Evolutionary Many-Objective Optimisation: Pushing the Boundaries

Miqing Li

for the degree of Doctor of Philosophy

of the

Department of Computer Science

Brunel University London

December 2015

Declaration

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

To my wife, Su.

Abstract

Many-objective optimisation poses great challenges to evolutionary algorithms. To start with, the ineffectiveness of the Pareto dominance relation, which is the most important criterion in multi-objective optimisation, results in the underperformance of traditional Pareto-based algorithms. Also, the aggravation of the conflict between proximity and diversity, along with increasing time or space requirement as well as parameter sensitivity, has become key barriers to the design of effective many-objective optimisation algorithms. Furthermore, the infeasibility of solutions' direct observation can lead to serious difficulties in algorithms' performance investigation and comparison.

In this thesis, we address these challenges, aiming to make evolutionary algorithms as effective in many-objective optimisation as in two- or three-objective optimisation. First, we significantly enhance Pareto-based algorithms to make them suitable for many-objective optimisation by placing individuals with poor proximity into crowded regions so that these individuals can have a better chance to be eliminated. Second, we propose a grid-based evolutionary algorithm which explores the potential of the grid to deal with many-objective optimisation problems. Third, we present a bi-goal evolution framework that converts many objectives of a given problem into two objectives regarding proximity and diversity, thus creating an optimisation problem in which the objectives are the goals of the search process itself. Fourth, we propose a comprehensive performance indicator to compare evolutionary algorithms in optimisation problems with various Pareto front shapes and any objective dimensionality. Finally, we construct a test problem to aid the visual investigation of evolutionary search, with its Pareto optimal solutions in a two-dimensional decision space having similar distribution to their images in a higher-dimensional objective space.

The work reported in this thesis is the outcome of innovative attempts at addressing some of the most challenging problems in evolutionary many-objective optimisation. This research has not only made some of the existing approaches, such as Pareto-based or grid-based algorithms that were traditionally regarded as unsuitable, now effective for many-objective optimisation, but also pushed other important boundaries with novel ideas including bi-goal evolution, a comprehensive performance indicator

and a test problem for visual investigation. All the proposed algorithms have been systematically evaluated against existing state of the arts, and some of these algorithms have already been taken up by researchers and practitioners in the field.

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisors Prof. Xiaohui Liu and Prof. Shengxiang Yang for their great, persistent support and help on both my study and life during the four years. It is impossible to imagine better mentors for my PhD stage. Prof. Xiaohui Liu is very kind and always encourages, guides and helps me whenever and wherever I face difficulties. His kindness, encouragement, inspiration, and enthusiasm are invaluable during my whole PhD life. Prof. Shengxiang Yang is very responsible and strives for excellence. He has always made time to discuss my research and provided suggestions and hints to whatever problems I met. I was profoundly moved more than once by his rigorous scholarship.

I am truly grateful to my enlightenment teacher Prof. Jinhua Zheng, who has got me interested in scientific research, led me into the field of evolutionary computation, and given me continued support over these years.

Many thanks go to the people with whom I co-authored a number of papers during the period of my PhD study, namely Prof. Robert M. Hierons, Dr. Wei Zheng, Dr. Sergio Segura, Zhaomeng Zhu, Dr. Ke Li, Prof. Qingfu Zhang, Dr. Veronica Vinciotti, Prof. Jinhua Zheng, Dr. Ruimin Shen, and Kang Wang.

In addition, I would like to express my gratitude to the following people for useful discussions, suggestions, comments, and supports of my research during the PhD stage: Dr. Ruimin Shen, Prof. Zidong Wang, Liang Hu, Dr. Crina Grosan, Prof. Qingfu Zhang, Ovidiu Pârvu, Kang Wang, and Zuofeng Zhang. I am also grateful to the following people or groups for providing their experimental data or opening their source codes for my research: Prof. Hisao Ishibuchi, Dr. Markus Wagner, Prof. Gary G. Yen, Prof. Kalyanmoy Deb, Prof. Qingfu Zhang, Dr. Tsung-Che Chiang, Dr. Evan J. Hughes, PISA, Jmetal, and OTL.

Further thank-yous are offered to my colleagues and friends from the Centre for Intelligent Data Analysis for the pleasant and enjoyable working atmosphere: Liang Hu, Chuang Wang, Dr. Djibril Kaba, Dr. Valeria Bo, Neda Trifonova, Izaz Rahman, Mohsina Ferdous, Dr. Emma Haddi, Dr. Zujian Wu, Dr. Yuanxi Li, Dr. Haitao Duan, Dr. Qian Gao, Dr. Ali Tarhini, Dr. Ana Salazar-Gonzalez. In addition, a special thank-

you goes to Ela Heaney for her kind help with everything at Brunel.

I would like to express my warmest thanks to my parents and my wife. My parents have always respected my choices and given me unconditional love, support and encouragement through all my life. My wife Su Guo, who had to go through difficult time to help me fulfil my dream, is so kind, patient, loving and caring and has provided tremendous support for my graduate study.

Finally, I would like to thank the Department of Computer Science, Brunel University London for funding my four-year PhD research.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ASF	achievement scalarizing function
CM	convergence measure
DM	decision-maker
DTLZ	Deb, Thiele, Laumanns and Zitzler test suite
EaMO	evolutionary many-objective optimisation
EMO	evolutionary multi-objective optimisation
GCD	grid crowding distance
GCPD	grid coordinate point distance
GR	grid ranking
HV	hypervolume indicator
IGD	inverted generational distance
MCDM	multi-criteria decision making
MOP	multi-objective optimisation problem
MaOP	many-objective optimisation problem
PCA	principle component analysis
PCI	performance comparison indicator
ROI	region of interest
SBX	simulated binary crossover
SOP	single-objective optimisation problem
TSP	travelling salesman problem
UF	unconstrained function
WFG	walking fish group test suite
ZDT	Zitzler, Deb and Thiele test suite

Algorithms

AGE-II	approximation-guided evolutionary algorithm
AR	average ranking
AR+Grid	average ranking combined with grid
BiGE	bi-goal evolution
DMO	diversity management operator
EA	evolutionary algorithm
ϵ-MOEA	ϵ-dominance based MOEA
FD-NSGA-II	fuzzy dominance-based NSGA-II
GrEA	grid-based evolutionary algorithm
HypE	hypervolume estimation algorithm
IBEA	indicator based evolutionary algorithm
MOEA	multi-objective evolutionary algorithm
MOEA/D	multi-objective evolutionary algorithm based on decomposition
MOEA/D+PBI	MOEA/D with penalty boundary intersection function
MOEA/D+TCH	MOEA/D with Tchebycheff function
MSOPS	multiple single objective Pareto sampling
NSGA-II	non-dominated sorting genetic algorithm II
NSGA-II+SDE	NSGA-II with shift-based density estimation
NSGA-III	non-dominated sorting genetic algorithm III
PESA-II	Pareto envelope-based selection algorithm II
PESA-II+SDE	PESA-II with shift-based density estimation
POGA	preference order based genetic algorithm
SDE	shift-based density estimation
SMS-EMOA	$\mathcal{S}$-metric selection based EMO algorithm
SPEA2	strength Pareto evolutionary algorithm 2
SPEA2+SDE	SPEA2 with shift-based density estimation

Symbols

div	number of divisions in the grid
f_i	the ith objective value
G_i	grid coordinate in the ith objective
m	number of objectives
n	number of decision variables
N	population size
P	evolutionary population (solution set)
p_c	crossover probability
p_m	mutation probability
Q	archive set
$\mathbb{R}^n$	field of real numbers
TSP_{cp}	correlation parameter in TSP
x_i	the ith decision variable
η_c	distribution index in SBX crossover
η_m	distribution index in polynomial mutation
$\prec$	to Pareto dominate

Chapter 1

Introduction

An individual would like to maximise the chance of being healthy and wealthy while still having fun and time for family and friends. A software engineer would be interested in finding the cheapest test suite while achieving full coverage (e.g., statement coverage, branch coverage and decision coverage). When prescribing radiotherapy to a cancer patient, a doctor would have to balance the attack on tumour, potential impact on healthy organs, and the overall condition of the patient. These multi-objective optimisation problems (MOPs) can be seen in various fields, sharing the same issue of pursuing several, often conflicting, objectives at the same time.

In MOPs, due to the conflicting nature of objectives, there is usually no single optimal solution but rather a set of alternative solutions, known as Pareto optimal solutions. These solutions are optimal in the sense that there are no other solutions in the search space that are superior for all objectives considered.

Evolutionary algorithms (EAs) are a class of stochastic optimisation methods that simulate the process of natural evolution. EAs have been recognised to be well suited for MOPs due to its characteristics of 1) low requirements on the problem properties, 2) being capable of handling large and highly complex search spaces, and particularly 3) population-based property which can search for a set of solutions in a single optimisation run, each representing a particular performance trade-off amongst the objectives.

As a subcategory of MOPs, many-objective optimisation problems (MaOPs) refer to an optimisation problem having four or more objectives. Over the last decade, many-

objective optimisation has been gaining rapidly increasing attention in the evolutionary computation community, driven by a wide variety of real-world applications (see [168]).

However, there exists great difference between many-objective optimisation and two- or three-objective optimisation. Major challenges lie in the way of the use of EAs to deal with MaOPs. In this thesis, we present a number of approaches to address these challenges, paving the way for the effective use of EAs in many-objective optimisation.

In this chapter, we first explain the motivation that led to the undertaking of this research, then we outline the main contributions of our work and the overall structure of this thesis, and finally we detail the publications which have resulted from the thesis.

1.1 Motivation

Since the early 90s, evolutionary multi-objective optimisation (EMO) algorithms[1] have demonstrated their effectiveness in solving various two- or three-objective optimisation problems. However, in practice, it is not uncommon to face an optimisation problem with four or more objectives (sometimes up to 10 or 15 objectives). Thus, it is not surprising that handling MaOPs has been one of the main research activities in the EMO area during the past few years.

Many-objective optimisation poses a number of challenges to EMO algorithms. Most notably, the Pareto dominance relation, which is the most important criterion in multi-objective optimisation, loses its effectiveness to differentiate individuals (solutions) in many-objective optimisation [74, 50, 67]. This makes EMO algorithms that work under the principle of Pareto dominance fail to provide selection pressure towards the Pareto front (i.e., Pareto optimal solutions in the objective space) [216, 256, 128]. In these algorithms, the density-based selection criterion will play a leading role in determining the survival of individuals in the evolutionary process, thus resulting in the final individuals distributed widely over the objective space but distant from the desired Pareto front. In fact, some studies have shown that a random search algorithm may even achieve better results than Pareto-based algorithms in problems with around 10 objectives [216, 152, 164].

[1] EAs which are used in multi-objective optimisation are usually called EMO algorithms.

In multi-objective optimisation, an EMO algorithm pursues two basic but often conflicting goals, proximity (also called convergence in some literature) and diversity. Such conflict has a detrimental impact on an algorithm's optimisation process and is particularly aggravated in many-objective optimisation [216, 1]. The algorithms capable of achieving a good balance between proximity and diversity in two- or three-objective problems could easily fail in many-objective optimisation [256, 138, 88]. In addition, a high objective dimensionality can also give rise to difficulty for the crowding evaluation [162, 52], parameter settings [216, 88, 180], and data structure used in EMO algorithms [45, 256]. All of these bring a big challenge for the design of new algorithms in many-objective optimisation.

Performance assessment is an important issue in evolutionary many-objective optimisation. However, many performance indicators, which are designed in principle for any number of objectives, are invalid or infeasible in practice to be used in many-objective optimisation [174, 254]. For example, indicators which is based on Pareto dominance relation typically return undifferentiated results of two solution sets with high dimensions. Indicators whose time or space requirement exponentially increases with the number of objectives may not be suitable for many-objective optimisation. Indicators which require a substitution of the Pareto front (as a reference set for comparison) could become inaccurate since it is very hard to properly represent a high-dimensional Pareto front.

Finally, a visual observation of the population becomes difficult in many-objective optimisation. Even though some effort has been made along this line (see [259, 249]), there is still a lack of simple, intuitive way to visualise solutions in the objective space with four or more objectives. This directly affects the algorithm analysis and investigation and also subsequent decision-making process.

Overall, the above challenges cause great difficulties in the use of classic Pareto-based algorithms in many-objective optimisation, in the design of new algorithms for many-objective problems, in the assessment of solution sets obtained by many-objective optimisers, and in the visualisation of solutions in a high-dimensional space. All of these suggest the pressing need of new methodologies for evolutionary many-objective

optimisation.

With these in mind, this thesis explores a number of innovative approaches to address these challenges in evolutionary many-objective optimisation, including a general enhancement of Pareto-based algorithms to make them suitable for many-objective optimisation (Chapter 3), an evolutionary algorithm which exploits the potential of the grid in many-objective optimisation (Chapter 4), an optimisation framework for many-objective problems (Chapter 5), a performance indicator for assessing solution sets with any number of objectives (Chapter 6), and a test problem for visual investigation of high-dimensional evolutionary search (Chapter 7). In short, our aim is to make EAs be considered as an effective tool in many-objective optimisation as in low-dimensional multi-objective (i.e., two- or three-objective) optimisation.

1.2 Contributions

The main contributions of the thesis are listed as follows.

- We significantly enhance Pareto-based algorithms by introducing a shift-based density estimation (SDE) to make them suitable for many-objective problems (Chapter 3). Unlike most of the current work which typically relaxes the Pareto dominance relation to make more individuals comparable, SDE works on the density estimation operation in Pareto-based algorithms. In view of the preference of density estimators for individuals in sparse regions, SDE "puts" individuals with poor proximity into crowded regions. This way, these poorly-converged individuals will be assigned a high density value, thus having a better chance to be eliminated in the diversity maintenance process of Pareto-based algorithms. The implementation of SDE is very simple, with negligible computational cost, and it can be applied to any Pareto-based algorithm without the need of additional parameters.

  The application of SDE in three popular Pareto-based algorithms has shown its high effectiveness in many-objective optimisation, especially when working with an accurate density estimator. Furthermore, from a comprehensive comparison

with five state-of-the-art algorithms, SDE has been demonstrated to be very competitive in finding a well-converged and well-distributed solution set on various many-objective optimisation problems with up to 10 objectives[2].

- We propose a novel algorithm, GrEA, to deal with many-objective optimisation problems (Chapter 4). GrEA explores the potential of the use of the grid in many-objective optimisation. Specifically, a set of grid-based criteria are introduced to guide the search towards the optimal front, and a grid-based fitness adjustment strategy is proposed to maintain an extensive and uniform distribution among individuals. In particular, to measure the crowding of individuals, GrEA considers the distribution of their neighbours in a set of hyperboxes whose size increases with the number of objectives, thus providing an accurate evaluation of individuals' crowding degree.

  From systematic experiments on 52 test instances with many objectives, GrEA has demonstrated its effectiveness in balancing proximity and diversity. Moreover, an appealing property of the algorithm is that its computational cost is almost independent on the number of hyperboxes in the grid and only increases linearly with the number of objectives. This is against the commonly accepted view [45] that grid-based approaches are not suitable in many-objective optimisation given their operation relying on the hyperboxes that exponentially grow in size with the number of objectives.

- We propose a bi-goal evolution (BiGE) framework for addressing many-objective optimisation problems (Chapter 5). Inspired by two observations: 1) the conflict between proximity and diversity is aggravated with the increase of objective dimensionality and 2) the Pareto dominance loses its effectiveness for a high-dimensional space but works well on a low-dimensional space, BiGE converts a given many-objective optimisation problem into a bi-goal (objective) optimisation problem regarding proximity and diversity, and then handles it using the Pareto dominance relation in this bi-goal domain.

[2]In several recent studies [167, 283], SDE has been found to be promising even for 20 objectives.

Systematic experiments are carried out to compare BiGE with five state-of-the-art algorithms on three groups of continuous and combinatorial benchmark suites with 5 to 15 objectives as well as on a real-world problem. In contrast to its competitors which work well on only a fraction of the test problems, BiGE can achieve a good balance between individuals' proximity and diversity on the problems with various properties.

- We propose a novel performance comparison indicator (PCI) to assess solution sets obtained by stochastic search algorithms in multi-objective optimisation (Chapter 6). In doing so, we also make a detailed analysis of the difficulties of popular performance indicators encountering in many-objective optimisation. PCI provides a comprehensive assessment of solution sets' proximity and diversity, and it can be used in problems with various Pareto front shapes and any objective dimensionality. In contrast to current state of the art, PCI is particularly practical in many-objective optimisation, given its characteristics of 1) no need for a specified reference set, 2) quadratic time complexity, 3) providing higher selection pressure than Pareto dominance but still being compliant with the latter, and 4) no requirement of parameter setting in the assessment.

- Finally, we construct a test problem (called Rectangle problem) to aid the visual investigation of multi-objective search (Chapter 7). Key features of the Rectangle problem are that the Pareto optimal solutions 1) lie in a rectangle in the two-variable decision space and 2) are similar to their images in the four-dimensional objective space (in the sense of Euclidean geometry). In this case, it is capable of visually examining the behaviour of objective vectors in terms of both proximity and diversity, by observing their closeness to the optimal rectangle and their distribution in the rectangle, respectively, in the decision space. Fifteen well-established algorithms are investigated on the Rectangle problem. Interestingly, most state-of-the-art algorithms (including those designed specially for many-objective optimisation) struggle on this relatively low-dimensional problem (having only 4 objectives). This indicates that the Rectangle problem can also be

used as a challenging benchmark function to test algorithms' ability in balancing proximity and diversity.

Altogether, these five contributions represent a significant advance in evolutionary many-objective optimisation, which should provide considerable help for researchers and practitioners in both algorithm development and problem solving. When designing a Pareto-based algorithm, researchers only need to focus on tackling two- or three-objective problems; for an optimisation problem with many objectives, SDE (Chapter 3) could be easily used. When working out a many-objective algorithm, the developer can use the Rectangle problem (Chapter 7) to investigate the behaviour of the algorithm or/and the PCI indicator (Chapter 6) to assess the performance of the algorithm. When dealing with a many-objective problem in hand, the user can directly adopt the algorithm GrEA (Chapter 4) or design new proximity and diversity estimation methods under the bi-goal evolution framework (Chapter 5).

1.3 Thesis Structure

This thesis is organised as follows.

Chapter 2 provides the necessary background material for the thesis. Beginning with the basic concepts in multi-objective optimisation, the chapter introduces key parts of EMO, including algorithm components, mainstream methods, test problems, and performance indicators. Then, general issues on many-objective optimisation are introduced, with a particular focus on the difficulties in many-objective optimisation and visualisation approaches in a high-dimensional space. Finally, various evolutionary many-objective optimisation techniques are described from different perspectives of addressing MaOPs.

Starting by the motivation of the work, Chapter 3 introduces a general enhancement of density estimation in Pareto-based algorithms to make them suitable for many-objective problems. SDE is integrated into three popular Pareto-based algorithms, NSGA-II, SPEA2 and PESA-II. Three groups of experiments are carried out to separately investigate 1) whether SDE improves the performance of all the three Pareto-

based algorithms, 2) among the density estimators in NSGA-II, SPEA2 and PESA-II which one is most suitable for SDE, and 3) how Pareto-based algorithms, when integrated with SDE, compare with other state-of-the-art algorithms designed specially for MaOPs.

Chapter 4 begins with the motivation of the use of grid in many-objective optimisation. Then, three grid-based fitness criteria are introduced, followed by their use in the mating selection and environmental selection processes. Finally, the performance of GrEA is empirically verified in sequence by the comparative study, parameter investigation, and algorithm analysis.

In Chapter 5, we propose a bi-goal evolution framework for many-objective problems. We first give the motivation of this work and then detail the BiGE framework and its implementation. This implementation includes four parts: proximity estimation, diversity estimation, mating selection, and environmental selection. Next, experimental results of BiGE in comparison with five best-in-class algorithms are shown, and finally a further investigation is provided to verify the proposed framework as well as its implementation.

The above three chapters present three approaches to deal with many-objective optimisation problems. In Chapter 6, we suggest a performance indicator to assess many-objective optimisation approaches. We first review related works in multi-objective optimisation and analyse their difficulties in many-objective optimisation. Then, we detail the proposed indicator. Finally, two classes of comparative studies are conducted to analytically and empirically verify PCI, respectively.

Chapter 7 focuses on another important issue in many-objective optimisation: visual investigation of multi-objective search. There, we present a test problem whose Pareto optimal solutions in the 2D search space have a similar distribution to their images in the 4D objective space. The proposed problem is tested by 15 EMO algorithms, with three instances of the problem to provide different challenges for these algorithms in balancing proximity and diversity.

In Chapter 8, we summarise the work presented in this thesis and look at how this has contributed to the field of evolutionary many-objective optimisation. Furthermore,

we suggest several directions of future research which have arisen during the course of this thesis.

1.4 Publications

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

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

  (Resulting from Chapter 5)

- M. Li, S. Yang and X. Liu. A performance comparison indicator for Pareto front approximations in many-objective optimization. In Proceedings of the 17th Annual Conference on Genetic and evolutionary computation (GECCO), 703–710, 2015, ACM.

  (Resulting from Chapter 6)

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

  (Resulting from Chapter 3)

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

  (Resulting from Chapters 2 and 6)

- M. Li, S. Yang, and X. Liu. A test problem for visual investigation of high-dimensional multi-objective search. In Proceedings of IEEE Congress on Evolutionary Computation (CEC), 2140–2147, 2014, IEEE. (*Best Student Paper Award*)

  (Resulting from Chapter 7)

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

  (Resulting from Chapter 4)

- M. Li, S. Yang, X. Liu, and R. Shen. A comparative study on evolutionary algorithms for many-objective optimization. In Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization (EMO), 261–275, 2013, Springer.

  (Resulting from Chapter 2)

In addition to the above, other papers, produced during the course of my PhD research, can be seen as indirect results of the research discussed in this thesis, as listed below. They have either focused on the algorithm development of evolutionary multi-objective optimisation or applied multi- or many-objective evolutionary approaches (including some presented in this thesis) to real-world problems.

- M. Li, S. Yang, and X. Liu. Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization. IEEE Transactions on Evolutionary Computation, 2015, in press.

- Z. Zhu, G. Zhang, M. Li, and X. Liu. Evolutionary multi-objective workflow scheduling in cloud. IEEE Transactions on Parallel and Distribution Systems, 2015, in press.

- W. Zheng, R. Hierons, M. Li, X. Liu, V. Vinciotti. Multi-objective optimisation for regression testing. Information Sciences, 2015, in press.

- M. Li, S. Yang, J. Zheng, and X. Liu. ETEA: A Euclidean minimum spanning tree-based evolutionary algorithm for multiobjective optimization. Evolutionary Computation, 22(2): 189–230, 2014.

- M. Li, S. Yang, K. Li, and X. Liu. Evolutionary algorithms with segment-based search for multiobjective optimization problems. IEEE Transactions on Cybernetics, 44(8): 1295–1313, 2014.

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

- M. Li, S. Yang, X. Liu, and K. Wang. IPESA-II: Improved Pareto envelope-based selection algorithm II. In Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization (EMO), 143–155, 2013, Springer.

- R. Hierons, M. Li, X. Liu, S. Segura, and W. Zheng. An improved method for optimal product selection from feature models. ACM Transactions on Software Engineering and Methodology, under review.

Chapter 2

Background

This chapter provides a review of evolutionary multi- and many-objective optimisation. The literature on this particular topic is vast and we will highlight the most relevant to this study.

This chapter is organised as follows. In Section 2.1, we present basic concepts in multi-objective optimisation, and this is followed by the description of key parts in EMO in Section 2.2. Section 2.3 introduces many-objective optimisation, with particular focuses on the difficulties of EMO algorithms in many-objective optimisation and the visualisation of solutions in a high-dimensional space. Finally, Section 2.4 provides a thorough review of evolutionary approaches for many-objective optimisation.

2.1 Basic Concepts

In general, a multi-objective optimisation problem (MOP) includes a set of n decision variables, a set of m objective functions, a set of J inequality constraints, and a set of K equality constraints. Without loss of generality, a minimisation MOP is defined as

the following form:

$$\begin{aligned} \text{Minimize} \quad & f_j(\mathbf{x}), j = 1, 2, ..., m \\ \text{Subject to} \quad & g_j(\mathbf{x}) \leq 0, j = 1, 2, ..., J \\ & h_k(\mathbf{x}) = 0, k = 1, 2, ..., K \\ & L_i \leq x_i \leq U_i, i = 1, 2, ..., n \end{aligned} \tag{2.1}$$

where $\mathbf{x}$ is a vector of n decision variables: $\mathbf{x} = (x_1, x_2, ..., x_n)$, $\mathbf{x} \in \mathbb{R}^n$. The last constraint set is called variable bounds, restricting each decision variable x_i within the range of $[L_i, U_i]$. In an MOP, feasible solutions (denoted as $\mathbf{x} \in \mathbb{R}^n_f$) refer to those solutions which satisfy all inequality and equality constraints. In the following, we introduce some underlying concepts in multi-objective optimisation.

Definition 2.1.1 (Pareto dominance)**.** *For two decision variables* $\mathbf{x}$ *and* $\mathbf{y}$*,* $\mathbf{x}$ *is said to Pareto dominate* $\mathbf{y}$ *(denoted as* $\mathbf{x} \prec \mathbf{y}$*), if and only if*

$$\begin{aligned} &\forall i \in (1, 2, ..., m) : f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \wedge \\ &\exists j \in (1, 2, ..., m) : f_j(\mathbf{x}) < f_j(\mathbf{y}) \end{aligned} \tag{2.2}$$

Pareto dominance reflects the weakest assumption about the preference of the decision maker; a solution is always preferable to another solution if the former dominates the latter. Accordingly, those solutions that are not dominated by any other solution are denoted as *Pareto optimal*. Pareto optimal solutions are characterised by the fact that improving in any one objective means worsening at least one other objective. The set of Pareto optimal solutions in the decision space is denoted as the *Pareto set*, and the corresponding set of objective vectors is denoted as the *Pareto front*. Next, we give the formal definition of these concepts.

Definition 2.1.2 (Pareto optimality)**.** *A solution* $\mathbf{x} \in \mathbb{R}^n_f$ *is said to be Pareto optimal if and only if* $\nexists \mathbf{y} \in \mathbb{R}^n_f$*,* $\mathbf{y} \prec \mathbf{x}$*.*

Definition 2.1.3 (Pareto set)**.** *The Pareto set (PS) is defined as the set of all Pareto optimal solutions, namely,* $PS = \{\mathbf{x} \in \mathbb{R}^n_f | \nexists \mathbf{y} \in \mathbb{R}^n_f, \mathbf{y} \prec \mathbf{x}\}$*.*

Definition 2.1.4 (Pareto front)**.** *The Pareto front (PF) is defined as the set of all objective vectors corresponding to the solutions in PS, namely,* $PF = \{(f_1(\mathbf{x}), ..., f_m(\mathbf{x})) : \mathbf{x} \in PS\}$.

Note that the size of the Pareto optimal solutions might be infinite and it is often infeasible to obtain the whole Pareto front. In practice, we want to obtain an approximation of the Pareto front that contains as much information as possible of the Pareto front, so the decision maker can either choose one element of the approximation as the final solution, or use this information to specify preferences that help search and find a satisfied solution.

2.2 Evolutionary Multi-Objective Optimisation

This section introduces some key parts in EMO, including algorithm components (Section 2.2.1), mainstream EMO methods (Section 2.2.2), performance indicators (Section 2.2.3), and test problems (Section 2.2.4).

2.2.1 Introduction

Evolutionary algorithms (EAs) stand for a class of stochastic search and optimisation methods that mimic the process of natural evolution. Over the past two decades, there has been significant interest in the use of EAs to solve MOPs (usually called evolutionary multi-objective optimisation (EMO) algorithms or multi-objective optimisation algorithms (MOEAs)), with success in fields as diverse as engineering, physics, chemistry, biology, economics, marketing, operations research, and social sciences [50, 41, 40, 285, 209]. This can be attributed to two major advantages of EAs. One is that they have low requirements on the problem characteristics and are capable of handling large and highly complex search spaces. The other is that their population-based search can achieve an approximation of the problem's Pareto front, with each solution representing a particular trade-off amongst the objectives.

The goal of approximating the Pareto front is itself multi-objective. In general, an EMO algorithm, in the absence of any further information provided by the decision

maker, pursues two ultimate goals with respect to its solution set – minimising the distance to the Pareto front (i.e., proximity or convergence) and maximising the distribution over the Pareto front (i.e., diversity). These two goals run through the design of all components of an EMO algorithm.

In the following, we briefly introduce several components of an EMO algorithm which are closely related to this thesis: fitness assignment, diversity maintenance, and selection; of course, other components are also crucial, such as variation, population initialisation, and stop criterion of an evolutionary algorithm.

Fitness Assignment

In contrast to single-objective optimisation in which the objective function and fitness function are typically identical, fitness assignment in multi-objective optimisation must allow for several goals. Many EMO algorithms design the fitness function on the basis of the Pareto dominance relation in the sense that the information of individuals dominating, being dominated or nondominated is used to define a rank, such as dominance count, dominance rank, strength, and others [50, 19, 43]. In addition, since Pareto dominance fails to reflect the diversity of individuals in a population, density information is also recognised as an auxiliary consideration to incorporate into the fitness function. This enables the population to evolve towards the optimum and simultaneously to diversify its individuals uniformly along the trade-off front.

Recently, there has been increasing interest in the use of other criteria (instead of Pareto dominance) in fitness assignment, with the aggregation-based criterion and indicator-based criterion being important examples. Typically, they convert an objective vector into a scalar value, thus providing a totally-ordered set of individuals in the population. Compared with Pareto dominance, such criteria have clear advantages, e.g., providing higher selection pressure towards the Pareto front [17, 138] and being easier to work with local search techniques stemming from global optimisation [169, 13].

Diversity Maintenance

Most EMO algorithms try to maintain diversity by incorporating density information into the selection process (often for nondominated individuals): the higher the

density of the surrounding area of an individual in the population, the lower the chance of the individual being selected. At the beginning, the *niching* technique, introduced by Goldberg [86], has been used to estimate the crowding degree of individuals in a population. With the development of the area, more density estimation methods, along with new EMO algorithms, have been presented. Among them, the cluster [294], crowding distance [55, 205], k-th nearest neighbour [292, 62], grid crowding degree [44, 150, 273, 181], and harmonic crowding distance [266] are representative examples.

On the other hand, some EMO algorithms maintain diversity of a population from other perspectives, for example, by integrating proximity and diversity into a single criterion [291] or by decomposing an MOP into a number of scalar optimisation problems with a set of well-distributed weight vectors [278]. These methods have been found to be very effective on some MOPs [17, 278]. However, since they are not completely in line with individuals' density in the population, their performance may be dependant on the shape of the Pareto front of an MOP at hand to some extent [179].

Selection

In the evolutionary process, selection represents the competition for resources among individuals. Some individuals better than others are naturally more likely to survive and to reproduce their genetic information. According to this principle, the overall selection operation in EMO algorithms can be split into two processes, mating selection (i.e., selection for variation) and environmental selection (i.e., selection for survival) [215]. Mating selection aims at picking promising individuals for variation and is usually performed in a random way. Environmental selection determines which of the previously stored individuals and the newly created ones are preserved in the archive (or the next population) and is usually performed in a deterministic way.

Interestingly, most EMO algorithms do not pay much attention to this difference and often directly perform the selection operation according to the fitness value of individuals for both selection processes. Nevertheless, there do exist some preliminary studies taking this into account, designing distinct strategies for mating selection and environmental selection [182, 271].

2.2.2 Methods

As mentioned before, in the absence of the information provided by the decision maker, EMO algorithms are designed with regard to two common goals, proximity and diversity. To achieve these two goals, however, different algorithms are implemented in distinct ways. In general, EMO algorithms, based on their selection mechanisms, can be classified into three groups – Pareto-based algorithms, decomposition-based algorithms, and indicator-based algorithms [39, 256].

Pareto-based Algorithms

Since the optimal outcome of an MOP is a set of Pareto optimal solutions, the Pareto dominance relation naturally becomes a criterion to distinguish between solutions during the evolutionary process of an algorithm. Behind such Pareto-based algorithms, the basic idea is to compare solutions according to their dominance relation and density. The former is considered as the primary selection and favours nondominated solutions over dominated ones, and the latter is used to maintain diversity and is activated when solutions are incomparable using the primary selection.

Most of the existing EMO algorithms belong to this group. Among them, several representative algorithms, such as the nondominated sorting genetic algorithm II (NSGA-II) [55], strength Pareto evolutionary algorithm 2 (SPEA2) [292], and Pareto envelope-based selection algorithm II (PESA-II) [44], are being widely applied to various problem domains [244, 266, 285, 287].

Decomposition-based Algorithms

In decomposition-based algorithms, the objectives of an MOP are aggregated by a scalarizing function such that a single scalar value is generated. In these algorithms, the diversity of a population is maintained by specifying a set of well-distributed reference points (or directions) to guide its individuals to search simultaneously towards different optima [200]. As the earliest multi-objective optimisation approach that can be traced back to the middle of the last century [161], this group has become popular again in recent years. One of the important reasons is due to the appearance of

an efficient algorithm, the decomposition-based multi-objective evolutionary algorithm (MOEA/D) [278, 170].

In decomposition-based EMO techniques, one important issue is to maintain uniformity of intersection points of the specified search directions to the problem's true Pareto front. Uniformly-distributed weight vectors cannot guarantee the uniformity of the intersection points. In fact, it is challenging for decomposition-based algorithms to access a set of the well-distributed intersection points for any MOP, especially for a problem having a highly irregular optimal front (e.g., a discontinuous or degenerate front). Despite the difficulty, much effort has been made on this issue recently [140, 82, 87, 83, 52, 218, 172].

Indicator-based Algorithms

The idea of indicator-based algorithms, which was first introduced by Zitzler and Künzli [291], is to utilise a performance indicator to guide the search during the evolutionary process. An interesting characteristic is that, in contrast to Pareto-based algorithms that compare individuals using two criteria (i.e., dominance relation and density), indicator-based algorithms adopt a single indicator to optimise a desired property of the evolutionary population.

The indicator-based evolutionary algorithm (IBEA) [291] is a pioneer in this group. Recently, a number of performance indicators, such as the ϵ indicator [165], inverted generational distance (IGD) [42], and $R2$ indicator [28], have been used in indicator-based algorithms [26, 227, 29], Of these, the hypervolume indicator [294] is a representative example. Due to the good theoretical and empirical properties [295, 24, 76], the hypervolume indicator has been frequently used to guide the search of an evolutionary population, such as in the S metric selection EMO algorithm (SMS-EMOA) [17] and multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) [112]. Whereas super-polynomial time complexity is required in the calculation of the hypervolume indicator (unless $P = NP$) [22], lots of effort is being made to reduce its computational cost, in terms of both the exact computation [16, 23, 267, 142] and the approximate estimation [10, 25, 129].

2.2.3 Performance Indicators

With the rapid development of EMO algorithms, the issue of performance assessment has become increasingly important and has developed into an independent research topic. During the past two decades, a variety of performance indicators have been emerging [75, 92, 295, 207, 153, 290, 272, 174]. They mainly concentrate on three aspects: 1) the proximity of the Pareto front approximation (i.e., the solution set obtained by a stochastic search algorithm, typically an EMO algorithm), 2) the uniformity of the approximation, and 3) the spread (i.e., extensity) of the approximation. The latter two are closely related, and in general, they are called the diversity of the approximation.

Table 2.1 lists some performance indicators and their properties, including the performance aspect(s) assessed by the indicators, the number of Pareto front approximations handled by the indicators, the computational cost needed by the indicators, and the state whether a reference set is required by the indicators or not. As can be seen from the table, some indicators only involve one aspect of the performance of Pareto front approximations, some focus on the diversity (i.e., both uniformity and extensity) of approximations, while the others give a comprehensive assessment of approximations' performance in terms of proximity, uniformity and extensity. Next, we briefly introduce two well-known comprehensive performance indicators, hypervolume [294] and IGD [42], which are used in the following chapters.

The hypervolume (HV) indicator calculates the volume of the objective space enclosed by a Pareto front approximation and a reference point. A large value is preferable. It can be described as the Lebesgue measure Λ of the union hypercubes h_i defined by a solution p_i in the approximation and the reference point x_{ref} as follows:

$$\mathrm{HV} = \Lambda(\{\bigcup_i h_i \mid p_i \in P\}) = \Lambda(\bigcup_{p_i \in P} \{x \mid p_i \prec x \prec x_{ref}\}) \tag{2.3}$$

The IGD indicator measures the average distance from the points in the Pareto front to their closest solution in a Pareto front approximation. Mathematically, let P^* be a reference set representing the Pareto front, then the IGD value from P^* to the

Table 2.1: Performance Indicators and their Properties

	Quality Indicators	Proximity	Uniformity	Spread	Number of Approximations	Computational Cost	Reference Set Needed
1	Generational Distance [252], GD_p [238]	√			unary	quadratic	√
2	ONVG, GNVG [237, 251, 253]	√			unary	linear time	
3	ONVGR, GNVGR [251, 253]	√			unary	linear time	√
4	Convergence Measure [53]	√			unary	quadratic	√
5	RNI [245]	√			unary	linear time	
6	Error Ratio [251]	√			unary	quadratic	√
7	$C1_R$, $C2_R$ [92, 48]	√			unary	quadratic	√
8	Coverage [293, 294]	√			binary	quadratic	
9	Dominance Ranking [153]	√			arbitrary	quadratic	
10	Purity [12]	√			arbitrary	quadratic	
11	Spacing [237, 51], Minimal Spacing [12]		√		unary	quadratic	
12	Uniform Distribution [245]		√		unary	quadratic	
13	Entropy Measure [66]		√		unary	linear time	
14	Cluster [269]		√		unary	exponential in m	
15	Uniformity Assessment [186]		√		unary	quadratic	
16	Maximum Spread [289, 85, 1]			√	unary	linear time	
17	Overall Pareto Spread [269]			√	unary	linear time	
18	Spread Assessment [183]			√	unary	exponential in m	
19	Δ Metric [55]		√	√	unary	quadratic	
20	Sigma Diversity Metric [199]		√	√	unary	linear time	
21	Diversity Measure [53]		√	√	unary	exponential in m	√
22	DCI [174]		√	√	arbitrary	quadratic	
23	Hypervolume [294]	√	√	√	unary/binary	exponential in m	
24	Hyperarea Ratio [251]	√	√	√	unary	exponential in m	√
25	Coverage Difference [288]	√	√	√	binary	exponential in m	
26	IGD [19, 42] IGD_p [238]	√	√	√	unary	quadratic	√
27	ϵ indicator [295]	√	√	√	unary/binary	quadratic	
28	G-Metric [189, 188]	√	√	√	arbitrary	quadratic	
29	Averaged Hausdorff Distance Δ_p [238]	√	√	√	unary	quadratic	√
30	IGD^+ [121, 123]	√	√	√	unary	quadratic	√

obtained solution set P is defined as follows:

$$\text{IGD} = \sum_{z \in P^*} d(z, P) \Big/ |P^*| \tag{2.4}$$

where $|P^*|$ denotes the size of P^* (i.e., the number of points in P^*) and $d(z, P)$ is the minimum Euclidean distance from point z to P. A low IGD value is preferable, which indicates that the obtained solution set is close to the Pareto front as well as having a good distribution.

Although both HV and IGD imply a combined performance of proximity and diveristy, there do exist some performance biases of the two indicators. IGD, which is based on uniformly-distributed points along the entire Pareto front, prefers the distribution uniformity of the solution set; HV, which is typically influenced more by the boundary solutions, has a bias towards the extensity of the solution set.

2.2.4 Test Problems

A number of test problems have been developed to benchmark the performance of EMO algorithms [122]. This section reviews six widely used benchmark suites. They are four continuous suites, ZDT [289], DTLZ [57], WFG [105], UF [281], and two combinatorial ones, multi-objective 0-1 knapsack [294] and multi-objective TSP [45].

The ZDT suite consists of six bi-objective test problems, with ZDT1, ZDT4 and ZDT5 having a convex Pareto front, ZDT2 and ZDT6 having a concave Pareto front, and ZDT3 having a disconnected Pareto front. All the problems are separable in the sense that the Pareto optimal set can be obtained by optimising each decision variable separately. ZDT4 and ZDT6 are multi-modal (i.e., have a number of local Pareto fronts) and ZDT6 also has a non-uniform mapping.

One important property of the DTLZ suite is that the test problems are scalable to any number of objectives and decision variables. All the seven problems are separable. DTLZ1 has a plane Pareto front but has a huge numer of local optima ($11^5 - 1$) in the objective space. Despite having the same optimal front, DTLZ2, DTLZ3 and DTLZ4 are designed to investigate different abilities of EMO algorithms. DTLZ2 is an easy

test problem with a spherical Pareto front. Based on DTLZ2, DTLZ3 introduces a vast number of local optima ($3^{10} - 1$) and DTLZ4 introduces a non-uniform mapping from the search space to the objective space. The Pareto front of DTLZ5 and DTLZ6 is a degenerate curve in order to test the ability of an algorithm to find a lower-dimensional optimal front while working with a higher-dimensional objective space. The difference between the two problems is that DTLZ6 is much harder than DTLZ5 by introducing bias in the g function [57]. DTLZ7 has a number of disconnected Pareto optimal regions, which is able to test an algorithm's ability to maintain sub-populations in disconnected portions of the objective space.

The WFG suite has nine test problems, which are also scalable in the number of objectives and decision variables. Compared with the ZDT and DTLZ suites, the WFG suite is more challenging, introducing more problem attributes, e.g., separability/non-separability, unimodality/multimodality, and unbiased/biased parameters. In WFG, a solution vector contains k position parameters and l distance parameters, and so the number of decision variables $n = k + l$. In contrast to ZDT and DTLZ, most WFG test problems are non-separable. This provides a big challenge for algorithms to achieve the Pareto front. Another property of the WFG problems is that they have dissimilar ranges of the Pareto front.

The UF suite has 10 test problems, among which UF1–UF7 have two objectives and UF8–UF10 have three objectives. Like the ZDT, DTLZ and WFG suites, UF uses component functions for defining its Pareto front as well as introducing various characteristics. A major advantage of UF over other test problems is that the Pareto set can be easily specified. In the UF problems, complex Pareto sets are used, with a strong linkage in variables among the Pareto optimal solutions. This poses a big challenge for EMO algorithms to search for the whole Pareto front.

The multi-objective 0-1 knapsack problem is one standard combinatorial problem in multi-objective optimisation. Given a set of l items and a set of m knapsacks, the

multi-objective knapsack problem can be defined as follows:

$$\begin{aligned} \text{Maximize } f_i(x) &= \sum_{j=1}^{l} p_{ij}x_j, \quad i = 1, ..., m \\ \text{Subject to } &\sum_{j=1}^{l} w_{ij}x_j \leq c_i, \quad i = 1, ..., m \\ & x = (x_1, ..., x_l)^T \in \{0, 1\}^l \end{aligned} \tag{2.5}$$

where $p_{ij} \geq 0$ is the profit of item j in knapsack i, $w_{ij} \geq 0$ is the weight of item j in knapsack i, c_i is the capacity of knapsack i, and $x_j = 1$ means that item j is selected in the knapsacks. Typically, p_{ij} and w_{ij} are set to random integers in the interval $[10, 100]$, and the knapsack capacity to half of the total weight regarding the corresponding knapsack.

The multi-objective travelling salesman problem (TSP) is also a typical combinatorial optimisation problem and can be stated as follows: given a network $L = (V, C)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of n nodes and $C = \{c_k : k \in \{1, 2, ..., m\}\}$ is a set of m cost matrices between nodes $(c_k : V \times V)$, we need to determine the Pareto optimal set of Hamiltonian cycles that minimise each of the m cost objectives. According to [45]. the matrix c_1 is first generated by assigning each distinct pair of nodes with a random number between 0 and 1. Then the matrix c_{k+1} is generated according to the matrix c_k:

$$c_{k+1}(i, j) = TSPcp \times c_k(i, j) + (1 - TSPcp) \times rand \tag{2.6}$$

where $c_k(i, j)$ denotes the cost from node i to node j in matrix c_k and $rand$ is a function to generate a uniform random number in $[0, 1]$. $TSPcp \in (-1, 1)$ is a simple TSP "correlation parameter", where $TSPcp < 0$, $TSPcp = 0$, and $TSPcp > 0$ introduce negative, zero, and positive inter-objective correlations, respectively.

2.3 Many-Objective Optimisation

This section introduces many-objective optimisation, with particular focus on the difficulties of evolutionary algorithms in many-objective optimisation (Section 2.3.2) and

the visualisation of solutions in a high-dimensional space (Section 2.3.3).

2.3.1 Introduction

Many-objective optimisation refers to the simultaneous optimisation of more than three objectives. Many-objective optimisation problems (MaOPs) appear widely in real-world applications, such as water resource engineering [224, 58, 146, 145], industrial scheduling problem [243], radar waveform optimisation problem [109], control system design [71, 100, 204], molecular design [159], space trajectory design [137], and software engineering [210, 236, 97, 196, 101, 208, 284]. Readers seeking more applications of many-objective optimisation can refer to a recent survey of evolutionary many-objective optimisation (EMaO) algorithms [168].

The term "many-objective optimisation" has been coined in 2002 [67], although some earlier work had realised a rapid increase of problems' difficulty with objective dimensionality [74, 50, 113, 91]. After that, EMO researchers did not pay much attention to many-objective optimisation, with the appearance of only a few studies for almost half a decade (2002–2006) [147, 215, 71, 107, 56]. Since 2007, there has been rapidly increasing interest in the use of evolutionary algorithms on MaOPs, as witnessed by a range of studies, including the algorithm analysis [216, 152] and design [108, 156, 229, 164, 212, 198, 135], empirical studies [256, 45, 127, 138], and review work [128, 110]. These enable many-objective optimisation to become one of the most active research topics in the EMO area. In recent years, some researchers have investigated particular topics of many-objective optimisation, such as the visualisation [259, 69, 249], test problems [235, 119, 115], performance indicators [132, 174, 6], theoretical analysis [248, 27, 239], experimental comparison [88, 180], and real-world applications (see [168]), while others have concentrated on developing new EMaO techniques (see Chapter 2.4 for details). Very recently, a couple of survey and review articles of EMaO algorithms have also emerged [254, 168].

2.3.2 Difficulties in Many-objective Optimisation

There exists great difference between many-objective optimisation and multi-objective optimisation with two or three objectives. Many difficulties lie in the way of the use of EAs in many-objective optimisation. Summarised from the literature [56, 216, 45, 128, 79, 52, 177], there are the following nine difficulties in evolutionary many-objective optimisation.

- **Incomparability of individuals on the basis of Pareto dominance.** As the number of objectives increases, the proportion of Pareto nondominated individuals in a population becomes large. The portion of any two individuals being comparable in an m-dimensional objective space is $\eta = 1/2^{m-1}$. For a two- or three-dimensional space, η is equal to 0.5 or 0.25, respectively, but when m reaches six, η is already as low as 0.03125. This means that only around three individuals are comparable in a population with randomly-produced 100 individuals.

- **Ineffectiveness of Pareto-based algorithms.** In Pareto-based algorithms, when the Pareto dominance criterion fails to distinguish between individuals, the density-based selection criterion will play a leading role in determining the survival of individuals. This phenomenon is termed *active diversity promotion* [216]. Some empirical observations [128, 256] indicated that the active diversity promotion has a detrimental impact on the algorithm's proximity due to its preference for *dominance resistant solutions* [113] (i.e., the solutions with an extremely poor value in at least one of the objectives, but with near optimal values in some others). As a result, the individuals in the final population in Pareto-based algorithms may be distributed uniformly over the objective space, but far away from the desired Pareto front. In fact, some studies have shown that a random search algorithm may even achieve better results than Pareto-based algorithms in problems with around 10 objectives [216, 152, 164].

- **Aggravation of the conflict between proximity and diversity.** In multi-objective optimisation, an EA pursues two basic but often conflicting goals, proximity and diversity. Such conflict has a detrimental impact on an algorithm's

optimisation process and is particularly aggravated in many-objective optimisation [216, 1, 178]. The algorithms capable of achieving a good balance between proximity and diversity in 2- or 3-objective problems could easily fail in many-objective optimisation.

- **Inaccuracy of density estimation.** In many-objective optimisation, some density estimation methods may fail to accurately reflect the crowding degree of individuals in a population. For example, the well-known density estimator *crowding distance* [55] works well only on bi-objective problems [163, 175]. The density estimators in some grid-based EMO algorithms (such as PAES [149], PESA-II [44] and DMOEA [273]), which consider the number of individuals in one hyperbox of the grid, may not accurately reflect individuals' crowding degree in many-objective optimisation. This is because the number of hyperboxes in the grid exponentially increases with the number of objectives and thus the individuals are most likely to disperse in different hyperboxes. In addition, the Euclidean distance, which is typically used in density estimators to measure similarity of individuals, can be inaccurate in a high-dimensional space [260].

- **Inefficiency of recombination operation.** In a high-dimensional space, individuals are likely to be widely distant from each other. In this case, the effect of recombination operation becomes questionable [216, 52]. Two distant parent individuals are likely to produce offspring that are also distant from parents. This can slows down the search process and lead to the inefficiency of overall EMO algorithms.

- **Increasing sensitivity of parameter settings.** Purshouse and Fleming [216] have shown that the *sweet-spot* of algorithm parameter setting that produces good results could shrink markedly in many-objective optimisation. Similar observations have also been reported in [88]. In addition, empirical studies in [180] have demonstrated that "many-objective" brings about severe instability for a classic algorithm, ϵ-MOEA [54]. For some 10-objective problems (such as DTLZ3 and WFG8), no matter how setting the parameter ϵ, the size of the final archive set

of ϵ-MOEA in several runs could be completely different – for some runs, the archive set has over 1600 individuals, yet for some other runs, the archive set has only one individual.

- **Rapid increase of time or space requirement.** The storage or time requirement of some EMO algorithms, such as PAES [149], PESA-II [44] and SMS-EMOA [17], increases exponentially with the number of objectives. This also applies to some performance indicators, like those based on the exact hypervolume calculation [294] or grid-centred calculation [53]. The exponential increase of the resource required can largely limit their applicability on many-objective optimisation.

- **Difficulty of representing the trade-off surface.** The Pareto front of an m-objective MOP is typically an $(m-1)$-dimension hyper-surface. The number of points required to accurately represent such a surface increases exponentially with the number of objectives. In addition, it is also difficult for the decision maker to consider such a large number of trade-off points.

- **Difficulty of visualisation.** Finally, a visual observation of the population becomes difficult when the number of objectives is larger than three. This is in contrast to two- or three-objective problems, where the population can be easily and comprehensively visualised using graphical ways.

Overall, these difficulties pose big challenges in the aspects of the design of EMaO algorithms, assessment of Pareto front approximations, and individual choice of the decision maker. Table 2.2 gives the effect scope of the difficulties with respect to the above three aspects.

2.3.3 Visualisation in Many-objective Optimisation

Visualisation is an important issue in multi-objective optimisation. It can be used in many respects [249]: 1) understanding the location, range and shape of Pareto front approximations, 2) assessing conflicts and trade-offs among objectives, 3) comparing the

Table 2.2: Difficulties in many-objective optimisation and their scope of the effect in EMO, including algorithm design, algorithm assessment (investigation) or/and multi-criteria decision making (MCDM).

Difficulties	Algorithm design	Algorithm assessment	MCDM
Incomparability on the basis of Pareto dominance	√	√	
Ineffectiveness of Pareto-based algorithms	√		
Conflict aggravation between proximity and diversity	√		
Inaccuracy of density estimation	√		
Inefficiency of recombination operation	√		
Increasing sensitivity of parameter settings	√	√	
Rapid increase of time or space requirement	√	√	
Difficulty of representation of the trade-off surface	√	√	√
Difficulty of visualisation		√	√

performance of Pareto front approximations, 4) helping to choose candidate solutions for the DM, and 5) monitoring the progress of an EMO algorithm.

However, as stated before, visualisation becomes difficult in many-objective optimisation. In contrast to two- or three-objective problems where it is straightforward to show objective vectors by scatter plots, for problems with four or more objectives one cannot intuitively observe the behaviour of objective vectors, such as how they evolve, how they are distributed, and how close they are to the Pareto front.

Effort has been made to ease this difficulty (see [259, 249], two comprehensive reviews of multi-objective visualisation). In general, there exist two classes of methods to visualise a set of vectors in the objective space. One, stemming from the multiple criterion decision-making (MCDM) community, is on the direct display by using a plane plot in the sense that objective vectors are displayed with no modifications, such as scatter plot matrix [192], parallel coordinates [114], star coordinates [50], trade-off plot [187], radial coordinate visualization [102], and heatmap [213]. These methods, however, often come without information about the Pareto dominance relation between vectors.

The other class is on the mapping of high-dimensional objective vectors to two- or three-dimensional ones for visualization. Key concerns under such mapping include the maintenance of the Pareto dominance relation between vectors and the reflection of their location information in the population. Many current studies originate from this motivation, presenting various interesting attempts, such as self-organizing map

[206], Sammon mapping [250], two-stage mapping [157, 69], multidimensional scaling [47], Isomap [247, 160], interactive decision maps [190], level diagrams [18], hyper-radial visualization [37], Pareto shells [257], seriated heatmaps [259], and prosection method [249]. However, inevitable information loss associated with the dimension reduction will influence the observation and understanding of objective vectors.

In addition, several methods of constructing and/or mapping some "key" vectors of a Pareto front approximation have been developed [183, 240, 64]. Despite failing to display all vectors completely, these methods can provide an outline of the whole Pareto front approximation, e.g., the range and location of the approximation in the space.

Finally, it is worth mentioning that some researchers have constructed (or introduced) a particular class of test problems to help the visual investigation of evolutionary search. Specifically, Köppen and Yoshida [156] presented a class of many-objective test problems whose Pareto optimal set is in a regular polygon on a two-dimensional decision space. This allows easy visualisation and examination of the proximity of the obtained solutions to the optimal region and their distribution in the decision space. Later on, Ishibuchi *et al.* [119, 130] extended and generalized this class of problems (called distance minimisation problems), introducing multiple Pareto optimal polygons with same [119] or different shapes [115] as well as making decision variables' dimensionality scalable [130]. Overall, these problems have provided a good alternative to help understand the behaviour of multi-objective evolutionary search, and have been used to investigate many-objective algorithms in some recent studies [241, 175, 180].

2.4 Evolutionary Approaches for Many-objective Optimisation

Since the pioneering studies at the beginning of the century [67], a wide variety of evolutionary approaches have been proposed to address MaOPs [45, 128, 132, 88, 254, 168]. Of these, some concentrate on the investigation or improvement of the existing approaches for two- or three-objective MOPs, while the others are dedicated to developing

new techniques specially for MaOPs. In general, these approaches could be classified into ten categories: 1) modified Pareto dominance criteria, 2) modified diversity maintenance operations, 3) decomposition-based algorithms, 4) aggregation-based methods, 5) indicator-based algorithms, 6) modified recombination operations, 7) new algorithm frameworks, 8) dimensionality reduction, 9) preference-based search, and 10) hybrid approaches.

2.4.1 Modified Pareto Dominance Criteria

A straightforward idea to resolve the difficulty of EMO algorithms in many-objective optimisation is to modify or enhance the Pareto dominance criterion. This can make more individuals comparable in a high-dimensional space, thus increasing the selection pressure towards the Pareto front. There are a large number of studies based on this idea (see [263, 168]). They can generally be divided into two groups: one is to change the dominance region, e.g., by controlling the dominance angle; the other is to consider the dominance relation on a part of objectives (rather than all objectives in the original Pareto dominance).

Modification of Dominance Region

ϵ-dominance, developed by Laumanns et al. [165], is one representative in this group, although not designed particularly for many-objective optimisation. There are two manners to implement ϵ-dominance, namely, additive and multiplicative. Given two solutions $\mathbf{x}, \mathbf{y} \in \mathbb{R}_f^n$ and $\epsilon > 0$, $\mathbf{x}$ is said to additively (multiplicatively) ϵ-dominate $\mathbf{y}$ if and only if $\forall i \in \{1, ..., m\}$, $f_i(\mathbf{x}) - \epsilon \leqslant f_i(\mathbf{y})$ ($f_i(\mathbf{x})(1-\epsilon) \leqslant f_i(\mathbf{y})$). It is clear that both additive and multiplicative ϵ-dominance relations enlarge the dominance region by ϵ. ϵ-dominance has been frequently used in many-objective optimisation, such as to replace the crowding distance in NSGA-II [156], to classify the entire population into a number of individual sets [3], and to control the size of archive set during the evolutionary process [255]. On the other hand, a classic ϵ-dominance based algorithm, ϵ-MOEA [54], has been found to work well on MaOPs [256, 88, 180, 176]. Dividing the objective space into many hyperboxes, ϵ-MOEA assigns each hyperbox at most a

single solution according to ϵ-dominance and the distance from solutions to the utopia point in the hyperbox. However, one major issue for ϵ-dominance based methods is how to determine a suitable ϵ value, especially when the number of objective is large [180]. Moreover, ϵ-dominance based EMO algorithms (like ϵ-MOEA) often fail to find boundary individuals of a population and also are somewhat sensitive to the shape of the Pareto front [99, 271, 176].

Ikeda et al. [113] proposed another relaxed form of Pareto dominance relation, called α-dominance. The basic idea is that a small deterioration in one objective can be compensated by a large improvement in other objectives. The α-dominance uses linear trade-off functions to define the tolerance of dominance. A similar idea was also presented by Branke [21], where the author explained it as a preference incorporation method.

Sato et al. [229] developed a preference relation to control the dominance area, called CDAS. According to a user-defined parameter, the dominance area can be expanded or contracted. However, it is not straightforward to specify a proper value of this parameter as the optimal setting varies, heavily depending on the problem in hand. Recently, some improvements of CDAS have been made, including making the parameter adaptive during the search [230, 203], controlling the number of crossed genes in many-objective optimisation [231], and applying the dominance area control to particle swarm optimisation [49].

In addition, Batista et al. [14] proposed another dominance relation, called Pareto cone ϵ-dominance as an improvement of ϵ-dominance. The cone ϵ-dominance can be seen as a combination of ϵ-dominance and α-dominance, and has been demonstrated to be able to overcome some weaknesses of ϵ-dominance [14]. Yet, again, a problem is on the choice of a proper ϵ value and the angle of the cone.

Consideration of a part of objectives

Drechsler et al. [61] proposed a preference relation, called *favour*, to differentiate nondominated solutions. When comparing two solutions, the favour relation considers the number of objectives for which one is better than the other. Later, the favour

relation has been improved by Sülflow et al. [243] by taking into account not only the number of objectives for which one solution is better than the other but also their quantitative difference on each objective.

The $(1-k)$-dominance, proposed by Farina and Amato [67, 68], can be seen as an extension of the favour relation. The $(1-k)$-dominance uses a parameter $k \in [0,1]$ to control the number of objectives for which one solution performs better than the other. When $k=0$, the $(1-k)$-dominance is actually the Pareto dominance; when $k=1$, the $(1-k)$-dominance is equal to the favour relation (including the case that two solutions perform equally in terms of the number of objectives).

Di Pierro et al. [59] presented a new preference relation, preference order ranking, for many-objective optimisation. Unlike the previous methods which only involve the comparison between two solutions, the preference order ranking considers the efficiency of the solutions compared with all other solutions in a population. A solution with "efficiency" of order k means that it is not Pareto dominated by any solutions in a population for any of $\binom{m}{k}$ objectives. A small order of efficiency is preferred. An experimental study has demonstrated its advantage over Pareto-based algorithms for MaOPs [59].

Köppen and Yoshida [155] defined a fuzzy Pareto dominance relation. The fuzzy Pareto dominance considers all the objectives of one solution being worse than its competitor, and fuses the magnitudes of these objectives into a single value. An advantage of the fuzzy Pareto dominance is that it can imply the crowding of solutions in the population, thereby providing a good coverage on some MaOPs [156].

Aguirre and Tanaka [4] proposed an approach to partition the objective space into several subspaces for MaOPs. In this approach, the environmental and mating selections are independently performed in each subspace to emphasise the search in smaller regions of the objective space. Later on, several enhancements have been made for this approach, including an introduction of the conflict information among objectives [131] and an adaptive set of the parameters in space partitioning [133]. These enhancements have been demonstrated to be effective on both combinatorial and continuous MaOPs [131, 133].

2.4.2 Modified Diversity Maintenance Operations

As stated before, when Pareto dominance fails to distinguish between solutions, diversity maintenance plays a decisive role and could lead the population away from the Pareto front. Therefore, another way of adapting Pareto-based algorithms for MaOPs is to modify their diversity maintenance strategy. However, interestingly, in contrast to the popularity of the modification of Pareto dominance, the modification of diversity maintenance operation has received little attention. There only exist a couple of studies from this modification so far.

Wagner et al. [256] demonstrated that assigning the crowding distance of boundary solutions a zero value in NSGA-II can clearly improve the performance in terms of proximity, despite the risk of losing diversity among solutions [126].

Adra and Fleming [1] employed a diversity management operator (DMO) to adjust the diversity requirement in the mating and environmental selection. By comparing the boundary values between the current population and the Pareto front, the diversity maintenance mechanism is controlled (i.e., activated or inactivated) during the evolutionary process. DMO has shown its advantage over NSGA-II in terms of proximity and diversity for DTLZ2 with up to 20 objectives.

Recently, inspired by the study in [2], Wang et al. [260] have shown the failure of using the Euclidean distance (L_2 norm) and Manhattan distance (L_1 norm) to calculate the similarity degree of solutions in a high-dimensional space, and then proposed a L_p-norm based distance to maintain diversity for MaOPs, where p is set to $1/m$ (m denotes the number of objectives).

2.4.3 Decomposition-based Algorithms

As the earliest multi-objective optimisation technique that can be traced back to the middle of the last century [161], the decomposition-based approach can be a good alternative in dealing with MaOPs. Instead of searching the entire search space for Pareto optimal solutions, decomposition-based algorithms decompose an MOP into a set of scalar optimisation subproblems by a set of weight vectors and an achievement scalar-

izing function (ASF). Commonly-used ASFs include weighted sum, Tchebycheff, vector angle distance scaling, and boundary intersection [195, 278, 111]. In the decomposition-based approach, since the optimal point associated with each search direction (weight vector) is targeted, sufficient selection pressure forwards can be provided and also a good distribution among solutions can be maintained in a high-dimensional space. According to the predefined multiple targets, the decomposition-based approach can be further divided into search-directions based and reference-points based algorithms [52].

Search-Direction based Algorithms

As one of the most well-known EMO algorithms developed recently, MOEA/D, proposed by Zhang and Li [278], deals with a set of scalar subproblems (by specifying multiple search directions) in a collaborative manner. Neighbourhood relations among these subproblems are defined on the basis of the similarity between their weight vectors. When optimising a subproblem, the information from its neighbouring subproblems is adopted. In MOEA/D, each subproblem keeps one individual in its memory, which could be the best individual found so far for it. For each subproblem, the algorithm generates a new individual by performing variation operators on some of its neighbouring individuals (i.e., the individuals of its neighbouring subproblems). The memory of both the considered subproblem and its neighbouring subproblems will be updated if the new individual is better than their current one. Despite not designed for MaOPs, MOEA/D, due to the potential in balancing proximity and diversity, leads to a number of subsequent works in many-objective optimisation.

Ishibuchi et al. [124] observed that MOEA/D with the weighted sum ASF is suitable for MaOPs with a convex Pareto front, while MOEA/D with the Tchebycheff ASF works well for a non-convex Pareto front. Based on these observations, the authors proposed to automatically alternate between weighted sum and Tchebycheff ASFs in MOEA/D. Later, Ishibuchi et al. [117] have also examined the relation between the neighbourhood size and the performance of MOEA/D, and found that MOEA/D with a large neighbourhood has a high search ability in the objective space, while a small neighbourhood is beneficial for diversity maintenance in the decision space.

Tan et al. [246] adopts a uniform design method to set weight vectors of the subproblems in MOEA/D. The experimental results have demonstrated that this method can improve the performance of MOEA/D and NSGA-II on most many-objective problems, including those with complicated Pareto set shapes.

Sato [228] improved the penalty boundary intersection (PBI) (called the inverted PBI) in MOEA/D to enhance the spread of solutions in the objective space and improve the search performance of the algorithm in MaOPs. The inverted PBI uses the parameter θ which is similar to the original PBI function, yet with $\theta = 0$ having the similar effect to the weighted sum function. That is, the inverted PBI involves the weighted sum achieving high search performance in MaOPs and also is able to emphasise the directivity for the search directions (determined by weight vectors) by increasing θ. The comparative experiments have shown the competitiveness of the inverted PBI on both continuous and combinatorial MaOPs.

Li et al. [171] integrated the Pareto dominance and niche-based diversity maintenance into MOEA/D (called MOEA/DD), with a hierarchical update of the population based on the order of Pareto dominance, local density estimation, scalarization functions. In MOEA/DD, each weight vector, apart from being associated with a subproblem, specifies a unique subregion in the objective space. Also, a steady-state selection scheme is adopted in the sense of updating the population once one offspring is produced.

Very recently, Yuan et al. [276] analysed the problem of the contour lines of the Tchebycheff function in diversity maintenance, and proposed to explicitly exploit the perpendicular distance from the solution to the weight vector in the objective space. The strategy is implemented in MOEA/D with the Tchebycheff function as well as in an ensemble fitness ranking method [274]. The experimental results have verified the effectiveness of the proposed strategy in balancing proximity and diversity.

On the other hand, researchers have also designed a range of decomposition-based algorithms specially for many-objective optimisation. Hughes [106] used multiple single Pareto sampling (MSOPS) to address MaOPs. In MSOPS, a set of T weight vectors are used to evaluate each solution (using a weighted min-max method), which is in

contrast to MOEA/D where a solution corresponds to only one weight vector. MSOPS has been found to perform better than NSGA-II in a couple of MaOPs [106]. Later on, Hughes [108] presented MSOPS-II with two extensions to MSOPS. The first extension is a method that uses the current population as input to generate a set of target vectors, and the second one is to reduce the time complexity of fitness assignment of the original algorithm. Recently, the author combined the aggregation method from MSOPS with the directed line search based on approximated local gradient [111]. The proposed algorithm has demonstrated its competitiveness on a constrained function with a concave Pareto front having up to 20 objectives.

Giagkiozis et al. [84] introduced the concept of generalised decomposition. Generalised decomposition provides a framework with which the DM can guide the search algorithm toward the Pareto front with the desired distribution of optimal solutions. This approach allows decomposition-based algorithms to focus on only the proximity to the Pareto front rather than all the three performance goals (proximity, uniformity and extensity). Combined with cross-entropy method, the proposed approach has shown to perform better than MOEA/D and RM-MEDA [280].

Reference-Point based Algorihtms

The reference-point based approach can be seen as another class of decomposition algorithms, which uses multiple predefined reference points (instead of search directions) to specify the search targets. This may be suitable for non-commensurable MOPs as the specified reference points can be well scaled by the current population.

NSGA-III [52] is a representative of such reference-point based decomposition algorithms. NSGA-III uses Pareto dominance to promote proximity and a niche operation to maintain diversity by a set of well-distributed reference points. Since poorly-converged solutions are often far away from the reference lines (i.e., the lines connecting the reference points with the original point), this niche operation implicitly prefers solutions with good proximity, thus leading to the algorithm better than traditional Pareto-based algorithms in a high-dimensional space. NSGA-III has been shown to be promising on various MaOPs [52, 139], including those with an irregular or a discon-

tinuous Pareto front on which the decomposition-based approach typically encounters difficulty [180].

Figueira et al. [70] presented a reference-point based approach, which can be partitioned into two phases, preparation phase and running phase. The preparation phase consists of estimating the boundary of the Pareto front, generating multiple reference points, and determining a solver's version for each reference point. The running phase launches a solver for each reference point in every processor.

Moen et al. [197] proposed a taxi-cab surface evolutionary algorithm (TC-SEA) for MaOPs. Manhattan distance is used as the selection criterion as well as the basis for generating the reference points, which leads to a fast, efficient algorithm. In comparison with several modern algorithms on problems with up to 20 dimensions, TC-SEA has shown its effectiveness as a many-objective optimiser [197].

Asafuddoula et al. [5] introduced an improved decomposition based evolutionary algorithm (I-DBEA) to deal with MaOPs. I-DBEA uses a steady-state evolution form, and a newly-generated solution is allowed to enter the population only if it is nondominated with respect to individuals in the population. Like MOEA/D-PBI [278], I-DBEA uses two distance measures, perpendicular distance (d_2) from the reference direction and distance (d_1) along the reference direction, in sequence, to compare individuals. But, this comparison is quite similar to the method in NSGA-III as only d_2 is activated in most cases. Later on, the authors used I-DBEA in many-objective robust design optimisation [7].

Yuan et al. [275] proposed a new dominance relation (θ-dominance) on the basis of decomposition-based approach. The authors aimed to enhance the proximity of NSGA-III but inherit its strength in terms of diversity maintenance. In θ-dominance, solutions are allocated into different niches represented by well-distributed reference points, and the solutions within the same niche have the competitive relationship. From an extensive comparison with eight state-of-the-art algorithms, the proposed approach has been demonstrated to be a promising alternative for many-objective optimisation [275].

In summary, the performance of the decomposition-based approach (including both

search-directions based and reference-points based) could be largely dependent on two key factors: the specification of the targeted directions and the association of solutions to targeted directions.

The specification of the search targets includes the selection of weight vectors [83, 52, 246, 108, 279, 277] and the selection of the ASF functions [124, 228]. ASF functions behave differently for different MaOPs. The weighted sum ASF is suitable for MaOPs with a convex Pareto front [124], the Tchebycheff ASF for many-objective 0-1 knapsack problems [118], and the boundary intersection ASF for the DTLZ suites [52].

The association of solutions to targeted directions directly determines the selection operation in the evolutionary process. Two distance measures, the distance along the targeted direction and the perpendicular distance from the targeted direction, are widely used in environmental selection via different comparison strategies [278, 52, 5, 275]. In contrast, mating selection has received relatively little attention, with only a couple of works being presented so far [35].

2.4.4 Aggregation-based Methods

The aggregation of objective functions (or individual rankings) is an alternative to differentiate individuals (in contrast to Pareto dominance) in many-objective optimisation[1]. A key issue in such aggregation techniques is how to maintain individuals' diversity because the information aggregation is unlikely to reflect their distribution in the population.

Average Ranking (AR), proposed by Bentley and Wakefield [15], is a popular individual comparison strategy in many-objective optimisation [45, 138]. AR compares all solutions in each objective and independently ranks them. The final ranking of a solution is obtained by summing its rankings of all objectives. However, due to the lack of a diversity maintenance scheme, AR may lead the evolutionary population to converge into a small part of the Pareto front [138, 180, 176]. Recently, some effort has been made to enhance diversity for AR in terms of the objective space [217, 184] and

[1] Decomposition-based technique is also an aggregation-based approach, but in this section we discuss the aggregation techniques which do not decompose an MOP into a number of subproblems to deal with.

decision space [154].

Kruisselbrink et al. [159] used desirability functions to combine groups of objectives in order to recast the original MaOPs into an optimisation problem with a low number of objectives. The purpose of this remodelling is to transform the original objectives so that a good view on the trade-off between the different objectives and the satisfaction of constraints is obtained in an evolutionary search method. The proposed technique was evaluated in a case study on automated drug design where the authors aim to find molecular structures that could serve as oestrogen receptor antagonists [159].

Garza-Fabre et al. [80] proposed a clustering-based elitist genetic algorithm (CEGA), which adopts an aggregation scheme to promote proximity and a clustering technique to maintain diversity. CEGA introduces an aggregation method, called the global detriment [79], which accumulates the inferiority of a solution to every other solution with respect to each objective in the population. Then, the clustering is applied to guide the selection in order to enhance the exploratory capabilities of the algorithm. The proposed algorithm has been well verified in the DTLZ test suite.

Recently, Cheng et al. [36] proposed a many-objective evolutionary algorithm based on directional diversity and favourable convergence, with the former being used in environmental selection and the latter in both mating and environmental selection. The favourable convergence is defined on the basis of the concept of favourable weight [242], and the directional diversity is defined by the projection lines from individuals to a hyperplane.

In fact, many EMaO techniques, despite not mentioning "aggregation" in their study, can fall into this category, since they somewhat use the aggregation of objective functions or individual rankings, such as ranking dominance [164], favour relation [243], winning score [191], correlation-based weighted sum [201], path control [225], and some distance-based ranking methods [198, 268, 296].

2.4.5 Indicator-based Algorithms

Indicator-based EMO algorithms utilise a performance indicator to optimise a desired property of the evolutionary population [291]. Due to good theoretical characteristics,

the indicator hypervolume (HV) [294] is widely used as a drive to guide the evolutionary population towards the Pareto front [17, 112, 10].

Emmerich et al. [63] proposed the S-metric Selection EMO algorithm (SMS-EMOA), a steady-state algorithm that aims to maximise the hypervolume. SMS-EMOA combines the maximisation of the hypervolume contribution with the non-dominated sorting in NSGA-II and only considers the hypervolume contribution of individuals when they are located in the "critical" layer. Although SMS-EMOA has shown good results for MOPs with up to six objectives [256], an exponentially increasing computational cost is required with the increase of the objective dimensionality in the algorithm.

HypE [10] is a hypervolume-based algorithm designed specially for many-objective optimisation. Unlike SMS-EMOA, HypE adopts Monte Carlo simulation to approximate the hypervolume value. This significantly reduces the time cost of the hypervolume calculation and enables the hypervolume-based search to be easily applied on many-objective optimisation, even when the number of objectives reaches 50.

Recently, some other performance indicators have also been demonstrated to be promising in guiding a well-converged and well-distributed population, such as the $R2$ [92, 29], IGD [227] and distance-based indicators [26, 255]. Diaz-Manriquez et al. [60] integrated the $R2$ indicator into a variant of the nondominated sorting procedure and presented that the obtained algorithm is very competitive with SMS-EMOA on the DTLZ suite with up to 10 objectives, but consumes much less computational time. Similar performance has been achieved by another $R2$-based EMaO algorithm proposed by Hernandez and Coello [98].

Bringmann et al. [26] proposed an approximation-guided evolutionary (AGE) algorithm, which aims at minimising a particular indicator (called the α indicator). Using the best knowledge obtained so far during the evolutionary process, AGE improves the approximation quality of the current population. AGE has been shown to outperform state-of-the-art algorithms in dealing with MaOPs. Despite good performance, AGE suffers from heavy computational cost as new incomparable solutions can unconditionally insert into AGE's archive. To tackle this issue, a fast, effective AGE (called AGE-II) has been developed by Wagner et al. [255]. AGE-II introduces an adaptive

ϵ-dominance approach to balance the convergence speed and runtime. Also, the mating selection strategy is elaborately designed to emphasise the diversity of the population.

2.4.6 Modified Recombination Operations

In contrast to the above methodologies which mainly focus on the environmental selection (i.e., selection for survival) in EAs, this category works on the recombination operation. Since individuals in a high-dimensional space are likely to be widely distant from each other, their offspring could be produced to be distant from them, thus seldom inheriting their "good" genes. Modifying recombination operation for many-objective optimisation can be divided into two types: 1) the use of mating restriction and 2) the design of new recombination operators.

Mating restriction can be naturally used in the decomposition-based approach in which two individuals from neighbouring search targets are participated in the recombination operation [278, 52, 35]. This mating restriction has been found to be useful in many-objective problems, by the evidence that increasing the size of a neighbourhood structure for recombination operation can deteriorate the performance of MOEA/D [125]. In addition, some researchers have shown that the recombination of similar individuals can improve the performance of Pareto-based algorithms [229, 158] and indicator-based algorithms [116] on many-objective 0/1 knapsack problems.

Despite receiving less attention, developing recombination operators (according to the characteristics of MOPs [173]) could be very effective in many-objective optimisation. Sato et al. [231] controlled the number of crossed genes (CCG) in crossover operators by using a user-defined parameter α. They have shown that a small number of crossed genes can improve the search performance in many-objective optimisation, and also the effectiveness of the CCG operator becomes significant with the increase of the objective dimensionality [231]. Later on, the authors developed an adaptive CCG to avoid parameter tuning and automatically find an appropriate α during the search process [232]. Simulation results have shown that the value of α can converge to an appropriate value even when the adaptation is started from any initial value.

2.4.7 New Algorithm Frameworks

To deal with MaOPs, researchers also designed new algorithm frameworks, including the coevolutionary of multiple populations [211, 81, 167, 260, 179] and multiple phases to separately focus on different performance (i.e., proximity and diversity) [94],

Garza-Fabre et al. [81] proposed a parallel genetic algorithm (PGA) for many-objective optimisation. In PGA, individuals are divided into multiple sub-populations which evolve in isolation most of the time. Sub-populations are evolved by means of an elitist genetic algorithm which is driven by a weighted sum ranking to promote proximity.

Wang et al. [260] presented a two-archive algorithm for MaOPs. The authors assigned different selection criteria in the two archives, with one archive being guided by the indicator $I_{epsilon+}$ (from IBEA [291]) and the other by the Pareto dominance criterion. The experimental results have shown that the presented algorithm is able to deal with MaOPs with satisfactory proximity, diversity and complexity.

He and Yen [94] proposed a two-stage strategy for many-objective optimisation. Unlike the decomposition-based algorithms stated before, the authors only considered the decomposition for the boundary points of an MOP, In their two-stage strategy, first, the whole population quickly approaches a small number of "targeted" points which form the range of the estimated Pareto front, and then, a diversity improvement strategy is applied to diversify individuals over the estimated Pareto front.

2.4.8 Dimensionality Reduction

In view of the fact that an MaOP (or after somehow converted) may have a low-dimensional Pareto front, dimensionality reduction (also named objective reduction) aims at removing redundant objectives of the original MaOP. Over the last decade, there are a range of objective reduction techniques in the context of evolutionary many-objective optimisation [234, 168].

Deb and Saxena [56] proposed a principal component analysis (PCA) based objective reduction method, which progresses iteratively from the interior of the search

space towards the optimal front by adaptively finding the correct lower-dimensional interactions. The proposed method was integrated into NSGA-II to deal with a class of MaOPs with a low-dimensional Pareto front (i.e., DTLZ5(I, m) [56]). Later, on the basis of this work, the authors [233] developed two new non-linear dimensionality reduction methods: one based on the correntropy PCA concept and the other on the maximum variance unfolding principle. Recently, Saxena et al. [234] presented a framework for both linear and nonlinear objective reduction algorithms which is built upon the above two methods (PCA and maximum variance unfolding principle). The performance of the proposed algorithms has been investigated on 30 test instances and two real-world problems.

Brockhoff and Zitzler [30] proposed a class of objective reduction techniques based on the dominance structure. Specifically, they presented the minimum objective subset problem (MOSS) asking which objective functions are essential, introduced a general notion of conflicts between objective sets, and developed an exact algorithm and a greedy heuristic for the NP hard MOSS problem. Later on, they allowed slight changes of the problem dominance structure to obtain a smaller minimum set of objectives [31]. Recently, Brockhoff and Zitzler [33] proposed an objective reduction methodology that allows both to consider black-box optimisation criteria and to maintain and control the dominance structure. This includes 1) an investigation on how adding or omitting objectives affects the problem characteristics, 2) a general notion of conflict between objective sets as a theoretical foundation for objective reduction, 3) exact and heuristic algorithms to systematically reduce the number of objectives, while preserving as much as possible the dominance structure of MaOPs, and 4) an experimental usefulness demonstration of the proposed methods in the context of both decision making and search.

Feature selection was used by Jaimes et al. [135] to design objective reduction methods. The authors presented two algorithms to reduce the objective dimensionality by identifying the most conflicting objectives. One determines the minimum subset of objectives that yields the minimum possible error and the other finds a subset of objectives of a given size that yields the minimum error. Later, Jaimes et al. [134] proposed two

schemes to integrate an objective reduction technique into an EMO algorithm (online objective reduction). One scheme periodically reduces the number of objectives during the search process, but for the last generation of the algorithm the original objectives are used. The other scheme alternately uses the objective reduction and the original objective set during the search process. Both schemes have been demonstrated to be promising to improve the proximity of EMO algorithms in many-objective optimisation. Recently, inspired by [134], Bandyopadhyay and Mukherjee [11] developed an objective reduction method, which periodically reorders the objectives on the basis of their conflict status and selects a subset of conflicting objectives for further processing.

Singh et al. [240] proposed a Pareto corner search evolutionary algorithm (PCSEA) for objective reduction in many-objective optimisation. Instead of seeking for the entire Pareto front, PCSEA searches for the boundary solutions first and then use them to identify the relevant objectives, given the fact that the solutions lying on the Pareto front's boundaries are diverse in terms of the range of the objective values and should aid in accurate estimation of actual dimensionality of the Pareto front.

Walker et al. [258] presented a rank-based dimension reduction for solution sets with high dimensions. The authors investigated objective selection methods which aim to preserve the average rank of individuals but with fewer objectives. Applying the objective selection process to the Times Good University Guide 2009 dataset has demonstrated that the two most significant objectives (i.e., contributing the most to the overall structure) are research quality and entry standards.

Recently, Wang and Yao [261] proposed an objective reduction method based on nonlinear correlation information entropy (NCIE). The NCIE matrix is used to measure the linear and nonlinear correlation between objectives, and a simple method is presented to select the most conflicting objectives during the evolutionary process. The proposed approach has been found to be well suited for Pareto-based algorithms but not for indicator-based ones.

2.4.9 Preference-based Search

In general, the Pareto front of an m-objective MOP is an $(m-1)$-dimensional front. To accurately represent this front having one higher dimension, exponentially more solutions are required. However, the decision maker (DM) is typically interested in only particular regions of the Pareto front. Therefore, designing an EMO algorithm which focuses on a subset of the Pareto front according to the DM's preference is a good alternative in many-objective optimisation.

There are two key issues in the preference-based approach: when to incorporate the preference information and how to model the DM preferences. According to when DM preferences are incorporated, EMO algorithms can be divided into three classes – *a priori*, *interactive* and *a posteriori*. An *a priori* decision making approach incorporates the DM preferences prior to the search process; an interactive decision making approach does so progressively during the optimisation process; and an *a posteriori* decision making approach does so after the search in the sense of selecting a preferred solution (by the DM) from an approximation of the whole Pareto front. According to the study of Coello et al. [43], methods for modelling DM preferences can be categorised into six forms: aspiration level (reference point), weight information, trade-off information, utility function, outranking, and fuzzy logic. All of these forms, coupled with three approaches of the DM preference incorporation, give rise to a wide variety of preference-based EMO algorithms (see [38, 20, 221, 263, 214]), including those focusing on many-objective optimisation [8, 219, 136].

In addition, it is worth mentioning that there exist another class of preference-based algorithms which aim to obtain a satisfactory approximation of the whole Pareto front. That is, the preference information is used to provide discrimination between solutions in a high-dimensional space rather than steer the search toward a particular region preferred by the DM. For example, Purshouse et al. [217] and Wang et al. [264, 265] proposed preference inspired co-evolutionary algorithms (PICEAs) based on a concept of co-evolving the common population of candidate solutions with a family of decision-maker preferences. Two realisations of PICEAs (PICEA-g and PICEA-w) are developed. PICEA-g co-evolves goal vectors with candidate solutions and PICEA-w

co-evolves weight vectors with candidate solutions. Compared with several modern algorithms, both of them have been found to be competitive in balancing proximity and diversity for many-objective optimisation. In addition, Zhang et al. [282] presented a knee point driven evolutionary algorithm for MaOPs. The basic idea behind this algorithm is that knee points are naturally most preferred among nondominated solutions if no explicit user preferences are given. Due to the use of multiple knee points to guide the population, the presented algorithm can maintain good distribution among individuals.

2.4.10 Hybrid Approaches

Combining several of the above approaches could be a promising way to deal with many-objective problems. This can make good use of their strengths and compensate for each other's weaknesses.

Brockhoff and Zitzler [32] combined objective reduction with the hypervolume-based EMO algorithms, saving a lot of computational cost of the hypervolume calculation. The authors developed a general approach on how objective reduction techniques can be incorporated into hypervolume-based algorithms and presented that the omission of objectives can improve the algorithms drastically in terms of the hypervolume results.

Li et al [167] proposed an improved two-archive algorithm for many-objective optimisation, with one archive guided by a decomposition-based approach for proximity and the other archive guided by a modified diversity maintenance approach for diversity. The efficiency of the proposed algorithm was demonstrated by experimental studies on the DTLZ test suite.

2.4.11 Summary

In short, the above approaches have led to a range of possible solutions to many-objective problems, some of which have already demonstrated early promise in some domains. However, great improvements are needed before evolutionary algorithms can be considered as an effective tool for many-objective problems as for two- or three-

objective problems. As demonstrated in some studies [217, 88, 180], modern EMaO algorithms can even struggle on some easy many-objective problems. This suggests a pressing need for new methodologies in the area.

Chapter 3

Shift-based Density Estimation

It is well known that Pareto-based algorithms encounter difficulties in dealing with MaOPs. In Pareto-based algorithms, the ineffectiveness of the Pareto dominance relation for a high-dimensional space leads to diversity maintenance mechanisms to play the leading role during the evolutionary process, while the preference of diversity maintenance mechanisms for individuals in sparse regions results in the final solutions distributed widely over the objective space but distant from the desired Pareto front. Intuitively, there are two ways to address this problem: 1) modifying the Pareto dominance relation and 2) modifying the diversity maintenance mechanism in the algorithm. In this chapter, we focus on the latter and propose a shift-based density estimation (SDE) strategy. The aim of our study is to develop a general enhancement of density estimation in order to make Pareto-based algorithms suitable for many-objective optimisation.

This chapter is organised as follows. In Section 3.1, we introduce the motivation of SDE. Section 3.2 details the SDE approach, and then Section 3.3 shows how SDE is integrated in three popular Pareto-based algorithms, NSGA-II [55], SPEA2 [292], and PESA-II [44]. Section 3.4 is devoted to experimental studies, which is followed by some discussions of SDE in Section 3.5. Finally, we summarise the work in Section 3.6.

3.1 Motivation

In a population, the density of an individual represents the crowding degree of the area where the individual is located. Due to the close relation with diversity maintenance, density estimation is very important in EAs and is widely applied in various optimisation scenarios, such as multimodal optimisation [193], dynamic optimisation [270], and robustness optimisation [143].

In multi-objective optimisation, usually there is no single optimal solution but rather a set of Pareto optimal solutions. Naturally, density estimation plays a fundamental role in the evolutionary process of multi-objective optimisation for an algorithm to obtain a representative and diverse approximation of the Pareto front [19, 166].

There are a wide range of density estimation techniques that have been developed in the EMO community. They act on different neighbours of an individual, involve different neighbourhoods, and consider different measures [148]. For example, the niched Pareto genetic algorithm (NPGA) considers the niche of an individual and measures the degree of crowding in the niche [103]. The strength Pareto evolutionary algorithm (SPEA) uses a clustering technique to estimate the crowding degree of an individual [294]. NSGA-II defines a new measure, "crowding distance", to reflect the density of an individual, only acting on the two closest neighbours located in either side for each objective. Most grid-based EMO approaches, such as PESA-II and the dynamic multi-objective EA (DMOEA) [273], estimate the density of an individual by counting the individuals in the hyperbox where it is located [181], whereas some recent grid-based approaches consider the crowding degree of a region constructed by a set of hyperboxes whose range varies with the number of objectives [184, 185]. SPEA2 considers the k-th nearest neighbour of an individual in the population [292]. Instead of using the Euclidean distance in SPEA2, Horoba and Neumann used the Tchebycheff distance to determine the k-th nearest neighbor [104]. In [181], a Euclidean minimum spanning tree (EMST) of individuals in a population is generated, and the density of an individual is estimated by its edges in the EMST. Farhang-Mehr and Azarm calculated the entropy in the population, estimating the density of an individual by considering the

influences coming from all other individuals in the population [65]. Also, the harmonic distance has been used to reflect the crowding degree of individuals in the population [266].

Despite the variety of density estimation techniques, they all measure the similarity degree among individuals in the population, i.e., they estimate the density of an individual by considering the mutual position relation between it and other individuals in the population. Formally, the density of an individual p in a population P can be expressed as follows:

$$D(p, P) = D(dist(p, q_1), dist(p, q_2), ..., dist(p, q_{N-1})) \tag{3.1}$$

where $q_i \in P$ and $q_i \neq p$, N is the size of P and $dist(p, q)$ is the similarity degree between individuals p and q, usually measured by their distance, e.g., Euclidean distance. $D()$ is the function of the similarity degree between the interested individual and other individuals in the population. The specific implementation of $D()$, as stated above, depends on the density estimator used in an EMO algorithm.

In Pareto-based algorithms, in general, when two individuals are nondominated ones in a population, the one with the lower density is preferable. This rule is very effective for an MOP with 2 or 3 objectives since it can provide a good balance between proximity and diversity. However, in many-objective optimisation, this rule may fail to guide the population to search towards the optimal direction.

As mentioned before, the proportion of nondominated individuals in the population becomes considerably large when a large number of objectives are involved. In the extreme case, all individuals in the population may become nondominated with each other, where the density of individuals will play a leading or even unique role in distinguishing between them in the selection process of algorithms. As a result, individuals that are distributed in sparse regions (i.e., individuals that have a low similarity degree to other individuals) will be preferred as long as they are nondominated in the population. However, it is likely that such individuals are located far away from the optimal front (e.g., they are slightly better than or comparable with other individuals in some

objectives but are significantly worse in at least one objective). For example, considering a population of four nondominated individuals **A**, **B**, **C**, and **D** with their objective value $(0, 1, 1, 100)$, $(1, 0, 2, 1)$, $(2, 1, 0, 1)$, and $(1, 2, 1, 0)$, individual **A** performs worst regarding proximity but is preferable in Pareto-based algorithms.

This density-leading criterion severely deteriorates the search performance of algorithms, which is reflected in both the mating and environmental selection. In the mating selection, there will be a higher probability that those poorly-converged nondominated individuals (such as individual **A** in the above example) are selected to recombine and produce low performance offspring. In the environmental selection, the long-term existence of those poorly-converged individuals will lead to the elimination of some well-converged ones due to the restriction of the population size. Consequently, the solutions, at the end of the optimisation process, may be distributed widely over the objective space, but far away from the desired Pareto front. Figure 3.1 plots the evolutionary trajectories of the proximity results[1] of the original NSGA-II and its modified version where the density estimation procedure is removed for the 10-objective DTLZ2 [57]. Evidently, with the evolution process, the original NSGA-II gradually draws the population away from the Pareto front, while replacing the crowding distance-based selection with random selection noticeably improves the proximity performance of the algorithm.

The above observations indicate that one reason for the failure of Pareto-based algorithms in many-objective optimisation is their dislike for individuals in crowded regions. Then, can we "put" those poorly-converged individuals into crowded regions? In this case, any density estimator can identify these poorly-converged individuals as long as it can correctly reflect the crowding degree of individuals. Keeping this in mind, we present a new general density estimation methodology – shift-based density estimation (SDE); to facilitate contrast, we abbreviate traditional density estimation as TDE.

[1] The results are evaluated by the convergence measure (CM) metric [53]. CM assesses the proximity of a solution set by calculating the average normalised Euclidean distance from the set to the Pareto front.

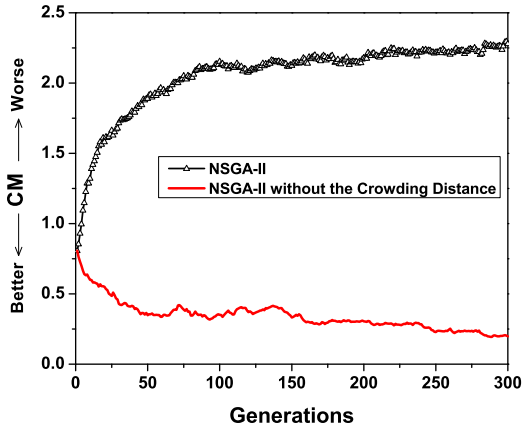


Figure 3.1: Evolutionary trajectories of the convergence metric (CM) for a run of the original NSGA-II and the modified NSGA-II without the density estimation procedure on the 10-objective DTLZ2.

3.2 The Proposed Approach

As stated previously, the density estimation of an individual in the population is based on the relative positions of other individuals with regard to the individual. In SDE, we adjust these positions, trying to reflect the proximity of the individual in the population.

When estimating the density of an individual p, SDE shifts the positions of other individuals in the population according to the proximity comparison between these individuals and p on each objective. More specifically, if an individual performs better[2] than p for an objective, it will be shifted to the same position of p on this objective; otherwise, it remains unchanged. Formally, without lose of generality, assuming that we consider a minimisation MOP, the new density $D'(p, P)$ of individual p in the population P can be expressed as follows:

$$D'(p, P) = D(dist(p, q'_1), dist(p, q'_2), ..., dist(p, q'_{N-1})) \tag{3.2}$$

where N denotes the size of P, $dist(p, q'_i)$ is the similarity degree between individuals p and q'_i, and q'_i is the shifted version of individual q_i ($q_i \in P$ and $q_i \neq p$), which is

[2]For minimisation MOPs, performing better means having a lower value; for maximisation MOPs, it means having a higher value.

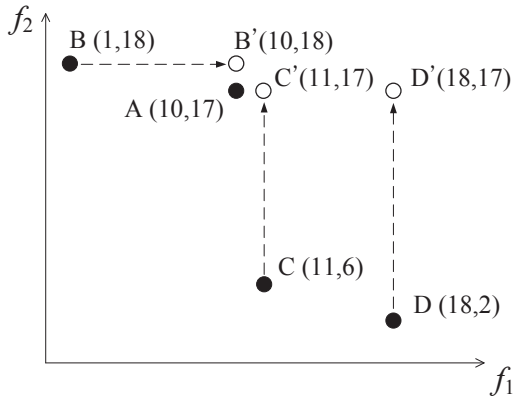


Figure 3.2: An illustration of shift-based density estimation in a bi-objective minimisation scenario. To estimate the density of individual **A**, individuals **B**, **C**, and **D** are shifted to $\mathbf{B}'$, $\mathbf{C}'$, and $\mathbf{D}'$, respectively.

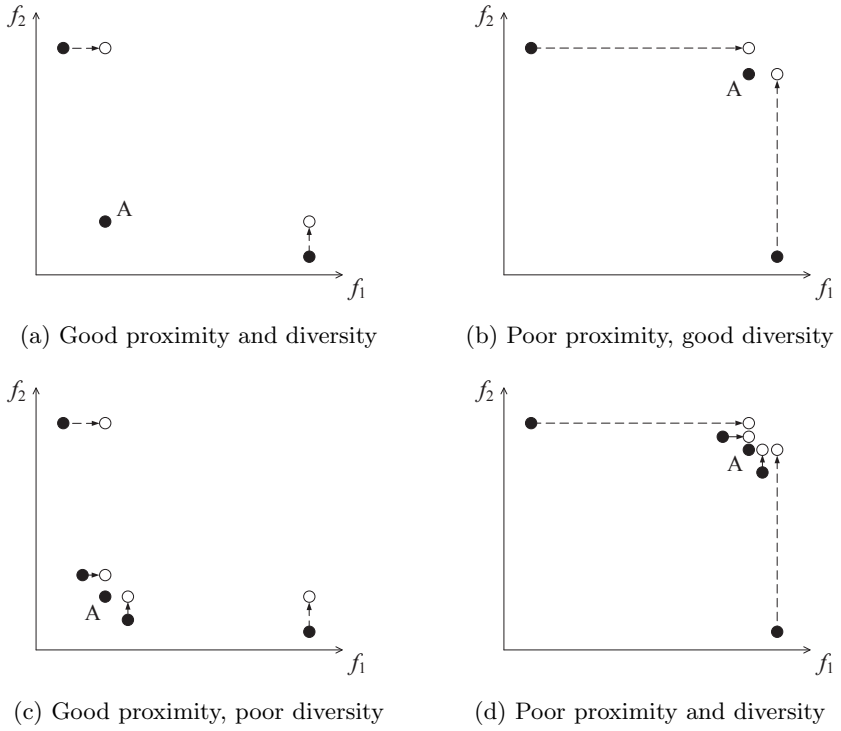
defined as follows:

$$q'_{i(j)} = \begin{cases} p_{(j)}, & \text{if } q_{i(j)} < p_{i(j)} \\ q_{i(j)}, & \text{otherwise} \end{cases}, \quad j \in (1, 2, ..., m) \tag{3.3}$$

where $p_{(j)}$, $q_{i(j)}$, and $q'_{i(j)}$ denote the j-th objective value of individuals p, q_i, and q'_i, respectively, and m denotes the number of objectives.

Figure 3.2 shows a bi-objective example to illustrate this shift-based density estimation operation. To estimate the density of individual **A** in a population composed of four nondominated individuals $\mathbf{A}(10, 17)$, $\mathbf{B}(1, 18)$, $\mathbf{C}(11, 6)$, and $\mathbf{D}(18, 2)$, **B** is shifted to $\mathbf{B}'(10, 18)$ since $\mathbf{B}_1 = 1 < \mathbf{A}_1 = 10$, and **C** and **D** are shifted to $\mathbf{C}'(11, 17)$ and $\mathbf{D}'(18, 17)$, respectively, since $\mathbf{C}_2 = 6 < \mathbf{A}_2 = 17$ and $\mathbf{D}_2 = 2 < \mathbf{A}_2 = 17$.

Clearly, individual **A**, which has a low similarity degree with other individuals in the original population, has two close neighbours in its new density estimation, and thus will be assigned a high density value. This occurs because there are two individuals **B** and **C** performing significantly better than **A** in terms of proximity (i.e., being slightly inferior to **A** in one or some objectives but greatly superior to **A** in the others). These individuals contribute large similarity degrees to **A** in its density estimation since the value on their advantageous objective(s) becomes equal to that of **A**. This means that



(a) Good proximity and diversity

(b) Poor proximity, good diversity

(c) Good proximity, poor diversity

(d) Poor proximity and diversity

Figure 3.3: Shift-based density estimation for four situations of an individual (**A**) in the population for a minimisation MOP.

the individuals which have no clear advantage over other individuals in the population will have a high density value in SDE.

In order to further understand SDE, we next consider four typical situations of the distribution of an individual in the population for a minimum MOP (i.e., performing well in proximity and diversity, performing well in diversity but poorly in proximity, performing well in proximity but poorly in diversity, and performing poorly in both proximity and diversity) in Figure 3.3.

As can be seen from Figure 3.3, only the individual with both good proximity and good diversity has a low crowding degree in SDE. The individual with either poor proximity or poor diversity has some close neighbours, and the individual with both poor proximity and poor diversity has the highest crowding degree in the four situations. In addition, note that the individuals with poor diversity (e.g., see Figure. 3.3(c) and Figure 3.3(d)) are always located in crowded regions no matter how well they perform

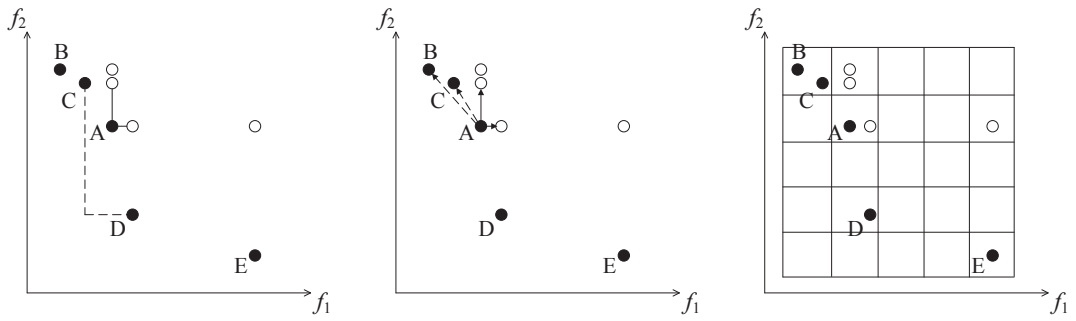
in terms of proximity, which means that SDE can maintain the distribution characteristic of individuals in the population while reflecting the proximity difference between individuals.

3.3 Integrating SDE into NSGA-II, SPEA2 and PESA-II

In this section, we apply SDE to three classical Pareto-based EMO algorithms: NSGA-II [55], SPEA2 [292] and PESA-II [44]. NSGA-II is known for its nondominated sorting and crowding distance-based fitness assignment strategies. SPEA2 defines a *strength* value for each individual, and combines it with the k-th nearest neighbor method to distinguish between individuals in the population. The main characteristic of PESA-II is its grid-based diversity maintenance mechanism, which is used in both the mating and environmental selection schemes. The density estimators (i.e., the crowding distance, k-th nearest neighbour and grid crowding degree) in the three algorithms are representative and are briefly described below.

To estimate the density of an individual in the population, NSGA-II considers its two closest points on either side along each objective. The crowding distance is defined as the average distance between the two points on each objective. The nearest neighbour technique used in SPEA2 takes the distance of an individual to its k-th nearest neighbor into account to estimate the density in its neighbourhood. This density estimator is used in both the fitness assignment and archive truncation procedures to maintain diversity. PESA-II uses an adaptive grid technique to define the neighbourhood of individuals. The density around an individual is estimated by the number of individuals in its hyperbox in the grid. Figure 3.4 illustrates the three density estimators used in TDE and SDE.

It is necessary to point out that since the crowding distance mechanism in NSGA-II separately estimates an individual's crowding degree on each objective, individuals may be overlapping on a single axis of the objective space in SDE. For example, when estimating the shift-based crowding degree of **A** on the f_1 axis in Figure 3.4(a), individuals **A**, **B**, and **C** are overlapping. Here, we keep the original order before individuals are



(a) Crowding Distance (b) k-th Nearest Neighbour (c) Grid Crowding Degree

Figure 3.4: An illustration of the three density estimators in traditional and shift-based density estimation, where individual **A** is to be estimated in the population.

shifted. That is, on the f_1 axis, individual **C** is still viewed as the left neighbour of **A**, and individual **D** is viewed as its right neighbour in the shift-based crowding distance calculation of **A**. In this case, the crowding distance of **A** in TDE (shown with a dashed line) is changed to the average distance between **A** and the shifted **C** and **D** in SDE (shown with a solid line).

In Figure 3.4(b), **C** and **B** are the two nearest neighbours of **A** in the original population (shown with a dashed arrow), but, to estimate the density of **A** in SDE, the two nearest individuals, the shifted **D** and **C**, are considered (shown with a solid arrow). Concerning Figure 3.4(c), there is no individual in the neighbourhood of **A** in TDE, but in SDE, the shifted **D** is the neighbour of **A**, thereby contributing to its grid crowding degree.

Overall, although the implementation of the three density estimators is totally different, the individuals (like individual **D** in Figure 3.4) which do not perform significantly worse than the considered individual will contribute a lot to its density estimation in SDE.

In the next section, we will empirically investigate the proposed method, trying to answer the following questions – Can SDE improve the performance of all the three Pareto-based algorithms? Among these three density estimators, which one is most suitable for SDE in many-objective optimisation? How would the Pareto-based algorithms, when integrated with SDE, compare with other state-of-the-art algorithms

Table 3.1: Properties of test problems and parameter setting in PESA-II, PESA-II+SDE, and ϵ-MOEA. The settings of *div* and ϵ correspond to the different numbers of objectives of a problem. m and n denote the number of objectives and decision variables, respectively

Problem	m	n	Properties	*div* in PESA-II	*div* in PESA-II+SDE	ϵ in ϵ-MOEA
DTLZ1	4, 6, 10	M+4	Linear, Multimodal	5, 40, 20	15, 12, 7	0.04, 0.054, 0.052
DTLZ2	4, 6, 10	M+9	Concave	5, 6, 7	11, 6, 4	0.105, 0.2, 0.275
DTLZ3	4, 6, 10	M+9	Concave, Multimodal	40, 40, 40	18, 16, 6	0.105, 0.2, 0.8
DTLZ4	4, 6, 10	M+9	Concave, Biased	6, 7, 10	13, 5, 4	0.105, 0.2, 0.275
DTLZ5	4, 6, 10	M+9	Concave, Degenerate	11, 7, 5	30, 20, 10	0.032, 0.11, 0.14
DTLZ6	4, 6, 10	M+9	Concave, Degenerate, Biased	9, 6, 6	23, 11, 5	0.095, 0.732, 1.48
DTLZ7	4, 6, 10	M+19	Mixed, Disconnected, Multimodal	7, 5, 3	13, 11, 5	0.09, 0.26, 0.73
TSP(–0.2)	4, 6, 10	30	Convex, Negative correlation	9, 6, 4	17, 9, 5	0.9, 1.9, 4.3
TSP(0)	4, 6, 10	30	Convex, Zero correlation	9, 5, 7	18, 10, 5	0.65, 1.3, 3.15
TSP(0.2)	4, 6, 10	30	Convex, Positive correlation	8, 5, 3	19, 10, 5	0.42, 0.85, 2.26

designed specially for many-objective problems?

3.4 Experimental Results

In this section, we first integrate SDE into the aforementioned three Pareto-based algorithms (which results in three new EMO algorithms, denoted NSGA-II+SDE, SPEA2+SDE, and PESA-II+SDE, respectively) and separately perform the comparison of NSGA-II vs NSGA-II+SDE, SPEA2 vs SPEA2+SDE, and PESA-II vs PESA-II+SDE. Thereafter, we compare three SDE-based algorithms and investigate the reason for their behaviour in many-objective optimisation. Finally, we test the competitiveness of SDE to existing state-of-the-art approaches by comparing SPEA2+SDE with five representative algorithms taken from different categories of solving MaOPs.

Two well-established test problem suites, DTLZ [57] and the multi-objective TSP [45], are used in this study (see Chapter 2.2.4 for details). In multi-objective TSP, $TSPcp$ is assigned to -0.2, 0 and 0.2 to represent different characteristics of the problem. The characteristics of all the tested problems are summarised in Table 3.1.

To compare the performance of the algorithms, the IGD [19, 278] and hypervolume (HV) indicators are considered. These two indicators have been introduced in Chapter 2.2.3 in detail. IGD is used to assess algorithms on the problems whose Pareto front are known (i.e., DTLZ) and HV on the problems whose Pareto front are unknown (i.e.,

the multi-objective TSP). In the calculation of HV, the determination of the reference point is a crucial issue. Choosing a reference point that is slightly larger than the worst value of each objective on the Pareto front has been found to be appropriate since the effects of proximity and diversity of the set can be well balanced [9, 119]. Since the range of the Pareto front is unknown in TSP, we regard the point with 22 for each objective (i.e., $r = 22^m$) as the reference point, given that it is slightly larger than the worst value of the mixed nondominated solution set constructed by all Pareto front approximation obtained. In addition, since the exact calculation of the HV metric is infeasible for a solution set with 10 objectives, we approximately estimate the HV result of a solution set by the Monte Carlo sampling method used in [10]. Here, 10^7 sampling points are used to ensure accuracy [10].

The algorithm PESA-II requires a grid division parameter (div). Due to the integration of SDE, the optimal setting for div in PESA-II+SDE is different from that in PESA-II. The settings of div in Table 3.1 can enable the two algorithms separately to achieve the best performance on the test instances.

All the results presented in this chapter are obtained by executing 30 independent runs of each algorithm on each problem with the termination criterion of 100,000 evaluations. Following the practice in [119], the population size was set to 200 for the tested algorithms, and the archive was also maintained with the same size if required. A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n denotes the number of decision variables) were used. For the continuous problem DTLZ, the simulated binary crossover (SBX) and polynomial mutation with both distribution indexes 20 [10] were used as crossover and mutation operators. For the combinatorial TSP, the order crossover and inversion mutation were used according to [194].

3.4.1 NSGA-II vs NSGA-II+SDE

Table 3.2 shows the results of the two algorithms on the DTLZ and TSP problems regarding the mean and standard deviation (SD) values, where IGD and HV were used for the DTLZ and TSP problems, respectively. The better result regarding the mean for each problem is highlighted in boldface. Moreover, in order to have statistically

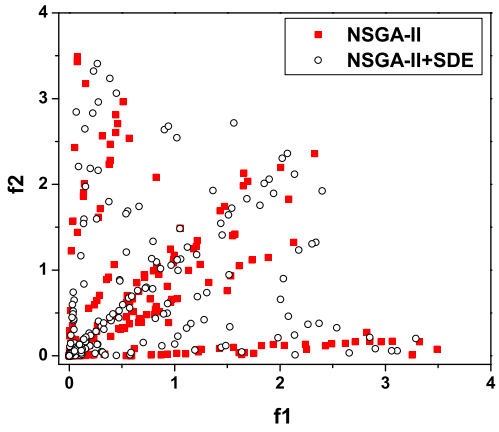


Figure 3.5: Result comparison between NSGA-II and NSGA-II+SDE on the 10-objective DTLZ2. The final solutions of the algorithms are shown regarding the two-dimensional objective space f_1 and f_2.

sound conclusions, the Wilcoxon's rank sum test [290] at a 0.05 significance level was adopted to test the significance of the differences between assessment results obtained by two competing algorithms.

As can be seen from Table 3.2, the performance of NSGA-II has an improvement when SDE is applied to the algorithm, achieving a better value in 24 out of all 30 test instances. Also, for most of the problems on which NSGA-II+SDE outperforms NSGA-II, the results have statistical significance (21 out of the 24 problems). Especially, for the TSP problem suite, NSGA-II+SDE shows a significant advantage over its competitor, with statistical significance for all 9 test instances.

Despite an improvement obtained, NSGA-II+SDE actually struggles to cope with MaOPs. Figure 3.5 plots the final solutions of the two algorithms in a single run regarding the two-dimensional objective space f_1 and f_2 of the 10-objective DTLZ2. Similar plots can be obtained for other objectives of the problem. This particular run is associated with the result which is the closest to the mean IGD value. Clearly, although NSGA-II+SDE tends to perform slightly better than NSGA-II in terms of diversity, both algorithms fail to approach the Pareto front of the problem, given that the range of the optimal front is $[0, 1]$ for each objective. A detailed explanation of the failure of NSGA-II+SDE will be given in Chapter 3.4.4.

Table 3.2: Performance comparison between NSGA-II and NSGA-II+SDE regarding the mean and standard deviation (SD) values on the DTLZ and TSP test suites, where IGD was used for DTLZ and HV for TSP. The better result regarding the mean for each problem instance is highlighted in boldface

Problem	4-objective		6-objective		10-objective	
	NSGA-II	NSGA-II+SDE	NSGA-II	NSGA-II+SDE	NSGA-II	NSGA-II+SDE
DTLZ1	8.894E−2 (6.6E−2)	**5.294E−2 (6.9E−3)**†	2.141E+1 (1.9E+1)	**1.512E+1 (7.8E+0)**	**4.471E+1 (3.3E+1)**	4.801E+1 (2.2E+1)
DTLZ2	1.199E−1 (5.0E−3)	**1.168E−1 (4.2E−3)**†	1.104E+0 (1.7E−1)	**6.160E−1 (8.0E−2)**†	2.112E+0 (1.5E−1)	**1.907E+0 (1.8E−1)**†
DTLZ3	5.099E+0 (2.8E+0)	**4.234E+0 (2.2E+0)**	2.668E+2 (9.1E+1)	**1.593E+2 (3.9E+1)**†	5.928E+2 (1.9E+2)	**3.802E+2 (1.4E+2)**†
DTLZ4	**1.096E−1 (3.6E−3)**	1.098E−1 (3.1E−3)	7.038E−1 (1.8E−1)	**3.388E−1 (3.9E−2)**†	2.357E+0 (1.8E−1)	**2.275E+0 (2.9E−2)**†
DTLZ5	**2.964E−2 (5.1E−3)**	3.650E−2 (1.0E−2)†	**1.030E−1 (3.6E−2)**	1.503E−1 (3.4E−2)†	**1.887E−1 (1.0E−1)**	3.880E−1 (1.7E−1)†
DTLZ6	3.367E+0 (1.9E−1)	**2.867E+0 (2.4E−1)**†	8.346E+0 (3.7E−1)	**7.772E+0 (4.5E−1)**†	**9.407E+0 (3.0E−1)**	9.701E+0 (2.8E−1)†
DTLZ7	1.626E−1 (6.1E−3)	**1.493E−1 (4.7E−3)**†	5.676E−1 (1.7E−2)	**5.227E−1 (1.9E−2)**†	2.288E+0 (6.1E−1)	**2.160E+0 (5.6E−1)**
TSP(−0.2)	4.786E+4 (2.2E+3)	**6.377E+4 (4.4E+3)**†	2.861E+6 (4.7E+5)	**4.274E+6 (5.2E+5)**†	1.040E+10 (1.8E+09)	**1.582E+10 (2.3E+09)**†
TSP(0)	5.488E+4 (3.4E+3)	**6.866E+4 (3.9E+3)**†	4.041E+6 (4.4E+5)	**5.669E+6 (6.0E+5)**†	1.801E+10 (2.6E+09)	**2.496E+10 (6.4E+09)**†
TSP(0.2)	6.162E+4 (3.1E+3)	**6.917E+4 (2.4E+3)**†	5.379E+6 (7.0E+5)	**7.580E+6 (8.0E+5)**†	2.545E+10 (4.5E+09)	**3.622E+10 (8.5E+09)**†

"†" indicates that the two results are significantly different at a 0.05 level by the Wilcoxon's rank sum test.

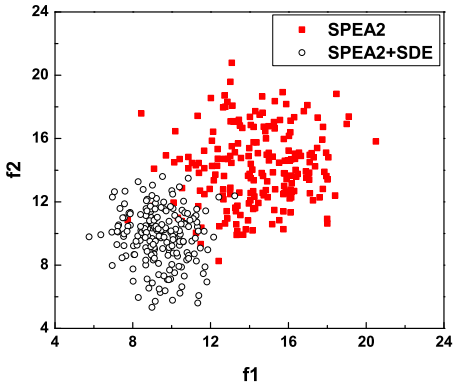


Figure 3.6: Result comparison between SPEA2 and SPEA2+SDE on the 10-objective TSP with $TSPcp = 0$. The final solutions of the algorithms are shown regarding the two-dimensional objective space f_1 and f_2.

3.4.2 SPEA2 vs SPEA2+SDE

Table 3.3 shows the comparative results of the two algorithms on the DTLZ and TSP test problems. In contrast to the slight difference between NSGA-II+SDE and NSGA-II, SPEA2+SDE significantly outperforms the original SPEA2. SPEA2+SDE achieves a better value for all the 30 test instances except the 4-objective DTLZ2, and with statistical significance on 28 instances. Moreover, the advantage of SPEA2+SDE becomes clearer as the number of objectives increases – more than an order of magnitude advantage of IGD is obtained for most of the 10-objective instances (i.e., DTLZ1, DTLZ3, DTLZ5, DTLZ6, and three TSP problems with different conflict degrees among the objectives). Figure 3.6 shows the final solutions of a single run of SPEA2 and SPEA2+SDE regarding the objective space f_1 and f_2 of the 10-objective TSP with $TSPcp = 0$. It is clear from the figure that the proximity performance of SPEA2 is significantly improved when SDE is applied to the algorithm.

3.4.3 PESA-II vs PESA-II+SDE

Using a grid technique to maintain diversity, PESA-II has been found to outperform NSGA-II and SPEA2 in many-objective optimisation [147]. In spite of this, SDE can significantly enhance the performance of PESA-II. Table 3.4 gives the compar-

Table 3.3: Performance comparison between SPEA2 and SPEA2+SDE regarding the mean and standard deviation (SD) values on the DTLZ and TSP test suites, where IGD was used for DTLZ and HV for TSP. The better result regarding the mean for each problem instance is highlighted in boldface

Problem	4-objective		6-objective		10-objective	
	SPEA2	SPEA2 +SDE	SPEA2	SPEA2 +SDE	SPEA2	SPEA2 +SDE
DTLZ1	7.567E−2 (6.0E−2)	**3.258E−2 (3.3E−4)**†	8.026E+1 (2.1E+1)	**6.223E−2 (5.0E−4)**†	1.916E+2 (3.2E+1)	**9.861E−2 (1.3E−3)**†
DTLZ2	**1.074E−1 (4.3E−3)**	1.121E−1 (2.1E−3)†	1.150E+0 (8.5E−2)	**2.703E−1 (4.0E−3)**†	2.457E+0 (1.7E−1)	**4.906E−1 (4.8E−3)**†
DTLZ3	7.200E+0 (6.1E+0)	**1.133E−1 (2.8E−3)**†	5.955E+2 (1.1E+2)	**2.703E−1 (3.3E−3)**†	1.526E+3 (1.5E+2)	**4.947E−1 (8.1E−3)**†
DTLZ4	1.242E−1 (1.2E−1)	**1.129E−1 (2.3E−3)**†	5.163E−1 (1.1E−1)	**2.722E−1 (2.9E−2)**†	2.485E+0 (2.6E−2)	**4.701E−1 (6.0E−3)**†
DTLZ5	8.516E−2 (1.3E−2)	**2.431E−2 (2.2E−3)**†	9.917E−1 (2.1E−1)	**8.052E−2 (1.3E−2)**†	2.261E+0 (3.4E−1)	**1.375E−1 (3.0E−2)**†
DTLZ6	2.444E+0 (1.8E−1)	**7.879E−2 (1.8E−2)**†	9.781E+0 (4.4E−2)	**1.470E−1 (1.9E−2)**†	9.993E+0 (1.5E−2)	**2.784E−1 (2.2E−2)**†
DTLZ7	1.336E−1 (2.7E−3)	**1.326E−1 (5.0E−3)**	7.059E−1 (2.9E−2)	**4.217E−1 (8.5E−3)**†	1.618E+0 (9.2E−2)	**8.868E−1 (4.7E−3)**†
TSP(−0.2)	6.973E+4 (2.4E+3)	**9.667E+4 (1.7E+3)**†	4.946E+6 (4.2E+5)	**1.825E+7 (5.1E+5)**†	1.493E+10 (2.2E+09)	**3.669E+11 (1.6E+10)**†
TSP(0)	6.735E+4 (2.4E+3)	**8.357E+4 (1.7E+3)**†	5.803E+6 (3.3E+5)	**1.550E+7 (3.5E+5)**†	1.683E+10 (2.3E+09)	**2.984E+11 (9.8E+09)**†
TSP(0.2)	6.741E+4 (2.1E+3)	**7.493E+4 (1.7E+3)**†	7.102E+6 (4.0E+5)	**1.357E+7 (3.1E+5)**†	2.423E+10 (4.0E+09)	**2.481E+11 (9.1E+09)**†

"†" indicates that the two results are significantly different at a 0.05 level by the Wilcoxon's rank sum test.

Table 3.4: Performance comparison between PESA-II and PESA-II+SDE regarding the mean and standard deviation (SD) values on the DTLZ and TSP test suites, where IGD was used for DTLZ and HV for TSP. The better result regarding the mean for each problem instance is highlighted in boldface

Problem	4-objective		6-objective		10-objective	
	PESA-II	PESA-II+SDE	PESA-II	PESA-II+SDE	PESA-II	PESA-II+SDE
DTLZ1	5.740E–1 (4.5E–1)	**3.729E–2 (3.4E–3)**†	1.341E+1 (5.5E+0)	**8.920E–2 (1.1E–2)**†	2.802E+1 (8.7E+0)	**1.552E–1 (4.4E–2)**†
DTLZ2	1.227E–1 (6.9E–3)	**1.113E–1 (5.3E–3)**†	2.857E–1 (1.1E–2)	**2.249E–1 (3.5E–3)**†	5.671E–1 (3.1E–2)	**3.756E–1 (3.2E–3)**†
DTLZ3	7.593E+0 (4.4E+0)	**2.365E–1 (2.5E–1)**†	1.303E+2 (3.8E+1)	**4.464E–1 (2.9E–1)**†	3.037E+2 (5.3E+1)	**1.222E+0 (1.4E+0)**†
DTLZ4	**1.192E–1 (7.8E–3)**	2.752E–1 (2.8E–1)	2.969E–1 (6.2E–3)	**2.579E–1 (6.4E–2)**†	6.929E–1 (4.3E–2)	**3.955E–1 (1.8E–2)**†
DTLZ5	1.008E–1 (1.5E–2)	**4.805E–2 (6.5E–3)**†	3.599E–1 (6.3E–2)	**3.283E–1 (7.4E–2)**	5.202E–1 (1.1E–1)	**4.087E–1 (7.0E–2)**†
DTLZ6	1.797E+0 (1.1E–1)	**1.510E–1 (2.7E–2)**†	6.689E+0 (2.4E–1)	**5.516E–1 (3.2E–2)**†	8.690E+0 (3.7E–1)	**8.679E–1 (1.2E–1)**†
DTLZ7	1.415E–1 (6.5E–3)	**1.352E–1 (4.8E–3)**†	6.164E–1 (6.5E–2)	**4.147E–1 (6.9E–2)**†	7.522E+0 (1.7E+0)	**1.259E+0 (3.6E–1)**†
TSP(–0.2)	7.406E+4 (4.1E+3)	**8.984E+4 (2.3E+3)**†	4.026E+6 (3.9E+5)	**1.533E+7 (6.4E+5)**†	8.564E+09 (8.8E+08)	**1.855E+11 (2.6E+10)**†
TSP(0)	7.177E+4 (2.3E+3)	**7.883E+4 (1.8E+3)**†	5.782E+6 (5.0E+5)	**1.354E+7 (5.5E+5)**†	9.186E+09 (1.1E+09)	**2.022E+11 (1.5E+10)**†
TSP(0.2)	6.867E+4 (2.0E+3)	**7.160E+4 (1.3E+3)**†	9.115E+6 (8.1E+5)	**1.249E+7 (3.8E+5)**†	1.286E+10 (2.1E+09)	**1.897E+11 (7.8E+09)**†

"†" indicates that the two results are significantly different at a 0.05 level by the Wilcoxon's rank sum test.

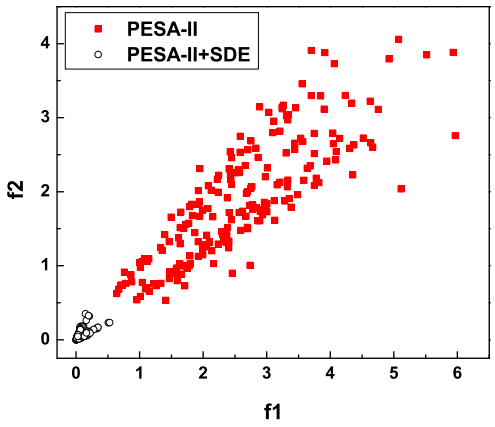


Figure 3.7: Result comparison between PESA-II and PESA-II+SDE on the 10-objective DTLZ6. The final solutions of the algorithms are shown regarding the two-dimensional objective space f_1 and f_2.

ative results of PESA-II and PESA-II+SDE on the DTLZ and TSP test problems. PESA-II+SDE achieves a better assessment result than the original PESA-II on all the 30 instances except the 4-objective DTLZ4, and with statistical significance for 28 instances. Especially, on some MOPs where big obstacles exist for an algorithm to converge into the Pareto front, such as DTLZ1, DTLZ3, and DTLZ6, more than an order of magnitude advantage is achieved for all the 4, 6, and 10-objective instances. Figure 3.7 plots the final solutions of a single run of the two algorithms regarding the objective space f_1 and f_2 of the 10-objective DTLZ6. A clear difference in terms of proximity between the two solution sets can be observed in the figure.

3.4.4 Comparison among NSGA-II+SDE, SPEA2+SDE and PESA-II+SDE

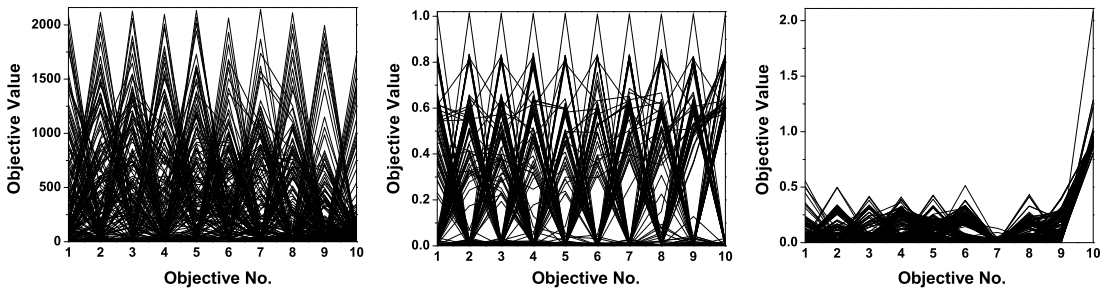
Previous studies presented different behaviours of the three Pareto-based algorithms when SDE is integrated to them in many-objective optimisation. This section compares the three new algorithms and tries to investigate: 1) why they behave differently and 2) which density estimator is more suitable for SDE. Table 3.5 gives the comparative results of the three algorithms on the DTLZ and TSP test problems.

As can be seen from Table 3.5, SPEA2+SDE and PESA-II+SDE significantly out-

Table 3.5: Performance comparison (mean and SD) of NSGA-II+SDE, SPEA2+SDE, and PESA-II+SDE on the DTLZ and TSP test suites, where IGD was used for DTLZ and HV for TSP. The best result regarding the mean value among the three algorithms for each problem instance is highlighted in boldface

Problem	Obj.	NSGA-II+SDE	SPEA2+SDE	PESA-II+SDE
DTLZ1	4	5.294E–2 (6.9E–2)†	**3.258E–2 (3.3E–4)**†	3.729E–2 (3.4E–3)†
	6	1.512E+1 (7.8E+0)†	**6.223E–2 (5.0E–4)**†	8.920E–2 (1.1E–2)†
	10	4.801E+1 (2.2E+1)†	**9.861E–2 (1.3E–3)**†	1.552E–1 (4.4E–2)†
DTLZ2	4	1.168E–1 (4.2E–3)†	1.121E–1 (2.1E–3)	**1.113E–1 (5.3E–3)**†
	6	6.160E–1 (8.0E–2)†	2.703E–1 (4.0E–3)†	**2.249E–1 (3.5E–3)**†
	10	1.907E+0 (1.8E–1)†	4.906E–1 (4.8E–3)†	**3.756E–1 (3.2E–3)**†
DTLZ3	4	4.234E+0 (2.2E+0)†	**1.133E–1 (2.8E–3)**†	2.365E–1 (2.5E–1)†
	6	1.593E+2 (3.9E+1)†	**2.703E–1 (3.3E–3)**†	4.464E–1 (2.9E–1)†
	10	3.802E+2 (1.4E+2)†	**4.947E–1 (8.1E–3)**†	1.222E+0 (1.4E+0)†
DTLZ4	4	**1.098E–1 (3.1E–3)**†	1.129E–1 (2.3E–3)†	2.752E–1 (2.8E–1)
	6	3.388E–1 (3.9E–2)†	2.722E–1 (2.9E–2)†	**2.579E–1 (6.4E–2)**†
	10	2.275E+0 (2.9E–2)†	4.701E–1 (6.0E–3)†	**3.955E–1 (1.8E–2)**†
DTLZ5	4	3.650E–2 (1.0E–2)†	**2.431E–2 (2.2E–3)**†	4.805E–2 (6.5E–3)†
	6	1.503E–1 (3.4E–2)†	**8.052E–2 (1.3E–2)**†	3.283E–1 (7.4E–2)†
	10	3.880E–1 (1.7E–1)†	**1.375E–1 (3.0E–2)**†	4.087E–1 (7.0E–2)
DTLZ6	4	2.867E+0 (2.4E–1)†	**7.879E–2 (1.8E–2)**†	1.510E–1 (2.7E–2)†
	6	7.772E+0 (4.5E–1)†	**1.470E–1 (1.9E–2)**†	5.516E–1 (3.2E–2)†
	10	9.701E+0 (2.8E–1)†	**2.784E–1 (2.2E–2)**†	8.679E–1 (1.2E–1)†
DTLZ7	4	1.493E–1 (4.7E–3)†	**1.326E–1 (5.0E–3)**	1.352E–1 (4.8E–3)†
	6	5.227E–1 (1.9E–2)†	4.217E–1 (8.5E–3)†	**4.147E–1 (6.9E–2)**†
	10	2.160E+0 (5.6E–1)†	**8.868E–1 (4.7E–3)**†	1.259E+0 (3.6E–1)†
TSP(–0.2)	4	6.377E+4 (4.4E+3)†	**9.667E+4 (1.7E+3)**†	8.984E+4 (2.3E+3)†
	6	4.274E+6 (5.2E+5)†	**1.825E+7 (5.1E+5)**†	1.533E+7 (6.4E+5)†
	10	1.582E+10 (2.3E+09)†	**3.669E+11 (1.6E+10)**†	1.855E+11 (2.6E+10)†
TSP(0)	4	6.866E+4 (3.9E+3)†	**8.357E+4 (1.7E+3)**†	7.883E+4 (1.8E+3)†
	6	5.669E+6 (6.0E+5)†	**1.550E+7 (3.5E+5)**†	1.354E+7 (5.5E+5)†
	10	2.496E+10 (6.4E+09)†	**2.984E+11 (9.8E+09)**†	2.022E+11 (1.5E+10)†
TSP(0.2)	4	6.917E+4 (2.4E+3)†	**7.493E+4 (1.7E+3)**†	7.160E+4 (1.3E+3)†
	6	7.580E+6 (8.0E+5)†	**1.357E+7 (3.1E+5)**†	1.249E+7 (3.8E+5)†
	10	3.622E+10 (8.5E+09)†	**2.481E+11 (9.1E+09)**†	1.897E+11 (7.8E+09)†

"†" indicates that the result of the considered algorithm is significantly different from that of its right algorithm (i.e., NSGA-II+SDE vs SPEA2+SDE, SPEA2+SDE vs PESA-II+SDE, and PESA-II+SDE vs NSGA-II+SDE) at a 0.05 level by the Wilcoxon's rank sum test.



(a) NSGA-II+SDE (b) SPEA2+SDE (c) PESA-II+SDE

Figure 3.8: The final solution set of the three algorithms on the ten-objective DTLZ3, shown by parallel coordinates.

performs NSGA-II+SDE. SPEA2+SDE outperforms NSGA-II+SDE for all test instances except for the 4-objective DTLZ4, and PESA-II outperforms NSGA-II+SDE in 26 out of all 30 instances. Especially, for the 6- and 10-objective DTLZ1 and DTLZ3 problems, the advantage of the first two algorithms over NSGA-II+SDE is more than two orders of magnitude. On the other hand, considering the results between SPEA2+SDE and PESA-II+SDE, the performance difference is also clear. The former has an advantage over the latter in 24 out of the 30 instances. More specifically, SPEA2+SDE achieves better results on all the DTLZ1, DTLZ3, DTLZ5, DTLZ6, and TSP instances as well as on most of the DTLZ7 instances, while PESA-II+SDE performs better on the three instances of DTLZ2, 6- and 10-objective DTLZ4, and 6-objective DTLZ7. In addition, the difference among three algorithms has statistical significance on most of all the 30 test instances: 30 for NSGA-II+SDE versus SPEA2+SDE, 28 for SPEA2+SDE versus PESA-II+SDE, and 28 for PESA-II+SDE versus NSGA-II+SDE.

Figure 3.8 shows the final solutions of the three algorithms on the 10-objective DTLZ3 by parallel coordinates based on the single run where the result is the closest to the mean IGD value. Parallel coordinates have been found to be a useful tool for visualising many-objective solutions in a two-dimensional graph [71]. Each line in the graph connects the objective values achieved by an individual of the population and represents a potential solution to the given problem. The DTLZ3 test problem, by introducing a vast number of local optima ($3^{10} - 1$), poses a stiff challenge for

an algorithm to search towards the global optimal front, especially when the number of objectives becomes large. The global optimal front of the problem is a spherical front satisfying $f_1^2 + f_2^2 + ... + f_M^2 = 1$ in the range $f_1, f_2, ..., f_M \in [0, 1]$. For this problem, NSGA-II+SDE fails to approach the Pareto front, with the upper boundary of its solutions exceeding 1600 on each objective, as shown in Figure 3.8. Most of the solutions of PESA-II+SDE can converge into the Pareto front, but fail to cover the whole optimal range. Only SPEA2+SDE can achieve a good balance between proximity and diversity, having a spread of solutions over $f_i \in [0, 1]$ for all the 10 objectives.

Due to the ineffectiveness of the Pareto dominance relation in distinguishing between individuals for many-objective optimisation, the performance differences among NSGA-II+SDE, SPEA2+SDE, and PESA-II+SDE can be attributed to the different behaviours of their density estimators (i.e., the crowding distance, k-th nearest neighbor, and grid crowding degree) in SDE. In the following, we will investigate them in detail.

Recall that to estimate the density of an individual, the crowding distance estimator considers its two closest points on either side along each objective. Due to this separate consideration of the neighbours on each objective, an incorrect estimation of an individual's density may be obtained when the number of objectives is larger than two[3]. This phenomenon has been reported in Kukkonen and Deb's study [163]. In this case, an individual which is far from other individuals in the population may be assigned a low (poor) crowding distance, leading to an incorrect estimation in SDE.

Consider a tri-objective scenario where a population is composed of four non-dominated individuals $\mathbf{A}(1, 1, 1)$, $\mathbf{B}(0, 10, 2)$, $\mathbf{C}(2, 0, 10)$ and $\mathbf{D}(10, 2, 0)$, as shown in Figure 3.9 by parallel coordinates. Clearly, individual $\mathbf{A}$ has a low similar degree with the other three individuals and performs significantly better than them in terms of proximity. However, $\mathbf{A}$ is assigned a poor crowding distance in both TDE and SDE. In TDE, $\mathbf{A}$ has two close neighbours on each objective (i.e., $\mathbf{B}$ and $\mathbf{C}$ on f_1, $\mathbf{C}$ and $\mathbf{D}$ on f_2, and $\mathbf{D}$ and $\mathbf{B}$ on f_3). In SDE, the upper neighbour of $\mathbf{A}$ on each objective remains

[3]For a bi-objective problem, the crowding distance can correctly estimate the density of an individual in the nondominated set since the property of the Pareto dominance relation (which implies a monotonic relation between individuals in the objective space) causes individuals to come close together along both the objectives.

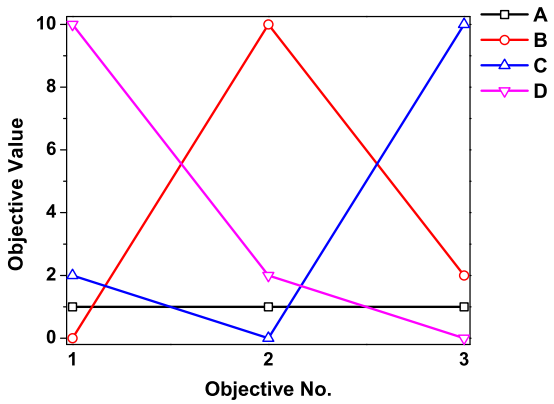


Figure 3.9: An illustration of the failure of the crowding distance in TDE and SDE on a tri-objective scenario, showed by parallel coordinates. In a nondominated set consisting of $\mathbf{A}(1,1,1)$, $\mathbf{B}(0,10,2)$, $\mathbf{C}(2,0,10)$ and $\mathbf{D}(10,2,0)$, individual $\mathbf{A}$ performs well in terms of proximity and diversity. But $\mathbf{A}$ will be assigned a poor density value in both TDE and SDE since the crowding distance separately considers its neighbours on each objective.

unchanged, while the lower neighbour moves to the position of $\mathbf{A}$. Thus, the crowding distance of $\mathbf{A}$ in SDE is $CD(\mathbf{A}) = ((\mathbf{A}_{f_1} - \mathbf{C}_{f_1}) + (\mathbf{A}_{f_2} - \mathbf{D}_{f_2}) + (\mathbf{A}_{f_3} - \mathbf{B}_{f_3}))/3 = 1$, which is clearly worse than $CD(\mathbf{B}) = CD(\mathbf{C}) = CD(\mathbf{D}) = 3$.

Unlike the crowding distance, the k-th nearest neighbour and grid crowding degree estimators consider an individual as a whole, thus avoiding the above misjudgement. The inferior performance of PESA-II+SDE against SPEA2+SDE may be due to the coarseness of the grid-based density estimator. As pointed out in [90], the judgement of density of an individual in grid depends partly on the size of a hyperbox and the position of the hyperbox where the individual is located.

As an explanation for the problem of the grid crowding degree, Figure 3.10 shows a bi-objective nondominated set consisting of individuals $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{E}$. Individual $\mathbf{D}$ performs worse than $\mathbf{C}$ and $\mathbf{E}$ in terms of proximity, thus having two very close neighbours $\mathbf{G}$ and $\mathbf{H}$ in SDE. However, since they are distributed in different hyperboxes, the grid crowding degree of $\mathbf{D}$ is still equal to one. In contrast, individual $\mathbf{C}$, which has a relatively distant neighbour $\mathbf{F}$ in its hyperbox, is assigned a higher grid crowding degree (2). Concerning the k-th nearest neighbour in SPEA2+SDE, it is clear that $\mathbf{D}$ has a higher density value than $\mathbf{C}$ since the Euclidean distance between $\mathbf{D}$ and

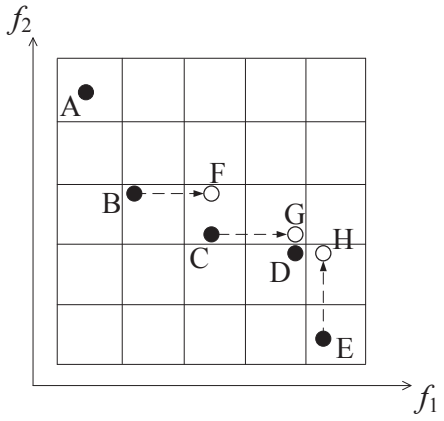


Figure 3.10: An illustration of the inaccuracy of the grid crowding degree. **D** has two very close neighbours **G** and **H** in SDE, but its grid crowding degree is smaller than that of **C** which has a relatively distant neighbour **F**.

its nearest neighbour **G** is smaller than that between **C** and its nearest neighbour **F**.

Overall, the performance difference among the algorithms is due to the difference in the degree of accuracy of their density estimators. A density estimator will be well suited in SDE as long as it can accurately estimate the density of individuals. Next, we will test the competitiveness of the proposed method to existing state-of-the-art methods in many-objective optimisation by comparing SPEA2+SDE with five representative algorithms taken from different categories solving MaOPs.

3.4.5 Comparison with State-of-the-Art Algorithms

We consider five peer algorithms, MOEA/D[4] [278], MSOPS [106], HypE [10], ϵ-MOEA [54], and DMO [1]. These algorithms are representative approaches in dealing with MaOPs, and their performance has been well verified in many-objective optimisation [81, 88, 256, 264, 180]. Readers can refer to Chapter 2.4 for their descriptions.

Parameters need to be set in some algorithms. According to their original papers, the neighbourhood size and the penalty parameter in MOEA/D were set to 10% of the population size and 5, respectively, and the number of sampling points in HypE

[4]Here, the penalty-based boundary intersection (PBI) function is used since MOEA/D with PBI has been found to be more competitive when solving problems with a high-dimensional objective space [52].

was set to 10,000. Since increasing the number of weight vectors with the number of objectives benefits the performance of MSOPS, 200 weight vectors were selected in MSOPS according to the experimental results in [256]. In ϵ-MOEA, the size of the archive set is determined by the ϵ value. In order to guarantee a fair comparison, we set ϵ so that the archive of ϵ-MOEA is approximately of the same size (200) as that of the other algorithms (given in Table 3.1). In addition, in MOEA/D the population size cannot be arbitrarily specified since it is equal to the number of weight vectors. As suggested in [119], we used the closest number to 200 among the possible values as the population size (i.e., 220, 252, and 220 for 4-, 6-, and 10-objective problems, respectively).

Comparison on the DTLZ Test Problems

Table 3.6 shows the comparative results of the six algorithms on the DTLZ problem suite. First, we consider the DTLZ1 problem which has an easy, linear Pareto front, but a huge number of local optima (11^5-1). For this problem, SPEA2+SDE and MOEA/D perform clearly better than the other four algorithms. More precisely, MOEA/D slightly outperforms SPEA2+SDE on the 4-objective instance, while SPEA2+SDE achieves a lower IGD value when a larger number of objectives are involved.

Although having the same optimal front, the problems DTLZ2, DTLZ3, and DTLZ4 are designed to challenge different capabilities of an algorithm. DTLZ2 is a relatively easy function with a spherical Pareto front. Based on DTLZ2, a vast number of local optima are introduced in DTLZ3, creating a big challenge for algorithms to search towards the global optimal front, and a non-uniform density of solutions are introduced in DTLZ4, creating a big challenge for algorithms to maintain diversity in the objective space. As can be seen from Table 3.6, for the DTLZ2 problem, generally, ϵ-MOEA performs best, followed by MOEA/D and SPEA2+SDE. On DTLZ3, MOEA/D and SPEA2+SDE are significantly superior to the other algorithms. The former reaches the best result on the two low-dimensional instances, and the latter outperforms the other algorithms when the number of objectives reaches 10. On DTLZ4, SPEA2+SDE is very competitive. Although MSOPS performs slightly better than SPEA2+SDE on

Table 3.6: IGD results (mean and SD) of the six algorithms on the DTLZ problems. The best result regarding the mean IGD value among the algorithms for each problem instance is highlighted in boldface

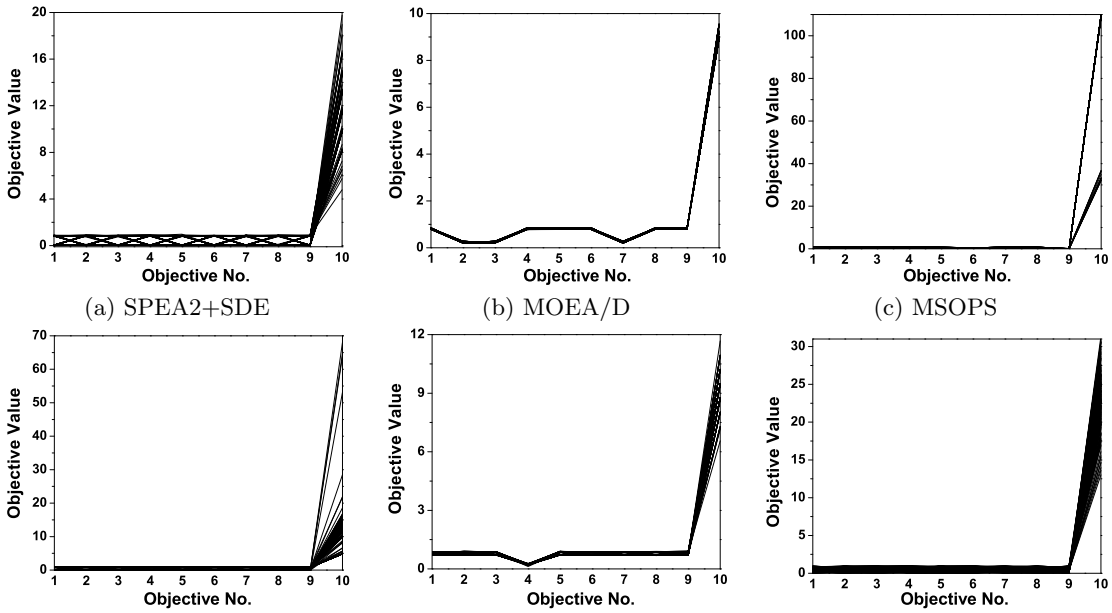
Problem	Obj.	SPEA2+SDE	MOEA/D	MSOPS	HypE	ϵ-MOEA	DMO
DTLZ1	4	3.258E−2 (3.3E−4)	**3.218E−2 (1.0E−4)**†	4.442E−2 (2.7E−3)†	8.728E−2 (9.6E−3)†	3.508E−2 (1.6E−3)†	4.249E−2 (1.8E−3)†
	6	**6.223E−2 (5.0E−4)**	6.489E−2 (3.7E−4)†	3.263E−1 (3.8E−1)†	3.267E−1 (2.4E−1)†	9.504E−2 (1.1E−1)†	4.336E−1 (2.8E−1)†
	10	**9.861E−2 (1.3E−3)**	1.017E−1 (1.6E−3)†	2.095E+0 (1.3E+0)†	4.000E−1 (1.7E−1)†	3.098E−1 (3.0E−1)†	2.709E+0 (3.8E+0)†
DTLZ2	4	1.121E−1 (2.1E−3)	**8.739E−2 (5.1E−6)**†	1.116E−1 (2.5E−3)	1.873E−1 (2.6E−2)†	1.051E−1 (1.7E−3)†	1.415E−1 (1.7E−2)†
	6	2.703E−1 (4.0E−3)	2.566E−1 (2.0E−5)†	3.228E−1 (9.2E−3)†	3.857E−1 (4.5E−2)†	**2.429E−1 (3.7E−3)**†	2.756E−1 (1.4E−2)
	10	4.906E−1 (4.8E−3)	4.921E−1 (6.9E−5)	6.852E−1 (4.9E−2)†	6.294E−1 (9.4E−2)†	**4.048E−1 (4.9E−3)**†	5.280E−1 (1.8E−2)†
DTLZ3	4	1.133E−1 (2.8E−3)	**9.172E−2 (3.2E−3)**†	2.101E+1 (1.1E+1)†	3.515E−1 (1.2E−1)†	1.172E−1 (5.5E−3)†	1.662E+0 (1.5E+0)†
	6	2.703E−1 (3.3E−3)	**2.616E−1 (8.6E−3)**†	5.702E+1 (1.4E+1)†	1.783E+0 (1.2E+0)†	3.365E−1 (1.1E−1)†	5.076E+1 (1.5E+1)†
	10	**4.947E−1 (8.1E−3)**	4.950E−1 (5.0E−3)	8.303E+1 (1.7E+1)†	2.242E+0 (1.3E+0)†	1.273E+1 (2.0E+1)†	2.368E+2 (8.2E+1)†
DTLZ4	4	1.129E−1 (2.3E−3)	3.537E−1 (2.9E−1)†	**1.116E−1 (3.2E−3)**	2.130E−1 (1.3E−1)†	1.531E−1 (1.3E−1)	1.285E−1 (1.1E−2)†
	6	**2.722E−1 (2.9E−2)**	5.231E−1 (1.2E−1)†	3.212E−1 (9.3E−3)†	4.887E−1 (5.5E−2)†	3.150E−1 (1.0E−1)	3.600E−1 (2.1E−2)†
	10	**4.701E−1 (6.0E−3)**	6.778E−1 (7.0E−2)†	6.755E−1 (2.4E−2)†	8.358E−1 (7.6E−2)†	4.725E−1 (5.8E−2)	6.822E−1 (3.8E−2)†
DTLZ5	4	2.431E−2 (2.2E−3)	1.607E−2 (1.8E−5)†	**1.557E−2 (1.4E−3)**†	9.803E−2 (1.8E−2)†	3.622E−2 (2.4E−3)†	3.062E−1 (5.5E−2)†
	6	8.052E−2 (1.3E−2)	2.361E−2 (1.4E−4)†	**1.476E−2 (1.8E−3)**†	8.630E−2 (2.1E−2)	1.214E−1 (1.5E−2)†	4.510E−1 (1.0E−1)†
	10	1.375E−1 (3.0E−2)	6.495E−2 (2.2E−6)†	**1.769E−2 (2.0E−3)**†	1.472E−1 (3.2E−2)	1.714E−1 (1.7E−2)†	4.656E−1 (1.1E−1)†
DTLZ6	4	**7.879E−2 (1.8E−2)**	8.636E−2 (3.0E−2)	2.495E+0 (5.8E−1)†	3.473E+0 (5.0E−1)†	6.318E−1 (4.5E−2)†	7.198E+0 (2.2E−1)†
	6	1.470E−1 (1.9E−2)	**1.205E−1 (3.6E−2)**†	7.788E+0 (3.0E−1)†	5.991E+0 (5.8E−1)†	2.904E+0 (2.4E−1)†	6.129E+0 (3.6E−1)†
	10	2.784E−1 (2.2E−2)	**1.532E−1 (4.0E−2)**†	7.738E+0 (2.8E−1)†	5.573E+0 (4.4E−1)†	3.175E+0 (1.9E+0)†	7.769E+0 (4.2E−1)†
DTLZ7	4	**1.326E−1 (5.0E−3)**	4.898E−1 (1.0E−1)†	4.873E−1 (1.2E−1)†	2.923E−1 (8.0E−3)†	2.189E−1 (9.9E−2)†	1.430E−1 (5.6E−3)†
	6	**4.217E−1 (8.5E−3)**	3.923E+0 (9.2E−1)†	9.989E+0 (1.5E+0)†	6.246E−1 (9.7E−2)†	6.815E−1 (1.3E−1)†	5.505E−1 (2.5E−2)†
	10	**8.868E−1 (4.7E−3)**	4.193E+0 (1.2E+0)†	2.208E+1 (3.0E+0)†	1.010E+0 (3.3E−2)†	1.879E+0 (1.8E−1)†	6.561E+0 (2.5E+0)†

"†" indicates that the results of the peer algorithm is significantly different from that of SPEA2+SDE at a 0.05 level by the Wilcoxon's rank sum test.

the 4-objective instance, SPEA2+SDE has a clear advantage over the other algorithms for the remaining instances. In addition, note that MOEA/D, which works very well on the first three problems (DTLZ1, DTLZ2, and DTLZ3), tends to struggle on DTLZ4, obtaining the worst IGD value on 4- and 6-objective test instances. Similar observations have been reported in [52].

The Pareto front of DTLZ5 and DTLZ6 is a degenerate curve in order to test the ability of an algorithm to find a lower-dimensional optimal front while working with a higher-dimensional objective space. The difference between the two problems is that DTLZ6 is much harder than DTLZ5 by introducing bias in the g function [57]. For such problems, MSOPS and MOEA/D work very well. The former performs best on DTLZ5, and the latter outperforms the other algorithms on most of the DTLZ6 instances. Nevertheless, SPEA2+SDE show advantages in the low-dimensional DTLZ6, and for the high-dimensional instances, it performs significantly better than the peer algorithms except MOEA/D. On DTLZ5, SPEA2+SDE is always in the third place, better than HypE, ϵ-MOEA, and DMO.

With a number of disconnected Pareto optimal regions, DTLZ7 tests an algorithm's ability to maintain sub-populations in disconnected portions of the objective space. For this problem, SPEA2+SDE has a clear advantage over the other five algorithms, obtaining the best IGD value for all the three instances. In contrast, two decomposition-based algorithms, MOEA/D and MSOPS, have great difficulty with this problem. The former performs worst on the 4-objective instance, and the latter obtains the worst IGD result for the 6- and 10-objective instances. Figure 3.11 plots the final solutions of the six algorithms in a single run on the 10-objective DTLZ7 by parallel coordinates. This particular run is associated with the result which is the closest to the mean IGD value. It is clear from the figure that the solutions of MSOPS, HypE, and DMO fail to converge into the optimal front (the upper bound of the last objective in the Pareto front of DTLZ7 is equal to $2 \times m$, i.e., $f_{10} \leq 20$ for the 10-objective instance). MOEA/D and ϵ-MOEA struggle to maintain diversity, with their solutions converging into a portion of the disconnected Pareto front. Only SPEA2+SDE achieves a good approximation and coverage of the Pareto front.



(a) SPEA2+SDE (b) MOEA/D (c) MSOPS

(d) HypE (e) ϵ-MOEA (f) DMO

Figure 3.11: The final solution set of the six algorithms on the ten-objective DTLZ7, shown by parallel coordinates.

Overall, SPEA2+SDE is very competitive on the DTLZ problem suite. It obtains the best IGD value in 9 out of the 21 test instances, followed by MOEA/D, MSOPS, and ϵ-MOEA, with the best value in 6, 4, and 2, respectively. Moreover, unlike MOEA/D and MSOPS, whose search ability has sharp contrasts on different problems, SPEA2+SDE has stable performance, ranking well for all the test instances.

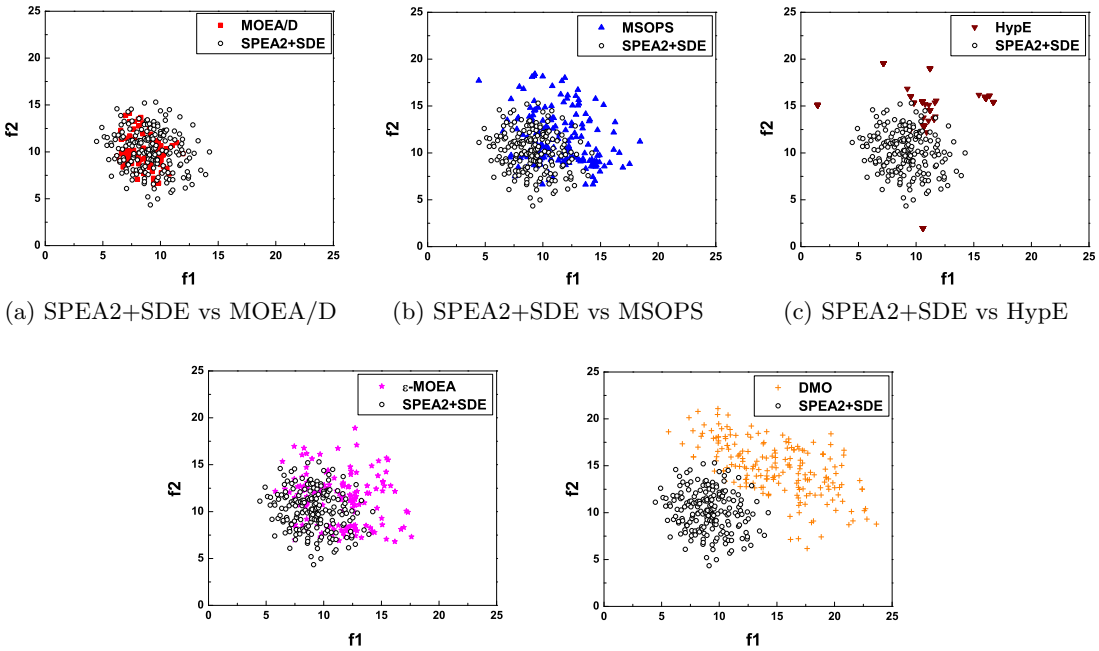
Comparison on the TSP Test Problems

EMO algorithms usually show different behaviour on combinatorial optimisation problems from on continuous ones. One important property of the multi-objective TSP problem is that the conflict degree among the objectives can be adjusted according to the parameter $TSPcp$, where a lower value means a greater degree of conflict. From the HV results shown in Table 3.7, the advantage of SPEA2+SDE over the other algorithms on the TSP seems to be greater than that on the DTLZ problems. SPEA2+SDE significantly outperforms the five peer algorithms for all the instances except the 4-objective TSP with $TSPcp = 0$, where MOEA/D performs slightly better

Table 3.7: HV results (mean and SD) of the six algorithms on the TSP problems. The best result regarding the mean HV value among the algorithms for each problem instance is highlighted in boldface

Problem	Obj.	SPEA2+SDE	MOEA/D	MSOPS	HypE	ϵ-MOEA	DMO
TSP(−0.2)	4	**9.667E+4 (1.7E+3)**	9.394E+4 (1.7E+3)†	7.574E+4 (2.8E+3)†	3.817E+4 (4.5E+3)†	9.044E+4 (1.7E+3)†	5.224E+4 (3.1E+3)†
	6	**1.825E+7 (5.1E+5)**	1.725E+7 (4.0E+5)†	1.061E+7 (6.5E+5)†	2.500E+6 (7.4E+5)†	1.428E+7 (6.6E+5)†	2.893E+6 (4.9E+5)†
	10	**3.669E+11 (1.6E+10)**	2.572E+11 (1.2E+10)†	1.980E+11 (1.7E+10)†	1.033E+10 (1.2E+09)†	1.337E+11 (1.2E+10)†	8.084E+09 (1.6E+09)†
TSP(0)	4	8.357E+4 (1.7E+3)	**8.358E+4 (1.4E+3)**	6.779E+4 (2.0E+3)†	3.973E+4 (2.5E+3)†	7.935E+4 (2.1E+3)†	5.380E+4 (2.9E+3)†
	6	**1.550E+7 (3.5E+5)**	1.458E+7 (3.3E+5)†	1.065E+7 (6.9E+5)†	3.930E+6 (4.8E+5)†	1.331E+7 (5.0E+5)†	3.965E+6 (4.9E+5)†
	10	**2.984E+11 (9.8E+09)**	1.969E+11 (1.3E+10)†	1.898E+11 (1.1E+10)†	1.613E+10 (9.8E+08)†	1.440E+11 (9.3E+09)†	1.691E+10 (3.2E+09)†
TSP(0.2)	4	**7.493E+4 (1.7E+3)**	7.427E+4 (1.7+3)	6.210E+4 (1.6E+3)†	4.639E+4 (3.7E+3)†	7.230E+4 (1.9E+3)†	6.203E+4 (3.2E+3)†
	6	**1.357E+7 (3.1E+5)**	1.264E+7 (3.1E+5)†	1.068E+7 (5.2E+5)†	5.468E+6 (5.3E+5)†	1.240E+7 (4.2E+5)†	5.355E+6 (7.6E+5)†
	10	**2.481E+11 (9.1E+09)**	1.580E+11 (1.0E+10)†	1.662E+11 (9.2E+09)†	4.136E+10 (7.2E+09)†	1.530E+11 (1.3E+10)†	2.504E+10 (5.2E+09)†

"†" indicates that the results of the peer algorithm is significantly different from that of SPEA2+SDE at a 0.05 level by the Wilcoxon's rank sum test.



(a) SPEA2+SDE vs MOEA/D (b) SPEA2+SDE vs MSOPS (c) SPEA2+SDE vs HypE

(d) SPEA2+SDE vs ϵ-MOEA (e) SPEA2+SDE vs DMO

Figure 3.12: Result comparison between SPEA2+SDE and the other algorithms on the 10-objective TSP with $TSPcp = -0.2$. The final solutions of the algorithms are shown regarding the two-dimensional objective space f_1 and f_2.

than SPEA2+SDE. Moreover, the performance difference can be better observed with the increase of the number of objectives.

To facilitate visual comparison, Figure 3.12 shows the final solutions of a single run of the six algorithms regarding the two-dimensional objective space f_1 and f_2 of the 10-objective TSP with $TSPcp = -0.2$. Clearly, the solutions of SPEA2+SDE have a good balance between proximity and diversity. In contrast, the solutions of MSOPS, ϵ-MOEA, and DMO are worse than those of SPEA2+SDE in terms of proximity. MOEA/D struggles to maintain diversity, making its solutions concentrated in a small region. Although there are several solutions distributed widely, most of the solutions of HypE have a poor proximity, thus leading to its low HV value.

Finally, it is worth mentioning that the difference between SPEA2+SDE and the peer algorithms on most of all the 30 DTLZ and TSP problems has statistical significance. Specifically, the proportion of the test instances where SPEA2+SDE outper-

forms MOEA/D, MSOPS, HypE, ϵ-MOEA and DMO with statistical significance is 15/19, 25/25, 28/30, 24/27 and 29/30, respectively.

3.5 Discussions

The impressive results of SPEA2+SDE motivate us to deeply explore the density estimation mechanism (i.e., the k-th nearest neighbour) in SPEA2. SPEA2 employs Euclidean distance to measure the similarity degree between individuals. The calculation of Euclidean distance can be viewed as an aggregation of each dimension's difference. In SDE, the dimensions in the aggregation are switched on (or off) when the interested individual performs better (or worse) than its opponent. This means that an individual which has no clear advantage over its opponents will have a high similarity degree with them, thus being assigned a high density value in the population.

In addition, the parameter k in the k-th nearest neighbour approach has no clear influence on the performance of SPEA2+SDE in many-objective optimisation. In SPEA2, k is used in the fitness assignment procedure, which serves the purpose of sorting individuals for archiving when the number of nondominated individuals is smaller than the archive size. However, in many-objective optimisation, most individuals are nondominated to each other, and usually the number of nondominated individuals is far larger than the archive size. In this case, the archive truncation procedure, which does not need to specify the parameter k, plays a decisive role in the algorithm's performance.

Finally, note that there is little additional computational cost of the proposed approach. Contrasting the calculation of SDE (Eqs. (3.2) and (3.3)) with that of TDE (Eq. (3.1)), only a comparison of objective values was added before estimating the similarity degree between individuals. This indicates a negligible computational difference between the original Pareto-based algorithms and their SDE version.

3.6 Summary

Many-objective optimisation presents great challenges for traditional Pareto-based algorithms. The imbalance of the role between the Pareto dominance relation and di-

versity maintenance suggests the need for new methodologies in the EMO community. In this chapter, we proposed a shift-based density estimation (SDE) strategy to enhance traditional density estimators in Pareto-based algorithms for MaOPs. By shifting individuals' positions according to their relative closeness to the Pareto front, SDE considers both proximity and diversity for each individual in the population. The implementation of SDE is simple and it can be applied to any specific density estimator without the need of additional parameters.

Systematic experiments were carried out by providing an extensive comparison on several groups of well-defined continuous and combinatorial test problems. SDE was separately applied to three popular Pareto-based algorithms, NSGA-II, SPEA2, and PESA-II. From the comparative results, it has been observed that all the three algorithms after the implementation of SDE achieve an improvement of performance with varying degrees. A further comparative study among NSGA-II+SDE, SPEA2+SDE, and PESA-II+SDE has revealed that SDE is well suited for the density estimator which can accurately reflect the density of individuals in the population. Moreover, five state-of-the-art EMO algorithms (MOEA/D, MSOPS, HypE, ϵ-MOEA, and DMO) for solving MaOPs from different perspectives were used as peer algorithms to verify the proposed SDE strategy. The experimental results have shown that SPEA2+SDE is very competitive against the peer algorithms in terms of providing a good balance between proximity and diversity. This leads to the two key findings of this chapter.

- Pareto-based algorithms, with a general enhancement, can be appropriate for many-objective optimisation, which refutes the common belief that the Pareto-based algorithm framework performs worse than the decomposition-based or indicator-based algorithm frameworks in dealing with MaOPs.

- When designing a Pareto-based algorithm, researchers only need to focus on tackling low-dimensional (i.e., 2-objective and 3-objective) optimisation problems; when addressing an MOP with many objectives, SDE may be easily and effectively adopted, provided that the algorithm's density estimator can accurately reflect the density of individuals.

Finally, it is important to note that using density estimators that reflect both proximity and density information will reduce the degree of accuracy of density estimation to some extent. This is a bit like the classical fitness sharing and penalty based approaches [86, 72], which change the original fitness value (with respect to proximity) of individuals to reflect their distribution information. Similarly, SDE changes the original density value of individuals to reflect their proximity information in the case of the Pareto dominance relation losing its effectiveness.

Chapter 4

A Grid-Based Evolutionary Algorithm

In multi-objective optimisation, the grid has an inherent property of reflecting the information of proximity and diversity simultaneously. Each solution in the grid has a deterministic location. The performance of a solution regarding proximity can be estimated by its grid location in comparison with other solutions, and the performance of a solution regarding diversity can be estimated by the number of solutions whose grid locations are identical with or similar to its grid location. Moreover, in contrast to the Pareto dominance criterion, a grid-based criterion can not only qualitatively compare solutions but also give the quantitative difference in each objective among them. This is well suited to MaOPs, given the increase of selection pressure obtained from the quantitative comparison of solutions' objective values [128, 45].

With these in mind, this chapter proposes a grid-based evolutionary algorithm (GrEA), which takes maximum advantage of the potential of the grid to deal with many-objective optimisation problems.

The chapter is organised as follows. In Section 4.1, we give the motivation of this work. Section 4.2 describes the proposed approach in detail. Section 4.3 is devoted to experimental studies, consisting of experimental setting, algorithm comparison, parameter investigation, and computational cost examination. Finally, Section 4.4 provides

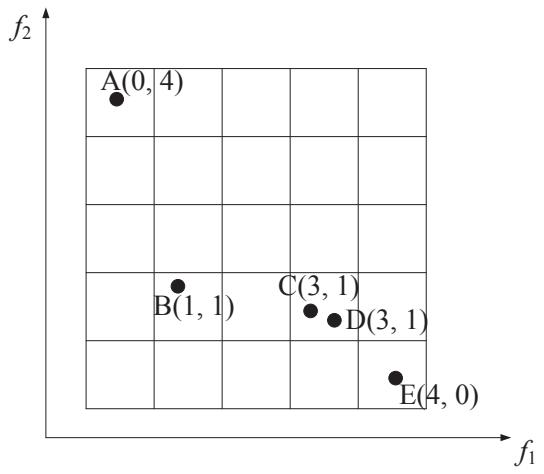


Figure 4.1: An illustration of individuals in grid for a bi-objective scenario.

some concluding remarks.

4.1 Motivation

Grid has a natural ability to reflect the distribution of individuals in the evolutionary process by their own grid locations (i.e., grid coordinates). The difference between grid coordinates of individuals indicates the distance between solutions and further implies the density information of individuals in the population. For example, Figure 4.1 illustrates individuals in grid in a bi-objective space. For individuals **A**, **B**, and **C** in the figure, their grid coordinates are $(0, 4)$, $(1, 1)$, and $(3, 1)$, respectively. The difference of grid coordinates between **A** and **C** (i.e., $(3 - 0) + (4 - 1) = 6$) is larger than that between **A** and **B** (i.e., $(1 - 0) + (4 - 1) = 4$), which indicates that **C** is farther away from **A** than **B**. In addition, given that there exists another individual (**D**) which has the identical grid coordinate with **C** (i.e., the difference of grid coordinates between them is 0), **C** can be considered to be of a higher crowding degree in comparison with **A** and **B**.

On the other hand, grid is also capable of indicating the evolutionary status of individuals in terms of proximity. The grid coordinate takes into account not only whether one individual is better than another but also the difference in objective values

between them. For example, considering individuals **A** and **B** with their own grid coordinates $(0, 4)$ and $(1, 1)$ in Figure 4.1, it is clear that the difference in objective f_2 between them is greater than that in objective f_1 (i.e., $(4 - 1) > (1 - 0)$). This means that grid can further distinguish between individuals when they are on a tie in the sense of Pareto dominance, thereby providing a higher selection pressure in the EMaO process.

Over the last decade, grid-based techniques have been widely applied in the EMO community, resulting in the appearance of several grid-based algorithms. Their behaviour has also been well studied, both theoretically and experimentally [151, 286, 165, 54, 220].

In the first known study of this kind, grid was introduced into the Pareto-based evolution strategy (PAES) proposed by Knowles and Corne [150] to maintain the diversity of the archive set. The crowding degree of an individual is estimated by the number of individuals sharing its grid location. When a nondominated candidate is to join an archive that is full, it replaces one of the individuals with the highest crowding degree if its own crowding degree is lower. Some extended theoretical and practical studies were also presented in [151, 46].

Yen and Lu [273] presented a dynamic multi-objective evolutionary algorithm, using adaptive grid-based rank and density estimation. Unlike PAES and PESA, grid, here, is regarded as an implement to store the information of both proximity and diversity of solutions. Each cell in grid is assigned a rank and a density value according to the Pareto dominance relation and grid location of solutions.

The concept of ϵ-dominance proposed firstly by Laumanns et al. [165] can be seen as a grid-based technique to combine the proximity properties of an EMO algorithm with the need to preserve a diverse set of solutions. Deb et al. [54] developed a steady-state algorithm ϵ-MOEA using ϵ-dominance. It divides the objective space into hyperboxes by the size of ϵ and each hyperbox contains at most a single individual. However, the boundary solutions may be lost in the evolutionary process of the algorithm due to the feature of ϵ-dominance [54, 99]. To address this issue, Hernández-Díaz et al. [99] proposed a variant of the algorithm, called Pareto adaptive ϵ-dominance.

Rachmawati and Srinivasan [220] introduced a dynamic grid resizing strategy, which is capable of shrinking or expanding hyperboxes as necessity dictates. The strategy uses two metrics, the mean occupancy and neighbor occupancy, to detect the setting of grid sizes and further adjusts them correspondingly.

More recently, Karahan and Köksalan [144] developed a territory-based EMO algorithm, TDEA, to solve MOPs. Similar to ϵ-MOEA, TDEA is also a steady-state algorithm. It defines a territory τ around an individual to maintain diversity. Its main difference from ϵ-MOEA lies in that the hyperbox of TDEA is based on individuals rather than being independent on them.

Overall, the above grid-based EMO algorithms are successful, and most of them perform very well on the problems with two or three objectives. However, it is interesting that their application to many-objective problems has received little attention. This may mainly be attributed to three reasons below.

- The need of data storage and computational time increases exponentially. The calculation of most existing grid-based algorithms revolves around hyperboxes in grid. Such box-centred calculation often needs to store the information of each hyperbox in grid (e.g., the number of individuals in each hyperbox). As pointed out by Corne and Knowles [45], these algorithms may not be suitable for many-objective problems since their operation relies on the data structures that exponentially grow in size with the number of objectives. Additionally, the computational cost for high dimensional problems would be tremendous when the box-centred calculation is implemented [147]. If we traverse each hyperbox in an m-dimensional grid, there will be r^m hyperboxes to be accessed, where r is the number of divisions in each dimension.

- The properties of grid are not utilised or exploited sufficiently. The selection criterion of some grid-based algorithms (such as PAES, PESA and TDEA), in the sense of proximity, is based on the Pareto dominance relation, and thus may fail to provide enough selection pressure towards the desired direction in the evolutionary process of many-objective optimisation.

- The density estimator may fail to reflect the distribution of solutions accurately. Since the number of hyperboxes in grid exponentially increases with the number of objectives, the solutions for many-objective problems are likely to disperse in different hyperboxes. Consequently, existing grid-based algorithms which only consider the number of individuals in a single hyperbox would not discriminate individuals by means of their distribution, as the density values are almost equal on the basis of this estimation method.

The above difficulties would largely limit the application of existing grid-based EMO algorithms to MaOPs. However, we argue that these difficulties should not be insurmountable. Firstly, box-centred calculation can be replaced by individual-centred calculation. In this case, grid is merely regarded as a "pointer" to depict the address of individuals. Secondly, a selection criterion based on the difference of grid coordinates can be introduced to strengthen the selection pressure. Finally, the failure of density estimation may also be addressed if the density value of an individual relies on the number of its neighbours not in a single hyperbox but rather in a region constructed by a set of hyperboxes whose range increases with the number of objectives.

Bearing these in mind, we propose a grid-based evolutionary algorithm for MaOPs.

4.2 The Proposed Algorithm

In this section, we first introduce some definitions used in GrEA. Then, we present the framework of the proposed algorithm. Next, we describe the fitness assignment process. Finally, the strategies for mating and environmental selection processes are presented.

4.2.1 Definitions and Concepts

In GrEA, grid is used as a frame to determine the location of individuals in the objective space. Therefore, its adaptability with the evolutionary population is advisable. In other words, when a new population is generated, the location and size of grid should be adapted and adjustable so that it just envelops the population. Here, we adopt

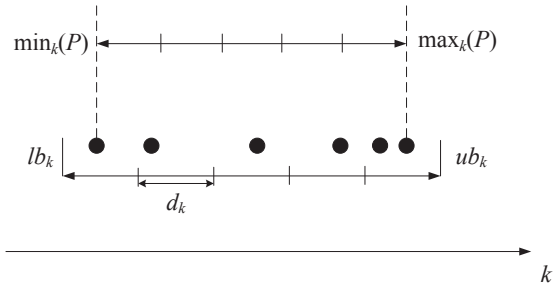


Figure 4.2: Setting of grid in the kth objective.

the adaptive construction of grid, borrowing ideas from the adaptive genetic algorithm (AGA) presented by Knowles and Corne [151].

The grid setting in the kth objective is shown in Figure 4.2. First, the minimum and maximum values regarding the kth objective among the individuals in a population P are found and denoted as $min_k(P)$ and $max_k(P)$, respectively. Then, the lower and upper boundaries of the grid in the kth objective are determined according to the following formulas:

$$lb_k = min_k(P) - (max_k(P) - min_k(P))/(2 \times div) \tag{4.1}$$

$$ub_k = max_k(P) + (max_k(P) - min_k(P))/(2 \times div) \tag{4.2}$$

where div denotes the number of the divisions of the objective space in each dimension (e.g., in Figure 4.2, $div = 5$). Accordingly, the original m-dimensional objective space will be divided into div^m hyperboxes. Thus, the hyperbox width d_k in the kth objective can be formed as follows:

$$d_k = (ub_k - lb_k)/div \tag{4.3}$$

In this case, the grid location of an individual in the kth objective can be determined by lb_k and d_k:

$$G_k(\mathbf{x}) = \lfloor (f_k(\mathbf{x}) - lb_k)/d_k \rfloor \tag{4.4}$$

where "$\lfloor \cdot \rfloor$" denotes the floor function, $G_k(\mathbf{x})$ is the grid coordinate of individual $\mathbf{x}$ in the kth objective, and $f_k(\mathbf{x})$ is the actual objective value in the kth objective. For

example, in Figure 4.2, the grid coordinate of individuals (from left to right) in the kth objective is 0, 1, 2, 3, 4, and 4. In the following, two concepts used in the comparison between individuals are defined on the basis of their grid coordinates.

Definition 4.2.1 (Grid dominance)**.** *Let* $\mathbf{x}, \mathbf{y} \in P, \mathbf{x} \prec_{grid} \mathbf{y}$*:* $\Leftrightarrow$

$$\begin{aligned} &\forall i \in (1, 2, ..., m) : G_i(\mathbf{x}) \leq G_i(\mathbf{y}) \wedge \\ &\exists j \in (1, 2, ..., m) : G_j(\mathbf{x}) < G_j(\mathbf{y}) \end{aligned} \tag{4.5}$$

where "$\mathbf{x} \prec_{grid} \mathbf{y}$" denotes that "$\mathbf{x}$ grid-dominates $\mathbf{y}$", m is the number of objectives, and the grid environment is constructed by the population P. Apparently, the concept of grid dominance is the same as that of Pareto dominance if the grid coordinates of individuals are replaced by their actual objective values. Their relation is as follows. If one solution Pareto-dominates another solution, the latter will not grid-dominate the former, and vice versa. On the other hand, the grid dominance relation permits one solution to dominate another solution if the former is slightly inferior to the latter in some objectives but largely superior to the latter in some other objectives, e.g., the individuals **B** and **C** in Figure 4.1.

The grid dominance relation is similar to the ϵ-dominance relation, in view of that both are the relaxed form of the Pareto dominance relation. But, one important difference is that the degree of relaxation of grid dominance is determined by the evolutionary status of the population. The division number *div* in GrEA is a fixed parameter set by the user beforehand, leading to the proximity and diversity requirements to be adjusted adaptively with the evolution of the population. A widely-distributed population in the objective space (often appearing at the initial stage of evolution) has a larger relaxation degree (i.e., a larger size of a cell in the grid), thereby providing a higher selection pressure; as the population evolves toward the more concentrated Pareto front region, the relaxation degree becomes lower, leading the diversity to be more emphasised.

In addition, the usage of grid dominance in our study is also different from that of ϵ-dominance in the ϵ-dominance based algorithms. In the ϵ-dominance based algorithms, ϵ-dominance is used to determine the survival of individuals. Only nondominated individuals can be preserved in the archive set. However, in GrEA, grid dominance is

Algorithm 4.1 Grid-based Evolutionary Algorithm (GrEA)

Require: P (population), N (population size)
1: $P \leftarrow Initialize(P)$
2: **while** termination criterion not fulfilled **do**
3: $Grid\_setting(P)$
4: $Fitness\_assignment(P)$
5: $P' \leftarrow Mating\_selection(P)$
6: $P'' \leftarrow Variation(P')$
7: $P \leftarrow Environmental\_selection(P \bigcup P'')$
8: **end while**
9: **return** P

mainly used to prevent individuals from being archived earlier than their competitors that grid-dominate them (this will be explained later in the fitness adjustment strategy part in Section 4.2.5). This means that grid-dominated individuals can also have the chance to enter the archive set, which is useful for the maintenance of boundary individuals in the population to some extent.

Definition 4.2.2 (Grid difference)**.** *Let* $\mathbf{x}, \mathbf{y} \in P$*, the grid difference between them is denoted as:*

$$GD(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{m} \mid G_k(\mathbf{x}) - G_k(\mathbf{y}) \mid \tag{4.6}$$

Grid difference is influenced by the number of divisions div, ranging from 0 to $m(div - 1)$. The larger the div, the smaller the size of a cell and the higher the grid difference value between individuals.

4.2.2 Basic Procedure

Algorithm 4.1 gives the basic framework of GrEA. The main procedure of the algorithm is similar to most generational EMO algorithms like NSGA-II [55] and SPEA2 [292]. Firstly, N individuals are randomly generated to form an initial population P. Then, the grid environment for the current population P is set as described in the previous section, and the fitness of individuals in P is assigned according to their location in the grid. Next, mating selection is performed to pick out promising solutions for variation. Finally, the environmental selection procedure is implemented to keep a record of the N best solutions (elitists) for survival.

4.2.3 Fitness Assignment

In order to evolve the population towards the optimum as well as diversify the individuals uniformly along the obtained trade-off surface, the fitness of individuals should contain the information in terms of both proximity and diversity. Here we take three grid-based criteria into account to assign the fitness of individuals. They are grid ranking (GR), grid crowding distance (GCD), and grid coordinate point distance (GCPD). The first and last criteria are used to evaluate the proximity of individuals while the middle one is concerned with the diversity of individuals in the population.

GR is a proximity estimator to rank individuals on the basis of their grid locations. For each individual, GR is defined as the summation of its grid coordinate in each objective:

$$GR(\mathbf{x}) = \sum_{k=1}^{m} G_k(\mathbf{x}) \tag{4.7}$$

where $G_k(\mathbf{x})$ denotes the grid coordinate of individual $\mathbf{x}$ in the kth objective, and m is the number of objectives.

GR can be considered as a natural tradeoff between the number of objectives for which one solution is better than another and the difference of values in a single objective between two solutions. On the one hand, if an individual performs better than its competitors in the majority of objectives, it would have a higher likelihood of obtaining a lower GR value. On the other hand, the difference in a single objective is also an important part of influencing the GR value. For instance, considering individuals **C** and **A** in Figure 4.3, **C** will obtain a worse GR value than **A** (6 against 4) since the advantage in f_2 is less than the disadvantage in f_1.

Note that the behaviour of GR is closely related to the shape of the Pareto front of a multi-objective problem; e.g., individuals around the centre of the Pareto front have good evaluations when the shape is convex, and individuals located in the edges of the Pareto front are preferable when the shape is concave. This may drive the population towards part of the optimal front, like the knee of the Pareto front. In the proposed algorithm, a GR adjustment strategy will be introduced to deal with this issue in the environmental selection process (Section 4.2.5).

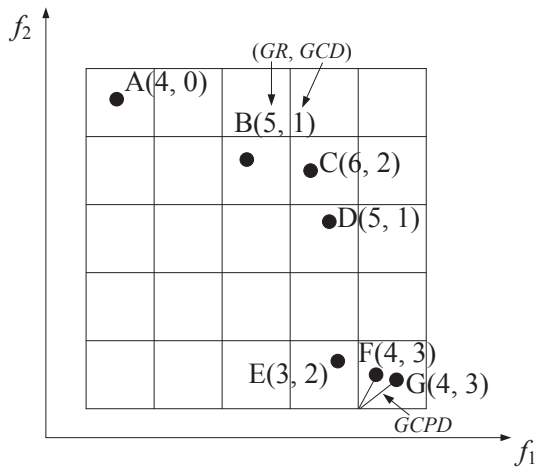


Figure 4.3: Illustration of fitness assignment. The numbers in the brackets associated with each solution correspond to GR and GCD, respectively.

Density estimation of solutions is an important issue in the fitness assignment process since a set of well-distributed solutions will play a crucial role in driving the search towards the entire Pareto front. However, the existing grid-based density estimators, which consider the number of solutions occupying a single hyperbox, may fail to reflect their distribution because of the exponential increase of the number of hyperboxes with the number of objectives. Here, we enlarge the range of regions considered and introduce the concept of the neighbours of a solution.

A solution $\mathbf{y}$ is regarded as a neighbour of a solution $\mathbf{x}$, if $GD(\mathbf{x}, \mathbf{y}) < m$, where $GD(\mathbf{x}, \mathbf{y})$ denotes the grid difference between $\mathbf{x}$ and $\mathbf{y}$, and m is the number of objectives. GrEA considers the distribution of neighbours of a solution with respect to its density estimation. Specifically, the density estimator, grid crowding distance (GCD), of $\mathbf{x}$ is defined as:

$$GCD(\mathbf{x}) = \sum_{\mathbf{y} \in N(\mathbf{y})} (m - GD(\mathbf{x}, \mathbf{y})) \tag{4.8}$$

where $N(\mathbf{x})$ stands for the set of neighbours of $\mathbf{x}$. For instance, in Figure 4.3, the neighbours of individual $\mathbf{G}$ are $\mathbf{E}$ and $\mathbf{F}$, and the GCD of $\mathbf{G}$ is 3, i.e., $(2-1)+(2-0) = 3$.

The GCD of a solution depends on both the range of neighbourhood (i.e., the region in which other solutions are regarded as its neighbours) and the grid difference between

it and other solutions. On the one hand, a larger neighbourhood range generally contains more solutions, thus contributing to a higher GCD value. Note that the neighbourhood range is determined by m. The number of considered hyperboxes will gradually increase with the number of objectives, which would be consistent with the total number of hyperboxes in the grid environment, hence providing a good distinction of the crowding degree among individuals. On the other hand, GCD also indicates the position information of solutions in the neighbourhood since the grid difference metric is involved. The farther the neighbours are located, the smaller the contribution to GCD is. For example, considering individuals **C** and **F** in Figure 4.3, the GCD of **C** is smaller than that of **F** (2 against 3) although the number of their neighbours is precisely equal.

Although GR and GCD have already provided a good measure of individuals in terms of proximity and diversity, they may still fail to discriminate individuals. Since their calculation is based on the grid coordinates of individuals, both GR and GCD have an integral value, which means that some individuals may have the same GR and GCD values, e.g., individuals **B** and **D** in Figure 4.3. Here, inspired by the strategy in ϵ-MOEA [54], we calculate the normalised Euclidean distance between an individual and the utopia point in its hyperbox (i.e., the best corner of its hyperbox), called grid coordinate point distance (GCPD), as follows:

$$GCPD(\mathbf{x}) = \sqrt{\sum_{k=1}^{m}((F_k(\mathbf{x}) - (lb_k + G_k(\mathbf{x}) \times d_k))/d_k)^2} \tag{4.9}$$

where $G_k(\mathbf{x})$ and $f_k(\mathbf{x})$ denote the grid coordinate and actual objective value of individual $\mathbf{x}$ respectively in the kth objective, lb_k and d_k stand for the lower boundary of the grid and the width of a hyperbox, respectively, for the kth objective, and m is the number of objectives. A lower GCPD is preferable. Individuals **F** and **G** in Figure 4.3 also illustrate this criterion.

According to the three grid-based criteria GR, GCD, and GCPD, the evolutionary status of individuals can be reflected effectively. In the following, we will employ these criteria to compare individuals in the selection process.

Algorithm 4.2 *TournamentSelection*

Require: individuals p, q randomly chosen from the population

if $p \prec q$ **or** $p \prec_{grid} q$ **then**
  return p
else if $q \prec p$ **or** $q \prec_{grid} p$ **then**
  return q
else if $GCD(p) < GCD(q)$ **then**
  return p
else if $GCD(q) < GCD(p)$ **then**
  return q
else if $random(0,1) < 0.5$ **then**
  return p
else
  return q
end if

4.2.4 Mating Selection

Mating selection which aims to make a good preparation for exchanging the information of individuals plays an important role in EMO algorithms. It is usually implemented by selecting promising solutions from the current population to form a mating pool. Here, we use a type of binary tournament selection strategy based on the dominance relation and density information to pick out individuals for variation.

Algorithm 4.2 gives a detailed procedure of this strategy. First, two individuals are randomly chosen from the population. If one Pareto-dominates or grid-dominates the other, the former is chosen. Otherwise, it indicates that these two solutions are non-dominated to each other in terms of both the Pareto dominance and grid dominance relations. In this case, we prefer the solution with a lower density estimation value (i.e., GCD). Finally, if GCD still fails to distinguish between the two solutions, the tie will be split randomly.

4.2.5 Environmental Selection

Environmental selection which aims to obtain a well-converged and well-distributed archive set is implemented by picking out the "best" solutions from the previous population and the newly created population. A straightforward way to do the selection is based on the fitness of solutions. However, a shortcoming of this way is that it may

lead to the loss of diversity since adjacent solutions often have similar fitness values. For example, solutions **E**, **F**, and **G** in Figure 4.3 have similar fitness values, and thus a high likelihood of being eliminated or preserved simultaneously. Here, we design a fitness adjustment mechanism to address this issue.

Fitness Adjustment

GrEA selects individuals by hierarchically comparing them according to the three fitness criteria: GR, GCD, and GCPD. GR is the primary criterion, GCD is regarded as the secondary one activated when the GR value of individuals is incomparable (i.e. equal), and when the first two criteria fail to discriminate individuals, the third one GCPD is used to break a tie. Here, we focus the adjustment on the primary criterion.

When an individual is selected into the archive, the GR value of its "related" individuals will be punished. However, how to implement the GR punishment operation (i.e., determine the "related" individuals and assign how much they would be punished) is not a trivial task. Several crucial factors need to be considered in order to achieve a good balance between proximity and diversity in the archive.

- A severe penalty should be imposed on individuals that have the same grid coordinate as the picked individual.
- The individuals grid-dominated by the picked individual should be punished more heavily than the individuals not grid-dominated by it. For instance, consider a set of individuals **A**, **B**, and **C** which have the grid coordinate $(0, 3)$, $(0, 5)$, and $(5, 0)$, respectively. Obviously, **C** is preferable to **B** after **A** has already entered the archive, because **C** is helpful to the evolution towards different directions.
- In order to further prevent crowding, the neighbours of the picked individual should be penalised, and the punishment degree should decline with the distance from them to the picked individual.
- When implementing penalty on the neighbours of the picked individual, the individuals grid-dominated by them may also need to be punished. For example, for a set of four individuals $\mathbf{A}(0, 0, 1)$, $\mathbf{B}(0, 1, 0)$, $\mathbf{C}(1, 0, 0)$, and $\mathbf{D}(3, 0, 0)$, assume

Algorithm 4.3 *GR_adjustment*(P, q)

Require: P (candidate set), q (picked individual), m (number of objectives), $PD(p)$ (maximum punishment degree of p), $E(q) := \{p \in P | GD(p, q) = 0\}$, $G(q) := \{p \in P | q \prec_{grid} p\}$, $NG(q) := \{p \in P | q \not\prec_{grid} p\}$, $N(q) := \{p \in P | GD(p, q) < m\}$

```
1: for all p ∈ E(q) do
2:    GR(p) ← GR(p) + (m + 2)
3: end for
4: for all p ∈ G(q) do
5:    GR(p) ← GR(p) + m
6: end for
7: for all p ∈ NG(q) ∧ p ∉ E(q) do
8:    PD(p) ← 0
9: end for
10: for all p ∈ N(q) ∩ NG(q) ∧ p ∉ E(q) do
11:    if PD(p) < m − GD(p, q) then
12:       PD(p) ← m − GD(p, q)
13:       for all r ∈ G(p) ∧ r ∉ G(q) ∪ E(q) do
14:          if PD(r) < PD(p) then
15:             PD(r) ← PD(p)
16:          end if
17:       end for
18:    end if
19: end for
20: for all p ∈ NG(q) ∧ p ∉ E(q) do
21:    GR(p) ← GR(p) + PD(p)
22: end for
```

that three individuals need to be selected into the archive. Apparently, the best choice is to eliminate the last individual **D**. However, individual **C** may fail to be selected after **A** and **B** are in the archive, since punishments were imposed on **C** as the neighbour of **A** and **B**, leading **C** to have a worse GR than **D**. Therefore, a punishment on the individuals grid-dominated by the neighbours of the picked individual is helpful, which can improve the proximity of the archive set largely.

Keeping the above factors in mind, a GR adjustment procedure is presented in Algorithm 4.3. From the procedure, individuals can be classified into three groups in the GR adjustment process: the individuals whose grid coordinate is equal to that of the picked individual (lines 1–3), the ones who are grid-dominated by the picked individual (lines 4–6), and the ones who are not grid-dominated by and have a different grid coordinate from the picked individual (lines 7–22). They correspond to the punishment degrees $m + 2$, m, and within $[0, m - 1]$, respectively, where m denotes the number of

Algorithm 4.4 *Environmental_selection*(P)

Require: N (archive size)
1: Generate an empty set Q for archive
2: $(F_1, F_2, ..., F_i, ...) \leftarrow Pareto\_nondominated\_sort(P)$
/* Partition P into different fronts $(F_1, F_2, ..., F_i, ...)$ by using the fast nondominated sorting approach and find the critical front F_i (i.e., $0 \leq N - |F_1 \cup F_2 \cup ... \cup F_{i-1}| < F_i$) */
3: $Q \leftarrow F_1 \cup F_2 \cup ... \cup F_{i-1}$
4: **if** $|Q| = N$ **then**
5: **return** Q
6: **end if**
7: $Grid\_setting(F_i)$ /* Set grid environment for F_i */
8: $Initialisation(F_i)$
9: **while** $|Q| < N$ **do**
10: $q \leftarrow Findout\_best(F_i)$
11: $Q \leftarrow Q \cup \{q\}$
12: $F_i \leftarrow F_i \setminus \{q\}$
13: $GCD\_calculation(F_i, q)$
14: $GR\_adjustment(F_i, q)$
15: **end while**
16: **return** Q

objectives. For individuals in the last group, a neighbour p of the picked individual q is imposed the punishment degree at least $m - GD(p, q)$ (lines 11 and 12), and correspondingly the individuals grid-dominated by p are imposed the punishment degree more than or equal to that of p (lines 13–17). This can prevent the individuals from being archived earlier than their better competitors in the sense of the grid dominance relation.

To sum up, by the fitness adjustment operation, GR will not be viewed as a simple proximity indicator, but rather a combination of information among proximity, density, and evolution direction of individuals in the archive set.

Main Procedure

Algorithm 4.4 shows the main procedure of environmental selection. Similar to NSGA-II [55], GrEA considers the critical Pareto nondominated front in the candidate set. The candidate solutions are divided into different fronts $(F_1, F_2, ..., F_i, ...)$ by using the fast nondominated sorting approach. The critical front F_i ($|F_1 \cup F_2 \cup ... \cup F_{i-1}| \leq N$ and $|F_1 \cup F_2 \cup ... \cup F_{i-1} \cup F_i| > N$, where N denotes the archive size) is found, and correspondingly the first $(i-1)$ nondominated fronts $(F_1, F_2, ..., F_{i-1})$ are moved into

Algorithm 4.5 *Initialization*(P)

```
1: for all p ∈ P do
2:     GR_assignment(p)          /* Assign GR according to equation (9) */
3:     GCPD_assignment(p)        /* Assign GCPD according to equation (11) */
4:     GCD(p) ← 0                /* Assign zero to GCD */
5: end for
```

Algorithm 4.6 *GCD_calculation*(P, q)

Require: P (candidate set), q (picked individual), $N(q) := \{p \in P | GD(p, q) < m\}$

```
1: for all p ∈ N(q) do
2:     GCD(p) ← GCD(p) + (m − GD(p, q))
3: end for
```

the archive (lines 2–6). In fact, since the solutions in many-objective problems are typically Pareto nondominated to each other, the critical front is often the first front, namely $i = 1$.

In Algorithm 4.4, function *Initialisation* (line 8) is used to initialise the information of individuals in the grid environment set in line 7. The fitness of individuals with regard to proximity (i.e., GR and GCPD) is calculated by Eqs. (4.7) and (4.9). It is necessary to point out that the initial density value of individuals (i.e., GCD) is assigned to zero in the function. Unlike the proximity estimator, which can be directly calculated by the own location of an individual, the diversity one has to be estimated by the relation to other individuals. It may be meaningless to consider the crowding relation among the individuals in the candidate set rather than in the archive set, since the latter is precisely the population to be preserved. Here, GrEA estimates the density of individuals by calculating their crowding degree in the archive (line 13). Algorithms 4.5 and 4.6 give the pseudocode of functions *Initialisation* and *GCD_calculation*, respectively.

Function *Findout_best* (line 10) in Algorithm 4.4 is designed to find out the best individual in the considered front. The pseudocode is shown in Algorithm 4.7. As stated previously, the function hierarchically compares the three criteria GR, GCD, and GCPD. A lower value is preferable in all the criteria.

The following example illustrates the working principle of the whole environmental selection process.

Algorithm 4.7 *Findout_best*(P)

Require: q (best solution in P), p_i (the ith solution in P)

```
1: q ← p_1
2: for i = 2 to |P| do
3:    if GR(p_i) < GR(q) then
4:       q ← p_i
5:    else if GR(p_i) = GR(q) then
6:       if GCD(p_i) < GCD(q) then
7:          q ← p_i
8:       else if GCD(p_i) = GCD(q) then
9:          if GCPD(p_i) < GCPD(q) then
10:            q ← p_i
11:         end if
12:      end if
13:   end if
14: end for
15: return q
```

An Example of Environmental Selection

Consider a four-objective scenario where a set of 12 candidate solutions **A**–**L** is the critical front to be archived. Their actual objective values are given in Figure 4.4.

Assume that the number of grid divisions is set to 6 and the archive size is 5. The grid coordinates of the individuals are calculated according to Eqs. (4.1)–(4.4), and the fitness is initialised by Algorithm 4.5. Figure 4.5 shows the environmental selection process of the individuals, and the grid coordinates and fitness values (GR, GCD, and GCPD) of individuals are also contained in the figure.

First, individual **A** is selected into the archive due to its best GCPD value (shown in Figure 4.5(b)), given that the other two criteria fail to completely distinguish between individuals **A**–**F**. Accordingly, individuals **C** and **H** are punished because the former has the same grid coordinate as **A** and the latter is grid-dominated by **A**. Their punishment degree is $m+2$ and m, respectively (i.e., $GR(\mathbf{C}) = GR(\mathbf{C}) + (m+2)$ and

A (5.0, 2.5, 0.0, 0.0)	**B** (0.0, 5.0, 2.5, 0.0)	**C** (4.8, 2.6, 0.0, 0.3)
D (0.0, 0.0, 5.0, 2.5)	**E** (0.0, 3.2, 4.1, 0.0)	**F** (2.5, 0.0, 0.0, 5.0)
G (1.2, 1.1, 0.0, 5.9)	**H** (6.3, 1.9, 0.0, 0.3)	**I** (0.8, 3.4, 3.2, 0.9)
J (0.8, 0.3, 6.1, 1.8)	**K** (2.3, 2.3, 2.3, 2.3)	**L** (0.4, 6.2, 1.7, 3.0)

Figure 4.4: A set of 4-objective individuals for archiving. The numbers in the brackets correspond to their objective values.

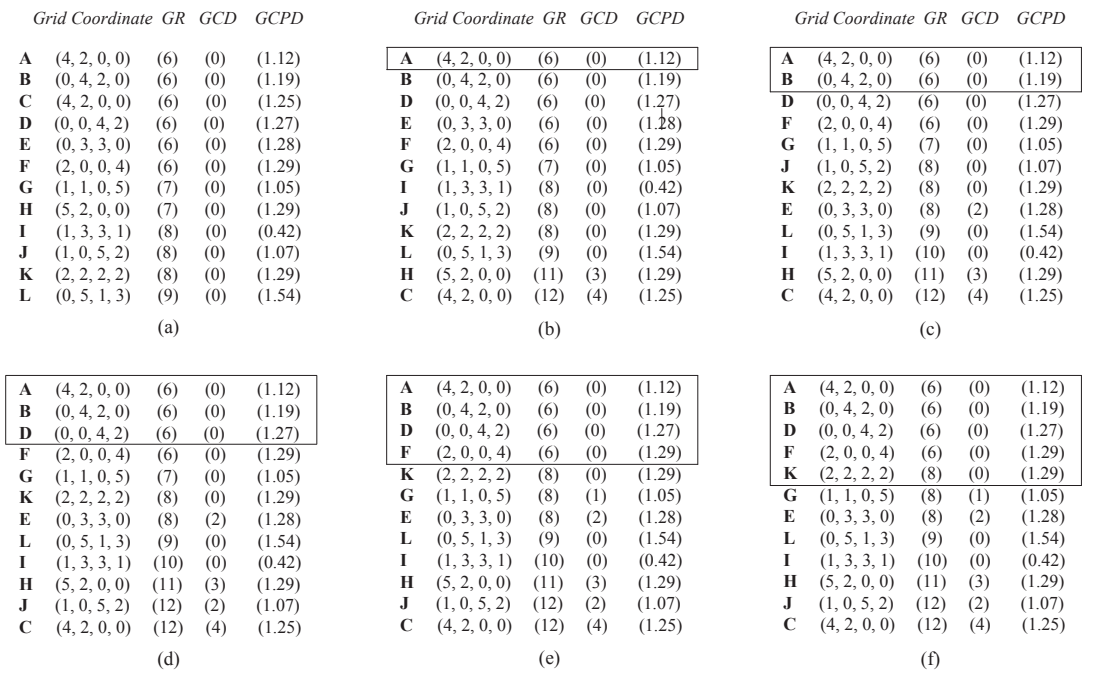


Figure 4.5: An illustration of the environmental selection process. Individuals are arranged in the order of their fitness values for observation. The framed individuals mean that they have entered the archive set. The archive size is set to 5.

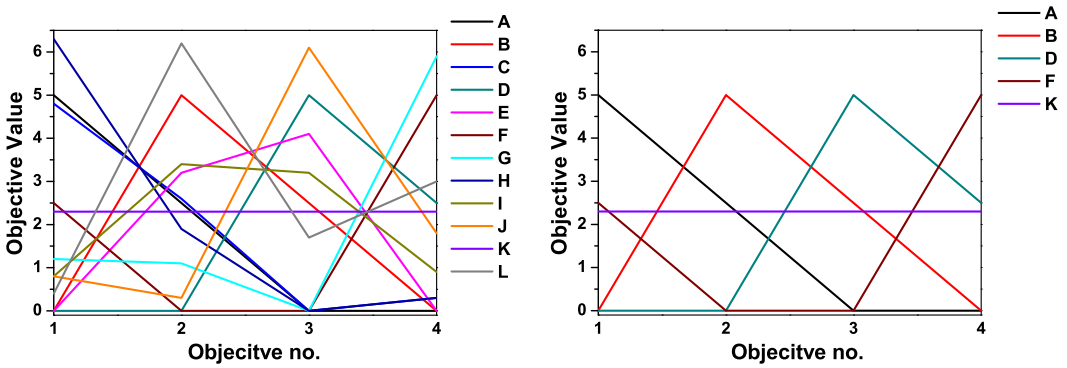
$GR(\mathbf{H}) = GR(\mathbf{H}) + m$). In addition, their GCD is also updated by Algorithm 4.6.

Second, individual **B** is picked out (Figure 4.5(c)). As the only neighbour of **B**, **E** is punished, and correspondingly **I** is also imposed a punishment since it is grid-dominated by **E**. Their punishment values are the same ($GR(\mathbf{E}) = GR(\mathbf{E}) + (m - GD(\mathbf{E}, \mathbf{B}))$ and $GR(\mathbf{I}) = GR(\mathbf{I}) + (m - GD(\mathbf{E}, \mathbf{B}))$). In addition, an update of GCD is made for **E**.

Then, individual **D** is chosen (Figure 4.5(d)). **J** is its neighbour and grid-dominated by it. So, $GR(\mathbf{J}) = GR(\mathbf{J}) + m$ and $GCD(\mathbf{J}) = GCD(\mathbf{J}) + (m - GD(\mathbf{J}, \mathbf{D}))$.

Next, individual **F** enters the archive set (Figure 4.5(e)). Accordingly, the GR and GCD of individual **G**, the only neighbour of **F**, are adjusted by adding $m - GD(\mathbf{G}, \mathbf{F})$ to each of them.

Finally, individual **K** is selected into the archive set (Figure 4.5(f)), and the fitness of all individuals in the candidate set remains unchanged since none of them is a neighbour of **K** or grid-dominated by it.



(a) Original candidate set (b) Final archive set

Figure 4.6: Distribution of the solution set for the 4-objective example by parallel coordinates.

In order to compare the results, we plot the original candidate set and the final archive set of the above example using parallel coordinates in Figure 4.6. As can be seen from the figure, after environmental selection a group of well-approximated and well-distributed solutions is obtained out of the original candidate set.

4.3 Experimental Results

This section is devoted to the performance verification of the proposed algorithm. We first introduce experimental settings in experimental studies. Then, we compare GrEA with five well-established algorithms: ϵ-MOEA [54], POGA [59], HypE [10], MSOPS [106] and MOEA/D [278], which are designed specially for many-objective problems or have been demonstrated to be promising in many-objective optimisation [256, 88, 180]. Next, the effect of the grid division parameter in GrEA is investigated. Finally, we analyse the time complexity of the proposed algorithm and show its computational cost.

4.3.1 Experimental Settings

In the experimental studies, three well-defined test suites, the DTLZ [57], DTLZ5(I, m) [234] and multi-objective travelling salesman problem (TSP) [45], are considered. All these problems can be scaled to any number of objectives and decision variables. In

Table 4.1: Settings of the test problems

Name	Number of Objectives (m)	Number of Variables (n)	Parameter
DTLZ1	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 5$
DTLZ2	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 10$
DTLZ3	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 10$
DTLZ4	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 10$
DTLZ5	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 10$
DTLZ6	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 10$
DTLZ7	$4, 5, 6, 8, 10$	$m - 1 + k$	$k = 20$
DTLZ5(I, m)	10	$m - 1 + k$	$k = 10$, $I = 3, 4, 5, 6, 7, 8, 9$
TSP	$5, 10$	30	$TSPcp = -0.4, -0.2, 0, 0.2, 0.4$

this study, we divided them into four groups according to their characteristics of testing different abilities of algorithms.

For the DTLZ suite, the problems DTLZ1 to DTLZ7 can be classified into two groups. One group, consisting of DTLZ2, DTLZ4, DTLZ5 and DTLZ7, is used to test the ability of an algorithm to cope with the MOPs with different shapes and locations. The other group, consisting of DTLZ1, DTLZ3 and DTLZ6, creates more obstacles for an algorithm to converge into the Pareto front [57].

The DTLZ5(I, m) suite, originating from DTLZ5, is a set of test problems where the actual dimensionality I of the Pareto front against the original number m of objectives in the problem can be controlled by the user. In these problems, all objectives within $\{f_1, ..., f_{m-I+1}\}$ are positively correlated, while the objectives in $\{f_{m-I+2}, ..., f_m\}$ are conflicting with each other.

The multi-objective TSP is a typical combinatorial optimisation problem (see Chapter 2.2.4 for details). In multi-objective TSP, $TSPcp \in (-1, 1)$ is a simple "correlation parameter". When $TSPcp < 0$, $TSPcp = 0$, or $TSPcp > 0$, it introduces negative, zero, or positive interobjective correlations, respectively. In our study, $TSPcp$ is assigned to -0.4, -0.2, 0, 0.2, and 0.4 respectively to represent different characteristics of the problem.

The summary of settings for all the test problems is shown in Table 4.1.

In order to compare the performance of the peer algorithms, two widely-used indicators, inverted generational distance (IGD) [19, 42] and hypervolume (HV) [294], are considered. The former requires a reference set of representing the Pareto front and

Table 4.2: Parameter settings in GrEA and ϵ-MOEA, where m is the number of objectives

Problem	$m=4$		$m=5$		$m=6$		$m=8$		$m=10$		Problem	$m=10$		Problem	$m=5$		$m=10$	
	div	ϵ	div	ϵ	div	ϵ	div	ϵ	div	ϵ	$(I,\ m)$	div	ϵ	$(TSPcp)$	div	ϵ	div	ϵ
DTLZ1	10	0.0520	10	0.0590	10	0.0554	10	0.0549	11	0.0565	DTLZ5(3,10)	9	0.06	TSP(−0.4)	12	2.4	11	6.5
DTLZ2	10	0.1312	9	0.1927	8	0.2340	7	0.2900	8	0.3080	DTLZ5(4,10)	10	0.12	TSP(−0.2)	11	1.9	10	4.8
DTLZ3	11	0.1385	11	0.2000	11	0.2270	10	0.1567	11	0.8500	DTLZ5(5,10)	10	0.16	TSP(0)	11	1.4	10	3.8
DTLZ4	10	0.1312	9	0.1927	8	0.2340	7	0.2900	8	0.3080	DTLZ5(6,10)	10	0.2	TSP(0.2)	11	0.95	10	2.6
DTLZ5	35	0.0420	29	0.0785	14	0.1100	11	0.1272	11	0.1288	DTLZ5(7,10)	8	0.24	TSP(0.4)	10	0.6	10	1.8
DTLZ6	36	0.1200	24	0.3552	50	0.7500	50	1.1500	50	1.4500	DTLZ5(8,10)	8	0.25					
DTLZ7	9	0.1050	8	0.1580	6	0.1500	5	0.2250	4	0.5600	DTLZ5(9,10)	9	0.26					

is used to evaluate algorithms on DTLZ and DTLZ5(I, m) since their optimal fronts are known. The latter is used to assess the performance of algorithms on TSP whose Pareto front is unknown.

A reference point is required in the HV calculation. Here, we regard the point with the integer value 22 for each objective as the reference point, given that it is slightly larger than the worst value of all the obtained solution sets. In addition, since the exact calculation of the hypervolume metric is infeasible for a solution set with 10 objectives, we approximately estimate the hypervolume result of a solution set by the Monte Carlo sampling method used in HypE [10]. According to [10], 10^7 sampling points are used.

General parameter settings for all experiments are given as follows unless otherwise mentioned.

- **Parameter setting for crossover and mutation**: A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n denotes the number of decision variables) were used. For DTLZ and DTLZ5(I, m), the operators for crossover and mutation are simulated binary crossover (SBX) and polynomial mutation with both distribution indexes 20 (i.e., $\eta_c = 20$ and $\eta_m = 20$) [50, 10]. As to the multi-objective TSP, the order crossover and inversion operator [194] are chosen as crossover and mutation operators, respectively.

- **Number of runs and stopping condition**: We independently run each algorithm 30 times on each test problem. The termination criterion of an algorithm is a predefined number of evaluations. For the first problem group (DTLZ2, DTLZ4, DTLZ5 and DTLZ7) and the third problem group (DTLZ5(I, m)), it was set to 30,000, and for the second group (DTLZ1, DTLZ3, and DTLZ6) and the fourth

group (TSP), it was set to 100,000.

- **Population and archive size**: For general EMO algorithms, the population size was set to 100, and the archive was also maintained with the same size if required. Note that the population size in MOEA/D is the same as the number of weight vectors. Due to the combinatorial nature of uniformly distributed weight vectors, the population size cannot be arbitrarily specified. Here, we use the closest integer to 100 among the possible values as the population size (i.e., 120, 126, 126, 120, and 55 for 4-, 5-, 6-, 8-, and 10-objective problems, respectively). In ϵ-MOEA, the population size is determined by the ϵ value. In order to guarantee a fair comparison, we set ϵ so that the archive of ϵ-MOEA is approximately of the same size as that of the other algorithms (shown in Table 4.2).
- **Parameter setting in MSOPS, MOEA/D, and GrEA**: In MSOPS, the number of weight vectors was set to 100 as suggested in [256]. Following the practice in [119], the Tchebycheff function in MOEA/D was selected as the scalarizing function and the neighbourhood size was specified as 10% of the population size. In GrEA, the setting of grid division div is shown in Table 4.2. A detailed study of div will be given in Section V-B.

4.3.2 Performance Comparison

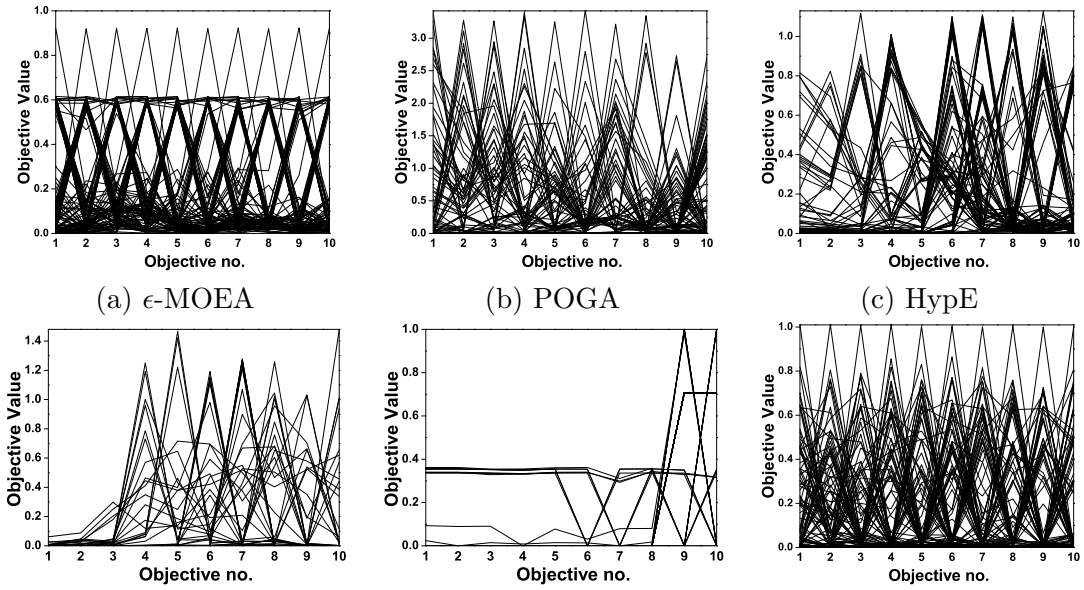
DTLZ2, DTLZ4, DTLZ5 and DTLZ7 Problems

Table 4.3 shows the IGD results of the six EMO algorithms on DTLZ2, DTLZ4, DTLZ5, and DTLZ7. The values in the table are the mean and standard deviation. The best mean for each problem is shown with a grey background.

Table 4.3 shows the IGD results in terms of the mean and standard deviation over 30 runs of the six EMO algorithms on DTLZ2, DTLZ4, DTLZ5 and DTLZ7, where the best mean for each problem is shown with a grey background. As can be seen from the table, GrEA performs best on DTLZ2 and DTLZ7 with respect to all the considered numbers of objectives. For the other problems, MSOPS and GrEA have their own strengths. For DTLZ4, MSOPS obtains the best IGD value in the

Table 4.3: IGD results of the six algorithms on DTLZ2, DTLZ4, DTLZ5, and DTLZ7

Problem	Obj.	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
DTLZ2	4	1.348E−1 (2.4E−3)	1.486E−1 (6.7E−3)	2.452E−1 (3.9E−2)	1.757E−1 (1.2E−2)	2.124E−1 (2.1E−3)	1.271E−1 (2.5E−3)
	5	1.945E−1 (1.2E−2)	2.758E−1 (1.8E−2)	4.245E−1 (7.5E−2)	3.625E−1 (2.8E−2)	2.674E−1 (1.1E−3)	1.750E−1 (2.9E−3)
	6	3.161E−1 (6.7E−3)	5.882E−1 (5.8E−2)	4.738E−1 (5.7E−2)	3.698E−1 (2.2E−2)	4.132E−1 (1.9E−2)	2.985E−1 (5.2E−3)
	8	4.408E−1 (1.3E−2)	9.903E−1 (9.3E−2)	6.147E−1 (5.0E−2)	5.681E−1 (3.6E−2)	6.365E−1 (7.1E−2)	3.957E−1 (4.6E−3)
	10	5.323E−1 (2.4E−2)	1.162E+0 (1.0E−1)	6.972E−1 (5.5E−2)	7.658E−1 (4.3E−2)	7.257E−1 (6.5E−2)	4.842E−1 (2.9E−3)
DTLZ4	4	4.150E−1 (2.7E−1)	2.663E−1 (6.8E−2)	4.976E−1 (3.4E−1)	1.449E−1 (4.4E−3)	5.382E−1 (2.8E−1)	1.913E−1 (1.1E−1)
	5	6.330E−1 (3.4E−1)	3.970E−1 (7.0E−2)	7.018E−1 (2.8E−1)	3.147E−1 (3.2E−2)	5.857E−1 (3.0E−1)	2.154E−1 (9.7E−2)
	6	6.035E−1 (1.7E−1)	9.541E−1 (1.6E−1)	6.672E−1 (1.0E−1)	3.685E−1 (1.1E−2)	6.498E−1 (1.6E−1)	3.007E−1 (4.6E−3)
	8	6.459E−1 (1.1E−1)	1.175E+1 (2.2E−1)	9.199E−1 (6.1E−2)	5.400E−1 (2.5E−2)	7.600E-1 (8.6E−2)	4.020E−1 (3.2E−3)
	10	6.267E−1 (9.2E−2)	8.668E−1 (2.7E−1)	1.074E+0 (6.1E−2)	8.178E−1 (4.5E−2)	8.311E−1 (8.4E−2)	4.928E−1 (3.9E−3)
DTLZ5	4	4.819E−2 (5.3E−3)	5.217E−2 (9.8E−3)	1.216E−1 (4.1E−2)	3.016E−2 (3.0E−3)	2.563E−2 (1.1E−4)	1.846E−2 (3.3E−3)
	5	8.956E−2 (7.5E−3)	7.437E−1 (1.6E−2)	1.459E−1 (5.3E−2)	3.002E−2 (4.1E−3)	4.544E−2 (1.5E−3)	4.333E−2 (2.2E−2)
	6	1.278E−1 (1.1E−2)	7.467E−1 (9.8E−4)	1.734E−1 (5.9E−2)	1.859E−2 (1.7E−3)	6.947E−2 (4.3E−3)	9.575E−2 (1.6E−2)
	8	1.589E−1 (2.1E−2)	7.473E−1 (1.3E−3)	1.754E−1 (6.9E−1)	2.525E−2 (2.5E−3)	1.088E−1 (7.4E−3)	2.327E−1 (3.5E−2)
	10	1.690E−1 (2.1E−2)	1.219E+0 (7.7E−1)	1.560E−1 (4.8E−2)	4.025E−2 (3.9E−3)	1.959E−1 (1.3E−2)	3.462E−1 (6.1E−2)
DTLZ7	4	3.492E−1 (1.8E−1)	2.151E−1 (8.6E−3)	4.846E−1 (1.9E−1)	1.547E+0 (4.1E−1)	5.152E−1 (7.2E−2)	1.897E−1 (6.8E−3)
	5	6.310E−1 (2.1E−1)	4.127E−1 (1.6E−2)	8.972E−1 (2.1E−1)	7.581E+0 (1.3E+0)	6.444E−1 (8.7E−2)	3.238E−1 (1.0E−2)
	6	5.856E−1 (1.9E−1)	6.608E−1 (3.0E−2)	9.894E−1 (1.9E−1)	1.085E+1 (2.5E+0)	7.551E−1 (6.1E−2)	4.888E−1 (1.6E−2)
	8	8.971E−1 (5.1E−1)	2.293E+0 (4.1E−1)	1.065E+0 (4.1E−2)	1.945E+1 (2.0E+0)	1.064E+0 (1.3E−1)	7.643E−1 (3.5E−2)
	10	1.180E+0 (3.6E−1)	4.211E+0 (7.5E−1)	1.224E+0 (8.1E−2)	2.670E+1 (3.4E+0)	1.546E+0 (2.0E−1)	1.057E+0 (3.8E−2)



(a) ϵ-MOEA (b) POGA (c) HypE

(d) MSOPS (e) MOEA/D (f) GrEA

Figure 4.7: The final solution set of the six algorithms on the ten-objective DTLZ2, shown by parallel coordinates.

low-dimension objective space, and GrEA outperforms the other algorithms when the number of objectives is larger than four. For DTLZ5, GrEA performs well on the 4-objective problem, and MSOPS obtains better results with the increase of the number of objectives. For a visual comparison, Figure 4.7 shows the typical distribution of the six algorithms on the 10-objective DTLZ2 by parallel coordinates.

Note that the MOEA/D algorithm, which has recently been found to be very successful in the literature, obtains a worse IGD value than GrEA, MSOPS, and ϵ-MOEA generally. In fact, the solution set obtained by MOEA/D is very close to the Pareto front for some test problems, such as DTLZ2 and DTLZ4. Yet, it has a worse coverage of the Pareto front than that obtained by GrEA, MSOPS, and ϵ-MOEA for most of the problems, thereby resulting in a worse IGD value on these problems. This occurrence may be attributed to the decomposition-based selection operation in MOEA/D. Although a set of uniformly-distributed weight vectors is selected to specify the search targets (i.e., the points on the Pareto front), it cannot ensure that these points are located uniformly, especially for some problems with irregular Pareto front. In addition,

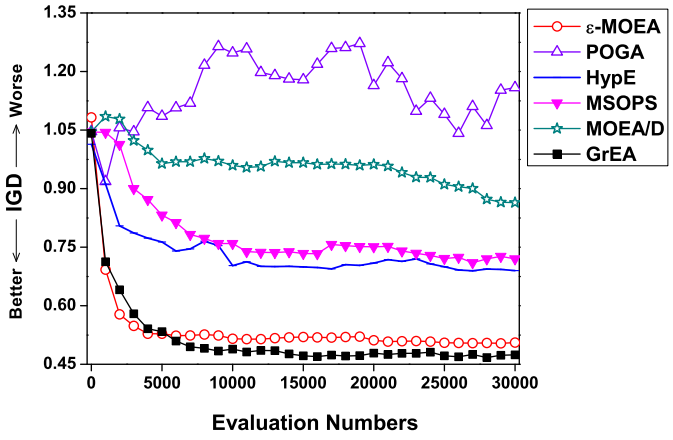


Figure 4.8: Evolutionary trajectories of IGD for the six algorithms on the ten-objective DTLZ2.

the Tchebycheff-based scalarizing function may not be a good tool to maintain diversity of solutions. Many weight vectors may correspond to only one Pareto optimal point by this scalarizing function [52]. Similar observations have been reported in [119, 52].

Further studies with these algorithms have been conducted to exhibit their evolutionary trajectories. Figure 4.8 plots the performance trajectories of IGD for the six algorithms on the 10-objective DTLZ2. As shown, GrEA performs better than the other five algorithms. Although ϵ-MOEA outperforms GrEA in the initial stage of evolution, the latter exceeds the former at around 6,000 evaluations and keeps a clear advantage until the end.

DTLZ1, DTLZ3 and DTLZ6 Problems

The IGD results of the six EMO algorithms on this problem group are shown in Table 4.4. As shown, GrEA and MOEA/D generally outperform the other four algorithms. They obtain the best IGD value in 6 and 7 out of 15 test instances, respectively. Specifically, GrEA performs best on the 8-objective DTLZ3, 4-objective DTLZ6, and DTLZ1 for all the considered numbers of objectives except 10. MOEA/D performs best on the 10-objective DTLZ1, 6- and 10-objective DTLZ3, and DTLZ6 with the number of objectives larger than 4. For the rest of the problems (i.e. the 4- and 5-objective DTLZ3), ϵ-MOEA obtains the best result.

Table 4.4: IGD results of the six algorithms on DTLZ1, DTLZ3, and DTLZ6

Problem	Obj.	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
DTLZ1	4	4.866E−2 (2.3E−3)	9.301E−1 (3.7E−2)	1.355E−1 (7.1E−2)	5.762E−2 (3.0E−3)	9.653E−2 (1.1E−4)	4.624E−2 (5.3E−3)
	5	6.930E−2 (8.1E−3)	4.470E+0 (4.3E+0)	3.107E−1 (3.8E−1)	8.673E−2 (3.8E−3)	1.192E−1 (1.3E−4)	6.257E−2 (7.5E−3)
	6	9.877E−2 (8.4E−2)	1.284E+1 (1.7E+1)	5.554E−1 (5.9E−1)	1.808E−1 (1.1E−1)	1.377E−1 (6.4E−3)	8.543E−2 (1.0E−2)
	8	3.069E−1 (3.4E−1)	1.012E+1 (6.9E+0)	1.017E+0 (1.4E+0)	7.770E−1 (6.8E-1)	1.852E−1 (5.8E−3)	1.060E−1 (4.8E−3)
	10	4.071E−1 (3.8E−1)	1.370E+1 (9.2E+0)	1.485E+0 (1.8E+0)	1.623E+0 (1.0E+0)	2.188E−1 (6.4E−3)	2.864E−1 (1.0E−1)
DTLZ3	4	1.417E−1 (8.9E−3)	1.317E+1 (5.7E+0)	1.029E+0 (7.1E−1)	1.072E+1 (6.8E+0)	2.130E−1 (1.7E−3)	1.522E−1 (4.7E−2)
	5	2.267E−1 (3.3E−2)	2.341E+1 (1.1E+1)	4.708E+0 (5.6E+0)	2.896E+1 (1.5E+1)	2.677E−1 (7.8E−4)	2.804E−1 (8.3E−2)
	6	4.578E−1 (1.4E−1)	3.239E+1 (1.2E+1)	2.689E+0 (1.7E+0)	4.665E+1 (1.6E+1)	4.085E−1 (3.2E−2)	4.368E−1 (1.5E−1)
	8	1.122E+1 (1.4E+1)	2.733E+1 (1.2E+1)	7.476E+0 (9.6E+0)	6.095E+1 (1.9E+1)	6.106E−1 (7.5E−2)	5.546E−1 (2.3E−1)
	10	2.040E+1 (2.7E+1)	3.157E+1 (1.4E+1)	6.226E+0 (6.2E+0)	6.312E+1 (1.8E+1)	6.599E−1 (5.5E−2)	7.743E−1 (2.5E−1)
DTLZ6	4	4.671E−1 (2.8E−2)	2.226E+0 (3.0E−1)	3.919E+0 (6.5E−1)	4.178E+0 (6.4E−1)	8.015E−2 (2.7E−2)	7.045E−2 (3.1E−2)
	5	1.678E+0 (1.5E−1)	1.801E+0 (4.5E−1)	5.431E+0 (5.6E−1)	6.560E+0 (5.0E−1)	1.194E−1 (3.8E−2)	1.429E−1 (4.3E−2)
	6	2.705E+0 (2.8E−1)	2.245E+0 (6.0E−1)	5.679E+0 (5.7E−1)	6.862E+0 (5.1E−1)	1.569E−1 (3.7E−2)	4.544E−1 (9.1E−2)
	8	1.987E+0 (1.3E+0)	5.807E+0 (3.9E+0)	6.165E+0 (6.2E−1)	6.813E+0 (4.4E−1)	1.830E−1 (2.8E−2)	5.971E−1 (3.8E−1)
	10	3.737E+0 (2.0E+0)	8.941E+0 (1.4E−1)	6.428E+0 (4.1E−1)	6.728E+0 (4.8E−1)	2.692E−1 (3.0E−2)	9.432E−1 (7.8E−1)

It is interesting to note that MOEA/D, unlike on the first group of test problems, is very competitive on this group of test problems which provide more obstacles for an algorithm to converge into the Pareto front. An important reason is that in contrast to some of the other algorithms (such as POGA, HypE and MSOPS) whose solution set fails to approximate the Pareto front, MOEA/D still works well in terms of proximity for most of the problems in this group, thereby obtaining better IGD results.

DTLZ5(I, m) Problem

In this section, we concentrate on the DTLZ5(I, m) problem suite, which tests the ability of an EMO algorithm to find a lower-dimensional Pareto front while working with a higher-dimensional objective space. Table 4.5 shows the IGD results of the six algorithms on DTLZ5($I, 10$), where I ranges from 3 to 9.

As shown in Table 4.5, GrEA and ϵ-MOEA perform better than the other four algorithms. Specifically, ϵ-MOEA outperforms the other algorithms when I is equal to 3 or 4, and GrEA performs best for the rest of the problems. MOEA/D has poor proximity of the Pareto front, leading to a worse IGD result than GrEA and ϵ-MOEA. Figure 4.9 gives a typical distribution of the six algorithms on DTLZ5$(6, 10)$ for observation. Although a few solutions of the set fail to reach the Pareto front, GrEA has a good approximation and coverage of the Pareto front.

Multi-objective TSP Problem

One important property of the multi-objective TSP is that the conflict degree among the objectives can be adjusted according to the parameter $TSPcp \in (-1, 1)$, where a lower value means a greater degree of conflict. From the HV results shown in Table 4.6, it can be seen that GrEA significantly outperforms the other algorithms for all the cases.

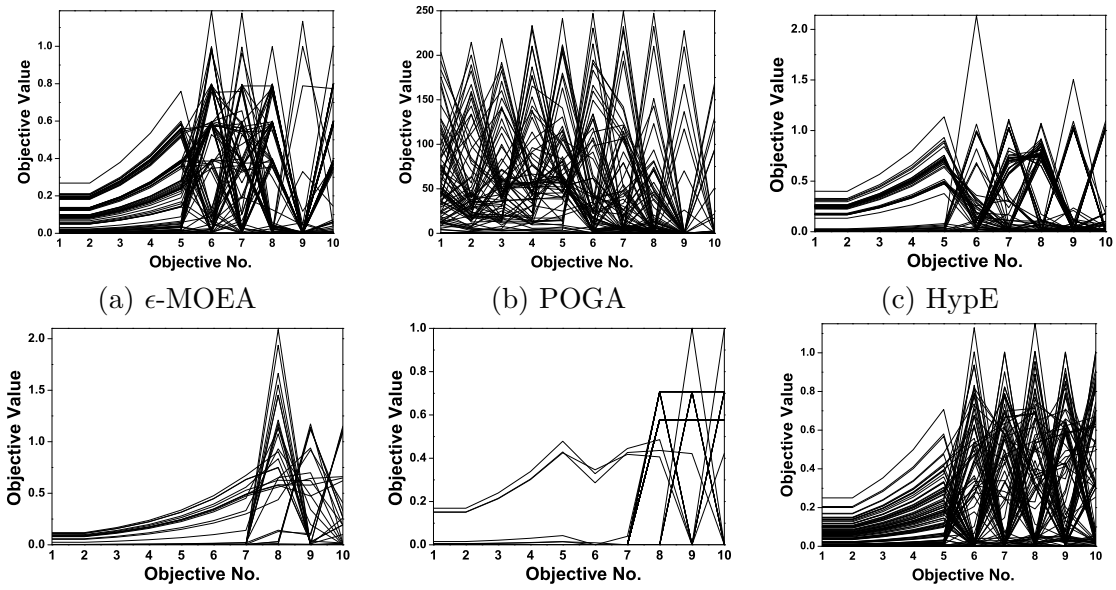
In addition, for this combinatorial optimisation problem the peer algorithms show some different behaviour from for the continuous ones. POGA, which generally performs worst in the previous test problems, works well when a greater conflict degree among the objectives of TSP is involved. ϵ-MOEA performs well for both the 5- and 10-objective problems, and HypE always obtains the worst HV value for all the 10 instances. MSOPS, which does not work very well in the 5-objective TSP, takes the

Table 4.5: IGD results of the six algorithms on DTLZ5(I, m), where $m = 10$.

I	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
3	6.435E−2 (2.2E−3)	1.810E+1 (1.9E+1)	2.068E−1 (2.5E−2)	5.433E−1 (5.1E−2)	3.060E−1 (8.7E−3)	1.418E−1 (6.7E−2)
4	1.402E−1 (1.7E−2)	2.271E+1 (1.7E+1)	2.997E−1 (2.9E−2)	3.291E−1 (9.7E−4)	3.586E−1 (3.3E−2)	1.528E−1 (8.8E−3)
5	2.605E−1 (1.1E−1)	1.354E+1 (1.7E+1)	3.923E−1 (9.4E−2)	2.982E−1 (9.4E−3)	3.753E−1 (5.0E−2)	1.932E−1 (1.4E−2)
6	4.024E−1 (1.2E−1)	8.689E+0 (9.6E+0)	6.885E−1 (1.2E−1)	6.031E−1 (1.1E−1)	5.838E−1 (7.5E−2)	3.214E−1 (1.5E−2)
7	4.894E−1 (1.3E−1)	1.061E+1 (1.1E+1)	7.911E−1 (1.0E−1)	7.934E−1 (1.1E−1)	6.343E−1 (8.6E−2)	3.808E−1 (1.1E−2)
8	5.119E−1 (9.0E−2)	8.520E+1 (9.8E+0)	8.584E−1 (8.9E−2)	9.273E−1 (1.1E−1)	6.520E−1 (6.5E−2)	4.352E−1 (6.8E−2)
9	5.694E−1 (1.1E−2)	5.823E+1 (9.0E+0)	9.180E−1 (7.5E−2)	1.062E+0 (1.3E−1)	7.411E−1 (7.6E−2)	4.564E−1 (5.1E−3)

Table 4.6: HV results of the six algorithms on the multi-objective TSP.

Obj.	TSP_{cp}	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
5	–0.4	1.125E+6 (5.3E+4)	1.128E+6 (5.2E+4)	1.962E+5 (5.3E+4)	7.963E+5 (4.5E+4)	9.615E+5 (5.8E+4)	1.231E+6 (4.8E+4)
	–0.2	1.014E+6 (4.5E+4)	1.012E+6 (3.7E+4)	2.493E+5 (4.8E+4)	8.048E+5 (4.2E+4)	9.687E+5 (4.8E+4)	1.121E+6 (3.8E+4)
	0	9.211E+5 (2.7E+4)	8.445E+5 (3.2E+4)	3.596E+5 (4.6E+4)	7.837E+5 (3.0E+4)	8.729E+5 (2.8E+4)	9.844E+5 (3.5E+4)
	2	8.543E+5 (3.1E+4)	7.565E+5 (3.2E+4)	4.251E+5 (6.7E+4)	7.651E+5 (3.4E+4)	7.530E+5 (3.1E+4)	8.836E+5 (2.3E+4)
	4	8.177E+5 (2.1E+4)	7.206E+5 (4.8E+4)	5.259E+5 (5.5E+4)	7.737E+5 (2.5E+4)	7.255E+5 (4.3E+4)	8.466E+5 (2.4E+4)
10	–0.4	1.026E+11 (1.4E+10)	2.520E+11 (5.7E+10)	3.634E+09 (1.7E+09)	1.706E+11 (2.4E+10)	1.539E+10 (3.2E+09)	3.772E+11 (1.6E+10)
	–0.2	1.203E+11 (1.3E+10)	1.761E+11 (4.9E+10)	1.198E+10 (3.7E+09)	1.820E+11 (1.8E+10)	3.376E+10 (5.1E+09)	3.097E+11 (1.0E+10)
	0	1.260E+11 (1.5E+10)	1.181E+11 (4.1E+10)	2.386E+10 (7.0E+09)	1.704E+11 (1.2E+10)	5.458E+10 (8.8E+09)	2.551E+11 (8.3E+09)
	2	1.400E+11 (1.2E+10)	8.500E+10 (2.5E+10)	3.404E+10 (8.3E+09)	1.529E+11 (1.0E+10)	7.450E+10 (1.1E+10)	2.136E+11 (5.9E+09)
	4	1.467E+11 (1.1E+10)	8.095E+10 (1.7E+10)	5.298E+10 (9.2E+09)	1.447E+11 (6.8E+09)	1.068E+11 (1.3E+10)	1.818E+11 (6.8E+09)



(a) ϵ-MOEA (b) POGA (c) HypE

(d) MSOPS (e) MOEA/D (f) GrEA

Figure 4.9: The final solution set of the six algorithms on DTLZ5(6,10), shown by parallel coordinates.

second place for most of the problems with 10 objectives. The result of MOEA/D for the 10-objective TSP is not as good as that for the 5-objective problem, with a worse HV value than MSOPS and ϵ-MOEA.

In order to demonstrate the evolutionary process of the six algorithms, Figure 4.10 plots their trajectories of HV during 100,000 evaluations on the 5-objective TSP with $TSPcp = -0.2$. As can be seen from the figure, the HV trajectory of GrEA rapidly increases in the initial stage of evolution, and keeps a clear advantage over the other algorithms during the whole evolutionary process.

Overall, from the study on the problems with different characteristics, we can conclude that the proposed algorithm has been successful in providing a balance between proximity and diversity in many-objective optimisation. GrEA outperforms the other five state-of-the-art algorithms in 36 out of all 52 test instances.

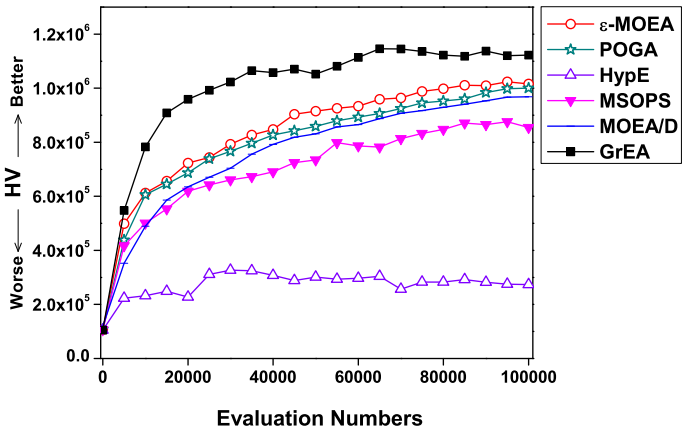


Figure 4.10: Evolutionary trajectories of HV for the six algorithms on the five-objective TSP, where $TSPcp = -0.2$.

4.3.3 Study of Different Parameter Configurations

In GrEA, a parameter, grid division (div), is introduced to divide the grid environment. This section investigates the effect of div and provides an appropriate setting for the user. Here, we show the results for the DTLZ2 problem. Similar results can be obtained for other test cases.

To study the sensitivity of the proposed algorithm to div, we repeat the experiments carried out in the previous section for $div \in [5, 50]$ with a step size 1. All other control parameters are kept unchanged. In addition, we expand the number of objectives for the problem to make a clearer observation of GrEA's performance with the variation of div. Figure 4.11 shows the results of IGD for different divisions on the problem with 3, 4, 5, 6, 8, 10, 12 and 15 objectives.

It is clear from the figure that the IGD value, in general, varies regularly with the number of divisions. In most cases, the trajectory of performance ranging from divisions 5 to around 9 rapidly descends and then gradually rises until the boundary. Moreover, the sensitivity of the algorithm increases with the number of objectives. For the number of objectives under eight, the performance trajectory remains smooth, and the algorithm appears to perform well during a segment of the range of divisions. When the number of objectives is larger than eight, the effect of division becomes

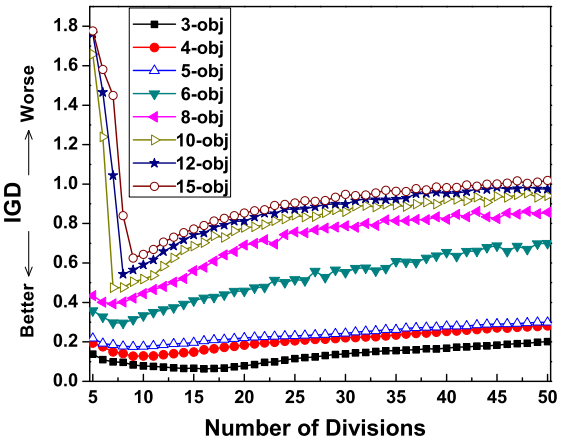


Figure 4.11: IGD of GrEA with different number of divisions on DTLZ2.

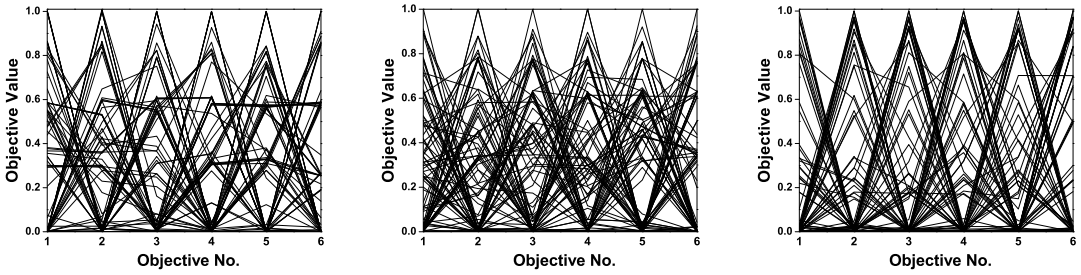
more apparent. A slight variation of *div* may result in a big change of the algorithm's performance. This indicates that a more careful setting of divisions should be conducted for a problem with a larger number of objectives.

On the other hand, considering the most suitable divisions for different numbers of objectives, similar results are obtained in general. GrEA performs best at 16, 9, 9, 8, 7, 7, 8 and 9 divisions for the 3-, 4-, 5-, 6-, 8-, 10-, 12- and 15-objective DTLZ2 problems, respectively. This phenomenon may be due to the adaptability of the fitness adjustment strategy in GrEA that tunes the GR of individuals adaptively according to the number of objectives.

IGD is a combined performance indicator of proximity and diversity, but fails to reflect them separately. In the following, we further investigate the grid division parameter via separately investigating its effect on proximity and diversity. In our study, two widely-used quality indicators, convergence measure (CM) and diversity measure (DM) [53], are selected. CM assesses the proximity of a solution set by calculating the average normalised Euclidean distance from the set to the Pareto front. A low value is preferable. DM measures the diversity of a solution set by comparing it with a reference set representing the Pareto front. It takes the value between zero and one, and a larger value means a better coverage of the Pareto front. Detailed descriptions of these two metrics can be found in [53]. In addition, in some trials, we observed that the

Table 4.7: Performance of GrEA with different number of divisions on the six-objective DTLZ2

Division	5	7	8	9	10	12	15	20	30	50
GD	3.944E–3 (1.0E–3)	3.643E–3 (7.5E–4)	3.475E–3 (8.0E–4)	2.724E–3 (7.0E–4)	2.112E–3 (6.6E–4)	1.823E–3 (6.1E–4)	1.729E–3 (6.1E–4)	1.516E–3 (4.5E–4)	1.464E–3 (4.6E–4)	1.110E–3 (4.4E–4)
DM	7.145E–1 (3.7E–2)	8.812E–1 (2.8E–2)	8.834E–1 (3.2E–2)	8.363E–1 (2.5E–2)	7.713E–1 (2.8E–2)	6.886E–1 (3.6E–2)	6.436E–1 (4.2E–2)	4.654E–1 (4.4E–2)	3.542E–1 (7.6E–1)	1.852E–1 (1.4E–1)
IGD	3.509E–1 (8.1E–3)	2.993E–1 (4.8E–3)	2.985E–1 (5.2E–3)	3.172E–1 (4.4E–3)	3.357E–1 (5.7E–3)	3.647E–1 (3.4E–3)	4.115E–1 (3.5E–3)	4.576E–1 (5.9E–3)	5.820E–1 (2.0E–2)	7.155E–1 (1.3E–1)



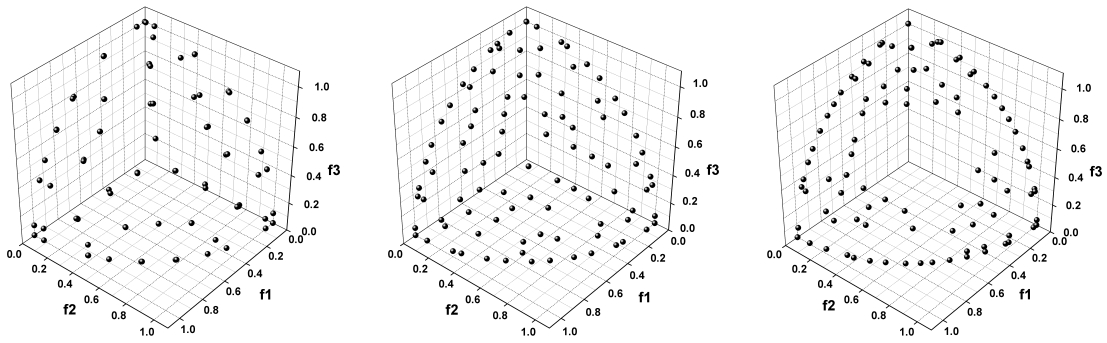
(a) $div = 5$ (b) $div = 8$ (c) $div = 20$

Figure 4.12: The final solution set of GrEA with different divisions on the six-objective DTLZ2, shown by parallel coordinates.

problems with different number of objectives have similar pattern with the variation of divisions. Here, only results on the 6-objective DTLZ2 are demonstrated for brevity.

Table 4.7 shows the results of CM and DM for different division settings on the 6-objective DTLZ2. As can be seen, the number of divisions has a different influence on proximity and diversity. For CM, the algorithm performs better with the increase of the number of divisions, although the degree of variation is not remarkable – it performs well even if the number of divisions decreases to five. The main influence is the distribution results of the algorithm. GrEA has a good distribution of solutions during a segment of the division settings, but when the number of divisions is smaller than 7 or greater than 10, poor performance will be obtained.

Figure 4.12 plots the final solution sets of GrEA with divisions 5, 8, and 20 by parallel coordinates on the 6-objective DTLZ2. In addition, for clearer understanding of these distributions on parallel coordinates, Figure 4.13 shows the final solution sets with different divisions by Cartesian coordinates on the 3-objective DTLZ2, which have



(a) $div = 9$ (b) $div = 16$ (c) $div = 20$

Figure 4.13: The final solution set of GrEA with different divisions on the three-objective DTLZ2, shown by Cartesian coordinates.

similar distributions to the sets in Figure 4.12.

As can be seen from the two figures, the main difference among the three sets is the distribution of solutions. The final solutions with a moderate number of divisions (16 and 8 for 3- and 6-objective DTLZ2 respectively) are well distributed over the Pareto front. However, the solutions with a too small number of divisions (9 and 5 for them) concentrate (or even coincide) in some scattered regions of the Pareto front, and the solutions with a too large number of divisions (20 for them) are located around the boundary of the optimal front.

This occurrence is due to the fitness assignment and adjustment strategies in GrEA. A large hyperbox size (i.e., a small division) would make many solutions located in a single hyperbox and hence assigned the same GR and GCD values (i.e. the former two criteria of fitness). In this case, the third criterion GCPD is activated to distinguish between solutions and further guide them evolving and gathering around the utopia point of a unit hyperbox. On the other hand, a small hyperbox size would lead to the increase of the difference of solutions in terms of the criterion GR. This means that the degree of the punishment for preventing crowding in the environmental selection process is decreased relatively, thus resulting in the failure of solutions covering the Pareto front.

Overall, although the performance varies with the number of grid divisions, GrEA can achieve a good balance between proximity and diversity under a proper setting.

Our experiments suggest that a division value around 9 may be reliable on an unknown optimisation problem. Furthermore, a slightly larger *div* is recommended if the problem in hand is found to be hard to converge, and a slightly lower value may be more suitable if the coverage of the solutions to the Pareto front is more emphasised.

4.3.4 Computational Complexity

The computational cost of GrEA is mainly determined by four steps: grid setting, fitness assignment, mating selection and environmental selection (we neglect the variation operation since it is not specific for the algorithm). The time complexity of the grid setting and mating selection procedures is low. The former needs to identify the maximum and minimum values in each objective for a population of size N, which requires $O(mN)$ comparisons, where m denotes the number of objectives. For the latter, selecting one individual requires $O(m)$ comparisons, so that the time complexity for filling the whole mating pool is $O(mN)$ as well.

The fitness of an individual in GrEA is formed by three criteria, GR, GCD and GCPD. The time complexity of calculating GR and GCPD for all individuals in the population are both $O(mN)$ according to Eqs. (4.7) and (4.9). Since the calculation of GCD for each individual needs to traverse the population for identifying its neighbours, the time complexity for all individuals is $O(mN^2)$, which will govern the computational cost of the fitness assignment procedure.

The computational cost of environmental selection can be divided into two parts (see Algorithm 4.4). The first one corresponds to the operations before the main loop (lines 1–8), including functions *Pareto_nondominated_sort*, *Grid_setting*, and *Initialisation*. The second one is related to the operations in the main loop (lines 9–15), mainly including functions *Findout_best*, *GCD_calculation*, and *GR_adjustment*. The time complexity of the first part is governed by the *Pareto_nondominated_sort* function, which requires $O(mN^2)$ comparisons.

As to the operations in the second part, functions *Findout_best* and *GCD_calculation* require $O(N)$ and $O(mN)$ computations, respectively, according to Algorithms 4.7 and 4.6. For function *GR_adjustment* (Algorithm 4.3), preconditioning (i.e., finding indi-

viduals who are neighbours of, grid-dominated by, not grid-dominated by, or identical with the picked individual) requires $O(mN)$ comparisons. The computational cost of the main procedure is governed by the punishment operation for preventing crowding (lines 10–19). Since the implementation of the punishment of an individual not only relates to itself but involves the individuals that it grid-dominates, the time complexity of this operation is $O(LN)$. Here, L denotes, for a set of Pareto nondominated individuals, the average number of the individuals in the set that are grid-dominated by one member of the set, where the grid environment is formed by the set. Obviously, $O(1) \leq O(L) \leq O(N)$. The former occurs when all individuals in the set are grid-nondominated to each other, and the latter occurs when a totally ordered relation regarding grid dominance holds for all individuals in the set. In fact, it may be a hard task to precisely determine L – it would be influenced not only by the population size and the number of objectives, but by the number of divisions (a larger division number would make more individuals grid-nondominated, and when $div \to \infty$, $L \to 0$). We leave it for a future study. Overall, from the above analysis, the average time complexity of the environmental selection procedure is bounded by $O(mN^2)$ or $O(LN^2)$, whichever is greater.

To sum up, the total complexity of the proposed algorithm is generally governed by the environmental selection procedure, thus bounded by $O(mN^2)$ or $O(LN^2)$ on average, whichever is greater. Since $O(1) \leq O(L) \leq O(N)$ and in most cases, $m < N$, the best and worst time complexity of the algorithm would be $O(mN^2)$ and $O(N^3)$, respectively.

4.4 Summary

In this chapter, we exploited the potential of the grid to handle many-objective optimisation problems. The proposed algorithm, GrEA, can mainly be characterised as:

- executing individual-centred calculation instead of grid-centred calculation throughout the algorithm;
- increasing the selection pressure towards the optimal front by introducing three

grid-based relations, GR, GCPD and grid dominance;

- estimating the density of individuals by using adaptive neighbourhood whose range varies with the number of objectives;
- adjusting the fitness of individuals in the environmental selection process by considering both neighbourhood and grid dominance relations.

Systematic experiments were carried out to make an extensive comparison of GrEA with five well-established EMO algorithms. Several groups of widely used test problems are chosen for challenging different capabilities of the algorithms. The results reveal that GrEA is very competitive against the peer algorithms in terms of finding a well-converged and well-distributed solution set in many-objective optimisation.

Furthermore, the effect of a key parameter in GrEA, the number of grid divisions, has been experimentally investigated. The results show that although the performance of GrEA varies with the number of grid divisions, GrEA can achieve a good tradeoff among proximity and diversity under a proper setting. The division parameter nine is recommended for an unknown MaOPs, and a slightly higher (or lower) value is suggested when the problem has been found to be hard to achieve a good proximity (or coverage) of the Pareto front. In addition, the study on the time complexity indicates that GrEA is suited to many-objective optimisation, given its 1) linear increase with objective dimensionality and 2) high independence of the number of hyperboxes in the grid.

Chapter 5

Bi-Goal Evolution

In this chapter, we present a new framework for evolutionary algorithms, called bi-goal evolution (BiGE), to deal with MaOPs. In multi-objective optimisation, it is generally observed that 1) the conflict between proximity and diversity requirements is aggravated with the increase of the number of objectives [216, 81, 1] and 2) the Pareto dominance loses its effectiveness for a high-dimensional space but works well on a low-dimensional space [45, 256, 128]. Inspired by these two observations, BiGE converts a given multi-objective optimisation problem into a bi-goal (objective) optimisation problem regarding proximity and diversity, and then handles it using the Pareto dominance relation in this bi-goal domain.

This chapter is organised as follows. In Section 5.1, the motivation of BiGE is given. Section 5.2 is devoted to the presentation of the BiGE framework and its implementation. Empirical results of BiGE in comparison with five best-in-class algorithms are shown in Section 5.3. Further investigation of the bi-goal evolution is carried out in Section 5.4. Finally, Section 5.5 summaries this chapter.

5.1 Motivation

An EMO algorithm pursues two basic but often conflicting goals, proximity and diversity. Such conflict has a detrimental impact on the algorithm's optimisation process and can be aggravated in many-objective optimisation. Figure 5.1 gives the comparison

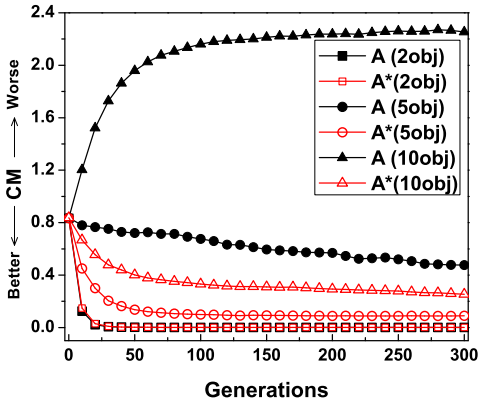
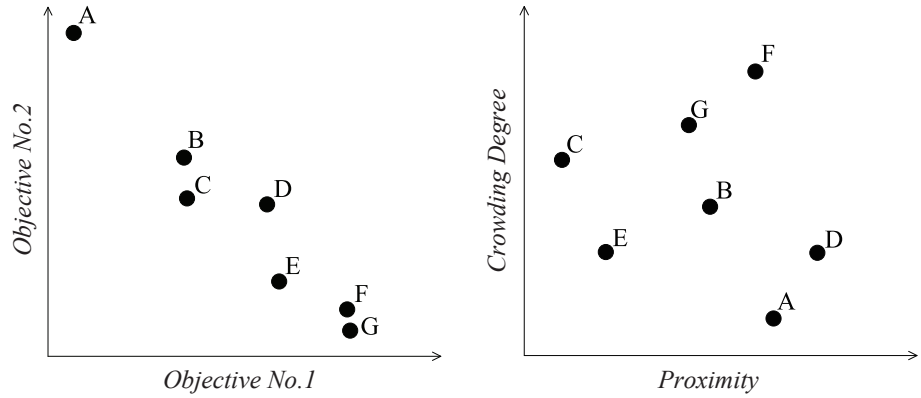


Figure 5.1: Evolutionary trajectories of the average convergence metric (CM) for 30 runs of the original NSGA-II (denoted as **A**) and the modified NSGA-II without the diversity maintenance mechanism (denoted as $\mathbf{A}^*$) on DTLZ2.

trajectories of the proximity results between the original NSGA-II [55] (involving both proximity and diversity maintenance mechanisms) and its modified version in which the diversity maintenance mechanism is removed, on the 2-, 5- and 10-objective DTLZ2 [57]. These results are evaluated by a convergence metric (CM) [53], which calculates the average normalised Euclidean distance from the solution set to the Pareto front.

As can be seen in Figure 5.1, the interval between the CM trajectories of the two algorithms becomes more visible with the increase of the number of objectives. For the 2-objective problem, both algorithms perform well, with their CM trajectories being virtually overlapping. For the 5-objective problem, the NSGA-II without the diversity maintenance mechanism achieves better CM results than the original NSGA-II during the evolutionary process, which means that diversity maintenance has an unfavourable impact on the proximity of the algorithm. For the 10-objective problem, the diversity maintenance mechanism in NSGA-II even makes the evolving population gradually move away from the Pareto front; the great interval between the two trajectories in Figure 5.1 indicates a serious conflict between proximity and diversity obtained.

On the other hand, Pareto dominance, which is popular and effective to distinguish between individuals in 2- or 3-objective MOPs, fails in many-objective optimisation. In fact, the proportion of any two individuals being comparable in an m-dimensional



(a) The actual objective space (b) The bi-goal space

Figure 5.2: An illustration of the conversion from the actual objective space to the bi-goal space of proximity and crowding degree on a bi-objective minimisation problem.

objective space is $\eta = 1/2^{m-1}$. For a 2- or 3-dimensional space, η is equal to 0.5 or 0.25, respectively, but when m reaches 6, η is already as low as 0.03125. Such exponential decrease of the portion leads to the dramatic decline of Pareto dominance's effectiveness with the number of objectives.

Given the above, it could then be viable to use Pareto dominance to only optimise the two goals (objectives) of proximity and diversity rather than to cope with all the objectives of an MOP. This way, sufficient selection pressure can be provided even in a very high-dimensional space. Bearing this in mind, we propose a bi-goal evolution framework, BiGE, to tackle many-objective optimisation problems.

5.2 The Proposed Approach

BiGE treats an MOP with many objectives as a bi-goal optimisation problem regarding minimising the proximity of individuals towards the optimal direction and minimising the crowding degree of individuals in the population. Figure 5.2 gives a bi-objective scenario to illustrate the conversion from the actual objective space to the bi-goal space.

As can be seen from Figure 5.2, by conversion, some of the nondominated individuals **A**–**G** in the objective space become comparable. In the bi-goal space, only three individuals (**C**, **A**, and **E**) are Pareto nondominated (i.e., the best individuals in the

Algorithm 5.1 Bi-Goal Evolution (BiGE)

Require: P (population), N (population size)
1: $P \leftarrow initialize(P)$
2: **while** termination criterion not fulfilled **do**
3: $proximityEstimation(P)$
4: $crowdingDegreeEstimation(P)$
5: $P' \leftarrow matingSelection(P)$
6: $P'' \leftarrow variation(P')$
7: $P \leftarrow environmentalSelection(P \bigcup P'')$
8: **end while**
9: **return** P

population), given that **C** and **A** perform best in terms of proximity and crowding degree, respectively, and the performance of **E** can be regarded as the tradeoff between that of **C** and **A**. In contrast, individual **F**, which performs poorly in both proximity and crowding degree, is dominated by most of the individuals in the population.

Below, we introduce the main procedure of BiGE and its specific implementations.

5.2.1 Basic Procedure

The aim of BiGE is to deal with the ineffectiveness of the Pareto dominance relation in the high-dimensional objective space. BiGE only considers the individuals when they are incomparable on the basis of Pareto dominance in the selection process. Algorithm 5.1 gives the basic procedure of BiGE. Firstly, N individuals are randomly generated to form an initial population P. Then, the proximity and crowding degree of individuals in the current population are estimated. Next, mating selection is performed to select promising solutions in the bi-goal space for variation. Finally, the environmental selection procedure is implemented to keep a record of the N best solutions with respect to the two goals for survival.

5.2.2 Proximity Estimation

Conversion from an MOP with a number of objectives to a bi-goal problem involves an integration of the objectives. In order to make the integration feasible (i.e., to be able to deal with an MOP with non-commensurable objective functions), in BiGE each objective of individuals is normalised (with respect to its minimum and maximum val-

ues in the current population) before estimating their proximity and crowding degree. For convenience, in the description of the proposed algorithm, the objective value of individuals refers to their normalised objective value in the range $[0, 1]$.

BiGE estimates the proximity (denoted as f_{pr}) of an individual p in the population by summing its value in each objective:

$$f_{pr}(p) = \sum_{k=1}^{m} f^k(p) \tag{5.1}$$

where $f^k(p)$ denotes the objective value of individual p in the kth objective, and m is the number of objectives. This estimation function is determined by two factors: the number of objectives and the performance in each objective. An individual with good performance in the majority of objectives is likely to obtain a lower (better) f_{pr} value.

It is worth pointing out that the proximity information of an individual with m objectives (i.e., an m-dimensional vector) cannot be completely reflected and represented by the scalar value f_{pr}. The accuracy of the estimation can be influenced by the shape of an MOP's Pareto front. For example, individuals around the knee of the Pareto front often have better estimation result than those far away from the knee even if they are non-dominated to each other. To solve this issue, we introduce the goal of minimising the crowding degree of individuals in the population. We consider the Pareto dominance relation of the two goals, preferring individuals with a good tradeoff between them.

5.2.3 Crowding Degree Estimation

Niching techniques are a kind of popular density estimation methods in the EA field. Bearing the idea of sharing resource in mind, niching techniques can effectively measure the crowding degree of an individual in the population. Here, we consider the following sharing function between two individuals p and q:

$$sh(p, q) = \begin{cases} (1 - \frac{d(p,q)}{r})^2, & \text{if } d(p, q) < r \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where $d(p, q)$ denotes the Euclidean distance between individuals p and q in the objective space, and r is the radius of a niche, determined by the population size N and the number of objectives m of a given MOP:

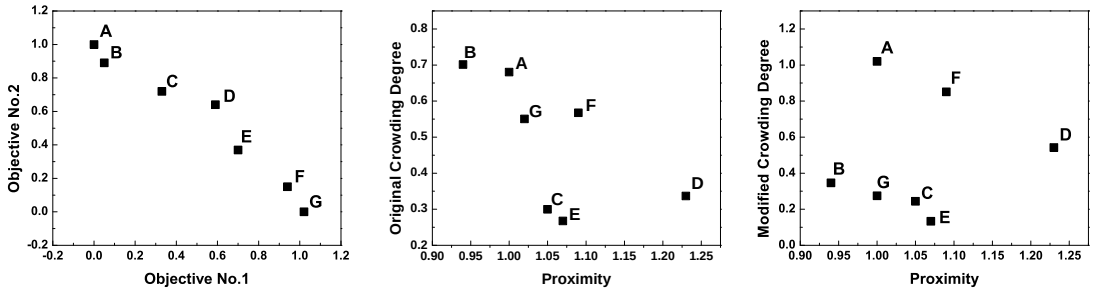
$$r = \frac{1}{\sqrt[m]{N}} \tag{5.3}$$

Note that the considered individuals are already normalised according to the range of the current population. Thus, the niche radius here is actually adaptive, varying with the evolutionary population. Using the sharing function in Eq. (5.2), the crowding degree (denoted as f_{cd}) of an individual p in a population P is defined as follows:

$$f_{cd}(p) = (\sum_{q \in P, q \neq p} sh(p, q))^{1/2} \tag{5.4}$$

Up to now, the performance of an individual in the population has been reflected by f_{pr} and f_{cd}. However, a problem may arise when applying these two estimation functions in the conversion from the actual objective space into the bi-goal space. Since the performance estimation of an individual depends on its position in comparison with other individuals in the population, the individuals located closely in the objective space may have similar behaviours regarding both proximity and crowding degree, thus also being situated closely in the bi-goal space. For example, similar nondominated individuals **A** and **B** in Figure 5.3(a), after conversion, are still located closely and nondominated to each other (shown in Figure 5.3(b)). In this case, it is likely that such individuals are preserved or eliminated simultaneously, which may result in congestion in some regions yet vacancy in some other regions.

To overcome this problem, we make a modification to the sharing function in Eq. (5.2) in order to distinguish between similar individuals. Two individuals will be assigned different sharing function values according to their performance comparison in terms of proximity. Specifically, we introduce a weight parameter (called the



(a) (b) (c)

Figure 5.3: An illustration of the case that similar individuals in the objective space may be located closely and nondominated to each other in the bi-goal space, and its remedy. (a) The actual objective space; (b) The bi-goal space with respect to the proximity and the original crowding degree; (c) The bi-goal space with respect to the proximity and the modified crowding degree. The numerical values of the individuals in these three spaces are given in Table 5.1.

Table 5.1: Individual values in the three spaces for the example of Figure 5.3.

	(Objective No. 1, Objective No. 2)	(Proximity, Original Crowding Degree)	(Proximity, Modified Crowding Degree)
A	(0.00, 1.00)	(1.00, 0.68031)	(1.00, 1.02047)
B	(0.05, 0.89)	(0.94, 0.70114)	(0.94, 0.34663)
C	(0.33, 0.72)	(1.05, 0.29965)	(1.05, 0.24422)
D	(0.59, 0.64)	(1.23, 0.33658)	(1.23, 0.54256)
E	(0.70, 0.37)	(1.07, 0.26737)	(1.07, 0.13369)
F	(0.94, 0.15)	(1.09, 0.56741)	(1.09, 0.85112)
G	(1.02, 0.00)	(1.02, 0.55022)	(1.02, 0.27511)

sharing discriminator) in the sharing function:

$$sh(p,q) = \begin{cases} (0.5(1 - \frac{d(p,q)}{r}))^2, & \text{if } d(p,q) < r, f_{pr}(p) < f_{pr}(q) \\ (1.5(1 - \frac{d(p,q)}{r}))^2, & \text{if } d(p,q) < r, f_{pr}(p) > f_{pr}(q) \\ rand(), & \text{if } d(p,q) < r, f_{pr}(p) = f_{pr}(q) \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

where the function $rand()$ means to assign *either* $sh(p,q) = (0.5(1 - \frac{d(p,q)}{r}))^2$ and $sh(q,p) = (1.5(1 - \frac{d(p,q)}{r}))^2$ *or* $sh(p,q) = (1.5(1 - \frac{d(p,q)}{r}))^2$ and $sh(q,p) = (0.5(1 - \frac{d(p,q)}{r}))^2$ randomly.

The sharing function now contributes differently to the crowding degree of individuals in the niche. An individual with better proximity than its neighbours will obtain a lower crowding degree. For two individuals which are the sole neighbour to each

Algorithm 5.2 Tournament selection

Require: individuals p, q
1: **if** $p \prec q$ in the bi-goal domain **then**
2: **return** p
3: **else if** $q \prec p$ in the bi-goal domain **then**
4: **return** q
5: **else if** $random(0, 1) < 0.5$ **then**
6: **return** p
7: **else**
8: **return** q
9: **end if**

other in a population, they had the same crowding degree before, but now the better individual (in terms of proximity) will only have half of the original crowding degree and the worse one will have one and a half of the original crowding degree.

In general, this modification enables adjacent individuals to be located distantly. More importantly, it could lead to similar individuals comparable on the basis of the Pareto dominance criterion of the proximity and diversity goals, which is well suited to BiGE. Figure 5.3(c) gives an illustration to explain the effect of this modification. As shown, individual **A** will become dominated by **B** when evaluated by the modified crowding degree. Table 5.1 shows the values of individuals in the three spaces for the example of Figure 5.3.

5.2.4 Mating Selection

Mating selection, which aims to make a good preparation for exchanging the information of individuals, picks out promising solutions from the current population to form a mating pool. BiGE uses a type of binary tournament selection strategy based on Pareto dominance in the bi-goal domain, as given in Algorithm 5.2. For two candidates, if they are Pareto-comparable in the two goal functions (e.g., $f_{pr}(p) < f_{pr}(q) \wedge f_{cd}(p) < f_{cd}(q)$), then the better one will be selected; otherwise, the tie will be split randomly.

5.2.5 Environmental Selection

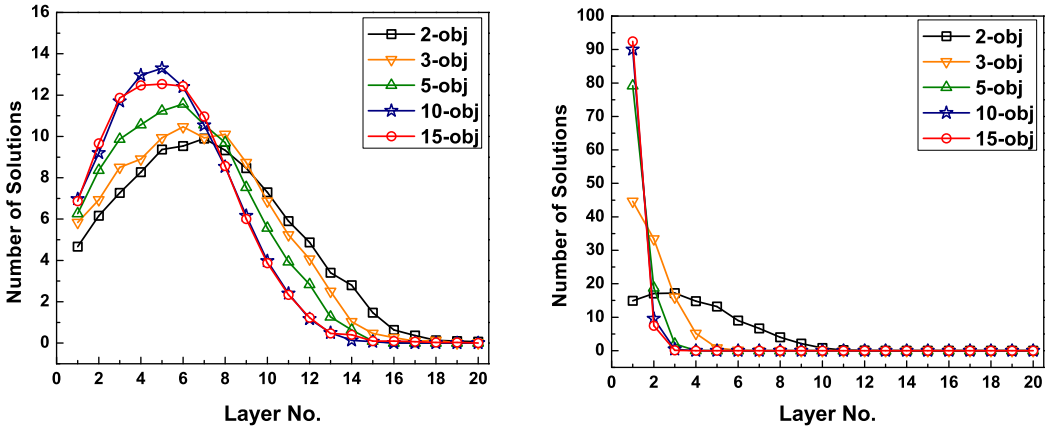
Environmental selection, which aims to obtain a well-approximated and well-distributed new population, chooses the "best" solutions from the previous population and newly

Algorithm 5.3 *environmentalSelection*(Q)

Require: N (population size)
1: Generate an empty population P
2: *proximityEstimation*(Q)
/* Compute proximity of each individual in Q by Eq. (5.1) */
3: *crowdingDegreeEstimation*(Q)
/* Compute crowding degree of each individual in Q by Eqs. (5.4) and (5.5) */
4: $\{L_1, L_2, ..., L_i, ...\} \leftarrow$ *nondominatedSorting*(Q)
/* Partition Q into different layers $(L_1, L_2, ..., L_i, ...)$ by using Pareto nondominated sorting regarding proximity and crowding degree, and find the critical layer L_i (i.e., $0 \leq N - |L_1 \cup L_2 \cup ... \cup L_{i-1}| < |L_i|$) */
5: $P \leftarrow L_1 \cup L_2 \cup ... \cup L_{i-1}$
6: **if** $|P| < N$ **then**
7: *randomSelection*$(P, L_i, N - |P|)$
/* Select $N - |P|$ individuals from L_i into P at random */
8: **end if**
9: **return** P

created individuals. BiGE implements the environmental selection according to individuals' Pareto dominance relation in the bi-goal domain. Here, we adopt a popular Pareto-based rank strategy in the area: nondominated sorting [86]. Nondominated sorting is an effective method to rank individuals in a low-dimensional space. First, the nondominated individuals in the population are identified as the first layer. Then, the remaining individuals are regarded as the current population, from which nondominated individuals are selected to form the second layer. This process is continued until the entire population is classified into different layers.

Algorithm 5.3 gives the environmental selection procedure of BiGE. First, individuals' performance regarding the proximity and crowding degree is estimated (Steps 2 and 3). Then, the candidate set Q is divided into different layers by the nondominated sorting procedure with respect to the two goals, and the first $(i-1)$ layers are moved into the population P, where $|L_1 \cup L_2 \cup ... \cup L_{i-1}| \leq N$ and $|L_1 \cup L_2 \cup ... \cup L_{i-1} \cup L_i| > N$ (Steps 4 and 5). Finally, the slots in P are filled randomly by individuals in L_i (Steps 6–8). Note that BiGE employs a randomly-selected mode on the layer L_i, rather than a density-based selection mode. This is because the density of individuals in this bi-goal space does not reflect their own performance. An individual with high density in the bi-goal space does not mean that it is worse than individuals with low density but rather that there are some other individuals having similar proximity and crowding degree



(a) Nondominated sorting of the two goals (b) Nondominated sorting of the actual objectives

Figure 5.4: The average number of solutions in all the nondominated layers under (a) the bi-goal Pareto nondominated sorting and (b) the original Pareto nondominated sorting, where the population size is 100, the number of runs is 30, and the test instance is DTLZ2.

with it in the population (cf. individual **C** in the example of Figure 5.3). Therefore, we randomly select individuals which are located in the same layer.

In order to investigate the effectiveness of the bi-goal nondominated sorting in providing the selection pressure, Figure 5.4 demonstrates the average number of solutions in all the nondominated layers on the 2-, 3-, 5-, 10- and 15-objective DTLZ2, where, for contrast, the average number of solutions in all the nondominated layers obtained by nondominated sorting of the actual objectives is shown as well. As can be seen from Figure 5.4(b), the number of individuals placed in the first layer (L_1) increases rapidly with the number of objectives, approximating 80% of the population size when the number of objectives reaches 5. In contrast, the individuals in Figure 5.4(a) are located in many different layers and distributed in a similar pattern. For example, L_1 is always small and has around 6 individuals. In all the instances, the number of individuals in L_i gradually increases until the total number of individuals in L_1 to L_i reaches around half of the population size. This similar pattern means that the bi-goal nondominated sorting can effectively distinguish between individuals, which is largely independent of the number of objectives.

Table 5.2: Properties of test problems in comparative studies

Problem	Number of Objectives (m)	Number of Variables (n)	Properties
WFG1	5, 10, 15	$2 \times (m-1) + 20$	Mixed, Flat Biased
WFG2	5, 10, 15	$2 \times (m-1) + 20$	Convex, Disconnected, Nonseparable
WFG3	5, 10, 15	$2 \times (m-1) + 20$	Linear, Degenerate, Nonseparable
WFG4	5, 10, 15	$2 \times (m-1) + 20$	Concave, Multimodal
WFG5	5, 10, 15	$2 \times (m-1) + 20$	Concave, Deceptive
WFG6	5, 10, 15	$2 \times (m-1) + 20$	Concave, Nonseparable
WFG7	5, 10, 15	$2 \times (m-1) + 20$	Concave, Parameter Dependant Biased
WFG8	5, 10, 15	$2 \times (m-1) + 20$	Concave, Nonseparable, Parameter Dependant Biased
WFG9	5, 10, 15	$2 \times (m-1) + 20$	Concave, Nonseparable, Deceptive, Parameter Dependant Biased
Knapsack	5, 10, 15	500	Convex, Constraint
TSP	5, 10, 15	30	Convex, Zero Correlation
Water	5	3	Convex, Degenerate, Constraint

5.3 Experimental Results

BiGE focuses on the comparison among the individuals which are nondominated to each other in the objective space. For the individuals that can be differentiated by Pareto dominance, any existing comparison strategy in the EMO area, such as the nondominated sorting [86], nondominated ranking [74], and strength [294], can be used. Here, the nondominated sorting strategy is chosen to cooperate with BiGE due to its simplicity and popularity [55].

We compare BiGE with five state-of-the-art algorithms, MOEA/D (with the Tchebycheff scalarizing function[1]), NSGA-III [52], HypE [10], FD-NSGA-II [96], and AGE-II [255]. All these algorithms have been well verified in dealing with MaOPs [180, 52, 264, 96, 255, 260].

5.3.1 Experimental Settings

In the experimental studies, three well-known continuous and combinatorial benchmark suites, the walking fish group (WFG) toolkit [105], the multi-objective 0–1 knapsack problem [294], and the multi-objective travelling salesman problem (TSP) [45], are included, with the objective number $m = 5$, 10, and 15. Also, a real-world constraint problem, the water problem [202], is considered. Their characteristics are summarized in Table 5.2.

[1]In order to obtain more uniform solutions, in the Tchebycheff scalarizing function, "multiplying the weight vector w_i" in the original MOEA/D [278] is replaced by "dividing w_i", as suggested and practised in recent studies [52, 172].

To compare the performance of the algorithms, the hypervolume (HV) metric [294] is used. Here, we present a normalised HV value of each algorithm with respect to the proportion of the optimal HV result achieved. This normalisation makes all of the obtained results reside in the range $[0, 1]$, with 1 representing the optimal value. For some of the test problems (i.e., WFG4–WFG9), the optimal HV value can be obtained by calculation; for the others, the optimal value is, as suggested in [89], approximately estimated by the HV result of the nondominated set with respect to the mixed population consisting of all the obtained solutions on a given problem.

In the calculation of HV, two crucial issues are the scaling of the search space [77] and the choice of the reference point [9, 78]. Since the objectives in the WFG and water problems take different ranges of values, we standardise the objective value of the obtained solutions according to the range of the problem's Pareto front. Following the recommendation in [119], the reference point is set to 1.1 times the upper bound of the Pareto front (i.e., $r = 1.1^m$) to emphasise the balance between proximity and diversity of the obtained solution set. For the two combinatorial optimisation problems, since the range of their Pareto front is unknown, we set the reference point slightly worse than the boundary values of the nondominated set with respect to the mixed population consisting of all the obtained solutions; that is, the points with 13000 and 22 for each objective (i.e., $r = 13000^m$ and $r = 22^m$) are fixed for the knapsack and TSP problems, respectively.

In addition, since the exact calculation of the HV indicator is generally infeasible for a solution set with 10 or more objectives, we approximately estimate the HV result of a solution set by the Monte Carlo sampling method used in [10]. Here, 10^7 sampling points are used to ensure accuracy [10].

All the results presented in this chapter were obtained by executing 30 independent runs of each algorithm on each problem. Following the practice in [81, 256], the population size was set to 100 and the termination criterion of a run was 30,000 evaluations (i.e., 300 generations) for the WFG, TSP and water problems. For the knapsack problem, more evaluations are required for one generation of an algorithm due to the repair method that deals with infeasible solutions, where we set 100,000 evaluations

as the termination criterion. Note that the size of the population in MOEA/D and NSGA-III is often the same as the number of weight vectors and it is impossible for the algorithms to generate uniformly distributed weight vectors at an arbitrary number. Here, we uniformly generate a set of around 5,000 weight vectors and then select 100 well-distributed weight vectors from the set by using the method in [279].

Parameters need to be set in some peer algorithms. According to the study in MOEA/D [278], the neighbourhood size was specified as 10% of the population size. For HypE, the number of sampling points in HypE was set to 10,000. Following the practice in [264], the reference point for calculating the hypervolume contribution in HypE was set to $2i + 1$ for all WFG problems, where i is the number of objectives; for other problems, the reference point was set to be the same as in the HV indicator. In FD-NSGA-II, parameter σ, which determines the spread of the Gaussian function, was set to 0.5, as suggested in [96]. In AGE-II, parameter ϵ_{grid}, which determines the size of the archive, was set to 0.1 since it can provide a good tradeoff between performance and runtime in many-objective problems [255].

A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n denotes the number of decision variables) were used. For continuous problems, operators for crossover and mutation are SBX crossover and polynomial mutation with both distribution indexes set to 20 [10, 278]. As to the combinatorial problems, following the studies in [45, 118], the uniform crossover and bit-flip mutation were used for the knapsack instance, and the order crossover and inversion mutation were used for the TSP instance.

5.3.2 Experimental Comparison

In this section, we verify the performance of BiGE according to the experimental design described previously. HV results in the tables are the mean and standard deviation (SD) over 30 independent runs, and the best and second best mean values among the algorithms for each problem instance are shown with dark and light grey background, respectively. Moreover, in order to have statistically sound conclusions, we adopt the Wilcoxon's rank sum test [290] at a 0.05 significance level to examine the significance of

Table 5.3: Normalised HV results (mean and SD) of the six algorithms on the WFG problem. The best and the second mean among the algorithms for each problem instance is shown with dark and light grey background, respectively

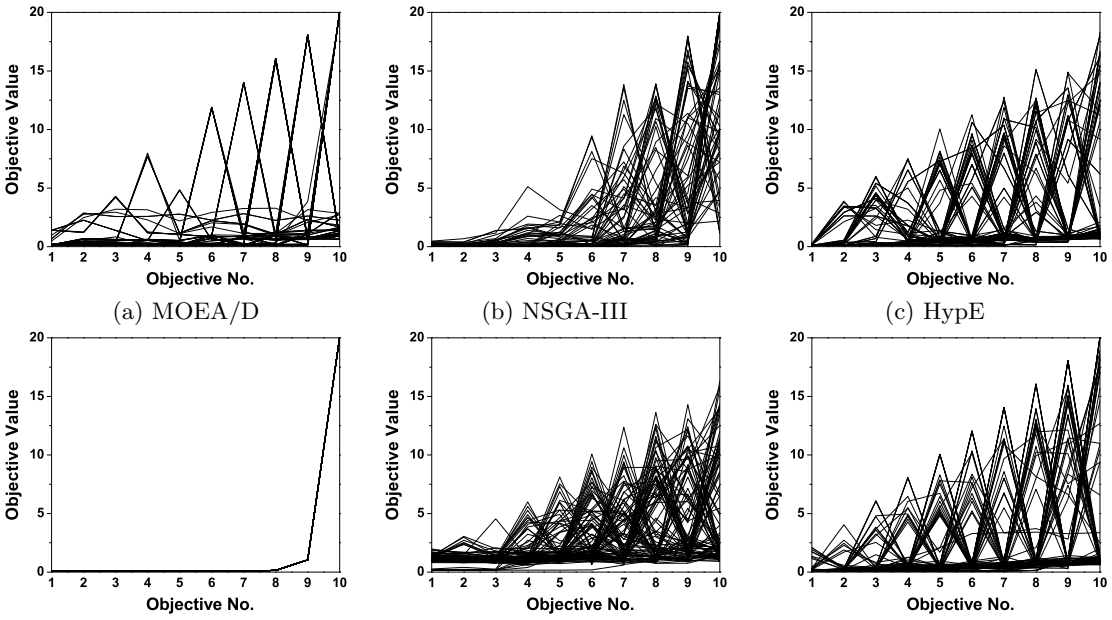
Problem	Obj.	MOEA/D	NSGA-III	HypE	FD-NSGA-II	AGE-II	BiGE
WFG1	5	6.892E-1 (3.6E-2)†	6.773E-1 (6.3E-2)†	7.827E-1 (1.6E-2)†	2.194E-1 (7.2E-2)†	5.219E-1 (2.4E-2)†	6.151E-1 (3.9E-2)
	10	6.421E-1 (3.2E-2)†	7.710E-1 (5.4E-2)	8.324E-1 (2.3E-2)†	4.294E-1 (6.9E-2)†	6.092E-1 (2.8E-2)†	7.786E-1 (4.7E-2)
	15	5.913E-1 (2.2E-2)†	7.208E-1 (3.1E-2)†	8.138E-1 (3.1E-2)†	4.775E-1 (7.8E-2)†	5.735E-1 (2.2E-2)†	7.612E-1 (3.4E-2)
WFG2	5	7.999E-1 (9.0E-2)†	8.881E-1 (1.5E-1)	8.819E-1 (8.6E-2)	2.977E-1 (7.4E-2)†	9.180E-1 (7.2E-2)	9.014E-1 (8.2E-2)
	10	7.576E-1 (8.4E-2)†	8.647E-1 (7.5E-2)†	8.864E-1 (8.0E-2)†	2.698E-1 (1.4E-1)†	9.168E-1 (7.1E-2)	9.475E-1 (5.3E-2)
	15	6.454E-1 (7.7E-2)†	8.834E-1 (8.1E-2)†	9.247E-1 (8.3E-2)	2.841E-1 (1.2E-1)†	8.535E-1 (7.1E-2)†	9.402E-1 (6.1E-2)
WFG3	5	6.450E-1 (2.0E-2)†	9.139E-1 (1.7E-2)	9.137E-1 (7.4E-3)	1.330E-1 (1.9E-3)†	8.139E-1 (2.5E-2)†	9.092E-1 (1.1E-2)
	10	4.553E-1 (3.2E-2)†	7.949E-1 (6.6E-2)†	9.149E-1 (1.2E-2)†	1.242E-1 (1.8E-3)†	6.673E-1 (3.4E-2)†	8.705E-1 (1.8E-2)
	15	2.431E-1 (2.3E-2)†	8.136E-1 (7.7E-2)†	8.964E-1 (1.8E-2)†	1.213E-1 (2.4E-3)†	4.713E-1 (2.6E-2)†	8.545E-1 (2.6E-2)
WFG4	5	5.899E-1 (2.9E-2)†	7.269E-1 (6.5E-2)†	8.171E-1 (7.6E-3)†	2.770E-1 (8.2E-2)†	6.381E-1 (1.9E-2)†	8.117E-1 (8.4E-3)
	10	4.359E-1 (4.4E-2)†	4.223E-1 (7.6E-2)†	8.049E-1 (2.9E-2)†	2.207E-1 (9.8E-2)†	3.248E-1 (2.3E-2)†	8.313E-1 (1.1E-2)
	15	3.000E-1 (3.8E-2)†	5.332E-1 (4.8E-2)†	7.831E-1 (2.5E-2)†	1.223E-1 (5.2E-2)†	2.336E-1 (2.3E-2)†	8.073E-1 (1.9E-2)
WFG5	5	6.418E-1 (1.8E-2)†	7.607E-1 (5.8E-3)†	7.701E-1 (8.5E-3)	8.776E-2 (2.0E-4)†	6.301E-1 (1.5E-2)†	7.709E-1 (6.2E-3)
	10	4.911E-1 (3.4E-2)†	5.235E-1 (6.5E-2)†	8.142E-1 (1.7E-2)†	7.930E-2 (2.6E-4)†	4.040E-1 (3.5E-2)†	7.990E-1 (1.9E-2)
	15	3.170E-1 (6.4E-2)†	6.250E-1 (5.4E-2)†	7.628E-1 (1.9E-2)†	7.892E-2 (2.8E-4)†	2.735E-1 (6.5E-2)†	7.715E-1 (1.4E-2)
WFG6	5	5.753E-1 (2.6E-2)†	7.718E-1 (3.9E-2)	7.750E-1 (1.2E-2)	1.040E-1 (3.5E-2)†	6.577E-1 (1.8E-2)†	7.728E-1 (8.9E-3)
	10	5.158E-1 (4.3E-2)†	5.389E-1 (5.1E-2)†	8.352E-1 (1.1E-2)†	8.434E-2 (1.3E-3)†	3.897E-1 (3.0E-2)†	8.270E-1 (1.3E-2)
	15	2.845E-1 (4.5E-2)†	6.528E-1 (4.5E-2)†	8.139E-1 (2.3E-2)†	8.367E-2 (1.2E-3)†	3.402E-1 (7.1E-2)†	8.339E-1 (1.4E-2)
WFG7	5	6.413E-1 (4.3E-2)†	7.878E-1 (3.7E-2)†	8.262E-1 (8.7E-3)†	1.011E-1 (3.7E-5)†	6.770E-1 (2.0E-2)†	8.356E-1 (5.5E-3)
	10	5.852E-1 (4.3E-2)†	5.637E-1 (6.7E-2)†	8.808E-1 (1.2E-2)	1.140E-1 (5.3E-2)†	3.869E-1 (5.6E-2)†	8.827E-1 (1.2E-2)
	15	2.057E-1 (5.8E-2)†	6.919E-1 (5.2E-2)†	8.305E-1 (2.4E-2)†	9.381E-2 (1.7E-2)†	1.160E-1 (2.5E-2)†	8.787E-1 (1.3E-2)
WFG8	5	3.536E-1 (2.7E-2)†	5.609E-1 (9.7E-2)†	6.893E-1 (7.7E-3)†	1.529E-1 (8.6E-2)†	5.587E-1 (2.8E-2)†	6.822E-1 (9.1E-3)
	10	3.756E-1 (3.1E-2)†	4.944E-1 (7.0E-2)†	7.628E-1 (1.2E-2)†	1.279E-1 (5.9E-2)†	2.976E-1 (7.1E-2)†	7.722E-1 (6.1E-3)
	15	2.479E-1 (6.3E-2)†	6.112E-1 (8.8E-1)†	7.872E-1 (2.4E-2)†	1.116E-1 (5.2E-2)†	1.215E-1 (4.5E-2)†	8.179E-1 (1.0E-2)
WFG9	5	4.727E-1 (3.5E-2)†	7.253E-1 (1.6E-2)†	6.971E-1 (3.2E-2)	8.059E-2 (4.6E-5)†	5.963E-1 (2.9E-2)†	6.903E-1 (1.4E-2)
	10	3.675E-1 (5.3E-2)†	6.127E-1 (3.6E-2)†	6.727E-1 (2.4E-2)†	7.417E-2 (2.3E-3)†	4.273E-1 (5.1E-2)†	6.824E-1 (1.3E-2)
	15	2.044E-1 (5.3E-2)†	5.570E-1 (4.2E-2)†	6.677E-1 (1.2E-2)†	7.435E-2 (3.3E-3)†	3.408E-1 (5.2E-2)†	6.893E-1 (3.1E-2)

"†" indicates that the result of the peer algorithm is significantly different from that of BiGE at a 0.05 level by the Wilcoxon's rank sum test.

the difference between the results obtained by BiGE and its competitors. The Wilcoxon test is a non-parametric alternative to the two-sample t-test with two advantages: 1) valid for data with a non-normal distribution and 2) much less sensitive to the outliers.

WFG Problems

Table 5.3 gives the comparative results of the six algorithms on the WFG problems with 5, 10, and 15 objectives. As shown, BiGE and HypE perform best, having a clear advantage over the other 4 algorithms on most of the test instances. Specifically, BiGE obtains the best and second best HV results on 14 and 10 out of the 27 instances respectively, and HypE on 10 and 15 respectively. NSGA-III performs best on the 5-objective WFG3 and WFG9, and also generally outperforms the other three algorithms. AGE-II and MOEA/D typically work fairly well on the 5-objective WFG, but struggle on the 10- and 15-objective instances. FD-NSGA-II, which fails to maintain the diversity of individuals in the population, has the worst HV results on the WFG problem suite.



(a) MOEA/D (b) NSGA-III (c) HypE

(d) FD-NSGA-II (e) AGE-II (f) BiGE

Figure 5.5: The final solution set of the six algorithms on the ten-objective WFG9, shown by parallel coordinates.

Concerning the statistical results, it can be observed that the difference between BiGE and the peer algorithms is significant on most of the test instances. Specifically, the proportion of the test instances where BiGE outperforms MOEA/D, NSGA-III, HypE, FD-NSGA-II and AGE-II with statistical significance is 26/27, 21/27, 11/27, 27/27, and 25/27 respectively. Conversely, the proportion of the instances where BiGE performs worse than MOEA/D, NSGA-III, HypE, FD-NSGA-II and AGE-II with statistical significance is 1/27, 2/27, 9/27, 0/27, and 0/27 respectively.

For a visual understanding of the solutions' distribution, Figure 5.5 plots the final solutions of one run with respect to the 10-objective WFG9 by parallel coordinates. This particular run is associated with the result that is the closest to the mean HV value. Although all considered solution sets appear to converge into the optimal front (the upper and lower bounds of objective i in WFG's Pareto front are 0 and $2 \times i$, respectively), the six algorithms perform differently in terms of diversity maintenance. The solutions obtained by FD-NSGA-II converge into one point of the Pareto front, while the solutions of MOEA/D concentrate in the boundaries of the optimal front.

Table 5.4: Normalised HV results (mean and SD) of the six algorithms on the Knapsack problem. The best and the second mean among the algorithms for each problem instance is shown with dark and light grey background, respectively

Obj.	MOEA/D	NSGA-III	HypE	FD-NSGA-II	AGE-II	BiGE
5	5.412E-1 (3.2E-2)†	4.657E-1 (1.6E-2)†	5.467E-1 (2.1E-2)†	5.319E-1 (2.7E-2)†	5.436E-1 (2.4E-2)†	5.738E-1 (2.1E-2)
10	1.385E-1 (3.6E-2)†	1.017E-1 (2.6E-2)†	2.529E-2 (3.2E-2)†	3.224E-1 (5.4E-2)†	3.171E-1 (4.0E-2)†	3.507E-1 (5.3E-2)
15	1.396E-1 (2.9E-2)†	4.793E-2 (1.4E-2)†	2.100E-1 (3.0E-2)	2.299E-1 (3.3E-2)	2.183E-1 (2.8E-2)	2.268E-1 (3.7E-2)

"†" indicates that the result of the peer algorithm is significantly different from that of BiGE at a 0.05 level by the Wilcoxon's rank sum test.

The solutions of AGE-II and NSGA-III seem to have a good uniformity, but fail to reach some regions of the Pareto front. HypE and BiGE perform similarly. The only difference between them is that the solutions of HypE struggle to cover the problem's boundary on some objectives, while the solutions of BiGE appear to have a good coverage over the whole Pareto front.

Knapsack Problem

Table 5.4 gives the results of the six algorithms on the 0-1 knapsack problem. As can be seen from the table, BiGE generally outperforms the five peer algorithms. Specifically, for the 5- and 10-objective instances, BiGE has the best HV value, and also the difference between BiGE and its competitors is statistically significant. For the 15-objective instance, BiGE ranks the second, only outperformed by FD-NSGA-II. In addition, it is interesting to note that FD-NSGA-II, which performs worst in the WFG problems, works quite well in the knapsack problem (also in the TSP problem, as shown in Table 5.5 later). This indicates the different characteristics between continuous and combinatorial optimisation problems. Some EMO algorithms may show better behaviour on combinatorial optimisation problems if their fitness assignment strategy is particularly suitable for the structure of the integral code in the problems.

TSP Problem

The normalised HV results of the six algorithms on the three TSP test instances are shown in Table 5.5. It can be observed that BiGE performs better on the problem with a larger number of objectives. For the 5-objective TSP, AGE-II has the highest HV value, and BiGE outperforms the other four algorithms with statistical significance.

Table 5.5: Normalised HV results (mean and SD) of the six algorithms on the TSP problem. The best and the second mean among the algorithms for each problem instance is shown with dark and light grey background, respectively

Obj.	MOEA/D	NSGA-III	HypE	FD-NSGA-II	AGE-II	BiGE
5	6.004E-1 (3.1E-2)†	4.173E-1 (3.9E-2)†	5.037E-1 (2.1E-2)†	5.636E-1 (3.8E-2)†	6.345E-1 (2.5E-2)†	6.186E-1 (2.1E-2)
10	2.246E-1 (3.9E-2)†	2.295E-2 (1.1E-2)†	3.000E-1 (1.8E-2)†	4.125E-1 (8.8E-2)†	3.477E-1 (2.8E-2)†	4.523E-1 (3.0E-2)
15	4.201E-2 (1.7E-2)†	8.876E-3 (6.4E-3)†	2.144E-1 (4.1E-2)†	2.446E-1 (6.3E-2)†	1.467E-1 (2.7E-2)†	2.860E-1 (4.7E-2)

"†" indicates that the result of the peer algorithm is significantly different from that of BiGE at a 0.05 level by the Wilcoxon's rank sum test.

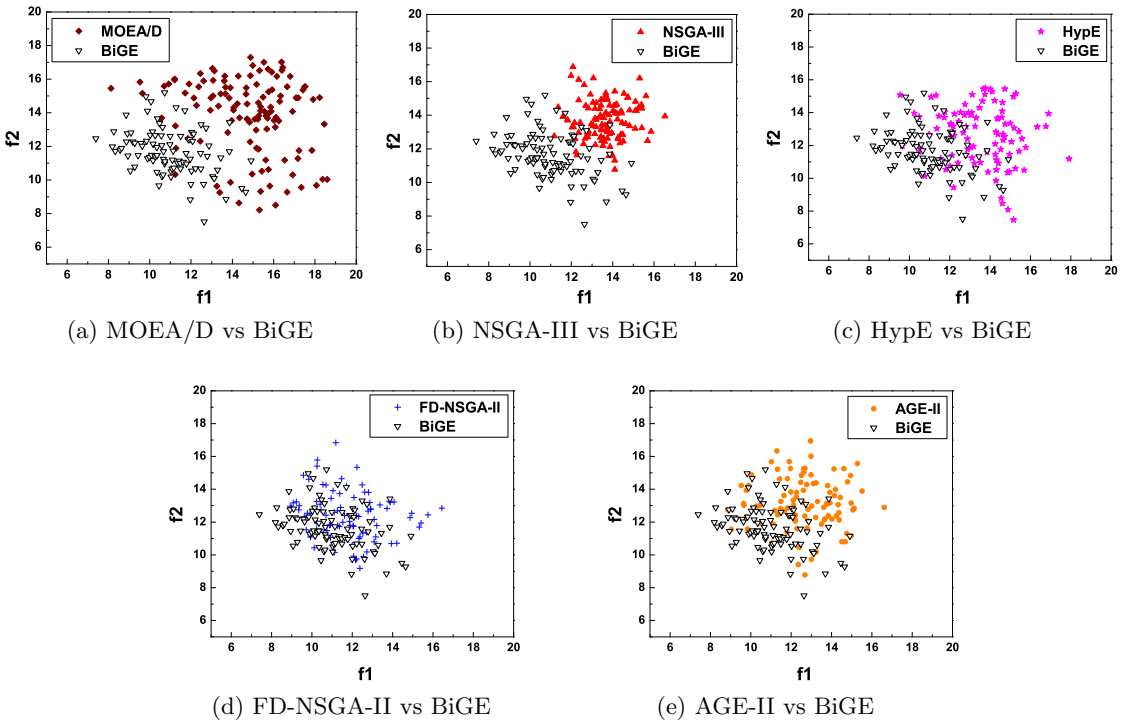
For the 10- and 15-objective instances, BiGE and FD-NSGA-II, like on the knapsack problem, perform better than the other four algorithms. A difference from the results on the knapsack problem is that here BiGE always obtains a higher HV value than FD-NSGA-II on the instances. It is worth mentioning that HypE and NSGA-III, which are competitive in the WFG problems, perform constantly worse than BiGE on all the 6 knapsack and TSP instances.

To facilitate visual comparison, Figure 5.6 plots the final solutions of a single run of the six algorithms regarding the two-dimensional objective space f_1 and f_2 of the 15-objective TSP. Similar plots can be obtained for other objectives of the problem. As shown, the solutions of BiGE have a good balance between proximity and diversity. In contrast, the five peer algorithms struggle in terms of proximity, with their solutions being generally distributed in the top-right region of the figures.

Water Problem

The water problem is a three-variable, five-objective, seven-constraint real-world problem [202, 224], which was designed to optimise the planning for a storm drainage system in an urban area. It is frequently used in the area to challenge EMO algorithms in dealing with a problem with many objectives and constraints [50, 240, 234, 139]. Table 5.6 gives the HV results of the six algorithms on this problem. As shown, BiGE outperforms the five peer algorithms with statistical significance. This indicates the effectiveness of the proposed bi-goal evolution in dealing with a problem with many objectives and constraints.

To sum up, BiGE generally outperforms the five state-of-the-art algorithms, with the best and second best HV results in 19 and 12 out of all the 34 test instances,



(a) MOEA/D vs BiGE (b) NSGA-III vs BiGE (c) HypE vs BiGE

(d) FD-NSGA-II vs BiGE (e) AGE-II vs BiGE

Figure 5.6: Result comparison between BiGE and each of the other five algorithms on the 15-objective TSP. The final solutions of the algorithms are shown regarding the two-dimensional objective space f_1 and f_2.

respectively. The five peer algorithms perform differently on problems with distinct properties. HypE and NSGA-III perform well on continuous MOPs, while FD-NSGA-II is competitive for combinatorial ones. AGE-II and MOEA/D work fairly well on 5-objective instances, but perform poorly in a higher-dimensional objective space. Similar observations have been reported in some recent studies [264, 180, 282, 260].

5.4 Further Investigations

The experimental results in the previous section have shown the effectiveness of BiGE on diverse problems. Next, we will further examine BiGE by investigating the effect of parameter setting on the algorithm performance and comparing it with some algorithms that have similar components to the proposed algorithm.

Table 5.6: Normalised HV results (mean and SD) of the six algorithms on the water problem. The best and the second mean among the algorithms for each problem instance is shown with dark and light grey background, respectively

MOEA/D	NSGA-III	HypE	FD-NSGA-II	AGE-II	BiGE
8.589E-1 (9.1E-3)†	9.176E-1 (2.9E-3)†	9.133E-1 (3.8E-3)†	1.982E-1 (1.8E-3)†	8.960E-1 (1.5E-3)†	9.273E-1 (4.1E-3)

"†" indicates that the result of the peer algorithm is significantly different from that of BiGE at a 0.05 level by the Wilcoxon's rank sum test.

5.4.1 Effect of the Population Size and Objective Dimensionality

In BiGE, two parameters, the population size and the number of objectives, play an important role. They determine the niche radius in the crowding degree estimation of the algorithm. In this section, we investigate the effect of these two parameters on the algorithm performance. Here, we show experimental results on WFG9, one of the most challenging test problems (this can be inferred from the HV values in Table 5.3). Similar results can also be observed on other problems.

First, we consider the effect of the population size on the performance of the six algorithms. The population size in the previous studies was fixed to 100. In this study, we give a wide range of the population size (from 50 to 1000) to test how the performance of the algorithm varies with it. Other parameters are kept unchanged in this study, except the function evaluations which are changed accordingly in order to keep the number of generations (300) fixed. Figure 5.7 shows the HV results on the 10-objective WFG9. Clearly, except FD-NSGA-II, the HV result of all the algorithms increases with the population size, which means that a larger population size generally leads to a better performance. This is shown more evidently in AGE-II, NSGA-III, and MOEA/D. On the other hand, HypE and BiGE always outperform other four algorithms under all the seven settings of the population size. More specifically, HypE has the best HV when the population size is 50, while BiGE performs best for the remaining cases. Overall, the above results indicate the insensitivity of the proposed algorithm to the population size – BiGE can work well under various sizes of the evolutionary population.

Next, we consider the effect of the objective dimensionality on the performance of the six algorithms. In the previous studies, the algorithms have already been tested

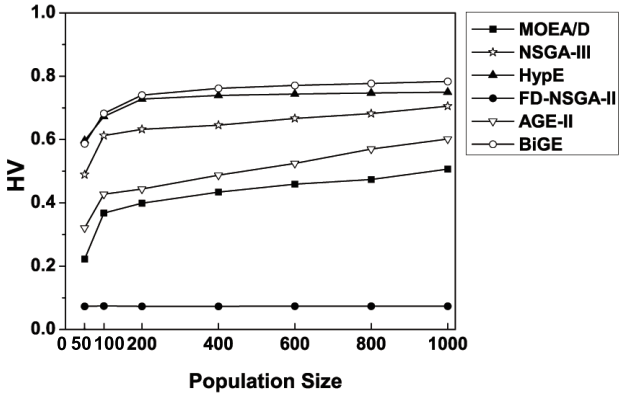


Figure 5.7: Normalised HV of the six algorithms with different settings of the population size on the 10-objective WFG9.

under 5, 10, and 15 objectives. Here, we extend the range of the number of objectives and investigate how the algorithms work in a lower- or higher-dimensional space. Figure 5.8 shows the HV results of the six algorithms on the 3-, 4-, 5-, 7-, 10-, 15-, and 20-objective WFG9. As shown, NSGA-III, HypE, and BiGE outperform the other algorithms under all the seven settings of the number of objectives. Taking a closer comparison among these three algorithms, NSGA-III and HypE perform best for the problem with 3 to 5 objectives, while BiGE shows its advantage when the number of objectives reaches 10. In addition, an interesting difference of BiGE from the other algorithms is that its HV value remains quite steady (rather than degrades) with the increase of the number of objectives. This occurrence could be attributed to the fact that the bi-goal evolution can provide a good balance between proximity and diversity, which is largely independent of a problem's objective dimensionality.

5.4.2 Effect of the Sharing Discriminator in the Sharing Function

A feature in BiGE is that a sharing discriminator is introduced to differentiate individuals in a niche. When calculating the sharing function of two neighbouring individuals, one with better proximity is encouraged by multiplying 0.5, while the other is discouraged by multiplying 1.5 (here we denote this sharing discriminator as (sd_e, sd_d)). This adjustment can lead the individual with better proximity to have a lower crowding degree and the individual with worse proximity to have a higher one. Now a straight-

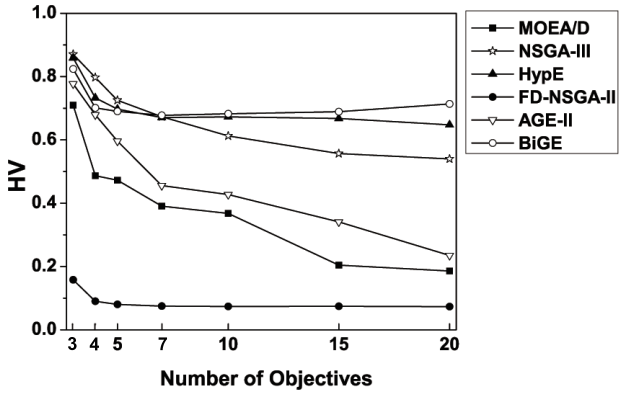


Figure 5.8: Normalised HV of the six algorithms with different settings of the number of objectives on WFG9.

forward question is how much (sd_e, sd_d) affects the performance of the algorithm. In addition, one may also ask if we can only discourage the individual with worse proximity while remaining the other unchanged, such as (sd_e, sd_d) being set to $(1.0, 1.5)$. In this case, two neighbouring individuals can also be well differentiated.

In this section, we investigate the effect of the sharing discriminator and attempt to answer the above two questions. Here, we show the results on the 10-objective WFG9. Similar results can be obtained for other problems. Here, we consider four representative settings of the discriminator: $(0.0, 2.0)$, $(0.25, 1.75)$, $(0.75, 1.25)$, and $(1.0, 1.0)$. The setting $(0.0, 2.0)$ is an extreme where the individual with better proximity is assigned zero sharing function value, while $(1.0, 1.0)$ is the other extreme where neither of the individuals' sharing function value is changed. The settings $(0.25, 1.75)$ and $(0.75, 1.25)$ are two middle values between the extremes and the setting $(0.5, 1.5)$ used in the experiments. Table 5.7 gives the HV results of BiGE with the above four settings, along with $(0.5, 1.5)$, on the 10-objective WFG9. As shown, the algorithm with the three settings $(0.25, 1.75)$, $(0.5, 1.5)$, and $(0.75, 1.25)$ performs very similarly, and all significantly outperform the algorithm with the two extreme settings $(0.0, 2.0)$ and $(1.0, 1.0)$. This indicates the insensitivity of the algorithm to the discriminator parameter within a certain range – BiGE can work well with different discriminator values, provided that they are away from the two extremes.

Next, we consider the case that only the individual with worse proximity is dis-

Table 5.7: Normalised HV of BiGE with different settings of the sharing discriminator on the 10-objective WFG9.

(sd_e, sd_d)	Normalised HV
(0.00, 2.00)	6.211E–1 (1.5E–2)
(0.25, 1.75)	6.822E–1 (1.4E–2)
(0.50, 1.50)	6.824E–1 (1.3E–2)
(0.75, 1.75)	6.806E–1 (1.3E–2)
(1.00, 1.00)	5.650E–1 (1.5E–2)

Table 5.8: Normalised HV of BiGE with the settings of the sharing discriminator that only discourage the individual with worse proximity on the 10-objective WFG9.

(sd_1, sd_2)	Normalised HV
(1.00, 1.25)	6.720E–1 (1.3E–2)
(1.00, 1.50)	6.773E–1 (1.2E–2)
(1.00, 1.75)	6.758E–1 (1.3E–2)
(1.00, 2.00)	6.732E–1 (1.2E–2)
(0.50, 1.50)	6.824E–1 (1.3E–2)

couraged in the sharing function. That is, sd_e is set to 1.0 and sd_d to larger than 1.0. Here, we consider four settings of the discriminator: $(1.0, 1.25)$, $(1.0, 1.5)$, $(1.0, 1.75)$, and $(1.0, 2.0)$. The HV results of BiGE with them are given in Table 5.8, where the result of the algorithm with $(0.5, 1.5)$ is repeated for comparison. As can be seen from the table, the algorithm where the individual with better proximity is not encouraged performs slightly worse than the original algorithm. A probable explanation for this is as follow. In general, for a group of individuals in a niche, it is ideal to select a representative individual (i.e., with the best proximity) from them into the next population. However, with the discriminator setting that only discourages individuals with worse proximity, all the individuals in the niche could have a high crowding degree (in comparison with those having no neighbour in their own niche). This may lead to none of the individuals in this niche surviving in the next population. Thus, an encouragement for the individual with better proximity in the niche is beneficial to the diversity of the population – it further differentiates similar individuals and enables a representative one to be preserved in the evolutionary process.

5.4.3 Comparison with Average Ranking Methods

In BiGE, the proximity of an individual is estimated by the sum of its normalised values across the objectives. This estimation could be viewed as a slightly more fine-grained version of the well-known average ranking (AR) method [15]. AR estimates the proximity of an individual by summing its ranks (in the population) across the objectives. The difference between these two estimations is that AR considers individuals' rank in the population on each objective, while the BiGE proximity estimation considers quantitative difference of individuals on each objective.

As an individual comparison criterion, AR is popular in many-objective optimisation. Corne and Knowles have demonstrated that AR can provide sufficient selection pressure towards the optimal front in a high-dimensional objective space [45]. However, due to the lack of a diversity maintenance scheme, AR may lead the evolutionary population to converge into a sub-area of the Pareto front [138]. Recently, some methods have been proposed to enhance the diversity for AR. For example, Purshouse *et al.* made a modification of the AR-based fitness by combining it with a sharing scheme based on the Epanechnikov kernel [217]. Li *et al.* imposed a punishment on individuals who are neighbours of the best-AR individual to prohibit or postpone their entry in the next population [184]. Kong *et al.* repeatedly initialised the population by a chaotic method after some generations, in order to enhance the diversity of individuals in the decision space [154]. Instead of considering the objectives in the original AR, Yuan *et al.* summed up the aggregation function values based on uniformly-distributed weight vectors [274].

A clear difference of BiGE from these AR-based algorithms is that BiGE uses the idea of Pareto dominance to deal with proximity and diversity. This could be well suited to many-objective optimisation where the conflict between proximity and diversity goals is more serious than that in bi- or tri-objective optimisation. Considering the dominance relation of these two goals can provide a good balance between them and lead the algorithm to be less affected by the increase of the objective dimensionality.

Next, we empirically investigate the difference between BiGE and some AR-based algorithms. Specifically, we consider three peer algorithms: (1) the original AR [15],

(2) AR combined with a fitness sharing scheme (called AR+sharing here) [217], and (3) a new version of BiGE where AR is used as the proximity estimation method, denoted as BiGE(AR). In [217], AR has been found to be competitive when combined with a sharing scheme based on the Epanechnikov kernel [73]. From some initial experiments, we found that replacing the Epanechnikov kernel with the proposed niching method, the algorithm can obtain very similar results. Therefore, here the proposed niching method is used in AR+sharing in order to investigate the difference of the algorithm framework. That is, AR+sharing and BiGE(AR) have the same proximity and crowding degree estimation methods and the only difference between them is the algorithm framework. In addition, it is worth noting that BiGE(AR) and BiGE have the same algorithm framework and the only difference between them is their proximity estimation.

Table 5.9 gives the HV results of the three algorithms on all the 34 test instances; for comparison, the results of BiGE are also included in the table. As shown, the diversity mechanism dramatically improves the HV results, with AR+sharing, BiGE(AR) and BiGE outperforming the original AR on all the 34 test instances. This suggests the importance of diversity maintenance in many-objective optimisation.

Regarding the two algorithms having same proximity and crowding degree estimators, BiGE(AR) performs better than AR+sharing in 31 out of the 34 instances. This clearly indicates the advantage of the bi-goal evolution framework for many-objective problems. In addition, note that AR+sharing has the best HV result for the three WFG2 instances, which have a disconnected Pareto front. This is because AR+sharing can always find all the optimal regions of the Pareto front in all the 30 runs, while BiGE(AR) and BiGE can only do so in around half of the 30 runs.

Finally, consider the comparison between two versions of the bi-goal evolution algorithm. BiGE(AR) outperforms BiGE on 10 test instances (including 7 five-objective instances), while BiGE has better HV results on the remaining 24 instances. An interesting observation is that the less fine-grained algorithm BiGE(AR) generally performs better on five-objective instances. One possible explanation for this is that BiGE(AR) could prefer some boundary individuals in a population. These boundary solutions, which perform rather poorly on one objective but best (or nearly best) on other ob-

Table 5.9: Normalised HV results (mean and SD) of the four algorithms on all the 34 test instances. The best and the second mean among the algorithms for each problem instance is shown with dark and light grey background, respectively

Problem	Obj.	AR	AR+sharing	BiGE(AR)	BiGE
WFG1	5	1.553E–2 (6.9E–2)†	5.001E–1 (1.4E–2)†	5.530E–1 (2.8E–2)†	6.151E–1 (3.9E–2)
	10	1.577E–1 (1.4E–1)†	4.842E–1 (2.9E–2)†	5.998E–1 (3.9E–2)†	7.786E–1 (4.7E–2)
	15	4.961E–1 (2.9E–1)†	4.716E–1 (3.1E–2)†	6.001E–1 (3.7E–2)†	7.612E–1 (3.4E–2)
WFG2	5	1.714E–1 (7.6E–2)†	9.663E–1 (8.2E–3)†	9.255E–1 (8.4E–2)	9.014E–1 (8.2E–2)
	10	2.276E–1 (9.0E–2)†	9.717E–1 (1.4E–2)†	9.715E–1 (5.2E–2)	9.475E–1 (5.3E–2)
	15	2.442E–1 (9.9E–2)†	9.587E–1 (1.9E–2)	9.541E–1 (6.3E–2)	9.402E–1 (6.1E–2)
WFG3	5	2.274E–1 (2.9E–2)†	7.470E–1 (3.7E–2)†	8.975E–1 (1.4E–2)†	9.092E–1 (1.1E–2)
	10	2.001E–1 (2.8E–2)†	7.486E–1 (5.5E–2)†	8.609E–1 (2.1E–2)†	8.705E–1 (1.8E–2)
	15	1.868E–1 (3.9E–2)†	7.559E–1 (4.7E–2)†	8.349E–1 (2.3E–2)†	8.545E–1 (2.6E–2)
WFG4	5	1.351E–1 (1.7E–2)†	6.745E–1 (1.5E–2)†	8.340E–1 (6.5E–3)†	8.117E–1 (8.4E–3)
	10	1.114E–1 (1.3E–2)†	6.455E–1 (2.3E–2)†	8.304E–1 (1.1E–2)	8.313E–1 (1.1E–2)
	15	1.053E–1 (1.1E–2)†	5.448E–1 (3.6E–2)†	7.829E–1 (2.7E–2)†	8.073E–1 (1.9E–2)
WFG5	5	1.210E–1 (2.1E–2)†	6.634E–1 (1.4E–2)†	7.778E–1 (7.4E–3)†	7.709E–1 (6.2E–3)
	10	9.355E–2 (1.1E–2)†	6.754E–1 (1.8E–2)†	7.909E–1 (1.1E–2)†	7.990E–1 (1.9E–2)
	15	9.344E–2 (1.0E–2)†	6.224E–1 (2.4E–2)†	7.870E–1 (1.3E–2)†	7.715E–1 (1.4E–2)
WFG6	5	1.196E–1 (2.3E–2)†	6.283E–1 (1.9E–2)†	7.891E–1 (9.3E–3)†	7.728E–1 (8.9E–3)
	10	9.997E–2 (2.4E–2)†	6.607E–1 (2.4E–2)†	8.190E–1 (1.0E–2)†	8.270E–1 (1.3E–2)
	15	1.019E–1 (2.6E–2)†	6.673E–1 (3.0E–2)†	8.120E–1 (1.8E–2)†	8.339E–1 (1.4E–2)
WFG7	5	2.239E–1 (7.4E–2)†	6.876E–1 (1.5E–2)†	8.426E–1 (6.8E–3)†	8.356E–1 (5.5E–3)
	10	2.166E–1 (6.0E–2)†	7.303E–1 (2.1E–2)†	8.489E–1 (1.2E–2)†	8.827E–1 (1.2E–2)
	15	1.734E–1 (5.1E–2)†	6.858E–1 (3.5E–2)†	8.182E–1 (1.9E–2)†	8.787E–1 (1.3E–2)
WFG8	5	1.832E–1 (3.0E–2)†	5.064E–1 (1.4E–2)†	6.833E–1 (8.9E–3)	6.822E–1 (9.1E–3)
	10	1.581E–1 (2.5E–2)†	5.769E–1 (2.7E–2)†	7.478E–1 (8.6E–3)†	7.722E–1 (6.1E–3)
	15	1.449E–1 (3.0E–2)†	6.427E–1 (2.8E–2)†	7.843E–1 (1.4E–2)†	8.179E–1 (1.0E–2)
WFG9	5	1.218E–1 (4.4E–2)†	6.078E–1 (1.4E–2)†	6.910E–1 (8.7E–3)	6.903E–1 (1.4E–2)
	10	9.933E–2 (3.7E–2)†	6.314E–1 (1.9E–2)†	6.742E–1 (2.1E–2)†	6.824E–1 (1.3E–2)
	15	1.131E–1 (5.7E–2)†	6.122E–1 (2.6E–2)†	6.616E–1 (3.5E–2)†	6.893E–1 (3.1E–2)
Knapsack	5	4.199E–1 (3.2E–2)†	4.538E–1 (2.9E–2)†	5.652E–1 (2.1E–2)	5.738E–1 (2.1E–2)
	10	1.409E–1 (3.6E–2)†	2.879E–1 (1.3E–2)†	3.126E–1 (3.9E–2)†	3.507E–1 (5.3E–2)
	15	7.471E–2 (1.9E–2)†	1.403E–1 (3.9E–2)†	2.096E–1 (3.3E–2)†	2.268E–1 (3.7E–2)
TSP	5	2.850E–1 (3.0E–2)†	3.295E–1 (3.2E–2)†	5.788E–1 (2.9E–2)†	6.186E–1 (2.1E–2)
	10	1.251E–1 (2.4E–2)†	1.837E–1 (2.4E–2)†	4.094E–1 (3.3E–2)†	4.523E–1 (3.0E–2)
	15	5.187E–2 (2.7E–2)†	1.013E–1 (5.5E–2)†	2.367E–1 (4.8E–2)†	2.860E–1 (4.7E–2)
Water	5	7.904E–1 (7.7E–3)†	8.668E–1 (5.6E–3)†	9.213E–1 (8.0E–3)†	9.273E–1 (4.1E–3)

"†" indicates that the result of the peer algorithm is significantly different from that of BiGE at a 0.05 level by the Wilcoxon's rank sum test.

jectives, play an important role in extending the search range. Due to having no consideration of quantitative difference of individuals, BiGE(AR) would be in favour of these solutions. Nevertheless, it is worth pointing out that a fine-grained estimation of individual proximity can be more important in a high-dimensional space, where there is more need of clear distinction between individuals.

5.5 Summary

In this chapter, we presented a new framework of EAs, called BiGE, to deal with many-objective optimisation problems. Converting many objectives of a given problem into two objectives of proximity and crowding degree, BiGE creates an optimisation problem in which the objectives are the goals of the search process itself.

Systematic experiments were carried out by providing extensive comparative studies between BiGE and five state-of-the-art algorithms on four groups of well-defined continuous and combinatorial benchmark suites with 5, 10, and 15 objectives. Unlike some peer algorithms, which work well on only a fraction of the test problems (e.g., AGE-II and MOEA/D on the 5-objective instances, HypE and NSGA-III on the continuous instances, and FD-NSGA-II on the combinatorial instances), BiGE can achieve a good balance between solutions' proximity and diversity on the test problems with different properties. In addition, the effect of several parameters on the algorithm was investigated. Experimental results have indicated the insensitivity of BiGE to the population size and objective dimensionality as well as the effectiveness of BiGE under different settings of the sharing discriminator within a certain range. Finally, a comparison with three AR-based algorithms has shown the advantage of the proposed framework and proximity estimation in dealing with MaOPs.

It is worth pointing out that the niche radius in the paper is a rough setting (estimate) according to the population size and the number of objectives. A finely tuned setting based on the characteristics of a given MOP, such as varying with the Pareto front's shape, may lead to a better performance of BiGE. Nevertheless, this fixed setting can benefit the applicability of BiGE to real-world problems as it could be hard (or even impossible) to know the problems' characteristics beforehand.

Chapter 6

A Performance Indicator

In this chapter, we present a performance comparison indicator (PCI) to assess the performance of Pareto front approximations (i.e., solution sets obtained by stochastic search algorithms) in multi-objective optimisation. PCI works for optimisation problems with any number of objectives and can be particularly practical in many-objective optimisation.

This chapter is organised as follows. Section 6.1 introduces the performance assessment in many-objective optimisation, with a particular focus on why there exists little work in this area. Section 6.2 reviews related works in multi-objective optimisation and further analyses their difficulties in many-objective optimisation. Section 6.3 is devoted to the description of the proposed indicator. Analytic and empirical studies are carried out in Section 6.4 and Section 6.5, respectively. Finally, Section 6.6 concludes the chapter.

6.1 Introduction

Performance indicators, which assess the performance of Pareto front approximations, play an important role in multi-objective optimisation. They not only are used to examine and compare EMO algorithms but also are capable of guiding the search during the evolutionary process of an algorithm [291].

During the past two decades, a variety of performance indicators have been proposed

in multi-objective optimisation [75, 92, 295, 207, 153, 290]. However, how to assess and compare Pareto front approximations in the context of many-objective optimisation has received scant concern [254]. Many performance indicators are designed in principle for any number of objectives, but in practice are invalid or infeasible when dealing with Pareto front approximations with high dimensions. In general, the difficulties of comparing high-dimensional Pareto front approximations can be summarised as follows:

- *Difficulties of visual comparison.* When the number of objectives is larger than three, visual and intuitive performance assessment becomes difficult or even misleading, despite the fact that it is a prevailing comparison tool in multi-objective optimisation.

- *Ineffectiveness of Pareto dominance.* Since the portion of the space that a solution dominates decreases exponentially with the number of objectives, most solutions in different approximation sets are likely to be incomparable under the criterion of Pareto dominance. This can lead to the ineffectiveness of many performance indicators based on the comparison of solutions' Pareto dominance relation, such as the *coverage* [294] and G-metric [189].

- *Rapid increase of time or space requirement.* The time or space required by some performance indicators increases exponentially with the number of objectives, such as the *hypervolume* [294], *hyperarea radio* [251] and *diversity measure* [53]. This can affect their application in many-objective optimisation.

- *Difficulties of parameter setting.* As shown in Purshouse and Fleming [216], the *sweet-spot* of algorithm parameter setting that produces good results could shrink markedly in many-objective optimisation. This could also apply to parameter setting of performance indicators. In general, the sensitivity of assessment results to an indicator's parameter(s) increases with the number of objectives. Two high-dimensional approximation sets could return completely contrary results when assessed by an indicator with slightly different settings of its parameter(s) [174].

- *Difficulties of the substitution of Pareto front.* Many performance indicators re-

quire a reference set as a substitution of the Pareto front, such as GD [251] and IGD [19, 42]. However, to represent a higher-dimensional Pareto front, exponentially more points are needed. As shown by Ishibuchi et al. [123], insufficient points can easily lead to an inaccurate or even a misleading assessment result.

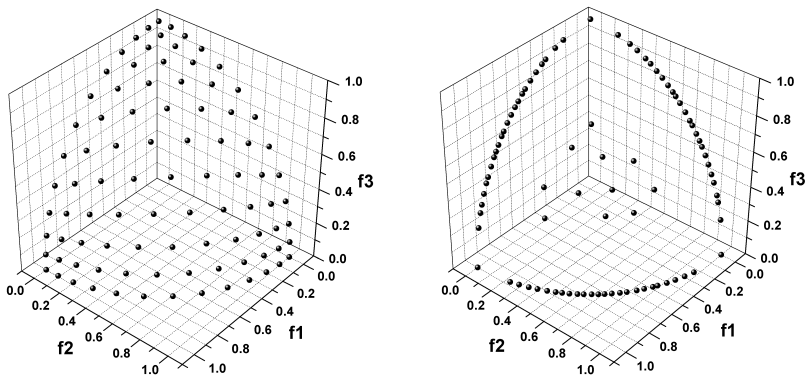
In this chapter, we propose a performance indicator to compare Pareto front approximations obtained by many-objective algorithms. The proposed indicator evaluates the relative quality of approximation sets with the aid of a reference set constructed by themselves. The points in the reference set are divided into many clusters, and the indicator estimates the minimum moves of solutions in the approximation sets to weakly Pareto dominate these clusters.

6.2 Related Work

Despite the difficulties in assessing high-dimensional Pareto front approximations, some effort has been made. Jaimes and Coello [132] measured the Tchebycheff distance of an approximation set to the "knee" of the Pareto front in many-objective optimisation. This measurement is based on the assumption that, in the absence of particular preference information, the decision maker may have more interest in the knee of a problem's Pareto front. Recently, Li et al. [174] proposed an indicator to compare the diversity of approximation sets. They put all approximation sets into a grid environment and calculated the contribution of each set to those non-empty hyperboxes. A problem in these indicators is that they only focus on one particular aspect of the performance of approximation sets, failing to provide a comprehensive evaluation of the sets' performance.

On the other hand, some classic performance indicators which measure the overall quality of approximation sets have been frequently used to test and compare many-objective optimisers [88, 180, 256]. Examples are the hypervolume [294], ϵ-indicator [295], and IGD [19] (also its variation IGD$^+$ [123]). Next, we will briefly discuss their pros and cons, especially when assessing high-dimensional approximation sets.

The hypervolume (HV) indicator [294] calculates the volume of the objective space



(a) MOEA/D (b) IBEA

Figure 6.1: An example that HV prefers the knee and boundary points on the Pareto front, where two sets of Pareto optimal solutions on DTLZ2 are obtained by MOEA/D and IBEA. The solution set with better distribution (obtained by MOEA/D) has a worse (lower) HV result, as given in Table 6.1.

enclosed by an approximation set and a reference point. HV has good theoretical properties and can give a general evaluation of the set, but its computational complexity increases exponentially with the number of objectives. These lead to its dominant application in the 2- or 3-objective problems. While the Monte Carlo method can largely reduce the time cost [10, 25], how to choose a proper reference point in the HV calculation is an important issue [151] and its difficulty also increases with the number of objectives. In addition, the HV indicator generally prefers the knee and boundary points of the Pareto front to well-distributed ones. Figure 6.1 gives such an example, where we consider two sets of Pareto optimal solutions[1] obtained by two well-known EMO algorithms, MOEA/D [278] and IBEA [291], on problem DTLZ2 [57]. Table 6.1 gives the HV results of these two solution sets with various settings of the reference point. As shown, the solution set with better distribution (obtained by MOEA/D) has a worse (lower) HV evaluation value than its competitor, regardless of the choice of the reference point.

The unary ϵ-indicator [295] measures the minimum factor ϵ for an approximation set such that any point in a reference set is ϵ-dominated [165] (additively or multi-

[1]Here, the number of DTLZ2's decision variables n is set to $m-1$ (m is the number of objectives) to ensure that all solutions produced by the algorithms are Pareto optimal.

Table 6.1: HV results of the two sets in Figure 6.1 under different reference points. The range of DTLZ2's Pareto front is $[0, 1]$ for all objectives.

Reference point	MOEA/D	IBEA
(1.0, 1.0, 1.0)	4.1413E–1	4.1525E–1
(1.1, 1.1, 1.1)	7.4484E–1	7.4596E–1
(1.2, 1.2, 1.2)	1.1418E+0	1.1430E+0
(1.4, 1.4, 1.4)	2.1578E+0	2.1590E+0
(1.7, 1.7, 1.7)	4.3268E+0	4.3280E+0
(2.0, 2.0, 2.0)	7.4138E+0	7.4150E+0
(2.5, 2.5, 2.5)	1.5039E+1	1.5040E+1
(3.0, 3.0, 3.0)	2.6414E+1	2.6415E+1

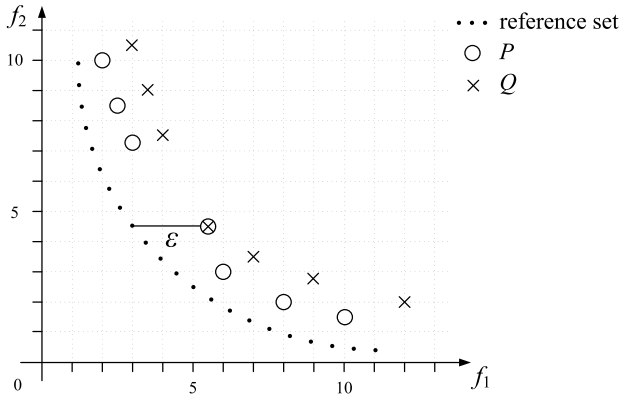


Figure 6.2: An example that the unary additive ϵ-indicator fails to distinguish between two approximation sets. P and Q have the same evaluation result ($\epsilon = 2.5$).

plicatively) by at least one solution in the approximation set. One weakness in the ϵ-indicator is that its evaluation value is only related to one particular solution in an approximation set, which could lead to an inaccurate estimation of the set's performance. Figure 6.2 gives an example that the unary additive ϵ-indicator fails to distinguish between approximation sets (P and Q). As can be seen from the figure, P clearly outperforms Q, but the two sets have the same evaluation result.

Zitzler et al. [295] also presented a binary ϵ-indicator and stated its desirable features, such as being free from the limitations of unary performance indicators, having a low computational cost, and representing natural extension to the evaluation of approximation schemes in theoretical computer science. However, the ϵ-indicator (both unary and binary) only considers one particular objective of the problem (i.e., the ob-

jective where the considered approximation sets have the biggest difference), ignoring the difference on the others. This unavoidably leads to an information loss, especially for optimisation problems with many objectives. Consider two 10-objective solutions, $p = (0, 0, 0, ..., 0, 1)$ and $q = (1, 1, 1, ..., 1, 0)$. p performs better on nine objectives and q does better only on the last objective, but they have the same evaluation result ($\epsilon = 1$).

IGD [19] calculates the average Euclidean distance from each point in a reference set (a substitution of the Pareto front) to its closest solution in an approximation set. This overcomes the ϵ-indicator's problem of only returning one particular objective value of one particular solution. However, one weakness of IGD is its non-compliance with the Pareto dominance relation. Ishibuchi et al. [123] presented several examples that an approximation set P could obtain a worse IGD result than an approximation set Q even if P Pareto-dominates Q. This even happens when the reference set is exactly the Pareto front. In fact, a reference set with sufficient, well-distributed points can only reduce the possibility of this misleading result, rather than eliminating it completely.

To solve this problem, Ishibuchi et al. [121, 123] proposed a modified IGD (IGD^+), only considering the superior objective values of the reference point to the solution of the approximation set in their distance calculation. This can enable the indicator to be weakly Pareto compliant[2] [123]. However, like IGD, IGD^+ also needs a reference set specified by the user. While IGD^+ alleviates the severe sensitivity of IGD to the reference set, different reference sets can cause IGD^+ to prefer different approximation sets [120]. This may lead to inconsistent evaluation results among these sets. In fact, how to specify a proper preference set in many-objective optimisation is a challenging issue [121]; this also applies to artificial test functions with a known Pareto front, given that exponentially increasing points are needed to accurately represent a higher-dimensional Pareto front.

For an MOP with an unknown Pareto front, a practical method of constructing a reference set is to use the nondominated solutions of all solutions in the considered approximation sets. This method is widely used in many-objective optimisation [108, 121]. However, in this method, there is no information about the location and

[2]For a weakly Pareto compliant indicator I, if a set P weakly dominates a set Q, then $I(P)$ is not worse than $I(Q)$ [295].

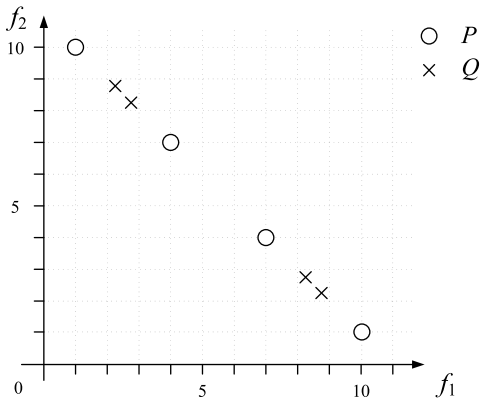


Figure 6.3: An example that IGD and IGD$^+$ fails to reflect the performance difference between approximation sets, where the reference set is constructed by the approximation sets themselves. P and Q have the same IGD and IGD$^+$ evaluation results (0.884 and 0.625 respectively).

distribution of the points in the reference set. Some points could be far away from the Pareto front, like the *dominance resistant solutions*[3] [57, 113] which are preserved more likely in a higher-dimensional objective space. Some points could be located closely or even overlapping, and this will result in some areas overcrowded and some others empty. Overall, such a reference set with an arbitrary distribution of nondominated solutions can significantly decrease the accuracy of the evaluation result. Figure 6.3 gives an example that with the reference set constructed by the tested approximation sets, IGD and IGD$^+$ fail to reflect the performance difference between the sets. As shown, the points of the reference set are not well distributed, which leads to the result that P with uniformly-distributed solutions has the same evaluation value as Q whose solutions concentrate in two small areas.

6.3 The Proposed Approach

As indicated before, the binary ϵ-indicator directly considers the performance difference of approximation sets, but only returns one particular objective value of one particu-

[3]Dominance resistant solutions in a set are the solutions with a quite poor value in at least one of the objectives but with (near) optimal values in some others, and thus are nondominated in the set.

lar solution; IGD^+ (or IGD) considers the whole approximation set and multiple (or all) objectives, but requires a reference set which is hard to specify in many-objective optimisation. So, an intuitive idea of developing an indicator is to consider their combination, that is, using the calculation method of IGD^+ to directly compare two approximation sets (i.e., two approximation sets are viewed as mutual reference sets in the IGD^+ calculation). This indicator, though, also suffers from the effect of a poorly-distributed reference set, e.g., still failing to distinguish between the approximation sets in Figure 6.3. In fact, since each point (regardless of its location) in the reference set is equally treated in the evaluation, the areas with many similar (or duplicate) points have more effect than the same-size ones with few points on the evaluation result. This naturally leads such an indicator to prefer an approximation set having similar distribution with the considered reference set. In addition, since a binary indicator manipulates only two approximation sets, many comparisons are required when more than two approximation sets are involved ($\binom{n}{2}$ comparisons for n sets).

Given the above, this chapter presents a performance comparison indicator (PCI), capable of assessing multiple approximation sets (any number) in a single run. PCI constructs a reference set by using all the tested Pareto front approximations and assesses each approximation on the basis of this reference set. Rather than dealing with each point in the reference set, PCI divides all points into many clusters according to their distribution (the method of the division will be explained later), and considers the relationship between the clusters and solutions in the approximation set. Specifically, PCI measures the minimum move distance of one solution to weakly dominate[4] all points in a cluster, as defined as follows.

Definition 6.3.1 (Dominance distance of a point to a point set)**.** *Let p be a point and Q be a set of points $\{q_1, q_2, ..., q_k\}$ ($k \geq 1$). The dominance distance $D(p, Q)$ of p to Q is defined as the minimum move of p in the objective space such that p weakly dominates all points in Q.*

[4]For two solutions p and q, p weakly dominates q ($p \preceq q$) if and only if p is not worse than q in all objectives.

$D(p, Q)$ can be calculated as:

$$D(p, Q) = \sqrt{\sum_{i=1}^{m} (p^{(i)} - d(p^{(i)}, Q))^2} \tag{6.1}$$

$$d(p^{(i)}, Q) = \begin{cases} \min\{q_1^{(i)}, q_2^{(i)}, ..., q_k^{(i)}\}, & \text{if } p^{(i)} > \min\{q_1^{(i)}, q_2^{(i)}, ..., q_k^{(i)}\} \\ p^{(i)}, & \text{otherwise} \end{cases}$$

where $p^{(i)}$ denotes the objective value of point p in the ith objective and m is the number of objectives.

Dominance distance $D(p, Q)$ only considers the objectives where Q performs better than p (i.e., the best value of the points in Q is better than that of p), regardless of the advantage of p over Q. This can lead to the indicator free from the effect of poorly-converged reference points, such as the dominance resistant solutions. The range of $D(p, Q)$ is from 0 to ∞; the smaller the better. If p performs slightly worse than Q in only a few objectives, $D(p, Q)$ will be small. Only when p weakly dominates all points in Q, $D(p, Q) = 0$.

Note that when Q has only one point q, the dominance distance of p to Q becomes $D(p, q)$, i.e., the minimum move of p to weakly dominate q in the objective space. This method has already been used in the area to measure the difference between two solutions. For example, IGD$^+$ used this measure to replace the Euclidean distance in the IGD calculation, making the indicator compliant with weak Pareto dominance. In fact, $D(p, Q)$ can also be viewed as the dominance distance of p to a point, q_{ideal}, constructed by the best value of each objective for all k points in Q.

In contrast to IGD and IGD$^+$ which consider each point in the reference set, PCI replaces a set of similar points with an imaginary (ideal) point constructed by themselves. This avoids the overlapping effect of similar points and also takes into account the superiority of these points on each objective. In fact, the dominance distance of a point to a set of points is larger than or equal to the maximum dominance distance of the point to all members in the set, but is smaller than their summation, i.e., $\max\{D(p, q_1), ..., D(p, q_k)\} \leq D(p, Q) < D(p, q_1) + ... + D(p, q_k)$.

Given that the dominance distance of a point to a cluster reflects the advantage of

the cluster over the point, one may think to use this dominance distance to consider all clusters in the reference set to assess an approximation set. However, this evaluation may lead to an inaccurate result when in a cluster there is more than one solution from the assessed approximation set. To explain this, we introduce the following definition.

Definition 6.3.2 (Dominance Distance of a point set to another)**.** *Let P be a set of points and Q be a set of points. The dominance distance $D(P,Q)$ is defined as the minimum total distance of the move of points of P in the objective space such that for any point $q \in Q$, there is at least one point $p \in P$ that weakly dominates q.*

The dominance distance of two sets reflects their performance difference. If P (weakly) dominates Q, then $D(P,Q) = 0$.

In the proposed indicator, since the reference set consists of all the approximation sets, a cluster can contain points from different approximation sets. Let a cluster C be comprised of P and Q, where $P = \{p_1, ..., p_i\}$ is from the assessed approximation set and $Q = \{q_1, ..., q_j\}$ from other approximation sets. Apparently, $D(P,C) = D(P,Q)$. When $i = 1$, $D(P,C)$ is the dominance distance of p_1 to the ideal point of C. When $i \geq 2$, $D(P,C)$ could be smaller than $\min\{D(p_1,C), ..., D(p_i,C)\}$.

Figure 6.4 gives an example to illustrate this situation. Consider the dominance distance of three solution sets P_1 ,P_2, and P_3 to three clusters C_1 ,C_2, and C_3, respectively in the figure, where $P_1 \in C_1, P_2 \in C_2, P_3 \in C_3, P = P_1 \cup P_2 \cup P_3$. For P_1 which has only one solution, $D(P_1,C_1)$ is the dominance distance of the solution to the cluster ($D(P_1,C_1) = (0.5^2 + 0.5^2)^{0.5} \approx 0.707$). For P_2 which has two solutions, $D(P_2,C_2)$ is the dominance distance of the upper solution to the ideal point of the two Q solutions, and this is smaller than the minimum of the dominance distance of the two single P_2 solutions to the cluster ($0.559 < \min\{1.031, 1.25\}$). Cluster C_3 gives an extreme situation where no move of the P_3 solutions is needed to weakly dominate all solutions in the cluster (i.e., $D(P_3,C_3) = 0$), but the dominance distance of the two single P_3 solutions to the cluster is 1.0.

From the above, it is clear that for a cluster $C = P \cup Q$ ($P = \{p_1, ..., p_i\}, i \geq 2, Q = \{q_1, ..., q_j\}$), the required move of P to weakly dominate C can be (much) smaller than the minimum move of any one point in P to weakly dominate C. This is because

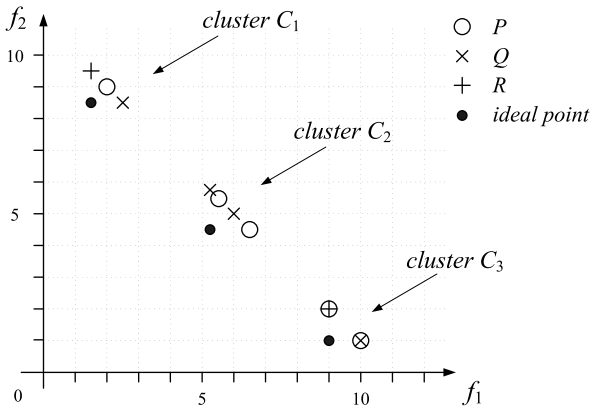


Figure 6.4: An example that the dominance distance of a set of solutions to a cluster can be smaller than the minimum of their single dominance distance to the cluster. For three sets P_1, P_2, and P_3 ($P_1 \in C_1, P_2 \in C_2, P_3 \in C_3, P = P_1 \cup P_2 \cup P_3$), their dominance distance to C_1, C_2 and C_3 is 0.707, 0.559 and 0.0, respectively, while the minimum of their single solution's dominance distance to C_1, C_2 and C_3 is 0.707, 1.031 and 1.0, respectively.

the points in C can be "divided and conquered" by multiple solutions in P. However, how to determine $D(P, C)$ (i.e., the minimum move of $\{p_1, ..., p_i\}$ to weakly dominate $\{q_1, ..., q_j\}$) could be quite time consuming; there are i^j possibilities for $p_1, p_2, ..., p_i$ to divide $q_1, q_2, ..., q_j$. Here, we approximately measure it by

$$\begin{aligned} D'(P, C) = &\max\{\min\{D(p_1, q_1), \ldots, D(p_i, q_1)\}, \\ &\ldots, \min\{D(p_1, q_j), \ldots, D(p_i, q_j)\}\} \end{aligned} \tag{6.2}$$

when $i \geq 2$. This only requires $i \times j$ comparisons. Although $D'(P, C) \leq D(P, C)$, their difference is generally slight when the size of C is small. In the example of Figure 6.4, $D'(P_2, C_2) = 0.5 < D(P_2, C_2) = 0.559$ and $D'(P_3, C_3) = D(P_3, C_3) = 0$.

Algorithm 6.1 gives the main procedure of the proposed indicator. It is necessary to mention here that we normalise approximation sets before the evaluation, in order to enable PCI to deal with MOPs with non-commensurable objective functions. Specifically, the range of an MOP's Pareto front is used to implement the normalisation if it is available; otherwise, the boundary of the constructed reference set is used. From Algorithm 6.1, it can be seen that PCI treats the clusters differently. For a cluster

Algorithm 6.1 Performance Comparison Indicator (PCI)

Require: $P^1, P^2, \ldots, P^{pn}$ (tested approximation sets)

1: $S \leftarrow NondominanceSelection(P^1, P^2, \ldots, P^{pn})$ /* Finding out the non-dominated solutions of the mixed set consisting of all the approximation sets */
2: $(C_1, C_2, ..., C_{cn}) \leftarrow Clustering(S)$
/* Clustering the points in S according to Algorithm 6.2 */
3: **for all** $P^i \in \{P^1, P^2, \ldots, P^{pn}\}$ **do**
4: PCI$(P^i) \leftarrow 0$
5: **for all** $C_j \in \{C_1, C_2, \ldots, C_{cn}\}$ **do**
6: $P^i_j \leftarrow P^i \cap C_j$
7: **if** $|P^i_j| < 2$ **then**
8: PCI$(P^i) \leftarrow$ PCI$(P^i) + \min\{D(p^i_1, C_j), ..., D(p^i_n, C_j)\}$ /* Finding out the minimum dominance distance of one solution in $P^i = \{p^i_1, ..., p^i_n\}$ to C_j */
9: **else**
10: PCI$(P^i) \leftarrow$ PCI$(P^i) + D'(P^i_j, C_j)$
/* Estimating the dominance distance of P^i_j to C_j */
11: **end if**
12: **end for**
13: PCI$(P^i) \leftarrow$ PCI$(P^i)/cn$
14: **end for**
15: **return** PCI(P^1), PCI(P^2), ..., PCI(P^{pn})

where the number of solutions from the assessed approximation set is less than two, PCI takes account of the minimum move of one solution in the approximation set to weakly dominate the cluster (step 8), on the basis of Eq. (6.1). Otherwise, PCI estimates the minimum move of the set's solutions in the cluster to weakly dominate the cluster by Eq. (6.2) (step 10).

An important issue in the proposed indicator is to cluster the points in the reference set before assessing approximation sets (step 2). Here, we use a greedy method to stepwise merge points according to their dominance distance. Algorithm 6.2 gives the details of clustering. A threshold σ is required in this clustering. We set σ to be the interval (in the sense of the dominance distance) of two neighbouring points in the normalised hyperplane with an ideal distribution of N points, where N is the size of the reference set. Since we consider two neighbouring points in the normalised hyperplane with an ideal distribution, they have an equal dominance distance to each other, also

Algorithm 6.2 *Clustering*(S)

Require: $S = \{s_1, s_2, \ldots, s_{sn}\}$, σ (threshold)

1: $(p_1, p_2, ..., p_{pn}) \leftarrow FindSortPair(S, \sigma)$
   /* Find all pairs of solutions in S satisfying that $\max\{D(s_i, s_j), D(s_j, s_i)\} \leq \sigma$ where s_i and s_j are the two solution of a pair, and then sort the pairs in the ascending order according to $\max\{D(s_i, s_j), D(s_j, s_i)\}$ */
2: $C_1 \leftarrow s_1, C_2 \leftarrow s_2, ..., C_{sn} \leftarrow s_{sn}$ /* Cluster initialization */
3: **for all** $p \in \{p_1, p_2, \ldots, p_{pn}\}$ **do**
4: $(C_m, C_n) \leftarrow Locate(p)$
   /* Let C_m and C_n be clusters which the two solutions of pair p fall into */
5: **if** $m \neq n$ **then**
6: **if** $\forall s_i \in C_m, s_j \in C_n : D(s_i, s_j) < \sigma \wedge D(s_j, s_i) < \sigma$ **then**
7: $C_m \leftarrow C_m \cup C_n$
8: $C_n \leftarrow \emptyset$ /* Merging two clusters if the dominance distance of any two solutions in them is not larger than σ */
9: **end if**
10: **end if**
11: **end for**
12: **return** $\{C_1, C_2, ..., C_{cn}\} : |C_i| > 0$

equal to their difference on one objective. In this case, we have

$$\sigma = \frac{1}{h} \tag{6.3}$$

where h denotes the divisions on one objective. So now our aim is to determine h.

On the other hand, the total number of points over the hyperplane can be calculated by $N = \binom{m-1+h}{m-1}$ (m is the number of objectives), that is

$$\begin{aligned} N &= \binom{m-1+h}{m-1} \\ &= \frac{(h+m-1) \times (h+m-2) \times ... \times (h+1)}{(m-1) \times (m-2) \times ... \times (1)} \end{aligned} \tag{6.4}$$

Since it is hard to accurately determine h from the above expression, we make an approximation by $(h+m-1) \times (h+m-2) \times ... \times (h+1) \approx (h+m/2)^{m-1}$. Then h can be approximately obtained by

$$h \approx \sqrt[m-1]{N(m-1)!} - (m/2) \tag{6.5}$$

Finally, from Eqs. (6.3) and (6.5), we have

$$\sigma \approx \frac{1}{\sqrt[m-1]{N(m-1)!} - (m/2)} \tag{6.6}$$

6.4 Comparison with State of the Art

Table 6.2 summarises the properties of PCI and some popular performance indicators. As can be seen, PCI and the ϵ-indicator perform similarly in terms of many aspects, such as neither specifying a reference set nor requiring parameter setting in the evaluation. The only difference between them shown in the table is that PCI is capable of dealing with more than two approximation sets in a single run. In fact, one significant weakness of the ϵ-indicator to other performance indicators, as indicated before, is its return only being the difference on one particular objective of one particular solution for two approximation sets.

Like IGD^+, PCI only considers the inferiority of solutions in the evaluation. However, one difference is that PCI considers the comparison between a point to a point set (or two point sets), which could be viewed as a more general case of two-points comparison in IGD^+. In addition, since the reference set in PCI consists of the tested approximation sets, the evaluation result of one approximation set depends entirely on its performance difference with other sets. In contrast, in IGD^+ the reference set is specified, and the evaluation result of the approximation sets not only depends on their performance difference but also could be largely affected by the reference set (such as its distribution).

Finally, Table 6.3 gives the evaluation results of PCI and the peer indicators on the approximation sets in Figures 6.1–6.3. As shown, compared with HV, ϵ-indicator and IGD^+, the proposed indicator can accurately reflect the difference of solution sets in terms of uniformity, proximity and diversity, respectively.

Table 6.2: Properties of some performance indicators

Indicators	Proximity	Diversity	Number of sets	Time complexity	Pareto compliant	Reference set specification	Parameter(s)
Coverage [294]	√		binary	quadratic	weakly	no	no
DCI [174]		√	arbitrary	quadratic	no	no	yes
HV [294]	√	√	unary	exponential	yes	no	yes
ϵ-indicator [295]	√	√	unary/binary	quadratic	weakly/yes	no	no
IGD [19]	√	√	unary	quadratic	no	yes	no
IGD^+ [123]	√	√	unary	quadratic	weakly	yes	no
PCI	√	√	arbitrary	quadratic	weakly or yes*	no	no

*PCI is Pareto compliant when two approximation sets are considered, but weakly Pareto compliant when more sets are involved.

Table 6.3: Evaluation results of PCI and the peer indicator (HV, ϵ-indicator, or IGD$^+$) on the approximation set instances in Figures 6.1–6.3. The reference point 1.1 is used in the HV calculation of Figure 6.1's instance. A better result is highlighted in boldface.

Two sets	Peer indicator	PCI
Figure 6.1(a) vs Figure 6.1(b)	0.7448 vs **0.7460** (HV)	**0.0159** vs 0.0170
P vs Q in Figure 6.2	2.5 vs 2.5 (ϵ-indicator)	**0.0000** vs 0.1204
P vs Q in Figure 6.3	0.625 vs 0.625 (IGD$^+$)	**0.0648** vs 0.0926

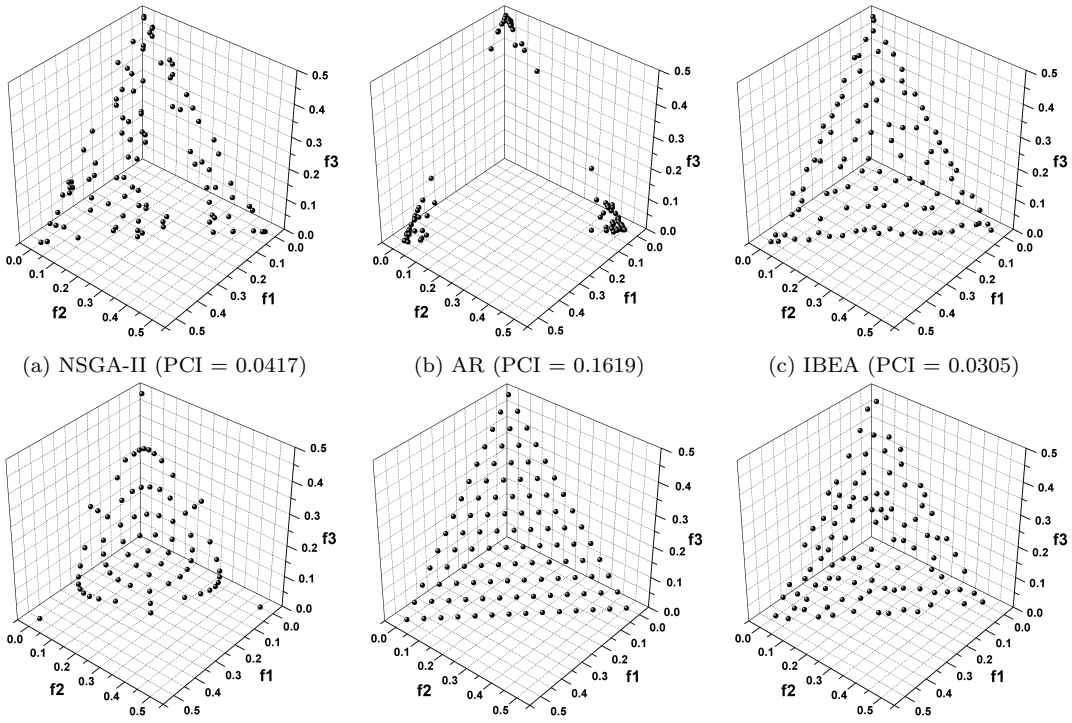
6.5 Experimental Results

In this section, we verify the proposed indicator by assessing Pareto front approximations obtained by six well-established EMO algorithms, NSGA-II [55], AR [45], IBEA [291], MOEA/D-TCH [278], MOEA/D-PBI [278] and SPEA2+SDE (proposed in Chapter 3). These algorithms are representative in terms of both proximity and diversity in dealing with many-objective optimisation problems.

A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n is the number of decision variables) are used. The crossover and mutation operators are simulated binary crossover (SBX) and polynomial mutation with both distribution indexes 20. The population size is set to 100 (or approximately 100 for MOEA/D due to its property [278]) and the termination criterion is $30,000$ evaluations.

To start with, we consider a tri-objective MOP, DTLZ1 [57], whose Pareto front is the positive part of a hyperplane satisfying $f_1 + f_2 + f_3 = 0.5$. Here, we ease the difficulty of the problem by using $g = \sum_{i=m}^{n}(x_i - 0.5)$ in DTLZ1 [57] in order to focus on the diversity verification of the indicator.

Figure 6.5 shows the approximation sets of the six algorithms as well as the corresponding PCI result. As can be seen from the figure, the solutions of MOEA/D-PBI are perfectly distributed over the whole Pareto front, thus having the best evaluation result. Although the solution sets of SPEA2+SDE, IBEA, and NSGA-II cover the triangle, their uniformity is different, which is consistent with the PCI results. The solution set of MOEA/D-TCH is of great regularity, but fails to cover the boundary region of the Pareto front, thus leading to a worse PCI than the above four sets. The



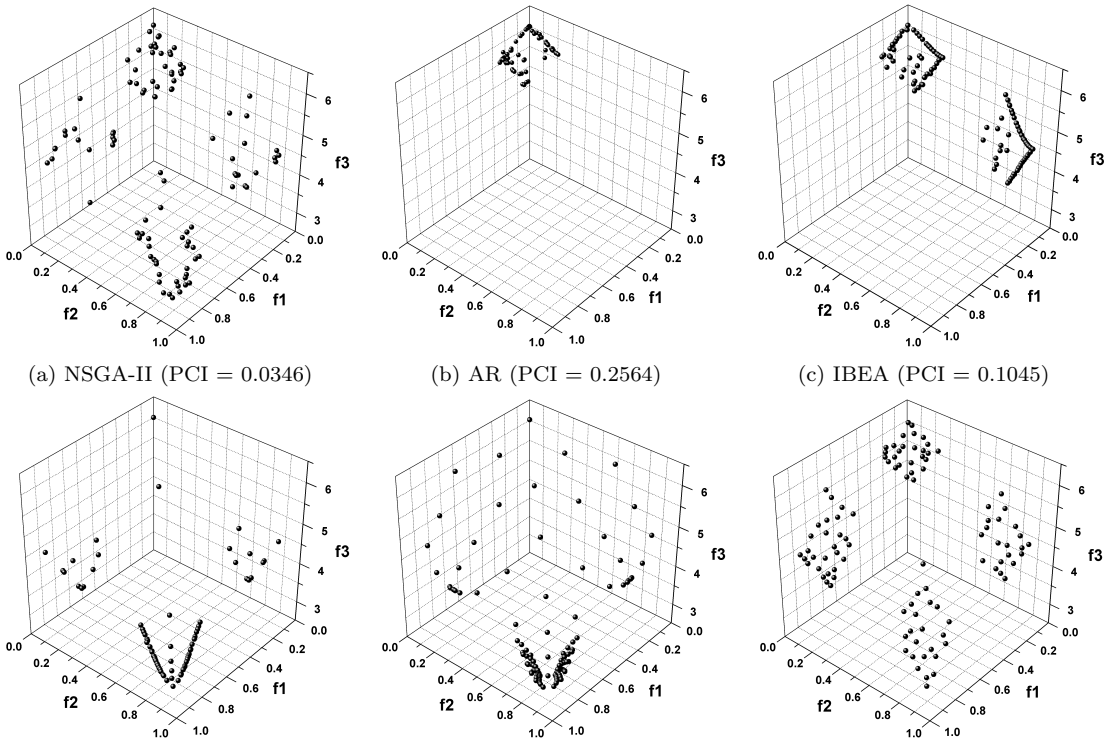
(a) NSGA-II (PCI = 0.0417) (b) AR (PCI = 0.1619) (c) IBEA (PCI = 0.0305)

(d) MOEA/D-TCH (PCI = 0.0434) (e) MOEA/D-PBI (PCI = 0.0270) (f) SPEA2+SDE (PCI = 0.0291)

Figure 6.5: Approximation sets of the six algorithms and their PCI result on the modified tri-objective DTLZ1.

solutions of AR only concentrate around three extreme points and obtain the worst PCI value.

To examine the proposed indicator on MOPs with an irregular Pareto front, we introduce the test problem DTLZ7 [57]. The tri-objective DTLZ7 has a disconnect Pareto front consisting of four regions with both convex and concave shapes. Figure 6.6 gives the approximation sets and the evaluation results. As shown, the set of SPEA2+SDE and NSGA-II is located in the four optimal regions, with a nearly equal number of solutions. The solutions of the two MOEA/D algorithms are mainly distributed in the bottom region of the Pareto front, and IBEA and AR fail to find all the four regions. It is clear that the PCI results confirm the observations. The solution set with a lower PCI value means that it performs better regarding the combined performance in finding multiple Pareto optimal regions and in maintaining solutions' uniformity and diversity in each region.



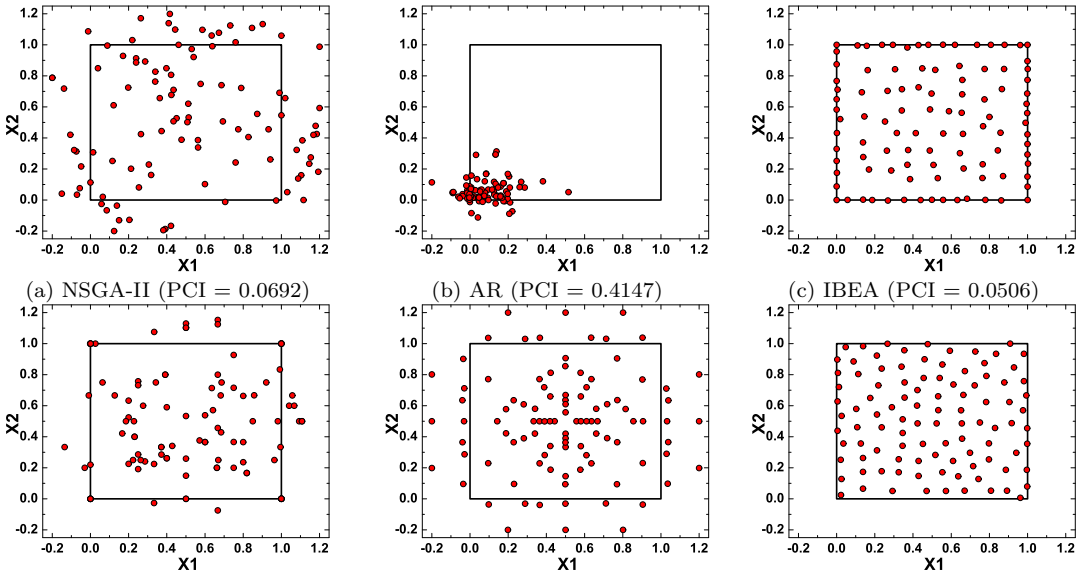
(a) NSGA-II (PCI = 0.0346) (b) AR (PCI = 0.2564) (c) IBEA (PCI = 0.1045)

(d) MOEA/D-TCH (PCI = 0.0638) (e) MOEA/D-PBI (PCI = 0.0483) (f) SPEA2+SDE (PCI = 0.0163)

Figure 6.6: Approximation sets of the six algorithms and their PCI result on the tri-objective DTLZ7.

Next, we consider a four-objective MOP, Rectangle problem[5] [176]. The Rectangle problem has two features: 1) its Pareto optimal solutions lie in a rectangle in the two-variable decision space and 2) they are similar (in the sense of Euclidean geometry) to their images in the four-dimensional objective space. These make the visual examination of Pareto front approximations feasible by observing their behaviour in the decision space. Figure 6.7 shows the approximation sets and the evaluation results on an instance of the Rectangle problem where the search range and Pareto optimal range of x_1 and x_2 are $[-0.2, 1.2]$ and $[0, 1]$, respectively. As can be seen, the performance of these approximation sets is consistent with the PCI results. SPEA2+SDE and IBEA have a set of well-converged and well-distributed solutions, thus having better PCI results than the other four algorithms. IBEA obtains a slightly worse PCI value than

[5]This test problem is also part of our work and will be presented in Chapter 7



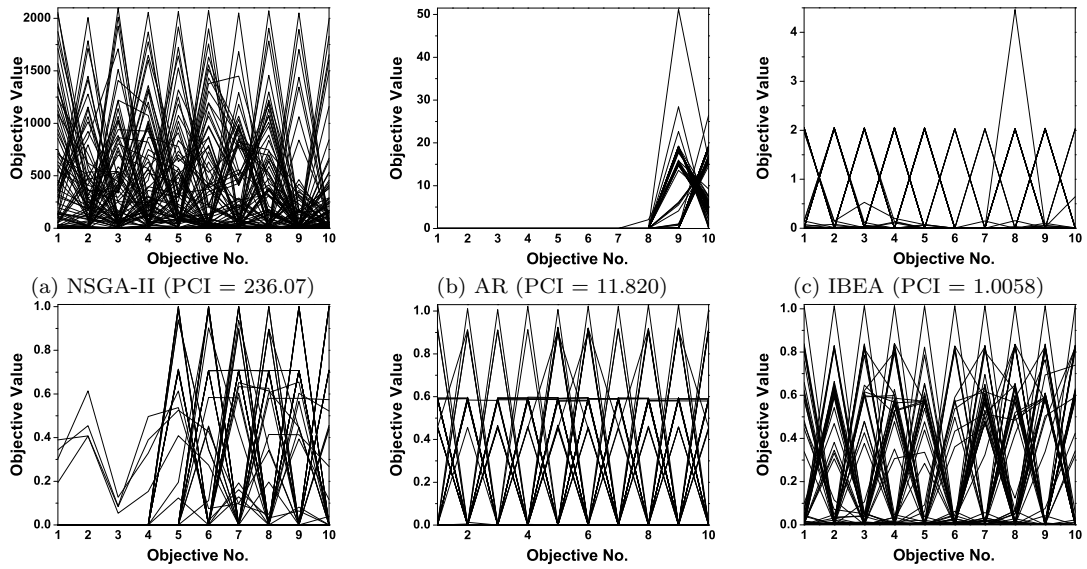
(a) NSGA-II (PCI = 0.0692) (b) AR (PCI = 0.4147) (c) IBEA (PCI = 0.0506)

(d) MOEA/D-TCH (PCI = 0.0666) (e) MOEA/D-PBI (PCI = 0.0798) (f) SPEA2+SDE (PCI = 0.0498)

Figure 6.7: Approximation sets of the six algorithms in the two-variable decision space and their PCI result on the four-objective Rectangle problem, where the Pareto optimal solutions in the decision space are similar to their images in the objective space in the sense of Euclidean geometry.

SPEA2+SDE, with more solutions being located onto the boundary of the optimal region. MOEA/D-TCH, NSGA-II and MOEA/D-PBI perform similarly in terms of proximity and diversity. Among them, MOEA/D-TCH has fewer solutions out of the optimal rectangle and MOEA/D-PBI has many solutions concentrated into the center of the rectangle. This leads to the difference of their PCI results. All solutions of AR are located around the lower left corner of the rectangle and thus have the worst evaluation result.

Finally, the 10-objective DTLZ3 is used to verify PCI in assessing approximation sets for high-dimensional challenging MOPs. DTLZ3 has a vast number of local optimal fronts and a global one satisfying $f_1^2 + f_2^2 + ... + f_m^2 = 1$ in the range $f_1, f_2, ..., f_m \in [0, 1]$. Figure 6.8 shows the approximation sets by parallel coordinates as well as the corresponding evaluation results. As shown, the solutions of only three algorithms SPEA2+SDE, MOEA/D-PBI, and MOEA/D-TCH can converge into the optimal front. SPEA2+SDE and MOEA/D-PBI achieve a good balance between prox-



(a) NSGA-II (PCI = 236.07) (b) AR (PCI = 11.820) (c) IBEA (PCI = 1.0058)

(d) MOEA/D-TCH (PCI = 0.2218) (e) MOEA/D-PBI (PCI = 0.0936) (f) SPEA2+SDE (PCI = 0.0845)

Figure 6.8: Parallel coordinate plot of approximation sets of the six algorithms and their PCI result on the ten-objective DTLZ3.

imity and diversity, while MOEA/D-TCH fails to cover the whole Pareto front. Almost all solutions of IBEA converge into the boundary of a local optimum ($f_1^2 + ... + f_{10}^2 = 2$), and most of AR's solutions concentrate around two boundary points of a local optimum ($f_1^2 + ... + f_{10}^2 = 20$). NSGA-II completely fails to approach the optimal front, with the upper boundary of its solutions exceeding 2,000 on each objective. The PCI results reflect the performance of the approximation sets. A set with a lower PCI value means that it performs better with respect to the tradeoff between proximity and diversity.

6.6 Summary

In this chapter, we present a performance comparison indicator (PCI) to compare Pareto front approximations of population-based search algorithms. PCI evaluates the quality of approximation sets to a reference set which is constructed by themselves. The points in the reference set are clustered according to their distribution, and PCI estimates the minimum moves of one solution (or a set of solutions) in one approximation set to weakly dominate these clusters. The proposed indicator can be practical

in many-objective optimisation, given its characteristics such as a combined evaluation of proximity and diversity, (weak) compliance with Pareto dominance, applicability for any number of approximation sets, no need of a specified reference set, and no requirement of parameter setting in the assessment.

A weakness of the proposed indicator is that we approximately measure the dominance distance of a point set to another, which can affect the accuracy of the indicator. Now, the question is whether there exists an efficient method to accurately measure the dominance distance of two point sets. If the answer is yes, we could directly compare two Pareto front approximations by their dominance distance to each other, rather than dividing solutions into many clusters.

Chapter 7

A Test Problem for Visual Investigation

In this chapter, we present a test problem, called the Rectangle problem, to aid the visual investigation of multi-objective search. The Rectangle problem has two key features: 1) the Pareto optimal solutions lie in a rectangle in a two-dimensional decision space, and 2) the Pareto optimal solutions are similar (in the sense of Euclidean geometry) to their images in a four-dimensional objective space. In this case, it is capable of visually examining the behaviour of objective vectors in terms of both proximity and diversity, by observing their closeness to the optimal rectangle and their distribution in the rectangle, respectively, in the decision space.

This chapter is organised as follows. Section 7.1 is devoted to the introduction of this work. Section 7.2 details the proposed test problem. Section 7.3 presents experimental results of visual investigation of 15 algorithms on three problem instances. Finally, Section 7.4 draws some conclusions.

7.1 Introduction

An inherent problem in multi-objective optimisation is that a direct observation of solution vectors with four or more objectives is infeasible, which brings a difficulty for a visual investigation of EMO algorithms. In contrast to two- or three-objective

problems where it is straightforward to track objective vectors during the evolutionary process, for problems with four or more objectives we cannot visually monitor how a set of objective vectors evolve nor visually understand their distribution in the space and their proximity to the Pareto front.

A natural way of dealing with this issue is to map high-dimensional objective vectors to a two- or three-dimensional space. Key concerns under such mapping include the maintenance of the Pareto dominance relation between vectors and the reflection of their location information in the population. Inevitable information loss associated with the dimension reduction will influence the observation and understanding of these vectors.

Recently, some researchers ease this visualisation challenge from another perspective. They constructed a particular class of test problems to help the visual investigation of multi-objective search [156, 226, 119, 239]. Specifically, Köppen and Yoshida [156] presented a class of many-objective test problems whose Pareto optimal set is in a regular polygon in a two-dimensional decision space. This allows easy visualisation and examination of the proximity of the obtained solutions to the optimal region and their distribution in the decision space. Later, Ishibuchi et al. [119, 130] extended and generalised this class of problems (called distance minimisation problems), introducing multiple Pareto optimal polygons with same [119] or different shapes [115] as well as making decision variables' dimensionality scalable [130]. Overall, these problems provide a good alternative to help understand the behaviour of multi-objective evolutionary search, and have been used to compare many-objective algorithms in recent studies [241, 175, 180].

However, one weakness of such a class of test problems is that from the decision space aspect it fails to exactly reflect the behaviour and performance of objective vectors, i.e., their proximity and distribution with respect to the Pareto front. Even if a set of objective vectors are distributed perfectly over the optimal front, we cannot know this fact via the observation of the corresponding decision variables in the polygon(s).

In this chapter, we construct a four-objective test problem with its Pareto optimal region being a rectangle in a two-dimensional decision space, called the Rectangle

problem. The key feature in this problem is that the Pareto optimal solutions in the decision space and their images in the objective space are similar (in the sense of Euclidean geometry). In other words, the ratio of the distance between any two Pareto optimal solutions to the distance between their corresponding objective vectors in the Pareto front is a constant. This way, we can easily understand the behaviour and performance of the objective vector set (e.g., its uniformity and coverage over the Pareto front) by observing the solution set in the two-dimensional decision space.

Using three instances of the proposed problem, we investigate the behaviour of 15 EMO algorithms, including well-known multi-objective algorithms and recently-developed many-objective ones. Interesting observations indicate that the Rectangle problem not only is a good tool to help visually understand the behaviour of multi-objective search in a high-dimensional objective space but also can be used as a challenging benchmark function to test algorithms' ability in balancing the proximity and diversity of solutions.

7.2 The Proposed Test Problem

The distance minimisation problems, proposed by Köppen and Yoshida [156] and generalised by Ishibuchi et al. [119], are a class of many-objective optimisation problems that minimise the Euclidean distance from a solution to a given set of points in the two- or three- dimensional Euclidean space, where the distance to any of these points is treated as an independent objective. Figure 7.1 gives a four-objective example of the distance minimisation problem with a set of points **A**, **B**, **C**, and **D**.

A significant feature of the distance minimisation problems is that their Pareto optimal region is a convex polygon determined by the given point set [156]. This allows a clear observation of whether the considered solution set has converged into the Pareto optimal region or not. However, a weakness of such problems is that they are unavailable for the distribution investigation of a solution set in the objective space. There is no explicit distribution relation between decision variables and their corresponding objective images in the problems.

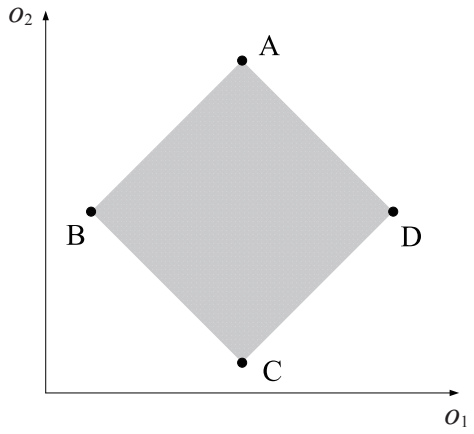


Figure 7.1: An illustration of a four-objective distance minimisation problem whose Pareto optimal region is determined by the four points.

Motivated by the above, we construct a test problem whose Pareto optimal solutions lie in a rectangle in the decision space and more importantly are similar (in the sense of Euclidean geometry) to their images in the objective space. Unlike the distance minimisation problems which consider the distance to a set of points, the proposed Rectangle problem takes into account the distance to a set of lines parallel to the coordinate axes. Figure 7.2 gives an example of the Rectangle problem where the Pareto optimal solutions are in the region enclosed by four lines **A**, **B**, **C**, and **D** (including the boundary).

Formally, the Rectangle problem minimises the Euclidean distance from a solution, $\mathbf{x} = (x_1, x_2)$, to four lines parallel to the coordinate axes ($x_1 = a_1$, $x_1 = a_2$, $x_2 = b_1$, and $x_2 = b_1$):

$$\min \begin{array}{l} f_1(\mathbf{x}) = |x_1 - a_1| \\ f_2(\mathbf{x}) = |x_1 - a_2| \\ f_3(\mathbf{x}) = |x_2 - b_1| \\ f_4(\mathbf{x}) = |x_2 - b_2| \end{array} \tag{7.1}$$

Next, we explain the geometric similarity of the Rectangle problem between the Pareto optimal solutions and their images in the objective space. Let $\mathbf{x}^1 = (x_1^1, x_2^1)$ and $\mathbf{x}^2 = (x_1^2, x_2^2)$ be two Pareto optimal solutions for the problem given in Eq. (7.1)

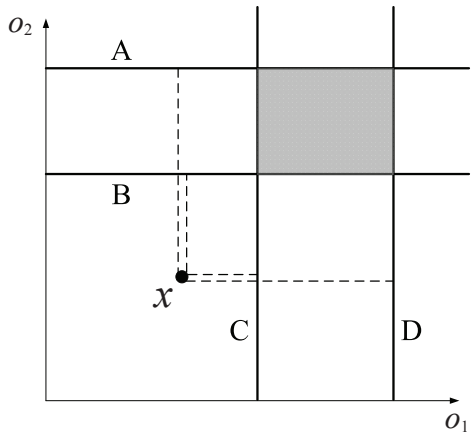


Figure 7.2: An illustration of a Rectangle problem whose Pareto optimal region is determined by the four lines.

(here, without loss of generality, assuming $a_1 < a_2$ and $b_1 < b_2$). Then, their Euclidean distance in the decision space is as follows:

$$D(\mathbf{x}^1, \mathbf{x}^2) = \left[(x_1^1 - x_1^2)^2 + (x_2^1 - x_2^2)^2\right]^{0.5} \tag{7.2}$$

Also, the distance of their images in the objective space is

$$\begin{aligned} & D\big(f(\mathbf{x}^1), f(\mathbf{x}^2)\big) \\ = & \left[\big(f_1(\mathbf{x}^1) - f_1(\mathbf{x}^2)\big)^2 + \big(f_2(\mathbf{x}^1) - f_2(\mathbf{x}^2)\big)^2 + \right. \\ & \left. \big(f_3(\mathbf{x}^1) - f_3(\mathbf{x}^2)\big)^2 + \big(f_4(\mathbf{x}^1) - f_4(\mathbf{x}^2)\big)^2\right]^{0.5} \\ = & \left[(|x_1^1 - a_1| - |x_1^2 - a_1|)^2 + (|x_1^1 - a_2| - |x_1^2 - a_2|)^2 + \right. \\ & \left. (|x_2^1 - b_1| - |x_2^2 - b_1|)^2 + (|x_2^1 - b_2| - |x_2^2 - b_2|)^2\right]^{0.5} \end{aligned} \tag{7.3}$$

Since $\mathbf{x}^1$ and $\mathbf{x}^2$ are two Pareto optimal solutions of the problem, it holds that $a_1 \leq$

$x_1^1, x_1^2 \leq a_2$ and $b_1 \leq x_2^1, x_2^2 \leq b_2$. So Eq. (7.3) can be further expressed as:

$$\begin{aligned} & D\big(f(\mathbf{x}^1), f(\mathbf{x}^2)\big) \\ &= \big[(x_1^1 - x_1^2)^2 + (x_1^1 - x_1^2)^2 + (x_2^1 - x_2^2)^2 + (x_2^1 - x_2^2)^2\big]^{0.5} \\ &= \sqrt{2}\big[(x_1^1 - x_1^2)^2 + (x_2^1 - x_2^2)^2\big]^{0.5} \\ &= \sqrt{2}D(\mathbf{x}^1, \mathbf{x}^2) \end{aligned} \tag{7.4}$$

The above equation indicates that the ratio of the distance between any two Pareto optimal solutions to the distance between their corresponding objective vectors is a constant. As such, it is easy to understand the distribution of the objective vectors in a Pareto front approximation by observing their position and crowding degree in the rectangle in the two-dimensional decision space.

Note that this two-dimensional problem with respect to decision variables can be extended to the three-dimensional scenario. In this way, the proposed problem will minimise the Euclidean distance from a solution (i.e., $\mathbf{x} = (x_1, x_2, x_3)$) to six lines parallel to the three coordinate axes, and the Pareto optimal region will become a cuboid enclosed by these six lines. In addition, it is necessary to point out that unlike in the distance minimisation problems where the objective dimensionality can be set freely, in the Rectangle problem the number of objectives (i.e., the considered lines) is determined by the number of decision variables (two lines corresponding to one coordinate axis). It might not be easy to add new lines while keeping the dimensionality of decision space unchanged, because the geometric similarity between the Pareto optimal solutions and their objective images will be violated when one coordinate axis corresponds to more than two lines.

7.3 Experimental Results

In this section, we investigate the behaviour of EMO algorithms on the Rectangle problem. In all, 15 algorithms are investigated, including those well-known and developed specially for many-objective optimisation. They are NSGA-II [55], SPEA2 [292], MSOPS [106], IBEA [291], ϵ-MOEA, SMS-EMOA [17], MOEA/D [278], AR

Table 7.1: The parameter setting and the source of the tested algorithms

Algorithm	Parameter(s)	Source
NSGA-II [55]		http://www.iitk.ac.in/kangal
SPEA2 [292]		http://www.tik.ee.ethz.ch/pisa
MSOPS [106]	weight vectors 200	http://code.evanhughes.org/
IBEA [291]	$\kappa = 0.05$	http://www.tik.ee.ethz.ch/pisa
ϵ-MOEA [54]	$\epsilon = 0.85$	http://www.iitk.ac.in/kangal
SMS-EMOA [17]		http://jmetal.sourceforge.net/index.html
MOEA/D-TCH [278]	neighborhood size 10%	http://dces.essex.ac.uk/staff/qzhang/
MOEA/D-PBI [278]	neighborhood size 10%, penalty 2.0	http://dces.essex.ac.uk/staff/qzhang/
AR [15]		written by ourselves
AR+Grid [184]	grid division 30	http://www.brunel.ac.uk/~cspgmml1/
HypE [10]	sampling point $10,000$	http://www.tik.ee.ethz.ch/pisa
DMO [1]		written by ourselves
GrEA [271]	grid division 25	http://www.brunel.ac.uk/~cspgmml1/
FD-NSGA-II [96]	$\sigma = 0.5$	provided by its authors
SPEA2+SDE [181]		http://www.brunel.ac.uk/~cspgmml1/

[15], AR+Grid [184], HypE [10], DMO [1], GrEA [271] (also Chapter 4), FD-NSGA-II [96], and SPEA2+SDE [181] (also Chapter 3). Readers seeking more details on these algorithms may refer to their original literature.

A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n denotes the number of decision variables) were used. The operators for crossover and mutation are simulated binary crossover (SBX) and polynomial mutation with both distribution indexes 20. The population size was set to 120 (also the archive set maintained with the same if required) and the termination criterion of a run was 30,000 evaluations (i.e., 250 generations) for all the algorithms. In ϵ-MOEA, the size of the archive set is determined by the ϵ value. For a fair comparison, we set ϵ so that the archive set is approximately of the same size as that of the other algorithms. Table 7.1 summarizes parameter settings as well as the source of all the algorithms. The setting of these parameters in our experimental studies either follows the suggestion in their original papers or has been found to make the algorithm perform well on the tested problem.

To investigate the proximity and diversity of the algorithms, we introduce three instances of the Rectangle problem. These instances have same objective lines ($x_1 = 0$, $x_1 = 100$, $x_2 = 0$, and $x_2 = 100$), and the only difference lies in the range of their search (decision) space, thus providing different challenges for an algorithm to balance

proximity and diversity.

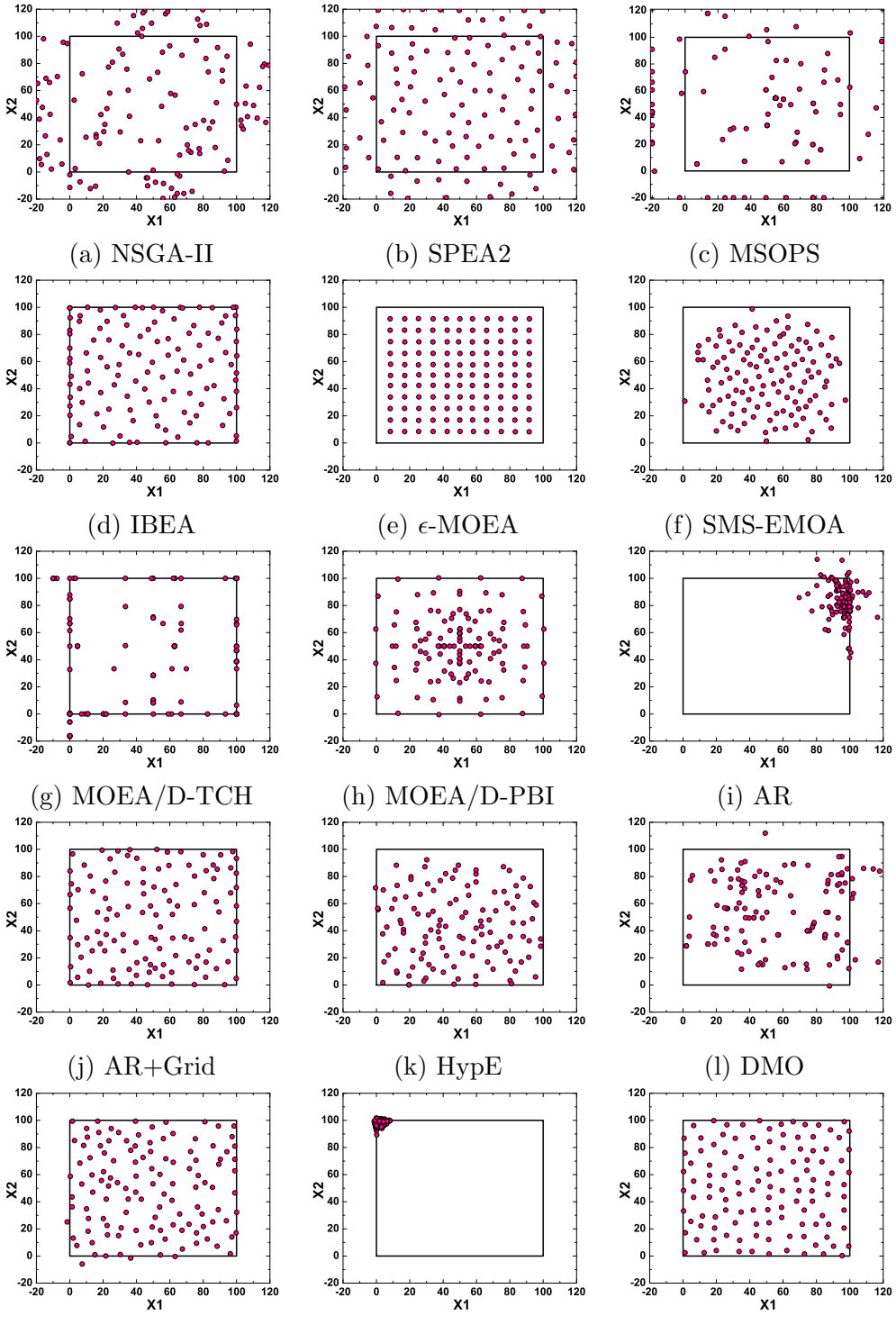
7.3.1 Instance I

Figure 7.3 shows the final solution sets obtained by one typical run of the 15 algorithms on a problem instance where the search range of both x_1 and x_2 is $[-20, 120]$. From different behaviours of their solutions in the figure, these algorithms can be divided into four groups.

The first group corresponds to the algorithms which fail to converge, including NSGA-II, SPEA2, and MSOPS. Among them, SPEA2 performs better than the other two algorithms in terms of diversity, although its solutions seem to cover the whole search space rather than the optimal region. Two algorithms, AR and FD-NSGA-II, belong to the second group where the obtained solutions concentrate in a small part of the Pareto optimal region. The algorithms in the third group struggle to maintain uniformity although most of their solutions can converge into the optimal region. SMS-EMOA, MOEA/D-TCH, MOEA/D-PBI, HypE, and DMO fall into this group: their solutions overcrowded in some regions of the rectangle, thus leading to vacancy in other ones. It is worth noting that the solutions of SMS-EMOA here concentrate on the middle part of the Pareto optimal region. This observation is interesting, given that SMS-EMOA has been reported to perform well in maintaining solutions' extensity [17, 256].

The last group includes the remaining algorithms (i.e., IBEA, ϵ-MOEA, AR+Grid, GrEA and SPEA2+SDE) which perform well in terms of proximity and diversity. More specifically, the solutions obtained by ϵ-MOEA tend to be perfectly uniform, but fail to cover the boundary of the optimal region. Although the solutions of IBEA, AR+Grid, and GrEA can reach the boundary of the optimal region, they are not so uniform as those of ϵ-MOEA and SPEA2+SDE. SPEA2+SDE appears to be the only algorithm with excellent performance in terms of both extensity and uniformity, and its solutions are distributed uniformly over the whole Pareto optimal region.

In addition, it is worth mentioning that among different implementations of MOEA/D regarding decomposition functions, MOEA/D-PBI with the penalty parameter value 2



(a) NSGA-II (b) SPEA2 (c) MSOPS

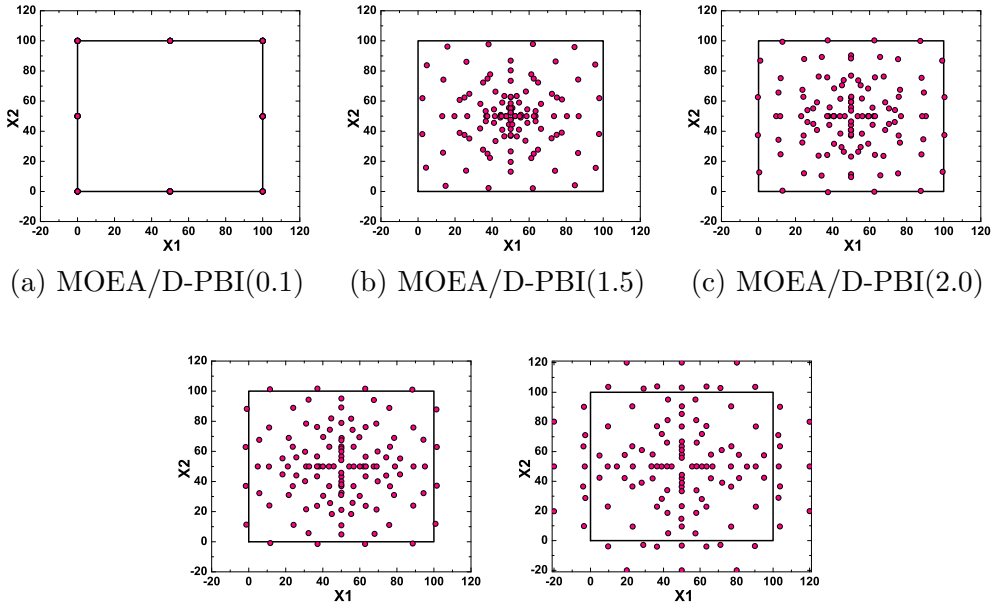
(d) IBEA (e) ϵ-MOEA (f) SMS-EMOA

(g) MOEA/D-TCH (h) MOEA/D-PBI (i) AR

(j) AR+Grid (k) HypE (l) DMO

(m) GrEA (n) FD-NSGA-II (o) SPEA2+SDE

Figure 7.3: The final solution set of the 15 algorithms on the Rectangle problem where $x_1, x_2 \in [-20, 120]$.



(a) MOEA/D-PBI(0.1) (b) MOEA/D-PBI(1.5) (c) MOEA/D-PBI(2.0)

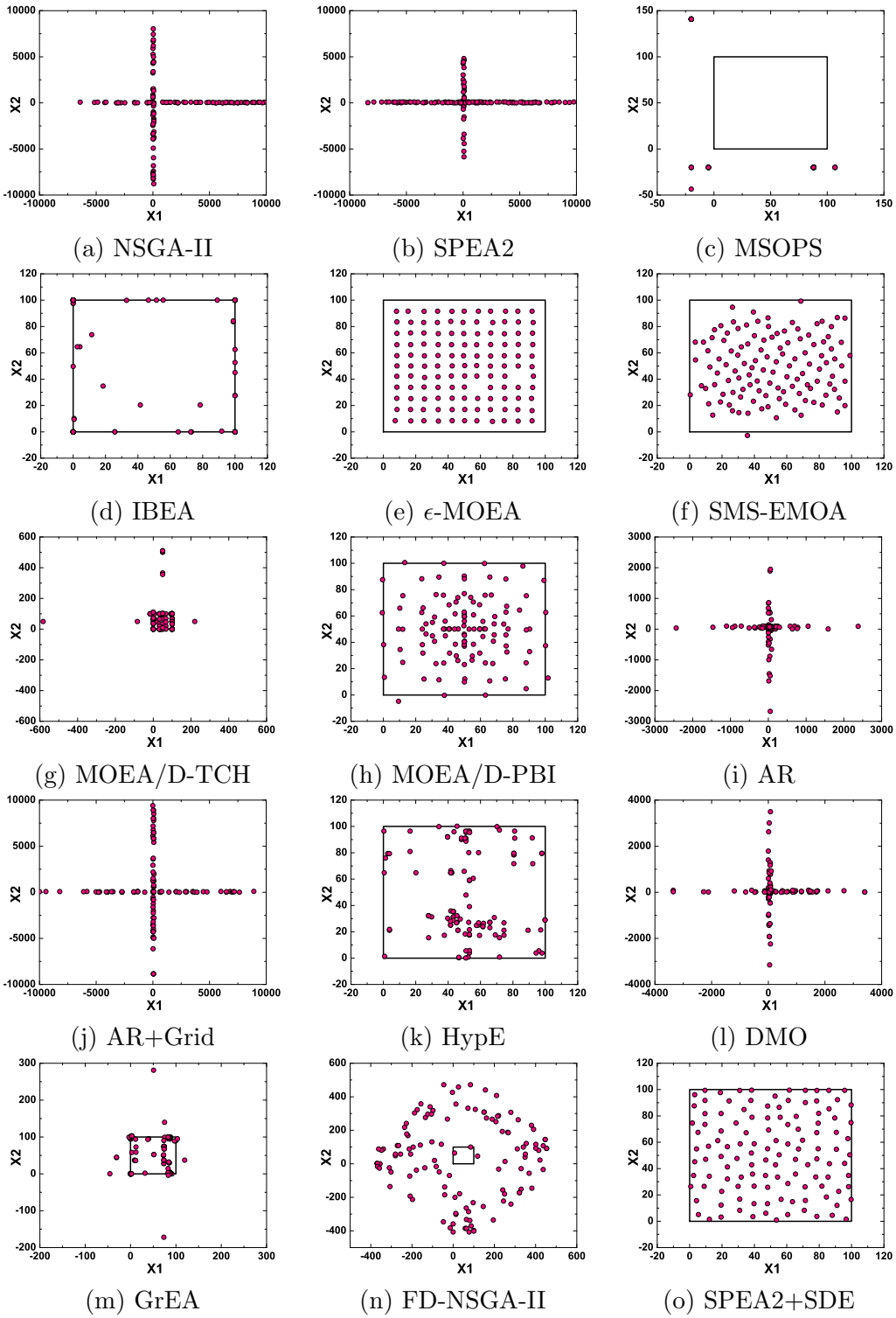
(d) MOEA/D-PBI(2.5) (e) MOEA/D-PBI(5.0)

Figure 7.4: The final solution set of the five implementations of MOEA/D-PBI with different penalty parameter values on the Rectangle problem where $x_1, x_2 \in [-20, 120]$. The number in the bracket denotes the penalty parameter value of the algorithm.

(i.e., the setting considered here) performs best on the rectangle problem; this is not the case for the distance minimisation problem where MOEA/D-TCH and MOEA/D-PBI with the penalty parameter value 0.1 (denoted as MOEA/D-PBI(0.1)) have been found to work well [117, 181]. Figure 7.4 shows the result of five implementations of MOEA/D-PBI with different penalty parameter values on the tested instance. Clearly, MOEA/D-PBI(0.1) fails to maintain diversity, while MOEA/D-PBI(5.0) struggles to have all of its solutions into the optimal region. Despite having similar distribution with MOEA/D-PBI(1.5) and MOEA/D-PBI(2.5), MOEA/D-PBI(2.0) seems to achieve a better balance between proximity and diversity, some of its solutions located exactly in the boundary of the rectangle.

7.3.2 Instance II

The Rectangle problem instance considered in this section greatly enlarges the search space of Instance I, with $x_1, x_2 \in [-10000, 10000]$. Figure 7.5 shows the final solution



(a) NSGA-II (b) SPEA2 (c) MSOPS

(d) IBEA (e) ϵ-MOEA (f) SMS-EMOA

(g) MOEA/D-TCH (h) MOEA/D-PBI (i) AR

(j) AR+Grid (k) HypE (l) DMO

(m) GrEA (n) FD-NSGA-II (o) SPEA2+SDE

Figure 7.5: The final solution set of the 15 algorithms on the Rectangle problem where $x_1, x_2 \in [-10000, 10000]$.

sets obtained by one typical run of the 15 algorithms on this instance.

It is clear from the figure that most of the algorithms face big challenges in balancing proximity and diversity on this problem instance. The solution sets obtained by NSGA-II, SPEA2, AR, AR+Grid, DMO, and FD-NSGA-II fail to approach the Pareto optimal region. The first five sets are distributed in the form of a cross and the last one is located in a rhombic region. All solutions of MSOPS overlap in six points near the rectangle. MOEA/D-TCH and GrEA perform similarly – most of their solutions can converge into the Pareto optimal region, but there still exist several solutions far away from the optimal rectangle. Although all the solutions obtained by IBEA and HypE are the Pareto optimal solutions, the two algorithms struggle to maintain uniformity, leading their solutions to concentrate (or even coincide) in some areas of the rectangle.

The remaining algorithms, ϵ-MOEA, SMS-EMOA, MOEA/D-PBI and SPEA2+SDE, perform significantly better than the previous ones. The solutions of ϵ-MOEA have almost perfect uniformity. Despite some downsides in terms of extensity or uniformity, the solution set obtained by SMS-EMOA and MOEA/D-PBI largely covers the whole optimal region. SPEA2+SDE, like the case on the problem Instance I, achieves the best performance in balancing solutions' uniformity and extensity.

Contrast the results on Instance II with those on Instance I – only the four algorithms (i.e., ϵ-MOEA, SMS-EMOA, MOEA/D-PBI and SPEA2+SDE) perform similarly; some algorithms' solution set, like NSGA-II's and SPEA2's, is far away from the optimal region and distributed crosswise. This is not the case for the distance minimisation problem where the solution set obtained by Pareto-based algorithms can easily approach the Pareto optimal region even when the number of objectives reaches 10 [119]. Figure 7.6 gives an illustration to explain why this happens. Let $\mathbf{x}^1$ and $\mathbf{x}^2$ be two solutions for the Rectangle problem in the figure. $\mathbf{x}^1$ is located in the middle of the two objective lines parallel to coordinate axis o_2, and $\mathbf{x}^2$ in the right upper area to the four objective lines. The region that Pareto-dominates $\mathbf{x}^1$ is a line segment, far smaller than that dominating $\mathbf{x}^2$, although $\mathbf{x}^2$ is closer to the optimal region than $\mathbf{x}^1$. In fact, any solution located between two parallel objective lines (except the Pareto optimal solutions) is dominated only by a line segment parallel to the two objective

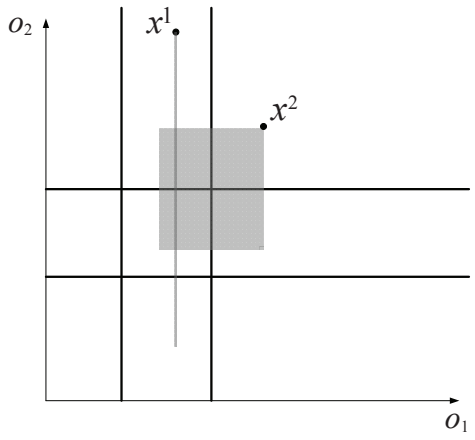


Figure 7.6: An illustration of the difficulty for algorithms to converge on the Rectangle problem. The shadows are the regions that dominate $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively.

lines, given that any improvement of the solution's distance to the one objective line will lead to the degradation to the other. This characteristic of the Rectangle problem (i.e., some non-Pareto optimal solutions dominated by only a linear region) will bring a great challenge for algorithms which use Pareto dominance as the sole selection criterion in terms of proximity, usually leading their solutions to be distributed crisscross in the space.

In addition, it is necessary to mention that on Instance II the optimal setting of the algorithms' parameter(s), if existing, is likely to be different from that on Instance I. The characteristic of the Rectangle problem explained above can make the optimal setting of the parameter(s) vary for the search space with different ranges. For instance, a significantly large grid division of GrEA (say 500) can make the algorithm's solutions converge into the optimal rectangle as well as having good diversity on Instance II. Similar cases occur for the algorithms IBEA, AR+Grid, and FD-NSGA-II.

7.3.3 Instance III

From the result comparison on Instance I and Instance II, the four algorithms, ϵ-MOEA, SMS-EMOA, MOEA/D-PBI, and SPEA2+SDE, have been found to perform steadily with the change of the search space. In this section, we significantly enlarge the search

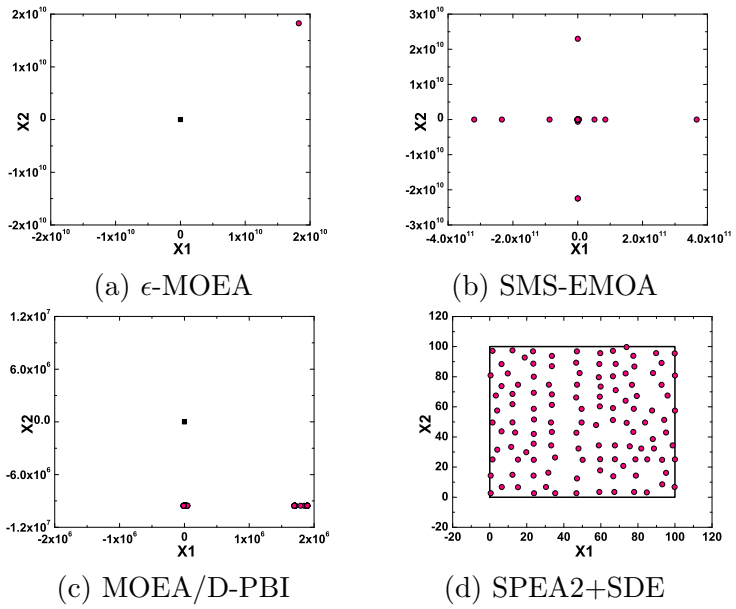


Figure 7.7: The final solution set of the four algorithms on the Rectangle problem where $x_1, x_2 \in [-10^{12}, 10^{12}]$.

range of solutions in order to further test the algorithms' ability of leading solutions to converge towards the Pareto optimal region when working in a huge space. Figure 7.7 shows the final solution sets obtained by one typical run of the four algorithms on the problem instance with $x_1, x_2 \in [-10^{12}, 10^{12}]$.

Clearly, only SPEA2+SDE works well on this instance, with its solutions located in the rectangle as well as having good coverage. The archive set of ϵ-MOEA has only one individual far from the optimal region. In fact, no matter how to set the ϵ value of the algorithm, there is only one solution left in the final archive set when the problem's search space becomes huge. Likewise, SMS-EMOA and MOEA/D-PBI fail to lead their solution set to approach the optimal region. The solutions of SMS-EMOA have a crisscross distribution and the solutions of MOEA/D-PBI gather into two clusters parallel to the horizontal axis.

7.4 Summary

Visual investigation of evolutionary search in a high-dimensional space is an important issue in the EMO area, which can help understand the behaviour of the existing EMO algorithms, facilitate their modification, and further develop new algorithms for many-objective optimisation problems. Unlike the existing studies which mainly focus on the mapping of high-dimensional objective vectors to two- or three-dimensional ones for visualisation, this chapter constructs a test function, called the Rectangle problem, where the Pareto optimal solutions in the two-variable decision space have similar distribution to their images in the four-dimensional objective space. In this case, it is feasible to visually investigate high-dimensional objective vectors of the problem by observing their behaviour in the decision space.

Fifteen EMO algorithms have been investigated on three instances of the proposed problem with varying range of search spaces which present different challenges for an algorithm to converge. Different behaviours of the tested algorithms have been demonstrated. The Pareto-based algorithms (i.e., NSGA-II and SPEA2) fail to guide their solutions evolving towards the Pareto optimal region even if the optimal region accounts for a large proportion of the whole search space. IBEA, AR+Grid, and GrEA can achieve a good balance between proximity and diversity on Instance I, but struggle when the search space becomes larger. Although ϵ-MOEA, SMS-EMOA, and MOEA/D-PBI work well on Instances I and II, their solutions fail to approach the optimal rectangle when a huge problem's search space is introduced. SPEA2+SDE is the only algorithm with good performance on all three test instances, with its solutions distributed uniformly over the whole Pareto optimal region all along.

A weakness of the proposed problem is its fixed dimensionality (four) in the objective space. Despite this, the Rectangle problem poses big challenges to many-objective algorithms. This is different from the case that on most existing four-objective problems (e.g., the DTLZ problem suite [57]), the algorithms designed specifically for many-objective optimisation (or even only based on Pareto selection criterion) perform fairly well [256]. This indicates that the Rectangle problem could be used as an effective

benchmark function to challenge the search ability of optimisation algorithms in the area.

Chapter 8

Conclusion

In this chapter, we present a global view on the contributions of this thesis. First, we summarise the research carried out within each chapter in Section 8.1. In so doing, we explain how we deal with the challenges of many-objective optimisation that were stated at the beginning of the thesis. Then, we outline some directions for further research in Section 8.2.

8.1 Summary of Results

Many-objective optimisation poses great challenges to evolutionary algorithms. The ineffectiveness of the Pareto dominance relation leads to the underperformance of traditional Pareto-based algorithms. The aggravation of the conflict between proximity and diversity, along with increasing time or space requirement as well as parameter sensitivity, has become key barriers to the design of new algorithms. The infeasibility of solutions' direct observation can result in serious difficulty of algorithms' performance investigation and assessment.

In this thesis, we have proposed a series of approaches to these challenges, aiming to make evolutionary algorithms as effective in many-objective optimisation as in low-dimensional multi-objective (i.e. two- or three-objective) optimisation. Specifically, a shift-based density estimation strategy (Chapter 3) has been proposed to make traditional Pareto-based algorithms suitable for many-objective problems; a grid-based

algorithm (Chapter 4) and a bi-goal evolution framework (Chapter 5) have been developed, as an independent algorithm and a general framework, respectively, to deal with many-objective problems; a performance indicator (Chapter 6) and a test problem (Chapter 7) have also been introduced for algorithm comparison and visualisation investigation, respectively, in the high-dimensional objective space. In the following, we provide a summary of the research results for each of these chapters.

Chapter 3 presented a general enhancement of Pareto-based algorithms to make them suitable for many-objective problems. Unlike most of the current work which relaxes the Pareto dominance relation to make more individuals comparable, the proposed approach acts on the density estimation operation in Pareto-based algorithms (thus no need of the parameter(s) of determining relaxation degree). The proposed shift-based density estimation (SDE) was applied to three well-known Pareto-based algorithms, NSGA-II, SPEA2 and PESA-II. It has been observed that after the implementation of SDE all three algorithms achieve a performance improvement. A further comparative study among NSGA-II+SDE, SPEA2+SDE and PESA-II+SDE has revealed that SDE is well suited for the density estimator which can accurately measure individuals' density in the population. Moreover, a comprehensive comparison with five state-of-the-art algorithms which tackle many-objective problems from different perspectives has shown that SPEA2+SDE is very promising in providing a good balance between proximity and diversity. Overall, the proposed approach is highly practical in many-objective optimisation, in view of its competitiveness against well-established indicator-based and decomposition-based approaches, applicability for any Pareto-based algorithm, simple implementation, negligible additional computational cost, and no need of additional parameters.

Chapter 4 presented a grid-based evolutionary algorithm (GrEA), which exploits the potential of the grid to address many-objective optimisation problems. In GrEA, three grid-based criteria were introduced to guide the search towards the optimal front and a fitness adjustment strategy was developed to maintain an extensive and uniform distribution among individuals. In particular, to measure the crowding of individuals, GrEA considers the distribution of their neighbours in a set of hyperboxes whose

size increases with the number of objectives (rather than a single hyperbox, typically used in existing grid-based approaches), thus providing a clear distinction of individuals' crowding degree. Systematic experiments were carried out to make an extensive comparison of GrEA with five well-established algorithms on 52 test instances. The comparative results have demonstrated the competitiveness of the proposed algorithm in finding a well-converged and well-distributed solution set. Moreover, an appealing property of GrEA is that its computational cost is almost independent on the number of hyperboxes in the grid and only increases linearly with the number of objectives. This is against the common belief that grid-based approaches are not suitable in many-objective optimisation in view of their operation relying on the hyperboxes that exponentially grow in size with the number of objectives [45].

In Chapter 5, we presented a bi-goal evolution (BiGE) framework for many-objective optimisation. BiGE converts a many-objective problem into a bi-goal (objective) optimisation problem regarding proximity and diversity, and then handles it using the Pareto dominance relation in this bi-goal domain. This idea was motivated by two difficulties of EAs facing in many-objective optimisation: 1) the conflict between proximity and diversity goals is aggravated with the increase of the number of objectives and 2) Pareto dominance which works well on bi-objective problems loses its effectiveness in a high-dimensional space. To verify BiGE, we carried out systematic experiments by comparison with five state-of-the-art algorithms on three groups of continuous and combinatorial benchmark suites with 5, 10 and 15 objectives as well as on a real-world problem. The experimental results have shown that unlike the peer algorithms which work well on only a fraction of the test problems, BiGE can achieve a good balance between solutions' proximity and diversity on the test problems with various properties. This indicates the promise of bi-goal evolution as a new way of addressing many-objective problems, which may open up many possibilities for further development in the future.

Having begun with a detailed analysis of the difficulties of popular performance indicators encountering in many-objective optimisation, Chapter 6 proposed a performance comparison indicator (PCI) to assess Pareto front approximations obtained by

stochastic search algorithms. In contrast to current state of the art, PCI has several strengths, including no need of a specified reference set (this is against IGD [42] and IGD$^+$ [121]), no requirement of parameter setting (against hypervolume [294]), a comprehensive consideration of all solutions in the approximation set (against ϵ-indicator [295]), (weak) compliance with Pareto dominance (against IGD), and quadratic time complexity (against hypervolume). These strengths enable PCI to be well suited for performance assessment of Pareto front approximations in many-objective optimisation.

Finally, Chapter 7 proposed a test problem (Rectangle problem) to aid the visual investigation of high-dimensional multi-objective search. Key features of the Rectangle problem are that the Pareto optimal solutions 1) lie in a rectangle in the two-variable decision space and 2) are similar (in the sense of Euclidean geometry) to their images in the four-dimensional objective space. In this case, it is easy to examine the behaviour of objective vectors in terms of both proximity and diversity, by observing their closeness to the optimal rectangle and their distribution in the rectangle, respectively, in the decision space. Fifteen algorithms were investigated on three instances of the Rectangle problem. Unlike on many existing four-objective problems (e.g., the DTLZ problem suite [57]) where modern many-objective algorithms (or even classic Pareto-based algorithms) perform fairly well, on the proposed problem most of the tested algorithms struggled, either in leading solutions into the optimal region or in maintaining solutions' diversity. This indicates that the Rectangle problem can also be used as a challenging benchmark function to test algorithms' ability in many-objective optimisation. Finally, it is worth mentioning that among all the 15 algorithms, only SPEA2+SDE obtained a well-converged and well-distributed solution set on all the three Rectangle problem instances. This, in turn, verifies the effectiveness of our SDE strategy (Chapter 3) in achieving good performance in terms of both proximity and diversity.

Thus, when taken together, these contributions presented in this thesis represent a significant advance in the state of the art of evolutionary many-objective optimisation, which provides considerable help for researchers and practitioners in both algorithm de-

velopment and problem solving. When designing a Pareto-based algorithm, researchers only need to focus on tackling two- or three-objective problems; for a problem with many objectives, SDE (Chapter 3) could be easily used. When working out a many-objective algorithm, the developer can use the Rectangle problem (Chapter 7) to investigate the behaviour of the algorithm or/and the PCI indicator (Chapter 6) to assess the performance of the algorithm. When facing a many-objective problem in hand, the user can directly adopt the algorithm GrEA (Chapter 4) or design new proximity and diversity estimation methods under the bi-goal evolution framework (Chapter 5).

Finally, it is worth mentioning that some of the approaches presented in this thesis have already been taken up by researchers and practitioners recently. This includes them being investigated in various test problems [282, 141, 283, 275], examined by different performance metrics [93, 95], integrated with other techniques [262, 167], used as benchmark algorithms in experimental studies [171, 6, 276, 36, 34], and applied to some real-world problems [222, 223].

8.2 Future work

Average time complexity of GrEA. We have presented the time complexity of GrEA to be bounded by $O(mN^2)$ or $O(LN^2)$ on average, whichever is greater (Section 4.3.4). There, m and N are the number of objectives and the population size, respectively, and L denotes, for a set of Pareto nondominated solutions, the average number of the solutions in the set that are grid-dominated by one member of the set, where the grid environment is formed by the set. We have $O(1) \leq O(L) \leq O(N)$. The former occurs when all solutions in the set are grid-nondominated to each other, and the latter occurs when a totally ordered relation regarding grid dominance holds for all solutions in the set. However, it would be interesting to know the specific relationship between L and N as well as other parameters like the number of objectives and grid divisions.

Instantiation of different performance estimation and selection operations in BiGE. In Chapter 5, we have presented a bi-goal evolution framework and imple-

mented two specific estimation methods of individuals' performance (i.e. proximity and crowding degree). One area of further research is to explore (or introduce) other proximity and crowding degree estimation methods. Also, in the environmental selection, BiGE used the nondominated sorting of the proximity and diversity goals to rank individuals. Other classic comparison strategies could also be used, such as nondominated ranking [74] and strength [294]. All of these allow us to have a deeper understanding of the behaviour of the presented bi-goal evolution mechanism.

Accurate calulation of dominance distance in PCI. In PCI, we approximately measured the dominance distance of a point set to another (which only requires quadratic time complexity), since there are exponentially increasing possibilities for points in one set to divide another (Section 6.3). Now, the question is whether there exists an efficient method to accurately measure the dominance distance of a set to another. If the answer is yes, we could directly compare two Pareto front approximations by their dominance distance of each other, rather than dividing solutions into many clusters.

Changeability of objective dimensionality in the Rectangle problem. In Chapter 7, we have presented a test problem with two variables and four objectives. Despite such a "low" objective dimensionality, the Rectangle problem poses big challenges to many-objective algorithms. Nevertheless, it is naturally desirable to construct new problems (or to improve the proposed problem) whose number of objectives is changeable, i.e., can be set by the user freely.

Recombination operation. A difficulty in evolutionary many-objective optimisation is the inefficiency of recombination operation. In a high-dimensional space, individuals are likely to be widely distant from each other. Thus, two distant parent individuals are more likely to produce offspring that are also distant from the parents, which can slow down the search process [52]. A straightforward way to deal with this difficulty is to select neighbouring solutions to perform the recombination operation [116, 118]. None of the presented works in this thesis involves this. However, the grid environment can be a good tool to identify neighbouring individuals in a population. Thus, such a neighbouring-solutions based recombination operation could be directly implemented

in the grid-based evolutionary algorithm GrEA.

Incorporation of decision-maker preferences. A major purpose of EMO algorithms is to assist the decision maker to select a single solution (or a few solutions) that fits her/his preferences [38, 20, 214]. However, since EMO algorithms usually equip the decision-maker with an approximation of the whole Pareto front, it might be difficult for the decision maker to choose her/his preferred one(s), especially in many-objective optimisation. One focus of our future research is to work with the preferences supplied by the decision maker. This includes the incorporation of preference information into the presented many-objective evolutionary approaches and the design of new performance indicators for preference-based search.

Real-world application. In this thesis, all the presented evolutionary many-objective approaches (GrEA, SDE and BiGE) were investigated mainly on well-defined continuous and combinatorial benchmark suites. These test problems, despite their diversity, cannot always match precisely the characteristics of real-world ones. Recently, we have made an attempt of using SDE to conduct the optimal product selection in the software product line (SPL) [101]. In the future, we will apply and adapt the proposed many-objective optimisation approaches to more real-world problems.

Bibliography

[1] S. F. Adra and P. J. Fleming. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, 2011.

[2] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. *On the Surprising Behavior of Distance Metrics in High Dimensional Space*. Springer, 2001.

[3] H. Aguirre and K. Tanaka. Adaptive ε-ranking on many-objective problems. *Evolutionary Intelligence*, 2(4):183–206, 2009.

[4] H. Aguirre and K. Tanaka. Many-objective optimization by space partitioning and adaptive ϵ-ranking on MNK-landscapes. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 407–422. 2009.

[5] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460, 2015.

[6] M. Asafuddoula, T. Ray, and H. K. Singh. Characterizing Pareto front approximations in many-objective optimization. In *Proceedings of the 17th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 607–614, 2015.

[7] M. Asafuddoula, H. K. Singh, and T. Ray. Six-sigma robust design optimization using a many-objective decomposition based evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 19(4):490–507, 2015.

[8] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 555–562, 2009.

[9] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In *Proceedings of the 10th ACM SIGEVO workshop on Foundations of Genetic Algorithms (FOGA)*, pages 87–102. 2009.

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

[11] S. Bandyopadhyay and A. Mukherjee. An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution. *IEEE Transactions on Evolutionary Computation*, 19(3):400–413, 2015.

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

[13] M. Basseur and E. K. Burke. Indicator-based multi-objective local search. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3100–3107, 2007.

[14] L. S. Batista, F. Campelo, F. G. Guimarães, and J. Ramírez. Pareto cone ϵ-dominance: Improving convergence and diversity in multiobjective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 76–90, 2011.

[15] P. J. Bentley and J. P. Wakefield. Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms. In *Soft Computing in Engineering Design and Manufacturing*, chapter 5, pages 231–240. 1997.

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

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

[18] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences*, 178(20):3908–3924, 2008.

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

[20] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer-Verlag, Berlin, Heidelberg, 2008.

[21] J. Branke, T. Kaußler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32(6):499–507, 2001.

[22] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry: Theory and Applications*, 43(6-7):601–610, 2010.

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

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

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

[26] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *Proceedings of the 22nd*

International Joint Conference on Artificial Intelligence (IJCAI), pages 1198–1203, 2011.

[27] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. On the effects of adding objectives to plateau functions. *IEEE Transactions on Evolutionary Computation*, 13(3):591–603, 2009.

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

[29] D. Brockhoff, T. Wagner, and H. Trautmann. $R2$ Indicator-Based Multiobjective Search. *Evolutionary Computation*, 2015.

[30] D. Brockhoff and E. Zitzler. Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature (PPSN) IX*, pages 533–542. 2006.

[31] D. Brockhoff and E. Zitzler. Dimensionality reduction in multiobjective optimization with (partial) dominance structure preservation: Generalized minimum objective subset problems. *TIK Report*, 247, 2006.

[32] D. Brockhoff and E. Zitzler. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2086–2093, 2007.

[33] D. Brockhoff and E. Zitzler. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation*, 17(2):135–166, 2009.

[34] L. Cai, S. Qu, Y. Yuan, and X. Yao. A clustering-ranking method for many-objective optimization. *Applied Soft Computing*, 35:681–694, 2015.

[35] S. Chen and T. Chiang. Evolutionary many-objective optimization by MO-NSGA-II with enhanced mating selection. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1397–1404, 2014.

[36] J. Cheng, G. G. Yen, and G. Zhang. A many-objective evolutionary algorithm with enhanced mating and environmental selections. *IEEE Transactions on Evolutionary Computation*, 19(4):592–605, 2015.

[37] P. Chiu and C. L. Bloebaum. Hyper-Radial Visualization (HRV) method with range-based preferences for multi-objective decision making. *Structural and Multidisciplinary Optimization*, 40(1-6):97–115, 2010.

[38] C. A. C. Coello. Handling preferences in evolutionary multiobjective optimization: A survey. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 1, pages 30–37, 2000.

[39] C. A. C. Coello. Evolutionary multi-objective optimization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5):444–447., 2011.

[40] C. A. C. Coello and R. L. Becerra. Evolutionary multiobjective optimization in materials science and engineering. *Materials and Manufacturing Processes*, 24(2):119–129, 2009.

[41] C. A. C. Coello and G. B. Lamont. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, 2004.

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

[43] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, Heidelberg, second edition, 2007.

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

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

[46] D. W. Corne, J. D. Knowles, and M. J. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature (PPSN) VI*, pages 839–848. 2000.

[47] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. CRC Press, 2000.

[48] P. Czyzak and A. Jaszkiewicz. Pareto simulated annealing – A metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, 1998.

[49] A. B. De Carvalho and A. Pozo. Measuring the convergence and diversity of CDAS multi-objective particle swarm optimization algorithms: A study of many-objective problems. *Neurocomputing*, 75(1):43–51, 2012.

[50] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley, New York, 2001.

[51] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature (PPSN) VI*, pages 849–858. 2000.

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

[53] K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. Technical Report 2002004, KanGAL, Indian Institute of Technology, 2002.

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

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

[56] K. Deb and D. K. Saxena. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3353–3360, 2006.

[57] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Berlin, Germany: Springer, 2005.

[58] F. di Pierro. *Many-Objective Evolutionary Algorithms and Applications to Water Resources Engineering.* PhD thesis, University of Exeter, 2006.

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

[60] A. Diaz-Manriquez, G. Toscano-Pulido, C. A. C. Coello, and R. Landa-Becerra. A ranking method based on the $R2$ indicator for many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1523–1530, 2013.

[61] N. Drechsler, R. Drechsler, and B. Becker. Multi-objective optimisation based on relation favour. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 154–166, 2001.

[62] A. Elhossini, S. Areibi, and R. Dony. Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evolutionary Computation*, 18(1):127–156, 2010.

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

[64] R. M. Everson, D. J. Walker, and J. E. Fieldsend. Edges of mutually non-dominating sets. In *Proceeding of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 607–614, 2013.

[65] A. Farhang-Mehr and S. Azarm. Entropy-based multi-objective genetic algorithm for design optimization. *Structural and Multidisciplinary Optimization*, 24(5):351–361, 2002.

[66] A. Farhang-Mehr and S. Azarm. An information-theoretic metric for assessing multi-objective optimization solution set quality. *Transactions of the ASME, Journal of Mechanical Design*, 125(4):655–663, 2003.

[67] M. Farina and P. Amato. On the optimal solution definition for many-criteria optimization problems. In *NAFIPS Fuzzy Information Processing Society 2002 Annual Meeting of the North American*, pages 233–238, 2002.

[68] M. Farina and P. Amato. Fuzzy optimality and evolutionary multiobjective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 58–72. 2003.

[69] J. Fieldsend and R. Everson. Visualising high-dimensional Pareto relationships in two-dimensional scatterplots. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 558–572, 2013.

[70] J. Figueira, A. Liefooghe, E. Talbi, and A. P. Wierzbicki. A parallel multiple reference point approach for multi-objective optimization. *European Journal of Operational Research*, 205(2):390–400, 2010.

[71] P. Fleming, R. Purshouse, and R. Lygoe. Many-objective optimization: An engineering design perspective. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 14–32. 2005.

[72] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA)*, volume 1, pages 416–423, 1993.

[73] C. M. Fonseca and P. J. Fleming. Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 42–52, 1995.

[74] C. M. Fonseca and P. J Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[75] C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Parallel Problem Solving from Nature (PPSN)*, pages 584–593. 1996.

[76] T. Friedrich, K. Bringmann, T. Voß, and C. Igel. The logarithmic hypervolume indicator. In *Proceedings of the 11th ACM SIGEVO workshop on Foundations of Genetic Algorithms (FOGA)*, pages 81–92, 2011.

[77] T. Friedrich, C. Horoba, and F. Neumann. Multiplicative approximations and the hypervolume indicator. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 571–578, 2009.

[78] T. Friedrich, F. Neumann, and C. Thyssen. Multiplicative approximations, optimal hypervolume distributions, and the choice of the reference point. *Evolutionary computation*, 23(1):131–159, 2015.

[79] M. Garza-Fabre, G. Toscano-Pulido, and C. A. C. Coello. Ranking methods for many-objective optimization. In *MICAI 2009: Advances in Artificial Intelligence*, pages 633–645. 2009.

[80] M. Garza-Fabre, G. Toscano-Pulido, and C. A. C. Coello. Two novel approaches for many-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

[81] M. Garza-Fabre, G. Toscano-Pulido, C. A. C. Coello, and E. Rodriguez-Tello. Effective ranking + speciation = Many-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 2115–2122, 2011.

[82] S. B. Gee, X. Qiu, and K. C. Tan. A novel diversity maintenance scheme for evolutionary multi-objective optimization. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 270–277. 2013.

[83] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming. Generalized decomposition. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 428–442, 2013.

[84] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming. Generalized decomposition and cross entropy methods for many-objective optimization. *Information Sciences*, 282:363–387, 2014.

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

[86] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-wesley, 1989.

[87] F. Gu, H. Liu, and K. C. Tan. A multiobjective evolutionary algorithm using dynamic weight design method. *International Journal of Innovative Computing Information and Control*, 8(5):3677–3688, 2012.

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

[89] D. Hadka and P. Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, 21(2):231–259, 2013.

[90] N. Hallam, P. Blanchfield, and G. Kendall. Handling diversity in evolutionary multiobjective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 3, pages 2233–2240, 2005.

[91] T. Hanne. Global multiobjective optimization with evolutionary algorithms: Selection mechanisms and mutation control. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 197–212. 2001.

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

[93] Z. He and G. G. Yen. Comparison of many-objective evolutionary algorithms using performance metrics ensemble. *Advances in Engineering Software*, 76:1–8, 2014.

[94] Z. He and G. G. Yen. Many-objective evolutionary algorithm: Objective space reduction + diversity improvement. *IEEE Transactions on Evolutionary Computation*, 2015, in press.

[95] Z. He and G. G. Yen. Visualization and performance metric in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2015, in press.

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

[97] C. Henard, M. Papdakis, M. Harman, and Y. L. Traon. Combining multi-objective search and constraint solving for configuring large software product

lines. In *Proceedings of the 2015 International Conference on Software Engineering (ICSE)*, 2015.

[98] A. G. Hernández-Díaz and C. A. C. Coello. MOMBI: A new metaheuristic for many-objective optimization based on the $R2$ indicator. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2488–2495, 2013.

[99] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. C. Coello, and J. Molina. Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 15:493–517, 2007.

[100] J. G. Herrero, A. Berlanga, and J. M. M. López. Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters. *IEEE Transactions on Evolutionary Computation*, 13(1):151–168, 2009.

[101] R. M. Hierons, M. Li, X. Liu, S. Segura, and W. Zheng. An improved method for optimal product selection from feature models. *ACM Transactions on Software Engineering and Methodology*, 2015, under review.

[102] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. Dna visual and analytic data mining. In *Proceedings of IEEE Visualization*, pages 437–441, 1997.

[103] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *Proceeding of the First IEEE Conference on Evolutionary Computation (CEC)*, pages 82–87, 1994.

[104] C. Horoba and F. Neumann. Approximating Pareto-optimal sets using diversity strategies in evolutionary multi-objective optimization. In C. C. A. Coello, C. Dhaenens, and L. Jourdan, editors, *Advances in multi-objective nature inspired computing, Studies in Computational Intelligence (SCI) 272*, pages 23–44. Springer, 2010.

[105] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[106] E. J. Hughes. Multiple single objective Pareto sampling. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 4, pages 2678–2684, 2003.

[107] E. J. Hughes. Evolutionary many-objective optimisation: Many once or one many? In *IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 222–227, 2005.

[108] E. J. Hughes. MSOPS-II: A general-purpose many-objective optimiser. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3944–3951, 2007.

[109] E. J. Hughes. Radar waveform optimisation as a many-objective application benchmark. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 700–714, 2007.

[110] E. J. Hughes. Fitness assignment methods for many-objective problems. In *Multiobjective Problem Solving from Nature*, pages 307–329. Springer Berlin Heidelberg, 2008.

[111] E. J. Hughes. Many-objective directed evolutionary line search. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 761–768, 2011.

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

[113] K. Ikeda, H. Kita, and S. Kobayashi. Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal? In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 957–962, 2001.

[114] A. Inselberg and B. Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991.

[115] H. Ishibuchi, N. Akedo, and Y. Nojima. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 649–656, 2011.

[116] H. Ishibuchi, N. Akedo, and Y. Nojima. Recombination of similar parents in SMS-EMOA on many-objective 0/1 knapsack problems. In *Parallel Problem Solving from Nature (PPSN)*, pages 132–142. 2012.

[117] H. Ishibuchi, N. Akedo, and Y. Nojima. A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems. In *Learning and Intelligent Optimization Conference (LION)*, pages 231–246. 2013.

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

[119] 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 *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 91–100, 2010.

[120] H. Ishibuchi, H. Masuda, and Y. Nojima. A study on performance evaluation ability of a modified inverted generational distance indicator. In *Proceedings of the 17th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 695–702, 2015.

[121] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. In *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, pages 170–177, 2014.

[122] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems. In *2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, pages 178–184, 2014.

[123] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 110–125. 2015.

[124] 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 *Evolutionary Multi-Criterion Optimization (EMO)*, pages 438–452. 2009.

[125] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima. Simultaneous use of different scalarizing functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 519–526, 2010.

[126] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima. Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 649–656, 2008.

[127] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Behavior of evolutionary many-objective optimization. In *Computer Modeling and Simulation (UKSIM)*, pages 266–271, 2008.

[128] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 2419–2426, 2008.

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

[130] H. Ishibuchi, M. Yamane, N. Akedo, and Y. Nojima. Many-objective and many-variable test problems for visual examination of multiobjective search. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1491–1498, 2013.

[131] A. L. Jaimes, H. Aguirre, K. Tanaka, and C. A. C. Coello. Objective space partitioning using conflict information for many-objective optimization. In *Parallel Problem Solving from Nature (PPSN) XI*, pages 657–666. 2010.

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

[133] A. L. Jaimes, C. A. C. Coello, H. Aguirre, and K. Tanaka. Adaptive objective space partitioning using conflict information for many-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 151–165. 2011.

[134] A. L. Jaimes, C. A. C. Coello, and J. E. U. Barrientos. Online objective reduction to deal with many-objective problems. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 423–437. 2009.

[135] A. L. Jaimes, C. A. C. Coello, and D. Chakraborty. Objective reduction using a feature selection technique. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 673–680, 2008.

[136] A. L. Jaimes, A. A. Montano, and C. A. C. Coello. Preference incorporation to solve many-objective airfoil design problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1605–1612, 2011.

[137] A. L. Jaimes, A. Oyama, and K. Fujii. Space trajectory design: Analysis of a real-world many-objective optimization problem. In *2013 IEEE Congress on Evolutionary Computation (CEC)*, pages 2809–2816, 2013.

[138] A. L. Jaimes, L. V. S. Quintero, and C. A. C. Coello. Ranking methods in many-objective evolutionary algorithms. In R. Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, pages 413–434. Berlin, Germany: Springer, 2009.

[139] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, Aug 2014.

[140] S. Jiang, Z. Cai, J. Zhang, and Y. S. Ong. Multiobjective optimization by decomposition with Pareto-adaptive weight vectors. In *2011 International Conference on Natural Computation (ICNC)*, volume 3, pages 1260–1264, 2011.

[141] S. Jiang and S. Yang. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts. *IEEE Transactions on Cybernetics*, 2015, in press.

[142] S. Jiang, J. Zhang, Y. S. Ong, A. N. Zhang, and P. S. Tan. A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics*, 2015, in press.

[143] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - A survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317, 2005.

[144] İ. Karahan and M. Köksalan. A territory defining multiobjective evolutionary algorithm and preference incorporation. *IEEE Transactions on Evolutionary Computation*, 14(4):636–664, 2010.

[145] J. R. Kasprzyk, P. M. Reed, G. W. Characklis, and B. R. Kirsch. Many-objective de Novo water supply portfolio planning under deep uncertainty. *Environmental Modelling and Software*, 34:87–104, 2012.

[146] 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 Resources Research*, 45(12), 2009.

[147] V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 376–390. 2003.

[148] E. F. Khor, K. C. Tan, T. H. Lee, and C. K. Goh. A study on distribution preservation mechanism in evolutionary multi-objective optimization. *Artificial Intelligence Review*, 23(1):31–56, 2005.

[149] J. D. Knowles and D. W. Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *Congress on Evolutionary Computation (CEC)*, volume 1, 1999.

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

[151] J. D. Knowles and D. W. Corne. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, 2003.

[152] J. D. Knowles and D. W. Corne. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 757–771. 2007.

[153] J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report No. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2006.

[154] W. Kong, J. Ding, T. Chai, and J. Sun. Large-dimensional multi-objective evolutionary algorithms based on improved average ranking. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pages 502–507, 2010.

[155] M. Köppen, R. Vicente-Garcia, and B. Nickolay. Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 399–412. Springer Berlin / Heidelberg, 2005.

[156] M. Köppen and K. Yoshida. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 727–741. 2007.

[157] M. Köppen and K. Yoshida. Visualization of Pareto-sets in evolutionary multi-objective optimization. In *International Conference on Hybrid Intelligent Systems (HIS)*, pages 156–161, 2007.

[158] 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 *Proceedings of the 6th Learning and Intelligent Optimization Conference (LION)*, pages 86–100. 2012.

[159] J. W. Kruisselbrink, M. T. Emmerich, T. Bäck, A. Bender, A. P. Ijzerman, and E. Horst. Combining aggregation with Pareto optimization: A case study in evolutionary molecular design. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 453–467. 2009.

[160] F. Kudo and T. Yoshikawa. Knowledge extraction in multi-objective optimization problem based on visualization of Pareto solutions. In *2012 IEEE Congress on Evolutionary Computation (CEC)*, pages 860–865, 2012.

[161] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. Berkeley, CA, University of California Press., 1951.

[162] S. Kukkonen and K. Deb. A fast and effective method for pruning of non-dominated solutions in many-objective problems. In *Parallel Problem Solving from Nature (PPSN)*, pages 553–562. 2006.

[163] S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI)*, pages 1179–1186, 2006.

[164] S. Kukkonen and J. Lampinen. Ranking-dominance and many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3983–3990, 2007.

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

[166] M. Laumanns, E. Zitzler, and L. Thiele. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 181–196, 2001.

[167] B. Li, J. Li, K. Tang, and X. Yao. An improved two archive algorithm for many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2869–2876, 2014.

[168] B. Li, J. Li, K. Tang, and X. Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys*, 48(1):1–35, 2015.

[169] H. Li and D. Landa-Silva. An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 481–494, 2009.

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

[171] K. Li, K. Deb, Q. Zhang, and S. Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 2015, in press.

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

[173] M. Li, S. Yang, K. Li, and X. Liu. Evolutionary algorithms with segment-based search for multiobjective optimization problems. *IEEE Transactions on Cybernetics*, 44(8):1295–1313, 2014.

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

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

[176] M. Li, S. Yang, and X. Liu. A test problem for visual investigation of high-dimensional multi-objective search. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2140–2147, 2014.

[177] M. Li, S. Yang, and X. Liu. A performance comparison indicator for Pareto front approximations in many-objective optimization. In *Proceedings of the 17th Confernece on Genetic and Evolutionary Computation (GECCO)*, pages 703–710, 2015.

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

[179] M. Li, S. Yang, and X. Liu. Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2015, in press.

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

[181] M. Li, S. Yang, X. Liu, and K. Wang. IPESA-II: Improved Pareto envelope-based selection algorithm II. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 143–155, 2013.

[182] M. Li, S. Yang, X. Liu, and J. Zheng. ETEA: A Euclidean minimum spanning tree-based evolutionary algorithm for multiobjective optimization. *Evolutionary Computation*, 22(2):189–230, 2014.

[183] M. Li and J. Zheng. Spread assessment for evolutionary multi-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 216–230, 2009.

[184] M. Li, J. Zheng, K. Li, Q. Yuan, and R. Shen. Enhancing diversity for average ranking method in evolutionary many-objective optimization. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 647–656. 2010.

[185] M. Li, J. Zheng, R. Shen, K. Li, and Q. Yuan. A grid-based fitness strategy for evolutionary many-objective optimization. In *Proceedings of the 12th Annual Confernece on Genetic and Evolutionary Computation (GECCO)*, pages 463–470, 2010.

[186] M. Li, J. Zheng, and G. Xiao. Uniformity assessment for evolutionary multi-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 625–632, 2008.

[187] M. Liebscher, K. Witowski, and T. Goel. Decision making in multi-objective optimization for industrial applications–data mining and visualization of Pareto data. In *Proceedings of 8th World Congress on Structural and Multidisciplinary Optimization*, pages 1–5, 2009.

[188] G. Lizárraga-Lizárraga. *On the Evaluation of the Quality of Non-dominated Sets.* PhD thesis, Center for Research in Mathematics A.C., Computer Science Area, Guanajuato, Mexico, 2009.

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

[190] A. V. Lotov and K. Miettinen. Visualizing the Pareto frontier. In *Multiobjective optimization*, pages 213–243. Springer, 2008.

[191] K. Maneeratana, K. Boonlong, and N. Chaiyaratana. Compressed-objective genetic algorithm. In *Parallel Problem Solving from Nature (PPSN) IX*, pages 473–482. 2006.

[192] W. S. Meisel. Tradeoff decision in multiple criteria decision making. *Multiple Criteria Decision Making (MCDM)*, pages 461–476, 1973.

[193] O. J. Mengshoel and D. E. Goldberg. The crowding approach to niching in genetic algorithms. *Evolutionary Computation*, 16(3):315–354, 2008.

[194] Z. Michalewicz and D. B. Fogel. *How to solve it: Modern heuristics*. Springer, 2000.

[195] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.

[196] W. Mkaouer, M. Kessentini, A. Shaout, P. Koligheu, S. Bechikh, K. Deb, and A. Ouni. Many-objective software remodularization using NSGA-III. *ACM Transactions on Software Engineering and Methodology*, 24(3):17, 2015.

[197] H. J. F. Moen, N. B. Hansen, H. Hovland, and J. Tørresen. Many-objective optimization using taxi-cab surface evolutionary algorithm. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 128–142, 2013.

[198] S. Mostaghim and H. Schmeck. Distance based ranking in many-objective particle swarm optimization. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 753–762. 2008.

[199] S. Mostaghim and J. Teich. A new approach on many objective diversity measurement. In *Practical Approaches to Multi-Objective Optimization*, 2005.

[200] T. Murata, H. Ishibuchi, and M. Gen. Specification of genetic search directions in cellular multi-objective genetic algorithms. In *Evolutionary Multi-Criterion optimization (EMO)*, pages 82–95, 2001.

[201] T. Murata and A. Taki. Examination of the performance of objective reduction using correlation-based weighted-sum for many objective knapsack problems. In *10th International Conference on Hybrid Intelligent Systems (HIS)*, pages 175–180, 2010.

[202] K. Musselman and J. Talavage. A trade-off cut approach to multiple objective optimization. *Operations Research*, 28(6):1424–1435, 1980.

[203] K. Narukawa. Effect of dominance balance in many-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 276–290, 2013.

[204] K. Narukawa and T. Rodemann. Examining the performance of evolutionary many-objective optimization algorithms on a real-world application. In *the Sixth International Conference on Genetic and Evolutionary Computing (ICGEC)*, pages 316–319, 2012.

[205] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham. AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 12(4):439–457, 2008.

[206] S. Obayashi and D. Sasaki. Visualization and data mining of Pareto solutions using self-organizing map. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 796–809. 2003.

[207] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 878–885, 2003.

[208] A. Panichella, F. M. Kifetew, and P. Tonella. Reformulating branch coverage as a many-objective optimization problem. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–10, 2015.

[209] A. Ponsich, A. L. Jaimes, and C. A. C. Coello. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and

other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, 17(3):321–344, 2013.

[210] K. Praditwong, M. Harman, and X. Yao. Software module clustering as a multi-objective search problem. *IEEE Transactions on Software Engineering*, 37(2):264–282, 2011.

[211] K. Praditwong and X. Yao. A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm. In *International Conference on Computational Intelligence and Security*, volume 1, pages 286–291, 2006.

[212] K. Praditwong and X. Yao. How well do multi-objective evolutionary algorithms scale to large problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3959–3966, 2007.

[213] A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap visualization of population based multi-objective algorithms. In *Evolutionary multi-criterion optimization (EMO)*, pages 361–375, 2007.

[214] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang. A review of hybrid evolutionary multiple criteria decision making methods. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1147–1154, 2014.

[215] R. C. Purshouse and P. J. Fleming. Evolutionary many-objective optimisation: An exploratory analysis. In *IEEE Congress on Evolutionary Computation (CEC)*, volume 3, pages 2066–2073, 2003.

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

[217] R. C. Purshouse, C. Jalbă, and P. J. Fleming. Preference-driven co-evolutionary algorithms show promise for many-objective optimisation. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 136–150. 2011.

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

[219] F. Qiu, Y. Wu, L. Wang, and B. Jiang. Bipolar preferences dominance based evolutionary algorithm for many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.

[220] L. Rachmawati and D. Srinivasain. Dynamic resizing for grid-based archiving in evolutionary multi-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 3975–3982, 2007.

[221] L. Rachmawati and D. Srinivasan. Preference incorporation in multi-objective evolutionary algorithms: A survey. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 962–968, 2006.

[222] A. Ramírez, J. R. Romero, and S. Ventura. On the performance of multiple objective evolutionary algorithms for software architecture discovery. In *Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 1287–1294, 2014.

[223] A. Ramírez, J. R. Romero, and S. Ventura. A comparative study of many-objective evolutionary algorithms for the discovery of software architectures. *Empirical Software Engineering*, 2015, in press.

[224] T. Ray, K. Tai, and K. C. Seow. Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, 33(4):399–424, 2001.

[225] P. C. Roy, M. Islam, K. Murase, and X. Yao. Evolutionary path control strategy for solving many-objective optimization problem. *IEEE Transactions on Cybernetics*, 45(4):702–715, 2015.

[226] G. Rudolph, B. Naujoks, and M. Preuss. Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 36–50, 2007.

[227] G. Rudolph, H. Trautmann, S. Sengupta, and O. Schütze. Evenly spaced Pareto front approximations for tricriteria problems based on triangulation. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 443–458. Springer, 2013.

[228] H. Sato. Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization. In *Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 645–652, 2014.

[229] H. Sato, H. Aguirre, and K. Tanaka. Controlling dominance area of solutions and its impact on the performance of MOEAs. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 5–20. 2007.

[230] H. Sato, H. Aguirre, and K. Tanaka. Self-controlling dominance area of solutions in evolutionary many-objective optimization. In *Simulated Evolution and Learning (SEAL)*, pages 455–465. 2010.

[231] H. Sato, H. Aguirre, and K. Tanaka. Improved S-CDAs using crossover controlling the number of crossed genes for many-objective optimization. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 753–760, 2011.

[232] H. Sato, C. A. C. Coello, H. Aguirre, and K. Tanaka. Adaptive control of the number of crossed genes in many-objective evolutionary optimization. In *Learning and Intelligent Optimization (LION)*, pages 478–484. 2012.

[233] D. Saxena and K. Deb. Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 772–787, 2007.

[234] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation*, 17(1):77–99, 2013.

[235] D. Saxena, Q. Zhang, J. Duro, and A. Tiwari. Framework for many-objectivet test problems with both simple and complicated Pareto-set shapes. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 197–211. 2011.

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

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

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

[239] O. Schütze, A. Lara, and C. A. C. Coello. On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 15(4):444–455, 2011.

[240] H. K. Singh, A. Isaacs, and T. Ray. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 15(4):539–556, 2011.

[241] H. K. Singh, A. Isaacs, T. Ray, and W. Smith. A study on the performance of substitute distance based approaches for evolutionary many-objective optimization. In *Simulated Evolution and Learning (SEAL)*, pages 401–410. 2008.

[242] B. Soylu and M. Köksalan. A favorable weight-based evolutionary algorithm for multiple criteria problems. *IEEE Transactions on Evolutionary Computation*, 14(2):191–205, 2010.

[243] A. Sülflow, N. Drechsler, and R. Drechsler. Robust multi-objective optimization in high dimensional spaces. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 715–726. 2007.

[244] K. C. Tan, E. F. Khor, and T. H. Lee. *Multiobjective Evolutionary Algorithms and Applications*. London, 2005.

[245] K. C. Tan, T. H. Lee, and E. F. Khor. Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial Intelligence Review*, 17:251–290, 2002.

[246] Y. Tan, Y. Jiao, H. Li, and X. Wang. MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & Operations Research*, 40(6):1648–1660, 2013.

[247] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[248] O. Teytaud. On the hardness of offline multi-objective optimization. *Evolutionary Computation*, 15(4):475–491, 2007.

[249] T. Tusar and B. Filipic. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, 2015.

[250] J. J. Valdés and A. J. Barton. Visualizing high dimensional objective spaces for multi-objective optimization: A virtual reality approach. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 4199–4206, 2007.

[251] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1999.

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

[253] D. A. Van Veldhuizen and G. B. Lamont. On measuring multiobjective evolutionary algorithm performance. In *Congress on Evolutionary Computation (CEC)*, volume 1, pages 204–211, 2000.

[254] C. von Lücken, B. Barán, and C. Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756, 2014.

[255] M. Wagner and F. Neumann. A fast approximation-guided evolutionary multi-objective algorithm. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 687–694, 2013.

[256] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 742–756. 2007.

[257] D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualisation and ordering of many-objective populations. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

[258] D. J. Walker, R. M. Everson, and J. E. Fieldsend. Rank-based dimension reduction for many-criteria populations. In *Proceedings of the 13th Conference Companion on Genetic and Evolutionary Computation (GECCO)*, pages 107–108, 2011.

[259] D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualising mutually non-dominating solution sets in many-objective optimisation. *IEEE Transactions on Evolutionary Computation*, 17(2):165–184, 2013.

[260] H. Wang, L. Jiao, and X. Yao. TwoArch2: An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(4):524–541, 2015.

[261] H. Wang and X. Yao. Objective reduction based on nonlinear correlation information entropy. *Soft Computing*, 2015, in press.

[262] J. Wang, W. Zhang, and J. Zhang. Cooperative differential evolution with multiple populations for multiobjective optimization. *IEEE Transactions on Cybernetics*, 2015, in press.

[263] R. Wang. *Preference-inspired Co-evolutionary Algorithms*. PhD thesis, Faculty of Engineering Department of Automatic Control and Systems Engineering, University of Sheffield, December 2013.

[264] R. Wang, R. C. Purshouse, and P. J. Fleming. Preference-inspired co-evolutionary algorithms for many-objective optimisation. *IEEE Transactions on Evolutionary Computation*, 17(4):474–494, 2013.

[265] R. Wang, R. C. Purshouse, and P. J. Fleming. Preference-inspired co-evolutionary algorithms using weight vectors. *European Journal of Operational Research*, 243(2):423–441, 2015.

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

[267] L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, 2012.

[268] U. K. Wickramasinghe and X. Li. Using a distance metric to guide PSO algorithms for many-objective optimization. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 667–674, 2009.

[269] J. Wu and S. Azarm. Metrics for quality assessment of a multiobjective design optimization solution set. *Transactions of the ASME, Journal of Mechanical Design*, 123:18–25, 2001.

[270] S. Yang and C. Li. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 14(6):959–974, 2010.

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

[272] G. G. Yen and Z. He. Performance metric ensemble for multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 18(1):131–144, 2014.

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

[274] Y. Yuan, H. Xu, and B. Wang. Evolutionary many-objective optimization using ensemble fitness ranking. In *Proceedings of the 16th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 669–676, 2014.

[275] Y. Yuan, H. Xu, B. Wang, and X. Yao. A new dominance relation based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2015, in press.

[276] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 2015, in press.

[277] S. Zapotecas-Martinez, H. E. Aguirre, K. Tanaka, and C. A. C. Coello. On the low-discrepancy sequences and their use in MOEA/D for high-dimensional

objective spaces. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 2835–2842, 2015.

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

[279] Q. Zhang, W. Liu, and H. Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 203–208, 2009.

[280] Q. Zhang, A. Zhou, and Y. Jin. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63, 2008.

[281] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari. Multiobjective optimization test instances for the CEC 2009 special session and competition. Working Report CES-487, School of CS & EE, University of Essex, 2009.

[282] X. Zhang, Y. Tian, and Y. Jin. A knee point driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(6):761–776, 2015.

[283] J. Zheng, H. Bai, R. Shen, and M. Li. A comparative study use of OTL for many-objective optimization. In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, pages 1411–1412, 2015.

[284] W. Zheng, R. M. Hierons, M. Li, X. Liu, and V. Vinciotti. Multi-objective optimisation for regression testing. *Infomation Sciences*, 2015, in press.

[285] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[286] Y. Zhou and J. He. Convergence analysis of a self-adaptive multi-objective evolutionary algorithm based on grids. *Information Processing Letters*, 104(4):117–122, 2007.

[287] Z. Zhu, G. Zhang, M. Li, and X. Liu. Evolutionary multi-objective workflow scheduling in cloud. *IEEE Transactions on Parallel and Distributed Systems*, 2015, in press.

[288] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Zurich, Switzerland: Swiss Federal Institute of Technology (ETH), 1999.

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

[290] E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of Pareto set approximations. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization*, volume 5252, pages 373–404. Springer Berlin / Heidelberg, 2008.

[291] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 832–842. 2004.

[292] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation and Control*, pages 95–100, 2002.

[293] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - A comparative case study. In *Parallel Problem Solving from Nature (PPSN) V*, pages 292–301. 1998.

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

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

[296] X. Zou, Y. Chen, M. Liu, and L. Kang. A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(5):1402–1412, 2008.