UNIVERSITY OF SHEFFIELD

Preference-inspired Co-evolutionary Algorithms

by

Rui Wang

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering
Department of Automatic Control and Systems Engineering

December 2013

Abstract

The simultaneous optimisation of many objectives (say, in excess of 3), in order to obtain a full and satisfactory set of trade-off solutions to support a posteriori decision-making, remains challenging. To solve many-objective optimisation problems (MaOPs), a novel class of algorithms, namely, preference-inspired co-evolutionary algorithms (PICEAs) is proposed based on a concept of co-evolving the common population of candidate solutions with a family of decision-maker preferences.

Two realisations of PICEAs, namely, PICEA-g and PICEA-w, are studied. PICEA-g co-evolves goal vectors with candidate solutions. The algorithm is demonstrated to perform better than or competitively with four of the best-in-class multi-objective evolutionary algorithms (MOEAs) on the benchmark MaOPs. PICEA-w co-evolves weight vectors with candidate solutions. PICEA-w performs better than or competitively with other leading decomposition based algorithms on the benchmark MaOPs. Moreover, PICEA-w eliminates the need to specify appropriate weights in advance of performing the optimisation, which leads the algorithm to be less sensitive to the problem geometries. As performance of MOEAs is often affected by the associated parameter configurations, parameter sensitivities of both the PICEAs are empirically studied, and some suggestions on the settings of parameters are provided.

This research also proposes a novel and unified approach, namely, interactive PICEA-g (iPICEA-g) for *a priori* or *progressive* multi-objective optimisation and decision-making. This approach is derived from PICEA-g by co-evolving goal vectors that are exclusively generated in regions of interest to a decision-maker. iPICEA-g, to the best of the author's knowledge, is the first approach that is simultaneously able to handle multiple preferences types such as aspirations, weights or even via visually brushing, and that is also able to support multiple regions of interest. The iPICEA-g is demonstrated to be effective on different benchmark problems as well as a real-world problem – aircraft control system design problem.

Statement of Originality

Unless otherwise stated in the text, the work described in this thesis was carried out solely by the candidate. None of this work has already been accepted for any other degree, nor is it being concurrently submitted in candidature for any degree.

Candidate: ______________________________

Candidate Name

Supervisor: ______________________________

Supervisor Name

Acknowledgements

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

First and foremost, I would like to express deepest appreciation to my supervisors Prof. Peter Fleming and Dr. Robin Purshouse. Thanks for their persistent support and valuable advice within the last three years. Prof. Peter Fleming is very kind and always encourages me whenever and wherever I face difficulties. Dr. Robin Purshouse is very handsome, nice and strict. His invaluable input and guidance has been supporting me during the whole research. Thanks for them. Without them I would not have the opportunity to do postgraduate research and been able to finish this thesis in time.

Many thanks go to my colleagues who have been involved in discussions that have helped to shape not only this thesis but also general research studies: Martha Arbayani Bin Zaidan, Maszatul Munirah Mansor, Vigneshwaran Venugopalan, Hasanain Bashar, Rishi Relan, Andy Mills, Rajesh Kudikala and Ioannis Giagkiozis. A very special thank you goes to Renata Ashton for her kindest help in everything in Sheffield.

I would also like to thank my co-referee Dr. Roderich Gross and Prof. Yaochu Jin for their willingness to read and judge the thesis, for the challenging but fruitful comments on the manuscript and for coming to Sheffield for the final exam.

In addition, a thank you to Prof. Carlos Artemio Coello Coello for his kindest support and valuable advices during my visit to Cinvestav-IPN. Thanks also to people in Cinvestav-IPN, notably, Adriana Menchaca Mendez and Arturo Yee.

Further thank-yous are offered to my friends: Chang Liu, Biao Yang, HaoYu Wang, Yu Gu, Zuli Chen, Jiacheng Liu, Xueyan Zhao, Xiliang Zhang, Yuzhu Guo and Ramida Rattanakam. Thanks for their welcome distraction. Without them this thesis could have been finished earlier.

I would also like to thank my parents for their endless love and support.

This work is funded by a three year PhD Studentship from the Department of Automatic Control and System Engineering, The University of Sheffield, UK and the China Scholarship Council.

To my parents
Liu Yanfang and Wang Ningning
with love and respect
...
...
To ber
...
...
To something I never know
...
...

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

BCI	bootstrapping confidence interval
EMO	evolutionary multi-objective optimization
DE	differential evolution
DM	decision-maker
DTLZ	Deb, Thiele, Laumanns and Zitzler test suite
DTLZn-Y	The n-th DTLZ problem with Y objectives
MOP	multi-objective problem
MaOP	many-objective problem
PCA	principle component analysis
PM	polynomial mutation
ROI	region of interest
SBX	simulated binary crossover
SOP	single-objective problem
SPX	simplex crossover
WFG	walking fish group test suite
WFGn-Y	The n-th WFG problem with Y objectives
ZDT	Zitzler, Deb and Thiele test suite

Algorithms

COEA	competitive-cooperation co-evolutionary algorithm
CMOGA	cellular multi-objective genetic algorithm
EA	evolutionary algorithm
ε-MOEA	ε-dominance based MOEA
g-NSGA-II	g-dominance based non-dominated sorting genetic algorithm
GA	genetic algorithm
G-MOEA	guided multi-objective evolutionary algorithm
HypE	an algorithm for fast hypervolume-based many-objective optimization
iPICEA-g	interactive preference-inspired co-evolutionary algorithm using goal vectors

IBEA	indicator based evolutionary algorithm
MCDM	multi-criteria decision-making
MOEA	multi-objective evolutionary algorithm
MOEA/D	multi-objective evolutionary algorithm based on decomposition
MOGA	multi-objective genetic algorithm
MOGLS	multi-objective genetic local search
MSOPS	multiple single objective Pareto sampling
MOPSO	multi-objective particle swarm optimization
NPGA	niched Pareto genetic algorithm
NSGA	muilti-objective optimization using non-dominated sorting in genetic algorithms
NSGA-II	A fast and elitist multi-objective genetic algorithm
PAES	Pareto archived evolution strategy
PBEA	preference-based evolutionary algorithm by thiele et al.
PESA	Pareto envelope-based selection algorithm
PESA-II	Pareto envelope-based selection algorithm
PICEA-g	preference-inspired co-evolutionary algorithm using goal vectors
PICEA-w	preference-inspired co-evolutionary algorithm using weight vectors
PROMETHEE	preference ranking organisation method for enrichment evaluations
ELECTRE	elimination and choice expressing and reality
R-NSGA-II	reference point based evolutionary multi-objective algorithms by deb and sundar
r-NSGA-II	r-dominance based NSGA-II
SMS-EMOA	$\mathcal{S}$-metric selection based evolutionary multi-objective optimization algorithm
SPEA	strength Pareto evolutionary algorithm
SPEA2	improving strength Pareto evolutionary algorithm
VEGA	vector evaluated genetic algorithm

Symbols

$\mathbf{g}$	goal vector
$\mathbf{g}_{min}$	lower bound of goal vector
$\mathbf{g}_{max}$	upper bound of goal vector
g^{ws}	weighted sum scalarising function
g^{te}	weighted Chebyshev scalarising function
k	position parameter of WFG problem
l	distance parameter of WFG problem
n	number of decision variables
nr	replacement neighbour size used in MOEA/D
$\mathbf{s}$	a candidate solution
$\mathbf{w}$	a weight vector

x_i	the ith decision variable
x_{li}	lower bound on the ith decision variable
x_{ui}	upper bound on the ith decision variable
p_m	probability of applying PM
p_c	probability of applying SBX
v	genetic-variation process
$\mathbf{z}^*$	reference point in a scalarising function
$\mathbf{z}_R$	reference point in iPICEA-g
$\mathbf{z}^{\mathbf{ide}}$	*ideal* vector
$\mathbf{z}^{\mathbf{nad}}$	*nadir* vector

Fit_s	fitness of a candidate solution
Fit_g	fitness of a goal vector
$JointS$	joint candidate solutions
$JointG$	joint goal vectors
$JointW$	joint weight vectors
G	goal vector set
Gc	offspring of goal vector set
GD	General Distance
HV	Hypervolume metric
HV^*	True hypervolume value
M	number of objective, problem dimension
$maxGen$	maximum number of generations
N_p	number of preferences
N_g	number of goal vectors
Nsp	number of sampling points used in HypE
N_w	number of weight vectors
NFS	number of function evaluations
$P(t)$	population at iteration t
S	candidate solution set
Sc	offspring of candidate solution set
S_v	selection-for-survival process
S_s	selection-for-variation process
T	selection neighbour size used in MOEA/D
W	weight vector set
Wc	offspring of weight vector set
$\mathbb{R}^n$	field of real numbers

θ	Search range

Δ	Spread metric
η_c	SBX distribution parameter
η_m	PM distribution parameter
$\boldsymbol{\Omega}$	all examined solutions during the search

Chapter 1

Introduction

1.1 Overview

Optimisation problems arise in various disciplines such as engineering, economics, manufacturing, mathematics, chemistry, robotics, logistics, medicine, architecture, etc.. These problems are often multi-objective in nature, that is, two or more objectives are required to be optimised simultaneously. In a multi-objective problem (MOP), objectives are often in competition with one another. Thus, the optimal solution of MOPs is a set of trade-off solutions, rather than a single solution. Amongst these trade-off solutions, none of them is better in all objectives.

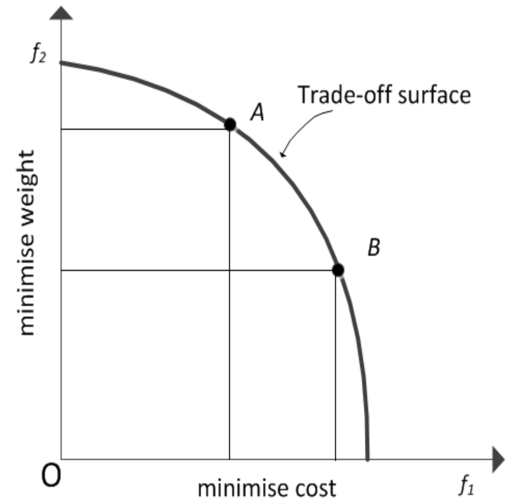


Figure 1.1: Illustration of a multi-objective problem.

As a general example, let us consider the task of purchasing a laptop. One always prefers a cheap and light laptop. That is, the first objective is to minimise the cost, and the second objective is to minimise the weight. However, these two objectives are typically contradictory, i.e. the lighter the laptop, the more the cost. Thus, it comes as no surprise that the optimal solution set is composed of a set of trade-off solutions as shown in Figure 1.1. For any two solutions of this set, e.g. A and B, neither A nor B is better in both objectives. A is cheaper but is heavier. B is lighter but costs more.

Without loss of generality, a minimisation MOP is defined as follows:

$$\begin{gathered} minimise \quad f_m(\mathbf{x}) \quad m = 1, 2, ..., M \\ subject\ to\ g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, ..., J \\ h_k(\mathbf{x}) = 0, \quad k = 1, 2, ..., K \\ x_{li} \leq x_i \leq x_{ui}, \quad i = 1, 2, ..., n \end{gathered} \tag{1.1}$$

A solution $\mathbf{x}$ is a vector of n decision variables: $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, $\mathbf{x} \in \mathbb{R}^n$. Each decision variable x_i is subject to a lower bound x_{li}, and an upper bound x_{ui}. f_m represents the m-th objective function. M is the number of objectives (generally, $M \geq 2$). J and K are the number of inequality and equality constraints, respectively. Feasible decision vectors refer to those decision vectors that satisfy all inequality and equality constraints. Some fundamental concepts in multi-objective optimisation are introduced as follows:

Definition 1.1 (**Pareto-dominance**): For two feasible decision vectors $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$ (denoted as $\mathbf{x} \preceq \mathbf{y}$) if and only if $\forall m \in 1, 2, \cdots, M$, $f_m(\mathbf{x}) \leq f_m(\mathbf{y})$ and $\exists m \in 1, 2, \cdots, M$, $f_m(\mathbf{x}) < f_m(\mathbf{y})$. Seen from Figure 1.2, we can find that $A \preceq B$, $A \preceq C$, $A \preceq D$ and $E \preceq A$. However, A and P, A and Q are non-dominated.

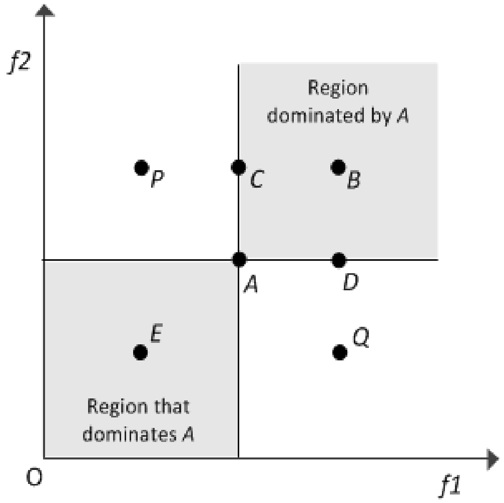


FIGURE 1.2: Illustration of Pareto-dominance relation.

Definition 1.2 (**Pareto optimality**): A solution $\mathbf{x} \in \mathbb{R}^n$ is said to be Pareto optimal in $\mathbb{R}^n$ if and only if $\nexists \mathbf{y} \in \mathbb{R}^n$, $\mathbf{y} \preceq \mathbf{x}$.

Definition 1.3 (**Pareto optimal set**): The Pareto optimal set (POS) is defined as the set of all Pareto optimal solutions, i.e. $POS = \{\mathbf{x} \in \mathbb{R}^n | \nexists \mathbf{y} \in \mathbb{R}^n, \mathbf{y} \preceq \mathbf{x}\}$.

Definition 1.4 (**Pareto optimal front**): The Pareto optimal front (POF) is defined as the set of all objective functions values corresponding to the solutions in POS, i.e., $POF = \{(f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_M(\mathbf{x})) : \mathbf{x} \in POS\}$.

In addition, a *nadir* point is a vector composed of all the worst (e.g., maximum for minimization problems) Pareto optimal objective values in a MOP. A *ideal* point is a vector composed of all the best (e.g., minimum for minimization problems) values of each objective (Miettinen, 1999, p.15).

Multi-objective evolutionary algorithms (MOEAs), inspired from the natural evolution, are well suited for solving multi-objective optimisation problems: their population based nature leads naturally to the generation of an approximate trade-off surface in a single run (Deb, 2001; Coello Coello et al., 2007); their robustness on problem characteristics enables them to be used on a wide range of application domains. Similar to evolutionary algorithms (EAs), in general a MOEA is composed of *genetic-variation* operator, *environment-selection* operator and *selection-for-variation* operator, which can be described as follows (Purshouse and Fleming, 2007):

$$P(t+1) = S_s(v(S_v(P(t))), P(t)) \tag{1.2}$$

where $P(t)$ are the candidate solutions (population) at iteration t , S_s is the *environment-selection* operator, v is the *genetic-variation* (recombination and mutation) operator, S_v is the *selection-for-variation* operator and $P(t+1)$ are the newly generated solutions. Thus, a set of candidate solutions is evolved by successively applying recombination, mutation, and selection to yield better solutions in an iterative process.

Evolutionary multi-objective optimisation (EMO) has enjoyed popularity in the last two decades. Research on EMO mainly focuses on issues such as the design of efficient algorithmic methods for the approximation of efficient solutions and the issue of measuring the quality of the approximated solutions. The IEEE Transactions on Evolutionary Computation by IEEE Press and Evolutionary Computation journal by MIT Press are two of the most important journals in the core evolutionary computation field. There are also some conferences for EMO. The bi-annual international Evolutionary Multi-Criterion Optimization conference series is devoted especially to EMO. Three other important evolutionary computation related conferences that also report EMO research developments are:

- IEEE Congress on Evolutionary Computation (CEC).
- Genetic and Evolutionary Computation Conference (GECCO).
- Parallel Problem Solving from Nature (PPSN).

1.2 Motivation

The central topic of this thesis is evolutionary multi-objective optimisation which links all the chapters. There are two motivations of this research: (i) to address many-objective optimisation problems; and (ii) to provide a unified approach for *a priori* or *progressive* evolutionary multi-objective optimisation and decision-making.

- **Many-objective optimisation**: Pareto-dominance based MOEAs were one of the earliest approaches for multi-objective optimisation. It is accepted that they perform well on MOPs with 2 and 3 objectives. However, studies have shown that their search capability often degrades significantly as the number of objectives increases (Purshouse and Fleming, 2003c; Hughes, 2005). This is because the proportion of Pareto-optimal (or non-dominated) objective vectors in the population becomes large when MOPs have more than 3 objectives, i.e. many-objective problems (MaOPs). That is, the ability of the Pareto-dominance relation in offering comparability between alternative solutions is reduced. As a result, insufficient selection pressure is generated toward the Pareto front. Unfortunately, MaOPs occur frequently in practice (Fleming et al., 2005; Coello Coello et al., 2007). In another sense, it is often easier to specify as many objectives as possible in the early stage of an optimisation process. Thus, it is desirable to develop more effective and efficient algorithms to handle MaOPs. This serves as the core motivation of this thesis.

  It is known that by incorporating decision-maker preferences we can potentially gain comparability between otherwise incomparable solutions (Miettinen, 1999). By holding multiple sets of hypothetical preferences simultaneously, we can obtain multiple comparison perspectives simultaneously, which might be sufficient to adequately describe the whole Pareto front. Existing approaches have tended to explore the use of some types of preferences. For example, Fonseca and Fleming (1998a) incorporated decision-maker preferences specified as goals and priorities into their early work MOGA (Fonseca and Fleming, 1993) (a Pareto-dominance based MOEA). The use of preferences enables MOGA to approximate a part of the Pareto optimal front even when the problem dimension is high. Hughes (2003) proposed a ranking method based on how each solution performed against a set of pre-defined preferences expressed as weighted min-max formulations. However, none of the studies has attempted to answer the question below

  - how can a set of suitable preferences be maintained adaptively in order to be employed to obtain a good approximation of the Pareto optimal front?

These observations motivate us to consider the type of preferences and how these preferences can be used. In this research, two of the most commonly used preferences, goals and weights are studied. The usefulness of preferences is maintained by co-evolving them with candidate solutions during the search.

- **a unified approach for preference-based MOEAs**[1]: Having obtained a good approximation of the Pareto optimal front, the MOP remains unsolved as a decision-maker (DM) has to incorporate his/her[2] preferences, and selects the most preferred solution. In other words, the ultimate goal of solving a MOP is not to obtain a good approximation of the Pareto optimal front but it is to help a DM to consider multiple objectives simultaneously and identify a single Pareto optimal solution that satisfies him/her, i.e. assisting multi-criteria decision making (MCDM).

  However, the ultimate goal of solving a MOP is not to approximate the entire Pareto optimal front but to help a decision maker (DM) to consider the multiple objectives simultaneously and to identify a single Pareto optimal solution which is satisfactory. Therefore, in order to select a single solution to a MOP, it is necessary to resolve the trade-off between competing solutions by introducing subjective preferences that a DM has for varying levels of performance across the objectives.

  Incorporation of the DM preferences is an important part for a real-world decision support system. However, current methods developed for preference-based multi-objective optimisation are unable to handle, comprehensively, the range of ways in which a DM likes to articulate his preferences. Moreover, in some cases it is easier for a DM to express his preferences by aspiration levels and in others by weights or other forms. These observations motivate us to develop a unified approach which enables the decision-maker to articulate multiple preference types.

1.3 Outline of the thesis

There are seven chapters in this thesis, beginning with this Introduction. A literature review Chapter follows. Chapters 3 and 4 introduce two novel algorithms for multi-objective optimisation, in particular, many-objective optimisation. Chapter 5 analyses parameter sensitivities of the proposed algorithms. Chapter 6 proposes a unified approach for *a priori* and *progressive* multi-objective optimisation and decision-making. Chapter 7 concludes. The content of each of these chapters is outlined below.

- Chapter 2 provides a comprehensive review of current literature relevant to this research. First, a historical overview of MOEAs is provided, followed by an introduction to multi-objective benchmark problems and performance metrics. Then,

[1] MOEAs that incorporate decision-maker preference *a priori* or *progressively* is termed as preference-based MOEAs.

[2] For brevity we use he instead of he/she, and his instead of his/her in this thesis.

general issues on many-objective optimisation are introduced. Following this, a detailed review of the existing methods developed for solving many-objective problems is provided. Finally, multi-objective evolutionary algorithms developed for *a priori* and *progressive* decision-making are critically reviewed.

- Chapter 3 proposes a novel algorithm, called a preference-inspired co-evolutionary algorithm using goal vectors (PICEA-g) for MOPs (especially for MaOPs). This Chapter begins with an introduction to the concept of a preference-inspired co-evolutionary algorithm. Secondly, as a realisation of this concept, PICEA-g is proposed in which candidate solutions are co-evolved with a set of goal vectors. Thirdly, the performance of PICEA-g is empirically assessed by comparing it with four of the best-in-class MOEAs on multi-objective benchmark problems up to 10 objectives. Following that, some further discussions such as the effect of co-evolution and the performance of goal vectors are provided. In the last section of this Chapter, an effective strategy named the *cutting* plane is proposed to further improve the performance of PICEA-g.

- Chapter 4 proposes another realisation of the preference-inspired co-evolutionary algorithm, denoted as PICEA-w, where candidate solutions are co-evolved with weight vectors. The use of co-evolution enables PICEA-w to effectively handle the difficulties encountered by decomposition based methods. This Chapter first introduces the basics and general issues in decomposition based methods. Following that, PICEA-w is described in detail. Thirdly, the performance of PICEA-w is assessed on a set of problems with different Pareto front geometries. Following that, some further discussions such as the performance of weight vectors are provided.

- Chapter 5 analyses the sensitivities of parameters in PICEAs. Parameter sensitivity analysis is helpful to gain insight into the robustness of an algorithm and to provide suggestions with respect to the parameter settings for non-expert users. First, a global sensitivity method is used to analyse the sensitivities of all parameters simultaneously, identifying the key parameters of PICEAs. Following that, further analysis is conducted on some key parameters. Finally, the settings of genetic operator parameters are studied.

- Chapter 6 proposes a novel approach, namely, iPICEA-g for *a priori* and *progressive* evolutionary multi-objective optimisation and decision-making. iPICEA-g is derived from PICEA-g. It co-evolves candidate solutions with goal vectors that are exclusively generated in the regions of interest. This Chapter first introduces the preference formulation used in iPICEA-g and the rationale of iPICEA-g. Secondly, the performance of iPICEA-g is assessed on a set of benchmark problems. Finally, iPICEA-g is applied to solve a real-world multi-objective problem– aircraft control system design.

- Chapter 7 concludes the thesis. First, the main contributions of this thesis are summarised in depth. Second, the future work and some open questions in multi-objective optimisation are identified.

1.4 Contributions

The main contributions of this thesis are:

- - **A novel multi-objective evolutionary algorithm, PICEA-g for many-objective optimisation.** PICEA-g is based on the concept of co-evolution of candidate solutions and preferences. It co-evolves candidate solutions with preferences specified as goal vectors. A systematic empirical study is conducted to assess the performance of PICEA-g. Experimental results show that PICEA-g outperforms the four best-in-class MOEAs (NSGA-II, ϵ-MOEA, HypE and MOEA/D) on many-objective WFG problems.
  - **The use of the *cutting plane* to further enhance PICEA-g performance.** PICEA-g is presented using an effective strategy named the *cutting plane*. The *cutting plane* works with the assumption of knowing the *ideal* point beforehand. Using this strategy, suitable goal vector bounds can be determined adaptively along the search process. By co-evolving goal vectors generated within different goal vector bounds, PICEA-g can then adaptively distribute appropriate search effort towards different objectives, leading a better performance. Experimental results conducted on bi- and 7-objective problems have demonstrated the effectiveness of this strategy.

  Related publications:

  - Wang, R., Purshouse, R. C., Fleming, P. J., Preference-inspired co-evolutionary algorithms for many-objective optimisation, IEEE Transactions on Evolutionary Computation., 17 (4), 474-494, 2013.
  - Wang, R., Purshouse, R. C., Fleming, P. J., Preference-inspired co-evolutionary algorithm using adaptively generated goal vectors, Evolutionary Computation (CEC), 2013 IEEE Congress on. IEEE, Cancun, Mexico, 2013: 916- 923.
  - Wang, R., Purshouse, R. C., Fleming, P. J., On Finding Well-Spread Pareto Optimal Solutions by Preference-inspired Co-evolutionary Algorithm, Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference (GECCO 2013). ACM, Amsterdam, The Netherlands, 2013: 695-702.

- **A preference-inspired co-evolutionary algorithm using weights (PICEA-w).** PICEA-w is a new decomposition based algorithm that eliminates the need to specify appropriate weights in advance of performing the optimisation. Adopting the concept of PICEA, in PICEA-w weights are adaptively modified by being co-evolved with candidate solutions during the search process. The co-evolution enables suitable weights to be adaptively constructed along the search process and thus guiding candidate solutions towards the Pareto optimal front more effectively and efficiently. Through rigorous empirical testing, we demonstrate that compared to other leading decomposition-based algorithms, PICEA-w is less sensitive to the Pareto front geometry of a problem and performs better on many-objective problems.

Related publications:

  - Wang, R., Purshouse, R. C., Fleming, P. J., Preference-inspired co-evolutionary algorithms using weight vectors for many-objective optimisation, Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference (GECCO 2013), ACM, Amsterdam, The Netherlands, 2013, pp. 101-102.
  - Wang, R., Purshouse, R. C., Fleming, P. J., Preference-inspired co-evolutionary algorithms using weight vectors, European Journal of Operational Research. (Under review)

- **A novel and unified method, iPICEA-g for *a priori* and *progressive* multi-objective optimisation and decision making.** iPICEA-g is the first approach which can cater simultaneously to different types of decision-maker preferences, and that is also able to support multiple regions of interest. In particular, the use of the brushing technique is the first time, to the best of the author's knowledge, that the decision-maker is able to articulate his/her preferences without using numerical values but simply by drawing on the Cartesian or Parallel coordinates in objective-space. As no direct elicitation of numbers is required, the cognitive burden on the decision-maker is reduced. The iPICEA-g is demonstrated to perform well on a set of benchmark problems as well as a real world application.

  Related publications:

  - Wang, R., Purshouse, R. C., Fleming, P. J., whatever works best for you-a new method for *a priori* and *progressive* multi-objective optimisation, in: Evolutionary Multi-Criterion Optimization, Springer, 2013, pp. 337-351.
  - Wang, R., Purshouse, R. C., Fleming, P. J., On finding preferred solutions by preference-inspired co-evolutionary algorithms, European Journal of Operational Research. (Under review)

- **A parameter sensitivity analysis of PICEA-g and PICEA-w.** An empirical study is conducted to analyse sensitivities of parameters used in both the PICEAs. Some valuable suggestions on parameter settings are provided, which facilitates the use of PICEAs for non-experts.

  - Wang, R., Purshouse, R. C., Fleming, P. J., Preference-inspired co-evolutionary show robustness on parameter settings, International Journal of System Science. (Under review)

Some additional contributions resulted during the development of this thesis. For the coherency of the thesis, these contributions are not explicitly documented within the monograph itself but are briefly described as follows.

- **A general framework for localised multi-objective evolutionary algorithms.** This framework seeks to improve both the accuracy and the diversity of these solutions through the local application of evolutionary operators to selected sub-populations. A local operation-based implementation framework is presented in which a population is partitioned, using hierarchical clustering, into a predefined number of sub-populations. Environment-selection and genetic-variation are then applied to each sub-population. The effectiveness of this approach is demonstrated on 2- and 4-objective benchmark problems. The performance of each of four best-in-class MOEAs is compared with their modified local operation-based versions derived from this framework. In each case the introduction of the local operation-based approach improves performance. Further, it is shown that the combined use of local environment-selection and local genetic-variation is better than the application of either local environment-selection or local genetic-variation alone. This work has been published as

  - Wang, R., Fleming, P. J., Purshouse, R. C., General framework for localised multi-objective evolutionary algorithms, Information sciences, 258(2), 29-53.
  - Wang, R., Purshouse, R. C., Fleming, P. J., Local preference-inspired co-evolutionary algorithms, in: GECCO 2012: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, Philadelphia, USA, 2012, pp. 513-520.

- **Enhancement of the MOGA Matlab toolbox of the University of Sheffield.** The toolbox was first created by Andrew J. Chipperfield, Carlos M. Fonseca and Peter J. Fleming, then was enhanced by Robin C. Purshouse. During this PhD research, some new components are developed for the toolbox. For example, a set of differential evolution operators for doing recombination and mutation; the ε-dominance archiving strategy for obtaining evenly distributed solutions; some functions for generating randomly/evenly distributed points; a function to measure the neighbourhood. These components will further facilitate the design of MOEAs.

Chapter 2

Literature review

Many real-world problems have multiple conflicting objectives that are required to be optimised simultaneously. A variety of methods have been developed in the literature for solving such optimisation problems. Prior to introducing our methods, it is important to review the current literature relevant to this thesis. This review discusses the limitations of the current state-of-the-art algorithms and provides motivations for this research.

This Chapter starts with an historical overview of MOEA, where co-evolutionary algorithms are also introduced (Section 2.1). Secondly, in Section 2.2, three multi-objective test suites are introduced, followed by an introduction to performance metrics in Section 2.3. These test problems and performance metrics are used as means of comparing the performance of MOEAs. Section 2.4 provides a comprehensive review of many-objective optimisation, including the issues for MaOPs as well as the existing methods proposed for many-objective optimisation. Lastly, multi-objective evolutionary algorithms developed for *a priori* and *progressive* decision-making are critically reviewed in Section 2.5.

2.1 Historical overview of MOEAs

Since the 1950s the operational research community has proposed approaches for solving MOPs. Typically, these approaches convert a MOP into a single objective optimisation problem by using some user-defined procedures. A variety of such approaches are available in specialised literature such as Miettinen (1999) and Ehrgott (2005). Some representatives approaches are the weighted sum, weighted metric, ε-constraint, goal programming (e.g., weighted goal programming, Lexicography goal programming and Min-Max programming). These approaches have the following limitations:

i) Only one Pareto optimal solution can be expected be found in an algorithm run. Thus, a number of runs (with different parameter settings) are required in order to obtain a set of Pareto optimal solutions.

ii) These approaches are susceptible to problem characteristics. For example, the weighted sum approach fails to find Pareto optimal solutions in non-convex Pareto regions.

iii) These approaches require some problem knowledge or additional parameter settings. For example, the weight vector used in the weighted sum and weighted metric approaches, the parameter ε used in the ε-constraint approach and the target vector used in the goal programming approach.

These limitations greatly inhibit the applications of the traditional approaches and drive us to develop some more effective methods. Multi-objective evolutionary algorithms have generated considerable interest and acceptance. Their population based nature is helpful to generate a set of trade-off solutions in a single run. Moreover, MOEAs tend to be robust to underlying cost function characteristics (Deb, 2001; Coello Coello et al., 2007). Over the last twenty years, many different MOEA approaches were proposed. The following sections provide a historical overview of these approaches based on the study of Coello Coello (2006).

2.1.1 MOEAs from 1985 to 1998

Vector Evaluated Genetic Algorithm (VEGA) (Schaffer, 1985) is often recognised as the first MOEA. VEGA is developed by modifying the selection mechanism of a basic genetic algorithm (GA). At each generation, proportional selection is performed to select a sub-population based on each objective function in turn. These sub-populations would be pooled together to form a new population. The genetic operations, i.e. crossover and mutation, are then performed on the new population in the usual way.

Later, Goldberg (1989) suggested two fundamental concepts for multi-objective optimisation: the Pareto-dominance relation, selecting non-dominated solutions, and the niching technique, maintaining diversified solutions along the non-dominated frontier. These two concepts opened a new avenue and generated an overwhelming interest in MOEAs. Fonseca and Fleming (1993) proposed, to the best of the author's knowledge, the first Pareto-dominance based MOEA, i.e. Multi-Objective Genetic Algorithm (MOGA). After that many other MOEAs that use Goldberg's idea were implemented. Two representative algorithms are Non-dominated Sorting Genetic Algorithm (NSGA (Srinivas and Deb, 1994)) and Niched-Pareto Genetic Algorithm (NPGA (Horn et al., 1994)). Among the three algorithms, MOGA is the most effective approach, followed by NPGA, then NSGA and VEGA (Zitzler and Thiele, 1999).

In addition to the mentioned MOEAs, Tanaka and Tanino (1992) proposed the first mechanism to incorporate decision-maker preferences into MOEAs. Fonseca and Fleming (1995b) provided the first survey of MOEAs, and introduced the first performance metric, i.e. attainment surface (Fonseca and Fleming, 1996). Moreover, Fonseca and Fleming (1998a) proposed one of the earliest preference-based MOEA where the decision-maker preferences are specified as goals and priorities and can be interactively incorporated into MOGA so as to guide solutions towards a region of interest. However, in this period few studies were carried out on the development of performance metrics and benchmark problems. Comparisons were visual in most cases and most problems tackled were bi-objective. The main achievements in this period are summarised in Figure 2.1.

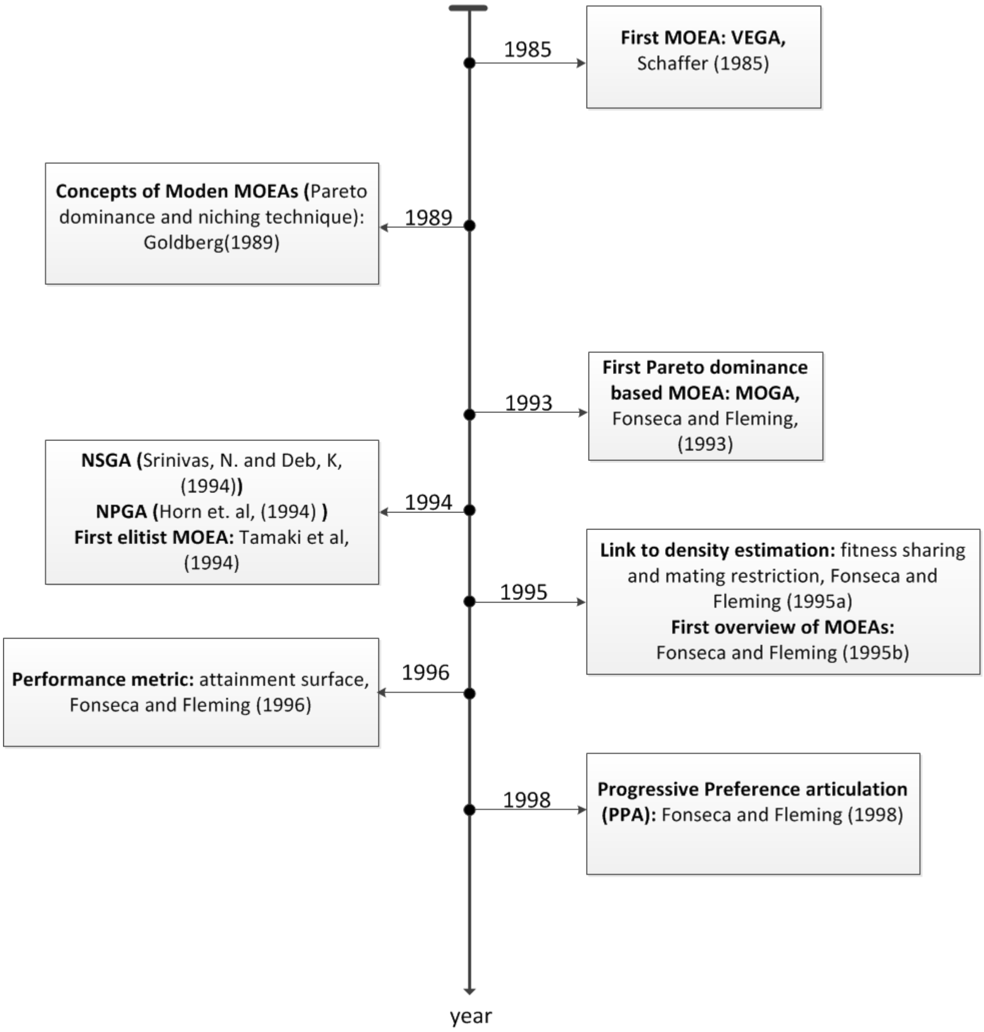

1985
First MOEA: VEGA, Schaffer (1985)
1989
Concepts of Moden MOEAs (Pareto dominance and niching technique): Goldberg(1989)
1993
First Pareto dominance based MOEA: MOGA, Fonseca and Fleming, (1993)
1994
NSGA (Srinivas, N. and Deb, K, (1994))
NPGA (Horn et. al, (1994))
First elitist MOEA: Tamaki et al, (1994)
1995
Link to density estimation: fitness sharing and mating restriction, Fonseca and Fleming (1995a)
First overview of MOEAs: Fonseca and Fleming (1995b)
1996
Performance metric: attainment surface, Fonseca and Fleming (1996)
1998
Progressive Preference articulation (PPA): Fonseca and Fleming (1998)
year

Figure 2.1: MOEAs from 1985 to 1998.

2.1.2 MOEAs from 1999 to 2003

The second generation of MOEAs starts when the elitism strategy gets popularised (Coello Coello, 2006). Specifically, after the publication of Strength Pareto Evolutionary

Algorithm (SPEA) (Zitzler and Thiele, 1999) many researchers started to incorporate elitism into the design of MOEAs. Elitism became the third common practice (the first two are Pareto-dominance and diversity maintenance) in MOEAs. Typically, elitism is implemented as an external archive that stores all non-dominated solutions found during the search. It can also be induced by a $(\mu + \lambda)$-selection framework, that is, parents and their offspring are first pooled together, then new parents are selected based on the pooled population.

In this period (from 1999 to 2003) researchers mainly focused on further improving the effectiveness and efficiency of MOEAs. Some representative algorithms include SPEA, SPEA2 (Zitzler et al., 2002), Pareto Archived Evolution Strategy (PAES (Knowles and Corne, 1999, 2000)), Pareto Envelope-based Selection Algorithm (PESA (Corne et al., 2000)) and its improved version PESA-II (Corne et al., 2001) and Non-dominated Sorting Genetic Algorithm II (NSGA-II (Deb et al., 2002a)). In SPEA the fitness of solutions is determined by the *Pareto strength value* (Zitzler and Thiele, 1999); a clustering method is employed to maintain the diversity of solutions. SPEA2 is an improved version of SPEA where an enhanced fitness assignment and clustering method is used. PAES, PESA and PESA-II apply a hyper-grid based scheme to conduct selection and diversity maintenance.

NSGA-II probably is the most widely used MOEA (9767 citations accessed on 09/10/2013). It is implemented within a $(\mu + \lambda)$ framework. Solutions are selected based on their dominance level and crowding distance. Specifically, the non-dominated sorting strategy is applied to rank all individuals. The assigned rank is used as a primary criterion in the binary tournament selection for parent selection. If more than μ solutions are non-dominated, solutions with the same rank are further evaluated by their crowding distance. Roughly speaking, solutions in a sparse region are preferred to those in a crowded region when they have the same rank.

In addition to these studies, another concept, ε-dominance, was proposed by Laumanns et al. (2002). It was demonstrated that the use of ε-dominance is helpful in finding solutions with good proximity and diversity (Deb et al., 2003, 2005). Several benchmark test suites and performance metrics were proposed which facilitates the assessment and comparison of MOEAs. The main achievements over this period are summarised in Figure 2.2.

MOEAs gained great developments during this period: the Pareto-dominance relation, diversity maintenance and elitism mechanism become three of the most common modules of MOEAs. MOEAs start to be applied in many application domains such as electrical engineering, aeronautical engineering, robotics control, scheduling, etc. (Deb, 2001; Fleming et al., 2005). However, with the increasing applications of MOEAs, researchers subsequently found that Pareto-dominance based MOEAs, e.g. MOGA, NSGA-II and

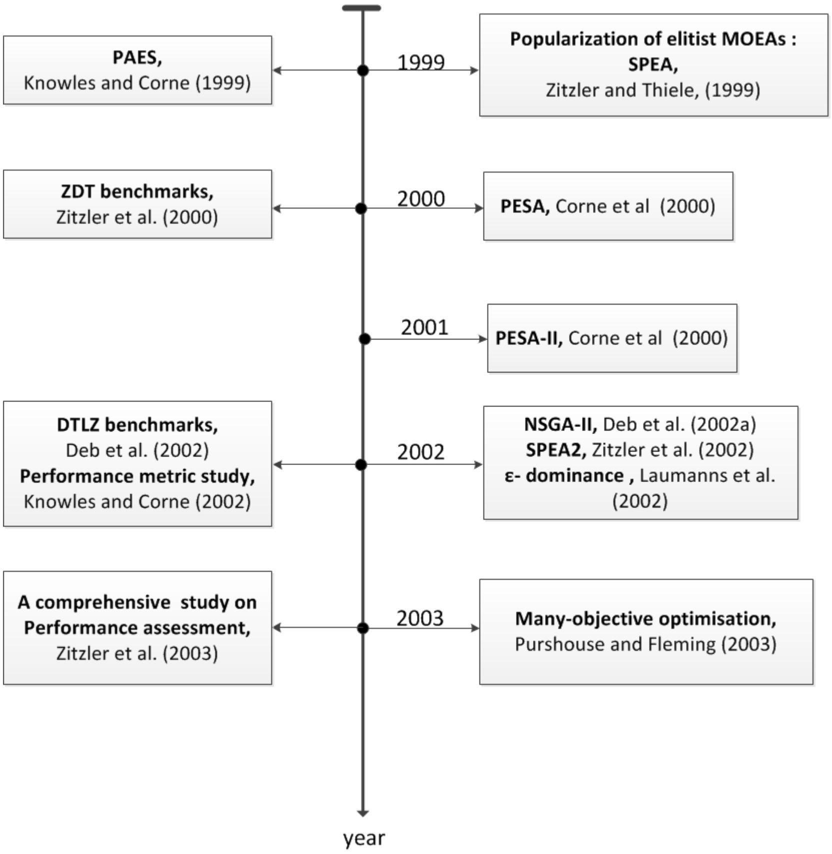

PAES,
Knowles and Corne (1999)
1999
Popularization of elitist MOEAs :
SPEA,
Zitzler and Thiele, (1999)
ZDT benchmarks,
Zitzler et al. (2000)
2000
PESA, Corne et al (2000)
2001
PESA-II, Corne et al (2000)
DTLZ benchmarks,
Deb et al. (2002)
Performance metric study,
Knowles and Corne (2002)
2002
NSGA-II, Deb et al. (2002a)
SPEA2, Zitzler et al. (2002)
ε- dominance , Laumanns et al. (2002)
A comprehensive study on
Performance assessment,
Zitzler et al. (2003)
2003
Many-objective optimisation,
Purshouse and Fleming (2003)
year

FIGURE 2.2: MOEAs from 1999 to 2003.

SPEA2, face difficulties on MOPs with more than three objectives, termed as many-objective (optimisation) problems (MaOPs). Purshouse and Fleming (2003c) first identified this issue and found that the search ability of Pareto-dominance based approaches degrades significantly as the number of objectives increases.

MaOPs arise in many applications, thus developing effective algorithms for MaOPs has become a hot topic. Additionally, researchers realised that often a decision-maker is not interested in obtaining the entire Pareto front but only a part of the Pareto front. Thus, some preference-based MOEAs were developed to incorporate decision-maker preferences into MOEAs. This thesis focuses on both the topics: many-objective optimisation and preference-based MOEAs. Relevant studies to these topics are critically reviewed in the next sections.

2.1.3 An introduction to co-evolutionary algorithms

Co-evolution refers to a process that simultaneously evolves different species. In co-evolution, the fitness of a solution in one species is measured based on its cooperative or competitive interactions with others (Goh et al., 2010). In cooperative co-evolution, different species evolve in a mutually beneficial relationship. In competitive co-evolution, different species have opposing interests and the success of one species depends on the failure of others.

A number of co-evolution based MOEAs have been proposed. These are briefly reviewed next. Following that, possible issues of implementation of co-evolution based algorithms are introduced.

Cooperative co-evolutionary algorithms

In cooperative co-evolution, an individual is formed by combining sub-individuals from different species (sub-populations). Sub-individuals are ranked based on the fitness of the combined individual. To better understand cooperative co-evolution based algorithms, we briefly describe some representative studies.

Potter and Jong (2000) proposed a cooperative co-evolution based genetic algorithm (CCGA). In CCGA a sub-population is initialized for each decision variable. Each individual sub-population is evolved using a traditional GA. Two versions of CCGA (i.e., CCGA1 and CCGA2) are implemented. The difference between the two versions is as follows. In CCGA1 a sub-individual is evaluated by combining it with the best known sub-individual from the other sub-populations. In CCGA2 a sub-individual is evaluated by combining it either with a random or the best sub-individual from the other sub-populations. The best fitness of the two individuals is assigned as the fitness of the current sub-individual.

However, CCGA only applies to single-objective optimisation. Keerativuttitumrong et al. (2002) proposed a multi-objective cooperative co-evolutionary genetic algorithm (MOCCGA) based on CCGA2. The only difference is that the multi-objective optimizer, MOGA, is applied to optimise each sub-population. In a similar way, Maneeratana et al. (2006) applied this cooperative co-evolution concept to NPGA and NSGA, and Iorio and Li (2004) applied this concept to NSGA-II. All the extended versions are reported to exhibit significant improvements over their canonical counterparts.

Coello Coello and Sierra (2003) proposed an alternative co-evolutionary MOEA called CO-MOEA. In their approach, some promising search spaces are identified during the search, and more search effort is distributed on these regions. Specifically, the search space is divided into some pre-defined sub-regions. The same number of individuals is initialised for each sub-region. The number of individuals in each sub-region is modified in proportion to the number of new non-dominated solutions discovered in the optimisation process. CO-MOEA is demonstrated to outperform PAES and NSGA-II on 2 and 3-objective problems. However, obviously the performance of CO-MOEA depends on the choice of sub-regions.

Competitive co-evolutionary algorithms

In competitive co-evolution, an inverse fitness interaction exists between different species, and different species compete to survive.

Laumanns et al. (1998) were the first to incorporate the concept of competitive co-evolution into MOEAs. Their proposed algorithm is based on the predator-prey model. Candidate solutions (acting as preys) are placed on an undirected and connected graph. Objective functions (act as predators) are selected at random to evaluate all the candidate solutions. In the neighbourhood of an objective function, the candidate solution that has the worst objective value is replaced by its offspring.

In Barbosa and Barreto (2001), another competitive co-evolutionary algorithm was proposed for solving a graph layout problem. In their approach, two populations are co-evolved – a graph layout population (candidate solutions) and a set of weight vectors. The two populations are evolved by a separate genetic algorithm in a round robin process. First candidate solutions are evolved for several generations while the weight vectors are kept constant. Then a decision-maker is asked to select his/her preferred solutions. These selected candidate solutions are then held constant while the weight vectors are evolved in an attempt to produce suitable weights that can reveal the decision-maker's preference.

Lohn et al. (2002) proposed an interesting competitive co-evolutionary algorithm called CGA, in which candidate solutions are co-evolved with target vectors. In CGA, the targets gain higher fitness by being satisfied by fewer candidate solutions, and the candidate solutions gain higher fitness by meeting as many targets as possible. Experimental results show that CGA performs better than SPEA and NSGA on 2-objective problems. However, Goh and Tan (2009) reported that CGA might be sensitive to the Pareto front geometries.

Combined cooperative and competitive co-evolutionary algorithms

There are also some algorithms that incorporate both competitive and cooperative co-evolutions, for example, the competitive-cooperation coevolutionary algorithm (COEA) Goh and Tan (2009). In COEA, sub-individuals are co-evolved both competitively and cooperatively. In the competitive mode, sub-individuals compete with each other to become a representative of a particular sub-population. In the cooperative mode, representatives from each sub-population co-evolve to generate better individuals. COEA is tested on three benchmark problems. Experimental results indicate that COEA performs well on problems with severe parameter interactions. A similar study is conducted by Goh et al. (2010) where this concept is applied to a multi-objective particle swarm optimisation (MOPSO) algorithm.

Possible issues of co-evolutionary algorithms

Co-evolution based algorithms potentially face the following issues the Red Queen Effect, cycling and disengagement (De Jong and Pollack, 2004; Bongard and Lipson, 2005).

(i) The Red Queen Effect means that the subjective fitness of the populations improves during the course of evolution but there is no consistent progress along the objective metric. Conversely, there is progress along an objective fitness but the improvement cannot be reflected by their subjective fitness. For example, consider students' scores in examinations and their true ability. A series of difficult examinations may result in a low score (subjective fitness) while in fact their ability is improved. Alternatively, students may get high scores in a series of easy examinations but such scores do not represent their abilities.

(ii) Cycling refers to a scenario that solution **a** is better than solution **b** in one subjective metric, and simultaneously, **a** is worse than **b** in another subjective metric. Cycling might occur when the subjective fitness criteria keeps changing over time. Cycling might arise in the situation that solutions keep searching the same space.

(iii) Disengagement refers to a loss of fitness gradient in the process of co-evolution. That is, one population is completely better than another population; the subjective fitness of solutions becomes constant in such a way that selection becomes random. Disengagement often leads to over-specialization, that is, candidate solutions only progress on some objectives.

Overall, although co-evolution is an interesting and promising concept, and some useful algorithms have been developed based on this concept, a particular attention should be paid to the potential for pathologies when implementing a co-evolution based algorithm.

2.2 Multi-objective test problems

A number of multi-objective test problems have been proposed to benchmark the performance of MOEAs (Huband et al., 2006). This section reviews three of the most widely used benchmarks. Detailed descriptions of the objective functions and the Pareto optimal fronts (for 2- and 3-objective problems) are provided in Appendix A.

Prior to describing these problems, we introduce the following four terminologies. "Multimodal" means there are multiple local optimal Pareto fronts. "Separable" means that the objective can be optimised by considering each parameter in turn. The global optimal solution is the cross-product of each of the optimal parameters. specifically, in a multi-objective case, the ideal points for separable objectives can be determined by considering only one parameter at a time. “Bias” means that an evenly distributed set of Pareto optimal solutions in the search space does not map to an evenly distributed set of objective vectors in fitness space[1]. “Deceptive” means a problem has at least two optima: a true optima and a deceptive optima. According to Deb (1999), deceptive functions are those in which low order building blocks do not combine to form higher-order building blocks: instead they form building blocks for a suboptimal solution.

2.2.1 ZDT test suite

Zitzler et al. (2000) proposed the first MOP test suite, the ZDT test suite. It contains six bi-objective minimisation problems. Characteristics of the six test problems are shown in Table 2.1.

Table 2.1: Properties of the ZDT test problems.

	Objective	Separable	Multi-modal	Pareto optimal front
ZDT1	f_1 f_2	Yes Yes	No No	Convex
ZDT2	f_1 f_2	Yes Yes	No No	Concave
ZDT3	f_1 f_2	Yes Yes	No Yes	Convex disconnected
ZDT4	f_1 f_2	Yes Yes	No Yes	Convex
ZDT5	f_1 f_2	Yes Yes	Yes Yes	Convex
ZDT6	f_1 f_2	Yes Yes	No Yes	Concave

All the problems are real-valued except for ZDT5 which is binary encoded. The decision variable x_i for all the real-valued problems is within $[0, 1]$ with the exception of ZDT4 where the $x_1 \in [0, 1]$ and all the other $x_i \in [-5, 5]$. All the problems are separable, that is, the Pareto optimal set can be achieved by optimising each decision variable separately. ZDT1, ZDT2, ZDT3 and ZDT6 problems have only one global optimal front and are not multi-modal while ZDT4 is multi-modal, having 2^{19} local Pareto optimal fronts. The Pareto optimal front of ZDT1 is convex, ZDT3 is convex and disconnected. All the other problems have a concave Pareto optimal front. ZDT5 describes a deceptive problem. However, as mentioned above, it is binary encoded. Moreover, the search space of ZDT6 is non-uniform and thus is more difficult compared with ZDT2.

[1] Note that bias is also called non-uniformity in some studies

2.2.2 DTLZ test suite

The DTLZ test suite was proposed by Deb et al. (2002b). Unlike the ZDT test suite, the DTLZ test problems are scalable to any number of objectives. Properties of these problems are described in Table 2.2.

TABLE 2.2: Properties of the DTLZ test problems.

	Objective	Separable	Multi-modal	Pareto optimal front
DTLZ1	$f_{1:M}$	Yes	Yes	Linear
DTLZ2	$f_{1:M}$	Yes	No	Concave
DTLZ3	$f_{1:M}$	Yes	Yes	Concave
DTLZ4	$f_{1:M}$	Yes	No	Concave
DTLZ5	$f_{1:M}$	Yes	No	a concave curve
DTLZ6	$f_{1:M}$	Yes	No	a concave curve
DTLZ7	$f_{1:M-1}$ f_M	No Yes	No Yes	Disconnected

For all DTLZ test problems, the domain of decision variables is within $[0, 1]$, and test problems are scalable with respect the number of decision variables n. For problems DTLZ1 to DTLZ6, $n = k+l$, where k is the number of distance parameters and $l = M-1$ is the position parameters. For DTLZ7, n is suggested to be set as $10M$. DTLZ1 has $115^k - 1$ local optimal fronts and can be used to investigate an MOEA's ability to converge to a hyper-plane; typically, DTLZ2, DTLZ3 and DTLZ4 are used to investigate an MOEA's ability to scale up its performance for a large number of objective, obtaining a set of well spread solutions that are also close to the Pareto optimal front. DTLZ5 and DTLZ6 are used to test an MOEA's ability to converge to a degenerated curve. DTLZ7 has $2^{(M-1)}$ disconnected Pareto-optimal regions in the search space and therefore is used to test an MOEA's ability to maintain sub-population in different Pareto optimal regions.

2.2.3 WFG test suite

The WFG test suite is proposed by Huband et al. (2006) which contains 9 test problems. These problems are all scalable to any number of objectives. Table 2.3 summarises the properties of these problems.

Compared with the ZDT and DTLZ test suites, the WFG test suite is more challenging as problem attributes include separability/non-separability, unimodality/multimodality, unbiased/biased parameters and convex/concave geometries. A solution vector $\mathbf{x}$ contains k position parameters and l distance parameters, that is, the number of decision variables, $n = k + l$. k should be divisible by $M - 1$ and l can be set to any positive integer, except for WFG2 and WFG3 where it must be a multiple of 2. The domain of x_i is $[0, 2i]$. All the ZDT and DTLZ problems are separable (Huband et al., 2006), however,

Table 2.3: Properties of the WFG test problems.

	Objective	Separable	Multi-modal	Bias	Pareto optimal front
WFG1	$f_{1:M}$	Yes	No	Polynomial flat	convex, mixed
WFG2	$f_{1:M-1}$ f_M	No No	No Yes		convex, disconnected
WFG3	$f_{1:M}$	No	No		linear degenerate
WFG4	$f_{1:M}$	Yes	Yes		concave
WFG5	$f_{1:M}$	Yes	Yes		concave
WFG6	$f_{1:M}$	No	No		concave
WFG7	$f_{1:M}$	Yes	No	parameter dependent	concave
WFG8	$f_{1:M}$	No	No	parameter dependent	concave
WFG9	$f_{1:M}$	No	Yes	parameter dependent	concave

the WFG2, WFG3, WFG6, WFG8 and WFG9 problems are all non-separable. With respect to multi-modality, problems WFG2, WFG4, WFG5 and WFG9 are multi-modal. None of the ZDT and DTLZ test problems contains bias. However, problems WFG1, WFG7, WFG8 and WFG9 all feature significant bias. Other than ZDT5, none of the ZDT or DTLZ test problems is deceptive. However, in the WFG test suite both WFG5 and WFG9 are deceptive. Additionally, all the WFG test problems have dissimilar trade-off ranges.

2.3 Performance metrics

Apart from the test problems, to measure the performance of MOEAs we still need some performance metrics. There are two distinct goals for multi-objective optimisation and these are shown in Figure 2.3:

- find solutions as close to the Pareto optimal front as possible, i.e. convergence;
- distribute solutions evenly along the Pareto optimal front as widely as possible, i.e. diversity.

Therefore, to evaluate the performance of MOEAs (the accuracy of the obtained approximation of the Pareto optimal front), both the convergence and diversity performance need to be assessed. According to the literature (Knowles and Corne, 2002; Zitzler et al., 2003), a number of performance metrics have been proposed to either separately measure convergence and diversity performance explicitly, or measure both convergence and diversity in an implicit manner.

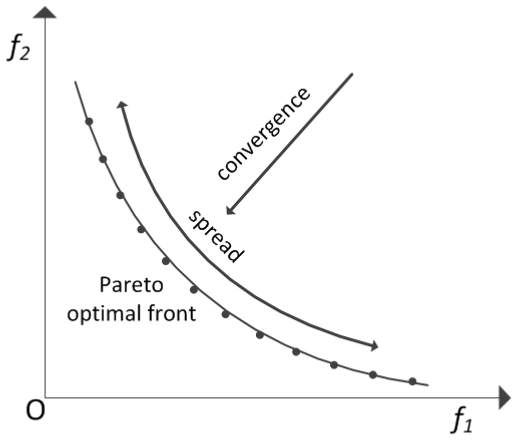


FIGURE 2.3: Illustration of two goals of multi-objective optimisation.

2.3.1 Convergence metrics

A convergence metric measures the closeness of solutions in the approximated set S to the Pareto optimal front. Typically, the Pareto optimal front is described by a reference set of representative Pareto optimal solutions, denoted as P^*. A number of metrics have been proposed such as, error ratio (Van Veldhuizen, 1999), generational distance (Van Veldhuizen, 1999) and $\mathcal{C}$ metric (Zitzler and Thiele, 1999). Note that $|S|$ denotes the cardinality of set S.

Error ratio: ER

The ER metric counts the number of solutions in S that are not members of P^*. The metric is defined by Equation 2.1:

$$ER = \frac{\sum_{i=1}^{|S|} e_i}{|S|} \tag{2.1}$$

where

$$e_i = \begin{cases} 0 & d_i(\mathbf{a}_i, P^*) < \delta \\ 1 & \text{otherwise.} \end{cases} \tag{2.2}$$

and d_i is the minimum Euclidean distance of the ith objective vector $\mathbf{a}_i$ in S to the reference set P^*; δ is a user-defined tolerance. $d_i(\mathbf{a}_i, P^*) < \delta$ refers that $\mathbf{a}_i$ could been seen as a member of P^*.

Generational distance: GD

The generational distance (GD) metric measures the mean distance from solutions in S to the nearest neighbour in P^* in objective-space (Knowles and Corne, 2002; Knowles, 2002). A favourable (smaller) GD value implies good proximity. The ideal GD value is 0. Mathematically, the GD metric is defined as follows:

$$GD_p(S, P^*) = \frac{\sum_{i=1}^{|S|} d_i(\mathbf{a}_i, P^*)}{|S|} \tag{2.3}$$

where d_i is the minimum Euclidean distance of the ith objective vector $\mathbf{a}_i$ in S to the reference set P^*.

$$d_i(\mathbf{a}_i, P^*) = \min_{\mathbf{p}\in P^*} \{\|\mathbf{a}_i - \mathbf{p}\|_2 : \mathbf{a}_i \in S\}, \tag{2.4}$$

Set converge metric: $\mathcal{C}(A, B)$

$\mathcal{C}(A, B)$ is a binary metric which calculates the proportion of solutions in B that are dominated by solutions in A, or mathematically:

$$\mathcal{C}(A, B) = \frac{|\mathbf{b} \in B | \exists \mathbf{a} \in A : \mathbf{a} \preceq \mathbf{b}}{|B|} \tag{2.5}$$

$|A|$ and $|B|$ represent the cardinality of set A and B, respectively. $\mathcal{C}(A, B) = 1$ means all solutions of B are weakly dominated by A. On the other hand, $\mathcal{C}(A, B) = 0$ none of solutions in B is weakly dominated by A. Note that $\mathcal{C}(A, B)$ is not necessarily equal to $1 - \mathcal{C}(B, A)$. Therefore, both $\mathcal{C}(A, B)$ and $\mathcal{C}(B, A)$ need to be calculated in order to understand which set of solutions converge better.

Amongst these three metrics, the GD metric is more widely used (when the Pareto optimal front is known) by researchers due to its simplicity. However, the $\mathcal{C}(A, B)$ gives a relative comparison result which was found to be more reliable (Zitzler et al., 2003).

2.3.2 Diversity metrics

Diversity metric measures the diversity of the obtained Pareto optimal solutions. Diversity has two components: the extent of the spread of solutions and the uniformity of the spread of solutions (Zitzler et al., 2000; Knowles and Corne, 2002). A number of metrics have been proposed such as spacing (Schott, 1995), maximum spread (Zitzler and Thiele, 1999) and spread (Deb et al., 2002a).

Spacing metric

The spacing metric measures relative distances between two solutions, the smaller the better. It is defined as follows:

$$space = \sqrt{\frac{1}{|S|}\sum_{i=1}^{|S|}(d_i - \bar{d})^2} \tag{2.6}$$

where

$$\begin{aligned} d_i &= \min_{k \in S \wedge k \neq i} \sum_{m=1}^{M} \sqrt{(f_m^i - f_m^k)^2} \\ \bar{d} &= \sum_{i=1}^{|S|} \frac{d_i}{|S|} \end{aligned} \tag{2.7}$$

This metric does not consider the extent of the spread of the solutions. As long as the obtained solutions are evenly distributed (uniformity), it will produce a smaller value. It is important to normalise all objective values within an identical range before applying this metric (Deb, 2001, p. 328).

Maximum spread: D

The maximum spread metric calculates the length of the diagonal of a hyper-box formed by the extreme objective values found in the Pareto set. It measures the extent of the obtained solutions:

$$D = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\frac{\max_{i=1}^{|S|} f_m^i - \min_{i=1}^{|S|} f_m^i}{F_m^{max} - F_m^{min}}\right)^2} \tag{2.8}$$

where F_m^{max} and F_m^{min} refers the maximum and minimum objective value of the m-th objective in the reference set P^*.

Spread metric: Δ

The spread metric Δ measures both the extent and uniformity of the solutions, which effectively alleviates the limitation of the spacing and maximum spread metrics and so is more preferred. Mathematically, Δ is defined as :

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{m=i}^{|S|} \sqrt{(d_i - \bar{d})^2}}{\sum_{m=1}^{M} d_m^e + |S|\bar{d}} \tag{2.9}$$

where d_m^e is the distance between the extreme solutions in P^* and S, in terms of the m-th objective function. d_i is a distance measure (e.g. Euclidean distance) between two consecutive solutions and $\bar{d}$ is the average value of all d_i (Deb et al., 2002a).

2.3.3 Measures for evaluating both convergence and diversity

Two widely used metrics that can measure both the closeness to, and the spread along the Pareto optimal front are hypervolume (Zitzler and Thiele, 1999) and inverted generational distance (Czyżżak and Jaszkiewicz, 1998).

Hypervolume: *HV*

The hypervolume metric (HV) measures the volume of the M-dimensional region in objective-space enclosed by the obtained non-dominated solutions S and a dominated reference point, see Figure 2.4. The hypervolume is shown as the shaded region. Typically, the reference point could be set as the *nadir* point (Deb et al., 2010) or a relaxed form of the *nadir* point. Mathematically,

$$HV = \lambda(\cup_{i=1}^{|S|} v_i) \tag{2.10}$$

where λ represents the Lebesgue measure (Knowles, 2002; Auger et al., 2009b) and v_i is a hypercube constructed by the reference point and the ith solution in S.

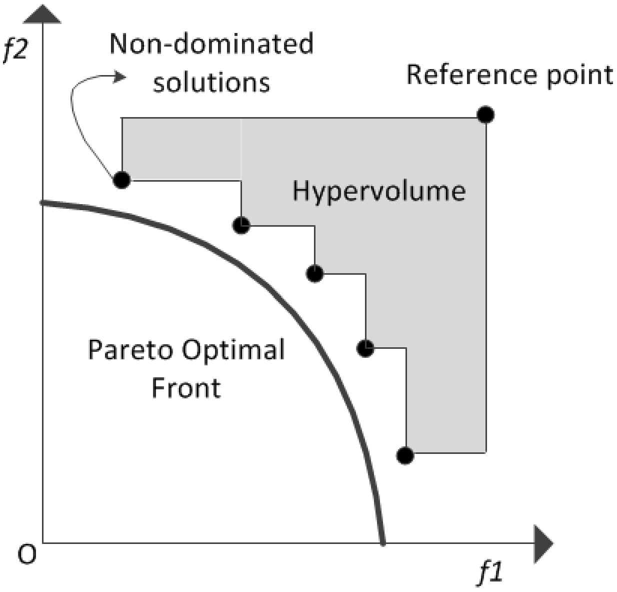
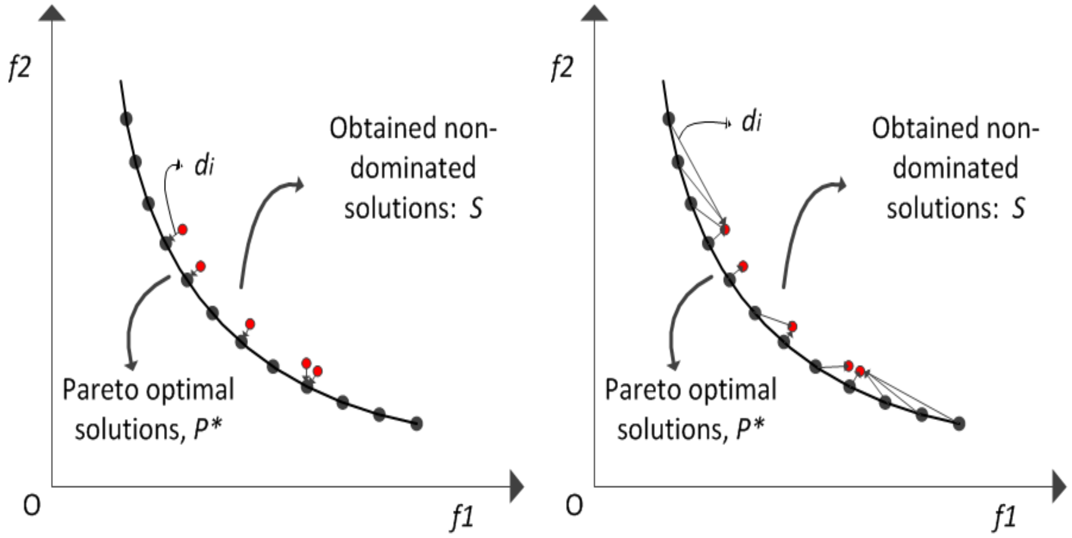


FIGURE 2.4: Illustration of the hypervolume metric.

HV is an effective unary quality measure which is strictly monotonic with regard to the Pareto-dominance. Moreover, calculation of the HV metric does not require the Pareto optimal front, which greatly facilitates the use of this metric in practice. However, there are two limitations: (i) the hypervolume calculation is computationally expensive (Beume et al., 2009), and (ii) the HV value is affected by the chosen reference point (Zitzler and Thiele, 1999; Auger et al., 2009b).

Inverted generational distance: IGD

As mentioned earlier, GD measures the mean distance of each of the obtained solutions in S to the nearest Pareto optimal solution in P^*. The inverted generational distance metric (IGD) measures the mean distance of each of the Pareto optimal solutions in P^* to the nearest solution in S, that is, $IGD = GD(P^*, S)$ (Czyzżak and Jaszkiewicz, 1998; Li and Zhang, 2009). In this sense, both the convergence and diversity performance are taken into account. For instance, in Figure 2.5 IGD value of the obtained non-dominated solutions is large since although these solutions converge well, see Figure 2.5(a), they spread poorly, see Figure 2.5(b).


f2
di
Obtained non-dominated solutions: S
Pareto optimal solutions, P*
O
f1
f2
di
Obtained non-dominated solutions: S
Pareto optimal solutions, P*
O
f1

(a) Good convergence

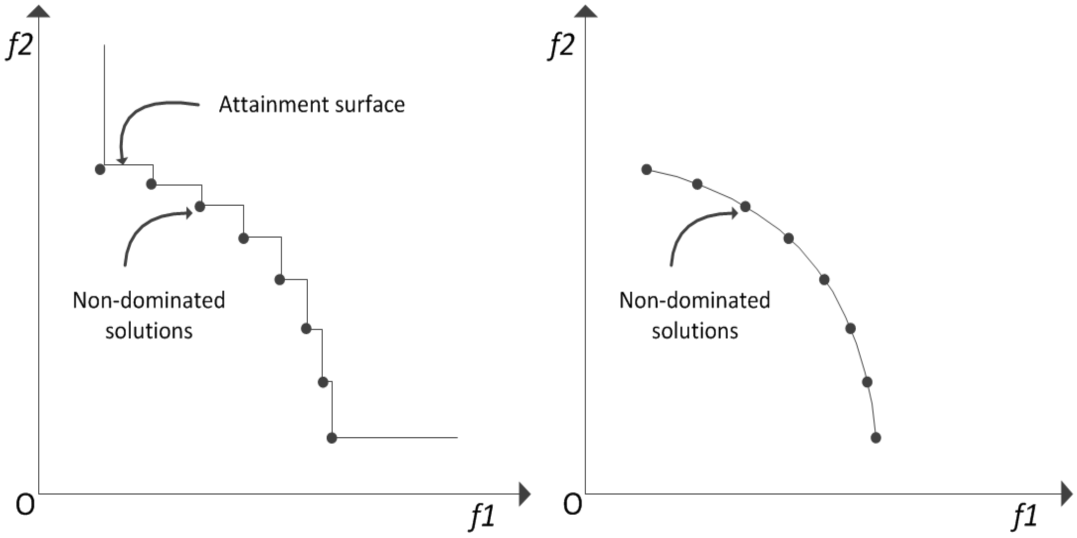
(b) Poor spread

FIGURE 2.5: Illustration of the IGD metric.

2.3.4 Visual inspection

In addition to the above quantitative metrics, to measure the quality of the obtained Pareto approximation set or compare the performance of different MOEAs, we can also simply plot the obtained solutions so as to visually examine their convergence and diversity performance. Fonseca and Fleming (1996) proposed to describe the Pareto front by an attainment surface. Instead of joining the obtained non-dominated solutions by a curve, see Figure 2.6(b), the obtained non-dominated solutions are joined by segmented lines, see Figure 2.6(a). The main limitation of the visual methods is that they are only convenient for 2- and 3-objective problems.

Overall this section provides a brief review of some representative performance metrics. It should be noted that there are other performance metrics and these can be found in survey studies of Knowles and Corne (2002) and Zitzler et al. (2003).



(a) Attainment surface (b) Joining non-dominated solutions with a curve

FIGURE 2.6: Illustration of methods to visualize the Pareto front.

To conclude this section, we discuss the usefulness of performance metrics on many-objective problems. As mentioned previously, performance metrics are used to measure the accuracy (i.e., convergence and diversity) of an approximation of the Pareto optimal front, and enable performance comparison of two MOEAs. In fact, the only unbiased way for comparing two sets of solution is to use dominance relation – whether one approximation set dominates another. This is because the only preference information that this metric relies on is a direction of preference in each objective. However, often such limited preference information is unable to distinguish between algorithms (Wang et al., 2013). Thus, we must rely on other performance metrics which unavoidably introduce an additional bias. However, as long as the bias ultimately reflects the decision-maker preferences, such comparison still makes sense. For example, the HV measures a volume enclosed by the Pareto front and a chosen reference point. The chosen reference point introduces bias, e.g., the hypervolume metric bias solutions placed at the edge on problems having concave Pareto front. However, if a decision-maker specifies his/her aspirations as the reference point, then the resulting hypervolume value is "unbiased".

Additionally, it is worth mentioning that the accuracy of an approximation set measured by a performance metric will be not reliable when the given approximation set only contains a limited number of solutions, i.e., the approximation set cannot represent the Pareto optimal front. In particular, this might happen on MaOPs as a large number of solutions are usually required to describe the entire Pareto optimal front. If there is only a limited number of solutions in the approximation set, the value derived from a performance metric becomes sensitive to the location of the solutions. In this case, the accuracy of an approximation measured by a performance metric needs careful consideration.

2.4 Many-objective optimisation

2.4.1 Challenges of many-objective optimisation

Ishibuchi et al. (2008b) identified three challenges of many-objective optimisation:

(i) The Pareto-dominance relation loses its effectiveness in offering comparability between alternative solutions in many-objective problems. Most of the objective vectors in a MOEA population become non-dominated when M is large (see Figure 2.7). The proportion of non-dominated vectors in a set of randomly generated vectors increases exponentially as the number of objective increases. As a consequence insufficient selective pressure can be generated towards the Pareto optimal front. This results in a poor performance of the Pareto-dominance based MOEAs such as MOGA, NSGA-II and SPEA2 (Purshouse and Fleming, 2003c; Hughes, 2005). Moreover, the "sweet-spot" of algorithm parameter settings that yield good performance contract greatly (Purshouse and Fleming, 2007), i.e. MOEAs become more sensitive to the user's choice of parameter settings.

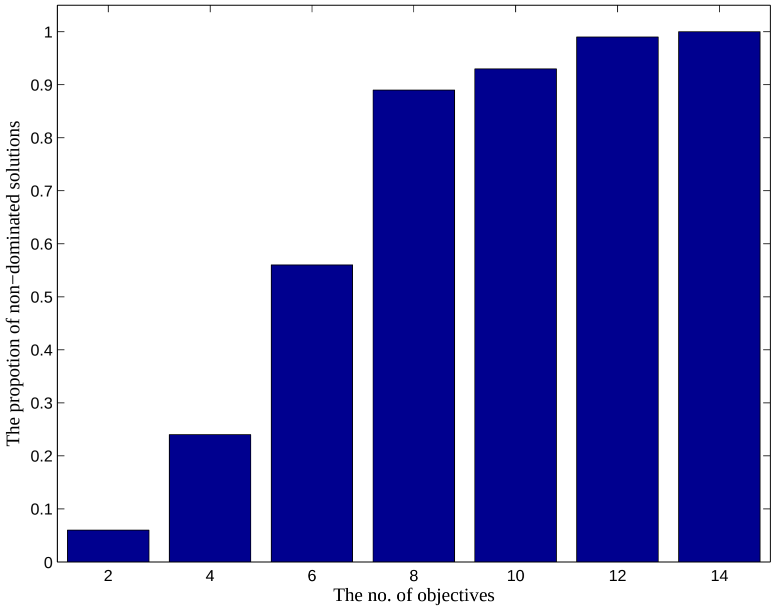


Figure 2.7: The proportion of non-dominated vectors in a set of random vectors for different number of objectives.

(ii) The Pareto optimal front of MaOPs is typically a $(M-1)$-dimension hyper-surface. The number of solutions required to represent such a surface at a given resolution grows exponentially with the number of objectives. Also it might be challenging for the DM to choose the most preferred solution from such large sets of candidate solutions.

(iii) Visualisation of the obtained Pareto approximation set is not as easy as that of two or three-objective problems. This creates difficulty for decision-makers to visually choose their preferred solutions.

A straightforward way to handle the first issue is to use a fine-grained Pareto-dominance relation which could exert more selective pressure. This will be discussed in detail together with some other alternatives in Section 2.4.2. With respect to the second difficulty, we could incorporate decision-maker preferences *a priori* or *progressively* into MOEAs during the search so as to search solutions only in the region(s) of interest to decision-makers. Regarding the third difficulty, some methods have been suggested such as visualising the Pareto approximation set using parallel coordinates (Inselberg, 1985), star coordinates (Kandogan, 2001), scatter-plot (Touchette et al., 1985), etc. Additionally, it is also helpful to apply some dimension reduction techniques to the approximated Pareto front *a priori* before visualisation so as to map the objective vectors into a low-dimensional space.

2.4.2 Methods for many-objective optimisation

A number of methods have been proposed to address the first challenging issue (Corne and Knowles, 2007; Ishibuchi et al., 2008b). These methods can be classified into five categories: modified Pareto-dominance based algorithms, ranking methods based algorithms, objective-reduction based algorithms, indicator based algorithms, and decomposition based algorithms.

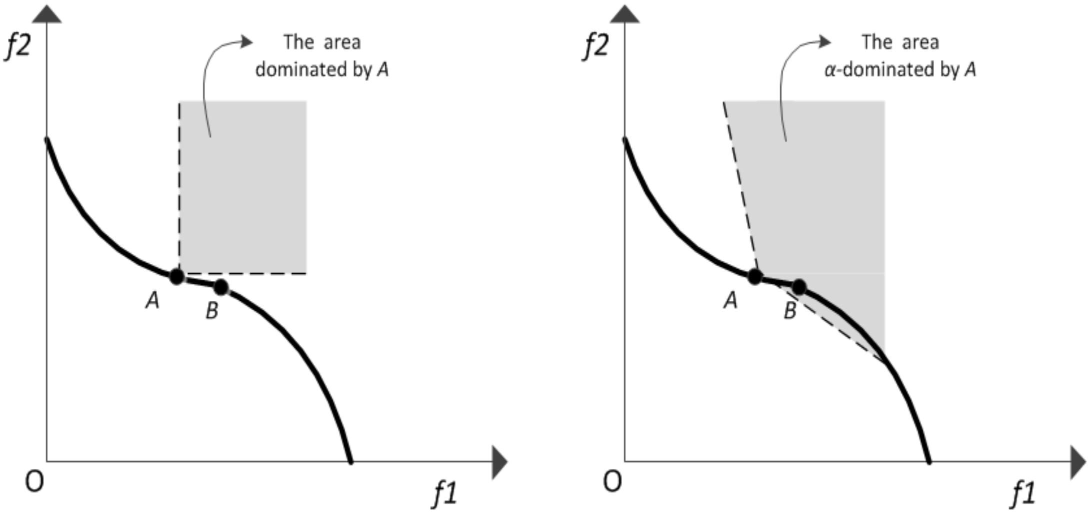
All these methods (considered as *a posteriori* approaches) focus on finding a representative approximation of the Pareto optimal front. There remains some debate about the viability of attempting to obtain such representative trade-off surfaces for MaOPs. This concerns the number of solutions required to adequately represent a trade-off surface at a desired level of resolution; this number is exponential in the number of conflicting objectives.

Mathematically, it is believed that given a reasonable size of archive set, such a representative subset can be achieved by storing all the non-dominated solutions. However, in practice, this target seems difficult to achieve and is restricted by the computational resource. Therefore, if we can use a very large number of solutions, we may be able to find a solution set which approximates the entire Pareto front very well. If we can use only a limited number of solutions, it is impossible to approximate the entire Pareto front very well. Furthermore, there could be a case that some solution sets are more representative than other solution sets. That is, we may be able to search for a solution set which is the most representative. Therefore, if we have a definition of "representative" this target is achievable by solving the question – what is the most representative solution set with a fixed number of solutions for a many-objective problem?

Modified Pareto-dominance based algorithms

According to Farina and Amato (2002), limitations of the Pareto-dominance relation on many-objective optimisation are (i) the number of objectives which are dominated; (ii) the degree of dominance; and (iii) the preferences among objectives are not considered. To overcome these limitations, various modified Pareto-dominance relations are proposed.

Ikeda et al. (2001) proposed a relaxed form of dominance relation called Pareto α-dominance. The basic idea is that a smaller improvement in one objective is not compensatable for a large deterioration in another objective. The trade-off rate between two objectives is also restricted by a pre-defined bound. Using this relation, two Pareto non-dominated solutions A and B, see Figure 2.8(a) are re-considered as A α-dominates B, see Figure 2.8(b). A similar idea is also proposed by Branke (2001), however, the author applies this concept to reveal the decision-maker preferences, thus guiding the search towards a preferred region, which will be described in Section 2.5. Experimental studies show that α-dominance works well when a proper trade-off rate is provided (Sato et al., 2007a; Fabre et al., 2010).

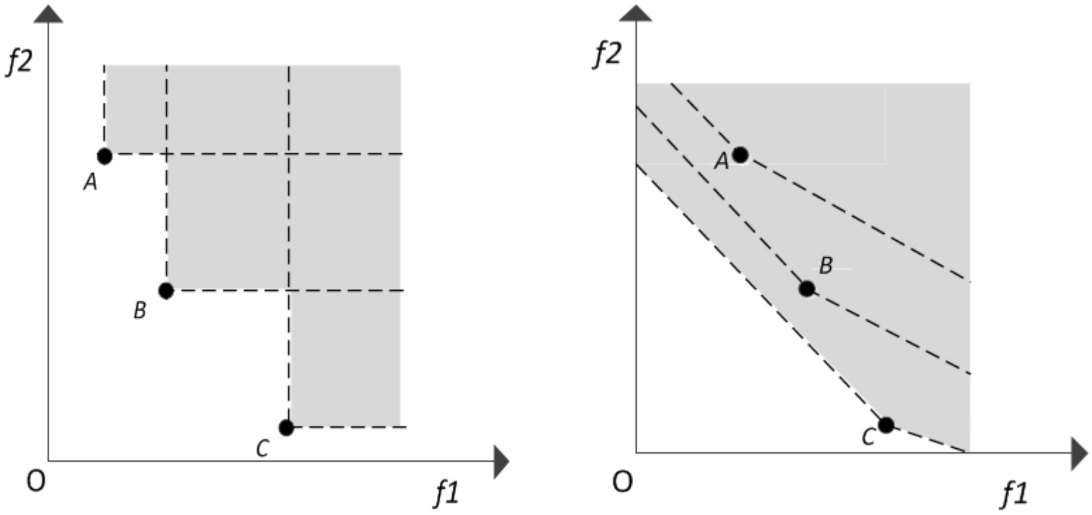


(a) A and B are non-dominated

(b) B is α-dominated by A

FIGURE 2.8: Illustration of the α-dominance relation.

Sato et al. (2007b) applied a preference relation to modify the dominance area of a solution. The dominated area can either be expanded or contracted according to a user defined vector. Figure 2.9 shows that solutions A and B are Pareto non-dominated while A dominates B when the dominance area is expanded. This modified dominance relation is applied to NSGA-II. Experimental results show that NSGA-II, when using the modified dominance relation performs well on MaOPs (Sato et al., 2007a). However, it is not straightforward to select a proper expanded or contracted degree.



(a) The standard Pareto-dominance (b) The expended Pareto-dominance

FIGURE 2.9: Illustration of a modified Pareto-dominance relation.

ε-dominance (Laumanns et al., 2002) is another widely used modified dominance relation. There are two schemes to implement ε-dominance, i.e., additive and multiplicative, as shown in Figure 2.10. A vector $\mathbf{x} \in \mathbb{R}^n$ is said to ε-dominate another vector $\mathbf{y} \in \mathbb{R}^n$, denoted as $\mathbf{x} \prec_\varepsilon \mathbf{y}$, if and only if (assuming a minimisation problem)

$$(additive) \quad \forall i \in 1, 2, \cdots, M, \quad x_i - \varepsilon \leq y_i \tag{2.11}$$

or in another scheme

$$(multiplicative) \quad \forall i \in 1, 2, \cdots, M, \quad x_i(1 - \varepsilon) \leq y_i \tag{2.12}$$

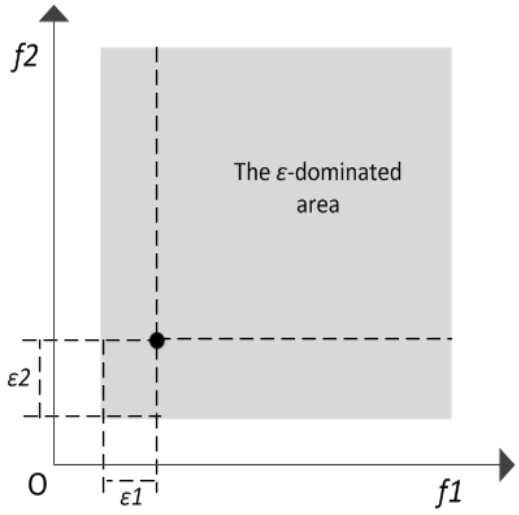


FIGURE 2.10: Illustration of the ε-dominance relation.

The value of ε can be different for different objectives, which reflects the scalings of a problem. ε-dominance based MOEAs (Deb et al., 2003, 2005) have been demonstrated to outperform NSGA-II on MaOPs (Wagner et al., 2007). However, the main issue for ε-dominance based methods is on the choice of a suitable ε value. Moreover, ε-dominance based MOEAs face difficulty on searching for extreme solutions, and the distribution of the obtained solutions is sensitive to Pareto front geometries (e.g., convex, concave, connected as well as disconnected Pareto front geometry) (Hernández-Díaz et al., 2007).

Recently, Batista et al. (2011) proposed another dominance relation named Pareto cone ε-dominance as shown in Figure 2.11. All the vectors in the shaded region are Pareto cone ε dominated by vector A. The shaded region is composed of a standard Pareto dominated area and an additional cone. This relaxation is helpful for obtaining some Pareto non-dominated solutions in some adjacent boxes that are ε-dominated. It is demonstrated that this relaxed dominance relation outperforms ε-dominance based MOEAs on MaOPs. Again the limitation is on the definition of a proper ε value and the angle of the cone.

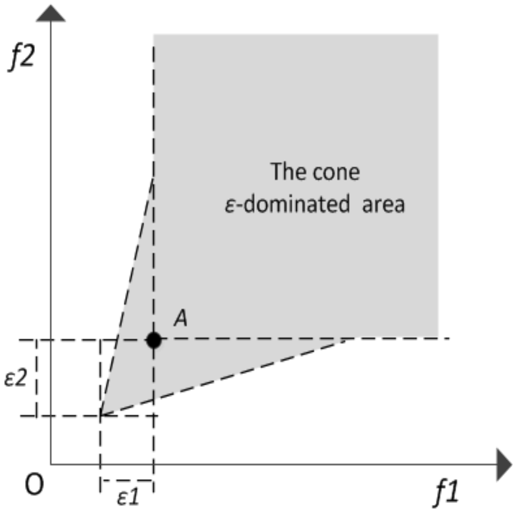


FIGURE 2.11: Illustration of the cone ε-dominance relation.

Overall, modified Pareto-dominance relations typically can generate more selective pressure towards the Pareto optimal front and therefore are more effective than the usual Pareto-dominance for many-objective optimisation. However, the main issue of this method concerns the choice of an appropriate parameter value, e.g. the ε value in the ε-dominance relation. Moreover, it is argued that modified dominance relations, while enhancing the selection pressure, might lead to degradation of solution diversity (Branke, 2001; Sato et al., 2007b).

Ranking methods based algorithms

In addition to the use of modified dominance relations, in order to obtain a partial ordering over non-dominated solutions, some ranking techniques can be used, e.g. average ranking, the *favour* relation, preference order ranking and the *winning_score* technique.

Average ranking (denoted as AR) was proposed by Bentley and Wakefield (1997), and was demonstrated to perform well on many-objective problems (Corne and Knowles, 2007). Solutions are ranked based on their performance on each objective in turn. This results in a $N \times M$ ranking matrix. For a solution $\mathbf{x}$, its ranking position is described by a vector $\mathbf{r}(\mathbf{x}) = [r_1(\mathbf{x}), r_2(\mathbf{x}), \cdots, r_M(\mathbf{x})]$, where $r_m(\mathbf{x})$ is the rank of $\mathbf{x}$ for the m-th objective. The global rank of $\mathbf{x}$ is then given by

$$AR(\mathbf{x}) = \sum_{i=1}^{M} r_m(\mathbf{x}) \tag{2.13}$$

Drechsler et al. (2001) proposed a new preference relation, known as *favour*. For two non-dominated solutions, the *favour* relation prefers the one which performs better on more objectives. Mathematically, $\mathbf{x}_i$ is favoured over $\mathbf{x}_j$, denoted as $\mathbf{x}_i \preceq_f \mathbf{x}_j$ if and only if

$$\begin{gathered} |\{m : f_m(\mathbf{x}_i) < f_m(\mathbf{x}_j)\}| > |\{n : f_n(\mathbf{x}_i) < f_n(\mathbf{x}_j)\}| \\ For\ m, n\ \in \{1, 2, \cdots, M\} \end{gathered} \tag{2.14}$$

Sülflow et al. (2007) slightly modified the *favour* relation by considering not only the number of objectives for which one candidate solution is better than the other but also the amount of the improvements in objective values between the two solutions. However, experimental results show that the *favour* relation is not as effective as AR on many-objective optimisation problems (Corne and Knowles, 2007).

Di Pierro et al. (2007) proposed another ranking method called preference order ranking (PO) for many-objective optimisation problems. Specifically, solutions are ranked based on their order of *efficiency*. A solution $\mathbf{x}_i$ with efficiency of order k means that it is not Pareto dominated by any other solutions for any of the $\binom{M}{k}$ objectives. The order of efficiency of a solution $\mathbf{x}_i$ is the minimum k value for which $\mathbf{x}_i$ is efficient. The order of efficiency is then used to rank solutions. The smaller the order of efficiency, the better the solution. Experimental results show that PO performs well for MaOPs (Di Pierro et al., 2007). However, as was demonstrated by Corne and Knowles (2007), PO is not as effective as AR.

Maneeratana et al. (2006) proposed another ranking method called *winning_score*. The *winning_score* of a solution is defined as the sum of its 'margins' over all the other solutions. For instance, if a solution $\mathbf{x}_i$ is better than $\mathbf{x}_j$ in five objectives while worse than $\mathbf{x}_j$ in two objectives, the 'margins' of $\mathbf{x}_i$ over $\mathbf{x}_j$ is 3, and the 'margins' of $\mathbf{x}_j$ over $\mathbf{x}_i$ is -3. Experimental results show that this method performs well on MaOPs. However, it is proved by Corne and Knowles (2007) that this method is essentially equivalent to the *AR* method.

Overall, the use of ranking methods enables a refined ordering to be obtained for non-dominated solutions and therefore can be used to handle MaOPs. Among these methods, *AR* is found to be the most simple, yet effective method (Corne and Knowles, 2007; Fabre et al., 2009, 2010). However, Kukkonen et al. (2007) demonstrated that the ranking methods also suffer from the problem of losing diversity .

Objective-reduction based algorithms

As is well known, almost all challenges in many-objective optimisation problems arise from the high dimension in objective-space (Purshouse and Fleming, 2003b). Therefore, a natural way to handle MaOPs could be to incorporate dimension reduction techniques into MOEAs.

In general, there are two types of dimension reduction techniques, i.e. feature extraction and feature selection. Feature extraction transforms data in a high-dimension space to a low-dimension space. Feature selection is to find a subset of the original variables that can cover the most of features of a problem (Fodor, 2002; Van der Maaten et al., 2009).

Deb and Saxena (2006) applied the linear principle component analysis (PCA), to transform high-dimension data into low-dimension date (i.e., feature extraction) for application in NSGA-II. The basic idea of the PCA based NSGA-II is to dismiss objectives that are highly correlated with others, simultaneously, maintaining the shape of the Pareto front in the remaining objective-space. This algorithm is shown to be able to identify the correct objective combination that covers most of the attributes of a problem, and therefore obtaining a good approximation of the Pareto optimal front. Their work was further extended by incorporating constraints reduction techniques (Saxena and Deb, 2008). Furthermore, a generalised framework is proposed with which both linear and non-linear objective reduction techniques can be applied (Saxena et al., 2010).

Brockhoff and Zitzler (2006) proposed a dominance preservation (feature selection) based dimension reduction technique for many-objective problems. The overall aim is to select a minimum set of objectives that can maintain the dominance structure of all considered solutions. This technique is applied to a hypervolume based algorithm which greatly improves the algorithm's performance (Brockhoff and Zitzler, 2007). Moreover, in Brockhoff and Zitzler (2009), some on-line and off-line objective reduction techniques are introduced.

Additionally, López Jaimes and his colleagues also carried out some work on this aspect. For instance, in López Jaimes et al. (2008) the authors introduced two ways to identify the most conflicting objectives. One is to find the minimum subset of objectives that yields the minimum error the other is to find a subset of objectives of a given size that yields the minimum error. Furthermore, they also proposed a simple on-line objective reduction method, which reduces the number of objectives periodically according to their correlation, i.e. the conflicting information between objectives (López Jaimes et al., 2009, 2011).

Lastly, Purshouse and Fleming (2003a) proposed an adaptive divide-and-conquer methodology for many-objective optimisation. This study is based on the understanding of that there are three different interactions between objectives: harmony, independent, conflict (Purshouse and Fleming, 2003b). A harmony relationship means the improvement in one objective is witnessed as an improvement in another objective; independent means the change in one objective has no impact on another objective; and a conflict relationship means an enhancement in one objective would result in a deterioration in another objective. The divide-and-conquer approach utilises the relationship between objectives and divides the objectives into some subsets, where possible. Note this method is also applicable to operations in decision-space. Experimental results show this method to be very encouraging.

Lygoe et al. (2010) proposed another PCA based dimension reduction method. The main feature of this approach is that it exploits local harmony between objectives to reduce dimensionality. Specifically, first, the k-means clustering approach is applied to divide the whole population into groups of like-solutions. Then, for each group of solutions, PCA together with some heuristic rules are applied to reduce objective dimensionality. This approach is applied to solve a real-world problem (i.e., automotive diesel engine calibration optimisation problem) with up to six objectives, and is shown to be effective.

The main issue of dimension-reduction based methods is that these approaches might lose advantages on problems which have many conflicting objectives. None of the introduced approaches (PCA, Pareto-dominance preservation, divide-and-conquer) has clearly shown effectiveness on problems with a large number of conflicts.

Indicator based algorithms

In addition to the above methods, another theoretically well-supported alternative to Pareto-dominance is the use of an indicator function to measure the quality of solution sets. This kind of MOEA is referred as an indicator based evolutionary algorithm (Zitzler and Künzli, 2004).

This idea was first used by Knowles and Corne (2002, 2003), where the hypervolume indicator is used for archive maintenance. Subsequently, Zitzler and Künzli (2004) proposed a general framework of indicator based evolutionary algorithm (IBEA). Wagner et al. (2007) demonstrated that IBEA outperforms Pareto-dominance based algorithms (e.g. NSGA-II) on MaOPs.

Several variants of IBEAs have been proposed amongst which the hypervolume indicator, which is the only unary quality measure that is strictly monotonic with regard to Pareto-dominance, is often used as an indicator. Emmerich et al. (2005) applied the hyervolume indicator as a selection criterion in their algorithm – $\mathcal{S}$ metric selection based evolutionary multi-objective optimisation algorithm (SMS-EMOA) (Beume et al., 2007). SMS-EMOA is a steady-state algorithm, considering one offspring per generation, in which non-dominated sorting is combined with a selection operator based on the hypervolume measure. Hypervolume is used as a second selection criterion that is similar to the use of the crowding distance in NSGA-II. Between two non-dominated solutions, the one that offers a larger hypervolume contribution is preferred. Experimental results show that SMS-EMOA outperforms NSGA-II and SPEA2 on multi-objective problems containing up to six objectives. However, the high computational effort required for the hypervolume calculation (Beume et al., 2009) significantly inhibits its application to MaOPs.

Regarding the calculation of hypervolume, Zitzler and Thiele (1999) proposed an underlying principle approach – "hypervolume by slicing objectives". The worst-case running complexity of both approaches is exponential in the number of objectives, i.e. $\mathcal{O}(N^{(M-1)})$. Some improved hypervolume calculation methods were also proposed. For example, Fonseca et al. (2006) proposed a "dimension-sweep" method with a worst-case runtime complexity of $\mathcal{O}(N^{(M-2)} \log N)$. Yang and Ding (2007) introduced an approach with a worst-case runtime complexity of $\mathcal{O}((M/2)^N)$ (as claimed by the authors). Additionally, some approximation methods have been proposed to compute the hypervolume value. Ishibuchi et al. (2009d) uses a set of achievement scalarising functions with evenly distributed weights to estimate the hypervolume value of a Pareto front. This estimation method is applied to SMS-EMOA (Ishibuchi et al., 2010). Experimental results show this method can greatly reduce the computation time of SMS-EMOA without severely affecting its performance. However, the findings are only based on 6-objective problems; as pointed out by the authors, this algorithm needs to be further tested on MOPs with more objectives. Encouragingly, Bader and Zitzler (2010) proposed an effective method in which the Monte Carlo Sampling method is used to estimate hypervolume contribution of a single solution. This approximation method significantly reduces the hypervolume computation load. The proposed algorithm "an algorithm for fast hypervolume-based many-objective optimisation" (HypE) was demonstrated to perform well on MaOPs with up to 20 objectives (Bader and Zitzler, 2011). The promising results lead HypE to be considered as a challenging competitor MOEA for many-objective optimisation.

Decomposition based algorithms[2]

Decomposition based algorithms transfer a MOP into a collection of single objective problems that are defined by means of scalarising functions (e.g. the weighted sum and the Chebyshev function) with different weight vectors. Solutions are evaluated by the weighted scalarising functions. The use of scalarising function based fitness evaluation enables decomposition based algorithms to become another well-supported approach for many-objective optimisation. Next we review some representative decomposition based algorithms. Note that decomposition based methods are also called aggregation based methods or scalarising function based methods in some studies (Wagner et al., 2007; Ishibuchi et al., 2008b).

Decomposition based algorithms were popularised after a paper (Zhang and Li, 2007) describing MOEA/D. The earliest decomposition based MOEA might be the genetic algorithm of Hajela et al. (1993) (denoted as HLGA). HLGA employs the weighted sum scalarising function for fitness assignment. Each objective is assigned a weight $w_i \in [0, 1]$ and $\sum w_i = 1$. The fitness scalar value of a solution is computed based on the weighted sum of all objective function values. A set of diversified solutions is found by employing a set of randomly generated weight vectors. However, it is known that the weighted sum scalarising function faces difficulty in searching for solutions in non-convex Pareto regions (Miettinen, 1999). To address this issue, Yaochu Jin and Sendhoff (2001) proposed a method called Evolutionary Dynamic Weighted Aggregation (EDWA). This strategy employs dynamic weights during the search, and stores non-dominated solutions in an archive. It is demonstrated that EDWA is useful for obtaining Pareto optimal solutions in both convex and non-convex region

Multi-objective genetic local search (MOGLS) can be seen as another decomposition based algorithm. MOGLS of Ishibuchi and Murata (1998) (denoted as I-MOGLS) uses a weighted sum fitness function with a randomly generated weight vector at each selection of parents. The use of random weights enables the algorithm to explore different regions of the Pareto front effectively. The weight applied in the selection procedure is also copied to the generated offspring. It is further used in the local search procedure to exploit the same search direction. I-MOGLS is improved by applying the local search probabilistically to a set of solutions selected by the 5-tournament (Ishibuchi et al., 2003).

Jaszkiewicz (2002) proposed another variant of MOGLS (denoted as J-MOGLS) where the search is also guided by random weights. The main difference between J-MOGLS and I-MOGLS is that I-MOGLS selects parents by roulette wheel selection among the whole population while J-MOGLS randomly selects parents from the best k solutions

[2] This topic is covered in considerable detail due to its relevance to the research described in Chapter 4.

in terms of the weighted scalar value. The author reported that J-MOGLS outperforms I-MOGLS on multi-objective knapsack problems.

The cellular multi-objective genetic algorithm (CMOGA) (Murata et al., 2000, 2001) can also be regarded as a decomposition based algorithm. In CMOGA each cell is associated with a weighted scalarising function, more precisely, a weighted sum function. Weights are evenly distributed in the search space, and this can effectively guide the search towards different parts of the Pareto optimal front. A neighbourhood, measured by the distance between two weight vectors, is defined for each cell. Mating restriction is applied, that is, for each cell a new individual is generated by applying genetic operators to neighbouring parents. The efficiency of CMOGA is demonstrated on a bi-objective flowshop scheduling problem. CMOGA is further extended by combing a local search procedure by Murata et al. (2002). The resulting algorithm, C-MOGLS, applies a local search procedure to all non-dominated solutions in every generation. The performance of C-MOGLS is tested on both flowshop scheduling and pattern classification problems. However, good results are only found for scheduling problems. The considered reason is that the balance between local search and global search is not appropriately controlled. C-MOGLS is further improved in Murata et al. (2003), where a suitable search direction is designed for each non-dominated solution based on its location. The performance of this modified C-MOGLS is tested on a 3-objective pattern classification problem, and is found to be effective.

All the above algorithms have the potential to handle MaOPs, however, none of them is developed with the concern of addressing MaOPs. The first decomposition-type approach that is also claimed as a many-objective optimiser is the multiple single objective Pareto sampling (MSOPS (Hughes, 2003)). In MSOPS a set of evenly distributed weighted vectors (target vectors) is employed to simultaneously guide the search towards multiple search directions. Each solution is evaluated by a set of weighted saclarising functions, which produces multiple ranks for each solution. These solutions are then ranked by using the lexicographical ordering method based on the rank. MSOPS is shown to perform well on both bi- and many-objective problems (Hughes, 2003; Wagner et al., 2007).

MOEA/D, as the state-of-the-art decomposition based algorithm, has attracted a lot of attention and has been shown to have a high search ability for continuous optimisation and combinatorial optimisation (Zhang and Li, 2007; Ishibuchi et al., 2009c; Zhang et al., 2010). It is also found to outperform NSGA-II on problems having complex Pareto sets (Li and Zhang, 2009). As the winner of the “Unconstrained multi-objective evolutionary algorithm” competition at the 2009 Congress on Evolutionary Computation (Zhang et al., 2009), MOEA/D is an important approach to consider for solving many-objective optimisation problems.

Overall, compared with Pareto-dominance based algorithms, decomposition based algorithms have a higher search ability and are more computationally efficient on fitness evaluation (Zhang and Li, 2007; Li and Zhang, 2009). However, it is also known that the performance of decomposition based algorithms depends on the employed weights - if inappropriate weights are specified, the obtained solutions might not be well-spread. Usually, determining a good set of weights *a priori* for real-world problems is not straightforward due to a lack of knowledge of the underlying problem structure. To overcome this limitation, some effort has been devoted to apply adaptive weights to decomposition based algorithms. Related studies are reviewed as follows.

Kim and De Weck (2005, 2006) proposed an adaptive weighted-sum approach which is shown to be able to find well-spread solutions for MOPs having complex geometries. The main idea is to impose an equality constraint on the original multi-objective problem, and solve it using the weighted sum method. Specifically, first the weighted sum method is applied to quickly approximate the bounds of the Pareto optimal front. Having determined the bounds, a mesh of the Pareto front patches is created. The Pareto front patch is then refined by searching for solutions that satisfy a defined equality constraint. The constraint guarantees the solution to be placed along the direction from the estimated *nadir* point to an expected Pareto optimal solution. The expected Pareto optimal solution is determined by the nodes of each patch with a specified weight factor. This method performs well on bi- and three-objective problems. However, no obvious evidence exists to show that this method would also work well on many-objective problems. Besides, the method itself is rather complex as solving MOPs with equality constraints is already difficult (Datta and Deb, 2011).

Hughes also extended his earlier work, i.e. MSOPS, to consider on-line generation of target vectors by bootstrapping these from the on-line archive of locally non-dominated solutions, with mixed results (Hughes, 2007). The target vector essentially is equivalent to weight vectors. Specifically, in this extended algorithm MSOPSII, target vectors are updated as follows. First the objective vectors of the current population are normalised so as to create valid weight vectors. Secondly, take the current weight vectors and augment each one with each of the created weight vectors in turn. Each time we calculate the angle between each pair of weight vectors and remove the most crowded weight vector.

Jiang et al. (2011) improved MOEA/D by using Pareto adaptive weights, $pa\lambda$. The approach $pa\lambda$ automatically adjusts weights according to the geometry characteristics of the Pareto front. Specifically, this approach assumes that the Pareto optimal front is symmetric of the form $f_1^p + f_2^p + \cdots, +f_M^p = 1$. Based on the non-dominated solutions in the archive, parameter p is estimated. Having determined p, evenly distributed points on the approximated symmetric shape are generated; these are then converted to valid weights. The use of $pa\lambda$ can significantly improve the performance of MOEA/D when the Pareto front geometry is close to the form: $f_1^p + f_2^p + \cdots, +f_M^p = 1$. However, this method faces difficulty on problems with an asymmetric and disconnected Pareto front.

Gu et al. (2012) proposed another algorithm which employs a dynamic weight design method in MOEA/D, denoted as DMOEA/D. This algorithm features in Chapter 4 for comparison purposes. Specifically, in every ten generations weights are re-generated according to the shape of the current non-dominated Pareto front. A piecewise linear interpolation method is applied to fit a curve (2-objective case) of the current non-dominated solutions. For each objective f_i, N_i interpolation points, of which the projection on the ith objective is uniform, are generated. N_i is determined by $\frac{10^{M-1}N\prod_{j=1,j\neq i}^{M} D_j}{\sum_{i=1}^{M}\prod_{j=1,j\neq i}^{M} D_j}$ where N is the population size and D_j is the maximum value of the projection of the current non-dominant solutions on the ith objective. At the same time, non-dominated solutions whose distance to the interpolation point is smaller than $\frac{D_j}{10}$ are removed. All the $\sum N_i$ interpolation points serve as an approximation to the Pareto front. These interpolation points are then ordered by an objective (e.g. f_1). Solutions that are adjacent in f_1 are clustered into one group. The maximum number of solutions in a group is $[\sum N_i/N']+1$, where N' is the size of current non-dominated solutions. A weight vector is then created by converting the point defined by the mean of the solutions in a group. Experimental results show DMOEA/D is able to find evenly distributed solutions for bi- and three-objective problems having different geometries. Moreover, this adaptive approach is reported to be applicable on many-objective problems, however, no experimental results are shown.

All the above methods attempt to maintain evenly distributed solutions on the fly. There are also some methods that aim to first obtain as many diversified solutions as possible and then apply some ad-hoc methods to the obtained solutions so as to get evenly distributed solutions. EMOSA (Li and Landa-Silva, 2011) belongs to this type. It hybridises MOEA/D with simulated annealing. The simulated annealing based local search is applied to improve the current solution of each single objective problem. In EMOSA weights are adaptively modified as follows: for each member $\mathbf{F\_s}_i$ in the current population, first find the closest neighbour (e.g., $\mathbf{F\_s}_j$) to $\mathbf{F\_s}_i$ and its associated weight vector $\mathbf{w}_j$. Secondly, identify the weights in the pre-defined weight set (in Li and Landa-Silva (2011) this weight set is formed by a set of evenly distributed weights generated by using the simplex-lattice design method) whose Euclidean distance to $\mathbf{w}_j$ is larger than the distance between $\mathbf{w}_i$ and $\mathbf{w}_j$. Furthermore, amongst the identified weights, select all the weights of which the distance between them and $\mathbf{w}_i$ is smaller than the distance between them and all the neighbours of $\mathbf{w}_i$. The definition of the neighbourhood (T) is the same as MOEA/D. If there are multiple weights then pick one randomly. Experimental results show that EMOSA outperforms three multi-objective simulated annealing algorithms (SMOSA (Serafini, 1994), UMOSA (Ulungu et al., 1999), CMOSA (Czyzżak and Jaszkiewicz, 1998) and three multi-objective memetic algorithms (i.e.. I-MOGLS, the improved I-MOGLS and MOEA/D) on 2- and 3-objective knapsack problems and travelling salesman problems. Due to the promising performance of EMOSA, it is also used in Chapter 4 for comparison purpose.

Derbel et al. (2013) proposed to dynamically compute a suitable search direction for each solution during the search. The search direction is designed in an attempt to adaptively fit the search process and approach the Pareto optimal more effectively. Five different strategies are proposed which are repulsive force-based directions (R-D), repulsive-attractive directions (RA-D), dominance based directions (D-D), non-backward directions (NB-D) and black hole directions (BH-D). These strategies are based on the so called "attraction-repulsion force-based rules", that is, solutions are attracted by other solutions dominating it and repelled otherwise. Observed from the experimental results, these five strategies are not very outstanding on MOPs and MaOPs.

Overall the use of adaptive weights is potentially helpful to handle the issue of Pareto front geometry for decomposition based algorithm. However, first none of the existing approaches has clearly shown their benefits on MaOPs; in addition, it is suspected that adapting weights during the search may affect the convergence performance of MOEAs (this will be discussed in Chapter 4).

2.4.3 Summary

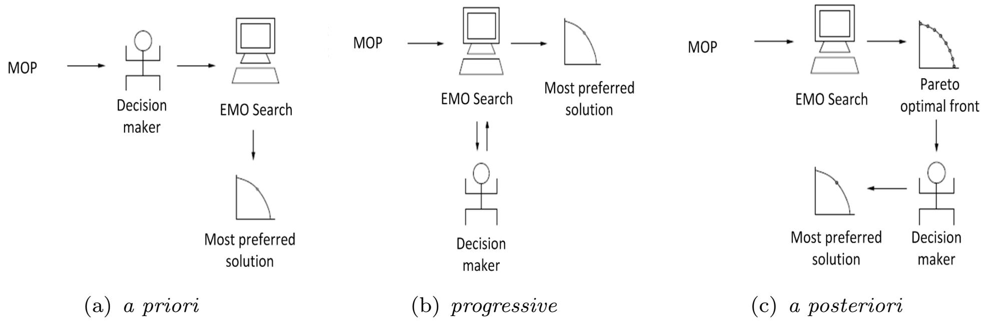
In this section we first summarise the challenges of many-objective optimisation. Following that, existing approaches proposed for solving many-objective optimisation problems, in terms of different classes (modified dominance relation based, ranking methods, objective-reduction based, indicator based and decomposition based), are reviewed together with their advantages and disadvantages. These observations motivate us to develop more effective methods for addressing many-objective optimisation problems. Such methods will be studied in Chapters 3 and 4.

2.5 Preference-based MOEAs

Once an approximation of the Pareto optimal front has been obtained, the MOP remains to be solved completely, a decision-maker (DM) has to reveal his/her preferences to choose one solution from the obtained solutions. This section reviews some representative studies for incorporating decision-maker preferences into MOEAs. Prior to reviewing these studies, we describe three different decision-making methods.

2.5.1 *A priori*, *progressive* and *posteriori* decision-making

According to when decision maker preferences are incorporated, i.e., before, during or after the search, decision-making approaches approaches can be divided into three classes – *a priori*, *progressive*[3] and *a posteriori*, respectively (Veldhuizen and Lamont, 2000).



(a) *a priori* (b) *progressive* (c) *a posteriori*

FIGURE 2.12: Illustration of different decision-making approaches in hybrid EMO-MCDM schemes.

(i) In an *a priori* decision-making approach, the DM preferences are incorporated prior to the search process as shown in Figure 2.12(a); the preferences are, in general, incorporated in the fitness calculation. The weighted sum approach is one of the most commonly employed *a priori* methods, where the DM preferences are formulated by a weight vector that indicates the relative importance of the objectives. When the DM preferences can be faithfully captured in a mathematical model, *a priori* method would be effective and efficient. However, this is rarely the case. The complete knowledge of the problem is often unavailable.

(ii) In an *progressive* decision-making approach, the DM preferences are incorporated progressively during the optimization process, see Figure 2.12(b). This enables a DM to learn about the problem and fine-tune his/her preferences if needed, effectively guiding the search towards regions of interest and away from exploring non-interesting solutions. The main limitation of this scheme is that DMs needs to be involved during the whole search process.

(iii) In an *a posteriori* approach, the DM preferences are incorporated after the search; an approximation of the Pareto optimal front is found first followed by selection of a preferred solution by the DM from the set of trade-off solutions previously obtained, see Figure 2.12(c). *A posteriori* approach is preferred for MOPs with 2 or 3 objectives since this approach performs well for these problems i.e. a good approximation of the Pareto optimal front can be obtained. Additionally, the availability of the entire Pareto optimal front enables the DM to confidently select

[3] Note that *progressive* is also termed *interactive* in some studies. These two terminologies are essentially equal and interchangeable. In this thesis, we use *progressive* following the study of Fonseca and Fleming (1998a).

his/her preferred solutions. However, *a posteriori* scheme becomes less effective on MaOPs Purshouse and Fleming (2003b). Not only does the computational burden for solving this problems becomes very expensive to estimate the whole Pareto optimal front, the approaches become more inefficient since the DMs often seek solutions within particular regions of the Pareto front. Furthermore, the number of Pareto optimal solutions required for describing the entire Pareto optimal front of a MaOP is usually very large. Selecting one preferred solution from all these solutions is cognitively difficult.

To date, a considerable effort has been spent on developing efficient EMO approaches for finding a well-converged and well-distributed set of Pareto optimal solutions, supporting *a posteriori* decision making, including, for example, MOGA (Fonseca and Fleming, 1993, 1998a), NSGA-II (Deb et al., 2002a), SPEA2 (Zitzler et al., 2002), and HypE (Bader and Zitzler, 2011). See Zhou et al. (2011) for a review. However, as mentioned above, this scheme faces difficulties when applied onto MaOPs, and such problems arise regularly in the real-world (Fleming et al., 2005). Thus, to facilitate the process of decision making, alternative is to consider incorporating DM preferences *a priori* or *progressively* into the EMO approaches. Such hybrid approaches might take advantages of both EMO algorithms and MCDM. methods.

2.5.2 Integrating DM preferences into MOEAs

Multi-criteria decision-making (MCDM) approaches (Miettinen, 1999) that are based on MOEAs (named preference-based MOEAs) have been discussed in a number of studies, for example, Coello Coello (2000), Rachmawati and Srinivasan (2006) and Ben Said et al. (2010). This section first describes methods for modelling DM preferences and then reviews some preference-based MOEAs.

Based on the study of Coello Coello et al. (2007), methods for modelling DM preferences can be categorised as follows:

(i) Aspiration level (reference point): this describes the desired levels of objectives (goal specification) from a decision-maker. The main advantage of the method is in the simplicity, that is, it does not demand a significant of amount of effort from the DM (Fonseca and Fleming, 1998a). A reference point can be seen as an alternative to aspiration level. It is often used to search for non-dominated solutions that are close to an expected solution specified as the reference point.

(ii) Weight: this describes the relative importance of objectives to a decision-maker. Lexicographical ordering could be seen as an extreme case of using weights, that is, a more important objective is infinitely more important than a less important objective. The optimum solution is obtained by minimising the objective functions in sequence, starting with the most important one and proceeding according to the assigned order of importance of the objectives. In addition, weight can also be interpreted as a reference direction, which is used to search for solutions along a given search direction.

(iii) Trade-off rate: this describes how much improvement in one or more objective(s) is comparable to a unit degradation in another objective. Modelling preferences as trade-offs between objectives is easy and effective on MOPs with fewer objectives, however, the generalisation of this approach to a higher number of objectives is difficult. This is because the involved dominance calculations become increasingly complex when the number of objectives is large (Branke and Deb, 2005).

(iv) Utility function: this is used to determine a preference index of one solution over another. The reference point, reference direction and trade-off information can all be formulated by a utility function. This approach is effective when a suitable utility function is provided. However, providing a proper utility function is difficult due to the lack of problem knowledge.

(v) Outranking: this approach determines a preference index through a series of pairwise comparisons of alternatives. Outranking is non-transitive, that is, if A outranks B and B outranks C, this does not infer that A outranks C. Therefore, it is difficult to generate the final rank of all alternatives.

(vi) Fuzzy Logic: the DM preferences are expressed linguistically. This is more friendly compared with using numerical values. The main disadvantage of this method is that formulation of the fuzzy inference system requires intensive tuning of membership functions and inference operators (Rachmawati and Srinivasan, 2006).

Preference-based MOEAs using aspiration levels

MOGA (Fonseca and Fleming, 1993) is the one of the earliest MOEAs that incorporates DM preferences in the search. In this study, the non-dominated ranking mechanism is extended to accommodate preferences (goal information, aspiration level) as an additional criterion. Individuals are ranked based on this modified ranking mechanism. The preferences can be incorporated either *a priori* or *progressively*. With the supplied preferences, the search is gradually guided towards the region of interest to the decision-maker.

Fonseca and Fleming (1998a) further extended their work by introducing a preferability operator, with which both goals and priorities (supplied by the DM) can be accommodated in the ranking scheme. This new ranking scheme provides a unification of Pareto optimality, the lexicographic method, goal programming, constraint satisfaction and constrained optimisation. MOGA has been successfully used in optimising a low-pressure spool-speed governor of a Pegasus gas turbine engine and many other applications (Fonseca and Fleming, 1998b; Fleming et al., 2005).

The main advantages of MOGA are in its simplicity of implementation, and effectiveness of scaling up to many-objective problems. The main weakness of this approach is that it requires a decision-maker to know the ranges of objective values in order to specify coherent aspiration levels. Some studies have been proposed to enhance the performance of MOGA. For example, Tan et al. (2003) improved MOGA by introducing a new goal-sequence domination scheme to allow advanced specifications such as hard/soft priorities and constraints to be incorporated.

Moreover, Molina et al. (2009) suggested a dominance relation called *g-dominance* (*g* refers to goals). Solutions satisfying all the aspirations and solutions fulfilling none of the aspirations are preferred over solutions satisfying some aspirations. An approach called g-NSGA-II that couples *g-dominance* and NSGA-II is proposed to search for solutions satisfying the specified aspirations. This algorithm works regardless of whether the specified goal vector is feasible or infeasible. However, Ben Said et al. (2010) demonstrate that g-NSGA-II faces difficulties when the provided goal vector is close to the true Pareto front (as the approach does not preserve a Pareto based ordering). Moreover, handling multiple regions of interest (ROIs) by g-NSGA-II is not discussed in their paper. Intuitively, the *g-dominance* relation is not easy to extend to handle multiple ROIs as an individual can *g-dominate* one goal vector, and simultaneously, be *g-dominated* by another goal vector.

In addition to the use of aspiration levels, a large body of approaches are developed based on the use of reference point, reference direction and light beam search (Miettinen, 1999, pp. 179-189). These will be reviewed in the following sections.

Preference-based MOEAs using reference point

One representative reference point based MOEA is R-NSGA-II proposed by Deb and Sundar (2006). R-NSGA-II hybridised the reference point approach with NSGA-II. The reference point is not applied in a classical way, i.e. together with an achievement scalarising function (Miettinen and Mäkelä, 2002), but rather by establishing a biased crowding scheme. Solutions near reference points are emphasised by the selection mechanisms. The extent and the distribution of the solutions is maintained by an additional parameter ε. Another representative reference point based MOEA is the preference-based evolutionary algorithm (PBEA) proposed by Thiele et al. (2009). It is a hybridisation of the reference point method with the indicator based evolutionary algorithm (IBEA). Preferences are incorporated using a binary quality indicator (the ε-indicator) which is also Pareto-dominance preserving. The spread range of solutions is controlled by an additional fitness scaling factor. Both R-NSGA-II and PBEA are shown to perform well in finding a set of preferred solutions in the regions of interest to the DM. However, for both the approaches, the spread range of the obtained solutions is controlled by an additional parameter which is not easy to configure.

The reference point method is also used in multi-objective particle swarm optimisation algorithm (MOPSO) (Allmendinger et al., 2008; Wickramasinghe and Li, 2008). The main idea is to apply the DM preferences (reference points) to the selection of leaders. This approach is shown to perform well on 2- and 3-objective test problems. However, the effectiveness of this approach has not been demonstrated on many-objective problems.

Ben Said et al. (2010) proposed another approach named r-NSGA-II. This approach employs the *r-dominance* relation to create a strict partial order over non-dominated solutions. It is another reference point based approach as the *r-dominance* relation prefers solutions that are closer to a specified reference point while preserving the order induced by Pareto-dominance relation. In r-NSGA-II, as well as a reference point, two additional parameters δ and $\mathbf{w}$ are introduced. $\delta \in [0, 1]$ is used to control the range of the ROIs, and $\mathbf{w}$ expresses the bias of the DM. The performance of r-NSGA-II is assessed on a set of benchmarks ranging from 2 to 10-objective problems and is shown to be good on searching for both single and multiple ROIs. Moreover, the authors compare r-NSGA-II with R-NSGA-II and PBEA, and report that r-NSGA-II offers competitive and better performance on most problems. However, as pointed out by the authors, r-NSGA-II faces difficulties on multi-modal problems, such as ZDT4.

Preference-based MOEAs using weights (reference direction or light beam search)

Researchers have also conducted studies on incorporating DM preferences expressed as weights into MOEAs. For example, Deb and Kumar (2007a) combined the reference direction with NSGA-II. Preferences are modelled using the reference direction from a starting point to a reference point. This approach is able to find Pareto optimal solutions along the reference direction and multiple ROIs can be obtained by using multiple reference directions. Deb and Kumar (2007b) also hybridised NSGA-II with the light beam search method. The hybridised approach is able to search part(s) of Pareto optimal fronts illuminated by the light beam emanating from a starting point to the reference point with a span controlled by a threshold.

Additionally, light beam search is also hybridised with MSPSO by Wickramasinghe and Li (2009). These approaches are all reported to have good performance on searching for solutions in the regions of interest to a decision-maker. However, again, the difficulty is how to appropriately control the spread range of the obtained solutions.

Preference-based MOEAs using trade-off rate and weighted hypervolume

Branke (2001) proposed an approach called guidance in evolutionary multi-objective optimisation (G-MOEA). In G-MOEA the DM preferences are manifested through a modification of the dominance relation, specifying an acceptable trade-off rate between objectives. G-MOEA works well for two objectives. However, as mentioned earlier, providing all pair-wise information for a problem with many objectives is cognitively intensive and needs $\frac{M^2-M}{2}$ comparisons. For $M = 2$ only one comparison is needed, for $M = 3$, three comparisons are needed. However, when M increases to 10, 45 comparisons are required.

Branke and Deb (2005) suggested a modified and controllable biased crowding approach. Their approach aims to search for a set of Pareto optimal solutions that are parallel to an *iso-utility* function defined by a specified reference direction. In this approach, a parameter is applied to control the range of ROI along the Pareto optimal front. The parameter is defined as the ratio of the real distances between neighbouring solutions on the Pareto optimal front and the projected distance of the same solutions on a plane with a user specified direction connoting a central linearly weighted utility function. Compared with G-MOEA, this approach is easier to scale up to high dimension problems.

In Zitzler et al. (2007), the authors integrate weight preferences in the calculation of hypervolume indicator. The weighted hypervolume indicator serves as a means of integrating the DM preferences. Auger et al. (2009a) implemented this idea on HypE and developed a new approach W-HypE. W-HypE articulates the DM preferences using a weight distribution function. It is demonstrated to perform well on searching for preferred solutions for both bi and many-objective problems. The only issue is that the spread range of the ROI is controlled by a deviation parameter in the weight distribution function. Defining a proper value for this parameter is not easy for a decision-maker.

Preference-based MOEAs using outranking

The outranking approach orders alternatives based on a series of pairwise comparisons. There has been some effort made to incorporate outranking approaches, such as PROMETHEE, ELECTRE I and PROMETHEE II into MOEAs.

Pereira (1997) integrates PROMETHEE into a MOEA, respectively. Having obtained a set of non-dominated solutions, pairwise comparisons of these solutions are performed. Specifically, an aggregating method is used to measure whether a solutions is preferred over another one. The final rank of these solutions is determined by the comparison results. This approach can also applied in a *progressive* way.

Massebeuf et al. (1999) combine PROMETHEE II with MOEAs. The DM is required to show their preferences (either to prefer one of two solutions or to be indifferent) for every pairwise comparison of alternatives (non-dominated solutions obtained by MOEAs). Simultaneously, the DM is required to specify the relative importance of different objectives. Having this information, both concordance and discordance indices of alternatives are computed. The outranking degree for each pairwise comparison is then created according to these indices. All alternatives are finally sorted based on the outranking degrees. Parreiras and Vasconcelos (2005) also integrated PROMETHEE II with NSGA-II. In their approach, a Gaussian preference function and an aggregating function are applied to compute a global preference for the non-dominated solutions obtained by NSGA-II.

There are also some studies that combine the outranking method ELECTRE I with MOEAs, e.g., Pereira (1995). First, alternatives are pairwise compared for each objective under consideration. Then, a global preference order is obtained by applying an aggregating function to the comparison results. Additionally, some studies have combined the PROMETHEE II with MOEAs (Rekiek., 2000; Rekiek et al., 2000). Similarly, the DM is asked to assign weight to each objective. The weight information is then incorporated into PROMETHEE II for calculating a net flow for all non-dominated solutions. These solutions are then ranked based on the value of net flow. It is also applied in a *progressive* manner. Weights can be changed along the optimisation process.

Overall, using the outranking method, the DM preferences can be easily incorporated during the pairwise comparison. The main advantage of this method is that pairwise comparison is cognitively easy for a decision-maker. However, its main weakness is that generation of the final ordering of alternatives is rather complex as outranking is non-transitive. Moreover, outranking methods often require many parameters which are not easy to configure (Coello Coello et al., 2007).

Preference-based MOEAs using Fuzzy logic

To solve multi-criteria decision-making problems, many researchers have also considered combining fuzzy logic methods with MOEAs. This combination aims to take advantages of both fuzzy logic (expressing preferences more naturally) and evolutionary computation (identifying a set of trade-off solutions in a single run).

Voget and Kolonko (1998) use a fuzzy controller, equipped with pre-defined goals, to adaptively control the selection pressure of MOEAs. The idea is similar to goal attainment or aspiration levels. However, in their approach goals are specified in vague terms by some membership function. Similar ideas are also implemented in studies by Lee et al. (1996) and Lee and Esbensen (1997).

Pirjanian (2000) transforms the DM preferences into weights using some fuzzy rules and then apply these weights to some traditional multi-objective optimisation methods, e.g. weighed sum and goal programming. This idea is further developed by Pirjanian and Mataric (2000), where fuzzy rules are integrated with standard fuzzy inferencing. Although this method is not combined with MOEAs, it is evident that MOEAs are promising alternatives to the traditional multi-objective optimisation approach.

Jin and Sendhoff (2002) proposed an approach to turn fuzzy preferences into weighted intervals. This approach is then incorporated into a MOEA using random/dynamic weighted sum techniques (Jin and Okabe, 2001). The resultant algorithm has shown good performance on bi-objective problems. However, this method has difficulty on non-convex problems due to the use of weighted sum function. To handle this limitation, the Chebyshev function can be used instead.

Wang and Terpenny (2003) use a fuzzy set-based aggregating function to model the DM preferences. The aggregating function is constructed based on the weights (indicating the relative importance of different objectives) and the trade-off information of a pair of objectives. This approach is used in a *progressive* way within an agent-based system. First, a set of non-dominated solutions are obtained by using a set of random weights. Then, the solutions are refined according to the defined fuzzy preference aggregating function.

Overall, applying fuzzy logic to incorporate the decision-maker preferences is non-trivial, however, the main difficulty lies in the construction of fuzzy inference system, i.e., tuning membership functions and inference operators.

2.5.3 Summary

This section reviewed some representative Preference-based MOEAs that are developed for assisting multi-criteria decision-making. All approaches have their own advantages and are able to find solutions in the region(s) of interest to a decision-maker. However, as previously mentioned, most of these approaches have limitations. Some cannot explore multiple ROIs, e.g., MOGA (aspiration level family) and G-MOEA (trade-off information); some face difficulties in scaling up to many-objective problems, e.g. G-MOEA, g-NSGA-II (aspiration level family) and outranking based approaches; and some require some additional parameters to control the range of ROIs, e.g., R-NSGA-II, PBEA, W-HypE and r-NSGA-II (reference point/direction family).

Another important observation is that none of the above approaches has catered simultaneously to the different ways that a DM can specify his/her preferences. However, this is important as in some cases it might be easier for the DM to express his/her preferences by aspiration levels and in some cases by weights or some other ways. In another perspective, different decision-maker may have his/her preferred approaches or be more confident in some particular way of expressing his/her preferences.

Overall, it would be helpful to develop an approach that can effectively overcome most of the limitations of the existing methods. Such an approach will be studied in Chapter 6.

2.6 Summary

This Chapter provided a comprehensive review of the current literature relevant to this thesis, such as the state-of-the-art MOEAs, multi-objective benchmark problems, performance metrics and particularly the studies on many-objective optimisation and Preference-based MOEAs.

Many-objective optimisation has become a hot topic in EMO field. The reason is that many-objective problems arise regularly in practice while the state-of-the-art MOEAs lose their effectiveness on such problems. Followed by a description of the challenges of many-objective problems, different types of algorithms proposed for solving many-objective problems are reviewed, together with their advantages and disadvantages. This analysis motivates us to develop effective algorithms for many-objective optimisation. These will be studied in Chapters 3 and 4 where two novel approaches will be proposed. Additionally, as the performance of MOEAs is often affected by their associated parameter configurations, Chapter 5 provides a parameter sensitivity analysis for the proposed two algorithms.

Preference-based MOEAs proposed for *a priori* and *progressive* decision-making are also discussed in terms of five categories in detail. Our review shows that most of the existing approaches have at least one weakness, such as handling multiple ROIs, scaling up to many-objective problems, controlling the spread range of ROI, etc.. Moreover, it is observed that none of approaches can cater simultaneously for the different ways that a decision-maker can specify his/her preferences. Limitations of the existing approaches motivate us to develop some more effective preference-based MOEAs. These will be studied in Chapter 6.

Chapter 3

Preference-inspired co-evolutionary algorithm using goal vectors

3.1 Introduction

The simultaneous optimisation of many objectives, in order to obtain a full and satisfactory set of trade-off solutions to support *a posteriori* decision-making, remains a challenging problem (Purshouse and Fleming, 2003c; Corne and Knowles, 2007). One of the major challenges identified for many-objective optimisation is the reduced ability of the Pareto-dominance relation in offering comparability between alternative solutions. This lack of comparability means that algorithms using Pareto-dominance struggle to drive the search towards the Pareto optimal front (Purshouse and Fleming, 2007). However, it has long been known that by using decision-maker preferences we can potentially gain comparability between otherwise incomparable solutions. A classic example can be found in the study of Fonseca and Fleming (1998a), where the effect of different specifications of decision-maker goals and priorities on the partial ordering of solutions in an enumerated search space is shown with striking clarity.

However, our interest here remains in assisting *a posteriori* decision making – that is, providing decision-makers with a satisfactory representation of the entire Pareto optimal front, prior to the elicitation and application of their preferences. In this sense, we are interested in holding multiple sets of hypothetical preferences simultaneously, to provide multiple comparison perspectives simultaneously, which are sufficient to adequately describe the whole front. The simultaneity is what differentiates this approach from the multiple restart strategies of conventional MCDM methods.

The notion of using preferences in this way is not to steer the search toward a specific subset of preferred solutions; rather, they are synthetic preferences that act only to provide discrimination between solutions in high-dimensional objective spaces. Thus, preferences here are not real decision-maker preferences, and can be interpreted as references. This is not new to evolutionary multi-objective optimisation research, but is certainly under-explored. Existing approaches have tended to focus on an aggregation-based formulation of preferences; Jin and Okabe (2001) considered preferences in the form of a weighted-sum, in which the weightings were varied over the course of the search. Hughes (2003) proposed a ranking method based on how each solution performed against a set of pre-defined preferences expressed as weighted min-max formulations. Zhang and Li (2007) developed a cellular algorithm in which each node represents a particular preference formulation, leading to a spatial distribution of preferences. The challenge for these methods is how to define a suitable family of preferences that will produce a full representation of the Pareto front. This issue is slightly different to that faced by multiple restart algorithms, since the focus is now on maintaining the usefulness of the preference family during the course of a population-based search.

A potential way of maintaining the relevance of the preference family as the search progresses is to co-evolve the family together with the usual population of candidate solutions. The solutions would gain fitness by performing well against the preferences (as in the approaches above), and the preferences would gain fitness by offering comparability between solutions. We refer to this type of approach as a preference-inspired co-evolutionary algorithm (PICEA).

In this Chapter we introduce the first instantiation of PICEA, i.e., preference-inspired co-evolutionary algorithm using goal vectors (PICEA-g). Section 3.2 elaborates the proposed algorithm PICEA-g. This is followed by an empirical study of PICEA-g on many-objective optimisation. Experiment description, results and discussions are provided in Section 3.3, Section 3.4 and Section 3.5, respectively. Section 3.6 concludes experiment findings. Section 3.7 proposes a new strategy called the *cutting plane* to further improve the performance of PICEA-g. Section 3.8 summarises this Chapter.

3.2 PICEA-g

3.2.1 Algorithm design: PICEA-g

Harnessing the benefits of co-evolution for optimisation purposes is known to be challenging (Bongard and Lipson, 2005; Kleeman and Lamont, 2006), although there are multi-objective examples, and we are aware of only one existing work that has attempted to implement a concept similar to PICEA. Lohn et al. (2002) considered co-evolving a family of target vectors as a means of improving diversity across the Pareto front. The paper was published shortly before the advent of many-objective optimisation in EMO and the authors did not consider the benefits of the target vectors for improving solution comparability *per se*. However, the paper can certainly be interpreted in such terms. The fitness assignment of Lohn et al.'s method is very interesting and we retain this in our study of the first realisation of a preference-inspired co-evolutionary algorithm: PICEA-g. PICEA-g considers a family of *goals* – a more natural terminology than target vectors when thinking about decision-maker preferences, but the two are essentially equivalent.

According to the fitness assignment in Lohn et al. (2002), candidate solutions gain fitness by dominating a particular set of goal vectors in objective-space, but the fitness contribution from goal vectors is shared between other solutions that also dominate those goals. In order to gain high fitness, candidate solutions need to dominate as many *valuable* goal vectors as possible. *Valuable* goal vectors refer to those which are dominated by few candidate solutions. To do so, candidate solutions must move towards the Pareto optimal front, either being closer to the Pareto optimal front (convergence) or spread in regions where few solutions exist (diversity). Goal vectors only gain fitness by being dominated by a candidate solution, but the fitness is reduced the more times the goal vectors are dominated by other candidate solutions in the population. Therefore, in order to gain high fitness, goal vectors should be placed in regions where only few solutions can dominate it, that is, either being closer to the Pareto optimal front or spread in a sparser region. The overall aim is for the goal vectors to adaptively guide the candidate solutions towards the Pareto optimal front. That is, the candidate solution population and the goal vectors co-evolve towards the Pareto optimal front.

We implement PICEA-g within a $(\mu+\lambda)$ elitist framework[1] shown as Figure 3.1. A population of candidate solutions and preference sets (goal vectors), S and G, of fixed size, N and N_g, are evolved for a number of generations. In each generation t, parents $S(t)$ are subjected to (representation-appropriate) genetic variation operators to produce N offspring, $Sc(t)$. Simultaneously, N_g new goal vectors, $Gc(t)$, are randomly generated. $S(t)$ and $Sc(t)$, and $G(t)$ and $Gc(t)$, are then pooled respectively and the combined populations are sorted according to their fitness. Truncation selection is applied to select the best N solutions as a new parent population $S(t+1)$ and N_g goal vectors as a new preference population $G(t+1)$.

The method to calculate the fitness, Fit_s, of a candidate solution $\mathbf{s}$ and fitness, Fit_g of a goal vector $\mathbf{g}$ is defined by Equations 3.1, 3.2 and 3.3:

$$Fit_s = 0 + \sum_{g \in G \uplus G_C | s \preceq g} \frac{1}{n_g} \tag{3.1}$$

[1] New parents (of size μ) are selected from a combined set of parents (of size μ) and offspring (of size λ).

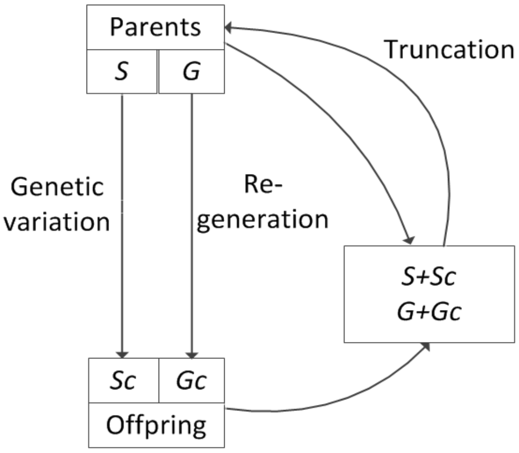


FIGURE 3.1: A (μ+λ) elitist framework of PICEA-g.

where n_g is the number of solutions that satisfy (i.e., dominate) goal vector $\mathbf{g}$ (note that if $\mathbf{s}$ does not satisfy any $\mathbf{g}$ then the Fit_s of $\mathbf{s}$ is defined as 0) and

$$Fit_g = \frac{1}{1+\gamma} \tag{3.2}$$

where

$$\gamma = \begin{cases} 1 & n_g = 0 \\ \frac{n_g-1}{2N-1} & \text{otherwise.} \end{cases} \tag{3.3}$$

In order to further explain the fitness assignment scheme, consider the bi-objective minimisation instance, shown in Figure 3.2, with two candidate solutions $\mathbf{s}_1$ and $\mathbf{s}_3$, their offspring $\mathbf{s}_2$ and $\mathbf{s}_4$, two existing preferences $\mathbf{g}_1$ and $\mathbf{g}_3$, and two new preferences $\mathbf{g}_2$ and $\mathbf{g}_4$ (i.e. $N = N_g = 2$).

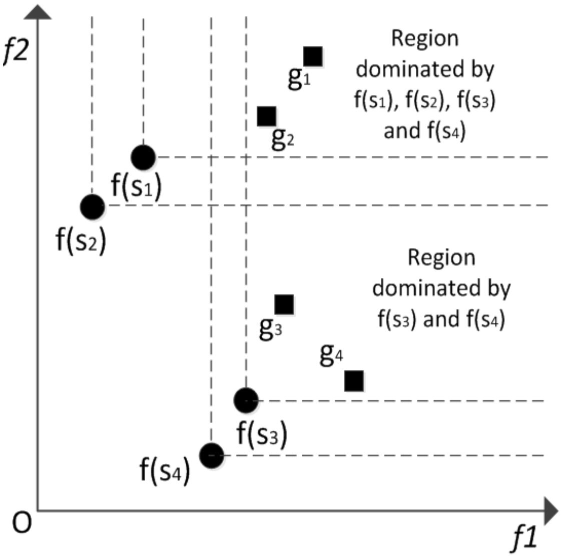


FIGURE 3.2: A simple bi-objective minimisation example.

In Figure 3.2, $\mathbf{g}_1$ and $\mathbf{g}_2$ are each satisfied by $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and $\mathbf{s}_4$ and so $n_{g_1} = n_{g_2} = 4$. $\mathbf{g}_3$ and $\mathbf{g}_4$ are satisfied by $\mathbf{s}_3$ and $\mathbf{s}_4$ only and therefore $n_{g_3} = n_{g_4} = 2$. In terms of fitness of solutions, from Equation 3.1, $Fit_{\mathbf{s}_1} = Fit_{\mathbf{s}_2} = \frac{1}{n_{g_1}} + \frac{1}{n_{g_2}} = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ and $Fit_{\mathbf{s}_3} = Fit_{\mathbf{s}_4} = \frac{1}{n_{g_1}} + \frac{1}{n_{g_2}} + \frac{1}{n_{g_3}} + \frac{1}{n_{g_4}} = \frac{1}{4} + \frac{1}{4} + \frac{1}{2} + \frac{1}{2} = \frac{3}{2}$. Considering the goal vector fitnesses, using Equation 3.2, γ for $\mathbf{g}_1$ and $\mathbf{g}_2$ is $\frac{n_{g_1}-1}{2N-1} = \frac{4-1}{4-1} = 1$ and so, using Equation 3.3, $Fit_{\mathbf{g}_1} = Fit_{\mathbf{g}_2} = \frac{1}{2}$. Similarly, γ for $\mathbf{g}_3$ and $\mathbf{g}_4$ is $\frac{2-1}{4-1} = \frac{1}{3}$ and therefore $Fit_{\mathbf{g}_3} = Fit_{\mathbf{g}_4} = \frac{3}{4}$.

Based on the fitness, $\mathbf{s}_3$ and $\mathbf{s}_4$ are considered as the best solutions, which will be selected into the next generation. However, obviously, $\mathbf{s}_3$ is dominated by $\mathbf{s}_4$. Compared with $\mathbf{s}_3$, although $\mathbf{s}_2$ has a lower fitness, it is non-dominated with $\mathbf{s}_4$. Therefore, $\mathbf{s}_2$ and $\mathbf{s}_4$ are desired to be kept in the population set. In order to do that, the classic Pareto-dominance relation is incorporated. After calculating fitness values using Equations 3.1, 3.2 and 3.3 we next identify all the non-dominated solutions in the set $S \biguplus Sc$. If the number of non-dominated solutions does not exceed the population size, then we assign the maximum fitness to all of the non-dominated solutions. However, if more than N non-dominated solutions are found, we then disregard the dominated solutions prior to applying truncation selection (implicitly, their fitness is set to zero). Based on fitness, the best N non-dominated solutions are selected to constitute the new parent $S(t+1)$. In the example in Figure 3.2, $Fit_{\mathbf{s}_1} = 0$, $Fit_{\mathbf{s}_2} = \frac{1}{2}$, $Fit_{\mathbf{s}_3} = 0$ and $Fit_{\mathbf{s}_4} = \frac{3}{2}$.

The pseudo-code of PICEA-g is presented in Algorithm 1 overleaf. Both the convergence and diversity are taken into account by the fitness assignment. In the following we explain the main steps of PICEA-g.

- Line 1 initialises the offline archive $BestF$ as $\emptyset$.
- In lines 2 and 3, N candidate solutions S are initialised and their objective values F_S are calculated. The offline archive $BestF$ is updated by function `updateArchive` in line 4.
- Line 5 applies function `goalBound` to determine goal vector bounds $GBounds$ for the generation of goal vectors. Line 6 generates N_g goal vectors by function `goalGenerator`.
- The offspring candidate solutions Sc are generated by function `geneticOperation` in line 8, and their objective values F_Sc are calculated in line 9. S and Sc, F_S and F_Sc are pooled together, respectively in line 10. Line 11 generates another set of goal vectors Gc based on the determined $GBounds$, and G and Gc are pooled together in line 12.
- Line 13 applies function `fitnessAssignment` to calculate the fitness of the combined solutions $JointS$ and goal vectors $JointG$.

Algorithm 1: Preference-inspired co-evolutionary algorithm using goals (PICEA-g)

```
Input: Initial candidate solutions, $S$ of size $N$, initial goal vectors, $G$ of size $N_g$,
       maximum number of generations, $maxGen$, the number of objectives, $M$, a
       scaling parameter, $\alpha$, archive size $ASize$
Output: $S, G$, offline archive, $BestF$
1  $BestF \leftarrow \emptyset$;
2  $S \leftarrow$ initializeS($N$) ;
3  $F\_S \leftarrow$ objectiveFunction($S$);
4  $BestF \leftarrow$ updateArchive($BestF, F\_S, ASize$);
5  $GBounds \leftarrow$ goalBound($F\_S, \alpha$);
6  $G \leftarrow$ goalGenerator($N_g, GBounds$) ;
7  for $t \leftarrow 1$ to $maxGen$ do
8      $Sc \leftarrow$ geneticOperation($S$);
9      $F\_Sc \leftarrow$ objectiveFunction($Sc$);
10     $(JointS, JointF) \leftarrow$ multisetUnion($S, Sc, F\_S, F\_Sc$);
11     $Gc \leftarrow$ goalGenerator($N_g, GBounds$) ;
12     $JointG \leftarrow$ multisetUnion($G, Gc$);
13     $(FitJointS, FitJointG) \leftarrow$ fitnessAssignment($JointS, JointG$);
14     find the index, $ixNom$ of all the non-dominated solutions from $JointF$ and count
       the number of non-dominated solutions, $numNomF$;
15     if $numNomF < N$ then
16         $FitJointS(ixNom) \leftarrow$ maxFitness($FitJointS$);
17         $(S, F\_S) \leftarrow$ truncation($JointS, FitJointS, JointF, N$);
18     else
19         $(S, F\_S) \leftarrow$
           truncation($JointS(ixNom), JointF(ixNom), FitJointS(ixNom), N$);
20     end
21     $G \leftarrow$ truncation($JointG, FitJointG, Ngoal$);
22     $BestF \leftarrow$ updateArchive($BestF, F\_S, ASize$);
23     $GBounds \leftarrow$ goalBound($BestF, \alpha$);
24 end
```

- Lines 14 to 21 select the best N candidate solutions and N_g goal vectors as new parents according to their fitness.
- Line 22 updates the offline archive with the F_S by function `updateArchive`.
- Line 23 updates goal vector bounds $GBounds$ based on the offline archive solutions.

Other basic functions

(i) Function `geneticOperation` applies genetic operators to generate offspring Sc. A number of genetic operators are available, for example, single point crossover, uniform crossover, simulated binary crossover (SBX) (Deb and Agrawal, 1994), simplex crossover (SPX), one bit-flip mutation, polynomial mutation (PM) (Deb et al., 2002a), etc.. These genetic operators have their own advantages and disadvantages. In this study the SBX and PM operators are chosen. It is worth mentioning that different genetic operators may lead different algorithm performance for different problems, that is, selecting suitable genetic operators is often algorithm- and problem-dependent (Srinivas and Patnaik, 1994). This is beyond the scope of this research but is worth investigating in future.

(ii) Note that using function `goalGenerator`, goal vectors in Gc are randomly generated as objective vectors directly in objective-space within the goal vector bounds, $GBounds$. $GBounds$ are determined by function `goalBound` based on all offline members, $BestF$. Specifically, the lower bound, $\mathbf{g}_{min}$, and the upper bound, $\mathbf{g}_{max}$, are estimated by Equation 3.4:

$$\begin{aligned} \mathbf{g}_{min} &= \min\left(BestF_i\right), \quad i = 1, 2, \cdots, M \\ \Delta F_i &= \max\left(BestF_i\right) - \min\left(BestF_i\right) \\ \mathbf{g}_{max} &= \min\left(BestF_i\right) + \alpha \times (\Delta F_i), \quad \alpha \geq 1, i = 1, 2, \cdots, M \end{aligned} \tag{3.4}$$

where $\alpha = 1.2$ is suggested. A further discussion on the configuration of parameter α is provided in Section 3.5.

Genetic operators are not applied to generate offspring goal vectors. This is because, based on our preliminary experiments, none of the classic genetic operators, such as SBX and SPX, work more effectively than the random method. A possible reason is due to the observation that applying genetic operators to goal vectors often produces arbitrary goal vectors. This is similar to the general observation that, recombining two dissimilar candidate solutions often does not produce a fruitful solution; for this reason, mating restriction is often considered in genetic operations (Ishibuchi et al., 2008a)). Although the random method might also generate some non-useful goal vectors, it guarantees that goal vectors in the entire objective-space are generated, and this helps the algorithm find solutions in all regions of the Pareto optimal front. It is expected that, in future, some effective genetic operators can be developed and so can be applied to goal vectors, further improving the performance of PICEA-g.

(iii) Function `updateArchive` updates the offline archive $BestF$ by F_S. For each solution (e.g. $\mathbf{F\_s}_i$) in the F_S, if $\mathbf{F\_s}_i$ is dominated by a solution in the archive, then $\mathbf{F\_s}_i$ is rejected. Otherwise it is accepted as a new archive member. Simultaneously, solutions in the archive that are dominated by $\mathbf{F\_s}_i$ are removed. When the number of archive solutions $cASize$ exceeds the archive size $ASize$, the clustering method employed in SPEA2 (Zitzler et al., 2002) (i.e., an archive truncation strategy) is invoked which iteratively removes solutions from the archive until $cASize = ASize$, and simultaneously maintains a set of evenly distributed solutions. The clustering method is described as follows: at each iteration, a member in the archive $\mathbf{F\_s}_i$ is chosen for removal for which $\mathbf{F\_s}_i \leq_d \mathbf{F\_s}_j$ for all $\mathbf{F\_s}_j \in BestF$ with

$$\mathbf{F\_s}_i \leq_d \mathbf{F\_s}_j :\Leftrightarrow \forall 0 < k < cASize : dist_i^k = dist_j^k \vee \\ \exists < k < cASize : [(\forall 0 < l < k : dist_i^l = dist_j^l) \wedge (dist_i^k < dist_j^k)] \tag{3.5}$$

where $dist_i^k$ denotes the distance of $\mathbf{F\_s}_i$ to its k-th nearest neighbour in $BestF$. Equation 3.5 means that the most crowded solution (the one that has the minimum distance to another solution) is chosen for removal at each stage; if there are multiple solutions with minimum distance, the tie is broken by considering the second smallest distances, and so on.

With respect to the time complexity of PICEA-g, evaluation of a population of candidate solutions runs at $\mathcal{O}(M \times N)$, where M is the number of objectives and N is the number of candidate solutions. Fitness assignment for candidate solutions and goal vectors which needs a crossed comparison (to determine which candidate solution dominates which goal vector) runs at $\mathcal{O}(M \times N \times N_g)$. Therefore, the overall time complexity of PICEA-g is $\mathcal{O}(M \times N \times N_g)$. When candidate solutions are evaluated by the same number of goal vectors, the time complexity of PICEA-g is $\mathcal{O}(M \times N^2)$ which is equivalent to the running time of NSGA-II (Deb et al., 2002a).

3.2.2 Algorithm analysis: PICEA-g

Co-evolution in PICEA-g

In PICEA-g, candidate solutions and goal vectors are co-evolved during the search process. In the co-evolution, candidate solutions act as predators and goal vectors act as preys. Candidate solutions try to catch (i.e., dominate) more goal vectors while goal vectors try to avoid being caught (i.e., dominated) by candidate solutions. Goal vectors act like a mechanical rabbit – keeping a certain distance from the dogs but never getting too far ahead or too close (Kleeman and Lamont, 2006).

Convergence: comparability with objective scaling

The use of goal vectors enables incomparable solutions (i.e. non-dominated solutions) to become comparable, thus generating selective pressure towards the Pareto optimal front and enabling a good proximity performance of PICEA-g. More specifically, whilst the use of goals does rely on Pareto-dominance comparison at the level of the individual goal, the presence of multiple goals can significantly mitigate the comparability issues observed when scaling the standard dominance relation. To see this, consider a population of 100 objective vectors randomly generated in the hypercube $(0, 1]^M$, where M is the dimension of objective space (number of objectives), and a direction of preference in each objective. We sort the 100 individuals into equivalence classes using a global Pareto-dominance relation (non-dominated sorting) and also using the fitness assignment scheme in PICEA-g for three populations of randomly generated goal vectors (of size 20, 100 and 500). We repeat our experiments 500 times and calculate the mean number of equivalence classes for each of the four approaches. The results are shown in Figure 3.3. It is evident that by using goals, a substantially greater level of comparability can be achieved than by using global Pareto-dominance. The more goals that are used, the greater the comparability that is achieved. Whilst the number of equivalence classes does reduce in the goal scheme as the number of objectives is increased, by 10 objectives a 100-goal approach (i.e. matched to the number of objective vectors) is still able to provide greater comparability than the global Pareto approach in 3 objectives. This provides some reassurance that the method has potential for many-objective optimisation, since Pareto-dominance based algorithms tend to still work well for 3-objective problems.

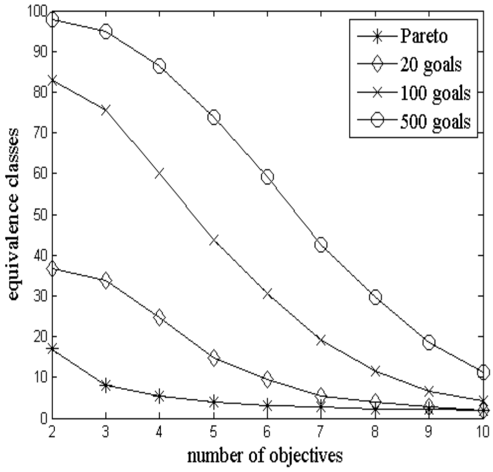


FIGURE 3.3: Changes in comparability with objective scaling: Pareto-dominance and goal approaches.

Implicitly, the use of goal vectors acts like a modified dominance relation – considering both the number of objectives for which one candidate solution is better than another and the amount of difference in objective values between two candidate solutions (Sülflow et al., 2007; Corne and Knowles, 2007). In Figure 3.4, $\mathbf{s}_1$ and $\mathbf{s}_2$ are non-dominated with each other. However, $\mathbf{s}_1$ dominates a larger region and therefore is more likely to dominate more goal vectors than $\mathbf{s}_2$ does. Therefore, $\mathbf{s}_1$ is likely to have a higher fitness than $\mathbf{s}_2$. Regarding the dominance relation, for a minimisation problem $\mathbf{s}_1$ is a bit worse than $\mathbf{s}_2$ in objective f_1 while it is much better than $\mathbf{s}_2$ in objective f_2. Therefore, $\mathbf{s}_1$ is more likely to be considered as a better solution by the modified dominance relation, e.g., the *favour* relation (Sülflow et al., 2007).

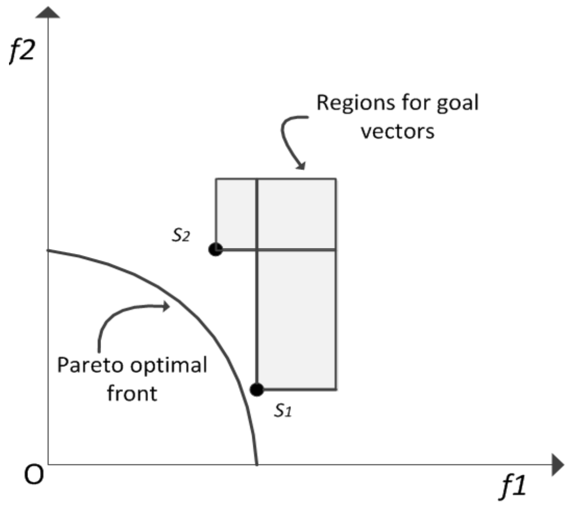


FIGURE 3.4: Convergence analysis of PICEA-g.

Diversity: fitness sharing

The diversity performance of PICEA-g is maintained by introducing the fitness sharing method (Rosin and Belew, 1997; Della Cioppa et al., 2007) into the fitness assignment of PICEA-g. That is, a candidate solution (e.g., $\mathbf{s}_i$) only receives $\frac{1}{n_g}$ credit from a goal vector $\mathbf{g}_i$ that is dominated by $\mathbf{s}_i$. n_g is the number of solutions that dominate $\mathbf{g}_i$. To be more specific, let us consider an example, as shown in Figure 3.5(a), to describe the fitness contribution of goal vectors in different regions. The fitness contribution of goal vectors in G_3 is shared by all the three candidate solutions, that is, each candidate solution receives $\frac{1}{3}$ credit from a goal vector in G_3. Similarly, each candidate solution receives $\frac{1}{2}$ credit from a goal vectors in G_2. The fitness contribution of goal vectors in G_1 is exclusively attributed to every single solution. Using the fitness sharing, goal vectors are implicitly labelled with different credits. The higher the credit labelled to a goal vector, the more challenging the goal vector to be dominated. That is, goal vectors in G_1 are more challenging compared with goal vectors in G_2. A solution should gain more credits by dominating a challenging goal vector. This is somehow more reasonable in nature.

Assuming that we have obtained four solutions as shown in Figure 3.5(b). $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$ are close to each other while $\mathbf{s}_4$ is distant from the three solutions. Disregarding the fitness sharing, $\mathbf{s}_4$ might not be selected as a new parent. This is because the credit gained by dominating an individual goal vector in different G_i is the same; and so $\mathbf{s}_4$ might not obtain higher fitness than the other solutions. However, considering the fitness sharing, $\mathbf{s}_4$ would gain more credits by dominating those challenging goal vectors in G_1, which results in higher fitness and so is more likely to be selected into the next generation. That is to say, using fitness sharing, solutions in a sparse region is more likely to survive and this improves the solution diversity.

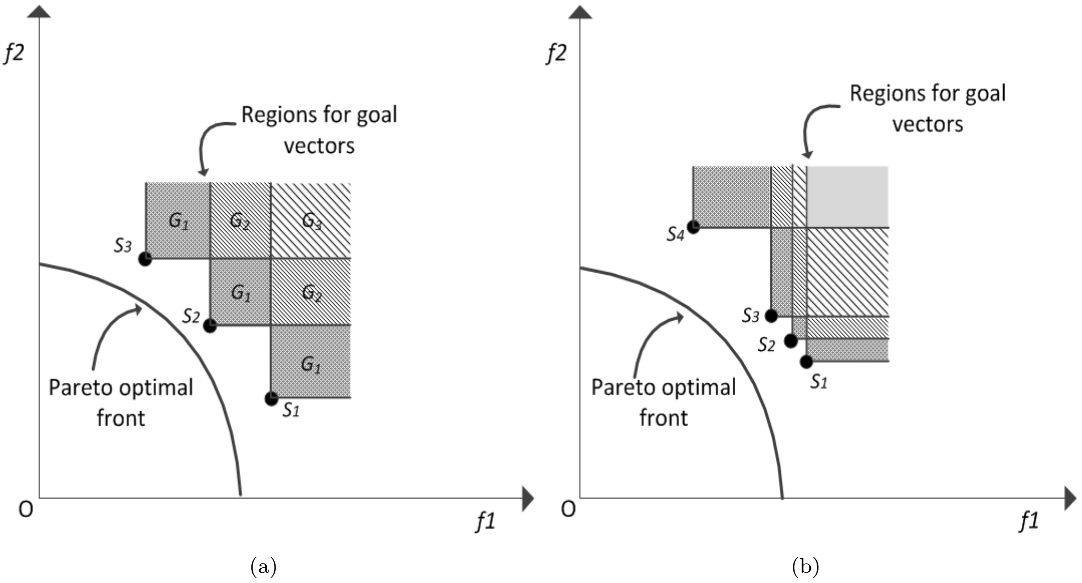


Figure 3.5: Diversity analysis of PICEA-g.

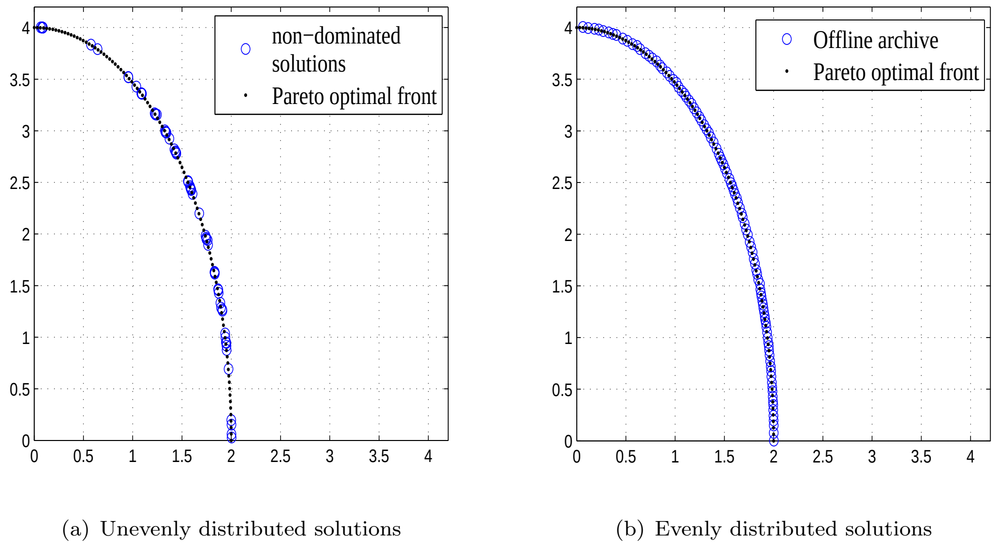
The use of archive

The fitness assignment applied in PICEA-g is susceptible to unevenness in the distribution. Neighbouring candidate solutions (e.g. $\mathbf{s}_1$ and $\mathbf{s}_2$, or $\mathbf{s}_3$ and $\mathbf{s}_4$) in general are of high probability to dominate similar goal vectors and so to gain similar fitness values. Therefore, neighbouring solutions may either be selected or disregarded at the same time (based on the truncation selection used in PICEA-g). This results in a set of unevenly distributed solutions along the Pareto optimal front as shown in Figure 3.6(a). Referring to the example shown in Figure 3.7, candidate solutions $\mathbf{s}_1$ and $\mathbf{s}_2$ are adjacent to one another and have the same fitness value $\frac{1}{2}$. Similarly, $\mathbf{s}_3$ and $\mathbf{s}_4$ gain the same fitness value $\frac{3}{2}$. $\mathbf{s}_3$ and $\mathbf{s}_4$ are selected in the end due to their high fitness. However, it is easy to see that a better distribution can be achieved if $\mathbf{s}_1$ or $\mathbf{s}_2$ is selected.

To handle this limitation, we propose to maintain an archive to store all non-dominated solutions found during the search. The clustering technique used in SPEA2 is then applied to obtain a user-defined size of evenly distributed solutions (Zitzler et al., 2002). Figure 3.6(b) shows a set of evenly distributed solutions obtained by applying the clustering technique to the offline archive solutions.

3.3 Experiment description

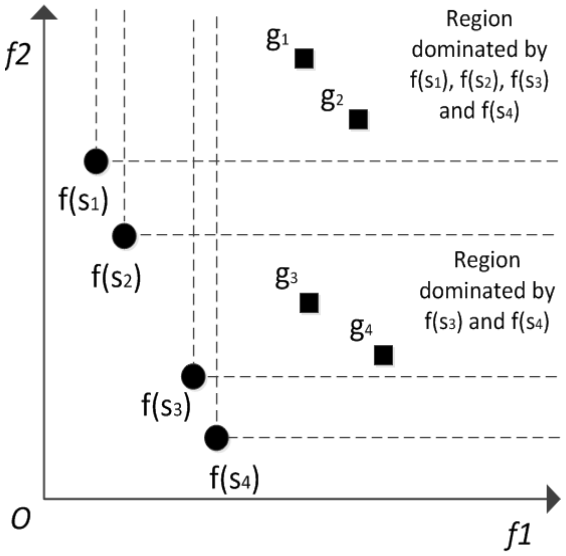
In this section, we explore the potential of the PICEA concept by comparing PICEA-g with four best-in-class MOEAs: the ε-dominance based algorithm, ε-MOEA (Deb et al., 2003), an indicator based algorithm, HypE (Bader and Zitzler, 2011), a decomposition



(a) Unevenly distributed solutions

(b) Evenly distributed solutions

Figure 3.6: Illustration of the limitation of the fitness assignment.



f2
f1
O
g₁
g₂
g₃
g₄
f(s₁)
f(s₂)
f(s₃)
f(s₄)
Region dominated by f(s₁), f(s₂), f(s₃) and f(s₄)
Region dominated by f(s₃) and f(s₄)

Figure 3.7: An example to illustrate the limitation of fitness assignment of PICEA-g.

based algorithm, MOEA/D (Zhang and Li, 2007) and the Pareto-dominance based algorithm, NSGA-II (Deb et al., 2002a), along with random search which is included as a baseline[2]. First, the selected test problems are presented. Then, we briefly introduce each of the competitor algorithms, together with some details of the specific implementations adopted for the comparison. Furthermore, the general parameter settings are shown. Lastly, the performance assessment is introduced.

[2] The ranking based approaches are not included in this comparison as Purshouse et al. (2011) has demonstrated that Lohn et al's method outperforms the average ranking based MOEA (one of the most promising ranking based approaches (Corne and Knowles, 2007) on MaOPs). Also, the objective reduction based approaches are not considered as competitor MOEAs. This is because all the selected test problems have high dimensions in objective-space and are highly conflicting.

3.3.1 Test problems

To benchmark the performance of the considered MOEAs, problems 2 to 9 from the Walking Fish Group (WFG) test suite (Huband et al., 2006) are invoked in 2, 4, 7 and 10-objective instances. In each case, the WFG parameters k and l are set to 18 and 14, providing a constant number of decision variables for each problem instance. Problem attributes covered include separability/non-separability, unimodality/multimodality, unbiased/biased parameters and convex/concave geometries. Please note that WFGn-Y refers to problem WFGn with Y objectives.

According to Huband et al. (2006), the Pareto optimal front of WFG2 is not a regular geometry (disconnected and convex), the Pareto optimal front of WFG3 is a linear line, and the Pareto optimal front of WFG4-WFG9 is the surface of an M-dimension hyper-ellipsoid with radius $r_i = 2i$ in the first quadrant. The *ideal* point and the *nadir* point of these problems are $[0, 0, \cdots, 0]$ and $[2, 4, \cdots, 2M]$, respectively.

3.3.2 The considered competitor MOEAs

ε-Multi-objective evolutionary algorithm: ε-MOEA

Laumanns et al. (2002) proposed the ε-MOEA algorithm, in which the ε-dominance concept is applied. Both the convergence property and the diversity property of this algorithm can be maintained by the setting of an appropriate value for ε. The objective space is divided into a grid of hyperboxes, whose size can be adjusted by the choice of ε. For each hyperbox that contains a solution (or solutions), the dominance of the hyperbox is checked. An archive strategy, suggested in Deb et al. (2003), is applied in ε-MOEA and is used to retain one solution for each non-dominated hyperbox. The specific dominance checking process is explained as follows (see Figure 3.8): firstly, if the hyperbox of a new solution (C) dominates another hyperbox (D) in the archive, the dominated archive members (D) are rejected. Secondly, if there is more than one solution in the same hyperbox (A, B), the dominated solutions are removed (B). Thirdly, if there is more than one non-dominated solution in a hyperbox (E, F), one of them is randomly selected. For the third step, Deb et al. (2005) suggested choosing the solution (E) which is the closest to the origin of the hyperbox.

In this study, ε-MOEA of Deb et al. (2005) is used and, for each test instance, different ε values are used[3] in order to obtain roughly 100 solutions after an allowed number of

[3] $M = 2$: WFG2: $\varepsilon =$ (0.004,0.008), WFG3: $\varepsilon =$ (0.0133,0.0266), WFG4 to WFG9: $\varepsilon =$ (0.02,0.04); $M = 4$: WFG2: $\varepsilon =$ (0.0500 0.1000 0.1500 0.2000), WFG3: $\varepsilon=$ (0.1000 0.2000 0.3000 0.4000), WFG4 to WFG9: $\varepsilon =$ (0.2857 0.5714 0.8571 1.1429); $M = 7$: WFG2: $\varepsilon=$ (0.0500 0.1000 0.1500 0.2000 0.2500 0.3000 0.3500) WFG3: $\varepsilon=$ (0.2000 0.4000 0.6000 0.8000 1.0000 1.2000 1.4000) WFG4 to WFG9: $\varepsilon =$ 0.4000 0.8000 1.2000 1.6000 2.0000 2.4000 2.8000) $M = 10$: WFG2: $\varepsilon=$(0.0571 0.1143 0.1714 0.2286 0.2857 0.3429 0.4000 0.4571 0.5143 0.5714) WFG3: $\varepsilon=$(0.1333 0.2667 0.4000 0.5333 0.6667 0.8000 0.9333

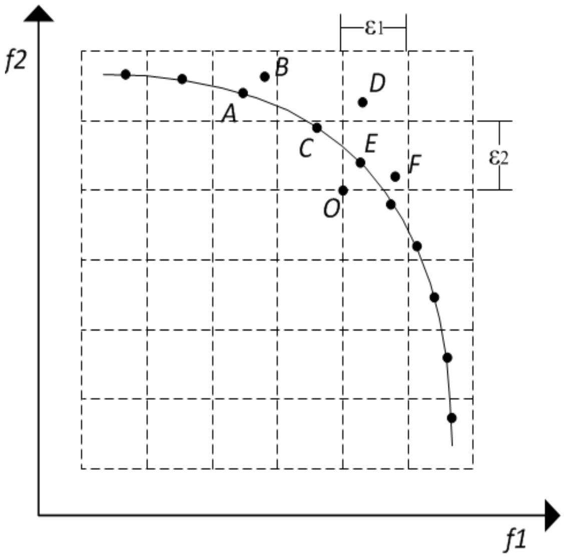


FIGURE 3.8: Illustration of the ε-dominance concept.

function evaluations (25 000 in this study). The value of ε varies with each objective, reflecting the scaling of the selected benchmark functions.

Indicator based evolutionary algorithm: HypE

Zitzler and Künzli (2004) proposed a general indicator based evolutionary algorithm (IBEA). Hypervolume, which has good properties with respect to set-based dominance comparisons (Zitzler et al., 2003), is often taken as an indicator in IBEA. However, the high computational effort required for its calculation inhibits the full exploitation of its potential (Beume et al., 2009). HypE proposed by Bader and Zitzler (2011) uses a hypervolume estimation algorithm for multi-objective optimisation (Monte Carlo simulation to approximate the exact hypervolume values), by which the accuracy of the estimates can be traded off against the available computing resources. There is evidence that the approach can be effective for many-objective problems. In the same way as a standard multi-objective evolutionary algorithm, it is based on fitness assignment schemes, and consists of successive application of mating selection, variation and environmental selection. The hypervolume indicator is applied in environmental selection.

In HypE the hypervolume-based fitness of a solution is not only calculated based on its own hypervolume contribution, but also the hypervolume contribution associated with other solutions. This is illustrated in Figure 3.9, where the portion of hypervolume that is dominated by **a** is fully attributed to **a**, the portion of hypervolume that is dominated by **a** and another solution **c** is attributed half to **a**. For example, the hypervolume contribution of the solution **a** is described as $Hyp(\mathbf{a})$ shown in Figure 3.9. Note that this is

1.0667 1.2000 1.3333) WFG4 to WFG9: $\varepsilon=$ (0.5000 1.0000 1.5000 2.0000 2.5000 3.0000 3.5000 4.0000 4.5000 5.0000) a more refined approach than that adopted in the other hypervolume-based approaches, such as the $\mathcal{S}$ metric selection based evolutionary multi-objective optimisation algorithm (SMS-EMOA) (Beume et al., 2007), in which contribution calculations are limited to single solutions, without consideration of the wider population context.

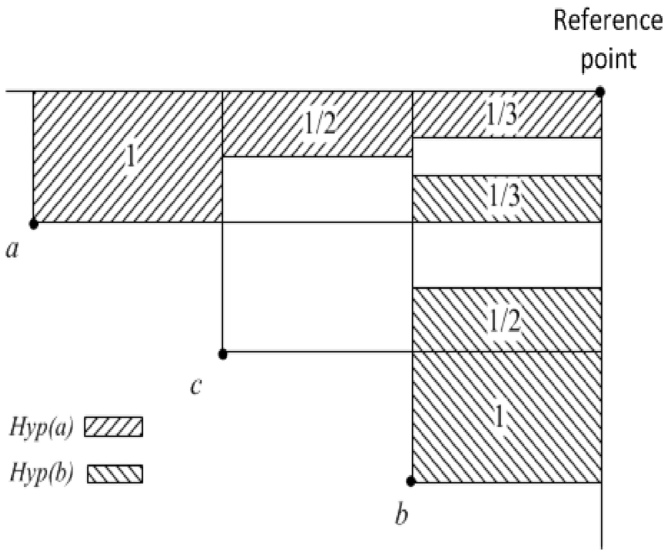


FIGURE 3.9: Illustration of the calculation of hypervolume contribution in HypE.

In this comparison study, we strictly follow the hypervolume contribution calculation method described in Bader and Zitzler (2011). On 2-objective problems, the exact hypervolume contribution of each solution is calculated. On 4-, 7- and 10-objective test instances, a Monte Carlo simulation method with 2000, 3500 and 5000 sampling points, respectively, is used to calculate the estimated hypervolume contribution.

Decomposition based algorithm: MOEA/D

MOEA/D, proposed by Zhang and Li (2007), is a simple yet powerful MOEA. It has a number of advantages over Pareto-dominance based algorithms such as its high search ability for combinatorial optimisation, computational efficiency of fitness evaluation, and high compatibility with local search (Zhang and Li, 2007; Li and Zhang, 2009; Ishibuchi et al., 2009a). The main characteristic feature of MOEA/D is the handling of a multi-objective problem as a collection of a number of single-objective problems (SOPs), which are defined by a scalarising function (e.g. weighted-sum or weighted Chebyshev) with different weight vectors. Each scalarising fitness function (defined by a specific weight vector) identifies a single solution which is the best with respect to that scalarising fitness function. For each SOP, a new solution is generated by performing genetic operators on several solutions from amongst its neighbours. Neighbours are defined based on the distance between the weight vectors. A SOP i is a neighbour of SOP j if the weight vector of SOP i is close to that of SOP j. The newly generated solution is compared with all of its neighbours. If the new solution is better, then some (or all) of its corresponding neighbours are replaced by the new solution. At the same time, the diversity of solutions is maintained by a number of uniformly distributed weight vectors in MOEA/D.

The weight vectors are generated according to Equations 3.6 and 3.7:

$$w_1 + w_2 + \cdots + w_M = 1 \tag{3.6}$$

$$w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \cdots \frac{H}{H}\right\}, \quad i = 1, 2, \cdots, M \tag{3.7}$$

where H is a user-definable positive integer, and M is the number of objectives. The number of weight vectors is calculated as $N_{wv} = C_{H+M-1}^{M-1}$, where C stands for the combination formula. For example, for 2-objective problems, if H is specified as 100, then we can generate $C_{101}^{1} = 101$ groups of weight vectors (0, 1), (0.01, 0.99), ... , (1, 0). Since each individual has a different weight vector, the population size is the same as the number of weight vectors. In the first MOEA/D version (Zhang and Li, 2007), the new solution will replace all the neighbours that are worse than itself. However, in order to maintain a better diversity, in Li and Zhang (2009), an upper bound is defined to limit the maximum number of replacements. In our experiments, the weighted Chebyshev scalarising function is used. The reference point z_i^* is updated according to Equation 3.8:

$$z_i^* = \min\left\{f_i(\mathbf{x})|\mathbf{x} \in \mathbf{\Omega}\right\}, \quad i = 1, 2, \cdots, M \tag{3.8}$$

where $\mathbf{\Omega}$ shows all the examined solutions during the optimisation. The reference point, $\mathbf{z}^*$ is updated once a better (smaller) value of f_i is found.

Population size and other required parameters in MOEA/D are set as shown in Table 3.1.

Table 3.1: Parameters for MOEA/D.

M	Population size (the number of weight vectors)	H
2	100	99
4	455	12
7	924	6
10	2002	5

Disregarding the stopping criterion (i.e. a fixed number of function evaluations) it is obvious that the more weight vectors used in MOEA/D, the better the performance that the algorithm can achieve. However, given a fixed number of function evaluations, it is not straightforward to decide how many groups of weight vectors are appropriate for each problem. In this comparison study, 100 and 455 groups of weight vectors are chosen for 2 and 4-objective WFG tests, as they are commonly used in MOEA/D studies (Ishibuchi et al., 2009c; Li and Zhang, 2009). However, on 7 and 10-objective

WFG tests, no related suggestions are given in the literature. The conventional method for deriving the H parameter for 2 and 4-objective problems is not appropriate for 7 and 10-objective cases due to the extremely large population size that results. For example, choosing $H = 12$ for 7-objective problems makes the population size as large as $C_{18}^{6} = 18564$. Therefore we performed a limited search through the parameter space of the H parameter to find more appropriate configurations of the MOEA/D weight vectors. Our choice of H then leads to the population sizes of 924 and 2002. No specific information is provided in the literature concerning the selection of neighbourhood size, T, and replacement neighbourhood size, nr, except that, in Li and Zhang (2009), the authors point out that T should be much smaller than the population size and nr should be much smaller than T. In this comparison study, therefore, $T = 10$ and $nr = 2$ are used for all the problems[4].

Pareto-dominance based evolutionary algorithm: NSGA-II

A wide variety of algorithms have been proposed, based on Pareto-dominance comparisons supplemented with diversity enhancement mechanisms. The most popular of these methods – the seminal NSGA-II algorithm (Deb et al., 2002a) – is selected in the study as representative of this class. NSGA-II is known to perform well on bi-objective problems but may experience difficulties in many-objective spaces (Purshouse and Fleming, 2003c, 2007). It is an elitist approach; the parent and offspring population are combined and evaluated, using a fast non-dominated sorting approach and an efficient crowding scheme. When more than N population members of the combined population belong to the non-dominated set, only those that are maximally apart from their neighbours according to the crowding measure are chosen.

Random search: rand

Evidence exists that random search can be competitive to evolutionary approaches in many-objective spaces (Hughes, 2005), making this a natural benchmark, at present, for comparison against any proposed new algorithm. A very crude random scheme is implemented in which $N \times maxGen$ candidate solutions are randomly generated and the dominated solutions are removed.

[4] It is demonstrated in Wang et al. (2013) that such setting yields good results on the selected WFG test problems.

3.3.3 General parameters

For each test problem, 31 runs of each algorithm test (each run for 25 000 function evaluations, this is to permit reasonable execution times, while providing meaningful results.) are performed in order to subject to statistical analysis. Some general parameter settings are listed in Table 3.2 and are fixed across all algorithm runs.

Table 3.2: Algorithm testing parameter settings.

Parameters	values
Population size of candidate solutions, N	100
The number of goal vectors, N_g	$100M$
The number of decision variables, n	$n = k + l = 32$
Crossover operator	SBX ($p_c = 1, \eta_c = 15$)
Mutation operator	PM ($p_m = 1/n, \eta_m = 20$)

The simulated binary crossover (SBX) and polynomial mutation (PM), as described in Deb et al. (2002a), are applied. The recombination probability p_c is set to 1 per individual and mutation probability p_m is set to $\frac{1}{n}$ per decision variable. The distribution indices $\eta_c = 15$ and $\eta_m = 20$.

3.3.4 Performance assessment

Firstly, median attainment surfaces (Fonseca and Fleming, 1996) are plotted to visualise the performance of algorithms on 2-objective instances. Secondly, approximation set comparisons are made using the hypervolume metric (HV). A favourable hypervolume (larger, for a minimization problem) implies good proximity with diversity.

Prior to calculating the hypervolume metric we normalise all objectives within the range $[0, 1]$ using the *nadir* vector. The reference point for calculating the hypervolume is chosen as $(1.2, 1.2, \cdots, 1.2)$. The software developed by Fonseca et al. (2006) is used to calculate the hypervolume. Note that throughout the thesis, the hypervolume value of an approximation set is computed based on the normalised objectives.

Additionally we also express the performance of each algorithm in terms of the proportion of the globally optimal hypervolume achieved (the method used to calculate optimal hypervolume depends on which WFG problem is being considered, and is described further in the Appendix B).

Performance comparisons between algorithms are made according to a rigorous non-parametric statistical framework, drawing on recommendations in Zitzler et al. (2003). The initial populations of candidate solutions are generated randomly for every replication of every algorithm on every problem instance – 31 replications are executed for each algorithm-instance pair. The approximation sets used in the comparisons are the members of the offline archive of all non-dominated points found during the search, since this is the set most relevant to *a posteriori* decision-making. For reasons of computational feasibility, prior to analysis the set is pruned to a maximum size of 100 using the SPEA2 truncation procedure (Zitzler et al., 2002).

For each problem instance, the performance metric values of each algorithm are calculated for each approximation set. Then the non-parametric statistical approach introduced in Hollander and Wolfe (1999) is conducted with the performance metric values. We first test the hypothesis that all algorithms perform equally using the Kruskal-Wallis test. If this hypothesis is rejected at the 95% confidence level, we then consider pair-wise comparisons between the algorithms using the Wilcoxon-ranksum two-sided comparison procedure at the 95% confidence level, employing the Šidák correction to reduce Type I errors (Curtin and Schulz, 1998).

3.4 Experiment results

In this section, we present and discuss the experimental results obtained by PICEA-g and the algorithms on the WFG benchmark problems. A comparison of the results, in terms of attainment surface and hypervolume, is provided in this section.

3.4.1 Attainment surface results

Considering as a starting point the bi-objective results, plots of median attainment surfaces across the 31 runs of each algorithm are shown in Figure 3.10 and Figure 3.11. These allows visual inspection of performance in terms of the dual aims of proximity to and diversity across the global trade-off surface. For clarity, PICEA-g, HypE, NSGA-II and MOEA/D are plotted on the left, while NSGA-II, ε-MOEA, and rand are plotted on the right. (NSGA-II serves as a cross-reference). A colour reproduction is available at `http://www.sheffield.ac.uk/acse/staff/rstu/ruiwang/index`.

Qualitatively, from inspection of Figure 3.10 and Figure 3.11, it is clear that rand is the worst performer. All the other MOEAs have comparable performances on convergence with different performances on diversity. Specifically, MOEA/D exhibits the best performance. PICEA-g and HypE have equivalent performances and both algorithms are slightly better than NSGA-II on all the benchmark functions. ε-MOEA can provide as good proximity as NSGA-II but its diversity performance is sometimes inferior. Upon closer examination, on WFG5-2, 6-2, 7-2 and 8-2, all the algorithms exhibit difficulties in converging to the true Pareto front. On WFG8-2, none of the algorithms considered is able to provide a good representation of the trade-off surface. On WFG9-2, only MOEA/D can converge to the most part of the Pareto front. From the results of WFG5-2 to WFG9-2 we can see that bi-objective problems featuring strong multi-modality or non-separable parameters still present a challenge for best-in-class MOEAs.

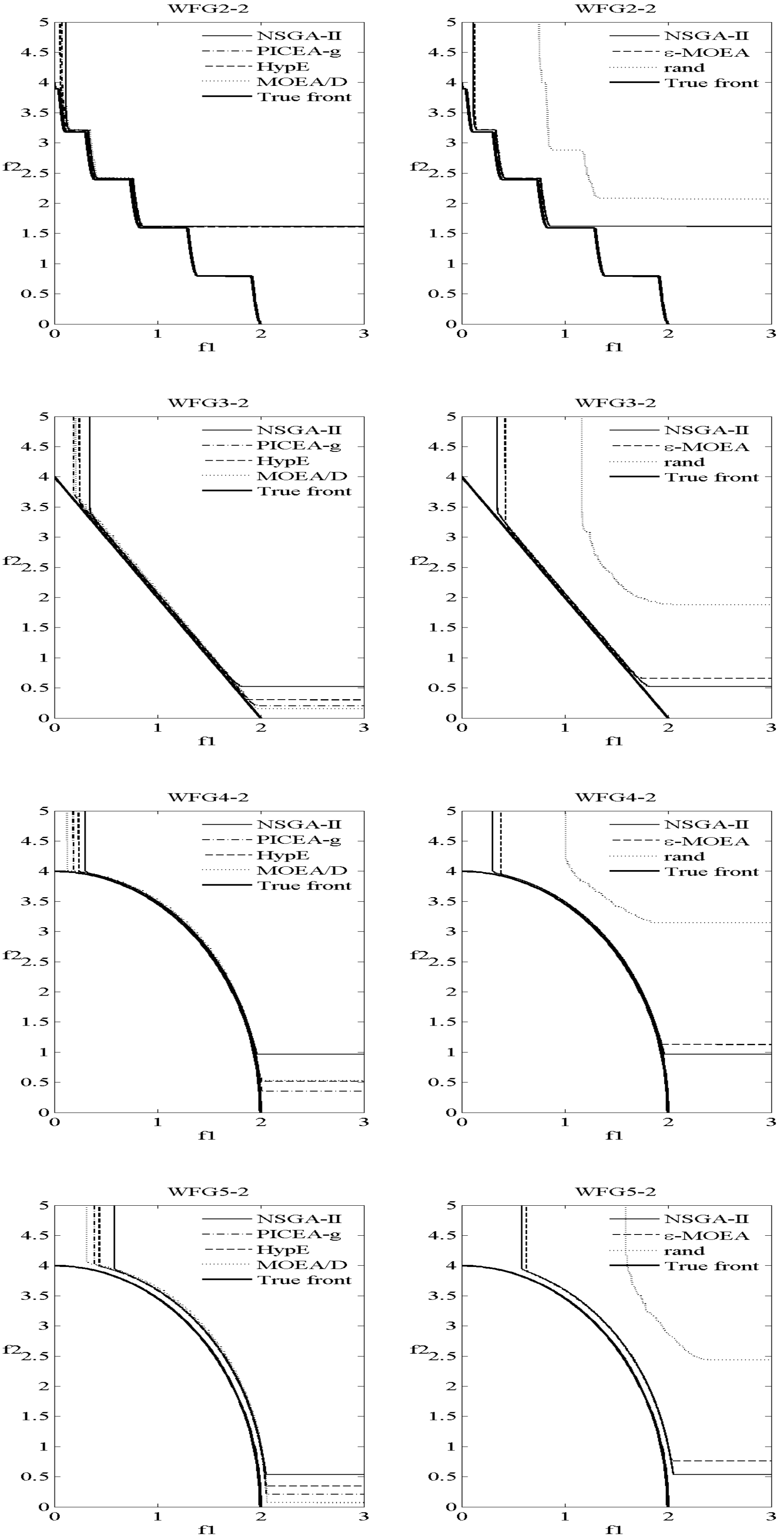


FIGURE 3.10: Attainment surfaces for 2-objective WFG test instances (a: WFG2-2 to WFG5-2).

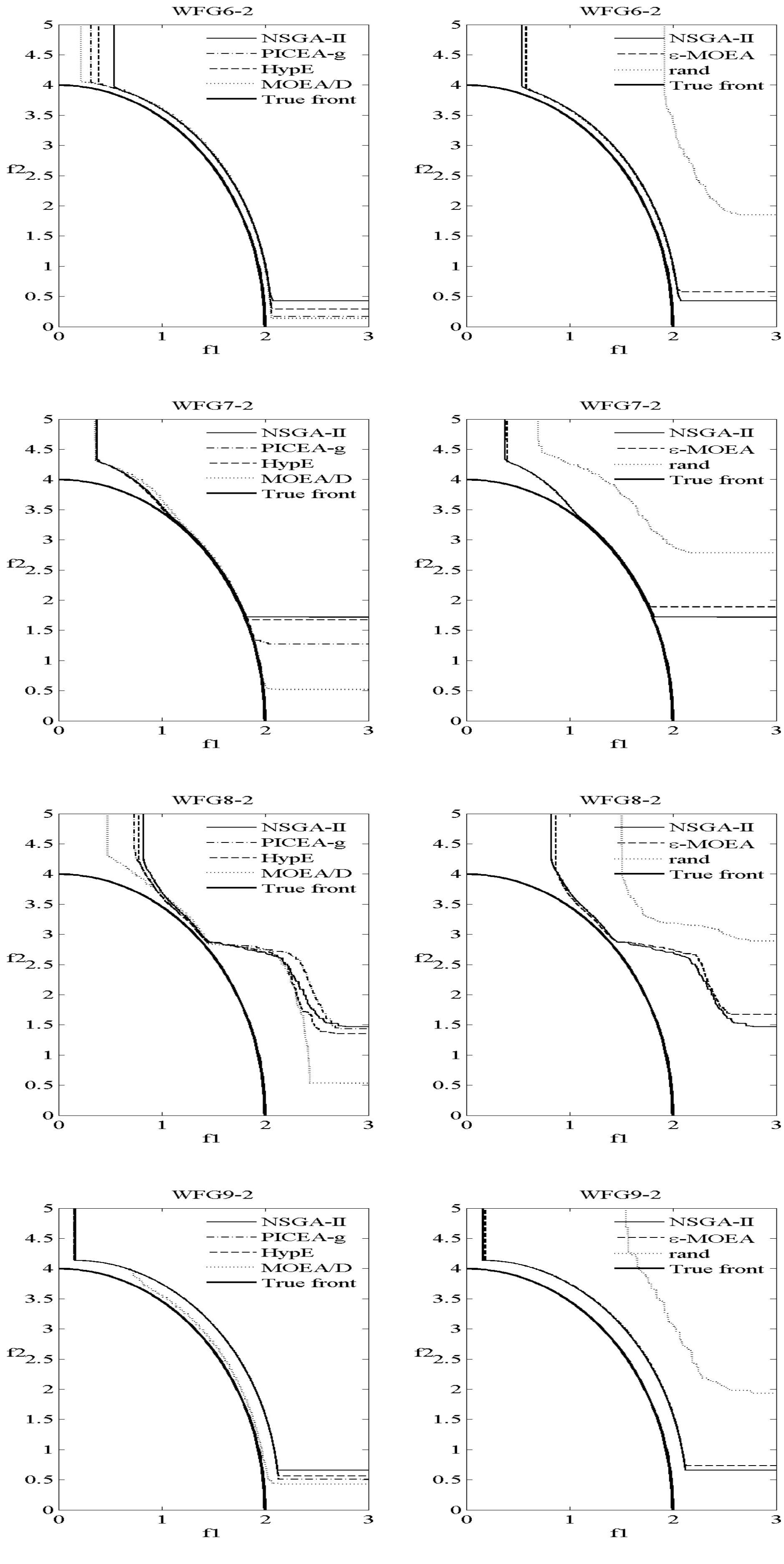


FIGURE 3.11: Attainment surfaces for 2-objective WFG test instances (b: WFG6-2 to WFG9-2).

3.4.2 Statistical results

Results of the Kruskal-Wallis tests followed by pair-wise Wilcoxon-ranksum plus Šidák correction tests based on the hypervolume metric are provided in this Section. The initial Kruskal-Wallis test breaks the hypothesis that all six algorithms are equivalent. Therefore the outcomes of pair-wise statistical comparisons for 2, 4, 7 and 10-objective WFG problems are shown in Tables 3.3, 3.4, 3.5 and 3.6 respectively. The related partial ordering of algorithms is constructed using the method previously described – again a smaller rank value indicates better performance; ordering within a rank is purely alphabetical.

Box plots of Figures 3.12, 3.13, 3.14 and 3.15 are used to visualize the distribution of the 31 hypervolume values for the associated problems. The upper and lower ends of the box are the upper and lower quartiles, while a thick line within the box encodes the median. Dashed appendages summarize the spread and shape of the distribution. Outlying values are marked as '+'. The box plots allow us to consider absolute performance (in terms of the proportion of the optimal hypervolume achieved by each algorithm) in addition to hypothesis testing around relative performance.

***HV* results for 2-objective WFG problems.**

TABLE 3.3: *HV* results for 2-objective instances.

WFG		Ranking by *HV*	WFG		Ranking by *HV*
2	1	HypE PICEA-g	3	1	MOEA/D PICEA-g
	2	ε-MOEA MOEA/D NSGA-II		2	HypE
	3	rand		3	NSGA-II
				4	ε-MOEA
				5	rand
4	1	HypE MOEA/D PICEA-g	5	1	MOEA/D
	2	NSGA-II		2	PICEA-g
	3	ε-MOEA		3	HypE
	4	rand		4	NSGA-II
				5	ε-MOEA
				6	rand
6	1	MOEA/D	7	1	MOEA/D
	2	PICEA-g		2	PICEA-g
	3	HypE		3	HypE NSGA-II
	4	NSGA-II		4	ε-MOEA
	5	ε-MOEA		5	rand
	6	rand			
8	1	MOEA/D	9	1	MOEA/D
	2	HypE		2	ε-MOEA HypE NSGA-II PICEA-g
	3	ε-MOEA NSGA-II PICEA-g		3	rand
	4	rand			

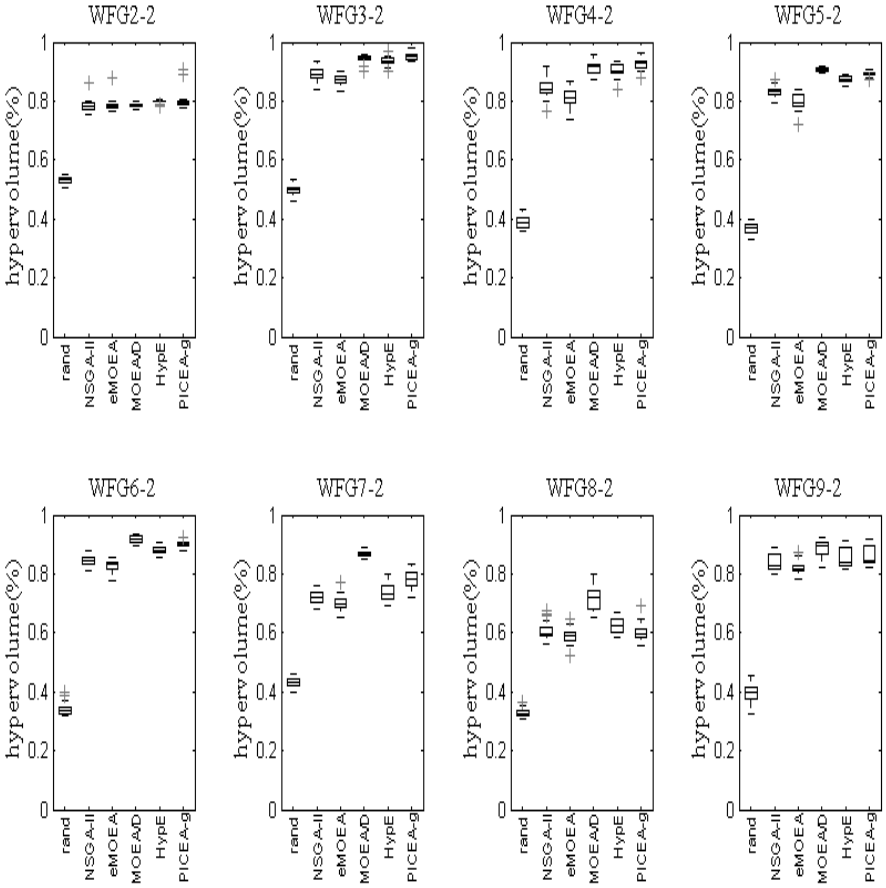


FIGURE 3.12: Box plots of hypervolume results for 2-objective instances.

From the statistical comparison results in Table 3.3 and box plots in Figure 3.12, the following key points can be observed for 2-objective WFG problems:

(i) MOEA/D is always amongst the top performing algorithms except for WFG2, where HypE and PICEA-g are the best performers.

(ii) PICEA-g outperforms ε-MOEA and NSGA-II on all the problems except for WFG8 and WFG9, where the three algorithms perform comparably.

(iii) HypE is inferior to PICEA-g on WFG3, 5 , 6 and 7, while it performs better on WFG8 and comparably on WFG9.

(iv) The performance of ε-MOEA is worse than (problems WFG3 to WFG7) or at best equivalent to (WFG2, 8 and 9) other MOEAs.

(v) NSGA-II exhibits mixed performance. It is ranked in the second class on WFG2 and WFG9, but it is worse than PICEA-g, HypE and MOEA/D on problems WFG3 to WFG6. Its performance is equivalent to PICEA-g, ε-MOEA on WFG8 and WFG9. Also, it performs worse than (WFG8) or comparably to (WFG7 and WFG9) HypE.

(vi) All five MOEAs outperform random search.

HV results for 4-objective WFG problems

Table 3.4: HV results for 4-objective instances.

WFG		Ranking by HV	WFG		Ranking by HV
2	1	HypE PICEA-g	3	1	PICEA-g
	2	MOEA/D NSGA-II		2	HypE
	3	ε-MOEA		3	MOEA/D
	4	rand		4	NSGA-II
				5	ε-MOEA
				6	rand
4	1	PICEA-g	5	1	PICEA-g
	2	HypE		2	HypE
	3	MOEA/D		3	MOEA/D
	4	ε-MOEA		4	ε-MOEA
	5	NSGA-II		5	NSGA-II
	6	rand		6	rand
6	1	HypE PICEA-g	7	1	HypE PICEA-g
	2	MOEA/D		2	MOEA/D
	3	ε-MOEA		3	ε-MOEA
	4	NSGA-II		4	NSGA-II
	5	rand		5	rand
8	1	MOEA/D	9	1	HypE PICEA-g
	2	PICEA-g		2	MOEA/D
	3	HypE		3	ε-MOEA
	4	ε-MOEA		4	NSGA-II
	5	NSGA-II		5	rand
	6	rand			

It is clear from Table 3.4 and Figure 3.13 that, for all 4-objective WFG problems, all MOEAs outperform random search. The performance of PICEA-g remains promising. In detail, except for WFG8 (MOEA/D is the best), PICEA-g is ranked first on all benchmark functions, exclusively for WFG3, 4 and 5, and jointly with HypE on WFG2, 6, 7 and 9. Reinforcing conclusions from previous studies, the Pareto-dominance based NSGA-II begins to struggle on 4-objective problems: it performs equivalently to MOEA/D on WFG2 and better than ε–MOEA on WFG3, but exhibits worse performance than both MOEA/D and ε–MOEA on the remaining benchmark functions.

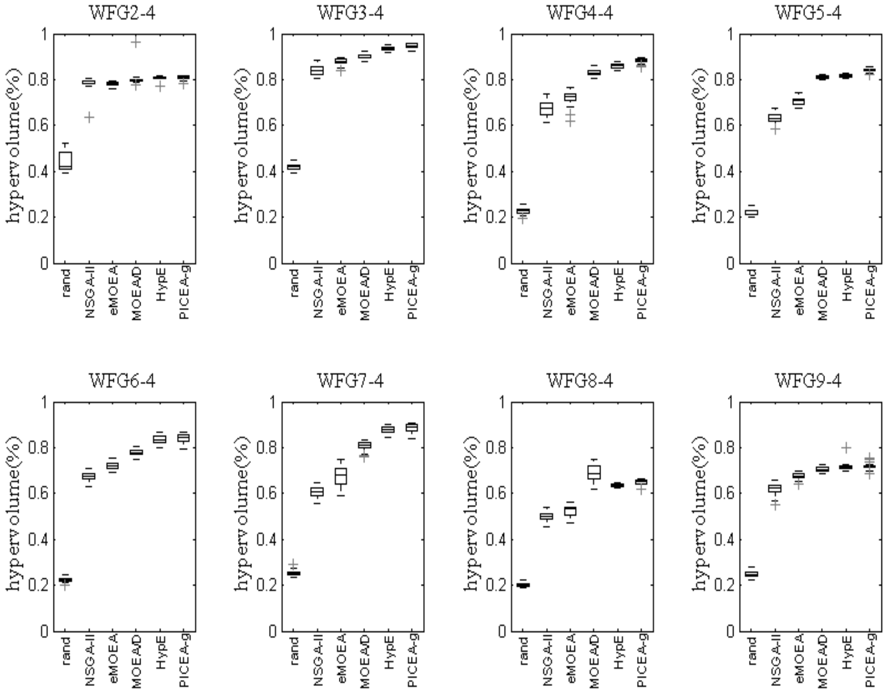


FIGURE 3.13: Box plots of hypervolume results for 4-objective instances.

TABLE 3.5: *HV* results for 7-objective instances.

WFG		Ranking by *HV*	WFG		Ranking by *HV*
2	1	ε-MOEA HypE MOEA/D NSGA-II PICEA-g	3	1	HypE PICEA-g
	2	rand		2	MOEA/D
				3	ε-MOEA
				4	NSGA-II
				5	rand
4	1	PICEA-g	5	1	PICEA-g
	2	HypE		2	HypE
	3	ε-MOEA MOEA/D		3	MOEA/D
	4	NSGA-II		4	ε-MOEA
	5	rand		5	NSGA-II
				6	rand
6	1	PICEA-g	7	1	PICEA-g
	2	HypE		2	HypE
	3	ε-MOEA MOEA/D		3	MOEA/D
	4	NSGA-II		4	ε-MOEA
	5	rand		5	NSGA-II
				6	rand
8	1	PICEA-g	9	1	HypE PICEA-g
	2	HypE		2	ε-MOEA MOEA/D
	3	ε-MOEA MOEA/D NSGA-II		3	NSGA-II
	4	rand		4	rand

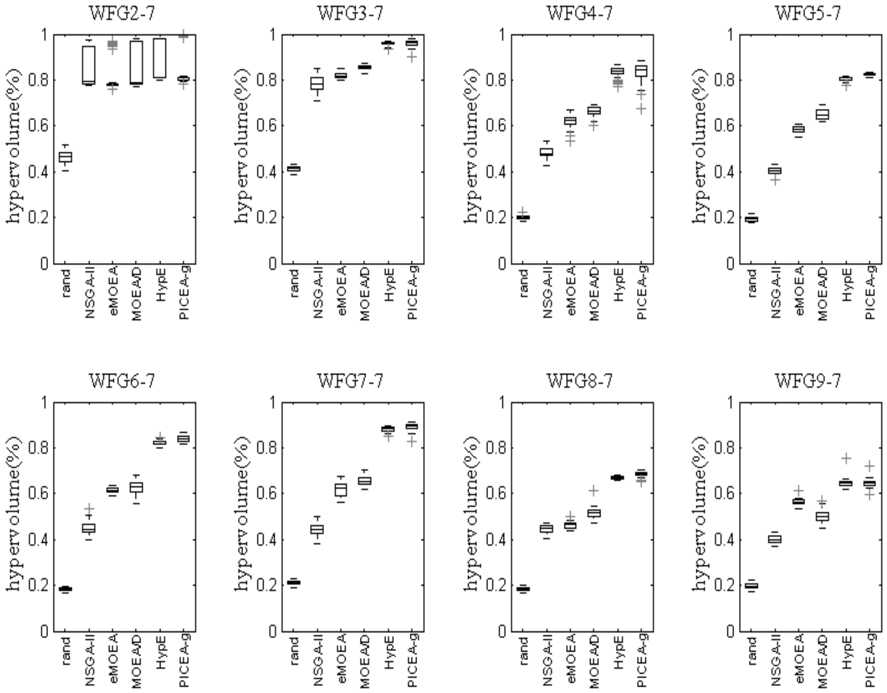


Figure 3.14: Box plots of hypervolume results for 7-objective instances.

HV results for 7-objective WFG problems

As the number of objectives increases to 7, we can observe (see Table 3.5 and Figure 3.14) that PICEA-g and HypE outperform the other three MOEAs. Interestingly, random search still performs the least well in all cases. Upon closer examination, the findings are as follows.

(i) PICEA-g and HypE are ranked first and second for WFG4, 5, 6, 7 and 8, respectively. The two algorithms are jointly ranked in the first class on WFG2, 3 and 9.

(ii) MOEA/D performs better than (on problems WFG3, 5 and 7) or at least comparably to (WFG2, 4, 6 and 9) ε–MOEA on all the benchmark functions. NSGA-II exhibits an inferior performance to ε–MOEA on all tests, except WFG2 and WFG8, where it gives comparable performance with MOEA/D and ε–MOEA.

(iii) All of the MOEAs have a comparable performance for WFG2, where absolute coverage of the globally optimal hypervolume remains at over 80%.

Table 3.6: HV results for 10-objective instances

WFG		Ranking by HV	WFG		Ranking by HV
2	1	ε-MOEA HypE MOEA/D NSGA-II PICEA-g	3	1	HypE PICEA-g
	2	rand		2	ε-MOEA MOEA/D
				3	NSGA-II
				4	rand
4	1	PICEA-g HypE	5	1	PICEA-g
	2	MOEA/D		2	HypE
	3	ε-MOEA NSGA-II		3	MOEA/D
	4	rand		4	ε-MOEA
				5	NSGA-II
				6	rand
6	1	PICEA-g	7	1	HypE PICEA-g
	2	HypE		2	MOEA/D
	3	ε-MOEA		3	ε-MOEA
	4	MOEA/D		4	NSGA-II
	5	NSGA-II		5	rand
	6	rand			
8	1	HypE PICEA-g	9	1	HypE
	2	MOEA/D		2	PICEA-g
	3	NSGA-II		3	ε-MOEA
	4	ε-MOEA		4	MOEA/D
	5	rand		5	NSGA-II
				6	rand

HV results for 10-objective WFG problems

Results for 10-objective WFG problems are shown in Table 3.6 and Figure 3.15. Considering relative comparisons, random search continues to perform badly and the superiority of PICEA-g and HypE is more established. The performances of these two algorithms are statistically better than all of the other algorithms on all benchmark functions except for WFG2, on which the five MOEAs are all in the first class (as for WFG2 with 7 objectives) and are better than random search. Specifically, PICEA-g and HypE jointly rank in the first class for WFG3, 4, 7 and 8. PICEA-g is the exclusive best for WFG5 and WFG6, and, likewise, HypE the exclusive best for WFG9. Among MOEA/D, ε–MOEA and NSGA-II, MOEA/D exclusively gives the best performance on WFG4, 5, 7 and 8. ε–MOEA is somewhat inferior to MOEA/D on most of the benchmark functions, however, for WFG6 and 9, it outperforms MOEA/D. The performance of both algorithms is comparable on WFG3. NSGA-II performs worst on all benchmark functions except for WFG8, where ε–MOEA is the worst. In terms of absolute performance, the box plots show that on all problems, except for WFG8 and 9, HypE and PICEA-g are still able to achieve over 80% of the global hypervolume value. A crude random search can achieve

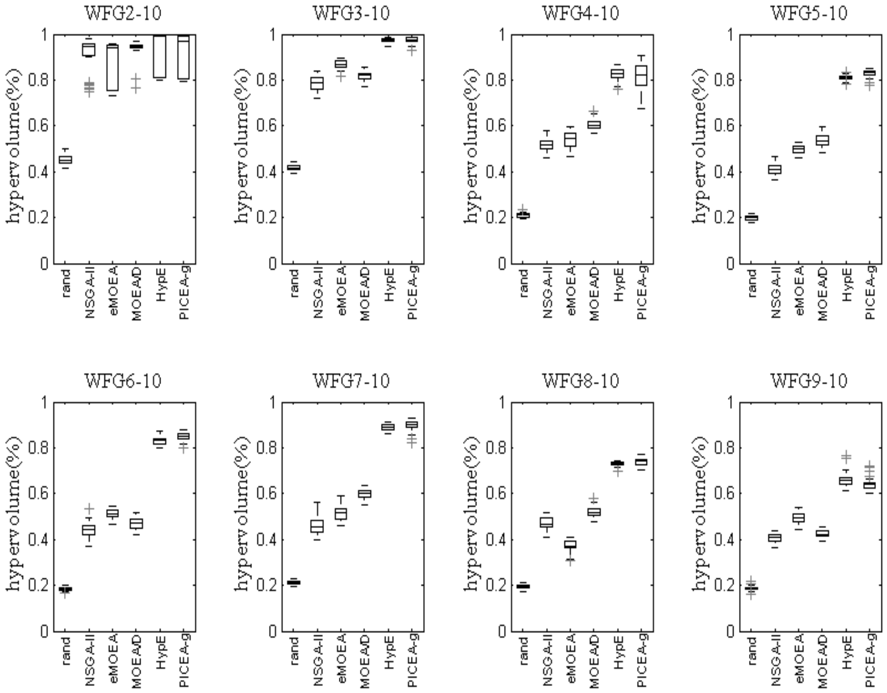


FIGURE 3.15: Box plots of hypervolume results for 10-objective instances.

approximately 20% coverage of the optimal hypervolume – with all MOEAs performing better than this for the equivalent number of candidate solution evaluations.

Supplementary results: *GD* and spread metric Δ

To further understand the performance of the algorithms, we have also separately calculated proximity (as measured by generational distance – *GD*) and diversity (as measured by the spread metric – Δ) measures for the 7-objective WFG4 to WFG9 benchmark functions.

The Pareto optimal front for these problems is a regular geometric shape (the surface of a hyper-ellipsoid) which is amenable to uniform sampling. We sample 20,000 points as the reference set for calculating the metrics. The statistical tests are based on the mean values of the performance indicators, and the same non-parametric procedures are adopted as earlier. The *GD* and Δ results are shown in Table 3.7 and Table 3.8 respectively. Similar to the *HV* calculation, approximation sets are normalised by the *nadir* point prior to the calculation.

MOEA/D is found to achieve the best proximity on four of the six problems considered; however it tends to rank quite poorly in terms of diversity. NSGA-II is found to provide inferior proximity to random search on all six benchmark problems, but with a diversity metric that is superior to random. PICEA-g consistently ranks amongst the top two

TABLE 3.7: *GD* results for selected 7-objective instances.

WFG		Ranking by *GD*	WFG		Ranking by *GD*
4	1	MOEA/D	5	1	MOEA/D
	2	PICEA-g		2	PICEA-g
	3	HypE		3	HypeE
	4	ε-MOEA		4	ε-MOEA
	5	rand		5	rand
	6	NSGA-II		6	NSGA-II
6	1	MOEA/D	7	1	PICEA-g
	2	PICEA-g		2	HypE MOEA/D
	3	HypE		3	ε-MOEA
	4	ε-MOEA		4	rand
	5	rand		5	NSGA-II
	6	NSGA-II			
8	1	PICEA-g	9	1	MOEA/D
	2	HypE		2	HypE PICEA-g
	3	MOEA/D rand		3	ε-MOEA
	4	ε-MOEA		4	rand
	5	NSGA-II		5	NSGA-II

TABLE 3.8: Δ results for selected 7-objective instances.

WFG		Ranking by Δ	WFG		Ranking by Δ
4	1	HypE	5	1	HypE
	2	ε-MOEA PICEA-g		2	ε-MOEA PICEA-g
	3	NSGA-II		3	NSGA-II
	4	MOEA/D		4	MOEA/D
	5	rand		5	rand
6	1	ε-MOEA HypE PICEA-g	7	1	HypE
	2	NSGA-II		2	PICEA-g
	3	MOEA/D		3	ε-MOEA
	4	rand		4	NSGA-II
				5	MOEA/D
				6	rand
8	1	ε-MOEA MOEA/D NSGA-II	9	1	ε-MOEA HypE
	2	HypE PICEA-g		2	PICEA-g
	3	rand		3	NSGA-II
				4	MOEA/D
				5	rand

for both proximity and spread, and is the only algorithm under test to achieve such a performance.

Additionally, as the true Pareto front of these problems is the surface of a hyper-ellipsoid, the GD metric can also be calculated by:

$$GD = \frac{\sum_{i=1}^{i=num}(\sqrt{\sum_{j=1}^{j=M}(normF_{ij}^2)} - 1)}{num} \tag{3.9}$$

where num is the number of solutions in the offline archive, $normF$ are the offline archive solutions normalised by the *nadir* point.

We have also examined the ranking results using GD values calculated by Equation 3.9. The same results as shown in Table 3.7 are found. This further confirms the convergence ranking results of algorithms. Also it suggests that the sampled solutions are sufficient to represent the entire Pareto optimal front and therefore the spread ranking results are reliable.

Supplementary results: $\mathcal{C}(A, B)$ for WFG2

According to the HV comparison results, NSGA-II is also found to perform well in absolute terms on the many-objective WFG2 problems (i.e., WFG2-7 and WFG2-10). It is known that standard Pareto-dominance based approaches can perform well when the dimensionality of the Pareto front is not many-objective (Schütze et al., 2011) or if the objectives are highly correlated (Ishibuchi et al., 2011), but these are not characteristics of the WFG2 problem. To understand this issue, we further compare NSGA-II to the other five algorithms in terms of the $\mathcal{C}$ metric. This aims to investigate if NSGA-II has achieved a comparable convergence performance as other algorithms. The comparison results are summarised in Table 3.9. The non-parametric Wilcoxon-ranksum two-sided comparison procedure at the 95% confidence level is employed to test if the two approaches performs comparably in terms of the $\mathcal{C}$ metric. The symbol '<', '=' or '>' means NSGA-II perform statistically worse, equal or better than a considered algorithm at 95% confidence level.

From Table 3.9 we observe that for each pair of comparisons, all the considered MOEAs outperform NSGA-II in terms of the C metric. Specifically, none of solutions achieved by PICEA-g, HypE or MOEA/D is dominated by NSGA-II for both WFG2-7 and WFG2-10. Regarding ε-MOEA, some of the obtained solutions are dominated by NSGA-II ($\mathcal{C}$(NSGA-II, ε-MOEA): 0.0006) on WFG2-7. However, for WFG2-10 again, none of the solutions from ε-MOEA is dominated by NSGA-II.

Based on these results, we conclude that NSGA-II does not have comparable convergence performance to the other MOEAs on WFG2 problem. The reason that NSGA-II

TABLE 3.9: C metric comparison results (mean±std) of NSGA-II and other MOEAs.

	$\mathcal{C}$(NSGA-II, PICEA-g)		$\mathcal{C}$(PICEA-g, NSGA-II)
WFG2-7	0±0	<	0.1819±0.1157
WFG2-10	0±0	<	0.1397±0.1013
	$\mathcal{C}$(NSGA-II, HypE)		$\mathcal{C}$(HypE, NSGA-II)
WFG2-7	0±0	<	0.2035±0.1193
WFG2-10	0±0	<	0.1177±0.0967
	$\mathcal{C}$(NSGA-II, ε-MOEA)		$\mathcal{C}$(ε-MOEA, NSGA-II)
WFG2-7	0.0006±0.0036	<	0.0348±0.0637
WFG2-10	0±0	<	0.0635±0.0865
	$\mathcal{C}$(NSGA-II, MOEA/D)		$\mathcal{C}$(MOEA/D, NSGA-II)
WFG2-7	0±0	<	0.1784±0.0914
WFG2-10	0±0	<	0.1858±0.1197

performs comparably with other algorithms on the 7- and 10-objective WFG2 problems might be that the selected reference point is not suitable for a disconnected Pareto optimal front which the WFG2 has.

3.5 Discussion

3.5.1 The effect of the scaling parameter α

On the estimation of goal vector upper bound, see Equation 3.4, a parameter α is used. In the comparison study, α is set to 1.2. This section experimentally examines the effect of this parameter. Five different settings are investigated: $\alpha = \{1, 1.2, 1.5, 2, 4\}$. The 2-, 4-, 7- and 10-objective WFG4 problems are selected as test problems. 31 runs are executed for each problem. Other parameters are set the same as adopted in Table 3.2. HV results of PICEA-g using different α are box plotted in Figure 3.16 for different problems.

From Figure 3.16 we can observe that the performance of PICEA-g is affected by the goal vector bound, itself arising from the setting of α. First, the performance of PICEA-g is not improved as the value of α increases. Next, $\alpha = 1$ produces the worst performance for WFG4-2. $\alpha = 4$ produces the worst performance on the other three many-objective problems. Relatively, $\alpha = 1.2$, $\alpha = 1.5$ or $\alpha = 2$ produces better performance.

The reason for the poor performance of $\alpha = 1$ is that goal vectors are generated exclusively in G_1 as shown in Figure 3.17. These goal vectors are not effective in exploring boundary areas and therefore lead to a poor diversity performance, in particular, a poor coverage of the Pareto optimal front. The reason for the poor performance of $\alpha = 4$ is that a large portion of goal vectors are generated in G_4, see Figure 3.17. However, goal vectors generated in this region do not offer comparability between solutions (as they are dominated by all the solutions). It is also worth mentioning that when $\alpha = 4$

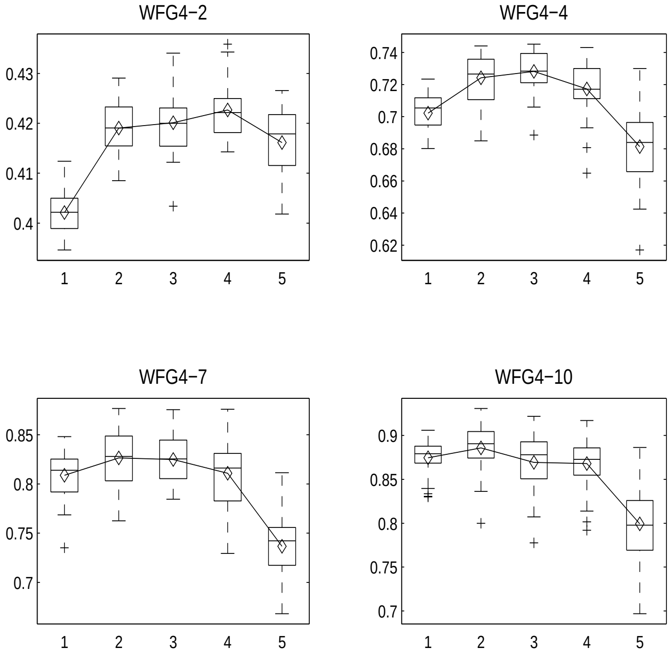


FIGURE 3.16: The influence of parameter α on PICEA-g performance in terms of HV metric (1: $\alpha = 1$, 2: $\alpha = 1.2$, 3: $\alpha = 1.5$, 4: $\alpha = 2$ and 5: $\alpha = 4$).

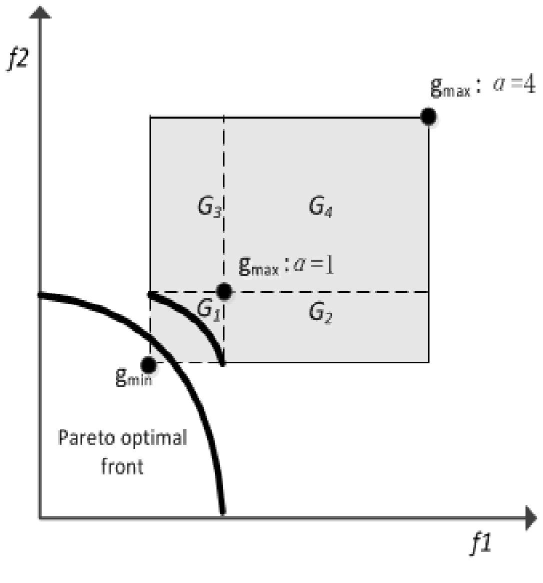


FIGURE 3.17: Illustration of the effect of α.

many goal vectors are generated in G_2 and G_3. These goal vectors are helpful to explore boundary areas since these goal vectors result in higher fitness for boundary solutions. Therefore, $\alpha = 4$ might enable the algorithm to have a better diversity performance, in particular, the coverage of the Pareto optimal front. The effect of goal vector bound is also discussed later in Section 3.7.1.

These observations can also be confirmed by the statistical comparison results made with the GD and Δ metric. Table 3.10 and Table 3.11 show the mean and standard deviation values of the two metrics for PICEA-g using different settings of α. The best mean value is marked in **boldface**.

TABLE 3.10: GD comparison results (mean±std) of PICEA-g with different α value.

α	WFG4-2	WFG4-4	WFG4-7	WFG4-10
1	**0.0008±0.0001**	**0.0021±0.0005**	**0.0151±0.0006**	**0.0194±0.0008**
1.2	0.0010±0.0002	0.0024±0.0005	0.0168±0.0008	0.0227±0.0008
1.5	0.0011±0.0002	0.0028±0.0006	0.0178±0.0008	0.0269±0.0010
2	0.0011±0.0002	0.0035±0.0004	0.0191±0.0008	0.0292±0.0010
4	0.0012±0.0002	0.0041±0.0005	0.0236±0.0011	0.0331±0.0018

TABLE 3.11: Δ comparison results (mean±std) of PICEA-g with different α value.

α	WFG4-2	WFG4-4	WFG4-7
1	0.1952±0.0051	2.5098±0.0849	6.0188±0.3941
1.2	0.1614±0.0049	**2.1436±0.0460**	**4.0147±0.2214**
1.5	0.1610±0.0052	2.4141±0.0490	5.3106±0.1887
2	**0.1601±0.0053**	2.2783±0.0480	5.6292±0.2117
4	0.1734±0.0053	2.4067±0.1214	4.6561±0.1960

Combining Table 3.10 and Table 3.11 we can observe that

- $\alpha = 1$ produces the best convergence, but the worst diversity performance;
- $\alpha = 4$ produces good diversity performance while its convergence performance is the worst.
- $\alpha = 1.2, 1.5$ or 2 produces neutral performance on both convergence and diversity.

According to the above analysis we report that α affects the performance of PICEA-g. More rigorously, the performance of PICEA-g is affected by the goal vector bounds. It is suggested, from the experimental results, that $\alpha \in (1, 2)$ is a good choice.

3.5.2 The effect of co-evolution

To clearly demonstrate the effect of co-evolution, PICEA-g is further compared with its variant: random-g. In random-g, goal vectors are not co-evolved but are randomly generated within goal vector bounds in each generation. Specifically, in random-g goal vectors in G are randomly generated the same as Gc. The experimental setup is the same as that adopted in Section 3.3.

The HV comparison results between random-g and PICEA-g are shown in Tables 3.12, 3.13 and 3.14 for the 2-, 4- and 7-objective WFG problems, respectively. In each table, the mean and the standard deviation HV values across the 31 independent runs are presented. The symbol '<', '=' or '>' shows PICEA-g is statistically worse, equal or better than random-g at 95% confidence level. Superior results are marked in **boldface**.

TABLE 3.12: HV comparison results (mean±std) of PICEA-g and random-g for 2-objective WFG instances.

	PICEA-g		random-g
WFG2-2	0.5712±0.0143	=	0.5635±0.0190
WFG3-2	**0.6132±0.0085**	>	0.5969±0.0079
WFG4-2	0.4194±0.0039	=	0.4210±0.0064
WFG5-2	0.3990±0.0052	=	0.3949±0.0054
WFG6-2	0.3921±0.0074	=	0.3923±0.0046
WFG7-2	0.3631±0.0076	=	0.3650±0.0101
WFG8-2	**0.2394±0.0066**	>	0.2163±0.0028
WFG9-2	0.3893±0.0191	=	0.3792±0.0219

TABLE 3.13: HV comparison results (mean±std) of PICEA-g and random-g for 4-objective WFG instances.

	PICEA-g		random-g
WFG2-4	0.8160±0.0088	=	0.8157±0.0246
WFG3-4	**0.8528±0.0099**	>	0.8269±0.0115
WFG4-4	**0.7516±0.0058**	>	0.7263±0.0143
WFG5-4	**0.7146±0.0071**	>	0.6842±0.0113
WFG6-4	0.7137±0.0170	=	0.7002±0.0180
WFG7-4	0.7603±0.0128	=	0.7594±0.0121
WFG8-4	**0.6126±0.0133**	>	0.5732±0.0124
WFG9-4	0.6234±0.0098	=	0.6095±0.0098

TABLE 3.14: HV comparison results (mean±std) of PICEA-g and random-g for 7-objective WFG instances.

	PICEA-g		random-g
WFG2-7	0.8764±0.0601	=	0.8785±0.0703
WFG3-7	**0.9321±0.0131**	>	0.9050±0.0145
WFG4-7	**0.8410±0.0361**	>	0.7716±0.0461
WFG5-7	**0.8314±0.0072**	>	0.7621±0.0097
WFG6-7	**0.8257±0.0137**	>	0.7885±0.0164
WFG7-7	**0.8487±0.0184**	>	0.8325±0.0221
WFG8-7	**0.7481±0.0120**	>	0.7029±0.0098
WFG9-7	**0.6619±0.0152**	>	0.6321±0.0203

According to the results presented in Tables 3.12, 3.13 and 3.14 we can observe that PICEA-g performs better than, or comparably, with random-g on all problems. Specifically, PICEA-g performs better than random-g for two bi-objective problems, four 4-objective problems and seven 7-objective problems. These results clearly demonstrate that co-evolution plays an important role in PICEA-g, and its effect becomes more significant as the number of objective increases.

3.5.3 The performance of goal vectors

A particular concern with optimisation-focused implementations of co-evolution is the potential for pathologies such as: the Red Queen Effect (subjective fitness improves without any corresponding improvement in objective fitness, or vice versa); cycling (subjective fitness exhibits limit cycle dynamics, without incremental improvement); and disengagement (loss of fitness discrimination due to the total superiority of one population) (De Jong and Pollack, 2004; Bongard and Lipson, 2005). Fortunately, the fitness assignment scheme in PICEA-g, based on the approach of Lohn et al. (2002), appears resistant to these issues. As anticipated, both the candidate solutions and the preferences can converge towards the Pareto optimal front. Figure 3.18 shows how hypervolume (normalised by the true hypervolume value) changes over the course of the evolution for candidate solutions and preferences. Note that the preference dynamics tend to slightly lag the solutions. The reason for the lag is that goal vectors that cannot be dominated by any solutions are assigned the worst possible fitness score in PICEA-g.

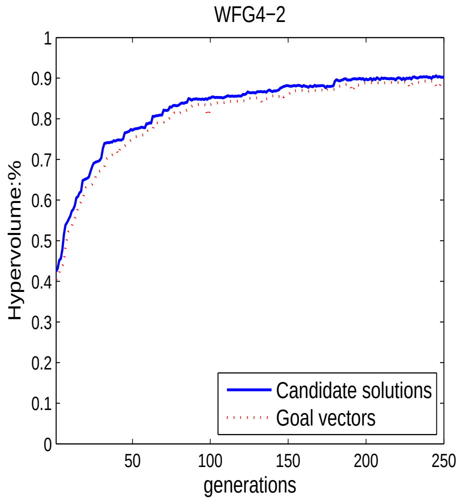


Figure 3.18: Illustration of the performance of candidate solutions and goal vectors on WFG4-2.

3.6 Summary of experiment findings

We have carried out a systematic comparison of six algorithms which included five different classes of multi-objective evolutionary algorithms: a preference-influenced co-evolutionary algorithm (PICEA-g), a Pareto-dominance based algorithm (NSGA-II), an ε-dominance based algorithm (ε-MOEA), a scalarising function based algorithm (MOEA/D) and an indicator based algorithm (HypE). Overall the empirical comparison has identified that, from the representative algorithms considered, PICEA-g and HypE are presently good options for many-objective optimisation. Meanwhile, MOEA/D exhibits outstanding performance on bi-objective problems, but has not performed so well in a many-objective context. Specifically, the findings are summarised as follows:

(i) The preference-inspired co-evolutionary algorithm (PICEA-g) exhibits promising performance for many-objective problems. It is found to be consistently among the top performing algorithms across the test problems considered. In addition to superior performance, as measured by the hypervolume indicator, on many-objective problems it also offers competitive performance with the popular NSGA-II in bi-objective environments.

(ii) The estimated hypervolume indicator based evolutionary algorithm (HypE) gives very competitive performance on both bi-objective and many-objective problems. Its performance is close to PICEA-g and better than other MOEAs on most of the selected problems.

(iii) The concept of ε-dominance is much more effective than pure Pareto-dominance in solving many-objective problems. Although ε-MOEA does not exhibit the best performance, it outperforms NSGA-II on most of the many-objective problems studied. However, the hypervolume measure of ε-MOEA is not found to be better than NSGA-II on bi-objective WFG tests. This may be due to the fact that ε-MOEA is not effective in obtaining extreme solutions on the Pareto front (Deb et al., 2003, 2005), or, perhaps, the ε value is set inappropriately.

(iv) MOEA/D performs well on 2-objective problems. However its performance is not particularly notable on the many-objective problems studied. Despite the unremarkable performance of MOEA/D on many-objective problems, according to hypervolume, the 7-objective results focusing separately on proximity and spread show that this algorithm is still very capable at finding solutions that are close to the global Pareto front. The issue is a loss of diversity, which is likely to be due to inappropriate specifications of *a priori* weight vectors, itself arising from a general lack of knowledge in the literature about how to configure the algorithm in many-objective spaces. (This issue will be further studied in Chapter 4) Moreover, this might be explained, partly, by the parameter settings used. Studies have demonstrated that MOEA/D is sensitive to parameters such as the selection and replacement neighbourhood size (Ishibuchi et al., 2009b).

(v) The classical Pareto-dominance and density based algorithm (NSGA-II) can perform well on bi-objective problems. However its performance is significantly degraded, in both relative and absolute terms, when dealing with many-objective problems. This result was first identified in Purshouse and Fleming (2003c) and confirmed in Hughes (2005) and is believed to be due to dominance resistance (in this case due to many-objectives) coupled with an active diversity promotion mechanism that favours remote solutions far away from the global Pareto front.

(vi) In terms of hypervolume, all the MOEAs considered offer better performance than a crude random strategy. Note that other studies have found that MOEAs can degenerate to random search (or possibly worse) on many-objective problems (Purshouse and Fleming, 2003c; Knowles and Corne, 2007). In this comparison study NSGA-II is seen to outperform random search in many-objective spaces according to hypervolume, further interrogation of the 7-objective results has confirmed that this tends to be based on approximation sets with equivalent or worse proximity to random search, yet retaining good diversity.

Overall the concept of co-evolving preferences with candidate solutions during the optimisation process is promising for solving many-objective optimisation problems. PICEA-g, as one specific algorithm of this concept, is demonstrated to show a highly competitive performance (compared to four best-in-class MOEAs) on WFG test problems with 2, 4, 7 and 10 objectives.

3.7 An enhanced PICEA-g using adaptively generated goal vectors

We have demonstrated the superiority of PICEA-g in approximating the entire Pareto optimal front in comparison with other MOEAs. However, all MOEAs have shortcomings. This section identifies one shortcoming of PICEA-g, that is, its performance is affected by the determined goal vector bounds. A strategy named the *cutting plane* is proposed to address this issue, and to improve the performance of PICEA-g even further.

The *cutting plane* is based on the assumption that the *ideal* point is known *a priori*. Unlike the *nadir* point (which is difficult to obtain (Deb et al., 2010)), the *ideal* point $\mathbf{z^{ide}} = (z_1^{ide}, z_2^{ide}, \cdots, z_M^{ide})$, being composed of all the best criteria available (minimum objective values for minimisation problems), can be easily obtained either via expert domain-specific knowledge or via preliminary single objective optimisation.

If the *ideal* point is known, the degree of exploration of an objective can be known by simply comparing objective value f_i with its corresponding *ideal* value z_i^{ide}. Such information might serve as a guide to appropriately distribute search effort towards different objectives during the search process. The earlier discussion in Section 3.5 p.81 shows that PICEA-g is affected by goal vector bounds. By co-evolving goal vectors generated in suitable goal vector bounds, PICEA-g can then distribute the search effort appropriately towards different objectives (this will be discussed in Section 3.7.1), and therefore lead to a better performance.

The rest of this research is structured as follows. First, we analyse the influence of goal vector bounds on the performance of PICEA-g. Following this, we describe our proposed strategy – the *cutting plane* that can adaptively configure the goal vector bounds. Finally, we experimentally examine the effectiveness of the proposed strategy.

3.7.1 An analysis of the effect of goal vector bounds

This section analyses the effect of goal vector bounds in PICEA-g. This analysis is conducted by considering six different sets of goal vectors generated using six different $\mathbf{g}_{max}$, see the left column of Figure 3.19 and Figure 3.20. In each case, PICEA-g is executed independently for 31 runs (each run for 25 000 function evaluations) on the 2-objective WFG4 problem. Plots of Pareto front which has median hypervolume value are shown in the right column of Figure 3.19 and Figure 3.20. In each figure, the shaded area represents the region dominated by a candidate solution. A and C represent the lower, $\mathbf{g}_{min}$ and upper, $\mathbf{g}_{max}$ bound, respectively. $\mathbf{g}_{min}$ is set to the coordinate origin as the *ideal* point for WFG4-2 is $(0, 0)$. In respect to $\mathbf{g}_{max}$, the six cases, respectively, represent that

(i) the generated goal vectors only cover the knee (EF) of the Pareto optimal front: Figure 3.19(a).

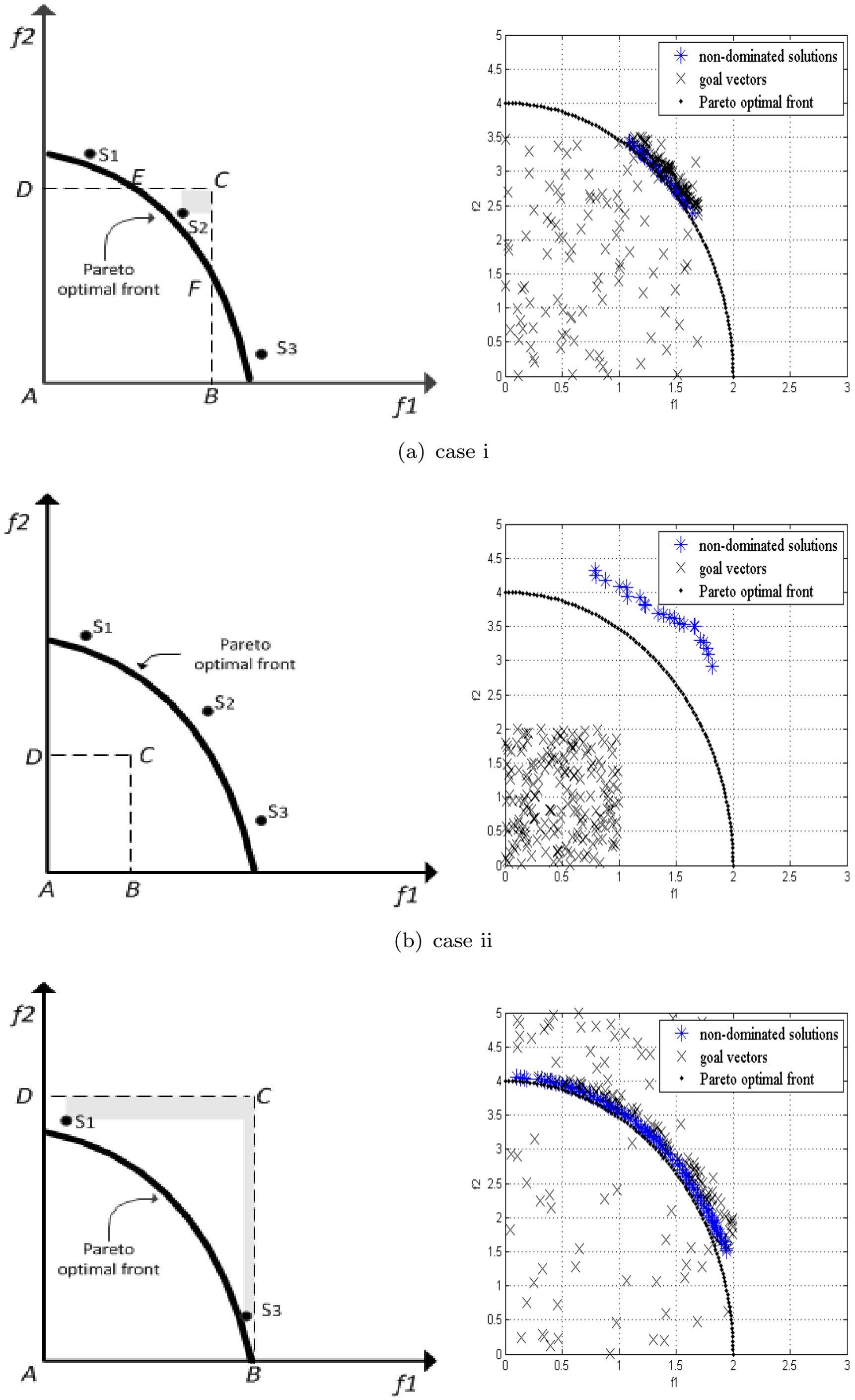
(ii) the generated goal vectors are all infeasible: Figure 3.19(b).

(iii) more goal vectors are generated in the top-left part of the Pareto optimal front: Figure 3.19(c).

(iv) more goal vectors are generated in the bottom-right part of the Pareto optimal front: Figure 3.20(a).

(v) goal vectors are now generated within the bounds of *ideal* and *nadir* vector: Figure 3.20(b).

(vi) goal vectors are generated within the bounds of *ideal* and a relaxed form of the *nadir* vector, i.e. $3 \times$ *nadir*: Figure 3.20(c).

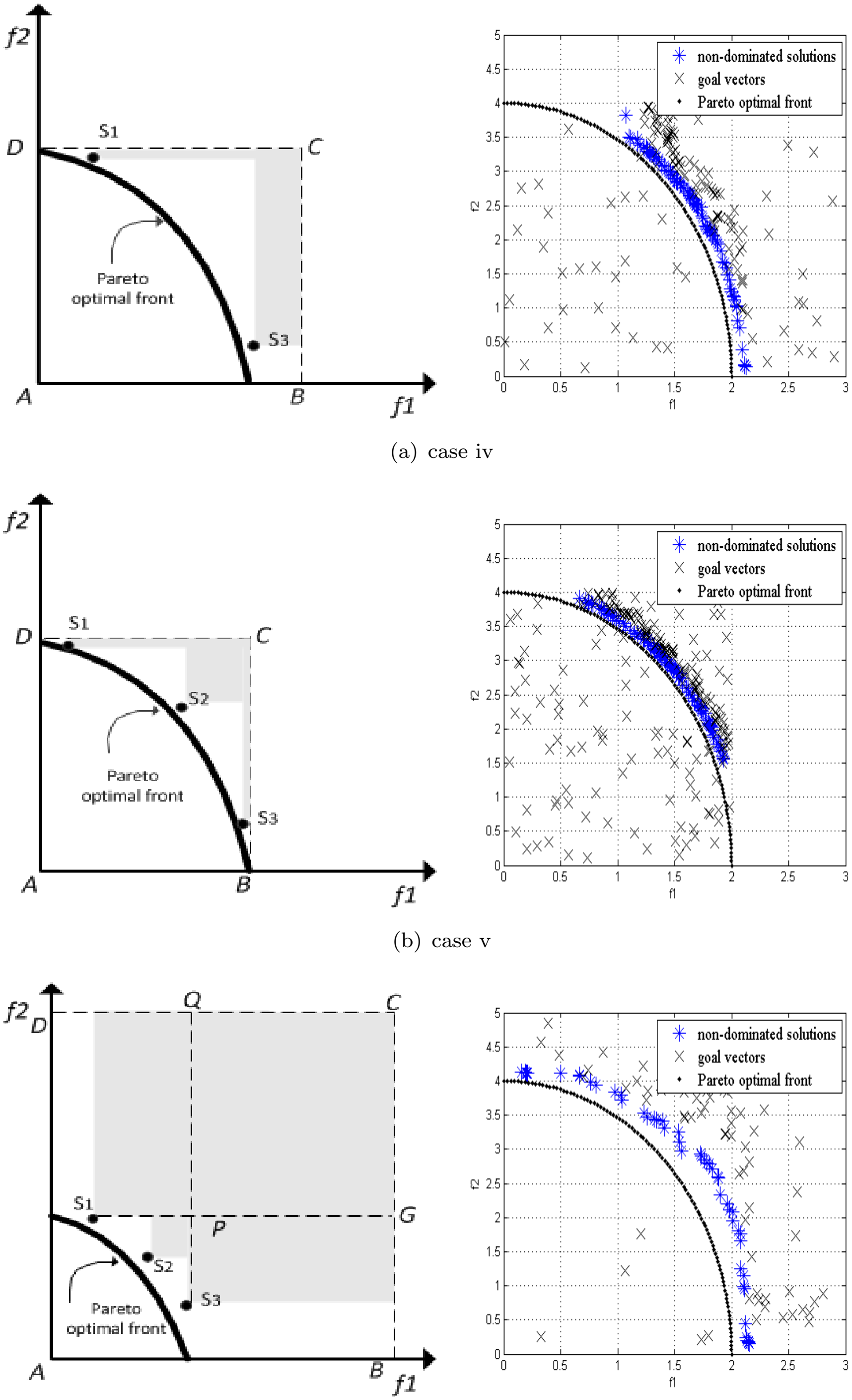


(a) case i

(b) case ii

(c) case iii

FIGURE 3.19: Illustration of different goal vectors (Left) and the corresponding Pareto front (Right): Part I



(a) case iv

(b) case v

(c) case vi

FIGURE 3.20: Illustration of different goal vectors (Left) and the corresponding Pareto front (Right): Part II

The obtained non-dominated solutions for case (i) are shown in Figure 3.19(a). It is observed from the results that the obtained solutions are all constrained within the region of CEF. The reason is that candidate solutions (e.g. $\mathbf{s}_2$) inside the region of $ABCD$ are more likely to dominate more goal vectors and therefore gaining higher fitness than candidate solutions placed outside this region (e.g. $\mathbf{s}_1$ and $\mathbf{s}_3$). Hence, candidate solutions inside the region of $ABCD$ are more likely to be propagated into the next generation. In addition, it is easy to understand that goal vectors in $ABFED$ are ineffective as these goal vectors cannot be dominated by any candidate solution, i.e., they have no contribution in offering comparability between alternative solutions; rather goal vectors should be generated in a feasible region so that there is a non-zero probability that at least one solution dominates the goal vectors.

The results for case (ii) are shown in Figure 3.19(b). Clearly, candidate solutions have not converged to the Pareto optimal front. The reason is that all goal vectors are infeasible. None of the candidate solutions is able to dominate any goal vector. Hence, candidate solutions are scored randomly and then PICEA-g behaviours in a similar way to random search.

Figure 3.19(c) shows the obtained solutions for case (iii). Solutions are biased towards the top-left of the Pareto optimal front as the generated goal vectors are biased to objective f_1. Specifically, candidate solutions with smaller value of f_1 (e.g., $\mathbf{s}_1$) would gain higher fitness by dominating more goal vectors than those with smaller value of f_2 (e.g., $\mathbf{s}_3$). Hence, solutions in the top-left are more likely to be propagated, leading a richer set of solutions in this region. In other words, more search effort is distributed on f_1. Note that, case (iii) is just an example to illustrate that the search effort would be biased if goal vectors are not appropriately generated. Whether $\mathbf{s}_1$ would dominate more goal vectors or not is also conditioned on the problem geometry.

Case (iv) is similar to case (iii), see Figure 3.20(a). More goal vectors are generated in the bottom-right part. Therefore, solutions in the bottom-right region (e.g. $\mathbf{s}_3$) are more likely to gain higher fitness (compared to $\mathbf{s}_1$) and so are more likely to survive in the evolution.

Results for case (v) are shown in Figure 3.20(b). A and C represent the *ideal* and *nadir* vector, respectively. It is observed that solutions have converged well to the knee part of the Pareto optimal front while the extreme part of the Pareto optimal front is not sufficiently explored. Again, the reason is that compared with solutions in the knee region (e.g. $\mathbf{s}_2$), candidate solutions (e.g. $\mathbf{s}_1$ and $\mathbf{s}_3$) in the extreme parts can only dominate a few goal vectors and so have low fitness. That is to say, solutions in the extreme region are more likely to be disregarded; this degrades the solution diversity.

Results for case (vi) are shown in Figure 3.20(c). This time A is set to the *ideal* point and C is set as $\alpha \times nadir$, $\alpha = 3$ vector. From the results, we observe that the extreme solutions converge slightly better than the knee solutions. Additionally, goal vectors generated in *PGCQ* are also ineffective as these goal vectors are dominated by all candidate solutions, i.e. having no contribution in offering comparability between solutions; therefore we should not generate goal vectors in such region.

Based on the above analysis we conclude that different $\mathbf{g}_{max}$ have different influences on the performance of PICEA-g. This enables us to adjust the search effort towards different objectives by using different $\mathbf{g}_{max}$. In addition, from the above analysis it is suggested that

- goal vectors that can be dominated by all candidate solutions or cannot be dominated by any candidate solution are ineffective, and should not be generated. Overall, the useful goal vectors are only those in the shaded region, see Figure 3.21.

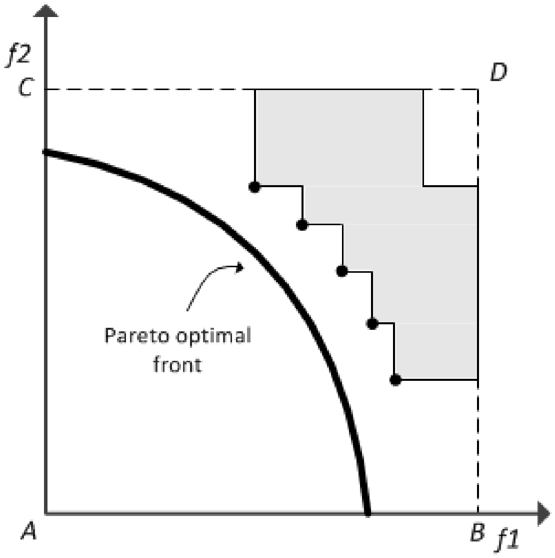


FIGURE 3.21: Illustration of the useful goal vectors.

- applying the *nadir* point as $\mathbf{g}_{max}$ is not a good choice. In any case, the *nadir* point is difficult to estimate. One reason has been mentioned in case (v), that is, the extreme part of the Pareto optimal front cannot be explored well. Another reason is that for some problems e.g., DTLZ1, the initial objective values are all outside the region enclosed by the *nadir* vector, even a relaxed form of *nadir* vector). In this case, PICEA-g would behave similarly to that of a random search, see Figure 3.19(c).

3.7.2 Adaptive goal vector bounds by the *cutting plane*

This section describes a proposed strategy named the *cutting plane* with which a suitable $\mathbf{g}_{max}$ can be determined adaptively. Prior to introducing this strategy, the limitation of the previous strategy used to estimate the $\mathbf{g}_{max}$ (Equation 3.4) is discussed.

Limitation of the previous method

In the comparison study, Equation 3.4 is applied to determine the goal vector bounds. This method is viable when no further information concerning the problem is available. However, if the *ideal* point is known beforehand, more effective methods can be developed to determine suitable goal vector bounds.

As mentioned earlier, by comparing the *ideal* point with the obtained solutions, we can determine which objective is less explored, that is, being far from the ideal value of this objective. According to the analysis in Section 3.7.1, to distribute more search effort towards the unexplored objective, we can set $\mathbf{g}_{max}$ with an offset to this objective so as to generate more goal vectors for the unexplored objective. The method given in Equation 3.4 simply takes the minimum value of $BestF_i$ as g_{min_i}, and $g_{min_i} + \Delta F_i$ as g_{max_i}, where ΔF_i is the value of the difference between the minimum and maximum $BestF_i$ multiplied by a scaling parameter α.

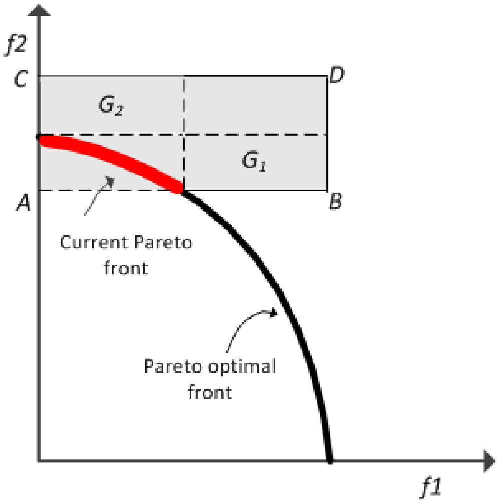


FIGURE 3.22: Limitation of the previous goal vector bounds estimation method.

In this method, knowledge of the *ideal* vector is not utilised. Moreover, this method has a limitation, this is explained with Figure 3.22. Let us assume that the existing solutions have converged to the top-left part of the Pareto optimal front, see Figure 3.22. This indicates that f_1 has been well explored while f_2 is not sufficiently well explored. According to Equation 3.4, the region for the generation of goal vectors towards each objective will have the same size, i.e., the size of G_1 for f_1 is equivalent to the size of G_2 for f_2. Thus, the search effort distributed towards each objective is equivalent. However, obviously, f_2 requires more exploration as its objective value is far from the ideal value.

A new method: the *cutting plane*

Next we introduce our proposed method, namely, the *cutting plane* which can effectively solve this problem and thus distribute the search effort more appropriately along the search process. To describe this strategy, let us consider, as an example, the bi-objective minimisation problem shown in Figure 3.23. The current Pareto front is shown as the dotted arc EF. The goal vector upper bound is calculated by the following two steps.

(i) Compute extreme points (E, F) of the current Pareto front and then obtain the *cutting plane*, i.e. the dashed straight line EF.

(ii) Obtain intercepts of the cutting plane on the coordinate axis. The obtained intercepts are labelled as K, L. The initial goal vector upper bound is determined by the intercepts, shown as point H.

Having obtained the goal vector upper bound, as shown in Figure 3.21, the final region for the generation of goal vectors should be the region closed by E, F, C, B, G, A and D (the shaded region shown in Figure 3.23). In this sense, all generated goal vectors are useful, that is, each of them can be dominated by at least one candidate solution.

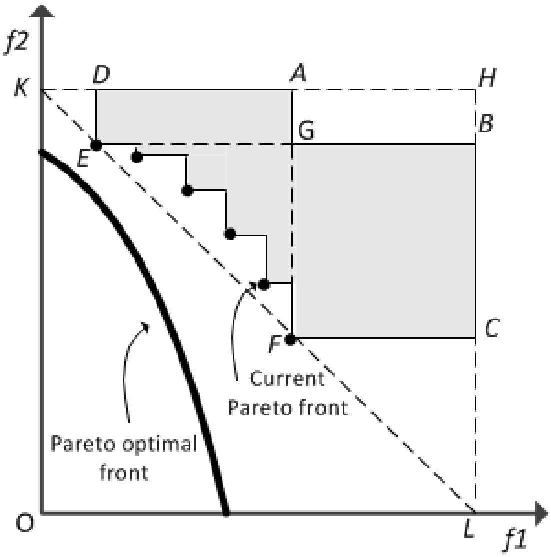


FIGURE 3.23: Illustration of the *cutting plane*.

This strategy is able to adaptively adjust the search effort towards different objectives based on the current Pareto front. Observing the current Pareto front EF shown in Figure 3.23, we find that more search effort should be assigned to objective f_2 as the value of f_2 is far from the ideal value. Here, the *ideal* vector is at the coordinate origin. Using the proposed method, a larger region $GFCB$ (compared with $DEGA$), as expected, is constructed, and more goal vectors are generated in this region. Therefore, according to the working process of PICEA-g, solutions in this region dominate more goal vectors, gaining higher fitness. Thus, these solutions are then more likely to be propagated into the next generation. That is to say, more search effort is distributed towards objective f_2.

Note that if the *ideal* point is not at the coordinate origin, we would need to do a transformation of the coordinate system first, i.e., set the *ideal* point as the new coordinate origin and compute the intercepts (K, L) based on the new coordinate system.

3.7.3 Experiments

This section empirically examines the effectiveness of the *cutting plane* approach. A comparison is made between PICEA-g using goal vector bounds determined by Equation 3.4 (denoted as PICEA-g1) and by the *cutting plane* (denoted as PICEA-g2). Note that for both the algorithms, the lower goal vector bound is set to the *ideal* point.

Experimental setup

Problems 2 to 9 from the WFG test suite are invoked in 2- and 7-objective instances. In each case, the WFG position parameter and distance parameter are set to 18 and 14, providing a constant number of decision variables ($n = 18 + 14 = 32$) for each problem instance. Parameters used for PICEA-g1 and PICEA-g2 are the same as those adopted in the comparison study, see Table 3.2. For each test problem, 31 runs of each algorithm test are performed, each run for 250 generations.

The generational distance metric (GD) and the spread metric (Δ) are used to measure the convergence and diversity performance, respectively. The hypervolume metric (HV) is applied to measure a combination of convergence and diversity performance. For all three metrics, the null hypotheses used are those of equality of mean values. The non-parametric Wilcoxon-ranksum two-sided comparison procedure at the 95% confidence level is employed to test the working hypotheses.

Experimental results

The comparison results for PICEA-g1 and PICEA-g2 in terms of the HV, GD and Δ metrics are shown in Tables 3.15 to 3.20. A favourable (smaller) GD value implies good proximity and also a favourable (smaller) spread metric value implies good diversity. A favourable hypervolume (larger, for a minimization problem) implies good proximity with diversity. In each table, the mean±deviation values of each performance metric across the 31 independent runs are shown. The symbol '<', '=' or '>' shows PICEA-g1 is statistically worse, equal or better than PICEA-g2 at 95% confidence level. Superior results are marked in **boldface**.

The HV comparison results for 2- and 7-objective problems are shown in Table 3.15 and Table 3.16, respectively. Key observations are as follows:

TABLE 3.15: *HV* comparison results (mean±std) of PICEA-g1 and PICEA-g2 for 2-objective WFG instances.

	PICEA-g1		PICEA-g2
WFG2-2	0.5712±0.0143	=	0.5736±0.0097
WFG3-2	0.6132±0.0085	<	**0.6254±0.0080**
WFG4-2	0.4194±0.0039	<	**0.4313±0.0031**
WFG5-2	0.3990±0.0052	<	**0.4123±0.0032**
WFG6-2	0.3921±0.0074	=	0.4017±0.0081
WFG7-2	0.3631±0.0076	<	**0.3710±0.0069**
WFG8-2	0.2394±0.0066	=	0.2402±0.0149
WFG9-2	0.3893±0.0191	=	0.3944±0.0204

TABLE 3.16: *HV* comparison results (mean±std) of PICEA-g1 and PICEA-g2 for 7-objective WFG instances.

	PICEA-g1		PICEA-g2
WFG2-7	0.8764±0.0601	=	0.8851±0.0591
WFG3-7	0.9321±0.0131	<	**0.9416±0.0171**
WFG4-7	0.8410±0.0361	<	**0.8578±0.0355**
WFG5-7	0.8314±0.0072	<	**0.8564±0.0059**
WFG6-7	0.8257±0.0137	<	**0.8471±0.0152**
WFG7-7	0.8487±0.0184	<	**0.8610±0.0161**
WFG8-7	0.7481±0.0120	<	**0.7656±0.0115**
WFG9-7	0.6619±0.0152	<	**0.6796±0.0121**

(i) For 2-objective problems PICEA-g2 performs better than PICEA-g1 on WFG3, 4, 5 and 7; while on the rest of four problems the two algorithms perform comparably.

(ii) For 7-objective problems PICEA-g2 performs better than PICEA-g1 on all the problems, except for WFG2-7 where the two algorithms have comparable performance.

These results clearly demonstrate the effectiveness of the *cutting plane*, especially for many-objective problems.

To better investigate the effect of this strategy, *GD* and Δ metrics are employed for a further comparison. These metrics are calculated for the 2 and 7-objective WFG4 to WFG9 benchmark problems. The same reference set as used in Section 3.4.2 is adopted here. That is, 20,000 uniformly sampled points from the surface of a M-dimension hyper-ellipsoid with radius $r_i = 2i,\ i = 1, 2, \cdots, M$ in the first quadrant.

The *GD* results for 2- and 7-objective problems are shown in Table 3.17 and Table 3.18, respectively. From the results, we observe that for 2-objective problems, PICEA-g2 performs comparably with PICEA-g1 on all the problems except for WFG6-2 where PICEA-g2 is worse. For 7-objective problems, the two algorithms performs comparably on four problems. On WFG8-7 PICEA-g1 is better while on WFG9-7 PICEA-g2 is better.

Table 3.17: *GD* comparison results (mean±std) of PICEA-g1 and PICEA-g2 for 2-objective WFG instances.

Problem	PICEA-g1		PICEA-g2
WFG4-2	0.0010±0.0002	=	0.0011±0.0002
WFG5-2	0.0034±0.0003	=	0.0033±0.0002
WFG6-2	**0.0035±0.0004**	>	0.0056±0.0005
WFG7-2	0.0035±0.0009	=	0.0032±0.0008
WFG8-2	0.0224±0.0013	=	0.0240±0.00219
WFG9-2	0.0042±0.0015	=	0.0038±0.0017

Table 3.18: *GD* comparison results (mean±std) of PICEA-g1 and PICEA-g2 for 7-objective WFG instances.

Problem	PICEA-g1		PICEA-g2
WFG4-7	0.0168±0.0006	=	0.0172±0.0008
WFG5-7	0.0179±0.0007	=	0.0184±0.0008
WFG6-7	0.0192±0.0008	=	0.0201±0.0012
WFG7-7	0.0204±0.0005	=	0.0211±0.0008
WFG8-7	**0.0210±0.0011**	>	0.0241±0.0015
WFG9-7	0.0174±0.0009	<	**0.0160±0.0011**

Table 3.19: Δ comparison results (mean±std) of PICEA-g1 and PICEA-g2 for 2-objective WFG instances.

Problem	PICEA-g1		PICEA-g2
WFG4-2	0.1614±0.0056	<	**0.1141±0.0049**
WFG5-2	0.2694±0.0201	<	**0.2014±0.0124**
WFG6-2	0.2517±0.0153	<	**0.2016±0.0197**
WFG7-2	0.5026±0.0429	<	**0.4392±0.0501**
WFG8-2	0.9136±0.0864	<	**0.6831±0.0546**
WFG9-2	0.2098±0.0191	<	**0.1831±0.0136**

Table 3.20: Δ comparison results (mean±std) of PICEA-g1 and PICEA-g2 for 7-objective WFG instances.

Problem	PICEA-g1		PICEA-g2
WFG4-7	4.0147±0.2214	<	**3.6104±0.2293**
WFG5-7	4.1134±0.2149	<	**3.9681±0.1990**
WFG6-7	**4.0893±0.2081**	=	4.1794±0.3026
WFG7-7	4.1614±0.2362	<	**3.6871±0.2095**
WFG8-7	4.9619±0.1620	<	**4.5912±0.1362**
WFG9-7	3.8218±0.1827	<	**3.5205±0.2391**

The Δ results for 2 and 7-objective problems are shown in Table 3.19 and Table 3.20, respectively. From the results, we can see that PICEA-g2 performs statistically better than PICEA-g1 on all the 2-objective problems, and on five out of the six 7-objective problems. Only for WFG6-7 does PICEA-g1 have a comparable performance to PICEA-g2.

Combining all the comparison results we can conclude that the use of *cutting plane* can significantly improve the performance of PICEA-g, especially for the diversity performance. Such results make a strong case for including this strategy when setting the goal vector bounds, when the *ideal* point of the problem is available.

The *ideal* point of a problem can be obtained either via expert domain-specific knowledge or preliminary single-objective optimisation. Given that the performance of PICEA-g is affected by the employed goal vector bounds, this section introduces a new strategy named the *cutting plane* which uses the information of the *ideal* point to adaptively construct suitable goal vector bounds. By setting proper goal vector bounds, PICEA-g is able to adjust the search effort towards different objectives appropriately during the search. The effectiveness of the *cutting plane* is examined on 2- and 7-objective WFG problems. Experimental results show that this strategy is helpful in improving the performance of PICEA-g for most of the problems.

3.8 Summary

Many-objective optimisation problems remain a challenging issue for most of the existing MOEAs, in obtaining a full and satisfactory approximation of Pareto optimal solutions. This Chapter has studied a novel class of algorithm, namely, the preference-inspired co-evolutionary algorithm, for many-objective problems. The main achievements are summarised as follows:

- Co-evolution of a family of decision-maker preferences with a population of candidate solutions is demonstrated to have promising performance characteristics for many-objective optimisation problems. After introducing the concept of the preference-inspired co-evolutionary algorithm (PICEA), a realisation of this concept, PICEA-g, is implemented and is rigorously compared to four best-in-class MOEAs (NSGA-II, ε-MOEA, MOEA/D and HypE) and a random search benchmark on the leading WFG test problems with 2, 4, 7 and 10 objectives. Experimental results show that PICEA-g has competitive performance with NSGA-II on bi-objective problems, and a better performance than other algorithms on many-objective problems.
- Based on the assumption that the *ideal* point of a problem is known *a priori*, an effective strategy named the *cutting plane* is proposed to further improve the

performance of PICEA-g. The *cutting plane* utilises knowledge of the *ideal* point to determine proper goal vector bounds and then leads to a generation of suitable goal vectors along the search process. This then enables PICEA-g to adaptively distribute the search effort towards different objectives appropriately. Thereby, leading to a better performance. The effectiveness of this strategy is examined on 2- and 7-objective WFG problems. Experimental results show this strategy has positive effect on PICEA-g.

The goal vector approach examined in this Chapter represents just one possible formulation of the preference-inspired co-evolutionary algorithm. In the next Chapter, we will study the potential of co-evolving another type of decision-maker preferences.

Additionally, there needs to be a systematic analysis of how performance varies with the tunable parameters of PICEA-g (this will be conducted in Chapter 5). Such analysis is important to gain an insight into the robustness of the algorithms on parameter settings and to provide further suggestions on parameter settings for non-expert use.

Chapter 4

Preference-inspired co-evolutionary algorithm using weight vectors

4.1 Introduction

Chapter 3 has demonstrated the benefits of co-evolution of candidate solutions with preferences. The co-evolution enables suitable preferences to be constructed adaptively and therefore guide candidate solutions towards the Pareto optimal front effectively. Given the success of PICEA-g, a natural consideration is that what might be the benefits of co-evolving candidate solutions with other types of preferences, such as weights.

A decomposition based algorithm is a popular class of evolutionary multi-objective optimiser that uses weights – it transfers a MOP into a set of single objective problems by means of scalarising functions with different weights. Weights are used to define the search directions. An approximation of the Pareto optimal front can be obtained by solving a range of the single objective problems.

Compared with Pareto-dominance based algorithms, decomposition based algorithms have a number of advantages, such as high search ability and computational efficiency on fitness evaluation (Zhang and Li, 2007; Li and Zhang, 2009). However, more recent studies have demonstrated that decomposition based algorithms, e.g., MOEA/D (Zhang and Li, 2007), face difficulties on problems that have complex Pareto optimal fronts (Gu et al., 2012); also though they perform well on bi-objective problems, are not particularly effective on many-objective problems due to a loss of diversity (Wang et al., 2013). These issues might be due to an inappropriate specification of weight vectors (this will be discussed later), itself arising from a lack of knowledge on the underlying problem structure.

In this Chapter, we study how decomposition based methods can benefit from the co-evolution of candidate solutions and weights. A new decomposition based algorithm called preference-inspired co-evolutionary algorithm using weights (PICEA-w) is proposed, which eliminates the need to specify appropriate weights in advance of performing the optimisation. The rest of this Chapter is structured as follows: in Section 4.2, basics of decomposition based algorithms are introduced. Section 4.3 analyses the issues of decomposition based algorithms. Section 4.4 elaborates our proposed algorithm PICEA-w. Section 4.5 describes the experimental setup. The experimental results and discussions are provided in Section 4.6 and Section 4.7, respectively. Section 4.8 summarises this Chapter.

4.2 Basics of decomposition based MOEAs

As previously mentioned, decomposition based algorithms handle a MOP by simultaneously solving a set of single objective problems defined by means of scalarising functions with different weights. Scalarising functions and weights are two basics for a decomposition based algorithm, and will be introduced in Section 4.2.1 and Section 4.2.2, respectively. Additionally, a decomposition based algorithm often requires a normalisation procedure; this will be introduced in Section 4.2.3.

4.2.1 Scalarising functions

A variety of scalarising functions have been introduced for decomposition based algorithms. The weighted sum and the weighted Chebyshev function are two frequently-used scalarising functions, and can be written as follows.

- The weighted sum scalarising function:

$$g^{ws}(\mathbf{x}|\mathbf{w}) = \sum_{i=1,2,\cdots,M} \{w_i(f_i(\mathbf{x}) - z_i^*)\} \tag{4.1}$$

- The weighted Chebyshev scalarising function:

$$g^{te}(\mathbf{x}|\mathbf{w}) = \max_{i=1,2,\cdots,M} \{w_i(f_i(\mathbf{x}) - z_i^*)\} \tag{4.2}$$

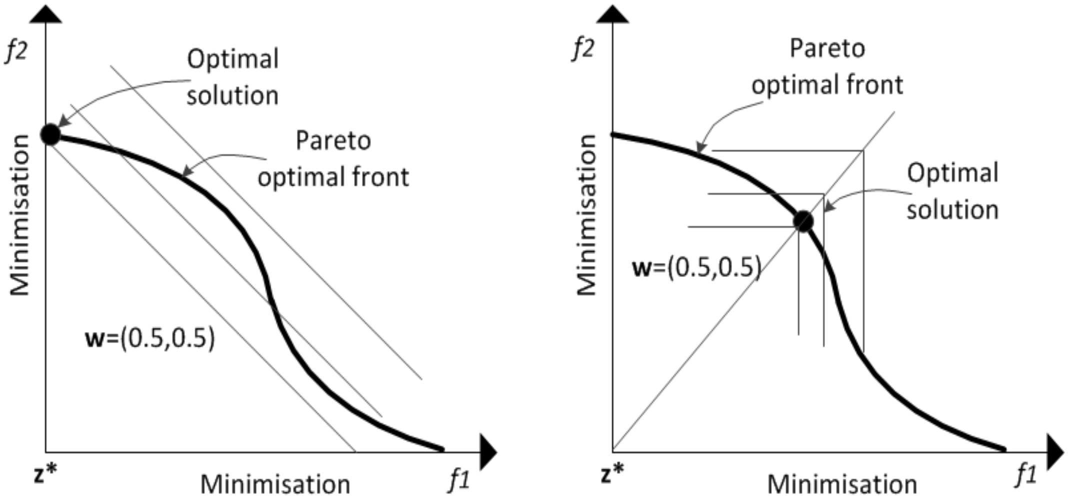
For minimisation problems, both $g^{ws}(\mathbf{x}|\mathbf{w})$ and $g^{te}(\mathbf{x}|\mathbf{w})$ should be minimised. In both Equation 4.1 and Equation 4.2, $\mathbf{w} = (w_1, w_2, \cdots, w_M)$ represents a weight vector, $w_i \geq 0$, $\sum_{i=1}^{i=M} w_i = 1$ and $\mathbf{z}^* = (z_1^*, z_1^*, \cdots, z_M^*)$ is a reference point. Typically, the reference point is updated once a better (smaller) value of f_i is found during the algorithm execution, see Equation 4.3.

$$z_i^* = \min \{f_i(\mathbf{x}) | \mathbf{x} \in \mathbf{\Omega}\} \tag{4.3}$$

where $\mathbf{\Omega}$ represents all the examined solutions during the algorithm execution. Ideally, the reference point should be set as the *ideal* point of the problem.

The optimal solution of each single objective problem, defined by a weighted scalarising function, corresponds to one Pareto optimal solution of a MOP. The weight vector defines a search direction for the scalarising function. Thus, one can use different weights to search for a set of diversified Pareto optimal solutions.

It is worth mentioning that the weighted sum scalarising function encounters difficulties with certain problem geometries[1], that is, the fact it is not possible to find Pareto optimal solutions in concave regions of the Pareto optimal front unless some addition technique (e.g., ε-constraint method) is applied (Kim and De Weck, 2005, 2006). The Chebyshev scalarising function does not have such issue, see Figure 4.1. However, its search ability is not as high as the weighted sum approach (Ishibuchi et al., 2009a).



(a) The weighted sum method

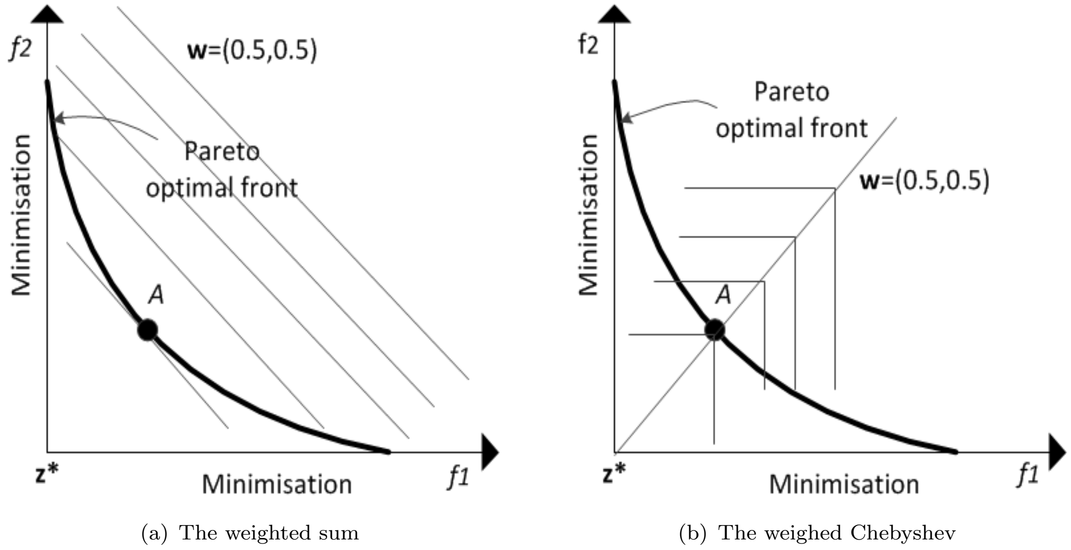
(b) The Chebyshev function

FIGURE 4.1: Behaviours of the weighted sum and Chebyshev scalarising function on non-convex Pareto front.

Next we briefly analyse the search abilities of the two scalarising functions based on the study of Giagkiozis and Fleming (2012). Figure 4.2 shows contour lines of each of the scalarising functions in a bi-objective case with reference point $\mathbf{z}^*$ at the origin and weight vector $\mathbf{w} = (0.5, 0.5)$. The objective-space is divided into two sub-spaces by the contour line. Solutions in one sub-space are better than solutions on the contour line while solutions in the other sub-space are worse. Solutions that lie on the same contour line have the same scalar objective value. In Figure 4.2, solution A is the optimal solution of $g^{ws}(\mathbf{x}|(0.5, 0.5))$ and $g^{te}(\mathbf{x}|(0.5, 0.5))$.

[1] Note that in this thesis problem geometry specifically refers to Pareto front geometry

The contour line of the weighted sum approach is a line, and the contour line of the weighted Chebyshev approach is a polygonal line (with vertical angle). According to the shape of the contour line we can observe that for the weighted sum approach the size of a better region is always half of the whole objective-space regardless of the number of objectives. This indicates that the probability of replacement of an existing solution by a newly generated solution is always $\frac{1}{2}$. That is, the search ability of the weighted sum approach is not affected by an increase in the number of objectives. With respect to the weighted Chebyshev function, a better region roughly equals to $(\frac{1}{2}^{M})$ of the M-dimensional objective-space. This indicates that the probability of replacement decreases significantly as the number of objective increases. That is, the search ability of the weighted Chebyshev function deteriorates as the number of objectives increases.



(a) The weighted sum

(b) The weighed Chebyshev

FIGURE 4.2: Contour lines of scalarising functions.

In fact the weighted sum and the weighted Chebyshev approach can be unified into a single formulation, known as the weighted L_p function.

$$g^{L_p}(\mathbf{x}|\mathbf{w}) = \sum_{i=1,2,\cdots,M} \left\{ w_i(|f_i(\mathbf{x}) - z_i^*|^p)^{1/p} \right\} \tag{4.4}$$

where $p \in (0, \infty]$. When $p = 1$, g^{L_p} represents the weighted sum approach, and when $p = \infty$, g^{L_p} represents the weighted Chebyshev function, minimising the largest deviation $|f_i(\mathbf{x}) - \mathbf{z}_i^*|$. Contour lines of the weighted L_p function with different p values are shown in Figure 4.3 (Giagkiozis and Fleming, 2012). It is observed that the search ability of the weighted L_p function decreases as p increases.

All the weighted L_p functions suffer from the geometry issue except for the Chebyshev function. Specifically, a L_p function cannot identify the whole Pareto optimal front of a MOP when the curvature of the Pareto optimal front is larger than the curvature of the

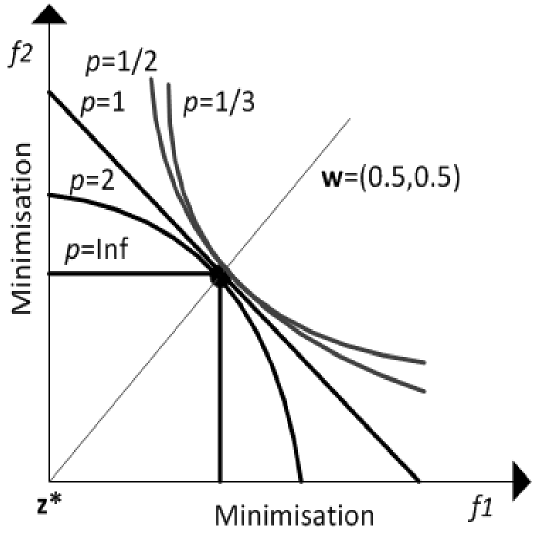


FIGURE 4.3: Contour lines of the weighted L_p function with different p values.

contour line of the chosen L_p function. Since the curvature of the Chebyshev function is ∞, it is able to identify Pareto optimal solutions for any type of geometry.

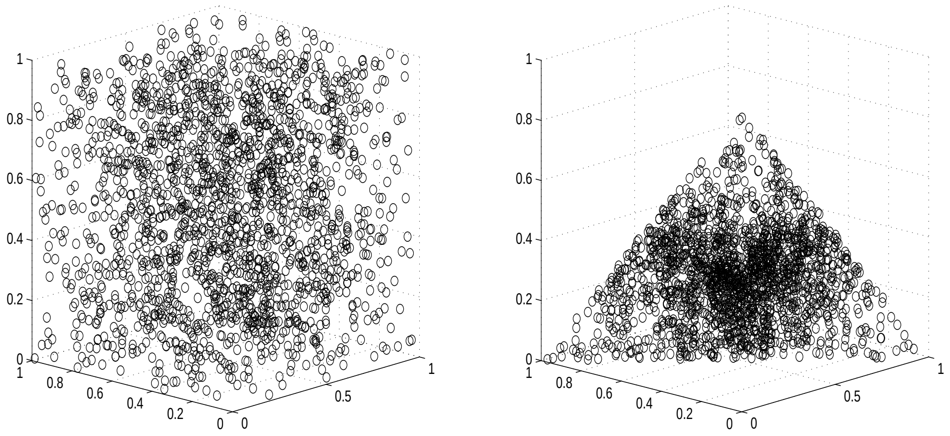
Overall, for a L_p scalarising function its search ability and its robustness on problem geometries are a trade-off – the higher the search ability, the lower the robustness. If we can estimate the curvature of the Pareto optimal front *a priori*, then we can easily select a suitable L_p function, based on its search ability. For example, if the curvature of the Pareto optimal front is quadratic, the most suitable L_p function should be the one using $p = 2$. Alternatively, for a new problem or a problem having a complex geometry, the use of the Chebyshev function is a good choice.

4.2.2 Weights

Diversified solutions can be obtained by employing different weights in decomposition based algorithms. Typically, the employed weights can either be pre-defined before the search or adaptively modified during the search. This section introduces two frequently-used methods that generate random weight vectors and evenly distributed weights.

Randomly generated weights

To generate a random weight vector, the easiest way that has been used in many studies, (e.g. Hajela et al., 1993; Ishibuchi and Murata, 1998) is first to randomly generate M numbers, e.g., $a_1, a_2, \cdots, a_M$ and $a_i \geq 0$ and then to normalise a_i by $\frac{a_i}{\sum_{i=1}^{M} a_i}$ to obtain a valid component of weight vector, i.e., $w_i = \frac{a_i}{\sum_{i=1}^{M} a_i}$. However, this method has a limitation, that is, the generated weights are dense in the centre while sparse at the edge. This is because the method equals to directly projecting all randomly distributed points in a hypercube at a hyperplane, see Figure 4.4.



(a) random points in a cube

(b) random weights on a plane

FIGURE 4.4: Illustration of the distribution of random weights.

Jaszkiewicz (2002) proposed another way to generate random weights which does not face the issue as shown in Figure 4.4. Components of a weight vector are calculated by Equation 4.5, where function $rand()$ returns a random value within the interval $(0, 1)$.

$$\begin{aligned} w_1 &= 1 - \sqrt[M-1]{rand()} \\ &\cdots \\ w_i &= (1 - \sum_{j=1}^{i-1} w_j)(1 - \sqrt[M-1]{rand()}) \\ &\cdots \\ w_M &= 1 - \sum_{j=1}^{i-1} w_j \end{aligned} \tag{4.5}$$

Evenly distributed weights

To generate evenly distributed weights on a hyperplane, many methods are available such as the simplex-lattice design, simplex-centroid design and axial design and uniform design (Tan et al., 2012). Amongst these methods, the simplex-lattice design method is frequently used (MOEA/D employs this method). Weights are formed by all normalised weight vectors with components chosen from the set $\{0, 1/H, \cdots, (H-1)/H, 1\}$ where H is a positive integer number, see Figure 4.5(a). For example, for 2-objective problems, if H is specified as 100, then we can generate $C_{101}^{1} = 101$ groups of weight vectors (0, 1), (0.01, 0.99), ... , (1, 0). The number of weights generated using this method increases significantly when M increases; it is determined by C_{M-1}^{H+M-1}. Given $H = 30$, the number of weights is 5456 for $M = 4$, while the number increases to 46376 when $M = 5$.

Hughes (2003) proposed another approach which generates evenly distributed points on a hypersphere by minimising the metric V defined by Equation 4.6, see Figure 4.5(b). These points can be transformed into valid weights by using the equation: $w_i = \frac{x_i}{\sum_{i=1}^{M} x_i}$ where x_i is the ith component of a point.

$$V = \max_{i=1}^{N} \max_{j=1,j\neq i}^{N} (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{4.6}$$

The metric V measures the worst-case angle of two nearest neighbours. The dot product $\mathbf{x}_i \cdot \mathbf{x}_j$ provides the cosine of angle between $\mathbf{x}_i$ and $\mathbf{x}_j$. The inner maximisation finds the nearest two neighbours in terms of the angle between them. The outer maximum operator finds the largest angle between two nearest neighbours. The optimal set of weights is produced when the outer maximum operator is minimised.

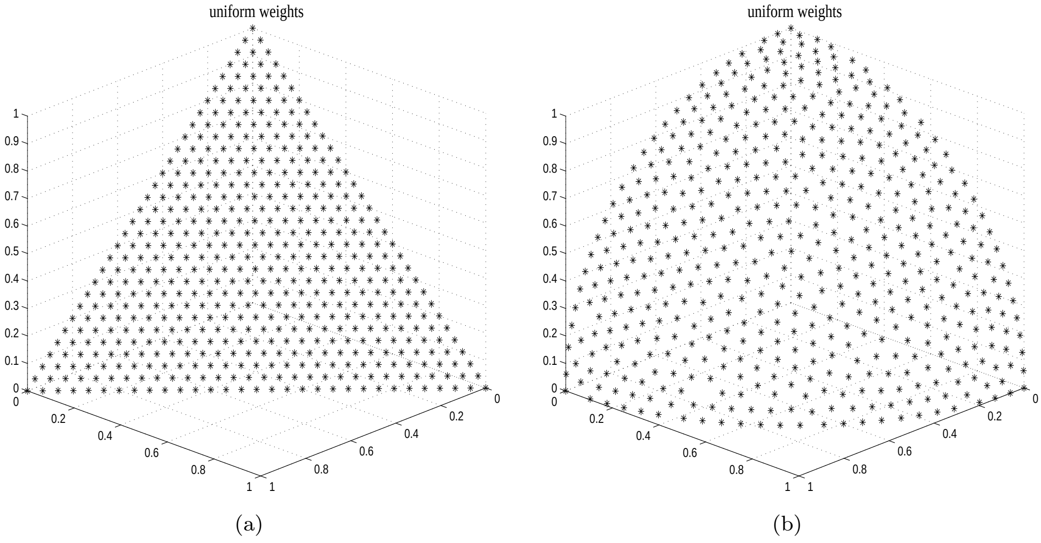


FIGURE 4.5: Evenly distributed weights on a hyperplane and hypersphere in 3-objective case.

4.2.3 Normalisation

Decomposition based algorithms combine different objectives into one metric. These objectives might have various units of measurement. Thus, it is important to rescale different objectives to dimension-free units before aggregation. Moreover, normalisation is useful for obtaining evenly distributed solutions when the objectives are disparately scaled.

Typically, the normalisation procedure transforms an objective value f_i (in Equations 4.1 or 4.2) by

$$\overline{f}_i = \frac{f_i - z_i^{ide}}{z_i^{nad} - z_i^{ide}} \tag{4.7}$$

where $\mathbf{z}^{\mathbf{ide}} = (z_1^{ide}, z_2^{ide}, \cdots, z_M^{ide})$ is the *ideal* point and $\mathbf{z}^{\mathbf{nad}} = (z_1^{nad}, z_2^{nad}, \cdots, z_M^{nad})$ is the *nadir* point. After normalisation, objective values are within $[0, 1]$.

If the true $\mathbf{z}^{\mathbf{ide}}$ and $\mathbf{z}^{\mathbf{nad}}$ are not available (they are often difficult to obtain, especially for $\mathbf{z}^{\mathbf{nad}}$ (Deb et al., 2010)), we can use the smallest and largest f_i of all non-dominated solutions found so far to estimate the z_i^{ide} and z_i^{nad}, respectively.

4.3 General issues of decomposition based algorithms

This section discusses two difficulties that are usually encountered by decomposition based algorithms: problem geometry and many-objective optimisation.

4.3.1 Problem geometry

In decomposition based algorithms, each Pareto optimal solution corresponds to an optimal solution of a single objective problem that is defined by a weighted scalarising function. That is to say, once the weighted scalarising function is determined, the distribution of the obtained Pareto optimal solutions is determined. Furthermore, if the scalarising function is also chosen, the distribution of solutions would only be affected by the distribution of the employed weights.

Gu et al. (2012) and Giagkiozis et al. (2013) discussed what the optimal distribution of weights is for a specified Pareto front when using different scalarising functions. For example, when the Chevbyshev scalarising function is used, the optimal weight for searching for a solution $\mathbf{x}$ is $(\frac{\frac{1}{f_1(\mathbf{x})}}{\sum_{i=1}^{M}\frac{1}{f_i(\mathbf{x})}}, \frac{\frac{1}{f_2(\mathbf{x})}}{\sum_{i=1}^{M}\frac{1}{f_i(\mathbf{x})}}, \cdots, \frac{\frac{1}{f_M(\mathbf{x})}}{\sum_{i=1}^{M}\frac{1}{f_i(\mathbf{x})}})$. That is, given a weight vector $\mathbf{w}$, the obtained Pareto optimal solution is along the search direction of $\frac{1}{\mathbf{w}}$ (as long as the search direction $\frac{1}{\mathbf{w}}$ does not point at a disconnected Pareto region).

Specifically, this relationship is described using Figure 4.6. The straight line from the reference point along the direction $(\frac{1}{w_{i,1}}, \frac{1}{w_{i,2}})$ can be described as $f_1 w_{i,1} = f_2 w_{i,2}$. Note that $w_{i,j}$ represents the jth component of weight vector $\mathbf{w}_i$. Line A intersects the Pareto front at point $\mathbf{s}_1$. It is easy to see that $\min \max(\mathbf{w}_1 f(\mathbf{x})) = \max(\mathbf{w}_1 f(\mathbf{x}_1))$, where $\mathbf{x}_1$ corresponds to the decision vector of $\mathbf{s}_1$. Thus, we can conclude that weight vector $\mathbf{w}_i = (w_{i,1}, w_{i,2})$ corresponds to a Pareto optimal solution along the direction of $(\frac{1}{w_{i,1}}, \frac{1}{w_{i,2}})$. In other words, the optimal weight vector corresponding to a Pareto optimal solution $\mathbf{x}_1$ is determined by the vector $(\frac{f_2(\mathbf{x}_1)}{f_1(\mathbf{x}_1)+f_2(\mathbf{x}_1)}, \frac{f_1(\mathbf{x}_1)}{f_1(\mathbf{x}_1)+f_2(\mathbf{x}_1)})$. It is worth mentioning that this conclusion is based on the assumption that there is a Pareto optimal

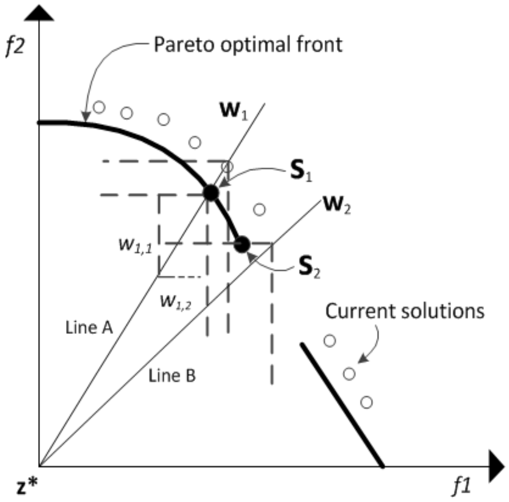


FIGURE 4.6: Illustration of the relation between weights and Pareto optimal solutions for a Chebyshev scalarising function.

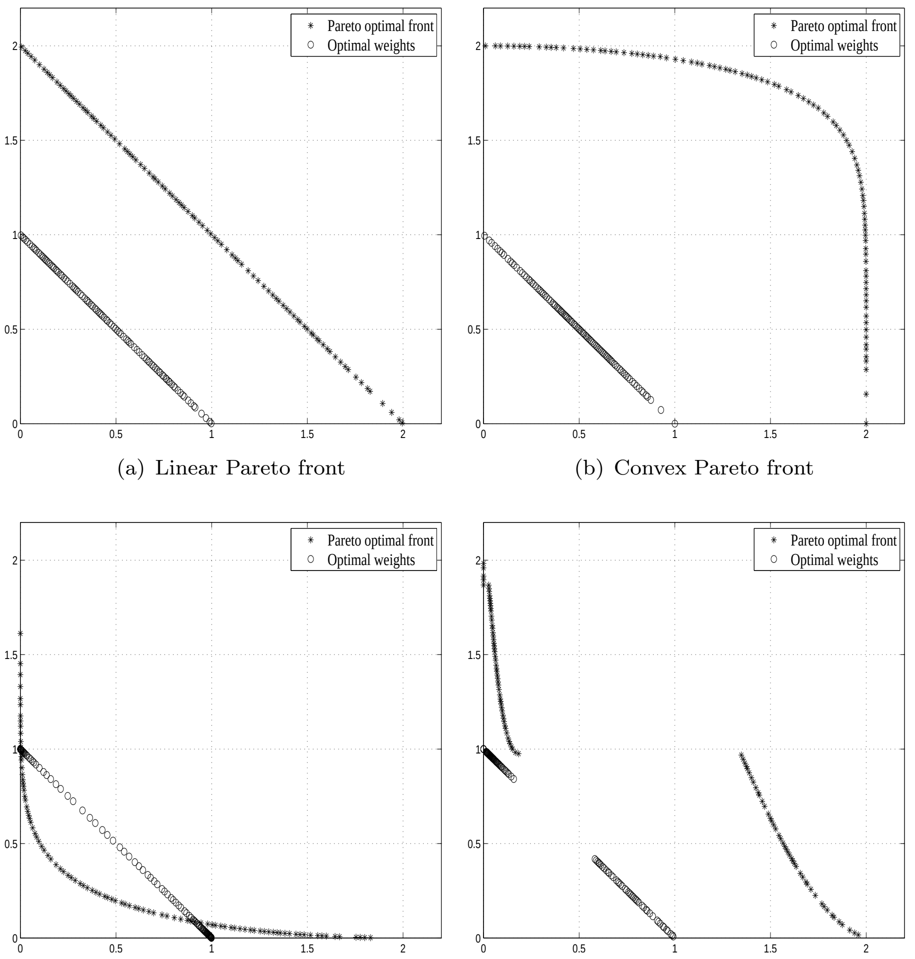
solution along the defined search direction. If the search direction defined by a weight vector (e.g., $\mathbf{w}_2$) points at a disconnected region (termed *boundary* search direction), a solution at the boundary (e.g. $\mathbf{s}_2$) will be identified as the optimal solution for this weight vector instead. This is because the decision vector $\mathbf{x}_2$ of $\mathbf{s}_2$ produces the minimal value for the expression $\max(\mathbf{w}_2 f(\mathbf{x}))$.

Given the above analysis, it is easy to see that the optimal distribution of weights for different problem geometries changes. Figure 4.7 illustrates the optimal distributions of weights for problems having linear, convex, concave or disconnected Pareto optimal fronts. Taking Figure 4.7(b) as an example, the optimal distribution of weights for a concave Pareto front is dense in the centre while sparse at the edge.

Overall, due to a lack of knowledge of the underlying problem geometry, it is usually not straightforward to determine a proper distribution of weights *a priori* for decomposition based algorithms so as to obtain a set of evenly distributed solutions. Although the use of adaptive weights is potentially helpful in handling this issue (Jiang et al., 2011; Gu et al., 2012), it is suspected that adaptive weights might have a deleterious effect on an algorithm's convergence performance (this will be discussed next).

4.3.2 Many-objective optimisation

Decomposition based algorithms using evenly distributed weights, such as MOEA/D, face difficulties on many-objective problems. This is because the number of Pareto optimal solutions that are required to describe the entire Pareto optimal front of a MaOP is very large (Ishibuchi et al., 2008b). In decomposition based algorithms each weight vector typically corresponds to one Pareto optimal solution. The evenly distributed weights are often initialised before the search and remain unchanged during the search. It



(a) Linear Pareto front

(b) Convex Pareto front

(c) Concave Pareto front

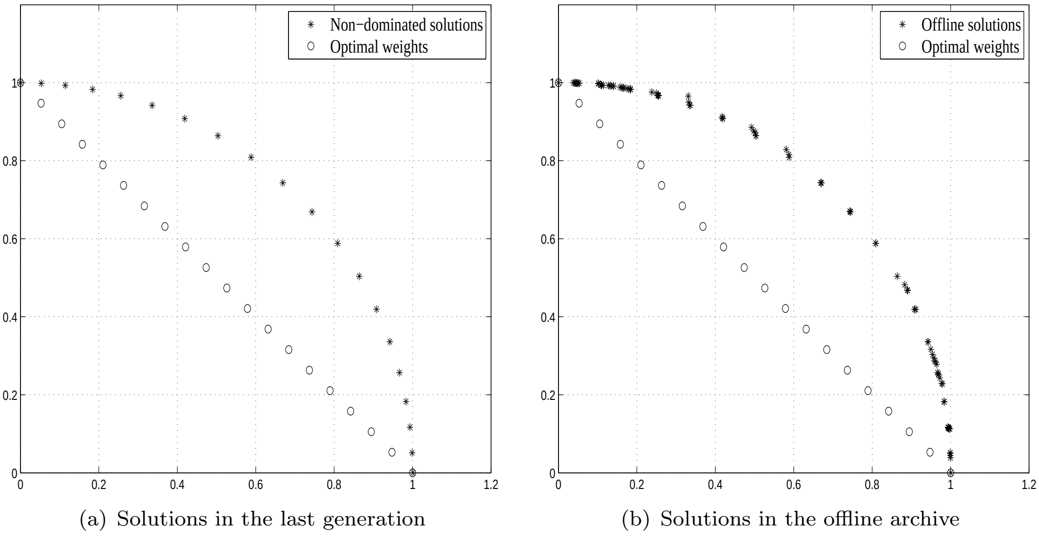
(d) Disconnected Pareto front

Figure 4.7: The optimal distributions of weights for different Pareto fronts in a 2-objective case using a Chebyshev scalarising function.

is therefore difficult to use a limited number of weights to obtain a full and representative approximation of the entire Pareto optimal front.

To illustrate this issue, we apply MOEA/D with 20 evenly distributed weights to solve the 2-objective DTLZ2 problem. Figure 4.8(a) and Figure 4.8(b) show the obtained non-dominated solutions in the last generation and in the archive respectively after running MOEA/D for 500 generations. It is obvious that the obtained solutions are not sufficient to cover the entire Pareto optimal front. It should be noted that due to the stochastic nature of MOEAs, neighbouring solutions of the $\mathbf{s}_w$ are likely to be obtained during the search. $\mathbf{s}_w$ is referred as the optimal solution of a single objective problem defined by the weighted scalarising function $g(\mathbf{x}|\mathbf{w})$. However, it is less likely to find solutions that are distant from $\mathbf{s}_w$, see Figure 4.8(b).

A natural way to address this limitation, i.e., a lack of solution diversity, is by employing a large number of weights. However, it is argued that compared with the number of solutions required to describe the entire Pareto optimal front, the number of employed

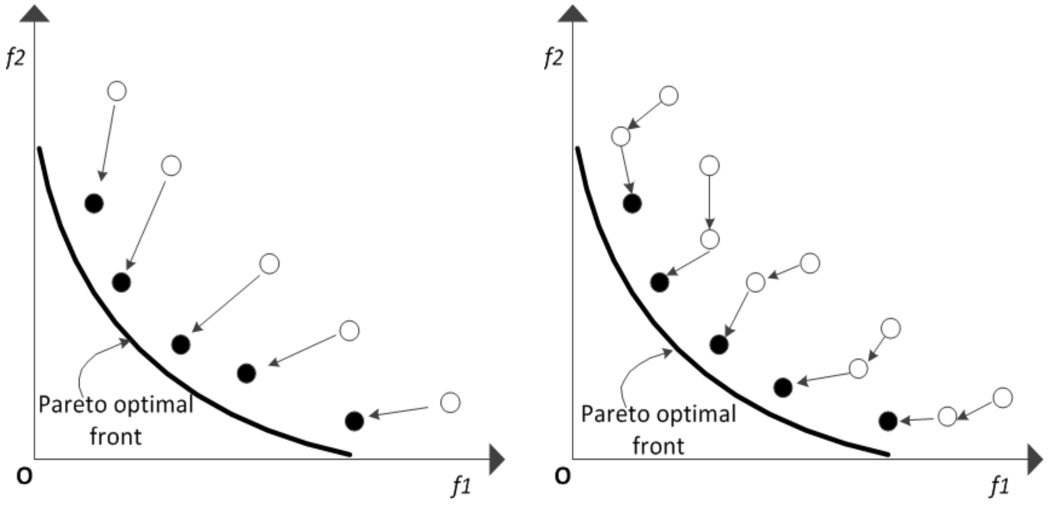


(a) Solutions in the last generation

(b) Solutions in the offline archive

Figure 4.8: An approximation of Pareto front of 2-objective DTLZ2 obtained by MOEA/D using 20 weights.

weights is always relatively small. Besides, for some decomposition based MOEAs, e.g., MOEA/D, the population size is required to be equal to the number of weights. It is not easy to strike an effective balance between the population size and the generations with a fixed computational budget – the larger the population size, the the more the beneficial dynamics of evolution are curtailed.



(a) fixed weights

(b) non-fixed weights

Figure 4.9: Illustration of the search behaviour using fixed weights and non-fixed weights.

Another alternative is to use non-fixed weights. Typically, non-fixed weights could be either randomly generated or adaptively modified during the search. The use of non-fixed weights enables MOEAs to have more opportunities to explore different regions, thereby obtaining a set of diversified solutions. However, this might slow down the convergence speed of an algorithm. When using fixed weights, solutions are guided towards the Pareto optimal front along the search directions constructed by the weights, see Figure 4.9(a). When using non-fixed weights, the constructed search directions keep changing. This suggests that solutions are guided towards the Pareto optimal front in a polyline trajectory as shown in Figure 4.9(b), that is, the convergence speed is degraded. Certainly, it should be admitted that in some case, e.g., multi-modal problems, the use of random/adaptive weights is helpful to maintain diversified solutions and to prevent the algorithm being trapped in the local optima, resulting in better convergence.

Overall, decomposition based algorithms face difficulties with many-objective optimisation. This issue has not received much attention. To the best of the author's knowledge, none of the existing decomposition based algorithms that use adaptive weights, such as Hughes (2007), Li and Landa-Silva (2011) and Gu et al. (2012) can effectively strike a balance between exploitation (convergence) and exploration (diversity). Therefore, it is proposed to develop an effective weights adaptation strategy to address this issue.

4.4 PICEA-w

This section introduces a novel weights adaptation method based on the concept of PICEA. Weights maintain their usefulness by being co-evolved with candidate solutions along the search process. The co-evolution enables suitable sets of weights to be constructed adaptively, and thus guide candidate solutions towards the Pareto optimal front effectively. We refer to this new strategy as a preference-inspired co-evolutionary algorithm using weight vectors (PICEA-w).

4.4.1 Algorithm design: PICEA-w

Similar to a general decomposition based algorithm, PICEA-w decomposes a MOP into a set of single objective problems that are defined by different weighted scalarising functions. The main feature of PICEA-w is that the scalarising functions' underlying weights are adaptively modified in a co-evolutionary manner during the search. Specifically, in PICEA-w candidate solutions are ranked by each of the weighted scalarising functions, creating a ranking matrix. The fitness of candidate solutions is then calculated based on the ranking matrix. Weights are co-evolved with the candidate solutions towards an optimal distribution, and are also responsible for striking a balance between exploration and exploitation.

We implement PICEA-w within a $(\mu+\lambda)$ elitist framework as shown in Figure 4.10. A population of candidate solutions and weight vectors, S and W, of fixed size, N and N_w, are co-evolved for a number of generations. At each generation t, parents $S(t)$ are subjected to genetic variation operators to produce N offspring, $Sc(t)$. Simultaneously,

N_w new weight vectors, $Wc(t)$, are randomly generated. $S(t)$ and $Sc(t)$, $W(t)$ and $Wc(t)$, are then pooled respectively and the combined populations are sorted according to their fitness. Truncation selection is applied to select the best N solutions as a new parent population, $S(t+1)$, and N_w solutions as a new preference population, $W(t+1)$. Additionally, an offline archive is employed to store all the non-dominated solutions found during the search. Evenly distributed solutions are obtained by using the clustering technique described in Zitzler et al. (2002).

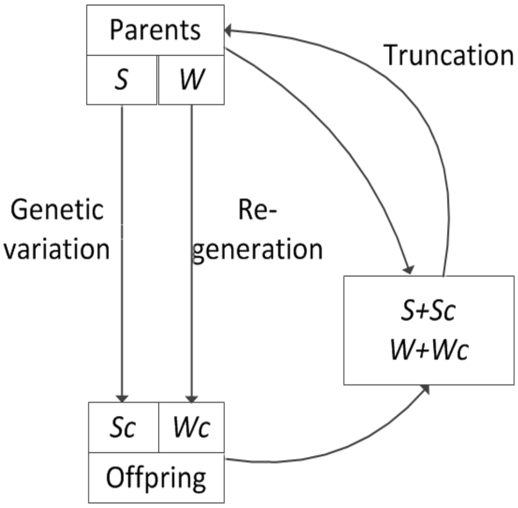


FIGURE 4.10: A (μ+λ) elitist framework of PICEA-w.

Algorithm 2: Preference-inspired co-evolutionary algorithm using weights

```
Input: Initial candidate solutions, $S$ of size $N$, initial weight vectors, $W$ of size $N_w$,
       the maximum number of generations, $maxGen$, the number of objectives, $M$,
       archive size, $ASize$
Output: $S, W$, offline archive, $BestF$
$BestF \leftarrow \emptyset$;
$S \leftarrow$ initialiseS($N$);
$F\_S \leftarrow$ objectiveFunction($S$);
$BestF \leftarrow$ updateArchive($BestF, F\_S$);
$W \leftarrow$ weightGenerator($N_w$);
for $gen \leftarrow 1$ to $maxGen$ do
    $Sc \leftarrow$ geneticOperation($S$,$F\_S$);
    $F\_Sc \leftarrow$ objectiveFunction($Sc$);
    $(JointS, JointF) \leftarrow$ multisetUnion($S, Sc$,$F\_S, F\_Sc$);
    $Wc \leftarrow$ weightGenerator($N_w$);
    $JointW \leftarrow$ multisetUnion($W, Wc$);
    $\theta \leftarrow$ thetaConfiguration($gen, \pi/2$);
    $(S, F\_S, W) \leftarrow$ coEvolve($JointF, JointS, JointW, \theta$);
    $BestF \leftarrow$ updateArchive($BestF, F\_S, ASize$);
end
```

The pseudo-code of PICEA-w is presented in Algorithm 2. In the following we explain the main steps of PICEA-w.

- Line 1 initialises the offline archive $BestF$ as $\emptyset$.
- In lines 2 and 3, N candidate solutions S are initialised and their objective values F_S are calculated. The offline archive, $BestF$ is updated by function `updateArchive` in line 4.
- Line 5 applies function `weightGenerator` to generate N_w random weights.
- The offspring candidate solutions Sc are generated by function `geneticOperation` in line 7. Their objective values F_Sc are calculated in line 8. S and Sc, F_S and F_Sc are pooled together, respectively in line 9.
- Line 10 generates another set of weights Wc. In line 11, W and Wc are pooled together.
- Line 12 sets the parameter θ which would be used in function `coEvolve` to implement a local operation.
- Line 13 co-evolves the joint candidate solutions $JointS$ and the joint weights $JointW$, and so to obtain new parents S and W.
- Line 14 updates the offline archive with newly obtained solutions F_S.

Note that function `weightGenerator` forms a new weight set Wc with N_w weight vectors that are generated according to Equation 4.5 (Jaszkiewicz, 2002). Alternatively, Wc can be formed by randomly sampling N_w weights from an evenly distributed candidate weight set, Ψ. Ψ can be created by the simplex-lattice design method (as described on p.107). Genetic operators are not applied to generate offspring weight vectors, the reason is the same as the case in PICEA-g, which is described in p. 56. In addition, functions `geneticOperation` and `updateArchive` are the same as that used in PICEA-g, see p. 56.

Function `thetaConfiguration` adjusts parameter θ by Equation 4.8, that is, θ increases linearly from a small value to $\frac{\pi}{2}$ radians.

$$\theta = \frac{\pi}{2} \times \frac{gen}{maxGen}; \tag{4.8}$$

The use of θ implements a local selection at the early stages of the evolution and implements a global selection at the later stages of the evolution. A local selection refers to a candidate solution that only competes with its neighbours, and a global selection refers to a candidate solution that competes with all the other candidate solutions. The benefits of this strategy will be illustrated later (in Section 4.7.2).

The core part of PICEA-w lies in the function `coEvolve` which will be elaborated next. Function `coEvolve` evaluates the performance of candidate solutions and weight vectors, and then constructs new parent populations S and W from the joint populations $JointS$

and $JointW$, respectively. A candidate solution gains high fitness by performing well on a large number of weighted scalarising functions. A weight vector only gains fitness by being rewarded from the candidate solutions that are ranked as the best by this weight. In other words, a weight vector is considered to be good only if it has contribution on the survived candidate solutions. The pseudo-code of function `coEvolve` is as follows.

```
Function coEvolve($JointS, JointF, W$)
Input: The joint populations $JointS, JointF, JointW$, the parameter $\theta$
Output: New parents, $S, F\_S, W$
1 $R \leftarrow$ rankingSW($JointF, JointW, \theta$);
2 $(F\_S, S, ix) \leftarrow$ selectS($JointF, JointS, R$);
3 $W \leftarrow$ selectW($JointW, F\_S, R, ix$);
```

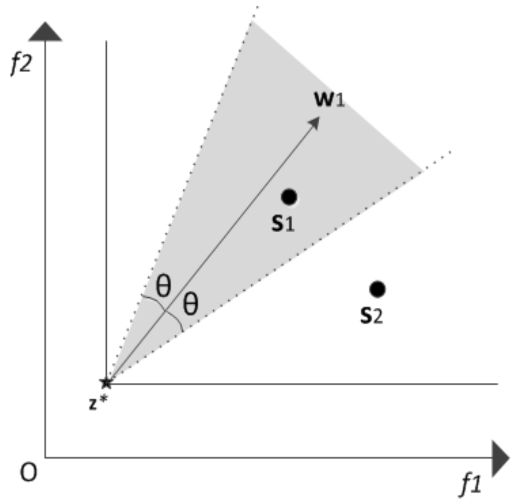


FIGURE 4.11: Illustration of the neighbourhood of candidate solutions and weight vectors.

(i) Line 1 applies function `rankingSW` to rank $JointF$ by each weighted scalarising function. Specifically, for each $\mathbf{w} \in JointW$, we first identify its neighbouring candidate solutions. The neighbourhood is measured by the angle between a candidate solution, $\mathbf{s}$ and a weight vector, $\mathbf{w}$. If the angle is smaller than θ, then $\mathbf{s}$ and $\mathbf{w}$ are defined as neighbours. For example, in Figure 4.11, $\mathbf{s}_1$ is a neighbour of $\mathbf{w}$ while $\mathbf{s}_2$ is not. Then we rank these neighbouring candidate solutions based on their performance measured by the corresponding weighted Chebyshev scalarising function. This produces a $[2N \times 2N_w]$ ranking matrix, denoted as R. R_{ij} represents the rank of the candidate solution $\mathbf{s}_i$ on the weighted Chebyshev function using $\mathbf{w}_j$, i.e., $g^{te}(\mathbf{s}_i|\mathbf{w}_j)$. The best solution is ranked 1. Ranking values for solutions that are not neighbours of the $\mathbf{w}$ are set as $\inf$. The Chebyshev scalarising function is used in PICEA-w due to its guarantee of producing a Pareto optimal solution for each weight vector, and also its robustness on problem geometries (Miettinen, 1999).

(ii) Line 2 applies function `selectS` to select the best N candidate solutions as new parents, S. Specifically, the following steps are executed.

- Sort each row of the ranking matrix R in an ascending order, producing another matrix R_{sort}. R_{sort} holds in the first column the best ranking (the smallest) value achieved by each candidate solution across all the weighted scalarising functions. Simultaneously, the second column holds the second smallest ranking result for each candidate solution and so on.
- All candidate solutions are lexicographically ordered based on R_{sort}. This returns a rank (denoted as r) for each candidate solution. The fitness of a candidate solution is then determined by $2N - r$. Truncation selection is applied to select the best N solutions as new parents.

(iii) Line 3 applies function `selectW` to select a suitable weight vector from $JointW$ for each of the survived candidate solutions, i.e., members in the new parent S. The basic idea of the selection is to balance exploration and exploitation. To do so, two criteria are set for the weights selection. First, for each candidate solution the selected weight must be the one that ranks the candidate solution as the best. Then, if more than one weight is found by the first criterion, the one that is the furthest from the candidate solution is chosen. The first criterion is helpful to drive the search quickly towards the Pareto front. The second criterion is helpful to guide the search to explore new areas.

More specifically, for a candidate solution $\mathbf{s}_i$, first we identify all the weights that rank $\mathbf{s}_i$ the best. If there is only one weight, then this weight is selected for $\mathbf{s}_i$. This guarantees that solution $\mathbf{s}_i$ will not lose its advantage in the next generation as there is still one weight vector that ranks this candidate solution the best, unless a better solution along this direction is generated. If so, the convergence performance is improved. If more than one weight is found, we select the weight that has the largest angle between $\mathbf{s}_i$ and itself. In this way, the algorithm is guided to investigate some unexplored areas, i.e., improving diversity. It is worth mentioning that the second criterion is not considered unless multiple weights are identified by the first criterion. This guarantees that when exploring for a better diversity, the exploitation for a better convergence is also maintained. The pseudo-code of the function `selectW` is described as follows.

- Line 1 initialises the new weight set as empty, $\emptyset$. Line 2 forms a new ranking matrix R' by selecting the ix-th row of R, where ix is the index of F_S in the $JointF$.
- Line 3 ranks the contribution of weights to each of the candidate solutions according to R'. The results are stored in matrix R''. R''_{ij} describes the relative order of contribution that a candidate solution $\mathbf{s}_i$ received from a weight vector $\mathbf{w}_j$. The lower the rank, the higher the contribution.

```
Function selectW($JointW, F\_S, R, ix$)
Input: the objective values $F\_S$ of the survived candidate solutions, the joint weight
       vectors, $JointW$, the ranking matrix, $R$, the index list of $F\_S$ in $JointF$, $ix$
Output: weight set, $W$
1  $W \leftarrow \emptyset$;
2  construct the ranking matrix $R'$ by extracting all the $ix$ rows of $R$;
3  rank weights for each $\mathbf{s}_i$ according to $R'$ and obtain another matrix $R''$;
4  foreach $\mathbf{F\_s}_i \in F\_S$ do
5      $J \leftarrow \{j | j = \arg\min_{j \in \{1,2,\cdots,N\}} R''_{ij}\}$ ;
6      if $|J| > 1$ then
7          $k \leftarrow \arg\max_{j \in J}$ angle$(\mathbf{w}_j, \mathbf{F\_s}_i)$ ;
8      else
9          $k \leftarrow j$;
10     end
11     $W \leftarrow W \cup \mathbf{w}_k$;
12     set the $k$-th column of $R''$ as inf;
13 end
```

- Line 5 finds all the weight vectors that give the highest contribution to solution $\mathbf{s}_i$, i.e., weights that have $R''_{ij} = 1$. If there is only one weight then we choose this weight for $\mathbf{s}_i$. If multiple weights are found, then we choose the weight $\mathbf{w}_k$ that has the largest angle with $\mathbf{F\_s}_i$ (lines 6 to 11). To avoid multiple selections of a weight, once the weight $\mathbf{w}_k$ is selected, the k-th column of R''_{ij} is set as inf.
- Additionally, function `angle` computes the angle between a candidate solution and a weight vector. The dot product of two normalised vectors returns the cosine of the angle between the two vectors.

To further explain the co-evolution procedure, let us consider a bi-objective minimisation instance shown in Figure 4.12 with four candidate solutions and four weight vectors, i.e. $N = N_w = 2$. In this example, it is assumed that $\mathbf{w}_1$ has two neighbours $\mathbf{s}_1$ and $\mathbf{s}_2$; $\mathbf{w}_2$ has only one neighbour $\mathbf{s}_3$; $\mathbf{w}_3$ has two neighbours $\mathbf{s}_3$ and $\mathbf{s}_4$; $\mathbf{w}_4$ has no neighbour.

Table 4.1 shows the selection process of candidate solutions. First, candidate solutions are ranked by each of the weighted Chebyshev functions. Then the fitness of each candidate solution is calculated according to the procedure of function `selectS`. Based on the fitness, $\mathbf{s}_1$ ($Fit_{\mathbf{s}_1} = 2$) and $\mathbf{s}_3$ ($Fit_{\mathbf{s}_3} = 3$) are selected as new parent candidate solutions S.

Next we select the best weight for each candidate solution in the S. First, we randomly select one solution from S, e.g. $\mathbf{s}_1$. Then we identify the weights that contribute the most to $\mathbf{s}_1$, that is, $\mathbf{s}_1$ is ranked the best on these weights. There is only one weight i.e., $\mathbf{w}_1$, that contributes to $\mathbf{s}_1$ and therefore $\mathbf{w}_1$ is selected for $\mathbf{s}_1$. After this, another candidate solution is randomly selected from the set $S \backslash \mathbf{s}_1$, e.g., $\mathbf{s}_3$. Similarly, we find

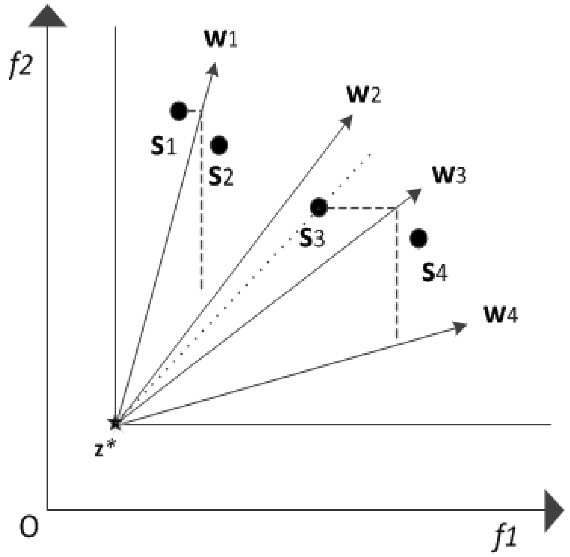


FIGURE 4.12: Illustration of the co-evolution procedure of PICEA-w with a bi-objective minimisation example.

TABLE 4.1: Candidate solutions selection process.

(a) Ranking matrix R

	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$
$\mathbf{s}_1$	1	inf	inf	inf
$\mathbf{s}_2$	2	inf	inf	inf
$\mathbf{s}_3$	inf	1	1	inf
$\mathbf{s}_4$	inf	inf	2	inf

(b) Ranking matrix R^1, r and $Fit_{\mathbf{s}_i}$

	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	r	$Fit_{\mathbf{s}_i}$
$\mathbf{s}_1$	1	inf	inf	inf	2	**2**
$\mathbf{s}_2$	2	inf	inf	inf	3	**1**
$\mathbf{s}_3$	1	1	inf	inf	1	**3**
$\mathbf{s}_4$	2	inf	inf	inf	3	**1**

TABLE 4.2: Weights selection process.

(a) Ranking matrix R'

	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$
$\mathbf{s}_1$	1	inf	inf	inf
$\mathbf{s}_3$	inf	1	1	inf

(b) Ranking matrix R''

	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	
$\mathbf{s}_1$	1	2	2	2	$\mathbf{w}_1$
$\mathbf{s}_3$	2	1	1	2	$\mathbf{w}_3$

the weights on which $\mathbf{s}_3$ performs the best. Both $\mathbf{w}_2$ and $\mathbf{w}_3$ satisfy the condition. Then we look at the second criterion. It is found that the angle between $\mathbf{s}_3$ and $\mathbf{w}_3$ is larger than that between $\mathbf{s}_3$ and $\mathbf{w}_2$ and so $\mathbf{w}_3$ is selected for $\mathbf{s}_3$. This procedure continues until each candidate solution in the S is assigned a weight vector.

Additionally, in PICEA-w the number of weight vectors N_w is not required to be equal to the number of candidate solutions N. However, since in each generation each of the survived candidate solution is assigned a different weight vector, it is required that $2N_w \geq N$.

4.4.2 Algorithm analysis: PICEA-w

PICEA-w co-evolves candidate solutions with weight vectors during the search. The candidate solutions are evaluated by the employed weights in a similar way to other decomposition based algorithms (e.g., MOGLS, MSOPS and MOEA/D). However, the weights are neither randomly generated nor initialised as an even distribution but are adaptively modified in a co-evolutionary manner during the search. It is expected PICEA-w would be less sensitive to problem geometry and perform better than other decomposition based algorithms that use random, evenly distributed or adaptive weights on many-objective problems.

The convergence of PICEA-w is affected by the chosen scalarising function (as described in Section 4.2.1). We apply the Chebyshev scalarising function to PICEA-w. Compared with other L_p scalarising functions, the Chebyshev scalarising function, though it leads to a slower convergence speed, is able to identify Pareto optimal solutions in both convex and non-convex regions.

The diversity performance of PICEA-w is affected by the distribution of the employed weights. Co-evolution enables suitable weights to be constructed adaptively and thus distributing the search effort appropriately towards different parts of the Pareto front. The two criteria used for the selection of weights effectively balances exploration and exploitation.

With respect to the time complexity of PICEA-w, evaluation of a population of candidate solutions runs at $\mathcal{O}(M \times N)$, where M is the number of objectives and N is the number of candidate solutions. The main cost of PICEA-w concerns function `coEvolve`, in which three sub-functions are involved. The sub-function `rankingSW` ranks all candidate solutions on each weight vector and so runs at $\mathcal{O}(N^2 \times N_w)$ (assuming that bubble sorting is used). The sub-function `selectS` selects the best N solutions from $2N$ solutions which runs at $\mathcal{O}(N^2)$. The sub-function `selectW` calculates the angle between each pair of candidate solution and weight vector, and runs at $\mathcal{O}(N \times N_w)$. Therefore, the overall time complexity of PICEA-w is $\mathcal{O}(N^2 \times N_w)$.

4.5 Experiment description

4.5.1 Test problems

Table 4.3 shows the employed test problems. These test problems are constructed by applying different shape functions provided in the WFG toolkit to the standard WFG4 benchmark problem (Huband et al., 2006). Details are provided in Appendix A.3.

Table 4.3: Problem geometries of the WFG4X test problems.

Test problem	Geometry
WFG41	concave
WFG42	convex
WFG43	strong concave
WFG44	strong convex
WFG45	mixed
WFG46	hyperplane
WFG47	disconnected, concave
WFG48	disconnected convex

These problems are invoked in 2-, 4-, 7- and 10- objective instances. The WFG parameters k (position parameter) and l (distance parameter) are set to 18 and 14, i.e., the number of decision variables is $n = k + l = 32$ for each problem instance.

Optimal solutions of these problems satisfy the condition in Equation 4.9:

$$x_{i=k+l:n} = 2i \times 0.35 \tag{4.9}$$

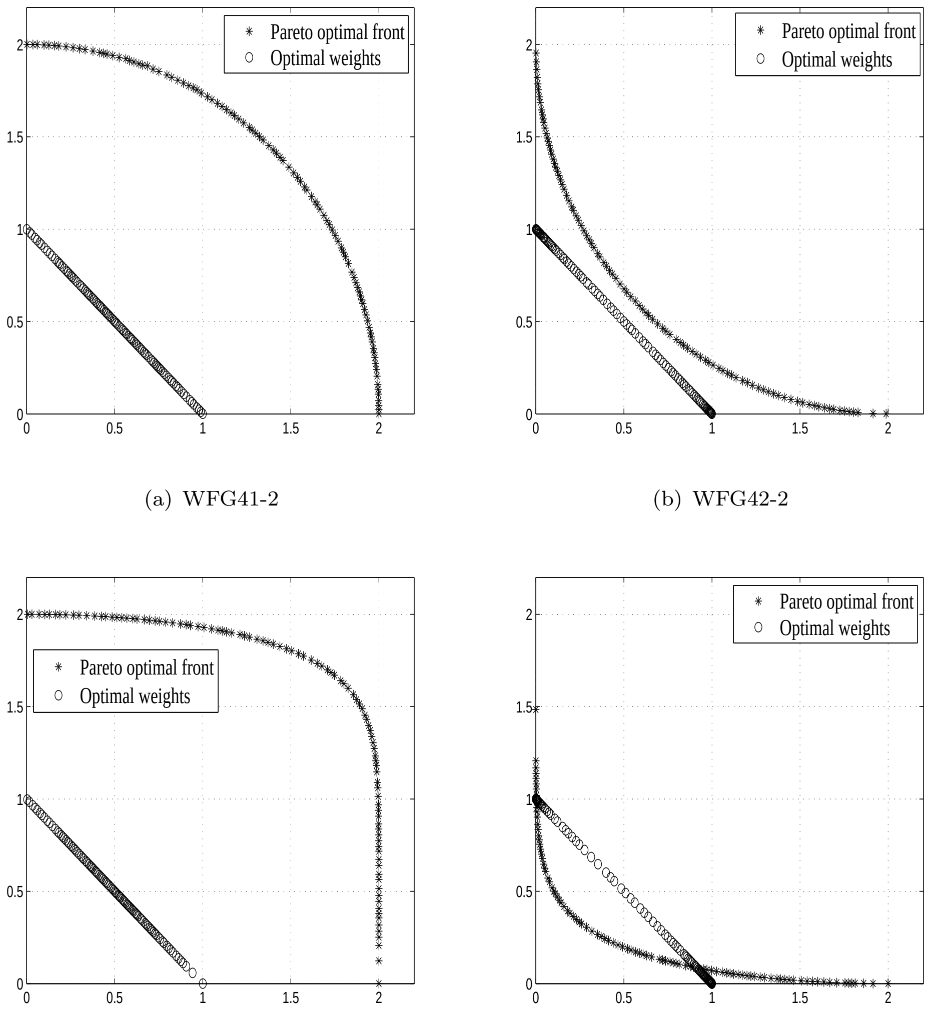
where n is the number of decision variables and $n = k + l$. To obtain an approximation of the Pareto optimal front, we first randomly generate $20,000$ optimal solutions for the test problem and compute their objective values. Second, we employ the clustering technique employed in SPEA2 to select a set of evenly distributed solutions from all the generated solutions.

The Pareto optimal front of the WFG4X problem has the same trade-off magnitudes, and it is within $[0, 2]$. Thus, the *nadir* point for these problems is $[2, 2, \cdots, 2]$. Figures 4.13 and 4.14 show the Pareto optimal front as well as the corresponding optimal distribution of weights for all the 2-objective WFG problems. We also plot the Pareto optimal fronts of the 3-objective WFG4X problems in Appendix A.3.

4.5.2 The considered competitor MOEAs

To benchmark the performance of PICEA-w, four competitor decomposition based algorithms are considered. All the competitors use the same algorithmic framework as PICEA-w, and the Chebyshev scalarising function is chosen. The only difference lies in the way of constructing $JointW$.

- The first algorithm (denoted as RMOEA) forms $JointW$ by combining N_w weights that are randomly selected from current $JointW$ and another set of N_w randomly generated weights. RMOEA represents decomposition based algorithms using random weights, e.g., I-MOGLS (Ishibuchi and Murata, 1998) and J-MOGLS (Jaszkiewicz, 2002).



(a) WFG41-2

(b) WFG42-2

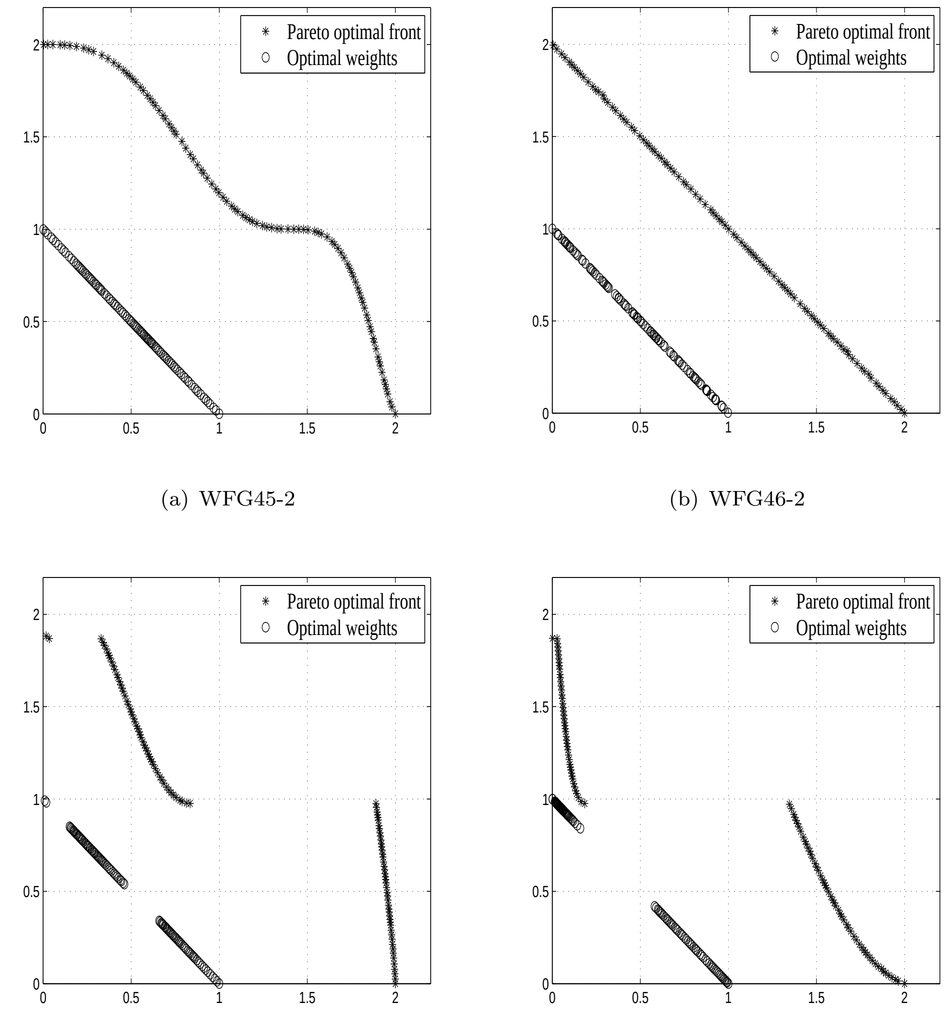
(c) WFG43-2

(d) WFG44-2

Figure 4.13: Pareto optimal fronts and the optimal distributions of weights for WFG41-2 to WFG44-2.

- The second competitor MOEA applies $2N_w$ evenly distributed weights as $JointW$ (denoted as UMOEA). UMOEA represents another class of decomposition based algorithms, such as MSOPS and MOEA/D, that use uniform weights.

- The other two considered competitors use adaptive weights. The weights adaptation strategies are extracted from DMOEA/D (Gu et al., 2012) and EMOSA (Li and Landa-Silva, 2011), respectively. The reason for choosing these two algorithms is that DMOEA/D has been shown to be able to obtain evenly distributed solutions for bi- and three-objective problems having complex geometries, and EMOSA is found to outperform MOGLS and MOEA/D on bi- and three-objective problems. Note that the neighbourhood size used in DMOEA/D and EMOSA is set as $T = 10$ which is demonstrated to offer a good performance (according to our previous comparative study (Wang et al., 2013)).



(a) WFG45-2

(b) WFG46-2

(c) WFG47-2

(d) WFG48-2

FIGURE 4.14: Pareto optimal fronts and the optimal distributions of weights for WFG45-2 to WFG48-2.

The weights adaptation strategies in DMOEA/D and EMOSA are briefly described as follows (for the readers' convenience), more details can be found in Chapter 2.4.2 p.40.

(i) DMOEA/D: first a piecewise linear interpolation method is used to fit a curve ($M = 2$) or hyper-surface ($M > 2$) for the current non-dominated solutions. Second, sample a set of evenly distributed points from the curve (or the hyper-surface). After that an optimal distribution of weights, corresponding to the evenly distributed points, is generated to guide the search.

(ii) EMOSA: for each member $\mathbf{F\_s}_i$ in the current population, first, find the closest neighbour (e.g., $\mathbf{F\_s}_j$) to $\mathbf{F\_s}_i$ and its associated weight vector $\mathbf{w}_j$. Second, identify the weights in $JointW$ whose Euclidean distance to $\mathbf{w}_j$ is larger than the distance between $\mathbf{w}_i$ and $\mathbf{w}_j$. Third, amongst the identified weights, select the weights for which the distance between them and $\mathbf{w}_i$ are closer than the distance between them and all the neighbours of $\mathbf{w}_i$. The definition of the neighbourhood is the same as MOEA/D. If there are multiple weights, then pick one randomly.

4.5.3 General parameters

Each algorithm is performed for 31 runs, each run for 25 000 function evaluations. For all algorithms the population size of candidate solutions and weights are set as $N = 100$ and $N_w = 100$, respectively. Simulated binary crossover (SBX) and polynomial mutation (PM) are applied as genetic operators. The recombination probability p_c of SBX is set to 1 per individual and mutation probability p_m of PM is set to $1/n$ per decision variable. The distribution indices η_c of SBX and η_m of PM are set as 15 and 20, respectively. These parameter settings are summarised in Table 4.4 and are fixed across all algorithm runs.

Table 4.4: Algorithm testing parameter setting.

Parameters	Values
N	100
N_w	100
$maxGen$	250
Crossover operator	SBX ($p_c = 1, \eta_c = 15$)
Mutation operator	PM ($p_m = \frac{1}{n}, \eta_m = 20$)

4.5.4 Performance assessment

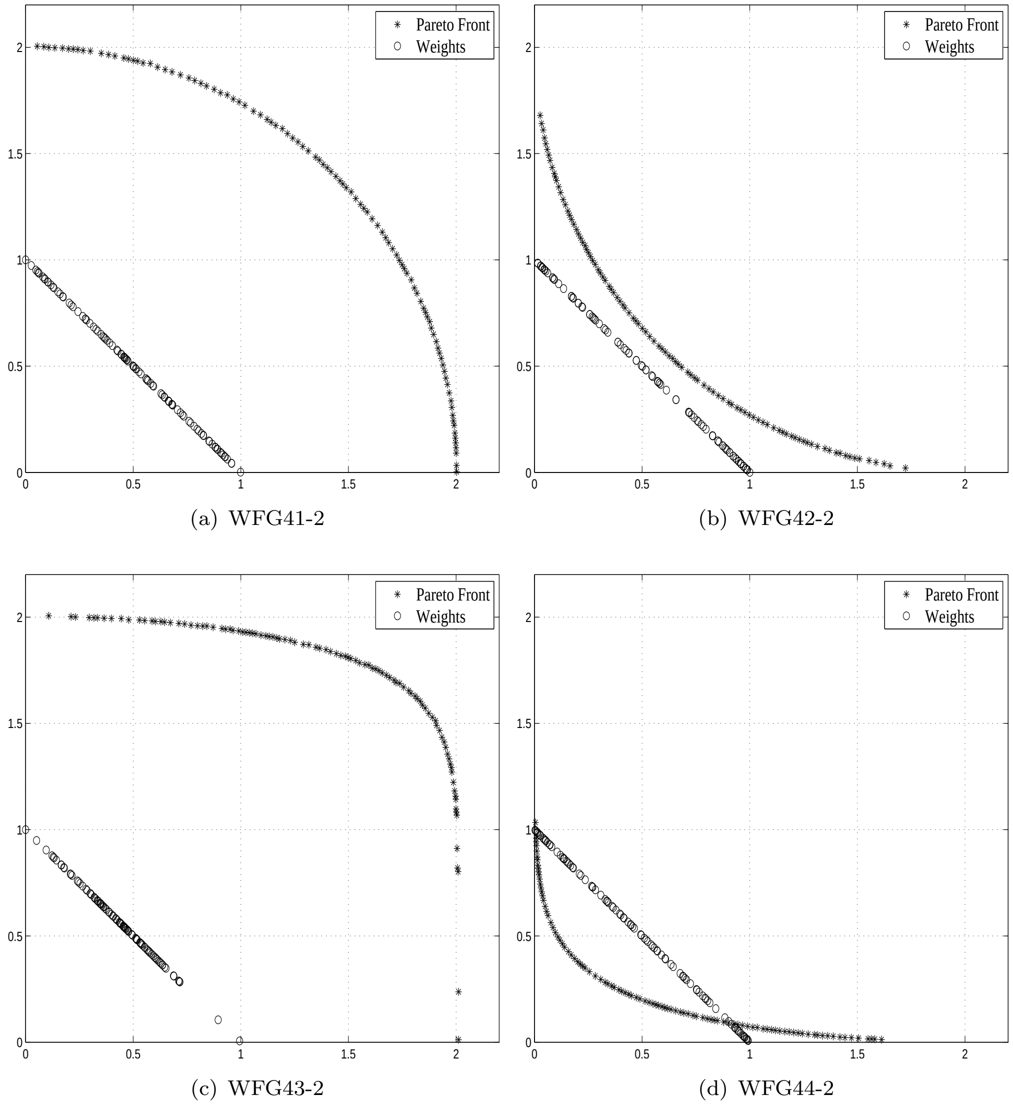
The hypervolume metric (HV) is used as a performance metric. A favourable hypervolume (larger, for a minimisation problem) implies a better combination of proximity and diversity. The approximation sets used in the HV calculation are the members of the offline archive of all non-dominated points found during the search, since this is the set most relevant to a *posteriori* decision-making. For reasons of computational feasibility, prior to analysis the set is pruned to a maximum size of 100 using the SPEA2 truncation procedure (Zitzler et al., 2002). Note that prior to calculating the HV, we normalize all objective values to be within the range $[0, 1]$ using the *nadir* point (which assumes equal relative importance of normalised objectives across the search domain). The reference point for the hypervolume calculation is set as $r_i = 1.2, \quad i = 1, 2, \cdots, M$.

Performance comparisons between algorithms based on the HV metric are made according to a rigorous non-parametric statistical framework, drawing on recommendations in Zitzler et al. (2003). Specifically, we first test the hypothesis that all algorithms perform equally using the Kruskal-Wallis test (Hollander and Wolfe, 1999). If this hypothesis is rejected at the 95% confidence level, we then consider pair-wise comparisons between the algorithms using the Wilcoxon-ranksum two-sided comparison procedure (Hollander and Wolfe, 1999) at the 95% confidence level, employing the Šidák correction to reduce Type I errors (Curtin and Schulz, 1998).

4.6 Experiment results

First, the Pareto fronts obtained by PICEA-w are visually plotted. Then, the statistical comparison results of PICEA-w and other competitor decomposition based algorithms are presented.

4.6.1 Non-dominated Pareto fronts and the co-evolved weights



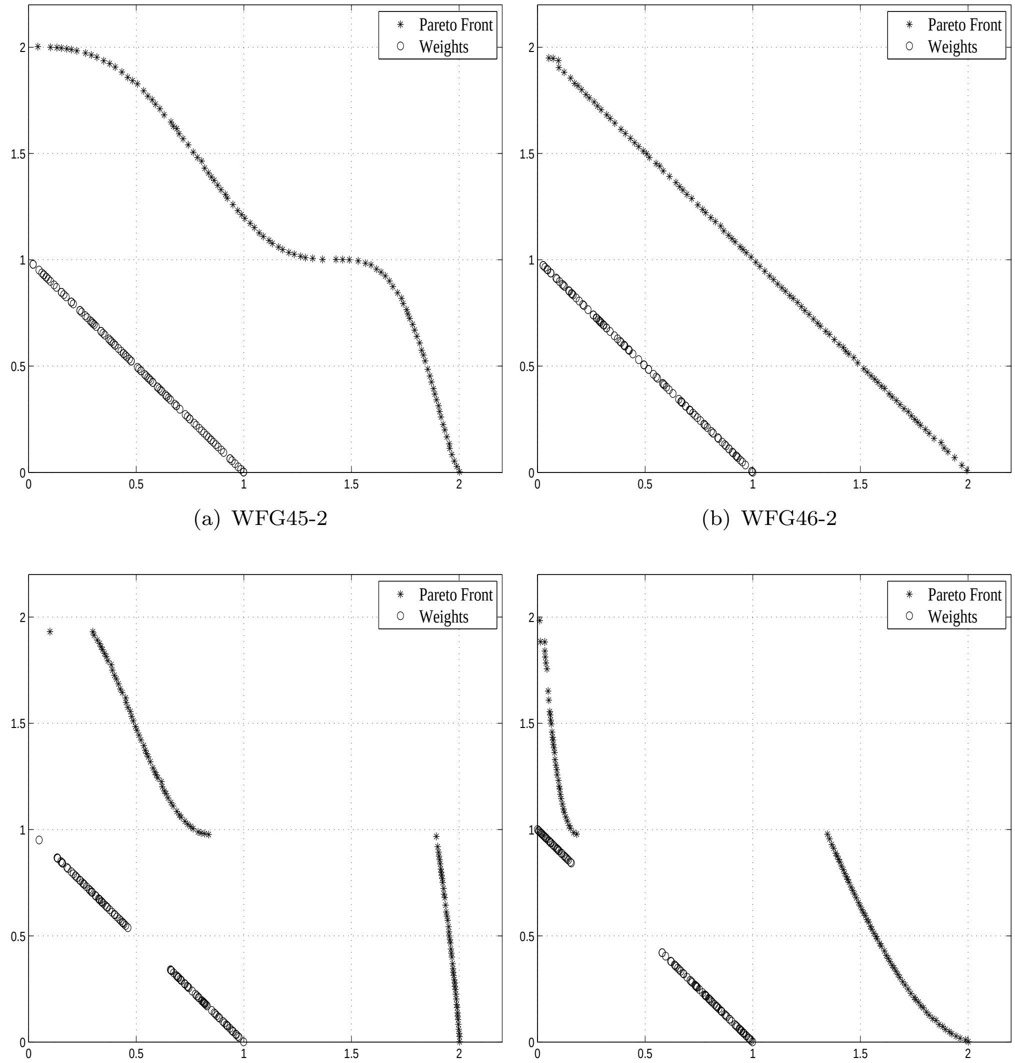
(a) WFG41-2

(b) WFG42-2

(c) WFG43-2

(d) WFG44-2

FIGURE 4.15: Pareto fronts and weights obtained by PICEA-w for WFG41-2 to WFG44-2.



(a) WFG45-2

(b) WFG46-2

(c) WFG47-2

(d) WFG48-2

Figure 4.16: Pareto fronts and weights obtained by PICEA-w for WFG45-2 to WFG48-2.

To visualise the performance of PICEA-w, we plot the obtained Pareto front (that has the median HV metric value across the 31 runs) as well as the distribution of the co-evolved weights of the 2-objective WFG4X problems in Figures 4.15 and 4.16.

From the results, it is observed that, for most of problems, the obtained solutions spread evenly along the Pareto front, that is, the performance of PICEA-w is robust with respect to the problem geometry. In addition, compared with Figures 4.13 and 4.14, the distribution of the obtained weights also approximates to the optimal distribution for each problem (this will be further discussed in Section 4.7.3). For example, problem WFG43-2 features strong concavity, and then the distribution of the obtained weights is dense in the centre while sparse at the edge. WFG44-2 features strong convexity, and thus the distribution of the co-evolved weights is sparse in the centre while dense at the edge. A much clearer observation can be made on problems WFG47-2 and WFG48-2.

On these two problems the co-evolution leads most of the weights to be constructed in the *right* place, intersecting the Pareto optimal front.

4.6.2 Statistical results

Results of the Kruskal-Wallis tests followed by pair-wise Wilcoxon-ranksum plus Šidák correction tests based on the performance metric are provided in this Section. The initial Kruskal-Wallis test breaks the hypothesis that all five algorithms are equivalent. Therefore the outcomes of pair-wise statistical comparisons for 2-, 4-, 7- and 10-objective WFG problems are shown in Tables 4.5, 4.6 4.7 and 4.8 respectively. A smaller rank value indicates better performance; ordering within a rank is purely alphabetical.

HV results for 2-objective problems

Table 4.5: *HV* results for 2-objective instances

WFG		Ranking by Performance	WFG		Ranking by Performance
41	1	UMOEA	42	1	UMOEA
	2	DMOEA/D EMOSA PICEA-w		2	PICEA-w EMOSA
				3	DMOEA/D
	3	RMOEA		4	RMOEA
43	1	DMOEA/D EMOSA PICEA-w	44	1	DMOEA/D EMOSA PICEA-w
	2	UMOEA		2	UMOEA
	3	RMOEA		3	RMOEA
45	1	DMOEA/D EMOSA PICEA-w UMOEA	46	1	UMOEA
				2	DMOEA/D EMOSA PICEA-w
	2	RMOEA			
				3	RMOEA
47	1	PICEA-w UMOEA	48	1	PICEA-w
	2	DMOEA/D EMOSA		2	UMOEA
	3	RMOEA		3	DMOEA/D EMOSA
				4	RMOEA

For 2-objective problems, it is observed from Table 4.5 that

(i) UMOEA performs the best for WFG41, WFG42, WFG45 and WFG46 for which the Pareto optimal fronts are neutral. However, UMOEA exhibits an inferior performance on the rest of the four problems. Specifically, its performance is worse than PICEA-w on three out of the four problems, and is worse than DMOEA/D and EMOSA on WFG43 and WFG44.

(ii) The three decomposition based algorithms that use adaptive weights have comparable performance on five problems: WFG41, WFG43, WFG44, WFG45 and

WFG46. On WFG47 and WFG48, DMOEA/D and EMOSA have comparable performance and both are worse than PICEA-w. On WFG42, PICEA-w and EMOSA have comparable performance and they both perform better than DMOEA/D.

(iii) RMOEA is the worst optimiser for all of the problems.

HV results for 4-objective problems

Table 4.6: *HV* results for 4-objective instances.

WFG		Ranking by Performance	WFG		Ranking by Performance
41	1	PICEA-w	42	1	PICEA-w EMOSA
	2	DMOEA/D EMOSA		2	DMOEA/D
	3	RMOEA UMOEA		3	RMOEA UMOEA
43	1	EMOSA PICEA-w	44	1	PICEA-w EMOSA
	2	DMOEA/D		2	DMOEA/D
	3	RMOEA		3	RMOEA
	4	UMOEA		4	UMOEA
45	1	DMOEA/D EMOSA PICEA-w	46	1	PICEA-w
				2	DMOEA/D EMOSA
	2	RMOEA		3	RMOEA UMOEA
	3	UMOEA			
47	1	PICEA-w	48	1	PICEA-w
	2	DMOEA/D EMOSA		2	DMOEA/D EMOSA
	3	RMOEA		3	RMOEA
	4	UMOEA		4	UMOEA

For 4-objective problems, it is observed from Table 4.6 that:

(i) The performance of UMOEA degrades. It is inferior to the three adaptive weights based algorithms for all the problems. It even performs worse than RMOEA for five out of the eight problems, and on the other three problems (WFG41, WFG42 and WFG46) UMOEA performs comparably with RMOEA.

(ii) PICEA-w is always amongst the top performing algorithms. It is exclusively the best for four problems, i.e., WFG41, WFG46, WFG47 and WFG48.

(iii) With respect to EMOSA and DMOEA/D, it is found that EMOSA exhibits a better or comparable performance for all of the problems.

(iv) RMOEA performs better than UMOEA, but is worse than the three adaptive weights based algorithms.

HV results for 7-objective problems

As the number of objectives increases to 7, we can observe from Table 4.7 that the performance of PICEA-w becomes more promising:

TABLE 4.7: *HV* results for 7-objective instances.

WFG		Ranking by Performance	WFG		Ranking by Performance
41	1	PICEA-w	42	1	PICEA-w
	2	DMOEA/D EMOSA		2	DMOEA/D EMOSA
	3	RMOEA		3	RMOEA
	4	UMOEA		4	UMOEA
43	1	PICEA-w	44	1	PICEA-w
	2	EMOSA		2	DMOEA/D EMOSA
	3	DMOEA/D		3	RMOEA
	4	RMOEA		4	UMOEA
	5	UMOEA			
45	1	EMOSA PICEA-w	46	1	EMOSA PICEA-w
	2	DMOEA/D RMOEA		2	DMOEA/D
	3	UMOEA		3	RMOEA
				4	UMOEA
47	1	PICEA-w	48	1	PICEA-w
	2	EMOSA		2	EMOSA
	3	DMOEA/D RMOEA		3	DMOEA/D RMOEA
	4	UMOEA		4	UMOEA

(i) PICEA-w ranks exclusively or jointly (on WFG45 and WFG46) the best for all of the problems.

(ii) Among the remaining four algorithms, EMOSA is the most effective one. It performs better than DMOEA/D for five out of the eight problems, and is comparable to DMOEA/D for the remaining three problems.

(iii) Although DMOEA/D is worse than PICEA-w and EMOSA, it is better than RMOEA for most of the problems.

(iv) RMOEA performs better than UMOEA for all the problems. UMOEA performs the least well.

HV results for 10-objective problems

Similar results to the 7-objective problems are found for the 10-objective problems. Key observations are as follows: PICEA-w consistently performs the best for all problems, followed by EMOSA and then DMOEA/D. With respect to RMOEA and UMOEA, RMOEA is better for all problems.

Supplementary convergence and diversity results

To further investigate the performance of the algorithms, we also separately calculated the proximity (as measured by generational distance – GD) and diversity (as measured by the spread metric – Δ) measures for the 2-, 4-, 7- and 10-objective WFG41 problems.

Table 4.8: *HV* results for 10-objective instances

WFG		Ranking by Performance	WFG		Ranking by Performance
41	1	PICEA-w	42	1	PICEA-w
	2	DMOEA/D EMOSA		2	EMOSA
	3	RMOEA		2	DMOEA/D
	4	UMOEA		3	RMOEA
				5	UMOEA
43	1	PICEA-w	44	1	PICEA-w
	2	EMOSA		2	EMOSA
	3	DMOEA/D		3	DMOEA/D
	4	RMOEA		4	RMOEA
	5	UMOEA		5	UMOEA
45	1	EMOSA PICEA-w	46	1	PICEA-w
	2	DMOEA/D RMOEA		2	DMOEA/D EMOSA
	3	UMOEA		3	RMOEA
				4	UMOEA
8	1	PICEA-w	9	1	PICEA-w EMOSA
	2	EMOSA DMOEA/D		2	DMOEA/D
	3	RMOEA		3	RMOEA
	4	UMOEA		4	UMOEA

The Pareto optimal front of WFG41 is the surface of an M-dimension hyper-sphere with radius $r = 2$ in the first quadrant, which is amenable to uniform sampling. We sample 20,000 points as the reference set for calculating the performance metrics.

Table 4.9: *GD* and Δ results for WFG41 problems

WFG		Ranking by *GD*		WFG		Ranking by Δ
41-2	1	UMOEA		41-2	1	DMOEA/D EMOSA PICEA-w
	2	DMOEA/D PICEA-w				
	3	EMOSA			2	UMOEA
	4	RMOEA			3	RMOEA
41-4	1	UMOEA		41-4	1	EMOSA PICEA-w
	2	PICEA-w			2	DMOEA/D
	3	DMOEA/D EMOSA			3	RMOEA
	4	RMOEA			4	UMOEA
41-7	1	UMOEA		41-7	1	PICEA-w
	2	PICEA-w			2	DMOEA/D EMOSA
	3	DMOEA/D EMOSA			3	RMOEA
	4	RMOEA			4	UMOEA
41-10	1	UMOEA		41-10	1	PICEA-w
	2	PICEA-w			2	EMOSA
	3	DMOEA/D EMOSA			3	DMOEA/D RMOEA
	4	RMOEA			4	UMOEA

From Table 4.9, it is found that

- in terms of the convergence performance, UMOEA performs the best while RMOEA performs the worst for all four problems. Among the three adaptive weights based algorithms, PICEA-w outperforms DMOEA/D and EMOSA for WFWG41-4, WFG41-7 and WFG41-10. For WFG41-2 DMOEA/D is comparable to PICEA-w, and both algorithms are better than EMOSA.

- in terms of the diversity performance PICEA-w is amongst the top performing algorithms for all the four problems. EMOSA performs competitively with PICEA-w on WFG41-2 and WFG41-4, however, it is worse than PICEA-w on the other two problems. DMOEA/D is worse than EMOSA and PICEA-w on the three many-objective problems. However, it is better than RMOEA on all problems except for WFG41-10 where DMOEA/D is comparable to RMOEA. Comparing UMOEA with RMOEA, RMOEA is worse on WFG41-2 while better on the other three many-objective problems.

4.7 Discussion

4.7.1 General discussion

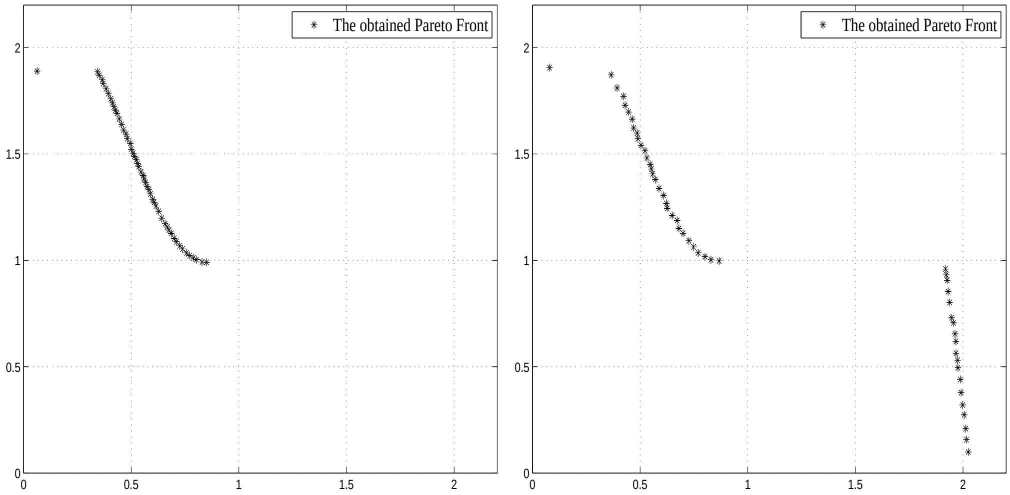
UMOEA, which employs a set of pre-defined evenly distributed weights, faces difficulties on many-objective problems or problems having extremely complex geometry. The reason is that, as mentioned earlier, evenly distributed weights lead to a poor distribution of solutions for problems whose geometry is not similar to a hyper-plane. Simultaneously, as the employed weights are fixed, UMOEA can only obtain a limited number of solutions. These are are insufficient to approximate the entire Pareto optimal front, in particular, for many-objective problems. Certainly, when the problem is low-dimension and has a neutral (i.e., not extremely complex) Pareto optimal front, UMOEA performs well, e.g., the 2-objective WFG41, WFG42, WFG45 and WFG46.

RMOEA, which employs weights that are randomly generated in each generation, performs worse than UMOEA for low-dimension problems. The reason is that, the search directions in RMOEA are not fixed as in UMOEA but keep changing during the whole search process. This leads RMOEA to have an inferior convergence performance compared with UMOEA. However, RMOEA tends to perform better than UMOEA for high-dimension problems. This is also due to the use of random weights, that is, random weights guide the algorithm to search different regions of the Pareto optimal front, therefore lead to a set of diversified solutions, i.e., a better diversity performance.

The three adaptive weights based decomposition based algorithms perform better than UMOEA and RMOEA for the four 2-objective problems (WFG43, WFG44, WFG47 and WFG48) that have complex Pareto optimal fronts and most of the many-objective problems. This indicates that the weights adaptation strategies employed in these algorithms are helpful to handle the issue of problem geometry and many-objective optimisation. Given a closer examination, it is found that amongst the three algorithms, the co-evolution based weights adaptation is the most effective one. It appropriately balances exploration and exploitation and so enables PICEA-w to perform well on most problems. Moreover, during co-evolution, weights gradually learn the geometry of the problem and evolve towards the optimal distribution. This leads PICEA-w to be less sensitive to the problem geometry.

4.7.2 The effect of the angle θ

On the evaluation of candidate solutions in PICEA-w, a parameter θ, which measures the angle between a candidate solution and a weight vector, is employed to implement local selection. As mentioned earlier, the use of θ, i.e., local selection, has a positive effect on improving the diversity performance of PICEA-w, and obtaining the entire Pareto optimal front (Wang et al., 2014). Next, we demonstrate the effectiveness of this strategy.



(a) PICEA-w1: WFG47-2

(b) PICEA-w2: WFG47-2

FIGURE 4.17: The Pareto front obtained by PICEA-w1 and PICEA-w2 for WFG47-2.

The 2- and 4-objective WFG47 are selected as test problems. Two different settings of θ are examined. The first setting configures θ as $\frac{\pi}{2}$ radians during the whole search process, which indicates that the evaluation of candidate solutions is performed globally. That is, every solution competes against all the other solutions. The second strategy configures θ as $\frac{\pi}{18}$ radians during the whole search process, which indicates that the evaluation of candidate solutions is always executed locally. That is, every solution competes against its neighbours. All the other parameters are the same as those adopted in Table 4.4. We use PICEA-w1 and PICEA-w2 to denote PICEA-w using $\theta = \frac{\pi}{2}$ and $\theta = \frac{\pi}{18}$ radians, respectively. These two variants are compared with PICEA-w in which θ is adjusted by Equation 4.8, that is, θ linearly increases from zero to $\frac{\pi}{2}$ radians. Experimental results in terms of the HV metric are shown in Table 4.10. The symbol '$-$', '$=$' or '$+$' means that the considered algorithm performs statistically worse, equal or better than PICEA-w at 95% confidence level.

TABLE 4.10: *HV* comparison results (mean±std) for WFG47 and WFG48.

	PICEA-w1	PICEA-w2	PICEA-w
WFG47-2	0.5028±0.0131 (−)	0.5730±0.0158 (=)	0.5910±0.0109
WFG47-4	0.6346±0.1212 (−)	0.7615±0.0219 (−)	0.7961±0.0182

From the results we can observe that (i) PICEA-w1 performs worse than PICEA-w for all problems; and (ii) PICEA-w2 performs comparably with PICEA-w on the 2-objective WFG47 while performing worse than PICEA-w on WFG47-4.

The poor performance of PICEA-w1 in terms of the *HV* metric is caused by the fact that PICEA-w1 has not found the entire Pareto optimal front, see Figure 4.17. The reason could be that each solution globally competes with other solutions, which leads to some potentially useful, though dominated, solutions to be removed at the early stage of the evolution. For example, see Figure 4.18, although solutions in region A are dominated, they are helpful to search for Pareto optimal solutions in region B. Using a small θ initially is helpful to keep those locally good solutions in region A and therefore obtaining the entire Pareto optimal front.

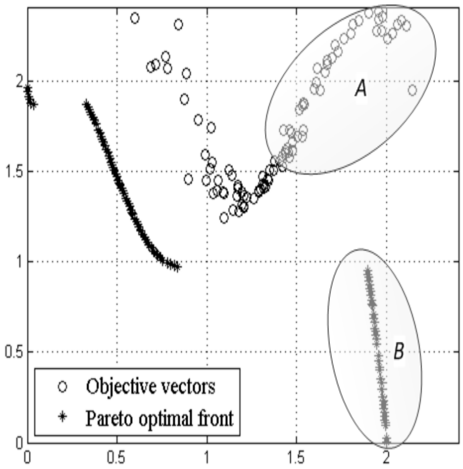


FIGURE 4.18: The initial objective vectors and the Pareto optimal front for WFG47-2.

The reason for the poor performance of PICEA-w2 is likely to be that local selection has a deleterious effect on convergence. This operation might assign higher fitness to some dominated solutions. Although this is helpful from a diversity perspective, it slows down convergence speed as some dominated solutions are stored during the search. This side effect becomes more significant on higher-dimension problems since convergence is inherently more difficult to achieve on many-objective fitness landscapes. Equation 4.8, used to adjust θ, though simple, is an effective strategy to balance convergence and diversity performance. In future, some other strategies will be investigated in order to provide a more effective strategy for configuring θ.

4.7.3 The distribution of the co-evolved weights

As previously mentioned, during the search weights are also evolved towards the optimal distribution. Comparing Figure 4.13 and Figure 4.14 with Figure 4.15 and Figure 4.16, we can observe that for most of the problems the distribution of the co-evolved weights approximates to an optimal distribution. Furthermore, to make a statistical analysis, the two-sample multi-dimensional Kolmogorov-Smirnov test at 95% confidence level (Justel et al., 1997) is employed to test the null hypothesis that the co-evolved weights and the optimal weights are drawn from the same distribution. Comparative results are presented in Table 4.11. $p \leq 0.05$ means that the hypothesis is rejected, that is, the tested two samples are not from the same distribution. Alternatively, $p \geq 0.05$ means that the hypothesis cannot be rejected.

To generate a set of optimal weights, first, N_w solutions from the Pareto optimal front are randomly sampled. The selected solutions are then scaled and normalised to create a valid weight. This procedure is repeated 31 times to get 31 sets of optimal weights. The co-evolved weight vectors from each of the 31 runs are compared with the 31 sets of the optimal weights, separately.

Table 4.11: Statistical comparison results of the distribution of optimal weights and the co-evolved weights for 2-objective problems.

	WFG41	WFG42	WFG43	WFG44
p	0.1231	0.1011	0.2035	0.1208
	WFG45	WFG46	WFG47	WFG48
p	0.1253	0.1217	0.1041	0.2868

Table 4.11 shows that none of the null hypotheses is rejected. This indicates that the co-evolved weights and the optimal weights are drawn from the same distribution. This could also serve as a reason for the good performance of PICEA-w.

4.8 Summary

Decomposition based algorithms comprise a popular class of evolutionary multi-objective optimiser, and have been demonstrated to perform well when a suitable set of weights are provided. However, determining a good set of weights *a priori* for real-world problems is usually not straightforward due to a lack of knowledge of the underlying problem structure. This Chapter proposes a new decomposition based algorithm for multi-objective optimisation, PICEA-w, that eliminates the need to specify appropriate weights in advance of performing the optimisation. Specifically, weights are adaptively modified by being co-evolved with candidate solutions during the search process. The co-evolution enables suitable weights to be adaptively constructed along the optimisation process and thus guiding the candidate solutions towards the Pareto optimal front effectively. Through rigorous empirical testing, we demonstrate the benefits of PICEA-w compared with other leading decomposition-based algorithms. The chosen test problems encompass the range of problem geometries likely to be seen in practice, including simultaneous optimisation of up to ten conflicting objectives. The main findings are as follows:

(1) PICEA-w is less sensitive to the problem geometry, and outperforms other leading decomposition based algorithms on many-objective problems. Moreover, it is shown that when guiding candidate solution towards the Pareto optimal front, weights also evolve towards the optimal distribution.

(2) UMOEA faces difficulties on problems having a complex Pareto optimal front and many-objective problems. The poor performance of UMOEA is due to a lack of solution diversity. However, UMOEA is found to perform the best in terms of convergence. This could be explained by the fact that the employed weights are kept fixed during the whole search process, which guide candidate solutions towards Pareto optimal front directly.

(3) RMOEA also faces difficulties on problems having a complex Pareto optimal front. Noticeably, although it performs worse than UMOEA on bi-objective problems, it performs better than UMOEA on many-objective problems. The reason is that, for bi-objective problems, the employed evenly distributed weights are sufficient to describe the entire Pareto optimal front and therefore the diversity performance of UMOEA is not much inferior to RMOEA. However, UMOEA demonstrates a better convergence performance than RMOEA. For many-objective problems, the limited number of weights employed in UMOEA is not sufficient to approximate the entire Pareto optimal front while the use of random weights enables the search to explore different regions of the Pareto optimal front, producing better solution diversity.

Lastly, although adaptive weights are helpful in dealing with the issues – problem geometry and many-objective optimisation – adapting weights degrades the algorithm's convergence performance. Therefore, it is worth mentioning that if the underlying problem knowledge, such as problem geometry, is known *a priori*, it is better to employ weights that are pre-defined with an optimum distribution.

Chapter 5

Parameter sensitivity study for PICEAs

5.1 Introduction

Chapters 3 and 4 have demonstrated the effectiveness of the PICEA concept. PICEA-g, which co-evolves goal vectors with candidate solutions, outperforms four best-in-class algorithms on many-objective problems, and PICEA-w, which co-evolves weight vectors with candidate solutions, overcomes some difficulties faced by decomposition based algorithms. However, it is unknown whether PICEAs are robust with respect to their chosen parameter settings. Eliminating the need for tuning parameters settings makes PICEAs robust and easy to use, especially for a non-expert user.

This Chapter studies the sensitivities of parameters in PICEAs. A variety of sensitivity analysis approaches are available, ranging from local methods that vary a single parameter at a time, to global methods that investigate the entire parameter space simultaneously. In this study, we first employ a global sensitivity analysis method, the Sobol' variance decomposition approach (Sobol, 1993, 2001; Saltelli et al., 2010), to determine the relative importance of the parameters controlling the performance of PICEAs. Arising from this approach, we investigate the effect of key parameters and the genetic operators. This analysis enables us to gain insight of the robustness of the algorithms and also to provide suggestions for parameter settings.

The rest of this Chapter is structured as follows. Section 5.2 discusses parameter sensitivities in a global manner using the Sobol' variance decomposition approach. Section 5.3 experimentally studies the sensitivities of key parameters. Section 5.4 summarises this Chapter.

5.2 Parametric sensitivity analysis: a global manner

5.2.1 The Sobol' variance decomposition method

A number of different global sensitivity analysis approaches have been proposed (Sheridan, 2008). The Sobol' variance decomposition (Sobol, 1993) is chosen due to its effectiveness compared with other global sensitivity analysis approaches (Tang et al., 2007).

The first-order effect S_i and the total-order effect S_i^T are two widely used measures in the Sobol' variance decomposition method. Considering a model $Y = f(x_1, x_2, \cdots, x_h)$ that has h parameters, S_i measures the expected reduction in variance that can be achieved if one input parameter, x_i, is fixed. Meanwhile, S_i^T measures the remaining expected variance if all parameters except x_i are fixed.

Mathematically, S_i is defined as follows (Hadka and Reed, 2012):

$$S_i = \frac{V_{x_i}[E_{x_{\sim i}}(Y|x_i)]}{V(Y)} \tag{5.1}$$

where x_i is the ith parameter and $x_{\sim i}$ represents a container that contains all the parameters except x_i. $E_{x_{\sim i}}(Y|x_i)$ in the numerator represents the expected value of Y, taken over all possible values of $x_{\sim i}$ when x_i is fixed. $V_{x_i}[E_{x_{\sim i}}(Y|x_i)]$ measures the variance of $E_{x_{\sim i}}(Y|x_i)$, taken over all possible values of x_i.

S_i only measures the effect of a single input parameter to the variance of the output. It has not taken into account the interactions V_{ij} between two input parameters. V_{ij} is defined as follows:

$$V_{ij} = V(E[Y|x_i, x_j]) - V(E[Y|x_i]) - V(E[Y|x_j]) \tag{5.2}$$

where $V(E[Y|x_i, x_j])$ describes the joint effect of the pair of parameters (x_i, x_j) on Y. This is known as the second-order effect. Higher-order effects can be computed similarly, e.g. the variance of the third-order effect of three parameters x_i, x_j and x_k is:

$$V_{ijk} = V(E[Y|x_i, x_j, x_k]) - V_{ij} - V_{jk} - V_{ik} - V_i - V_j - V_k \tag{5.3}$$

S_i^T denotes the sum of the first and higher order effects of x_i. Taking a model with three input parameters ($h = 3$) as an example, the total order effect of x_1 is $S_1^T = S_1 + S_{12} + S_{13} + S_{123}$, where S_{12} and S_{13} are the second-order effect of x_1, S_{123} is the third-order effect of x_1. Obviously, computing the total-order effect using Equation 5.3 is not appropriate when h is large. However, from Equation 5.1, $S_{\sim i}$ can be extended as:

$$S_{\sim i} = \frac{V_{x_{\sim i}}[E_{x_i}(Y|x_{\sim i})]}{V(Y)} \tag{5.4}$$

$S_{\sim i}$ represents the first order effect of $x_{\sim i}$. Therefore, $V(Y) - V_{x_{\sim i}}[E_{x_i}(Y|x_{\sim i})]$ is the contribution of all terms in the variance decomposition that includes x_i (Saltelli et al., 2010). Thus, S_i^T can be computed by:

$$S_i^T = 1 - \frac{V_{x_{\sim i}}[E_{x_i}(Y|x_{\sim i})]}{V(Y)} \tag{5.5}$$

In practice, S_i and S_i^T can be computed by evaluating the model at a number of sample points. These points can be selected by using the Monte Carlo method. Each point corresponds to a model output. In this study, all the Monte Carlo samples of the parameter space are generated using the Sobol' quasi-random sequence sampling method (Sobol and Kucherenko, 2005). S_i and S_i^T are calculated by the approach presented in Saltelli (2002) with $q \times (h+2)$ runs, where h is the number of parameters, and q is the sample size which is set to 2^h as suggested in Saltelli (2002).

To validate that the sensitivity results are caused by the effect of parametrisation rather than the stochastic effects, the bootstrap technique called the moment method is applied to evaluate the confidence level of the Sobol' indices, producing 95% confidence intervals. The reason for choosing the moment method is that it produces reliable results using a small resampling size (Archer et al., 1997). In this study the resampling size B is set to 2 000 as suggested in Saltelli et al. (2010). Assuming that we have determined S_i and S_i^T for each parameter of the model $Y = f(x_1, x_2, \cdots, x_h)$, the Bootstrap Confidence Interval (BCI) for each parameter is computed as follows, taking S_i as an example (BCI for S_i^T is calculated in a similar way):

(i) Resample B groups of the parameters, $(x_1, x_1, \cdots, x_h)$, with which we can then obtain B outputs of Y. S_i is recalculated at each sampling time. This produces a bootstrap estimate of the sampling distribution of the sensitivity indices, S_i^b, where $b = 1, 2, \cdots, B$.

(ii) BCI for S_i with 95% confidence is then determined by $S_i \pm 1.96 \times std(S_i)$, where $std(S_i) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(S_i^b - \frac{1}{B}\sum_{b=1}^{B} S_i^b\right)^2}$.

When the value of S_i or S_i^T exceeds the BCIs of S_i or S_i^T, we say the sensitivity indices cannot be reliably computed. In other words, the variance of S_i or S_i^T is mostly caused by stochastic effects. More precisely, a large confidence level of S_i or S_i^T is due to the fact that the effect of parameterization is not significantly stronger than stochastic effects (Hadka and Reed, 2012).

5.2.2 Experiment description

There are $h = 7$ parameters in both PICEA-g and PICEA-w. These parameters are the number of function evaluations, *NFS*, the population size of candidate solutions, N, the number of preferences, N_p, the crossover rate of SBX, p_c, the mutation rate of PM, p_m, the SBX distribution index, η_c and the PM distribution index, η_m. The sampled parameter space of these parameters is shown in Table 5.1. Parameter N_p is the number of preferences used in the PICEAs. N_p refers to N_g in PICEA-g, and N_w in PICEA-w.

Table 5.1: The sampled parameter space.

	values									
NFS	2500	5000	10000	15000	20000	25000	30000	35000	40000	50000
N	20	50	100	150	200	300	400	500	700	1 000
N_p	20	50	100	150	200	300	400	500	700	1 000
p_c	0.01	0.05	0.1	0.2	0.4	0.5	0.7	0.8	0.9	1
η_c	1	5	10	20	50	100	200	300	400	500
p_m	0.01	0.05	0.1	0.2	0.4	0.5	0.7	0.8	0.9	1
η_m	1	5	10	20	50	100	200	300	400	500

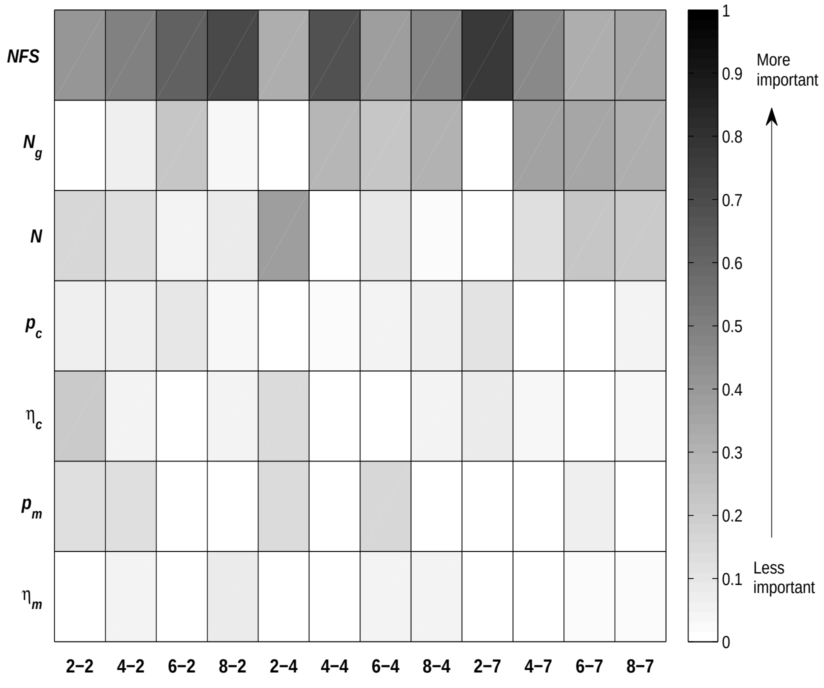
Parameter *NFS* is sampled from 2 500 to 50 000 times, and parameters N and N_p range from 20 to 1000. Such configurations aim to permit reasonable execution times, while providing meaningful results. The configuration of parameters p_c and p_m covers the entire feasible range [0,1]. The distribution indices of SBX (η_c) and PM (η_m) are based on the choice used by Purshouse and Fleming (2007).

As mentioned earlier, the sample size q is suggested to be 2^h, that is, $q = 2^7 = 128$ groups of parameter settings are sampled. For each test problem, the testing algorithm is run $q \times (h + 2) = 128 \times (7 + 2) = 1152$ times, which correspondingly produces 1152 Pareto approximation sets. The *HV* metric is then calculated for each Pareto approximate set. Thus, the final results comprise of 1152 parameter settings and their corresponding end-of-run *HV* values. Additionally, for each problem the testing algorithm is run independently 31 times in order to facilitate a statistical analysis.

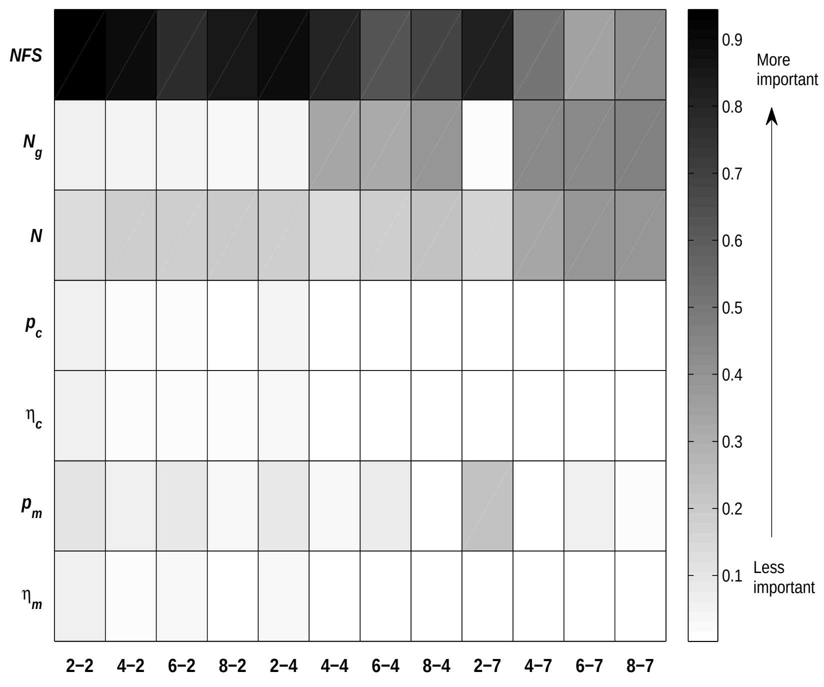
Test problems are chosen from the WFG test suite. Specifically, all the even-numbered WFG problems are invoked for 2-, 4- and 7-objective instances. For each test problem, the WFG position parameter and distance parameter are set to 18 and 14, that is, each problem has $n = 18 + 14 = 32$ decision variables.

5.2.3 Experiment results

Figure 5.1 shows the first-order (S_i) and total-order (S_i^T) effects of the parameters in PICEA-g in terms of the *HV* metric. Similarly, the results for PICEA-w are shown in Figure 5.2. S_i represents the contribution of a single parameter to the variance of the *HV* distribution. S_i^T measures the overall effect of a parameter to the variance of

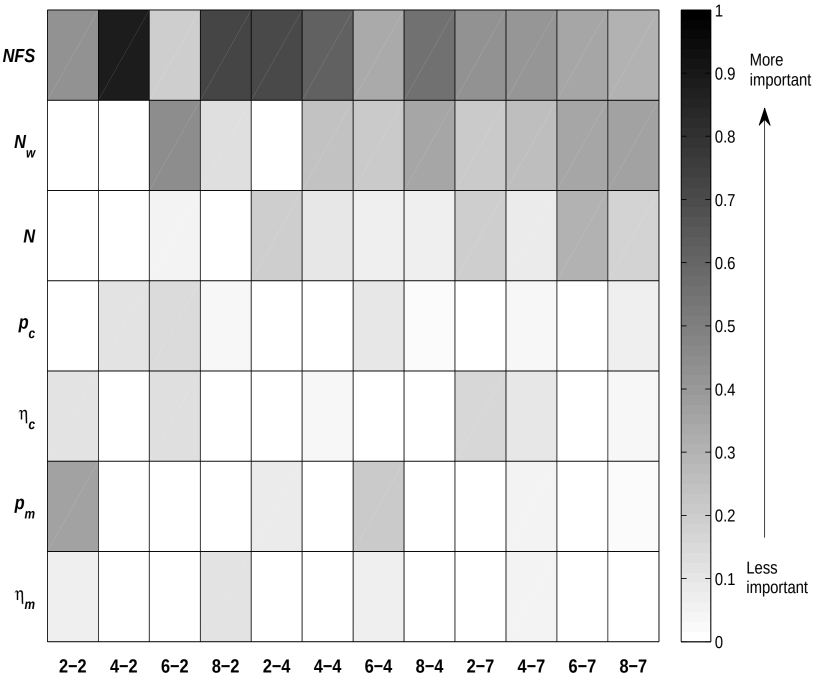


(a) the first-order effect

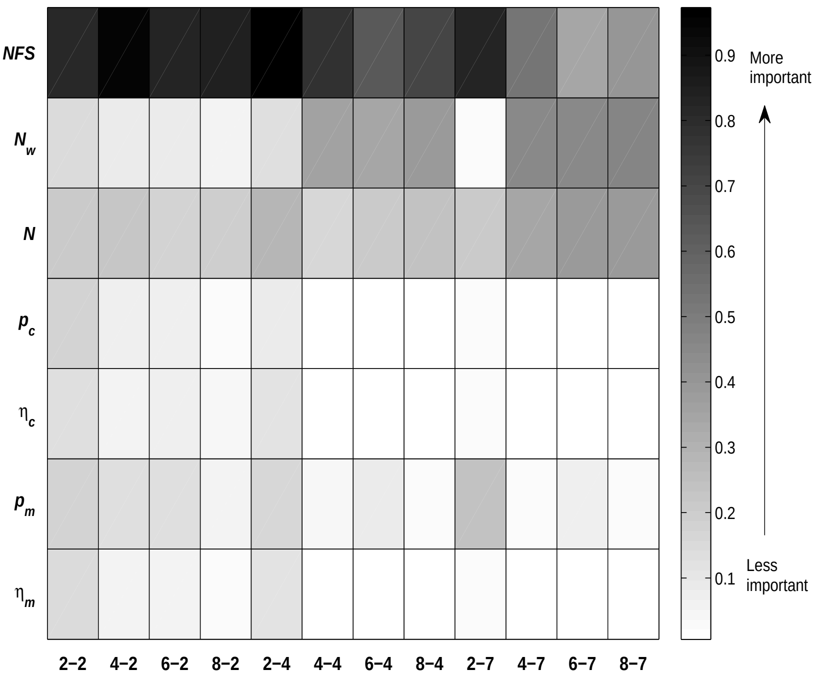


(b) the total-order effect

Figure 5.1: The sensitivity of an individual parameter in PICEA-g measured by the Sobol' variance decomposition method.



(a) the first-order effect



(b) the total-order effect

FIGURE 5.2: The sensitivity of an individual parameter in PICEA-w measured by the Sobol' variance decomposition method.

the *HV* distribution. In each figure, the shaded grid corresponds to a problem instance (x-axis) and one of the parameters (y-axis). For brevity in the x-axis label we use $n-Y$ to denote the problem WFG$n-Y$. For example, 2-4 refers to WFG2 with 4 objectives.

The intensity of the grid represents the importance of a parameter. A larger intensity indicates that the corresponding parameter has a significant effect on algorithm performance whereas a small intensity implies that the effect of changing the parameters is negligible. In Figures 5.1 and 5.2, all the indices are reliable as the calculated indices are all within the 95% BCIs. This implies that the sampling size $q = 2^h = 128$ is sufficient. Key observations are as follows:

(i) Combining both the first-order and the total-order effects, it is observed that the parameter *NFS* is the most significant parameter for both PICEAs across most problems. This indicates that PICEAs are user-friendly as the algorithm performance is controlled for the most part by a single parameter. Parameterising PICEAs should be easy in practice. A better performance can be obtained simply by lengthening the runtime of PICEAs, i.e., increasing the number of function evaluations. Additionally, it is worth mentioning that *NFS* is not always the most dominant parameter for all MOEAs. For example, it is demonstrated in Hadka and Reed (2012) that in some MOEAs, such as ε-MOEA (Deb et al., 2005) that utilises ε-dominance archives, the setting of ε-dominance is the most dominant component for controlling the algorithm performance.

(ii) Amongst the remaining six parameters, population size, N and preference size, N_p are relatively more significant to algorithm performance. As the number of objectives increases, these two parameters become more important. However, there is an exception, that is, the influence of N_p is not significant on WFG2 problems. One reason might be that the Pareto optimal front of WFG2 problem is disconnected and only occupies a small portion of the objective-space. Therefore, a small number of preferences are enough to guide the solutions approximate the entire Pareto optimal front. Another reason might be that the reference point chosen for the *HV* calculation is not appropriate; it cannot distinguish the quality of different Pareto approximation sets. With respect to the four genetic operators, p_m is the most dominant, especially on the WFG2 problems.

5.3 Parametric sensitivity analysis: a local manner

This section discusses the effect of parameters in a local manner. The influence of five parameters is investigated: one is the number of preferences and the other four parameters are associated with the genetic operator. The aim is to provide useful suggestions with respect to the parameter settings for non-expert users.

5.3.1 The effect of the number of preferences

Tables 5.2 and 5.3 present possible settings for N_g and N_w for different problems, respectively. Configurations for other parameters are shown in Table 5.4. The WFG benchmarks are selected as test problems. For each test problem, the algorithm is executed for 31 independent runs.

Table 5.2: Parameter settings of N_g.

	the number of goal vectors		
M	$M \times 10$	$M \times 100$	$M \times 500$
2	20	200	1000
4	40	400	2000
7	70	700	3500
10	100	1000	5000

Table 5.3: Parameter settings of N_w.

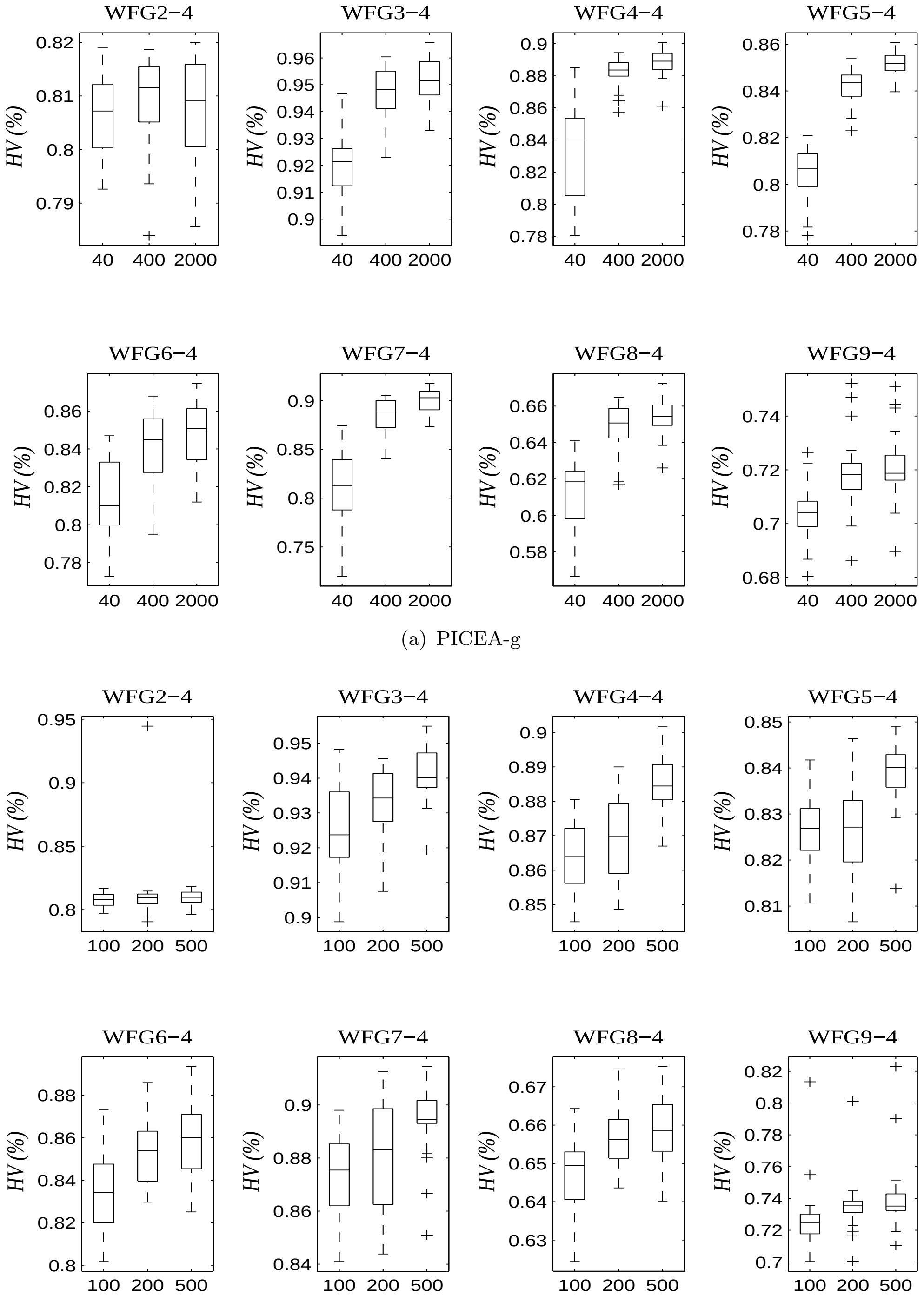
M	N_w		
2	100	200	500
4	100	200	500
7	100	200	500
10	100	200	500

Table 5.4: Parameter settings of PICEAs.

N	NFS	SBX (p_c)	SBX (η_c)	PM (p_m)	PM (η_m)
100	25000	1	15	$1/n = 1/32$	20

Figure 5.3(a) and Figure 5.3(b) present representative experimental results arising from these different settings when PICEA-g and PICEA-w are applied to 4-objective WFG problems, respectively. From the Figures, we can observe that for most problems (except for WFG2), the larger the number of preferences used, the better the performance that PICEAs can deliver. On the WFG2 problem, increasing the number of preferences does not necessarily produce better algorithm performance. This can also be observed from Figure 5.1 and Figure 5.2 where parameter N_p does not play a crucial role on WFG2 problems.

Additionally, it is obvious that as the number of preferences increases there is a corresponding increase in the resulting computational cost. Clearly, choice of a suitable value for N_p demands an appreciation of the trade-off between computational effort and performance. Although only results on 4-objective WFG problems are provided above, similar experiments were performed on WFG problems with 2, 7 and 10 objectives, with similar outcomes.



(a) PICEA-g

(b) PICEA-w

FIGURE 5.3: The performance of PICEAs on 4-objective problems for different numbers of preferences.

5.3.2 The effect of the genetic operator parameters

Experiment description

In PICEAs SBX and PM are applied as genetic operators. This section studies the effect of the two genetic operators. Two parameters (i.e., p_c and η_c) are considered for the SBX operator. p_c is the probability of applying crossover to a pair of parent solutions, and η_c is the magnitude of the expected variation from the parent values. The smaller the value for η_c, the larger the variation, although this also depends on the distance between the parent values (Deb et al., 2007). Note that originally there is another parameter p_e in the SBX operator which controls the probability of exchanging the child values for a decision variable between the child solutions. However, most of the time (e.g. Deb et al., 2002a; Zhang and Li, 2007) this parameter is set to zero. Thus, in this experiment, p_e is also set to zero for simplicity. There are two controllable parameters p_m and η_m in the PM operator: $-\ p_m$ is the probability of applying mutation to a pair of parent solutions, and η_m is the magnitude of the expected mutation from the parent values. The smaller the value for η_m, the larger the mutation.

The first experiment studies parameter settings for the SBX operator, i.e., p_c and η_c. The PM parameters are set constantly as $p_m = \frac{1}{n} = \frac{1}{32}$, $\eta_m = 20$. The second experiment studies parameter settings of the PM operator, i.e., p_m and η_m. The SBX operator parameters are set as $p_c = 0.9$, $\eta_c = 20$.

Table 5.5 and Table 5.6 show the genetic operator settings for each experiment, respectively. For each parameter, six values are sampled from a wide range of the parameter space, based on the study of Purshouse and Fleming (2007), and thus produce 36 groups of parameter settings for each experiment. The other parameters are set as $N = 100$, $N_p = 100$, $NFS = 25000$ for both experiments. For each group of parameter setting, the algorithm is executed for 31 independent runs in order to enable a statistical analysis.

TABLE 5.5: The SBX operator parameter settings.

SBX parameter	Values					
p_c	0.01	0.1	0.2	0.5	0.8	1
η_c	1	5	10	50	100	500

TABLE 5.6: The PM operator parameter settings.

PM parameter	Values					
p_m	0.01	0.1	0.2	0.5	0.8	1
η_m	1	5	10	50	100	500

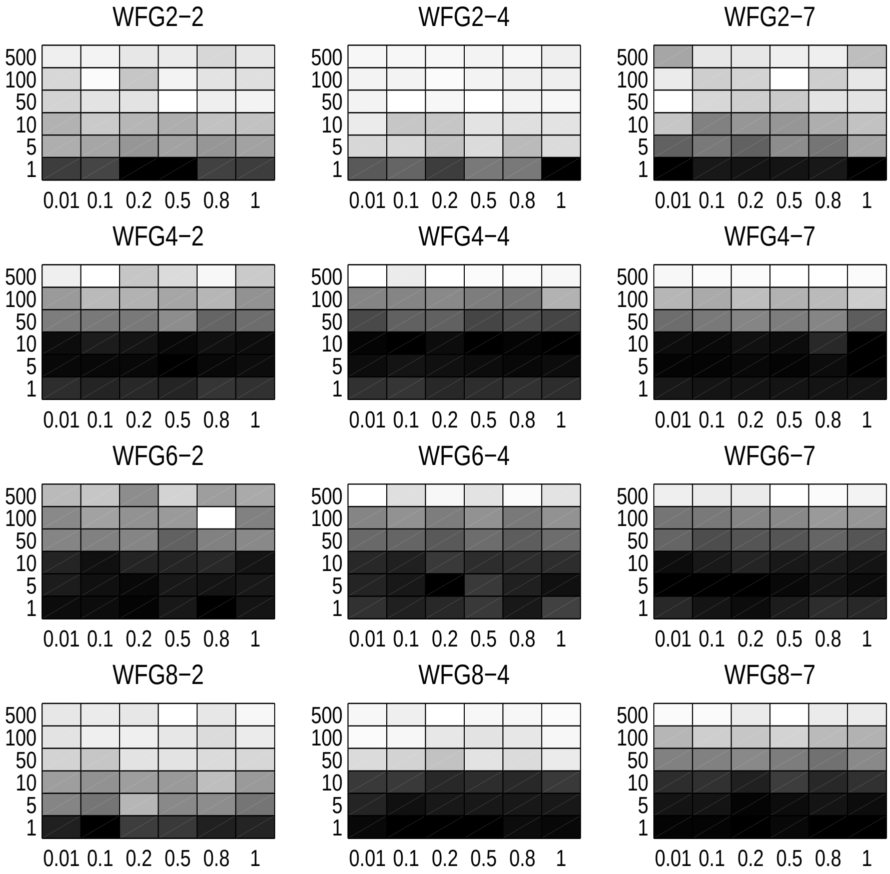


FIGURE 5.4: The SBX experimental results for PICEA-g.

Results of varying the recombination operator SBX

Algorithm performance associated with each pair of parameter settings is presented by a grayscale square on the control map. Performance is measured by the HV metric. The intensity of the shade represents algorithm performance – the darker the shade, the better the performance. Control maps shown in Figure 5.4 and Figure 5.5 show the results for PICEA-g and PICEA-w, respectively.

From Figure 5.4 and Figure 5.5 the following key observations can be made.

(i) The SBX parameters behave similarly on both the PICEAs. Different parameter combinations result in different algorithm performance. Among the test problems, algorithm performance varies more markedly for the WFG2 and WFG8 problems. The reason might be that the WFG2 problem has disconnected Pareto fronts and WFG8 problem features strong non-separability and parameter-dependency. These characteristics create difficulty in generating good solutions.

(ii) Both the PICEAs are less sensitive on parameter p_c. Their performance is more influenced by parameter η_c. For all the problems, the smaller the η_c, the better

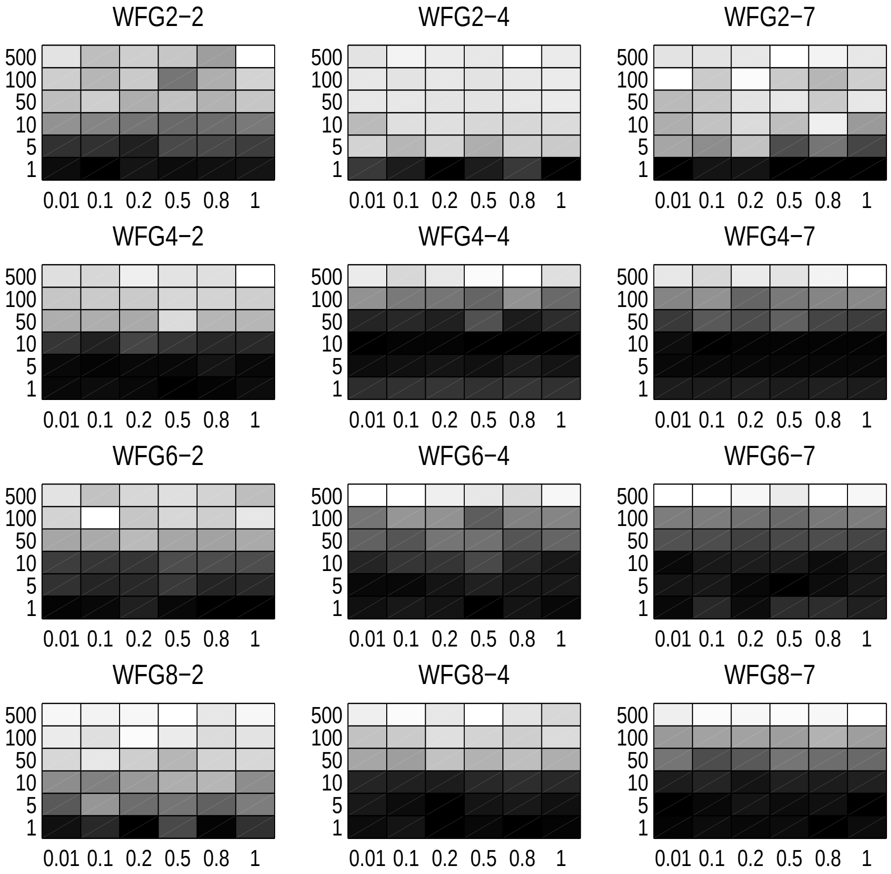


Figure 5.5: The SBX experimental results for PICEA-w.

the algorithm performance. To summarise, we find that $\eta_c \in [1,10]$ and $p_c \in [0.5,1]$ are good settings for the SBX operator.

Results of varying the mutation operator PM

Control maps of PICEA-g and PICEA-w with different mutation parameter settings are shown in Figure 5.6 and Figure 5.7. It is observed from the results that the PM parameters behave similarly on both the PICEAs. Different parameter combinations result in different algorithm performance. The algorithm performance degrades gradually from the top-left of the control map to the bottom-right of the control map. A PM operator configured with a small p_m and a large η_m leads to good performance. Overall, the suggested parameter settings for the PM operator are $p_m \in [0.01,0.1]$ and $\eta_m \in [10,500]$.

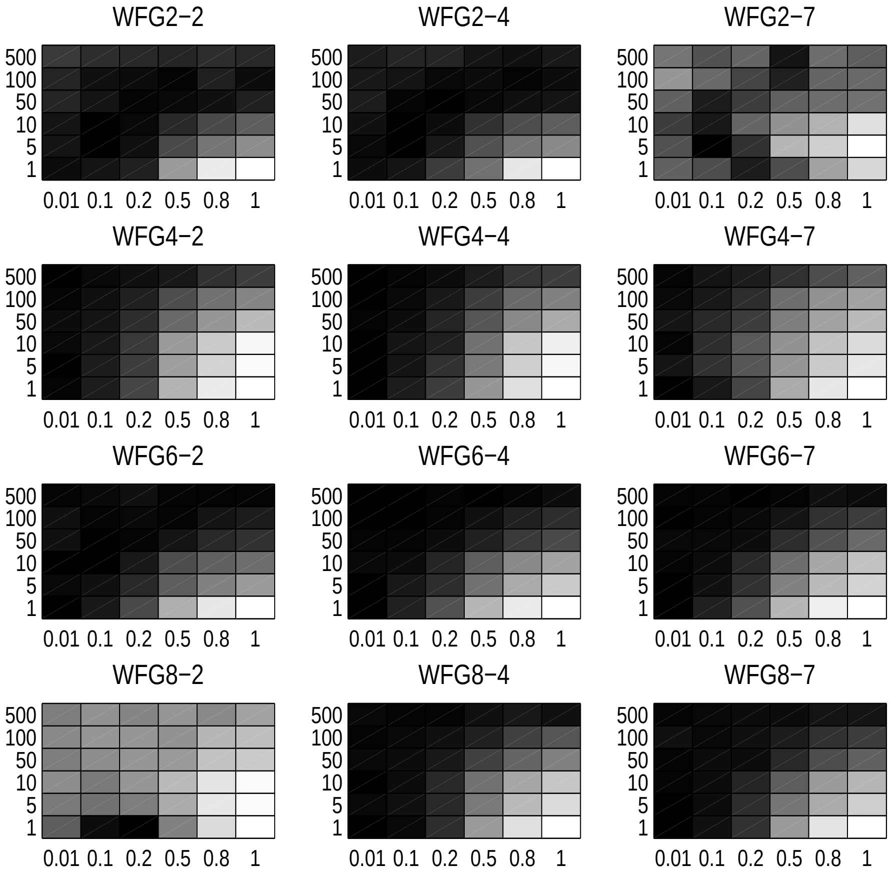


FIGURE 5.6: The PM experimental results for PICEA-g.

5.4 Summary

This Chapter analyses parameter sensitivities of PICEA-g and PICEA-w. It is found that the performance of both the PICEAs is largely determined by the number of function evaluations. This indicates that parameter configuration for PICEAs is easy in practice – increasing the number of function evaluations should directly result in better performance. In addition to the parameter *NFS*, the number of preferences also plays an important role on algorithm performance. The more preferences that are co-evolved, the better the performance that the algorithm delivers, although this improvement gradually becomes smaller.

In terms of the genetic operator configuration, it is found for the SBX operator that the more dominant parameter, η_c, should adopt a small value, while the less dominant parameter, p_c, could be set to a large value. We also suggest applying a small value for p_m and a large value for η_m, for the PM operator. These recommended settings are summarised in Table 5.7.

We have empirically studied the sensitivities of parameters in PICEA-g and PICEA-w, and provide some suggestions for the parameter settings, see Table 5.7. However, the

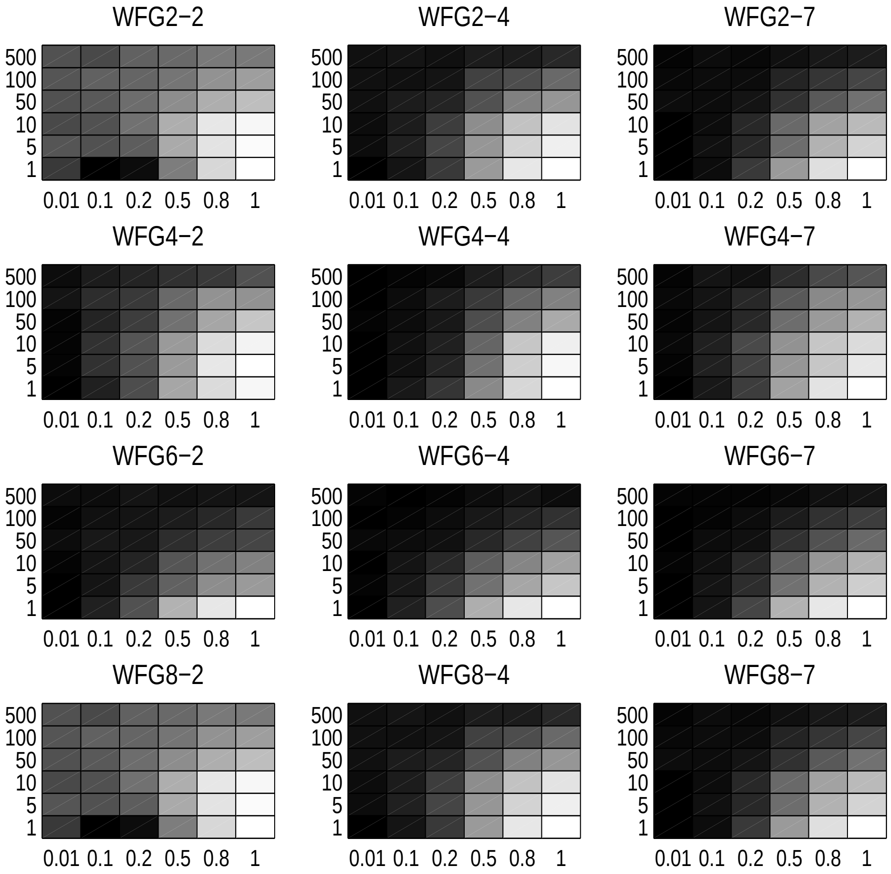


FIGURE 5.7: The PM experimental results for PICEA-w.

TABLE 5.7: The suggested SBX and PM parameter settings for PICEAs.

	SBX		PM	
	p_c	η_c	p_m	η_m
PICEAs	[0.5, 1]	[1, 10]	[0.01, 0.1]	[10, 500]

effects of parameter settings are often problem-dependent. A more reliable approach may be to develop a reliable and effective strategy, which enables the algorithm parameters to be updated according to the problem posed (e.g., Eiben et al., 1999; Brest et al., 2006). This have the potential to simultaneously reduce the burden of the users in terms of parameter configuration and improve the algorithm performance.

Chapter 6

"Whatever works best for you"- a unified approach for *a priori* and *progressive* multi-criteria decision-making

6.1 Introduction

Chapters 3 and 4 have proposed two effective MOEAs that are able to find a representative approximation of the Pareto optimal front. However, the ultimate goal of solving a MOP is not to obtain the entire Pareto optimal front but it is to help a decision-maker to consider the multiple objectives simultaneously and to identify a single Pareto optimal solution which is satisfactory (Coello Coello et al., 2007, pp. 31-47), that is, assisting multi-criteria decision-making (MCDM).

According to when decision-maker (DM) preferences are made, i.e., before, during or after the search, MCDM can be divided into three classes – *a priori*, *progressive* and *a posteriori* decision-making respectively. *A posteriori* decision-making attempts to first find a good approximation of the entire Pareto optimal front and subsequently to let the DM choose a preferred one. This scheme is effective on problems having 2 or 3 objectives. This is mainly because that obtaining a good approximation of the Pareto optimal front for bi- and tri-objective MOPs is not difficult. In addition, having known the entire Pareto optimal front, the DM can be confident of selecting his/her preferred solutions. However, *a posteriori* decision-making scheme becomes less effective on MaOPs. The reason is that approximation of the whole Pareto optimal front for MaOPs is computationally expensive, and the DM is usually only interested in some regions of the Pareto front (see Figure 6.1 for an example of a region of interest (ROI) to a decision-maker). Moreover, the number of Pareto optimal solutions required for describing the entire Pareto optimal front of a MaOP is usually very large. Selecting one preferred solution from all these solutions is cognitively difficult.

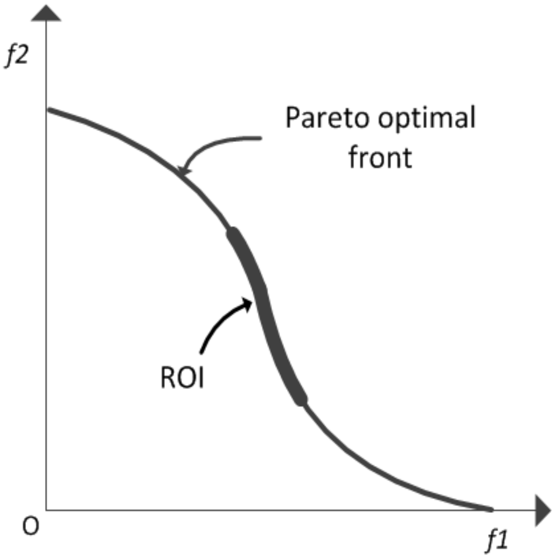


Figure 6.1: Illustration of a ROI.

Thus, to facilitate the process of decision-making, alternatives are to consider DM preferences in an *a priori* or *progressive* way. In these cases, the search can be guided by the preferences towards the ROI and away from exploring non-interesting solutions.

As reviewed in Section 2.5, a variety of methods have been introduced to model the preferences of a decision-maker, such as aspiration level, trade-off rate, reference point/direction and utility function. Coupled with these preference models, a number of preference-based MOEAs have also been proposed to search for solutions in the ROI to a decision-maker. For example, MOGA uses aspiration level (Fonseca and Fleming, 1998a) , G-MOEA uses trade-off rate (Branke, 2001), Thiele et al. (2009) incorporates reference point into IBEA, Deb and Kumar (2007a) incorporates reference direction into NSGA-II, Auger et al. (2009a) incorporates utility function into HypE. These approaches all have their own advantages and disadvantages. However, these approaches share some common points, that is, they are all based on a single preference model, and the DM preferences are specified using numerical values.

However, it is argued that a decision-maker might have his/her preferred way to express his/her preferences. In addition, in some cases it is easier for a decision-maker to express his preferences using aspiration levels and in others by weights or other forms. Furthermore, a decision-maker might find it much easier to specify his preferences simply by drawing, that is, brushing his preferred region in objective-space, rather than using numerical values. Therefore, it would be helpful to develop a unified approach which enables the decision-maker to articulate his preferences in different ways.

This Chapter proposes such a unified evolutionary multi-objective optimisation and decision-making approach. This approach is derived from the previously introduced algorithm PICEA-g, and is denoted as iPICEA-g. The rest of this Chapter is structured as follows. In Section 6.2 we describe our proposed approach, iPICEA-g. Section 6.3 shows the simulation results of using iPICEA-g with different preference formulations. Section 6.4 presents a case study – applying iPICEA-g to solve a real-world problem (aircraft control systems design) problem. Section 6.5 summarises this Chapter.

6.2 Articulating decision-maker preference into PICEA-g: iPICEA-g

To meet the needs of different DMs, this section describes our proposed approach with which iPICEA-g can cater simultaneously to different ways that a DM could specify his preferences. Section 6.2.1 presents the formulation of iPICEA-g and Section 6.2.1 explains the rationale of iPICEA-g.

6.2.1 The unified approach iPICEA-g: formulation

Three parameters are defined for the unified approach, iPICEA-g: a reference point in objective-space ($\mathbf{z}_R$), a weight ($\mathbf{w}$) and a search range (θ). $\mathbf{z}_R$ can be used to describe the aspiration level or simply serves as a reference point; $\mathbf{w} = (w_1, w_2, \cdots, w_M)$ is to introduce the DM's bias towards some objectives, $\sum_{i=1}^{M} w_i = 1$, $\forall i, w_i \geq 0$. By default w_i is set as $1/M$, indicating that there is no bias towards any objective. The θ parameter controls the size of ROI and its domain is $[0, \frac{\pi}{2}]$. $\theta \to \frac{\pi}{2}$ leads to a search over a large region of the Pareto optimal front while $\theta \to 0$ results in a search over a small and targeted region of the Pareto optimal front. θ is set to $\frac{\pi}{4}$ radians by default. An example in the bi-objective case is shown in Figure 6.2. Note that $\mathbf{z}_R$ can also be unattainable; this will be described later. Using the three parameters, the decision-maker can either express his preferences using aspirations or weights.

(i) If the DM specifies preferences as aspiration levels, then we set $\mathbf{z}_R$ as the aspirations, $w_i = 1/M, i = 1, \cdots, M$ and $\theta = \arccos(\sqrt{\frac{M-1}{M}})$. An example where $M = 2$, $w_1 = w_2 = 0.5$ and $\theta = \frac{\pi}{4}$ radians is shown in Figure 6.3(a).

(ii) If the DM specifies weights, then $\mathbf{z}_R$ is set to the *ideal* point (if the *ideal* point is unknown, an estimated value can be used instead), $\mathbf{w}$ represents the specified weight vector and θ controls the search range, see Figure 6.3(b). For example, when the DM specifies that objective f_1 is twice as important as objective f_2, $\mathbf{w}$ will be $(\frac{2}{3}, \frac{1}{3})$. Note that in this case, objective values need to be normalised: b

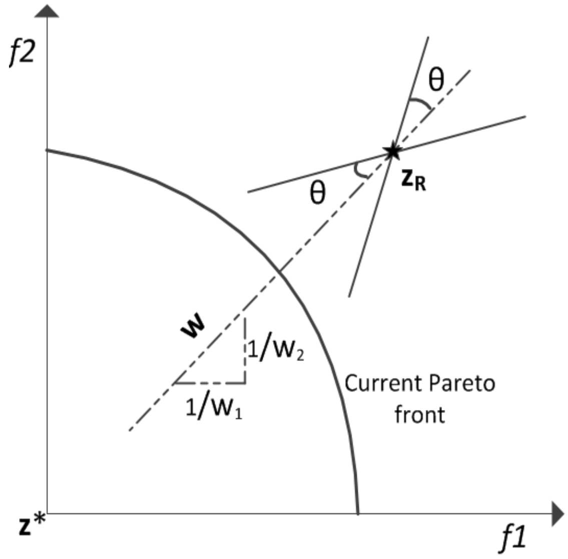
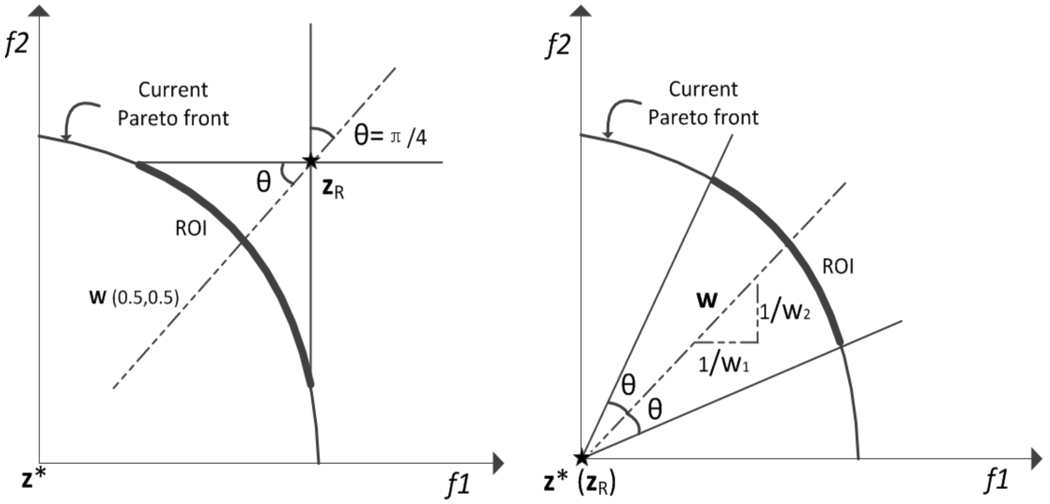


FIGURE 6.2: Illustration of the parameters $\mathbf{z}_R$, $\mathbf{w}$ and θ.



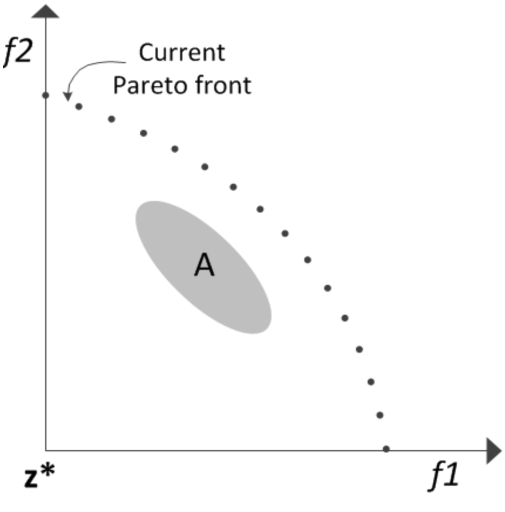
(a) Aspirations (b) Weights

FIGURE 6.3: Illustration of searching with aspiration level and weights.

$$\overline{f}_i = \frac{f_i - z_i^{ide}}{z_i^{nad} - z_i^{ide}} \tag{6.1}$$

(iii) Alternatively, the DM can specify his preferences using all three parameters together, see Figure 6.2.

In addition to specifying preferences using numerical values, the DM might find it cognitively easier to specify preferences visually by drawing, that is, "brushing" his preferred region in objective-space. According to the brushed region, see Figure 6.4(a), preference information is elicited and then incorporated into an algorithm to search for solutions in the preferred region, see Figure 6.4(b).

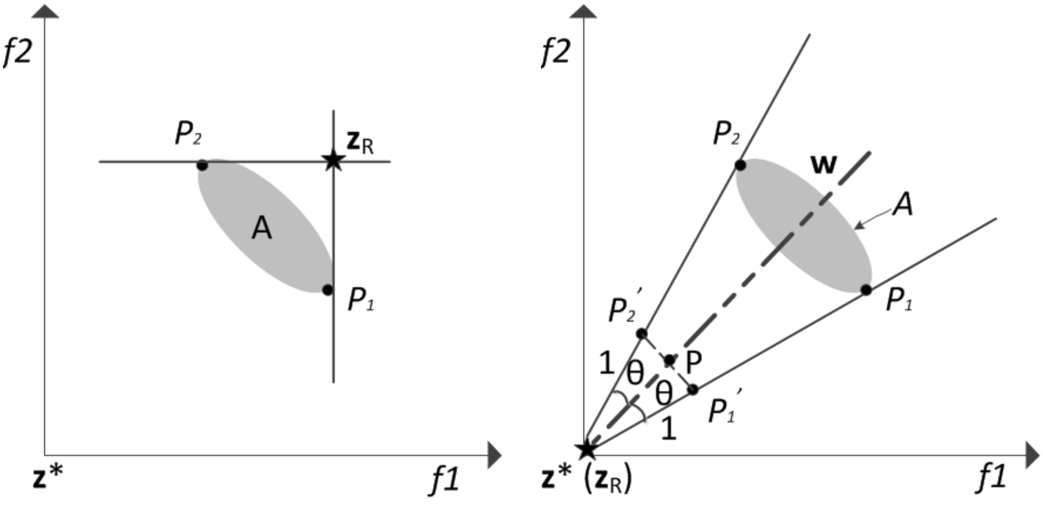


(a)

(b)

FIGURE 6.4: Illustration of the brushing technique.

Assuming that a set of solutions has been obtained and is presented to the DM, the DM then brushes his preferred region. Based on the brushed region, the three parameters are configured. Very naturally, there are two schemes to configure the parameters, as illustrated in Figure 6.5, the brushed region is labelled as A.



(a) Scheme 1

(b) Scheme 2

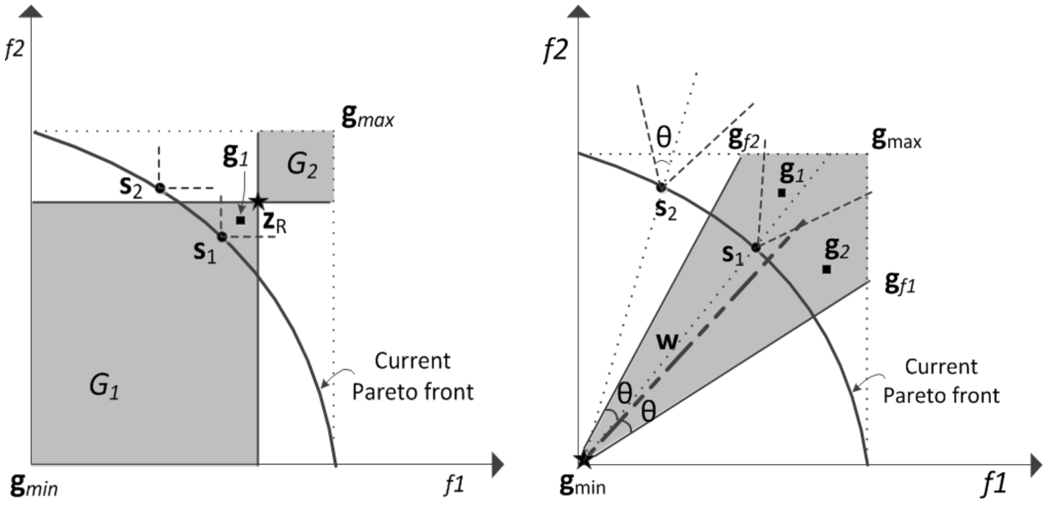
FIGURE 6.5: Illustration of parameters calculation for the brushing technique.

(i) When parameters are configured in Scheme 1, as shown in Figure 6.5(a), the search is similar to that using aspiration levels. Specifically, $\mathbf{z}_R$ is created by the maximum value of P_1 and P_2 in each objective. P_1 and P_2 are two extreme vectors of the brushed region A. The extreme vector has the largest value in one objective. Parameters $\mathbf{w}$ and θ are set as $w_i = 1/M, i = 1, \cdots, M$, $\theta = \arccos(\sqrt{\frac{M-1}{M}})$.

(ii) When parameters are configured in Scheme 2, as shown in Figure 6.5(b), the search is similar to that using weights. However, here weight $\mathbf{w}$ does not introduce a bias but describes the search direction from the *ideal* point to the centre of region A. Specifically, the three parameters are calculated as follows: first obtain the normalised vector P_1' and P_2' of the extreme points P_1 and P_2, i.e., $P_i = P_i/L_i, i = 1, 2$, where L_i is the Euclidean distance from $\mathbf{z}^*$ to P_i, and $|P_1'\mathbf{z}^*| = |P_2'\mathbf{z}^*| = 1$. $\mathbf{z}^*$ is the estimated *ideal* point. Then, the search direction $\mathbf{w}$ is determined by vector $\overrightarrow{P\mathbf{z}^*}$, where P is the centre of P_1' and P_2'. θ is then calculated by $\arccos(\overrightarrow{P\mathbf{z}^*} \cdot \overrightarrow{P_1'\mathbf{z}^*})$. Additionally, $\mathbf{z}_R$ is set to the *ideal* point $\mathbf{z}^*$.

6.2.2 The unified approach iPICEA-g: rationale

According to the fitness assignment scheme of PICEA-g, it can be observed that if the goal vectors are exclusively generated in a region then candidate solutions inside this region will be more likely to survive during the search. The reason is that these candidate solutions can dominate more goal vectors and so are more likely to gain higher fitness, while candidate solutions outside this region can only dominate a few goal vectors and so are likely to have lower fitness. Therefore, candidate solutions will exhibit a tendency to approach to the specified region. This has also been demonstrated in Section 3.7.1 that by using different goal vector bounds, PICEA-g can approximate different parts of the Pareto optimal front.



(a) Aspirations (b) Weights

FIGURE 6.6: Illustration of the goal vectors generated in different cases.

Inspired by this thinking, goal vectors in iPICEA-g are generated only within the region defined by $\mathbf{z}_R$, $\mathbf{w}$ and θ. Specifically, goal vectors are generated in the shaded region in Figures 6.6(a) and 6.6(b) when the preferences are modelled by aspirations and weights, respectively. By co-evolving candidate solutions with these specially generated goal vectors, candidate solutions are expected to be guided towards the ROIs. Similarly, when the preferences are specified using the brushing technique, goal vectors are generated in a similar way as shown in Figures 6.6(a) and 6.6(b) for the Scheme 1 and Scheme 2, respectively.

(i) When searching with aspiration levels, goal vectors are generated in both G_1 and G_2, extending both towards and away from the *ideal* point. This is to handle the case where the supplied $\mathbf{z}_R$ is unattainable. In this case, the Pareto-dominance relation is applied to determine whether a solution meets (i.e., dominates) a goal vector or not.

(ii) When searching with weights, goal vectors are generated along the direction $\mathbf{w}$ and spanned within θ radians, that is, goal vectors are generated in the shaded region closed by points $\mathbf{g}_{min}$, $\mathbf{g}_{f1}$, $\mathbf{g}_{f2}$ and $\mathbf{g}_{max}$.

Note that in this case, instead of using the Pareto-dominance relation, iPICEA-g employs the Pareto cone-dominance (Batista et al., 2011) to determine whether a goal vector $\mathbf{g}$ is met by a candidate solution $\mathbf{s}$ or not. Specifically, $\mathbf{g}_j$ is said to be Pareto cone-dominated by $\mathbf{s}_i$, if and only if the angle between the vector $\overrightarrow{\mathbf{s}_i\mathbf{g}_j}$ and the vector $\overrightarrow{\mathbf{s}_i\mathbf{g}_{min}}$ is not larger than θ. For example, in Figure 6.6(b), $\mathbf{g}_1$ is Pareto cone-dominated by candidate solution $\mathbf{s}_1$ while $\mathbf{g}_2$ is not Pareto cone-dominated by $\mathbf{s}_1$. Mathematically, given two feasible vectors $\mathbf{x}$ and $\mathbf{y}$ we say $\mathbf{x}$ Pareto cone-dominates $\mathbf{y}$, $\mathbf{x} \preceq_{cone} \mathbf{y}$: if and only if $\mathbf{y} \in \mathbb{C}$, where $\mathbb{C}$ is a generated cone, given by weight vector $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_M$, i.e. $\mathbb{C} = \{\mathbf{z} : \mathbf{z} = \lambda_1\mathbf{w_1} + \lambda_2\mathbf{w_2}, \cdots, \lambda_M\mathbf{w_M}, \sum\lambda_i = 1, \forall\lambda_i > 0\}$. Figure 6.7 illustrates the generated cone of $\mathbf{x}$ in a bi-objective space.

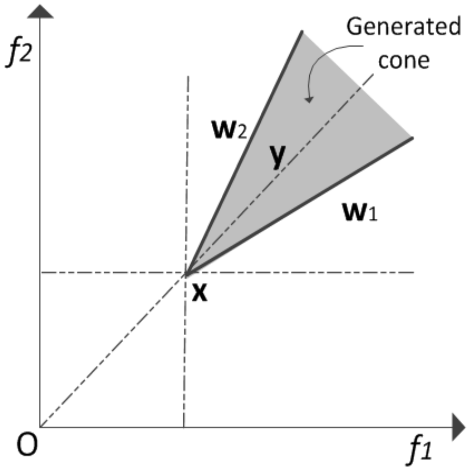


FIGURE 6.7: Illustration of a generated cone in 2-objective case.

Compared with Pareto-dominance, the use of Pareto cone-dominance can further emphasise the solutions along the reference direction, i.e., assigning higher fitness to these solutions. For example, in terms of Pareto-dominance, $\mathbf{s}_2$ can satisfy some goal vectors and therefore might be retained in the evolution. However, $\mathbf{s}_2$ is not in the ROI. When using Pareto cone-dominance, $\mathbf{s}_2$ cannot satisfy any goal vector, having the lowest fitness and then would be more likely to be disregarded in the evolution.

In both cases the lower, $\mathbf{g}_{min}$, and upper, $\mathbf{g}_{max}$, goal vector bounds are estimated based on the objective values of the non-dominated solutions $\mathbf{f}(S^*)$ found so far and the reference point, $\mathbf{z}_R$,

$$\begin{aligned} \mathbf{g}_{min} &= \min\left(\mathbf{z}_{Ri} \cup f_i(S^*)\right), i = 1, 2, \cdots, M \\ \mathbf{g}_{max} &= \max\left(\mathbf{z}_{Ri} \cup f_i(S^*)\right), i = 1, 2, \cdots, M \end{aligned} \tag{6.2}$$

In addition to the benefit that iPICEA-g can handle different types of DM preference, i.e. aspiration levels and weights, another major benefit of iPICEA-g is that multiple ROIs can be explored by simultaneously generating goal vectors for all the ROIs. It is also anticipated that iPICEA-g will perform well on many-objective problems because PICEA-g has been demonstrated to have good performance in this environment.

Finally, the interaction of the analyst and the decision-maker when using iPICEA-g is as follows:

(i) The analyst asks the decision-maker to specify his preferences. The preferences can be expressed by aspirations, weights or brushed regions. Based on these preferences, the three parameters ($\mathbf{z}_R$, $\mathbf{w}$ and θ) are configured.

(ii) Next, the algorithm sets the population size of candidate solutions, the number of goal vectors and the stopping criterion. Subsequently, iPICEA-g is executed until the stopping criterion is met.

(iii) The obtained solutions are presented to the decision-maker. If the decision-maker is satisfied with the provided solutions then stop the search process. Otherwise, ask the decision-maker to specify new preference information. For example, if the decision-maker would like to narrow the range of the obtained solutions, then reduce the setting of θ; if the decision-maker would like to bias one objective, e.g. f_i, then increase the value of w_i. Certainly, decision-maker can specify a different $\mathbf{z}_R$. After updating the preferences, return to step (ii).

An additional issue

Intuitively, driving the search towards a narrow region at the beginning of an optimisation process might cause a lack of population diversity as the candidate solutions may have similar phenotypes and thus result in low search efficiency and converge to a local optimum (Coello Coello et al., 2007, pp. 131-143). This issue is more likely to appear in *a priori* decision-making as the decision-maker might be only interested in a small Pareto region, i.e., the search range is small. In order to avoid this problem, we suggest starting the search with a large search range θ' and gradually, decreasing θ' to a preferred search range θ. The decreasing process is defined:

$$\theta_{use} = \theta' - (\theta' - \theta) \times (\frac{gen}{maxGen})^{\alpha}; \tag{6.3}$$

where gen is the current generation, $maxGen$ is the maximum generations, and α is to control the speed of decreasing. As shown in Figure 6.8, θ is decreased from $\frac{\pi}{4}$ radians to $\frac{\pi}{9}$ radians with different α. A large α corresponds to a slow decreasing speed in the early stages and a fast decreasing speed in the later stages of evolution. $\alpha = 2$ is suggested for use after a preliminary analysis. The benefit of this strategy will be illustrated next in Section 6.3.1.

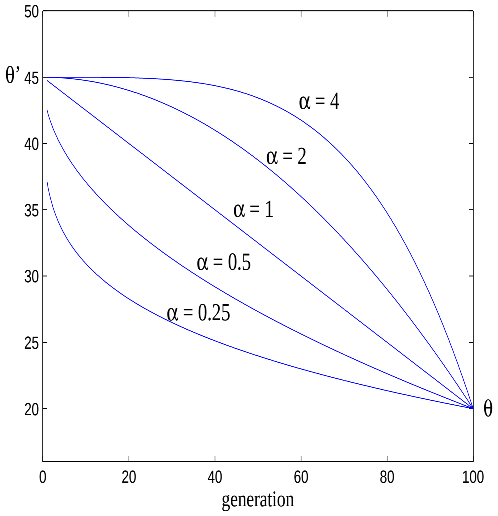


FIGURE 6.8: Illustration of the decrease process of θ with $maxGen = 100$.

6.3 Experiments

In this section, we examine the performance of iPICEA-g on different benchmarks from the ZDT, DTLZ and WFG test suites. The selected test problems have convex, concave, connected, as well as disconnected, Pareto front geometry. In all the experiments, the simulated binary crossover (SBX, $p_c = 1$, $\eta_c = 15$) and polynomial mutation (PM, $p_m = 1/n$ per decision variable and $\eta_m = 20$, where n is the number of decision variables) are applied as genetic operators. To quantitatively measure the performance of iPICEA-g, the generational distance (GD) metric is employed to evaluate the convergence of the obtained solutions.

To gain insight into the effect that the parameters $\mathbf{z}_R$, $\mathbf{w}$ and θ have in iPICEA-g, we study them in isolation in Section 6.3.1. In Section 6.3.2, we describe the performance of iPICEA-g when searching with aspiration levels or weights for regions of interest in an *a priori* preference articulation setting. In Section 6.3.3, we introduce what we refer to as the brushing technique (Buja et al., 1996; Hauser et al., 2002), which is useful for interactive preference articulation and decision-making.

6.3.1 Parameters and their effect in iPICEA-g

The bi-objective 20-variable DTLZ2, which has a concave Pareto optimal front is selected as the test problem to study the effect of the three parameters. Note that unless otherwise specified, in all the experiments of this study, the initial search range θ' is set as $\theta + \frac{\pi}{18}$ radians.

The effect of $\mathbf{z}_R$

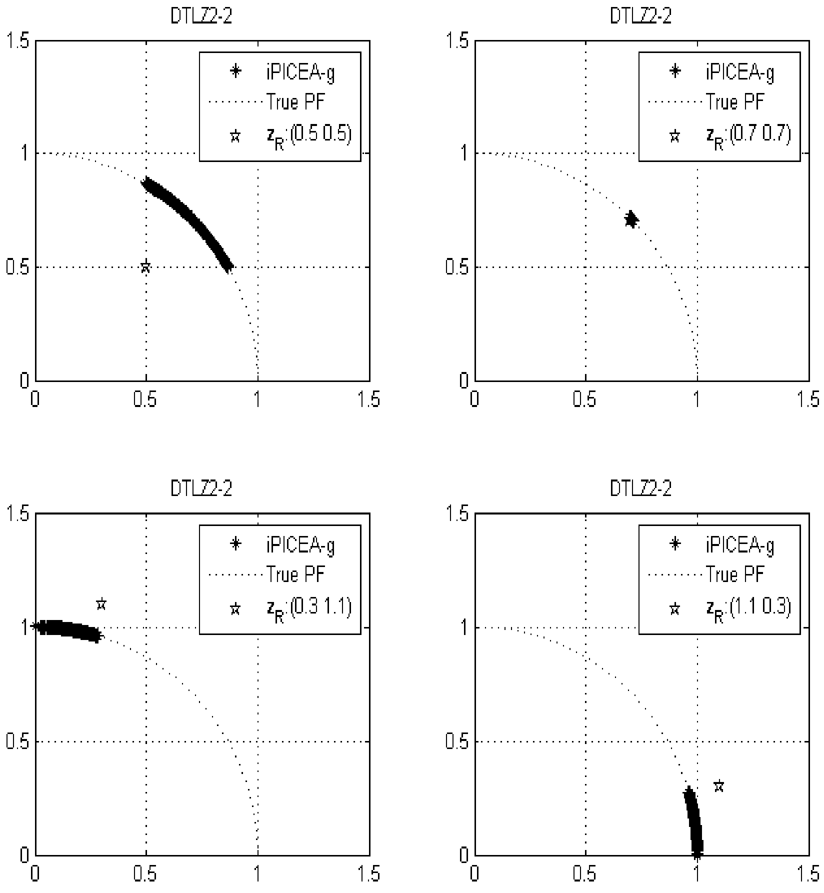


Figure 6.9: Solutions obtained by iPICEA-g for different $\mathbf{z}_R$.

If the decision-maker wishes to have solutions around a point then we set $\mathbf{z}_R$ as the specified point. For example, the DM specifies (i) one infeasible (0.5,0.5) reference point; (ii) one reference point (0.7,0.7) near the Pareto optimal front; and (iii) two feasible reference points, (0.3,1.1) and (1.1,0.3). Figure 6.9 shows the obtained results after an optimisation run for 250 generations with iPICEA-g for different cases. During the optimisation the search direction is set as $\mathbf{w} = (0.5, 0.5)$ and the search range is set as $\theta = \frac{\pi}{4}$. It can be seen in the Figure 6.9 that in every case iPICEA-g can find a set of solutions close to the true Pareto front. This result is encouraging as it suggests that iPICEA-g is able to handle both feasible and infeasible aspiration levels. Moreover, as the fitness calculation in iPICEA-g preserves the Pareto-dominance relation and therefore, unlike g-NSGA-II whose performance is degraded when $\mathbf{z}_R$ approaches to the Pareto optimal front (Ben Said et al., 2010), iPICEA-g performs well when $\mathbf{z}_R$ is close to the Pareto optimal front.

The effect of w

Assuming that decision-maker would like to specify a preference for one objective over another, we use $\mathbf{w}$. For example, the DM specifies that (i) both the objectives are equally important then $\mathbf{w} = (\frac{1}{2}, \frac{1}{2})$ or (ii) objective f_1 is twice as important as f_2 then $\mathbf{w} = (\frac{2}{3}, \frac{1}{3})$ or (iii) objective f_1 is half as important as f_2 then $\mathbf{w} = (\frac{1}{3}, \frac{2}{3})$. Figure 6.10 shows the obtained results after performing iPICEA-g for 250 generations with $\mathbf{w} = (\frac{1}{2}, \frac{1}{2})$, $\mathbf{w} = (\frac{1}{3}, \frac{2}{3})$ and $\mathbf{w} = (\frac{2}{3}, \frac{1}{3})$, respectively. During the simulation, $\mathbf{z}_R = (\frac{1}{2}, \frac{1}{2})$ and $\theta = \frac{\pi}{6}$ radians. As seen in Figure 6.10, the obtained solutions are along the search direction $\mathbf{w}$ and span over $\frac{\pi}{6}$ radians. Consider a case where $\mathbf{w} = (\frac{2}{3}, \frac{1}{3})$, in this scenario, a higher *importance* is ascribed to the first objective function, f_1, which leads the optimisation algorithm to be more sensitive to fluctuations in its value. Therefore, it is more likely to obtain lower values for the first objective.

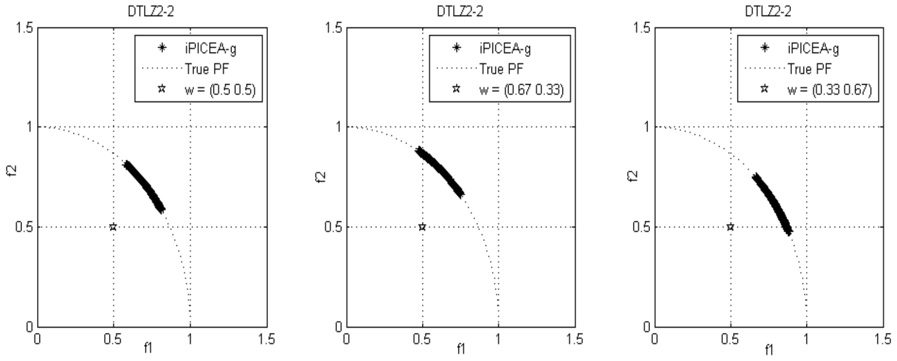


FIGURE 6.10: Solutions obtained by iPICEA-g for different $\mathbf{w}$.

The effect of the θ

If the decision-maker seeks to obtain solutions widely spread over the Pareto front then θ should be set to a relatively large value. For instance for $\theta = \frac{\pi}{2}$ radians, the whole Pareto front can be obtained, given of course that the reference point is chosen appropriately, see Section 6.3.1. If the DM has a preference towards a set of solutions in a narrow region of the Pareto front then $\mathbf{w}$ can be directed towards that region and the θ parameter should be set to a small value, $\frac{\pi}{180}$ radians, for example. The obtained results are depicted in Figure 6.11 and as in the previous experiments iPICEA-g was allowed to operate on the population of solutions for 250 generations for three values of the θ parameter, $\frac{\pi}{2}$, $\frac{\pi}{4}$ and $\frac{\pi}{180}$. During the optimisation, $\mathbf{z}_R$ and $\mathbf{w}$ have been set to $(0.3, 0.3)$ and $(0.5, 0.5)$ respectively. Clearly, the obtained portion of the Pareto front contracts for decreasing values of θ.

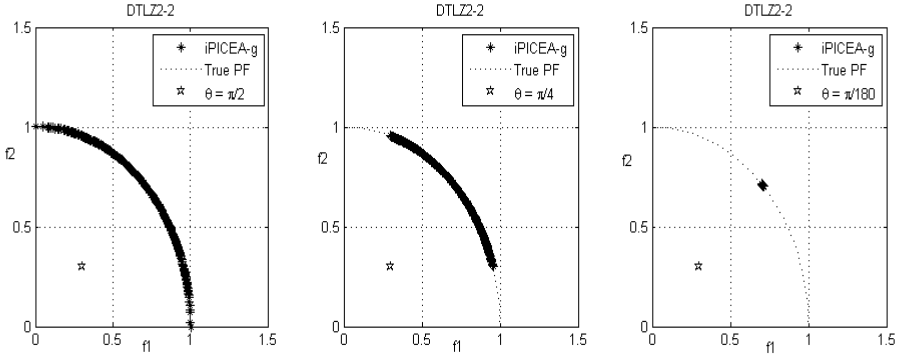


FIGURE 6.11: The distribution of solutions obtained by iPICEA-g with different θ values.

The effect of adaptive management of θ

In this section we explore the effect that adaptation of the θ parameter has on algorithm performance, see Equation 6.3. We take the 10-variable ZDT4, featuring multi-modality, as a test problem. Assuming the DM wishes to obtain optimal solutions around $\mathbf{z}_R = (0.1, 0.4)$, the search direction is the ray emanating from the *ideal* point (here it is the origin) to $\mathbf{z}_R$, i.e. $\mathbf{w} = (0.2, 0.8)$ and a relatively narrow spread range of ROI is required, say $\theta = \frac{\pi}{18}$. We perform two runs of iPICEA-g with the initial search range $\theta' = \frac{\pi}{4}$ and $\theta' = \frac{\pi}{9}$, respectively. In this experiment too, iPICEA-g is allowed to execute for 250 generations per run. The initial population size is set to $N = 100$ and $N_g = 100$. The obtained results are shown in Figure 6.12. iPICEA-g with $\theta' = \frac{\pi}{4}$ has obtained solutions near the Pareto optimal region. However, the solutions obtained by iPICEA-g with $\theta' = \frac{\pi}{9}$ are still far away from the Pareto optimal front. Such results clearly illustrate that the adaptive process of θ is beneficial to iPICEA-g. A large initial search range enables iPICEA-g to perform more exploration at the beginning, that is, a more diverse set of individuals can be evolved in the search process. Therefore the possibility of being trapped in a local optimum is reduced. The obtained solutions are more likely to converge to the true Pareto front.

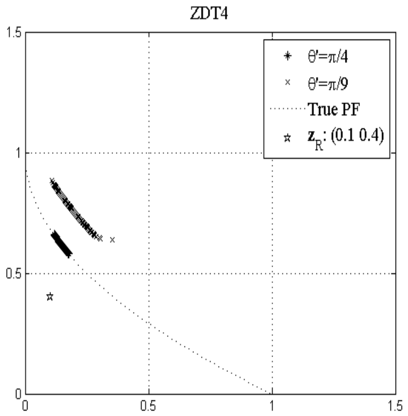


Figure 6.12: The obtained results with $\theta' = \pi/4$ and $\theta' = \pi/9$ radians.

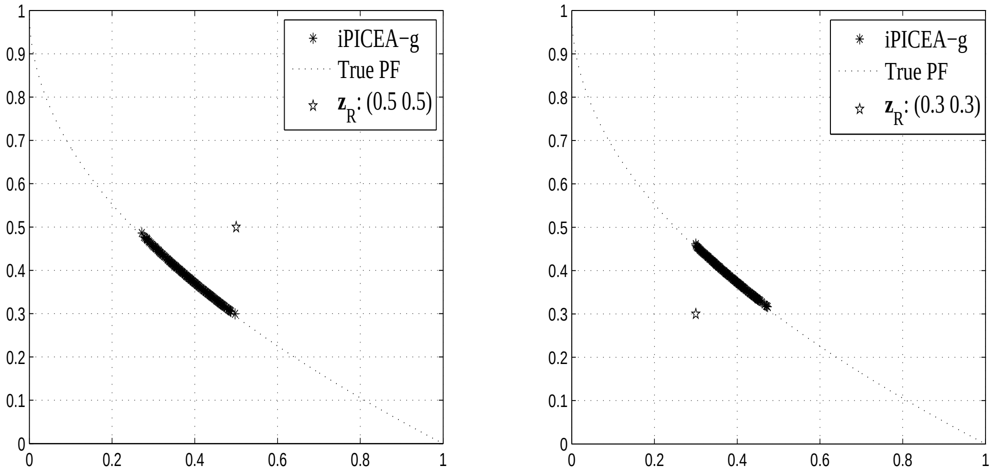
6.3.2 Results for *a priori* preference articulation

In this section we illustrate the scenario of searching with aspiration levels and searching with weights, respectively. Unless stated otherwise, the population size of candidate solutions and the number of co-evolved goal vectors are set to 200. For every test problem, the reported results are evaluated using 31 independent trials. The obtained Pareto front that has the median GD value over the 31 runs is shown.

Search with aspiration levels

Here we consider the case where the DM specifies preferences as aspiration levels. The ZDT1, DTLZ2 and DTLZ4 problems are used in the simulation and iPICEA-g is run for 250 generations on each problem.

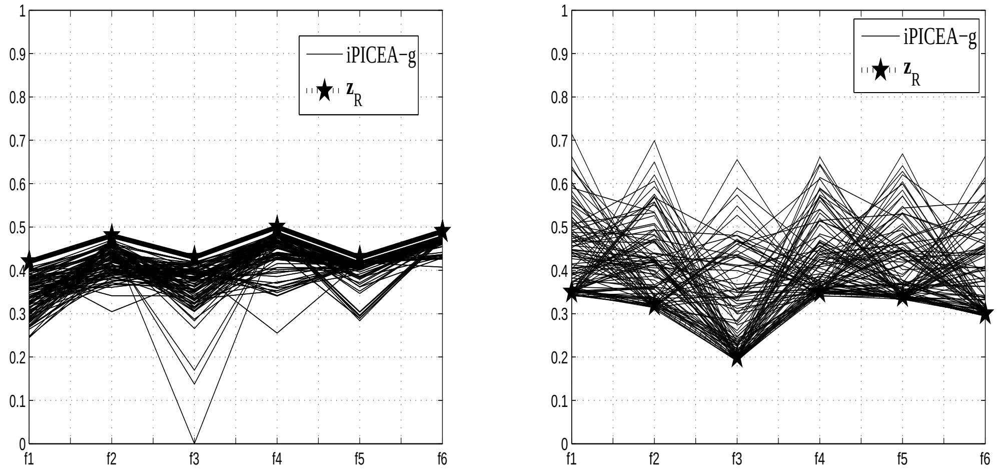
Searching for a ROI: the 30-variable ZDT1 is considered as the first test problem. We assume that the DM specifies his aspiration level as $(0.5, 0.5)$ and so $\mathbf{z}_R = (0.5, 0.5)$; correspondingly, $\mathbf{w}$ is configured as $(0.5, 0.5)$ and θ is set as $\frac{\pi}{4}$ radians. After running iPICEA-g for 250 generations a set of satisfied solutions is obtained shown in Figure 6.13(a). We can see that visually all the obtained solutions are very close to the true Pareto front. Quantitatively, the mean GD value is 0.0006 ± 0.0002 which indicates that the obtained solutions converge well. In some cases the DM might specify an infeasible aspiration level, i.e., $\mathbf{z}_R = (0.3, 0.3)$. After running iPICEA-g for 250 generations, a set of solutions are obtained, as shown in Figure 6.13(b). Although the obtained solutions have not met the aspiration level, they have approached the Pareto optimal front. This is also confirmed by the GD metric, where the mean value is 0.0007 ± 0.0002.



(a) ZDT1

(b) ZDT1

Figure 6.13: Illustration of searching with aspiration level on ZDT1.

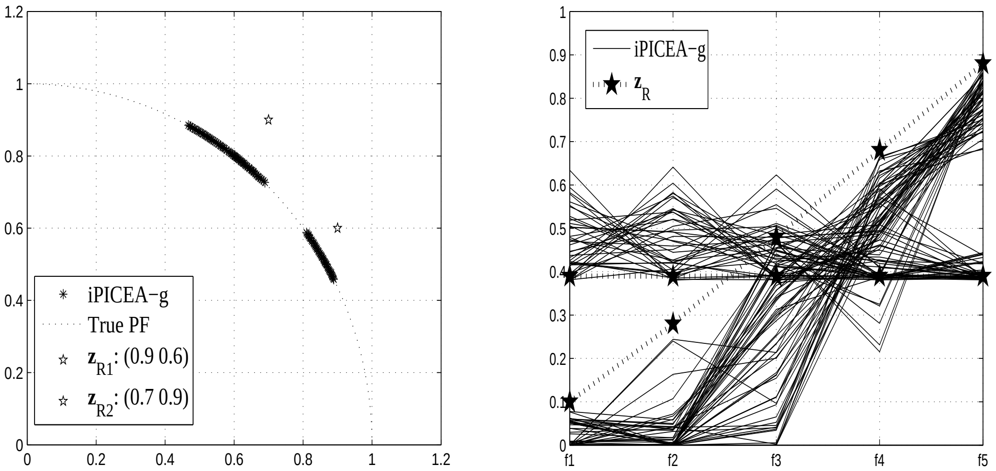


(a) DTLZ4-6

(b) DTLZ4-6

Figure 6.14: Illustration of searching with aspiration level on DTLZ4-6.

Similarly, iPICEA-g is tested on the 6-objective DTLZ4 problem for both cases: attainable aspirations, $\mathbf{z}_{R1} = (0.42, 0.48, 0.43, 0.5, 0.43, 0.49)$ and unattainable appropriations, $\mathbf{z}_{R2} = (0.35, 0.32, 0.2, 0.35, 0.34, 0.3)$. Figure 6.14 shows the obtained solutions after running iPICEA-g for 250 generations. The obtained solutions are shown in Figure 6.14(a) using Parallel coordinate system (An introduction to the Parallel coordinate system is offered in Appendix C). It is observed that these solutions have met the aspiration level. Moreover, the mean GD value is 0.0036 ± 0.0006, indicating that the solutions converge well to the Pareto optimal front. From Figure 6.14(b), it is observed that all the solutions have not met the aspiration level. However, after examining the GD value we find that the mean GD value is 0.0027 ± 0.0019 which indicates that the solutions have approached to the Pareto optimal front. Furthermore, the Pareto optimal front of DTLZ4 lies on the surface of a hyper-sphere with radius 1 ($\sum_{i=1}^{M} f_i^2 = 1$) in the first quadrant (Deb et al., 2002b). Having computed $\sum_{i=1}^{4} f_i^2$ for all the obtained solutions (across 31 runs), we find all values are within the range $[1.0141, 1.0528]$, which indicates that all the obtained solutions have almost converged to the true Pareto front.



(a) DTLZ2-2

(b) DTLZ4-5

FIGURE 6.15: Illustration of searching with multiple aspiration levels.

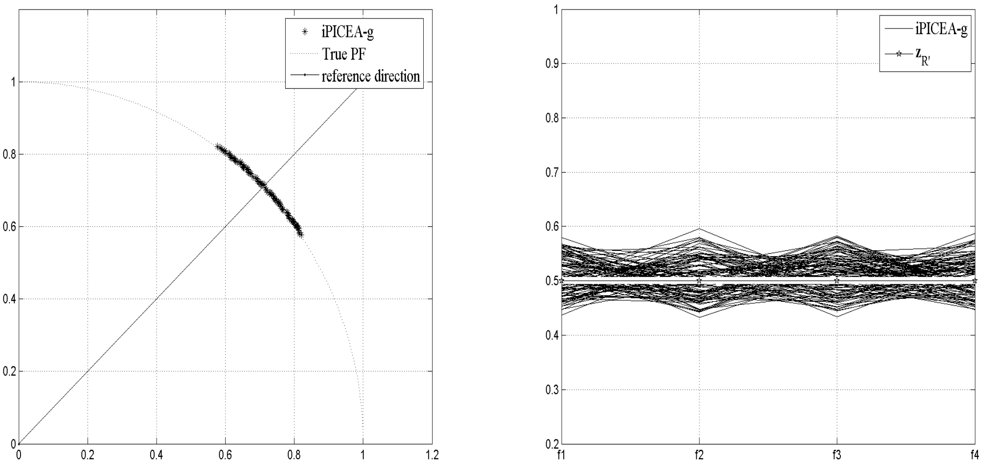
Searching for multiple ROIs: next we demonstrate the performance of iPICEA-g on searching for multiple ROIs. The 2-objective DTLZ2 is chosen as the first test problem. Assume that the DM specifies his preferences that objective f_1 should be better (smaller) than 0.7 and f_2 should be smaller than 0.9, or, alternatively, that f_1 is smaller than 0.9 and f_2 is smaller than 0.6. Therefore, the reference point $\mathbf{z}_R$ is set to $(0.7, 0.9)$ and $(0.9, 0.6)$, denoted as $\mathbf{z}_{R1}$ and $\mathbf{z}_{R2}$, respectively. After running iPICEA-g for 250 generations, two sets of solutions are obtained, and both sets of solutions have met the corresponding aspirations. Visually, both sets of solutions are close to the true Pareto front. Quantitatively, the mean GD value over the 31 runs is 0.0011 ± 0.0004 for $\mathbf{z}_{R1}$, and is 0.0010 ± 0.0003 for $\mathbf{z}_{R2}$, which shows that the obtained solutions converge well to the Pareto optimal front.

iPICEA-g is also tested on the 5-objective DTLZ4 problem. Assuming that the DM specifies two aspiration levels, where one is attainable $\mathbf{z}_{R1} = (0.10, 0.28, 0.48, 0.68, 0.88)$ and the other is unattainable, $\mathbf{z}_{R2} = (0.39, 0.39, 0.39, 0.39, 0.39)$. For each aspiration, a set of solutions is obtained, as shown in Figure 6.15(b). Visually, we observe that solutions have met the aspiration $\mathbf{z}_{R1}$ while they have not met $\mathbf{z}_{R2}$ (as it is unattainable). Quantitatively, the mean GD value is 0.0029 ± 0.0013 for $\mathbf{z}_{R1}$, and is 0.0024 ± 0.0011 for $\mathbf{z}_{R2}$, which confirms that the obtained solutions converge well to the Pareto optimal front.

These results confirm that iPICEA-g when searching with aspirations is able to achieve a set (or sets) of solutions with good convergence to the true Pareto front and that is also of interest to the decision-maker.

Search with weights

Here we consider the case where the DM specifies his preferences as weights. Again, the experiments are conducted for searching a single ROI and multiple ROIs, respectively.



(a) DTLZ4-2

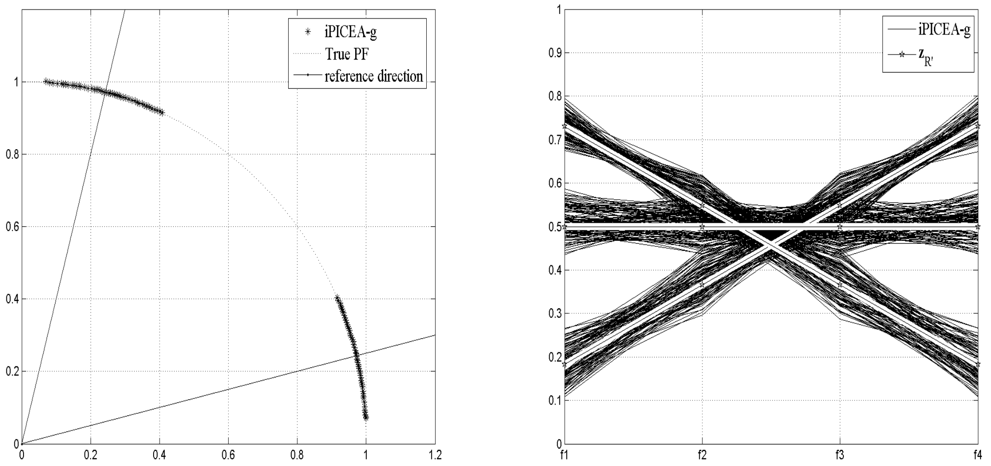
(b) DTLZ4-4

FIGURE 6.16: Illustration of searching with weights on DTLZ4.

Searching for a ROI: first, the bi-objective DTLZ4 problem is used as a test problem. Assuming that the DM specifies that the two objectives have equivalent importance, then the weight vector is configured as $\mathbf{w} = (0.5, 0.5)$. Correspondingly, θ is set as $\frac{\pi}{6}$ radians in order to obtain a moderate range of solutions. After running iPICEA-g for 250 generations a set of satisfied solutions are obtained, as shown in Figure 6.16(a). It is observed that all the obtained solutions are close to the Pareto optimal front, and the solutions show no bias towards any objective.

Second, the 4-objective DTLZ4 problem is chosen as another test problem to examine the scalability of iPICEA-g. Again, we assume that the DM has no bias towards any objective, then the weight vector is configured as $\mathbf{w} = (0.25, 0.25, 0.25, 0.25)$, and θ is set as $\frac{\pi}{36}$ radians in order to obtain a small range of solutions. After running iPICEA-g for 250 generations a set of solutions are obtained as shown in Figure 6.16(b). All the obtained solutions are concentrated around the projected reference point $\mathbf{z}_{R'}$ shown as $-\star-$ (the white line in the middle). $\mathbf{z}_{R'}$ represents the Pareto optimal solution along the search direction $(0.25, 0.25, 0.25, 0.25)$, namely it is the point of intersection of the ray emanating from the *ideal* point with direction $\mathbf{w}$. After computing $\sum_{i=1}^{4} f_i^2$ for all obtained solutions, the values lie within the range $[1.0391, 1.0903]$, therefore indicating that all solutions have converged to the true Pareto front.

Statistically, the mean GD values over the 31 independent runs are 0.0010 ± 0.0005 for DTLZ4-2 and 0.0041 ± 0.0012 for DTLZ4-4. Such results confirm that the obtained solutions converge well to the true Pareto front. Note that DTLZn-Y represents the DTLZ(n) problem with Y objectives.



(a) DTLZ2-2

(b) DTLZ2-4

Figure 6.17: Illustration of searching with weights on DTLZ2.

Searching for multiple ROIs: the bi- and 4-objective DTLZ2 problems are used as test problems. For the DTLZ2-2 problem, we assume that the DM bias one objective over another: either f_1 is four times as important as f_2, or, alternatively, f_2 is four times as important as f_1. Thus, two weight vectors are constructed, i.e., $\mathbf{w}_1 = (0.2, 0.8)$ and $\mathbf{w}_2 = (0.8, 0.2)$. Correspondingly, for both cases, θ is set as $\frac{\pi}{12}$ radians so as to obtain a close range of solutions. After running iPICEA-g for 250 generations two sets of solutions are obtained as shown in Figure 6.17(a). We can observe that all the obtained solutions are close to the Pareto optimal front. This is also confirmed by the GD metric, that is, the mean GD value is 0.0006 ± 0.0003.

For the DTLZ2-4 problem, we assume that the DM is satisfied with any of the following three cases: (i) either the four objectives are equally important, i.e., $\mathbf{w} = (0.25, 0.25, 0.25, 0.25)$, (ii) f_1 is four times as important as f_4, f_2 is three times as important as f_4, f_3 is twice as important as f_4, i.e., $\mathbf{w} = (0.4, 0.3, 0.2, 0.1)$, and (iii) f_1 is $\frac{1}{4}$ times as important as f_4, f_2 is $\frac{1}{2}$ times as important as f_4, f_3 is $\frac{3}{4}$ as important as f_4, i.e., $\mathbf{w} = (0.1, 0.2, 0.3, 0.4)$. Correspondingly, the search range is assumed to be $\theta = \frac{\pi}{20}$ radians. After running iPICEA-g for 250 generations three sets of solutions are obtained as shown in Figure 6.17(b). It is observed that in each set, solutions are concentrated around the projected reference point $\mathbf{z}_{R'}$ shown as $-\star-$. After computing $\sum_{i=1}^{4} f_i^2$ for all obtained solutions, the values lie within the range $[1.0391, 1.0903]$, indicating that all solutions have converged close to the true Pareto front. Statistically, the mean GD values over the 31 independent runs are 0.0024 ± 0.0011 for DTLZ2-4. Such results further confirm that the obtained solutions converge well to the true Pareto front.

These results demonstrate that iPICEA-g works well when searching with weights. It is able to achieve solutions that are close to the true Pareto front and are also in the regions of interest to the decision-maker.

A discussion on the use of weights

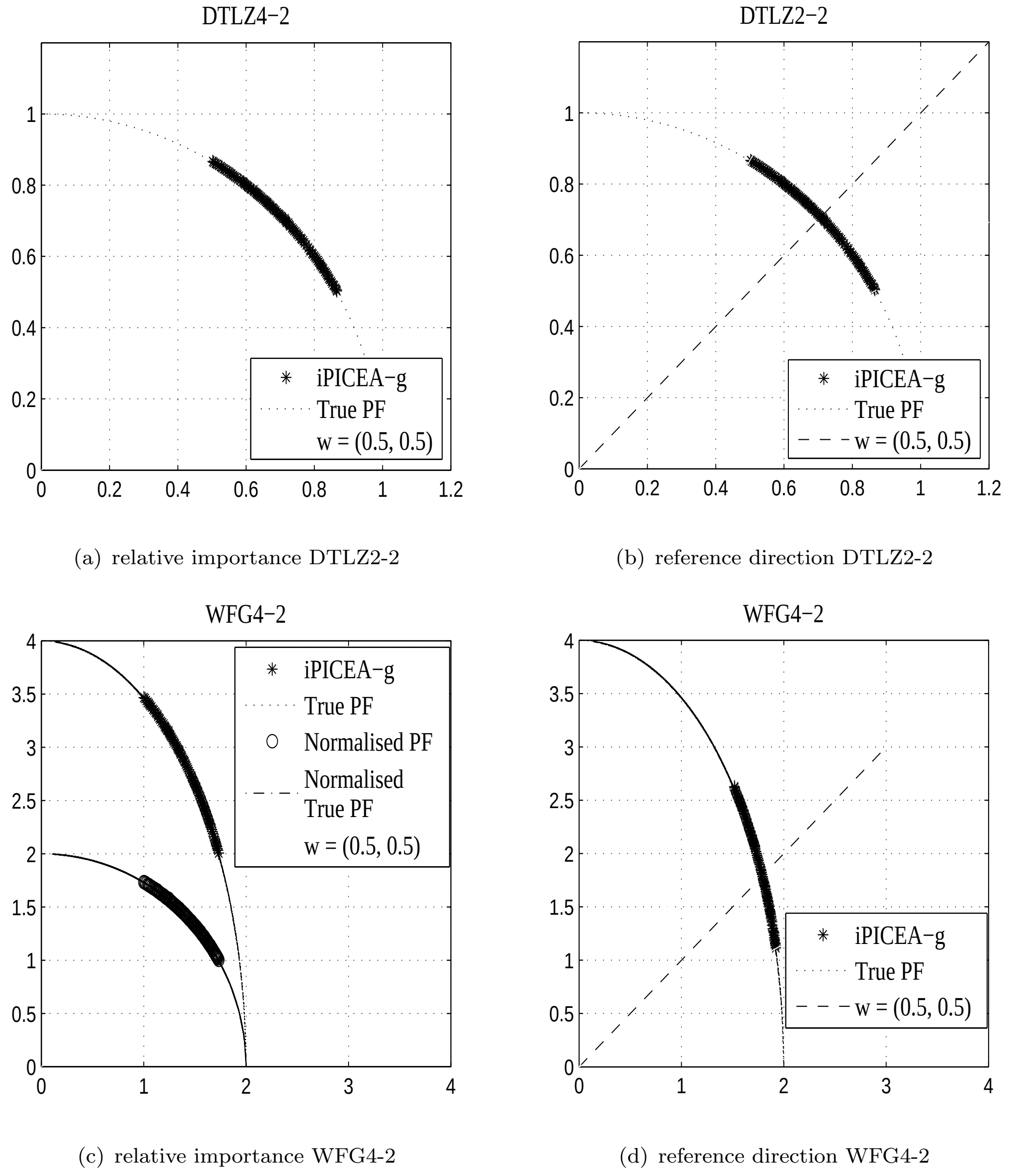
A weight vector $\mathbf{w}$ can either be interpreted as the relative importance of objectives or simply as a reference direction. Different interpretations might lead to different results. In the following, we illustrate the difference of these two interpretations. The 2-objective DTLZ2 and WFG4 benchmarks are used as test problems. The Pareto optimal front of DTLZ2-2 has the same trade-off magnitude, while the Pareto optimal front of WFG4-2 has a different trade-off magnitude.

Assuming that the weight vector is $(0.5, 0.5)$ (correspondingly, θ is set as $\frac{\pi}{12}$ radians and $\mathbf{z}_R$ is set as $[0, 0]$), when $\mathbf{w}$ is interpreted as relative importance, it means f_1 is the same as important as f_2; when it is interpreted as reference direction, it describes a direction from the ideal point to a reference point having the same objective values, e.g., (0.5,0.5). Figure 6.18 presents the obtained solutions for the two cases. Clearly we can observe that for DTLZ2-2 $\mathbf{w}$ used as reference direction or relative importance leads to the same results. However, for WFG4-2 the obtained results are different.

These results illustrate that for problems having the same trade-off magnitude (i.e., the same range of objective value), there is no difference between the two interpretations. However, the two interpretations lead to different results when a problem has a different trade-off magnitude. The reason is that when $\mathbf{w}$ is taken as the relative importance of objectives, implicitly, it is assumed that Pareto front geometry is disregarded, which means the search is conducted on the normalised solutions. When $\mathbf{w}$ is interpreted to be the reference direction, the decision-maker aims to find solutions along the specified direction, and normalisation is not required.

6.3.3 *Progressive* scenario

Although we have only demonstrated the effectiveness of iPICEA-g when searching with aspirations and weights in Section 6.3.2, it is reasonable to assume that iPICEA-g should also work well when the preferences are expressed using the brushing technique. This is because the brushing technique essentially simulates the search using aspirations (Scheme 1) and weights (Scheme 2).



(a) relative importance DTLZ2-2

(b) reference direction DTLZ2-2

(c) relative importance WFG4-2

(d) reference direction WFG4-2

Figure 6.18: Weights: relative importance and reference direction.

In this section, we simply illustrate the use of the brushing technique. To describe the scenario, we solve the bi-objective ZDT1 and four-objective DTLZ4 problems by simulating an interactive search process. The population size of candidate solutions and the number of goal vectors is set to $N = 200$ and $N_g = 200$. Note that in this illustration, the parameters are configured according to Scheme 2.

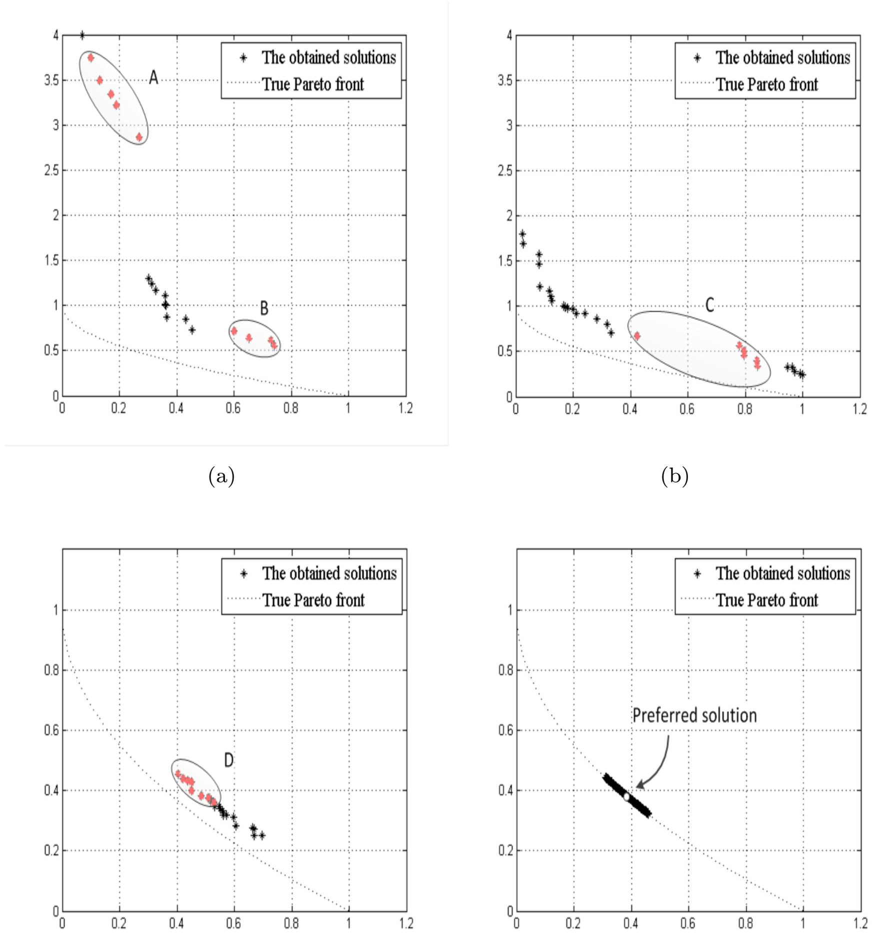
Bi-objective ZDT1

First, iPICEA-g is run for 10 generations without incorporating any preferences. The aim is to roughly know the range of the objectives so as to inform the DM's initial preferences. The obtained solutions are shown in Figure 6.19(a).

Second, the DM brushes his preferred regions, i.e., ellipses A and B, see Figure 6.19(a). The parameters of iPICEA-g are then configured based on the brushed solutions, which are $\mathbf{w} = (0.05, 0.95)$, $\theta = \frac{5\pi}{36}$ radians and $\mathbf{w} = (0.45, 0.55)$, $\theta = \frac{\pi}{18}$ radians for A and B, respectively. After running iPICEA-g for 50 more generations, two sets of improved solutions are found, as shown in Figure 6.19(b).

Third, we assume that the DM is not satisfied with the obtained solutions. He again brushes a set of solutions that are of his interest, see Figure 6.19(b), solutions in ellipse C. The related parameter settings are $\mathbf{w} = (0.76, 0.24)$, $\theta = \frac{\pi}{12}$ radians. By running iPICEA-g for another 50 generations, a set of solutions are found, see Figure 6.19(c).

Lastly, the DM remains dissatisfied, and, would like to focuses on a part of these solutions – a set of solutions that is in a more specific region of interest (that is in region D) is brushed (see Figure 6.19(c)). iPICEA-g is run for 50 more generations. The related parameters are configured as $\mathbf{w} = (0.5, 0.5)$, $\theta = \frac{\pi}{18}$ radians. A set of better solutions are found. The DM is now happy to choose a solution from this set, see Figure 6.19(d).



(a)

(b)

(c)

(d)

FIGURE 6.19: Interactive scenario on a 30-variable instance of the ZDT1 test problem (viewed in objective-space).

4-objective DTLZ2

Similarly, iPICEA-g is run without introducing any preference for 10 generations. A set of solutions is found, as shown in Figure 6.20(a).

Next, the DM brushes the preferred solutions for each objective as shown by the band in Figure 6.20(a). The selected solutions are shown in Figure 6.20(b). Parameters $\mathbf{w}$ and θ are then calculated as $(0.25, 0.25, 0.25, 0.25)$ and $\frac{\pi}{6}$ radians. iPICEA-g is run for 50 more generations. An improved set of solutions is obtained (see Figure 6.20(c)).

Thirdly, assuming that DM is dissatisfied with the obtained solutions. He brushes some solutions that are of interest (see Figure 6.20(d)). Based on the brushed solutions, two ROIs are identified. The related parameters are configured, that is, $\mathbf{w} = (0.3986, 0.3500, 0.1105, 0.1409)$, $\theta = \frac{\pi}{12}$ radians and $\mathbf{w} = (0.1124, 0.2249, 0.3498, 0.3128)$, $\theta = \frac{7\pi}{90}$ radians. The brushed solutions are shown in Figure 6.20(e). After running iPICEA-g for another 50 generations, more solutions are found, see Figure 6.20(f).

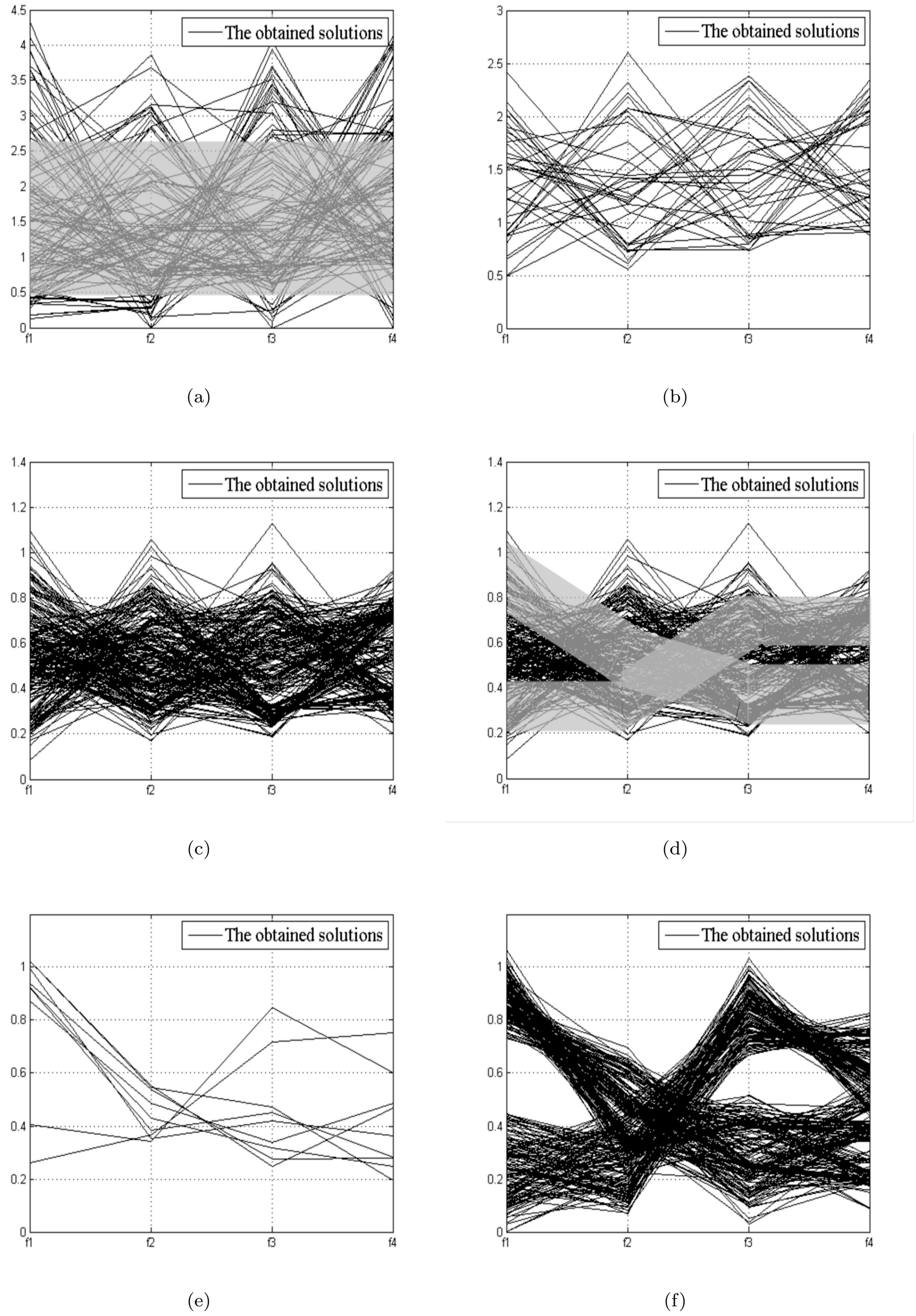
Fourthly, the DM is still not satisfied with the obtained solutions. He decides to explore one set of the obtained solutions. Again, he brushes his preferred region, as shown in Figure 6.21(a) (solutions existed in this region are shown in Figure 6.21(b)), and run iPICEA-g for 50 more generations. According to the brushed region, $\mathbf{w}$ is set to $(0.3691, 0.2773, 0.1383, 0.2153)$, θ is set as $\frac{\pi}{36}$ radians. As seen from Figure 6.21(c), a set of refined solutions are found in this preferred region. We compute $\sum_{i=1}^{4} f_i^2$ for all the obtained solutions. The value lies within the range of $[1.0190, 1.041]$ which indicates the obtained solutions have converged well to the true Pareto front. The DM is now happy to choose a solution from this set. The solution shown as the white dashed line is selected, see Figure 6.21(d).

6.4 Real-world application: aircraft control systems design

Section 6.3 has demonstrated the effectiveness of iPICEA-g on benchmark problems. This section assesses the performance of iPICEA-g using a real-world problem, i.e., aircraft control system design. This problem is taken from the study of Tabak et al. (1979). Section 6.4.1 describes this problem in detail where performance criteria and decision variables are described. Section 6.4.2 describes how this problem is solved by using iPICEA-g.

6.4.1 Problem description

The dynamic equations of the aircraft motion are often highly nonlinear. In practice, the customary treatment of these equations is to convert them into a linear problem



(a)

(b)

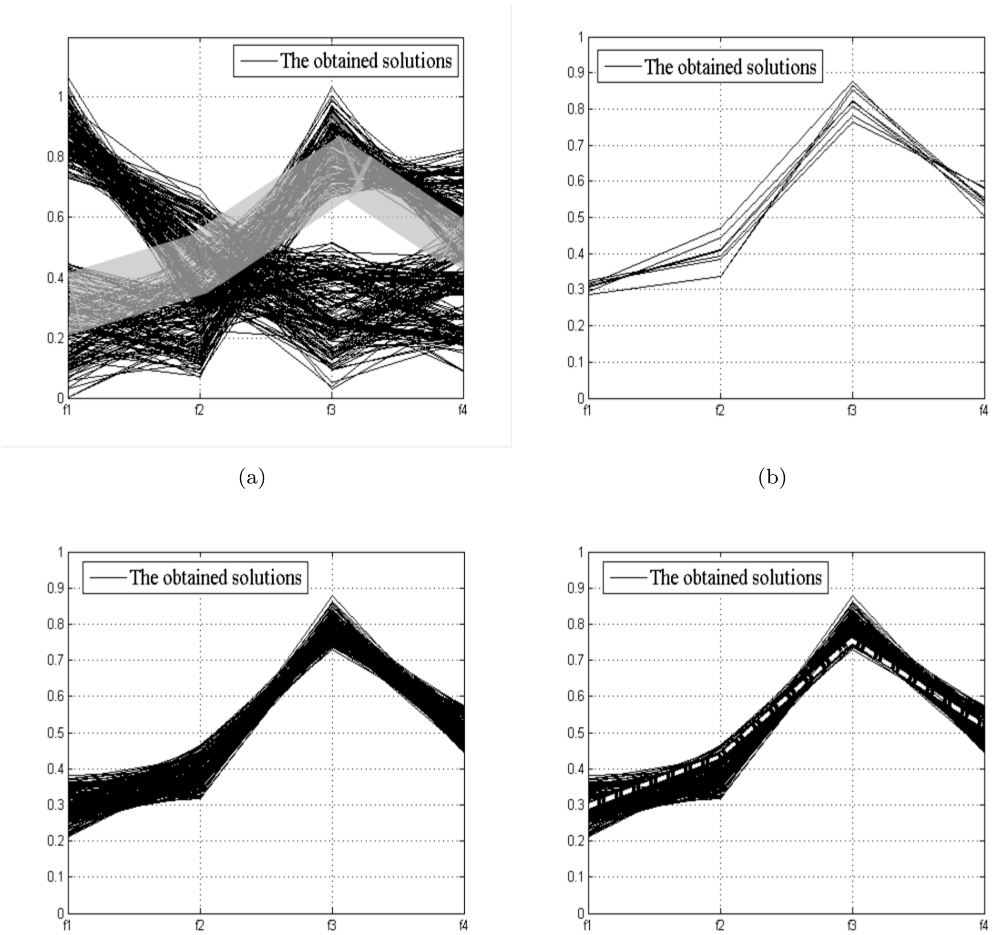
(c)

(d)

(e)

(f)

FIGURE 6.20: Interactive scenario for a 4-objective instance of the DTLZ2 test problem: part A (viewed in objective-space).



(a)

(b)

(c)

(d)

FIGURE 6.21: Interactive scenario for a 4-objective instance of the DTLZ2 test problem: part B (viewed in objective-space).

by using some small deviations from an equilibrium trajectory (Tabak et al., 1979). This study investigates the lateral response of the aircraft motion. According to Tabak et al. (1979), the lateral response can be formulated by a fourth order set of linear state equation:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\boldsymbol{\mu} \tag{6.4}$$

where $\mathbf{x}$ is a state vector containing the variable sideslip angle (β), yaw rate (r), roll rate (p) and bank angle (ϕ), $\boldsymbol{\mu}$ is the control vector that has variables, aileron control motions (δ_a) and rudder control motions (δ_r). The matrices A and B are set as follows according to Tabak et al. (1979):

$$A = \begin{bmatrix} -0.2482 & -0.9879 & -0.1574 & -0.0204 \\ -10.8574 & -0.5504 & -0.2896 & 0 \\ -199.8942 & -0.4840 & -1.6025 & 0 \\ 0 & 0.1566 & 1 & 0 \end{bmatrix} \tag{6.5}$$

$$B = \begin{bmatrix} 0 & 0.0524 \\ 0.4198 & -12.7393 \\ 50.5756 & 21.6573 \\ 0 & 0 \end{bmatrix} \tag{6.6}$$

The control operator $\boldsymbol{\mu}$ is further formulated as

$$\boldsymbol{\mu} = C\boldsymbol{\mu}_p + K\mathbf{x} \tag{6.7}$$

where $\boldsymbol{\mu}_p = [\delta_{ap}, \delta_{rp}]^T$ is the pilot's control input vector. As described in Tabak et al. (1979), δ_{ap} and δ_{rp} are specified as 16 and 0, respectively. C and K are two gain matrices of the form

$$C = \begin{bmatrix} 1 & 0 \\ k_5 & 1 \end{bmatrix} \tag{6.8}$$

$$K = \begin{bmatrix} k_6 & k_2 & k_1 & 0 \\ k_7 & k_3 & k_4 & 0 \end{bmatrix} \tag{6.9}$$

Experience shows that feedback from the bank angle (ϕ) is not crucial and so the last column of the K matrix is set to zero. Substituting Equation 6.7 into Equation 6.4 we obtain

$$\dot{\mathbf{x}} = (A + BK)\mathbf{x} + BC\boldsymbol{\mu}_p \tag{6.10}$$

Next, we introduce the decision variables and objective functions of this problem.

Decision variables

In this problem variables $k_1, k_2, \cdots, k_7$ are the decision variables that need to be established by the optimisation procedure. k_i represents gains applied to various signals, involved in the aircraft control system. High gains tend to cause sensitivity to sensor noise, and may also cause saturation of the control actuator response, so k_i should be configured as small as possible. In this experiment, k_i is set within the interval $(0, 1)$.

The stability of the system is determined by the eigenvalues of the matrix $A + BK$, that is, the stability characteristics of the system can be modified by changing the value of k_i.

Performance criteria

In this system, eight performance criteria ($M = 8$) are required to be satisfied:

(i) Four performance criteria are related to the stability parameters, that is, the spiral root λ_s, the damping in roll root λ_r, the damping ratio of the Dutch-roll complex pair ε_d and the Dutch roll frequency w_d. These parameters are derived from the characteristic roots of the matrix $A+BK$. The spiral root (λ_s) should be as small as possible, since a small real root, in general, corresponds to a tendency of returning to wings level straight flight after a small bank angle disturbance. A marginally stable roll damping root (λ_r) could be satisfactory, however, a larger negative real root of λ_r is preferred. The damping ratio ε_d should be as large as possible in order to avoid persistent oscillations and large overshoots in the response. A larger frequency w_d is preferred since this leads to a good stability.

(ii) The speed and the steadiness of the basic roll response to the aileron are two other criteria. These two criteria are determined by the bank angle ϕ at two specific times (i.e., $\phi(t_1)$ and $\phi(t_2)$) in response to maximum aileron input. The first time is chosen for providing a rapid initial response. The second time is chosen for insuring that the response remains steady.

(iii) Another criterion is to minimise the deviation of the overall sideslip angle, λ, i.e., $\min(\Delta\beta)$. $\Delta\beta$ must meet the response speed requirement. A small $\Delta\beta$ can decouple the roll response from the yaw-sideslip modes, producing smooth and precise roll responses.

(iv) The last criterion is to minimise the control effort. This is implemented by keeping the gains as small as possible, that is, minimising $\sum_{i=1}^{7} k_i^2$.

Some criteria are in conflict with each other, for example, maximising $\phi(t_1)$ requires high gains k_i and that is contrary to the goal of minimising the control effort. Moreover, maximising $\phi(t_1)$ is also conflicting with minimising the deviation of the sideslip, $\Delta\beta$.

Optimisation model

Overall, the optimisation model can be written as follows:

$$\begin{aligned}
\min F(\mathbf{k}) = (f_1(\mathbf{k}), f_2(\mathbf{k}), f_3(\mathbf{k}), f_4(\mathbf{k}), f_5(\mathbf{k}), f_6(\mathbf{k}), f_7(\mathbf{k}))\\
where\ f_1 = \sum_{i=1}^{7} k_i^2\\
f_2 = \Delta\beta = (\max(\mathbf{x}_4) - \min(\mathbf{x}_4))/2\\
f_3 = -\phi(t_1) = -\phi(1)\\
f_4 = -\phi(t_2) = -\phi(2.8)\\
f_5 = \lambda_s = r(D(1))\\
f_6 = \lambda_R = r(D(2))\\
f_7 = -\varepsilon_d = \cos(\arctan i(D(3), r(D(3))))\\
f_8 = -w_d = -real(D(4))
\end{aligned} \tag{6.11}$$

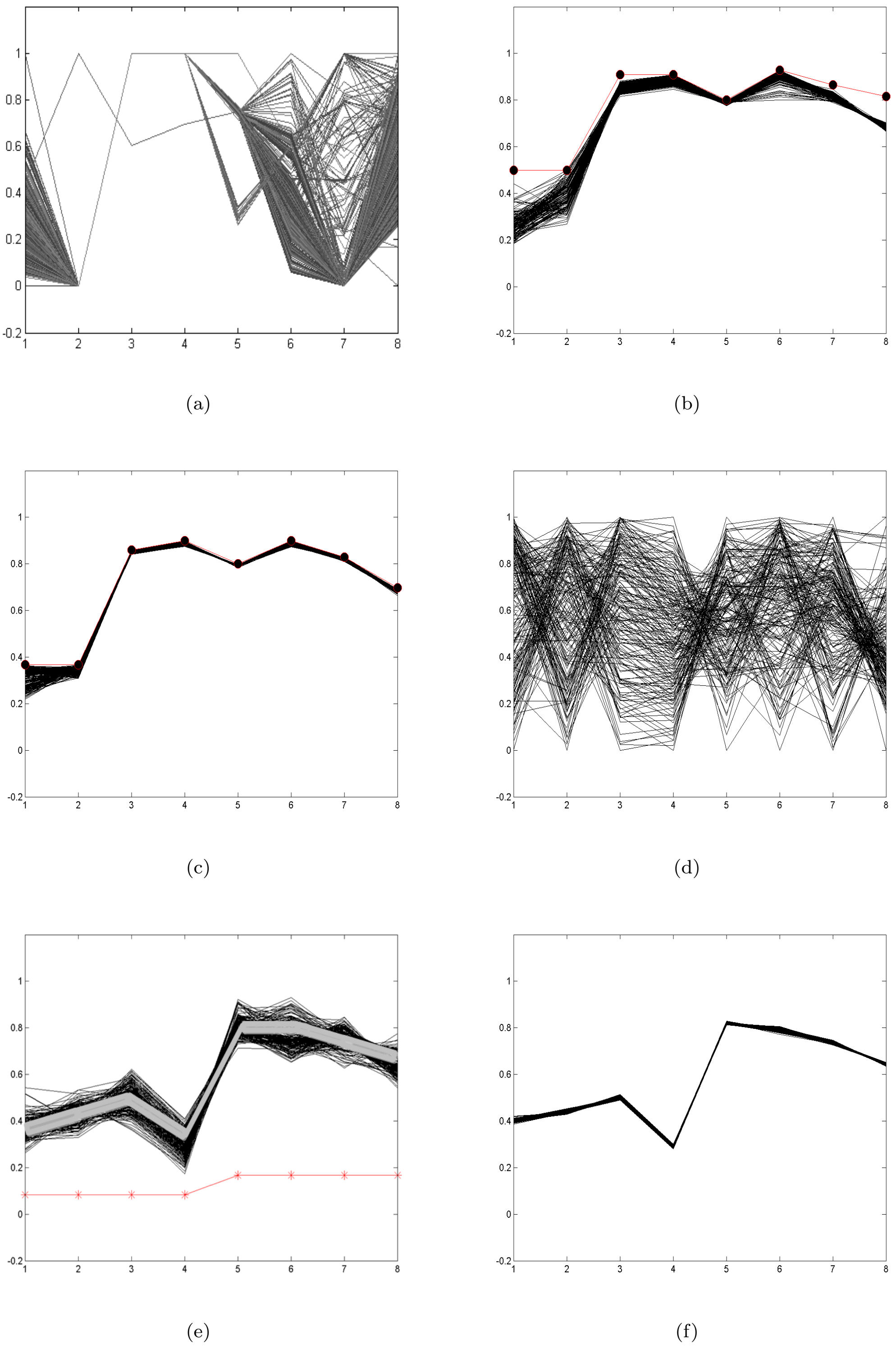
and $\mathbf{k} = (k_1, k_2, k_3, k_4, k_5, k_6, k_7)$ is the decision vector, $k_i \in (-1, 1)$. $\mathbf{x}_4$ is a vector formed by the fourth components (ϕ) of the state vector $\mathbf{x}$. D represents the eigenvalues of the matrix $A + BK$. K is the matrix formed by different $\mathbf{k}$ vectors. r and i represent the real and imaginary parts of a complex number.

6.4.2 Experiments

In this section we illustrate how to obtain a preferred solution for this design problem by using iPICEA-g. Note that to better visualise the results, we normalise the objectives values by two vectors $\mathbf{f}_{max} = (1.5, 1.5, -45, -180, 0.49, -2.5, 0.1, 6)$ and $\mathbf{f}_{min} = (0, 0, -540, -2160, -2, -20, -4, -32)$. The population size of candidate solutions and goal vectors are set as $N = 200$ and $N_g = 200$, respectively.

First, iPICEA-g is run without introducing any preference for 50 generations. The aim is to approximately learn the range of the objectives, i.e., the estimated *ideal* and *nadir* point, so as to inform the DM's initial preferences. The obtained solutions are shown in Figure 6.22(a).

Next, the DM specifies his aspiration levels for different performance criteria: the control effort (f_1) should be smaller than 0.75; the deviation of the overall sideslip angle (f_2) should be smaller than 0.75; according to the military specifications requirement $\phi(1s)$ is preferred to be larger than 90° and $\phi(2.8s)$ is preferred to be larger than 360°, i.e. ($f_3 \geq 90$, $f_4 \geq 360$); the spiral root (f_5) should be larger than 0.0005; the marginally stable roll damping root (f_6) should be smaller than -3.75; the damping ratio (f_7) should be larger than 0.45; the frequency (f_8) should be larger than 1. Therefore, $\mathbf{z}_R$ is set to $[0.75, 0.75, -90, -360, -0.005, -3.75, -0.45, -1]$, shown as $-\bullet-$ in Figure 6.22(b).



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 6.22: Interactive scenario for an aircraft control system design problem (viewed in objective-space).

Correspondingly, parameters $\mathbf{w}$ and θ are set as $w_1 = w_2 =, \cdots, = w_8 = 0.125$ and $\arccos(\sqrt{\frac{7}{8}}) = 0.3614$ radians. iPICEA-g is run for 50 more generations. A satisfied set of solutions is obtained, see Figure 6.22(b).

Next, assume that the DM is dissatisfied with the obtained solutions. To enable better stability of the aircraft, the DM wants to further minimise the control effort and the sideslip deviation and so the aspiration levels of f_1 and f_2 are modified to 0.55 and 0.55, respectively. Also, the DM decides to search for a larger frequency w_d to ensure a better stability. Thus, the aspiration level for f_8 is set to -5.5. Moreover, having obtained the exiting solutions, there is space to improve other objectives further. The DM then provides new aspiration levels for other objectives: bank angle at 1.0 second $f_3 \leq -115$, bank angle at 2.8 second $f_4 \leq -380$, roll damping $f_6 \leq -4.25$, dutch roll damping $f_7 \leq -0.6$. Overall, $\mathbf{z}_R$ is set as $(0.55, 0.55, -115, -380, -0.0050, \ -4.25, -0.6, -5.5)$. After running iPICEA-g for another 50 generations, a set of solutions is found as shown in Figure 6.22(c). Correspondingly, Figure 6.22(d) shows the objective values normalised by the estimated *ideal* and *nadir* points.

Next, the DM further specifies that the first four objectives are as half important as the last four objectives. Therefore, the weight vector in iPICEA-g is configured as $\mathbf{w} = (\frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$. Additionally, the spread range θ is set to $\frac{\pi}{18}$ radians. After running iPICEA-g for 50 more generations, a set of new solutions is found, see Figure 6.22(e).

Finally, assume that the DM is still dissatisfied with the obtained solutions. He brushes a set of solutions that are within region of interest. Based on the brushed solutions, the related parameters are configured as $\mathbf{w} = (0.1237, 0.1250, 0.1251, \ 0.1247, 0.1250, 0.1261, 0.1245, 0.1259)$, and $\theta = \frac{\pi}{60}$ radians. After running iPICEA-g for 50 more generations, a set of satisfactory solutions are found, see Figure 6.22(f). The DM is now able to choose a solution from this set.

6.5 Summary

Incorporation of decision-maker preferences is an important part of a real-world decision support system. However, current methods for preference-based evolutionary multi-objective approaches are unable to handle, comprehensively, the range of ways in which a decision-maker wishes to articulate his/her preferences. In this Chapter, we have presented, to the best of our knowledge, the first unified preference-based MOEA that is simultaneously able to handle decision-maker preferences specified either as aspirations or weights and that is also able to support multiple regions of interest. The effectiveness of iPICEA-g is demonstrated on a set of problems. Moreover, we also enhance the DM-friendliness by allowing preferences to be expressed via visually brushing in objective-space on a coordinate system. The main findings are summarised as follows:

(i) The effects of the three parameters $\mathbf{z}_R$, $\mathbf{w}$ and θ are sufficient to formulate the decision-maker preferences in different ways.

(ii) iPICEA-g can either incorporate the decision-maker preference expressed by aspiration levels or by weights. In both cases, iPICEA-g is able to find solutions that are better in the region of interest to the decision-maker, and that are also close to the Pareto optimal front.

(iii) The brushing technique allows decision-maker preferences to be expressed visually via drawing. It is illustrated to be effective. Since no direct elicitation of numerical values is required, a decision-maker may find it cognitively easier to use this method.

(iv) The performance of iPICEA-g is further examined on a real-world decision-making problem – aircraft control system design problem. Experimental results show that this unified approach iPICEA-g works well, and it enables the decision-maker to freely articulate his/her preferences in different ways.

Chapter 7

Conclusions

7.1 Key results

This research proposes a novel class of multi-objective evolutionary algorithm named the preference-inspired co-evolutionary algorithm (PICEA). PICEA co-evolves the population of candidate solutions with a family of decision-maker preferences during the optimisation process. Two instantiations of the PICEA, i.e., PICEA-g and PICEA-w, are proposed. PICEA-g takes goal vectors as preferences, and can handle many-objective problems successfully. PICEA-w takes weight vectors as preferences, and can effectively overcome the difficulties encountered by decomposition based algorithms. Additionally, sensitivities of parameters in PICEAs are studied. Lastly, a unified approach named iPICEA-g is proposed to support an *a priori* or *progressive* decision-making.

PICEA-g

Many-objective optimisation problems remain challenging in terms of obtaining a good and representative approximation of the Pareto optimal front. One of the main challenges identified for many-objective optimisation is the reduced ability of the Pareto-dominance relation in offering comparability between alternative solutions. This lack of comparability means that algorithms using Pareto-dominance struggle to drive the search towards the Pareto optimal front. PICEA-g co-evolves the usual population of candidate solutions with a family of decision-making preferences, formulated as goal vectors during the search. The incorporation of goal vectors enables incomparable solutions (in terms of the Pareto-dominance) to be comparable, thus creating selective pressure towards the Pareto optimal front.

During the co-evolution, candidate solutions act as predators that attempt to catch (i.e., dominate) as many goal vectors as possible. Goal vectors act as prey that attempt to avoid being caught (i.e., dominated) by candidate solutions. This scenario is realised by fitness assignment. Specifically, candidate solutions gain higher fitness by dominating more goal vectors and the fitness contribution of a goal vector to a candidate solution is shared between other solutions that also dominate this goal vector. Goal vectors gain higher fitness by being dominated by fewer candidate solutions. Both the candidate solutions and the goal vectors are co-evolved towards the Pareto optimal front. Key findings are:

(i) Compared with the four best-in-class MOEAs, that is, NSGA-II (Pareto-dominance class), ε-MOEA (modified Pareto-dominance class), MOEA/D (decomposition class) and HypE (indicator class), PICEA-g exhibits a better performance on many-objective benchmark problems, namely WFG.

(ii) An effective strategy named the *cutting plane* is proposed which can improve the performance of PICEA-g even further. This strategy utilises the knowledge of the *ideal* point, and enables PICEA-g to adaptively distribute the search effort towards different objectives appropriately. The effectiveness of this strategy is demonstrated on the WFG benchmark problems with up to seven objectives.

PICEA-w

Decomposition based algorithms transfer a MOP into a set of single objective problems that are determined by means of scalarising functions with different weights. They have been demonstrated to perform well when a suitable set of weights is provided. However, determining a good set of weights *a priori* for real-world problems is usually not straightforward due to a lack of knowledge of the underlying problem structure.

Adopting the concept of PICEA, a new decomposition based algorithm, PICEA-w, is proposed, which eliminates the need to specify appropriate weights before performing the optimisation. Specifically, weights are co-evolved with candidate solutions during the search process. The co-evolution enables suitable weights to be adaptively constructed along the optimisation process. Through rigorous empirical testing, we demonstrate the benefits of PICEA-w compared to other leading decomposition-based algorithms. The chosen test problems encompass the range of Pareto front geometries likely to be seen in practice, including simultaneous optimisation of up to ten conflicting objectives. Key findings are:

(i) PICEA-w is less sensitive to the Pareto front geometry, and performs well on many-objective problems. During the co-evolution, the employed weights also evolve towards an optimal distribution.

(ii) Decomposition based algorithms using pre-defined weights face difficulty on problems having a complex Pareto optimal front. The use of adaptive weights is helpful to solve this problem. However, adapting weights during the search would affect an algorithm's convergence performance.

Parameter sensitivity of PICEAs

It is known that performance of an algorithm is often affected by its associated parameter settings. We have performed both a global and a local sensitivity analysis on the parameters used in PICEA-g and PICEA-w. Key findings are:

(i) The performance of both PICEA-g and PICEA-w is controlled for the most part by a single parameter, i.e., the number of function evaluations. This indicates that parameterising PICEAs should be easier in practice for non-experts. A better performance can be obtained simply by lengthening the runtime of PICEAs, i.e., increasing the number of function evaluations.

(ii) Amongst the remaining parameters, the number of preferences is an important one. PICEAs can deliver a better performance if more preferences are co-evolved during the search. In terms of the genetic operator configurations, it is found that, for the recombination operator (SBX), parameter η_c is more dominant, and it is suggested that a small value of η_c is used. For the mutation operator (PM), a combination of a small p_m and a large η_m produces good results, and are suggested for use.

iPICEA-g

iPICEA-g is a unified approach for *a priori* or *progressive* evolutionary multi-objective optimisation and decision-making. It extends the existing algorithm PICEA-g by co-evolving candidate solutions with goal vectors that are exclusively generated in the regions of interest to the decision-maker. iPICEA-g distinguishes itself from other preference-based MOEAs by its ability to cater simultaneously in different ways, i.e., aspirations, weights or via visually brushing, which a decision-maker could specify his/her preferences.

The performance of iPICEA-g has been assessed on a set of benchmark problems as well as a real world problem. Experimental results show that iPICEA-g works effectively while incorporating different types of preferences. Moreover, by incorporation of the brushing technique, iPICEA-g allows a decision-maker to specify his/her preferences without using numerical values but simply by drawing in objective-space, which might prove cognitively easier for a decision-maker to use.

7.2 Future work

It is admitted that there are a number of ways in which the central contributions of this research are limited. These limitations suggest where more studies are needed in future.

Testing PICEAs on other problems

One of the main limitations is that all the findings in this research are based on real-parameter function optimisation problems. It would be valuable to assess the performance of PICEAs on other problem types, such as many-objective combinatorial problems, the travelling salesman problem and the graph partitioning problem, and also, crucially, real-world problems in future.

Constraints, dynamism and uncertainty handling techniques

Real-world problems are often involved with constraints, dynamism and uncertainty. In the literature, a number of methods have been proposed to handle problems having these properties (Jin and Branke, 2005; Mallipeddi and Suganthan, 2010; Mezura-Montes and Coello Coello, 2011; Cruz et al., 2011; Nguyen et al., 2012). However, none of these techniques has been investigated specifically for PICEAs. Further study on these important aspects should be conducted.

Estimation of *ideal* and *nadir* point

The importance of the *ideal* and *nadir* point in multi-objective optimisation has been mentioned in a number studies (Miettinen, 1999; Deb et al., 2006). These two vectors can be used to normalise the objectives such that multi-objective evolutionary algorithms can be used more reliably. Moreover, these two vectors are also required in multi-criteria decision-making methodologies. Additionally, they are helpful as an aid in visualising Pareto solutions.

To estimate the *ideal* point, one can perform single objective optimisation for each objective separately. However, this might not be effective since in some cases the single objective optimisation is also difficult. To estimate the *nadir* point, various methods have been provided such as *surface–to–nadir*, *edge–to–nadir* and *extreme–to–nadir* (Deb and Miettinen, 2009). However, these methods all have limitations, such as computationally expense. Thus, developing an effective strategy for estimating these two vectors would be valuable.

Adaptive parameter configurations

As previously mentioned, the performance of an algorithm is often affected by its associated parameter configurations. An appropriate parameter configuration is often problem-dependent. For example, given a fixed number of function evaluations, it is not easy to strike an effective balance between population size and the number of generations, for example, the larger the population size, the beneficial dynamics of evolution are curtailed. In most of the existing studies, population size is set as a constant value during the whole search. However, this has been demonstrated not to be the best choice (Lobo, 2011; Cook and Tauritz, 2010).

Therefore, developing adaptive approaches that can effectively configure parameters appropriately for an algorithm is valuable. Particularly, this will greatly facilitate the use of algorithms for non-experts in practice since the procedure of tuning parameter settings can be eliminated.

Genetic operator design

In order to improve the performance of MOEAs, most studies focus on designing a better selection mechanism. Undoubtedly, a good selection mechanism is important as it can provide good parent candidates for the generation of offspring. However, in addition to the selection mechanism, genetic operators are also crucial in terms of producing fruitful offspring. Therefore, it is worth studying how effective genetic operators can be designed. As a start, it is helpful to first assess the performance of different genetic operators, such as the one-point crossover, SBX, SPX, DE and PM, etc. This analysis might provide some insights on designing effective genetic operators.

Mating restriction

The use of mating restriction has been demonstrated to be beneficial in generating fruitful offspring (Fonseca and Fleming, 1995a). One of the frequently used mating restriction strategies is to recombine neighbouring solutions (Ishibuchi and Shibata, 2003, 2004). Neighbourhood can be measured by Euclidean distance, Manhattan distance and the like. However, the effectiveness of this method is highly dependent on a defined neighbourhood size. Few approaches are able to handle this issue effectively. Therefore, it is worth investigating how to determine a suitable neighbourhood size for different problems, or developing other effective mating restriction strategies.

Multi-criteria decision-making

We have proposed a unified approach iPICEA-g for *a priori* and *progressive* decision-making. However, it is known that decision-making is often a group rather than individual activity. Therefore, it would be useful to develop a methodology to support a group decision-making process. Moreover, since the decision-maker preferences are often expressed in fuzzy linguistic terms (Jin and Sendhoff, 2002; Rachmawati and Srinivasan, 2010), it would also be interesting to investigate how fuzzy preferences can be incorporated into iPICEA-g.

Appendix A

Multi-objective test benchmarks

A.1 ZDT test

General definitions of ZDT test problems are

$$\begin{aligned} f_1 &= x_1 \\ f_2 &= g * h \end{aligned} \tag{A.1}$$

where g and h are functions of the decision vector $\mathbf{x}$.

- ZDT1

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \cdots, x_n) &= 1 + 9\sum_{i=1}^{k} z_i/k \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{A.2}$$

  where $\forall x_i \in [0, 1]$. The Pareto optimal front is convex (see Figure A.1), and is formed with $g(x_2, \cdots, x_m) = 1$. The solutions are uniformly distributed in the search space:

- ZDT2

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \cdots, x_n) &= 1 + 9\sum_{i=1}^{k} z_i/k \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned} \tag{A.3}$$

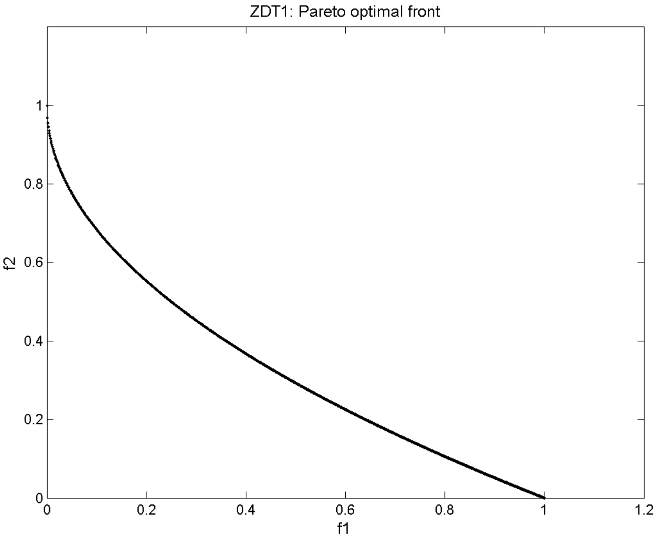


Figure A.1: The Pareto optimal front of ZDT1.

where $\forall x_i \in [0,1]$. The Pareto optimal front is concave (see Figure A.2), and is formed with $g(x_2,\cdots,x_m)=1$. The solutions are uniformly distributed in the search space.

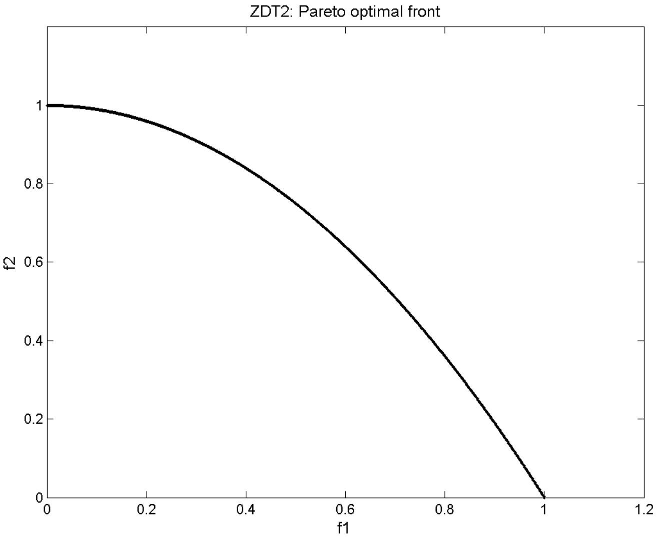


Figure A.2: Pareto optimal front of ZDT2.

- ZDT3

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \cdots, x_n) &= 1 + 9 \sum_{i=1}^{k} z_i/k \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_i) \end{aligned} \tag{A.4}$$

where $\forall x_i \in [0, 1]$. The Pareto optimal front is formed with $g(x_2, \cdots, x_m) = 1$. The Pareto optimal front is composed of five disconnected convex fronts as shown in Figure A.3. Although the Pareto optimal front is disconnected, there is no discontinuity in the search space.

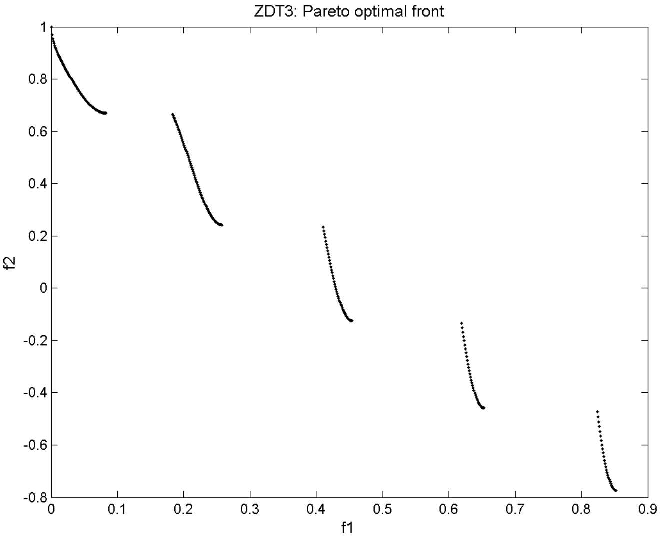


FIGURE A.3: Pareto optimal front of ZDT3.

- ZDT4

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \cdots, x_n) &= 1 + 10(m-1) + \sum_{i=2}^{m} (x_i^2 - 19\cos(4\pi x_i)) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \tag{A.5}$$

where $x_1 \in [0, 1]$ and $x_i \in [-5, 5], i = 2, \cdots, n$. ZDT4 is multi-modal which has 21^9 local Pareto optimal fronts. The Pareto optimal front (see Figure A.4) is formed with $g(x_2, \cdots, x_m) = 1$.

- ZDT6

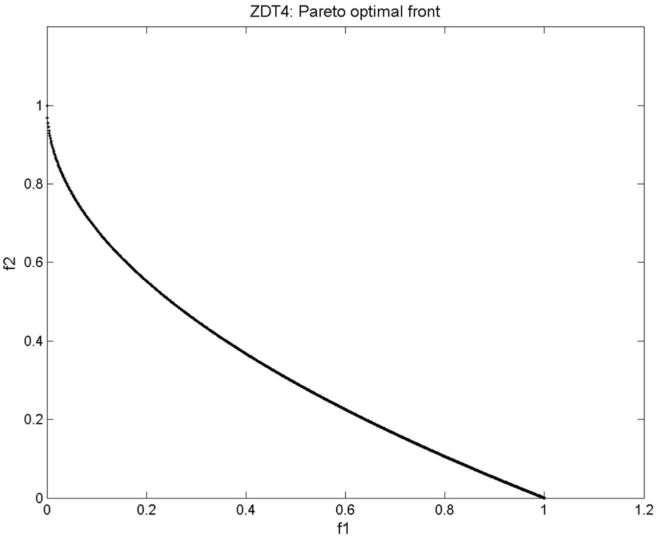


FIGURE A.4: Pareto optimal front of ZDT4.

$$
\begin{aligned}
f_1(x_1) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\
g(x_2, \cdots, x_n) &= 1 + 9\Big(\sum_{i=2}^{m} x_i\Big)^{0.25} \\
h(f_1, g) &= 1 - (f_1/g)^2
\end{aligned}
\tag{A.6}
$$

where $x_1 \in [0, 1]$ and $x_i \in [-5, 5], i = 2, \cdots, n$. Both the Pareto optimal front and the search space are non-uniform. The Pareto optimal front is convex shown as Figure A.5 and is formed with $g(x_2, \cdots, x_m) = 1$.

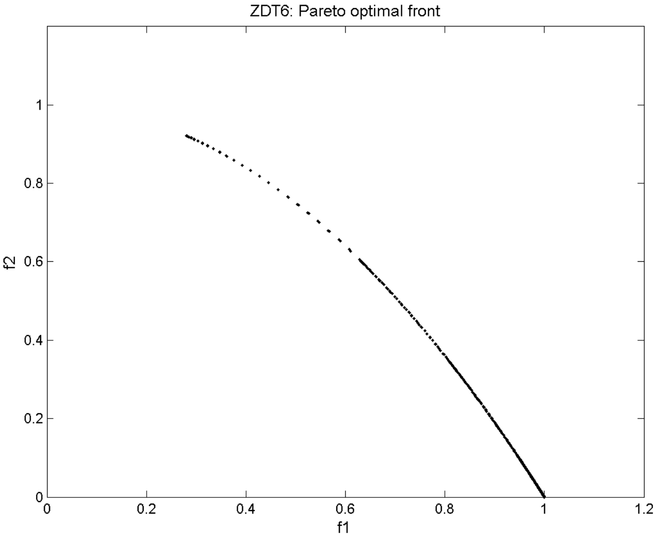


FIGURE A.5: Pareto optimal front of ZDT6.

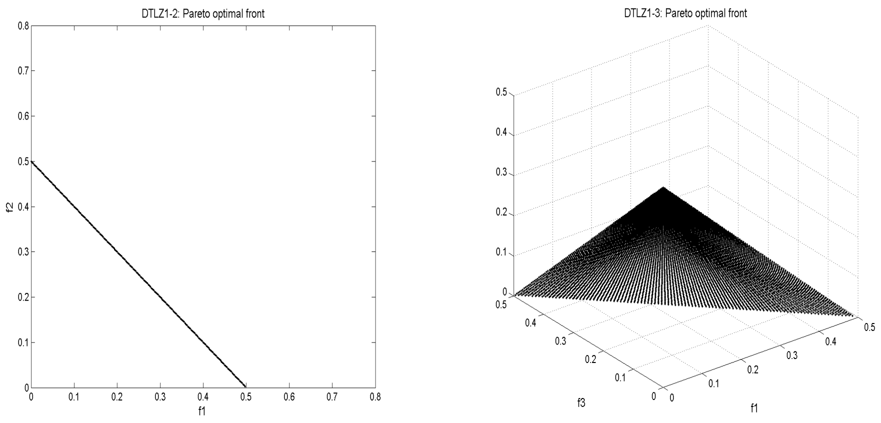
Note that ZDT5 is not listed here as the scope of this thesis is on real-parameter optimization. ZDT5 is a binary coded problem.

A.2 DTLZ test

- DTLZ1

$$\begin{aligned}
f_1 &= (1+g)\frac{1}{2}x_1x_2\cdots x_{M-1}\\
f_2 &= (1+g)\frac{1}{2}x_1x_2\cdots(1-x_{M-1})\\
&\vdots\\
f_{M-1} &= (1+g)\frac{1}{2}x_1(1-x_2)\\
f_M &= (1+g)\frac{1}{2}(1-x_1)\\
g &= 100(|X_M| + \sum_{x_i\in X_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5)))
\end{aligned} \tag{A.7}$$

where $|X_M| = k$ and $X_M = [x_M, x_{M+1}, \cdots, x_N]$. Parameter k is suggested to be 5 in (Deb et al., 2002b). The Pareto optimal front is a linear hyper-plane that satisfies $\sum_{i=1}^{M} f_i^* = 0.5$ as shown in Figure A.6. The Pareto optimal solutions corresponds to $x_i^* = 0.5, \quad x_i^* \in X_M$. DTLZ1 is a multi-modal problem which has $11^k - 1$ local optimal fronts.



(a) DTLZ1-2

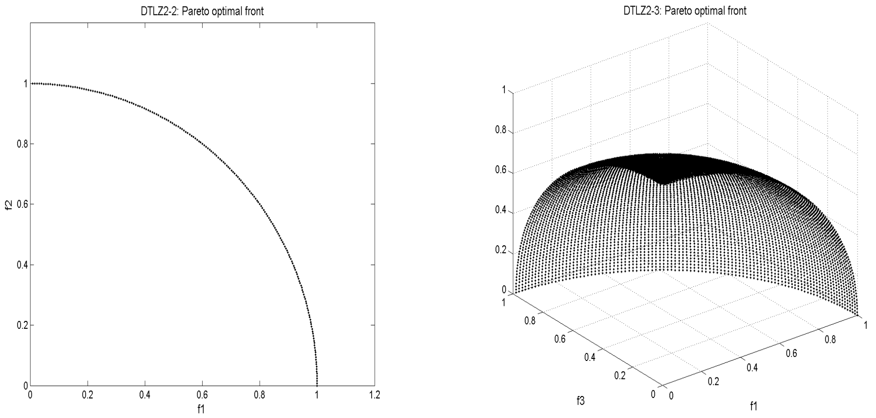
(b) DTLZ1-3

FIGURE A.6: Pareto optimal fronts of DTLZ1-2 and DTLZ1-3.

- DTLZ2

$$\begin{aligned}
f_1 &= (1+g)\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\cos(x_{M-1}\frac{\pi}{2})\\
f_2 &= (1+g)\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\sin(x_{M-1}\frac{\pi}{2})\\
&\vdots\\
f_{M-1} &= (1+g)\cos(x_1\frac{\pi}{2})\sin(x_2\frac{\pi}{2})\\
f_M &= (1+g)\sin(x_1\frac{\pi}{2})\\
g &= \sum_{x_i\in X_M}(x_i-0.5)^2
\end{aligned} \tag{A.8}$$

where $k = |X_M| = 10$ is suggested (Deb et al., 2002b). The Pareto optimal front satisfies $\sum_{i=1}^{M}(f_i^*)^2 = 1$ as shown in Figure A.7. The Pareto optimal solutions corresponds to $x_i^* = 0.5, \quad x_i^* \in X_M$.



(a) DTLZ2-2

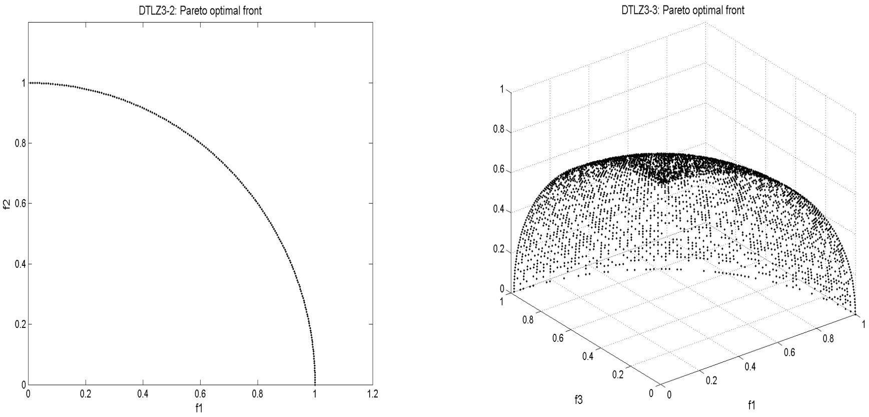
(b) DTLZ2-3

Figure A.7: Pareto optimal fronts of DTLZ2-2 and DTLZ2-3.

- DTLZ3

$$\begin{aligned}
f_1 &= (1+g)\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\cos(x_{M-1}\frac{\pi}{2})\\
f_2 &= (1+g)\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\sin(x_{M-1}\frac{\pi}{2})\\
&\vdots\\
f_{M-1} &= (1+g)\cos(x_1\frac{\pi}{2})\sin(x_2\frac{\pi}{2})\\
f_M &= (1+g)\sin(x_1\frac{\pi}{2})\\
g &= 100(|X_M| + \sum_{x_i\in X_M}(x_i-0.5)^2 - cos(20\pi(x_i-0.5)))
\end{aligned} \tag{A.9}$$

The Pareto optimal front of DTLZ3 is concave and satisfies $\sum_{i=1}^{M}(f_i^*)^2 = 1$ as shown in Figure A.8. The Pareto optimal solutions corresponds to $x_i^* = 0.5, \quad x_i^* \in X_M$. DTLZ3 is also multi-modal that has $3^k - 1$ local optimal fronts.



(a) DTLZ3-2

(b) DTLZ3-3

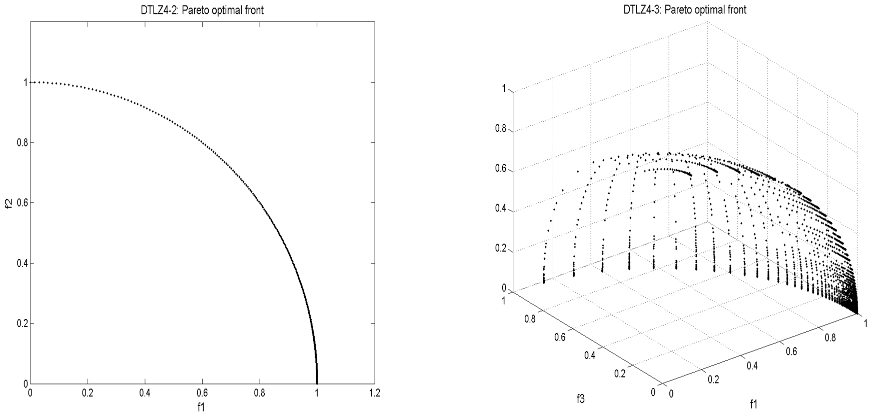
FIGURE A.8: Pareto optimal fronts of DTLZ3-2 and DTLZ3-3.

- DTLZ4

$$\begin{aligned}
f_1 &= (1+g)\cos(x_1^\alpha\frac{\pi}{2})\cos(x_2^\alpha\frac{\pi}{2})\cdots\cos(x_{M-2}^\alpha\frac{\pi}{2})\cos(x_{M-1}^\alpha\frac{\pi}{2})\\
f_2 &= (1+g)\cos(x_1^\alpha\frac{\pi}{2})\cos(x_2^\alpha\frac{\pi}{2})\cdots\cos(x_{M-2}^\alpha\frac{\pi}{2})\sin(x_{M-1}^\alpha\frac{\pi}{2})\\
&\vdots\\
f_{M-1} &= (1+g)\cos(x_1^\alpha\frac{\pi}{2})\sin(x_2^\alpha\frac{\pi}{2})\\
f_M &= (1+g)\sin(x_1^\alpha\frac{\pi}{2})\\
g &= \sum_{x_i\in X_M}(x_i-0.5)^2
\end{aligned} \tag{A.10}$$

DTLZ4 is constructed by substitute x_i in DTLZ2 with x_i^α. α is set as 100. The Pareto optimal front of DTLZ4 satisfies $\sum_{i=1}^{M}(f_i^*)^2 = 1$, however, the its distribution is not uniform as shown in Figure A.9. The Pareto optimal solutions corresponds to $x_i^* = 0.5, \quad x_i^* \in X_M$.

- DTLZ5



(a) DTLZ4-2

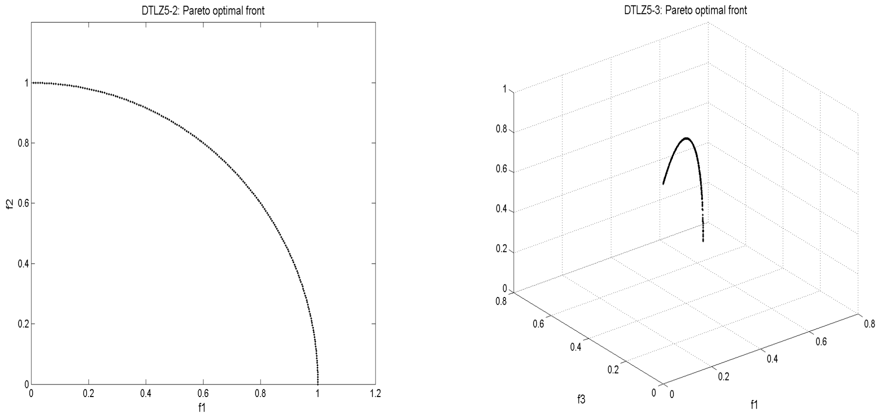
(b) DTLZ4-3

FIGURE A.9: Pareto optimal fronts of DTLZ4-2 and DTLZ4-3.

$$
\begin{aligned}
f_1 &= (1+g)\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\cos(\theta_{M-1}\frac{\pi}{2})\\
f_2 &= (1+g)\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\sin(\theta_{M-1}\frac{\pi}{2})\\
&\vdots\\
f_{M-1} &= (1+g)\cos(\theta_1\frac{\pi}{2})\sin(\theta_2\frac{\pi}{2})\\
f_M &= (1+g)\sin(\theta_1\frac{\pi}{2})\\
g &= \sum_{x_i\in X_M}(x_i-0.5)^2\\
\theta_i &= \frac{\pi}{4(1+g)}(1+2gx_i),\quad i=1,2,\cdots,M-1\\
\theta_1 &= x_1\frac{\pi}{2}
\end{aligned}
\tag{A.11}
$$

where $|k = X_M = 10|$ is suggested. The Pareto optimal front is a degenerated concave curve shown as in Figure A.10, satisfying $\sum_{i=1}^{M}(f_i^*)^2 = 1$, correspondingly, the Pareto optimal solutions satisfy $x_i^* = 0.5, \quad x_i^* \in X_M$.

- DTLZ6



(a) DTLZ5-2

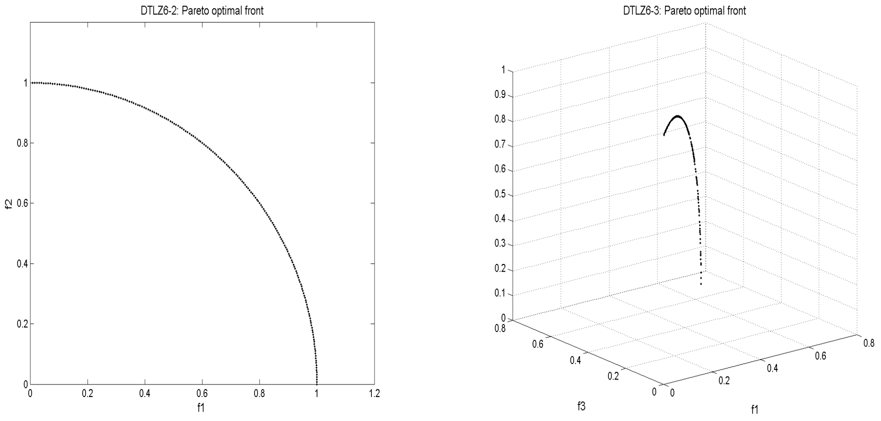
(b) DTLZ5-3

Figure A.10: Pareto optimal fronts of DTLZ5-2 and DTLZ5-3.

$$
\begin{aligned}
f_1 &= (1+g)\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\cos(\theta_{M-1}\frac{\pi}{2}) \\
f_2 &= (1+g)\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\sin(\theta_{M-1}\frac{\pi}{2}) \\
&\vdots \\
f_{M-1} &= (1+g)\cos(\theta_1\frac{\pi}{2})\sin(\theta_2\frac{\pi}{2}) \\
f_M &= (1+g)\sin(\theta_1\frac{\pi}{2}) \\
g &= \sum_{x_i \in X_M} x_i^{0.1} \\
\theta_i &= \frac{\pi}{4(1+g)}(1+2gx_i), \quad i = 1, 2, \cdots, M-1 \\
\theta_1 &= x_1\frac{\pi}{2}
\end{aligned}
\tag{A.12}
$$

where $k = |X_M| = 10|$ is suggested. The Pareto optimal front is the same as DTLZ5, see Figure A.11. The Pareto optimal solutions correspond to $x_i^* = 0, \; x_i^* \in X_M$.

- DTLZ7



(a) DTLZ6-2

(b) DTLZ6-3

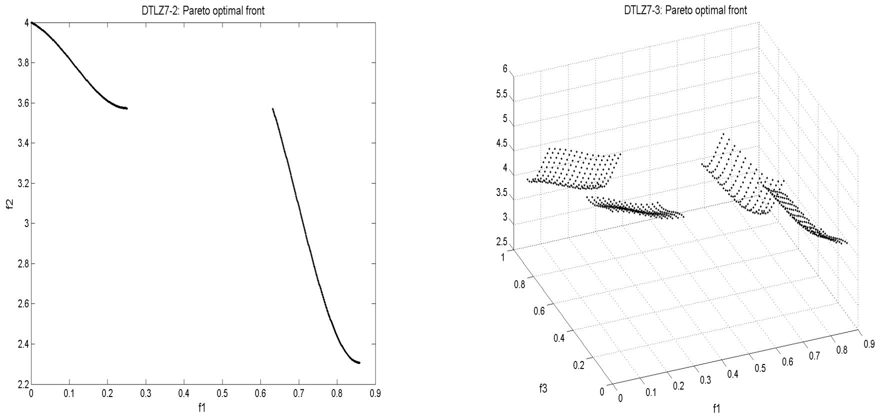
Figure A.11: Pareto optimal fronts of DTLZ6-2 and DTLZ6-3.

$$\begin{aligned} f_1 &= x_1 \\ f_2 &= x_2 \\ &\vdots \\ f_{M-1} &= x_{M-1} \\ f_M &= (1+g)(M - \sum_{i=1}^{M-1}(\frac{f_i}{1+g}(1+\sin(3\pi f_i)))) \\ g &= 1 + \frac{9}{|X_M|}\sum_{x_i \in X_M} x_i \end{aligned} \tag{A.13}$$

where $|k = X_M = 20|$ is suggested. The Pareto optimal front has 2^{M-1} disconnected regions as shown in Figure A.12. The Pareto optimal solutions correspond to $x_i^* = 0, \quad x_i^* \in X_M$.

A.3 WFG test

The general definition of WFG test suite can be described as



(a) DTLZ7-2

(b) DTLZ7-3

FIGURE A.12: Pareto optimal fronts of DTLZ7-2 and DTLZ7-3.

$$
\begin{aligned}
&\mathit{Given}\ \mathbf{z} = \{z_1, \cdots, z_k, z_{k+1}, \cdots, z_n\} \\
&\qquad \mathit{Minimise}\ f_{m=1:M}(\mathbf{x}) = Dx_m + S_m h_m(x_1, \cdots, x_{M-1}) \\
&\qquad\qquad \mathit{where}\ \mathbf{x} = \{x_1, \cdots, x_M\} \\
&\qquad\qquad\qquad = \{\max(t^p_M, A_1)(t^p_1 - 0.5) + 0.5, \cdots, \\
&\qquad\qquad\qquad \max(t^p_{M-1}, A_1)(t^p_{M-1} - 0.5) + 0.5, t^p_M\} \\
&\qquad\qquad \mathbf{t^p} = \{t^p_1, \cdots, t^p_M\} \\
&\qquad\qquad \mathbf{z}_{[0,1]} = \{z_{1,[0,1]}, \cdots, z_{n,[0,1]}\} \\
&\qquad\qquad\qquad = \{\frac{z_1}{z_{1,max}}, \cdots, \frac{z_n}{z_{n,max}}\}
\end{aligned}
\tag{A.14}
$$

where $\mathbf{z}$ is a decision vector, consisting $n = k + l$ decision variables. k is the number of position parameters and l is the number of distance parameters. h_m is the shape functions and S_m is the scaling constraints. $\mathbf{t^p}$ is the transition vector.

Priori to describing the WFG test problems, we introduce the *shape* functions, and the *transformation* functions used in the test suite.

The *shape* functions

The shape functions are introduced as follows, in all cases, $x_1, \cdots, x_{M-1} \in [0, 1]$, and A, α and β are constants:

- Linear

$$\begin{aligned} linear_1(x_1, \cdots, x_{M-1}) &= \prod_{i=1}^{M-1} x_i \\ linear_{m=2,\cdots,M-1}(x_1, \cdots, x_{M-1}) &= (\prod_{i=1}^{M-1} x_i)(1 - x_{M-m+1}) \\ linear_M(x_1, \cdots, x_{M-1}) &= 1 - x_i \end{aligned} \tag{A.15}$$

The Pareto front is a hyper-plane ($\sum_{m=1}^{M} h_m = 1$) when $h_{m=1,\cdots,M} = linear_m$

- Convex

$$\begin{aligned} convex_1(x_1, \cdots, x_{M-1}) &= \prod_{i=1}^{M-1} (1 - \cos(x_i \frac{\pi}{2})) \\ convex_{m=2,\cdots,M-1}(x_1, \cdots, x_{M-1}) &= \prod_{i=1}^{M-1} (1 - \cos(x_i \frac{\pi}{2}))(1 - \sin(x_{M-m+1} \frac{\pi}{2})) \\ convex_M(x_1, \cdots, x_{M-1}) &= 1 - \sin(x_1 \frac{\pi}{2}) \end{aligned} \tag{A.16}$$

The Pareto front is convex when $h_{m=1,\cdots,M} = convex_m$.

- Concave

$$\begin{aligned} concave_1(x_1, \cdots, x_{M-1}) &= \prod_{i=1}^{M-1} \sin(\_i \frac{\pi}{2}) \\ concave_{m=2,\cdots,M-1}(x_1, \cdots, x_{M-1}) &= \prod_{i=1}^{M-1} \sin(\_i \frac{\pi}{2}) \cos(x_{M-m+1} \frac{\pi}{2})) \\ concave_M(x_1, \cdots, x_{M-1}) &= \cos(x_1 \frac{\pi}{2} \end{aligned} \tag{A.17}$$

The Pareto front is concave when $h_{m=1,\cdots,M} = concave_m$, having $\sum_{m=1}^{M} (h_m)^2 = 1$.

- Mixed convex/concave ($\alpha > 0$, $A \in \{1, 2, \cdots\}$)

$$mixed_M(x_1, \cdots, x_{M-1}) = (1 - x_1 - \frac{\cos(2A\pi x_1) + \pi/2}{2A\pi})^{\alpha} \tag{A.18}$$

Parameter A is to control the number of concave or convex segments. If $\alpha > 1$, the overall front is convex; if $\alpha < 1$, the overall shape is concave; and if $\alpha = 1$ the overall shape is linear.

- Disconnected ($\alpha > 0$, $\beta > 0$ $A \in \{1, 2, \cdots\}$)

$$discon_M(x_1, \cdots, x_{M-1}) = (1 - x_1^{\alpha} - \cos(A(x_1)^{\beta}\pi)^2 \tag{A.19}$$

Similarly, A controls the number of disconnected segments. $\alpha > 1$ leads to a convex overall shape; $\alpha < 1$ leads to a concave shape; and $\alpha = 1$ leads to a linear shape. β impacts the locations of the disconnection.

The *transformation* functions

The transformation functions are listed below. In these functions y and $[y_1, \cdots, y_{|\mathbf{y}|}]$ is in the domain $[0, 1]$. A, B, C, α and β are constants.

- Bias: Polynomial ($\alpha > 0, \alpha \neq 0$)

$$b\_poly(y, \alpha) = y^{\alpha} \tag{A.20}$$

when $\alpha > 1$, y is bias towards 0 else towards 1.

- Bias: Flat region ($A, B, C \in (0, 1), B < C, B = 0 \rightarrow A = 0 \wedge C \neq 1 \rightarrow A = 1 \wedge B \neq 1$)

$$\begin{aligned} b\_flat(y, A, B, C) = A + \min(0, \lfloor y - B \rfloor)\frac{A(B-y)}{B} - \\ \min(0, \lfloor C - y \rfloor)\frac{(1-A)(y-C)}{1-C} \end{aligned} \tag{A.21}$$

Values of y between B and C (the area of flat region) are all mapped to A.

- Bias: Parameter dependent ($A \in (0, 1), 0 < B < C$)

$$\begin{aligned} b\_param(y, \mathbf{y}', A, B, C) &= y^{B+(C-B)v(u(\mathbf{y}'))} \\ v(u(\mathbf{y}')) &= A - (1 - 2u(\mathbf{y}'))|\lfloor 0.5 - u(\mathbf{y}') \rfloor + A| \end{aligned} \tag{A.22}$$

A, B, C and $\mathbf{y}'$ determine the degree to which y is biased by being raised to an associated power: values of $u(\mathbf{y}') \in [0, 0.5]$ are mapped linearly onto $[B, B + (C - B)A]$, and values of $u(\mathbf{y}') \in [0.5, 1]$ are mapped linearly onto $[B + (C - B)A, C]$.

- Shift: Linear ($A \in (0, 1)$)

$$s\_linear(y, A) = \frac{|y - A|}{|\lfloor A - y \rfloor + A|} \tag{A.23}$$

A is the value for which y is mapped to 0.

- Shift: Deceptive ($A \in (0, 1)$, $0 < B \ll 1, 0 < C \ll 1, A - B > 0$, $A + B < 1$)

$$\begin{aligned} s\_decept(y, A, B, C) = 1 + (|y - A| - B)\times \\ (\frac{\lfloor y - A + B \rfloor(1 - C + \frac{A-B}{B})}{A - B} + \frac{\lfloor A + B - y \rfloor(1 - C + \frac{1-A-B}{B})}{1 - A - B} + \frac{1}{B}) \end{aligned} \tag{A.24}$$

A is the value at which y is mapped to 0, and the global minimum of the transformation. B is the "aperture" size of the well/basin leading to the global minimum at A, and C is the value of the deceptive minima (there are always two deceptive minima).

- Shift: Multi-modal ($A \in \{1, 2, \cdots\}$), $B \geq 0$, $(4A+2)\pi \geq 4B, C \in (0,1)$)

$$s\_multi(y, A, B, C) = \frac{1 + \cos[(4A+2)\pi(0.5 - \frac{|y-C|}{2(\lfloor C-y \rfloor + C)})] + 4B(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)})^2}{B+2} \tag{A.25}$$

A controls the number of minima, B controls the magnitude of the "hill sizes" of the multi-modality, and C is the value for which y is mapped to 0. When $B = 1, 2A+1$ values of y (one at C) are mapped to 0, and when $B \neq 0$, there are $2A$ local minima, and one global minimum at C. Larger values of A smaller values of B create more difficult problems.

- Reduction: Weighted sum ($|\mathbf{w}| = \mathbf{y}$, $\forall w_i > 0$, $i = 1, 2, \cdots, |\mathbf{y}|$)

$$r\_sum(\mathbf{w}, \mathbf{y}) = \frac{\sum_{i=1}^{|\mathbf{y}|} w_i y_i}{\sum_{i=1}^{|\mathbf{y}|} w_i} \tag{A.26}$$

Be varying the constants of the weight vector $\mathbf{w}$, Evolutionary algorithms are forced to treat parameters differently.

- Reduction: Non-separable ($A \in \{1, \cdots, |\mathbf{y}|\}$, $\mathbf{y} mod A = 0$)

$$r\_nonsep(\mathbf{y}, A) = \frac{\sum_{i=1}^{|\mathbf{y}|}(y_j + \sum_{k=0}^{A-2} |y_j - y_{(1+j+k) mod \mathbf{y}}|)}{\frac{\mathbf{y}}{A}\lceil A/2 \rceil (1 + 2A - 2\lceil A/2 \rceil)} \tag{A.27}$$

The degree of non-separability is controlled by A, $r\_nonsep(\mathbf{y}, \mathbf{1}) = r\_sum(\mathbf{y}, \{1, 1, \cdots, 1\})$.

The standard WFG test suite

Prior to introducing the format of the WFG test suite, we describe some general parameters for all the WFG test problems. For any transition vector $\mathbf{t}^i$, we let $\mathbf{y} = \mathbf{t}^{i-1}$. For $\mathbf{t}^1$, let $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \cdots, z_n/(2n)\}$. The domain of x_i is $[0, 2i]$, where $i = 1, 2, \cdots, n$. The constants are set as follows:

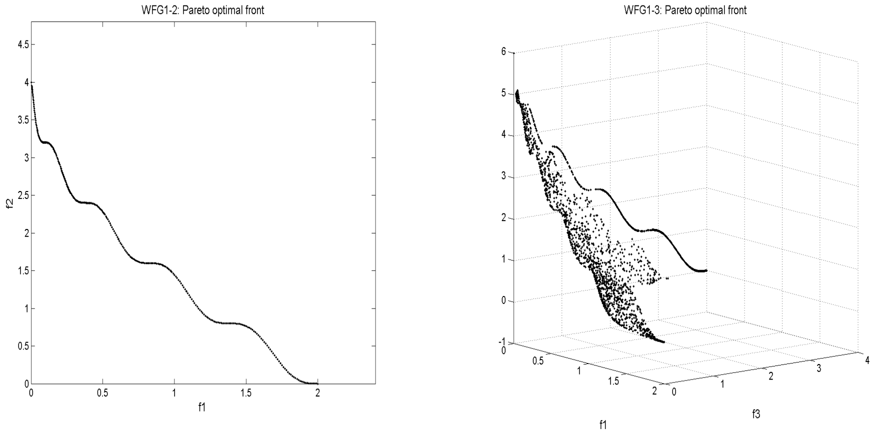
$$\begin{aligned} S_{m=1,\cdots,M} &= 2m \\ D &= 1 \\ A_1 &= 1 \end{aligned} \tag{A.28}$$

$A_{2:M-1}$ is set to 0 for WFG3, and 1 for the other problems. The settings for $S_{1:M}$ ensures Pareto optimal fronts have dissimilar trade-off magnitudes, and the settings for $A_{1:M-1}$ ensures Pareto optimal fronts are not degenerated, except for WFG3, which has a one dimensional Pareto optimal front.

- WFG1

$$\begin{aligned}
h_{m=1,\cdots,M-1} &= convex\_m \\
h_M &= mixed\_M\ (with\ \alpha = 1\ and\ A = 5) \\
t^1_{i=1:k} &= y_i \\
t^1_{i=k+1:n} &= s\_linear(y_i, 0.35) \\
t^2_{i=1:k} &= y_i \\
t^2_{i=k+1:n} &= b\_flat(y_i, 0.8, 0.75, 0.85) \\
t^3_{i=1:n} &= b\_poly(y_i, 0.02) \\
t^4_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \\
&\quad \{2((i-1)k/(M-1)+1), \cdots, 2ik/(M-1)\}) \\
t^4_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{2(k+1), \cdots, 2n\})
\end{aligned} \tag{A.29}$$

Pareto optimal fronts of 2 and 3-objective WFG2 are show in Figure A.13.



(a) WFG1-2

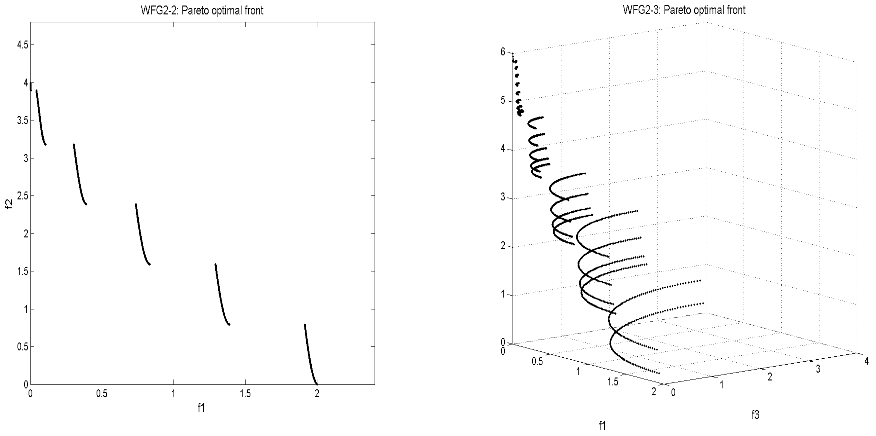
(b) WFG1-3

Figure A.13: Pareto optimal fronts of WFG1-2 and WFG1-3.

- WFG2

$$
\begin{aligned}
h_{m=1,\cdots,M-1} &= convex\_m \\
h_M &= disc\_M(with\alpha = \beta = 1 and A = 5) \\
\mathbf{t}^1 & \ is\ the\ same\ as\ WFG1\ (Linear\ shift) \\
t^2_{i=1:k} &= y_i \\
t^2_{i=k+1:k+l/2} &= r\_nonsep(\{y_{k+2(i-k)-1}, y_{k+2(i-k)}\}, 2) \\
t^3_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\
t^3_M &= r\_sum(\{y_{k+1}, \cdots, y_{k+l/2}\}, \{1, \cdots, 1\})
\end{aligned}
\tag{A.30}
$$

Pareto optimal fronts of 2 and 3-objective WFG2 are show in Figure A.14.



(a) WFG2-2

(b) WFG2-3

FIGURE A.14: Pareto optimal fronts of WFG2-2 and WFG2-3.

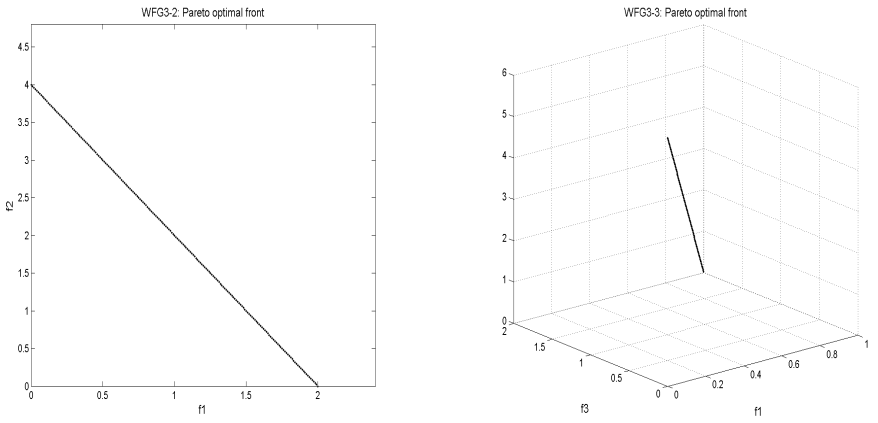
- WFG3

$$
\begin{aligned}
h_{m=1,\cdots,M} &= linear_m \quad (degenerate) \\
\mathbf{t}^{1:3} & \ is\ \ the\ same\ as\ \mathbf{t}^{1:3}\ in\ WFG2
\end{aligned}
\tag{A.31}
$$

Pareto optimal fronts of 2 and 3-objective WFG3 are show in Figure A.15.

- WFG4

$$
\begin{aligned}
h_{m=1,\cdots,M} &= concave\_m \\
t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\
t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\
t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\})
\end{aligned}
\tag{A.32}
$$

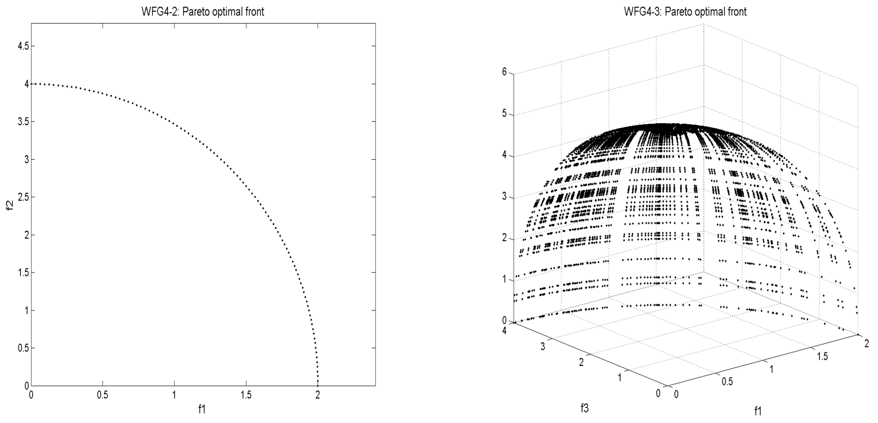


(a) WFG3-2

(b) WFG3-3

Figure A.15: Pareto optimal fronts of WFG3-2 and WFG3-3.

Pareto optimal fronts of 2 and 3-objective WFG4 are show in Figure A.16.



(a) WFG4-2

(b) WFG4-3

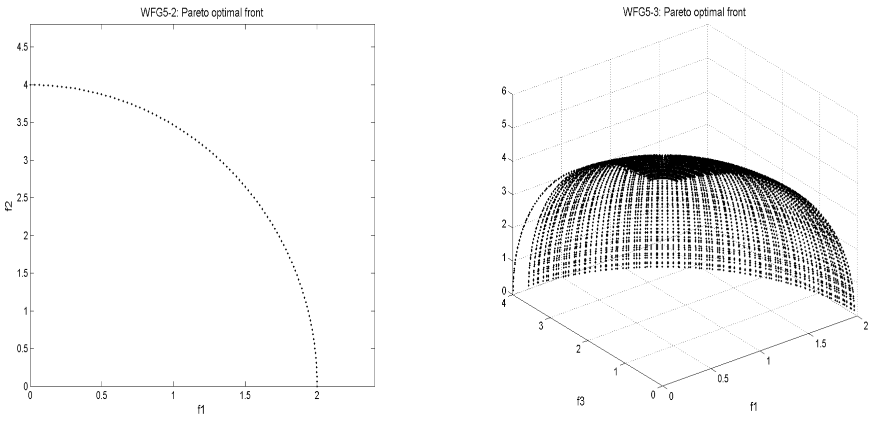
Figure A.16: Pareto optimal fronts of WFG4-2 and WFG4-3.

- WFG5

$$
\begin{aligned}
h_{m=1,\cdots,M} &= concave\_m \\
t^1_{i=1:n} &= s\_decept(y_1, 0.35, 0.001, 0.05) \\
&\mathbf{t}^2 \ is \ \ the \ same \ as \ \mathbf{t}^2 \ in \ WFG4 \ (Weighted \ sum \ reduction)
\end{aligned}
\tag{A.33}
$$

Pareto optimal fronts of 2 and 3-objective WFG5 are show in Figure A.17.

- WFG6

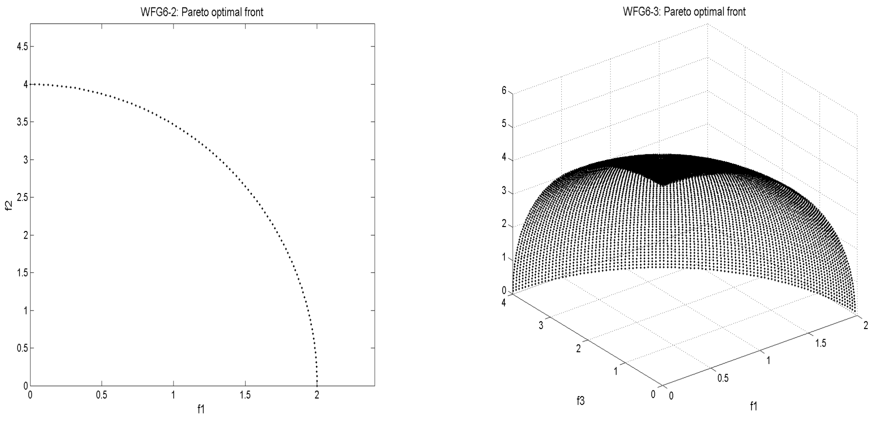


(a) WFG5-2 (b) WFG5-3

FIGURE A.17: Pareto optimal fronts of WFG5-2 and WFG5-3.

$$
\begin{aligned}
h_{m=1:M} &= concave\_m \\
\mathbf{t}^1 isthe\ same\ as\ \mathbf{t}^1\ in\ WFG1\ (Linear\ shift) \\
t^2_{i=1:M-1} &= r\_nonsep(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, k/(M-1)) \\
t^2_M &= r\_nonsep(\{y_{k+1}, \cdots, y_n\}, l)
\end{aligned}
\tag{A.34}
$$

Pareto optimal fronts of 2 and 3-objective WFG6 are show in Figure A.18.



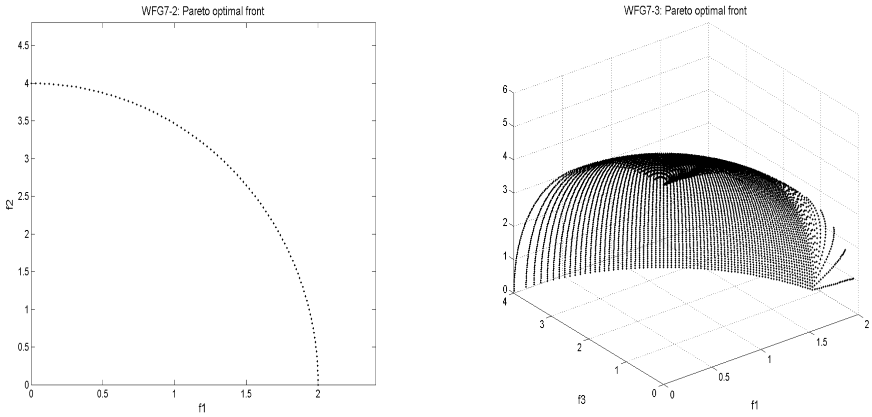
(a) WFG6-2 (b) WFG6-3

FIGURE A.18: Pareto optimal fronts of WFG6-2 and WFG6-3.

- WFG7

$$
\begin{aligned}
h_{m=1:M} &= concave\_m \\
t^1_{i=1:k} &= b\_param(y_1, r\_sum(\{y_{i+1}, \cdots, y_n\}, \{1, \cdots, 1\}), \frac{0.98}{49.98}, 0.02, 50) \\
t^1_{i=k+1:n} &= y_1 \\
&\mathbf{t}^2 \ is \ \ the \ same \ as \ \mathbf{t}^1 \ in \ WFG1 \ (Linear \ shift) \\
&\mathbf{t}^3 \ is \ \ the \ same \ as \ \mathbf{t}^2 \ in \ WFG4 \ (Weighted \ sum \ reduction)
\end{aligned}
\tag{A.35}
$$

Pareto optimal fronts of 2 and 3-objective WFG7 are show in Figure A.19.



(a) WFG7-2

(b) WFG7-3

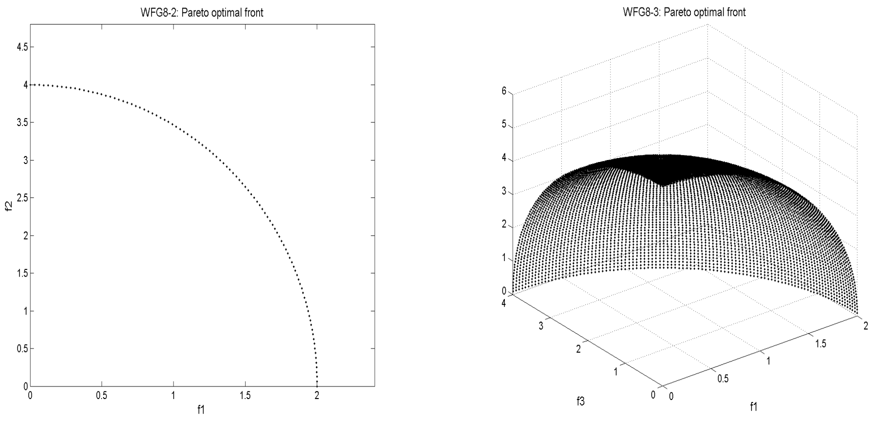
Figure A.19: Pareto optimal fronts of WFG7-2 and WFG7-3.

- WFG8

$$
\begin{aligned}
h_{m=1:M} &= concave\_m \\
t^1_{i=1:k} &= y_i \\
t^1_{i=k+1:n} &= b\_param(y_1, r\_sum(\{y_{i+1}, \cdots, y_n\}, \{1, \cdots, 1\}), \frac{0.98}{49.98}, 0.02, 50) \\
t^1_{i=k+1:n} &= y_1 \\
&\mathbf{t}^2 \ is \ \ the \ same \ as \ \mathbf{t}^1 \ in \ WFG1 \ (Linear \ shift) \\
&\mathbf{t}^3 \ is \ \ the \ same \ as \ \mathbf{t}^2 \ in \ WFG4 \ (Weighted \ sum \ reduction)
\end{aligned}
\tag{A.36}
$$

Pareto optimal fronts of 2 and 3-objective WFG8 are show in Figure A.20.

- WFG9

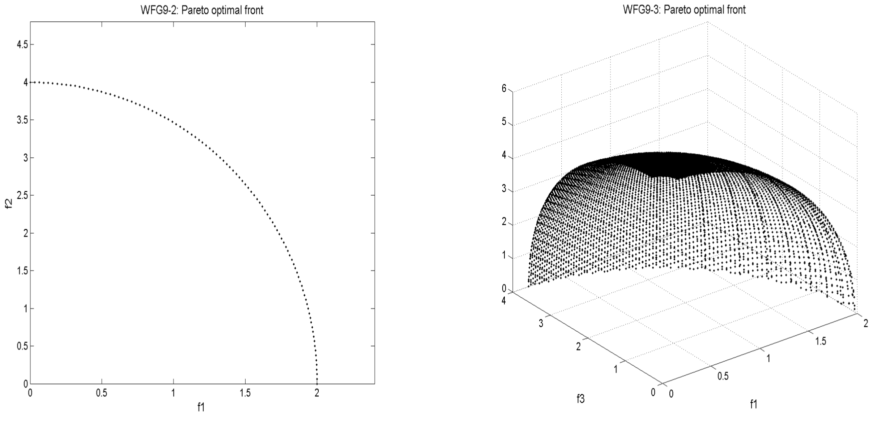


(a) WFG8-2 (b) WFG8-3

Figure A.20: Pareto optimal fronts of WFG8-2 and WFG8-3.

$$
\begin{aligned}
h_{m=1:M} &= concave\_m \\
t^1_{i=1:n-1} &= b\_param(y_1, r\_sum(\{y_{i+1}, \cdots, y_n\}, \{1, \cdots, 1\}), \frac{0.98}{49.98}, 0.02, 50) \\
t^1_n &= y_n \\
t^2_{i=1:k} &= s\_decept(y_i, 0.35, 0.001, 0.05) \\
t^2_{i=k+1:n} &= s\_multi(y_i, 30, 95, 0.35) \\
\mathbf{t}^3 &\ is\ the\ same\ as\ \mathbf{t}^2\ in\ WFG6\ (Non\ separable\ reduction)
\end{aligned}
\tag{A.37}
$$

Pareto optimal fronts of 2 and 3-objective WFG9 are show in Figure A.21.



(a) WFG9-2 (b) WFG9-3

Figure A.21: Pareto optimal fronts of WFG9-2 and WFG9-3.

The modified WFG4X test suite

The WFG4X test suite is developed by replacing the shape function used in WFG4 problem with different shapes functions. The Pareto optimal front of the WFG4X problems have the same trade-off magnitudes, that is, $S_{1:M} = 2$, see Equation A.28.

- WFG41 is the same as the WFG4 problem which has a concave Pareto optimal front.

$$\begin{aligned} h_{m=1,\cdots,M} &= concave\_m \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.38}$$

- WFG42 has a convex Pareto optimal front. It is built by replacing the concave shape function used in WFG4 with the convex shape function.

$$\begin{aligned} h_{m=1,\cdots,M} &= convex\_m \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.39}$$

- WFG43 has a strong concave Pareto optimal front. It is built by scaling the concave shape function with power $\frac{1}{4}$.

$$\begin{aligned} h_{m=1,\cdots,M} &= (concave\_m)^{\frac{1}{4}} \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.40}$$

- WFG44 has a strong convex Pareto optimal front. It is built by scaling the convex shape function with power $\frac{1}{4}$.

$$\begin{aligned} h_{m=1,\cdots,M} &= (convex\_m)^{\frac{1}{4}} \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.41}$$

- WFG45 has a mixed Pareto optimal front. It is built by replacing the concave shape function used in WFG4 with the mixed shape function.

$$\begin{aligned} h_{m=1,\cdots,M} &= mixed\_m(with\ \alpha = 1\ and\ A = 5) \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.42}$$

- WFG46

$$\begin{aligned} h_{m=1,\cdots,M} &= linear\_m \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.43}$$

  The Pareto optimal front of WFG46 is a hyperplane. It is built by replacing the concave shape function used in WFG4 with the linear shape function.

- WFG47

$$\begin{aligned} h_{m=1,\cdots,M-1} &= concave\_m \\ h_M &= discon\_m(with\ \alpha = \beta = 1\ and\ A = 2) \\ t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\ t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\ t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\}) \end{aligned} \tag{A.44}$$

  The Pareto optimal front of WFG47 is disconnected and concave. It is built by replacing the concave shape function used in WFG4 with the concave (for the first $M - 1$ objectives) and the disconnected (for the last objective) shape function. Parameters used in the disconnected function are set as $\alpha = \beta = \frac{1}{2}, A = 2$.

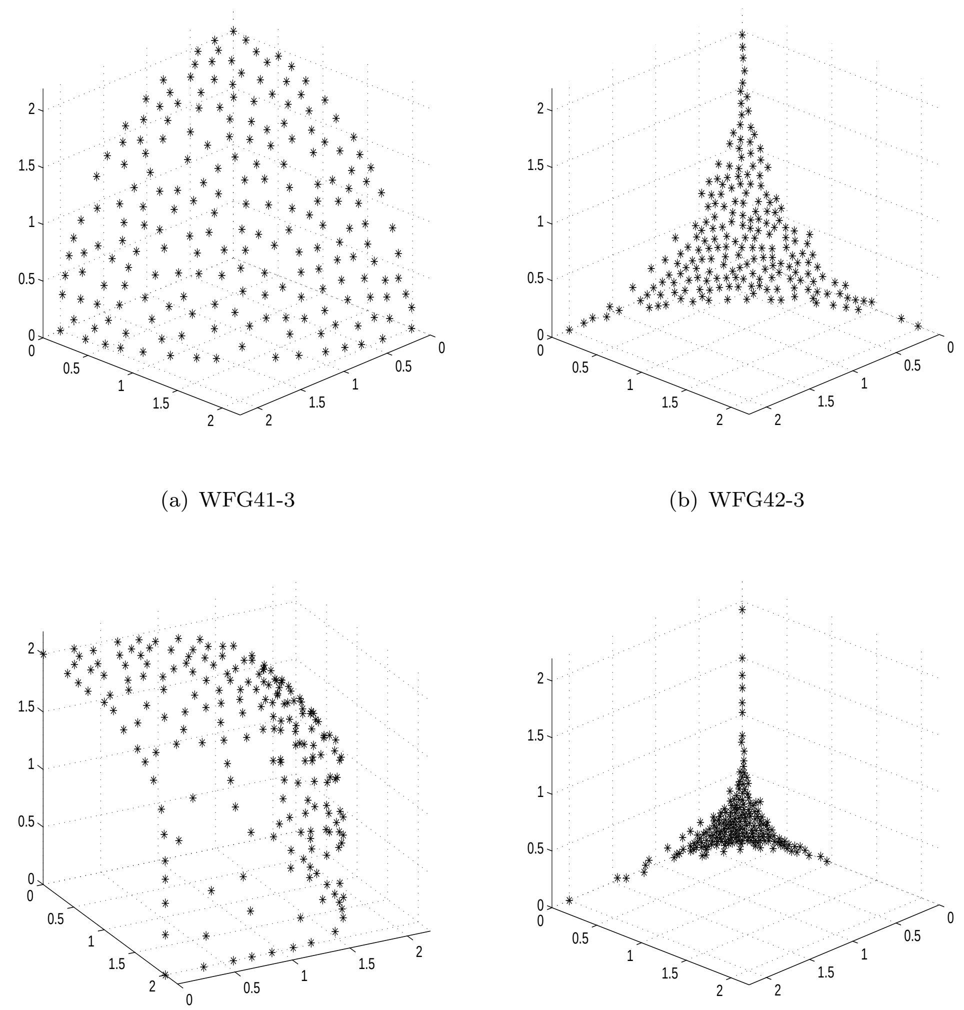
- WFG48

$$
\begin{aligned}
h_{m=1,\cdots,M-1} &= convex\_m \\
h_M &= discon\_m(with\ \alpha = \beta = 1\ and\ A = 2) \\
t^1_{i=1:n} &= s\_multi(y_1, 30, 10, 0.35) \\
t^2_{i=1:M-1} &= r\_sum(\{y_{(i-1)k/(M-1)+1}, \cdots, y_{ik/(M-1)}\}, \{1, \cdots, 1\}) \\
t^2_M &= r\_sum(\{y_{k+1}, \cdots, y_n\}, \{1, \cdots, 1\})
\end{aligned}
\tag{A.45}
$$

The Pareto optimal front of WFG48 is disconnected and convex. It is built by replacing the concave shape function used in WFG4 with the convex (for the first $M-1$ objectives) and the disconnected (for the last objective) shape function. Parameters used in the disconnected function are set as $\alpha = \beta = \frac{1}{2}, A = 2$.

Pareto optimal fronts of the 3-objective WFG4X problem are shown in Figure A.22 and Figure A.23.

The source code of all the above test problems is available in `http://www.sheffield.ac.uk/acse/staff/rstu/ruiwang/index`.



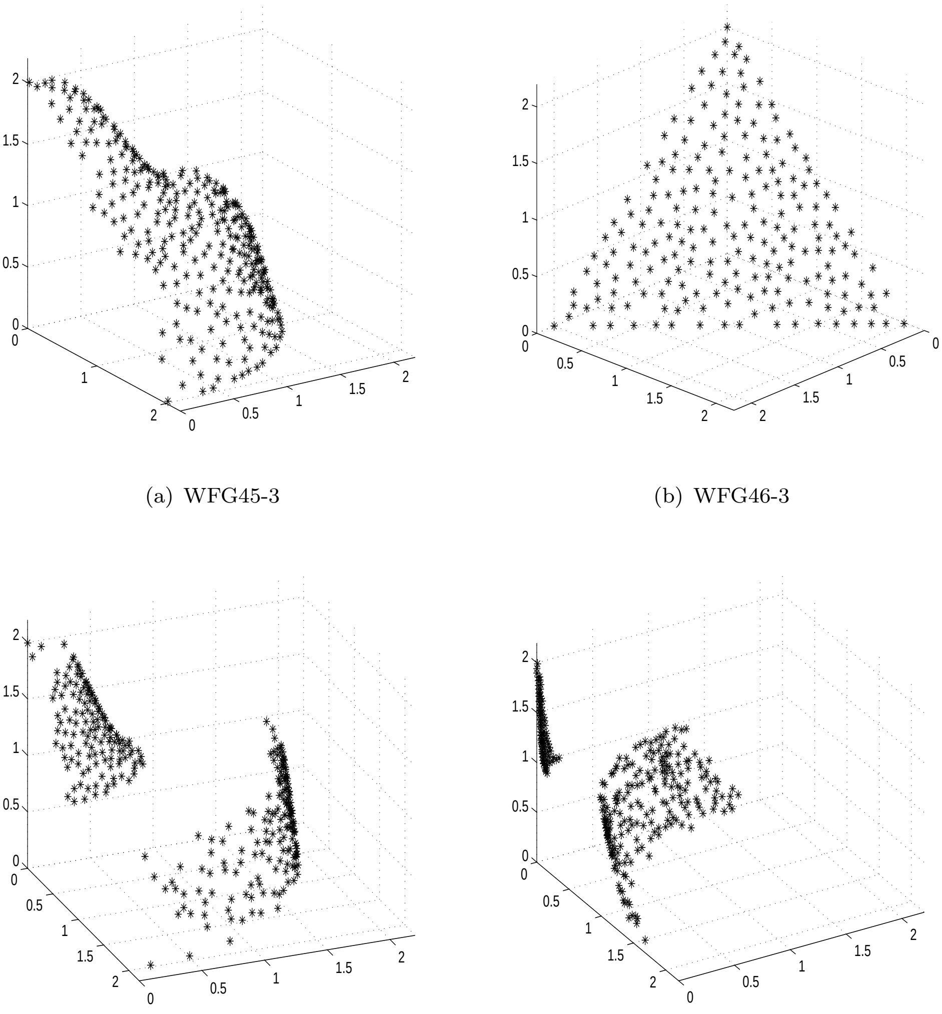
(a) WFG41-3

(b) WFG42-3

(c) WFG43-3

(d) WFG44-3

Figure A.22: Pareto optimal fronts of WFG41-3 to WFG44-3.



(a) WFG45-3

(b) WFG46-3

(c) WFG47-3

(d) WFG48-3

FIGURE A.23: Pareto optimal fronts of WFG45-3 to WFG48-3.

Appendix B

True hypervolume value of WFG problems

The true optimal hypervolume values (HV^*) for WFG problems vary by problem, due to the different Pareto front geometries employed. From Huband et al. (2006) the true Pareto-optimal front of test problems (WFG4 to WFG9) is the surface of M-dimension hyper-ellipsoid with radius $r_i = 2 \times i, i = 1, 2, \cdots, M$ in the first quadrant. Therefore HV^* can be computed by

$$HV^* = V_2 - \frac{V_1}{2^M} \tag{B.1}$$

where V_1 is the volume of M-dimension hyper-ellipsoid, and V_2 is the volume of M-dimension hypercube, which is constructed by the reference point and the coordinate origin. As the volume covered by the Pareto front is only in the first $\frac{1}{2^M}$ area, e.g. the first quadrant for 2 dimensions. So V_1 must be divided by 2^M. The formula of calculating V_1 is given as follows (Schrocker, 2008):

$$V_1 = \begin{cases} \frac{1}{(\frac{M}{2})!}\pi^{\frac{M}{2}} \prod_{i=1}^{M} r_i & M \text{ is even} \\ \frac{2^{(\frac{M+1}{2})}}{(M)!!}\pi^{\frac{M-1}{2}} \prod_{i=1}^{M} r_i & M \text{ is odd} \end{cases} \tag{B.2}$$

Taking the 2-objective WFG4 test as an example, see Figure B.1:

$V_1 = \pi r_1 r_2 = \pi \times 1 \times 1 = 1.44\pi$, and $V_2 = 1.2 \times 1.2 = 1.44$. So $HV^* = 1.44 - \frac{pi}{2^2} = 0.6546$.

As the Pareto front of WFG2 is disconnected, the above computation process is not applicable. However, in Huband et al. (2006) the authors point out that the optimal solutions of WFG2 satisfy the condition Equation B.3:

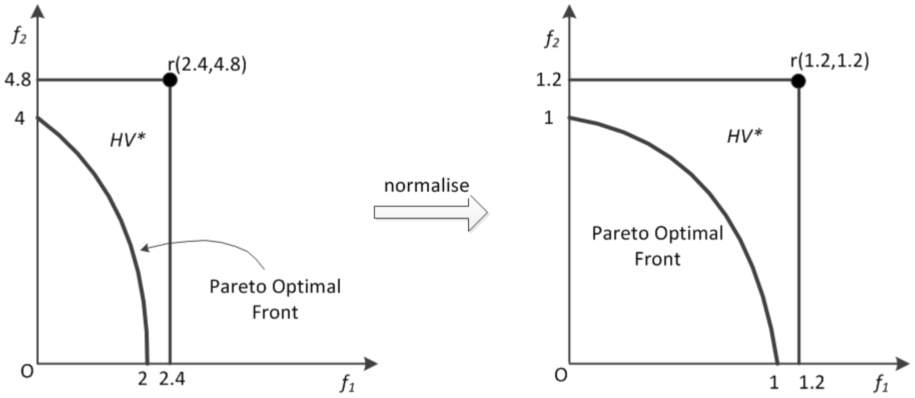


FIGURE B.1: True hypervolume calculation.

$$Z_{i=k+l:n} = 2 \times i \times 0.35 \tag{B.3}$$

where n is the number of decision variables and $n = k+l$, k and l are position and distance parameters. Therefore, we generate $10,000 \times M$ optimal solutions for M-dimension WFG2 and then the approximation of HV^* can be calculated by using conventional methods (Zitzler and Thiele, 1999; Fonseca et al., 2006).

Appendix C

An introduction to the Parallel coordinate system

The concept of Parallel coordinates was first introduced by Inselberg (1985) and subsequently applied to EMO by Fonseca and Fleming (1993). It places all the axes parallel to each other thus effectively reducing an arbitrary high-dimensional space to two-dimensions.

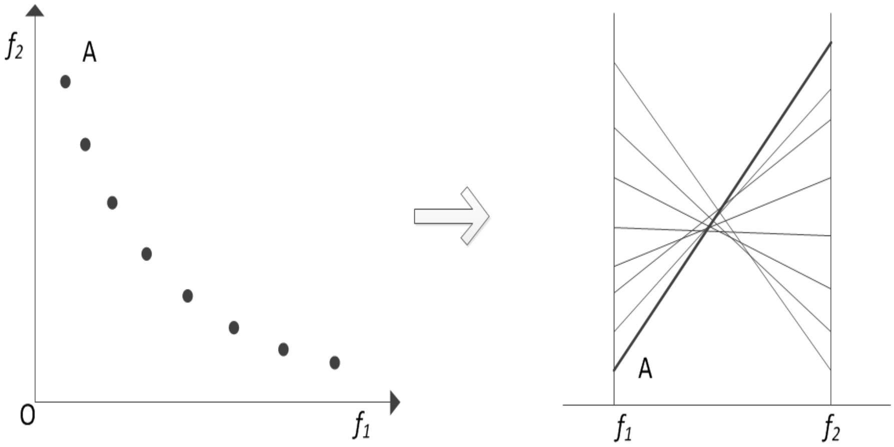


FIGURE C.1: Mapping from Cartesian into parallel coordinates.

In Cartesian system, a solution (in objective-space) is displayed by a single point. In the Parallel coordinates system, a solution is displayed by joining the objective values in all adjacent objectives by straight lines. Figure C.1 illustrates the mapping between the Cartesian system and the parallel coordinates, where point A in the coordinate system is represented by line A in the parallel coordinates representation.

Parallel coordinates also visually represent the relationships, such as conflict and harmony, between design objectives. Specifically, lines representing two adjacent objectives will cross if conflict is observed or will fail to cross if harmony is exhibited between the

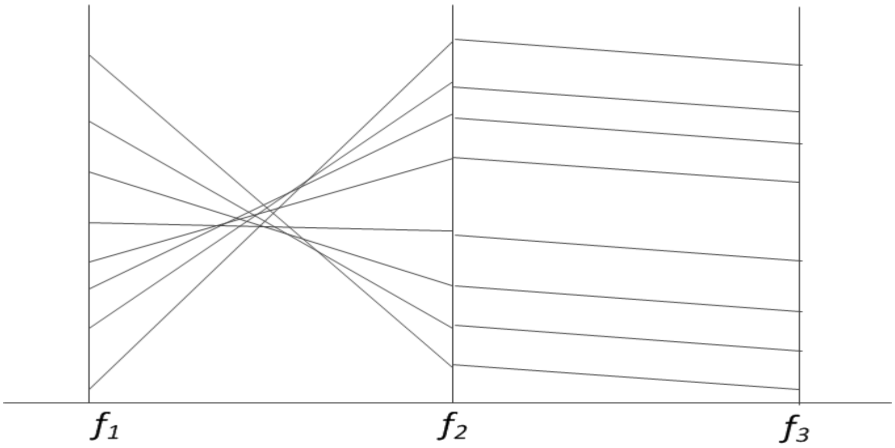


FIGURE C.2: Solutions displayed using parallel coordinates.

two objectives. For example, in Figure C.2, objectives f_1 and f_2 are in conflict while objectives f_2 and f_3 are harmony.

Bibliography

Allmendinger, R., Li, X., Branke, J., 2008. Reference point-based particle swarm optimization using a steady-state approach. In: Proceedings of the 7th International Conference on Simulated Evolution and Learning. Springer, pp. 200–209.

Archer, G., Saltelli, A., Sobol, I., 1997. Sensitivity measures, anova-like techniques and the use of bootstrap. Journal of Statistical Computation and Simulation 58 (2), 99–120.

Auger, A., Bader, J., Brockhoff, D., Zitzler, E., 2009a. Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In: GECCO 2009: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, Montreal,Canada, pp. 555–562.

Auger, A., Bader, J., Brockhoff, D., Zitzler, E., 2009b. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms. ACM, pp. 87–102.

Bader, J., Zitzler, E., 2010. A hypervolume-based optimizer for high-dimensional objective spaces. In: Jones, D., Tamiz, M., Ries, J. (Eds.), New Developments in Multiple Objective and Goal Programming. Vol. 638 of Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg, pp. 35–54.

Bader, J., Zitzler, E., Jan. 2011. HypE: an algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation 19 (1), 45–76.

Barbosa, H. J., Barreto, A. M., 2001. An interactive genetic algorithm with co-evolution of weights for multiobjective problems. In: GECCO 2001: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 203–210.

Batista, L. S., Campelo, F., Guimarães, F. G., Ramírez, J. A., 2011. Pareto cone ε-dominance: improving convergence and diversity in multiobjective evolutionary algorithms. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 76–90.

Ben Said, L., Bechikh, S., Ghédira, K., 2010. The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. Evolutionary Computation, IEEE Transactions on 14 (5), 801–818.

Bentley, P. J., Wakefield, J. P., 1997. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. Soft Computing in Engineering Design and Manufacturing 5, 231–240.

Beume, N., Fonseca, C., López-Ibáñez, M., Paquete, L., Vahrenhold, J., 2009. On the complexity of computing the hypervolume indicator. IEEE Transactions on Evolutionary Computation 13 (5), 1075–1082.

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

Bongard, J., Lipson, H., Aug. 2005. Nonlinear System Identification Using Coevolution of Models and Tests. IEEE Transactions on Evolutionary Computation 9 (4), 361–384.

Branke, J., Jun. 2001. Guidance in evolutionary multi-objective optimization. Advances in Engineering Software 32 (6), 499–507.

Branke, J., Deb, K., 2005. Integrating user preferences into evolutionary multi-objective optimization. Knowledge Incorporation in Evolutionary Computation, 461–477.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V., Dec. 2006. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation 10 (6), 646–657.

Brockhoff, D., Zitzler, E., 2006. Dimensionality reduction in multiobjective optimization: The minimum objective subset problem. In: Proc. of Operations Research 2006. Citeseer, pp. 423–429.

Brockhoff, D., Zitzler, E., 2007. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In: Evolutionary Computation (CEC), 2007 IEEE Congress on. IEEE, pp. 2086–2093.

Brockhoff, D., Zitzler, E., 2009. Objective reduction in evolutionary multiobjective optimization: theory and applications. Evolutionary computation 17 (2), 135–166.

Buja, A., Cook, D., Scientist, D. F. S. R., 1996. Interactive high-dimensional data visualization. Journal of Computational and Graphical Statistics 5 (1), 78–99.

Coello Coello, C. A., 2000. Handling preferences in evolutionary multiobjective optimization: A survey. In: Evolutionary Computation (CEC), 2000 IEEE Congress on. Vol. 1. IEEE, pp. 30–37.

Coello Coello, C. A., 2006. Evolutionary multi-objective optimization: a historical view of the field. Computational Intelligence Magazine, IEEE 1 (1), 28–36.

Coello Coello, C. A., Lamont, G., Van Veldhuizen, D., 2007. Evolutionary algorithms for solving multi-objective problems. Springer.

Coello Coello, C. A., Sierra, M., 2003. A coevolutionary multi-objective evolutionary algorithm. In: Evolutionary Computation (CEC), 2003 IEEE Congress on. IEEE, pp. 482–489.

Cook, J. E., Tauritz, D. R., 2010. An exploration into dynamic population sizing. In: GECCO 2012: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, pp. 807–814.

Corne, D., Knowles, J., 2007. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: GECCO 2007: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, pp. 773–780.

Corne, D., Knowles, J., Oates, M., 2000. The pareto envelope-based selection algorithm for multiobjective optimization. In: Parallel Problem Solving from Nature PPSN VI. Springer, pp. 839–848.

Corne, D. W., Jerram, N. R., Knowles, J. D., Oates, M. J., July 7-11 2001. Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In: GECCO 2001: Proceedings of the Genetic and Evolutionary Computation Conference. San Francisco, California, pp. 283–290.

Cruz, C., González, J. R., Pelta, D. A., 2011. Optimization in dynamic environments: a survey on problems, methods and measures. Soft Computing 15 (7), 1427–1448.

Curtin, F., Schulz, P., 1998. Multiple correlations and bonferronis correction. Biological psychiatry 44 (8), 775–777.

Czyzżak, P., Jaszkiewicz, A., 1998. Pareto simulated annealing–a metaheuristic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis 7 (1), 34–47.

Datta, R., Deb, K., 2011. A bi-objective based hybrid evolutionary-classical algorithm for handling equality constraints. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 313–327.

De Jong, E. D., Pollack, J. B., Jan. 2004. Ideal evaluation from coevolution. Evolutionary computation 12 (2), 159–92.

Deb, K., 1999. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary computation 7 (3), 205–230.

Deb, K., 2001. Multi-objective optimization using evolutionary algorithms. Wiley.

Deb, K., Agrawal, R. B., 1994. Simulated binary crossover for continuous search space. Complex Systems 9, 1–34.

Deb, K., Chaudhuri, S., Miettinen, K., 2006. Towards estimating nadir objective vector using evolutionary approaches. In: GECCO 2006: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, New York, New York, USA, pp. 643–650.

Deb, K., Kumar, A., 2007a. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In: GECCO 2007: Proceedings of the Genetic and Evolutionary Computation Conference. Vol. 1. ACM, London, England,UK.

Deb, K., Kumar, A., 2007b. Light beam search based multi-objective optimization using evolutionary algorithms. In: Evolutionary Computation (CEC), 2007 IEEE Congress on. IEEE, pp. 2125–2132.

Deb, K., Miettinen, K., 2009. A review of nadir point estimation procedures using evolutionary approaches: A tale of dimensionality reduction. In: Proceedings of the Multiple Criterion Decision Making (MCDM-2008) Conference. pp. 1–14.

Deb, K., Miettinen, K., Chaudhuri, S., 2010. Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. c 14 (6), 821–841.

Deb, K., Mohan, M., Mishra, S., 2003. Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 222–236.

Deb, K., Mohan, M., Mishra, S., Jan. 2005. Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. Evolutionary computation 13 (4), 501–25.

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

Deb, K., Saxena, D., 2006. Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proceedings of the World Congress on Computational Intelligence (WCCI-2006). pp. 3352–3360.

Deb, K., Sindhya, K., Okabe, T., 2007. Self-adaptive simulated binary crossover for real-parameter optimization. In: GECCO 2007: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, London, England,UK, pp. 1187–1194.

Deb, K., Sundar, J., 2006. Reference point based multi-objective optimization using evolutionary algorithms. In: GECCO 2006: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, New York, New York, USA, pp. 635–642.

Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2002b. Scalable multi-objective optimization test problems. In: Evolutionary Computation (CEC), 2002 IEEE Congress on. IEEE, pp. 825–830.

Della Cioppa, A., De Stefano, C., Marcelli, A., 2007. Where are the niches? dynamic fitness sharing. IEEE Transactions on Evolutionary Computation 11 (4), 453–465.

Derbel, B., Brockhoff, D., Liefooghe, A., 2013. Force-based cooperative search directions in evolutionary multi-objective optimization. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 383–397.

Di Pierro, F., Khu, S. T., Savic, D. A., 2007. An investigation on preference order ranking scheme for multiobjective evolutionary optimization. IEEE Transactions on Evolutionary Computation 11 (1), 17–45.

Drechsler, N., Drechsler, R., Becker, B., 2001. Multi-objective optimisation based on relation favour. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 154–166.

Ehrgott, M., 2005. Multicriteria optimization. Vol. 2. Springer Berlin.

Eiben, A. E., Hinterding, R., Michalewicz, Z., 1999. Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3 (2), 124–141.

Emmerich, M., Beume, N., Naujoks, B., 2005. An EMO algorithm using the hypervolume measure as selection criterion. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 62–76.

Fabre, M. G., Pulido, G. T., Coello, C., 2009. Ranking methods for many-objective optimization. MICAI 2009: Advances in Artificial Intelligence, 633–645.

Fabre, M. G., Pulido, G. T., Coello, C., 2010. Alternative fitness assignment methods for many-objective optimization problems. Artifical Evolution, 146–157.

Farina, M., Amato, P., 2002. On the optimal solution definition for many-criteria optimization problems. In: 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. IEEE, pp. 233–238.

Fleming, P., Purshouse, R., Lygoe, R., 2005. Many-objective optimization: An engineering design perspective. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 14–32.

Fodor, I. K., 2002. A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory 9, 1–18.

Fonseca, C., Fleming, P., 1993. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc., pp. 416–423.

Fonseca, C., Fleming, P., 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. In: Parallel Problem Solving from Nature – PPSN VI. Vol. 1141. Springer, pp. 584–593.

Fonseca, C., Fleming, P., 1998a. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 28 (1), 26–37.

Fonseca, C., Fleming, P., 1998b. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. II. Application example. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 28 (1), 38–47.

Fonseca, C., Paquete, L., López-Ibáñez, M., 2006. An improved dimension-sweep algorithm for the hypervolume indicator. In: Evolutionary Computation (CEC), 2006 IEEE Congress on. IEEE, pp. 1157–1163.

Fonseca, C. M., Fleming, P. J., 1995a. Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In: Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. IET, pp. 45–52.

Fonseca, C. M., Fleming, P. J., 1995b. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation 3 (1), 1–16.

Giagkiozis, I., Fleming, P., 2012. Methods for many-objective optimization: An analysis. Tech. Rep. No. 1030, Automatic Control and Systems Engineering, University of Sheffield.

Giagkiozis, I., Purshouse, R. C., Fleming, P. J., 2013. Generalized decomposition. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 428–442.

Goh, C., Tan, K., Feb. 2009. A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization. IEEE Transactions on Evolutionary Computation 13 (1), 103–127.

Goh, C., Tan, K., Liu, D., Chiam, S., Apr. 2010. A competitive and cooperative coevolutionary approach to multi-objective particle swarm optimization algorithm design. European Journal of Operational Research 202 (1), 42–54.

Goldberg, D., 1989. Genetic algorithms in search, optimization, and machine learning. Addison-wesley.

Gu, F., Liu, H., Tan, K., 2012. A multiobjective evolutionary algorithm using dynamic weight design method. International Journal of Innovative Computing Information and Control 8 (5B), 3677–3688.

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

Hajela, P., Lee, E., Lin, C., 1993. Genetic algorithms in structural topology optimization. Topology design of structures 227, 117–134.

Hauser, H., Ledermann, F., Doleisch, H., 2002. Angular brushing of extended parallel coordinates. In: Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on. IEEE, pp. 127–130.

Hernández-Díaz, A., Santana-Quintero, L., Coello Coello, C., Molina, J., 2007. Pareto-adaptive ε-dominance. Evolutionary computation 15 (4), 493–517.

Hollander, M., Wolfe, D., 1999. Nonparametric statistical methods. Wiley-Interscience.

Horn, J., Nafpliotis, N., Goldberg, D., 1994. A niched pareto genetic algorithm for multiobjective optimization. In: Evolutionary Computation (CEC), 1994 IEEE Congress on. IEEE, pp. 82–87.

Huband, S., Hingston, P., Barone, L., While, L., Oct. 2006. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 10 (5), 477–506.

Hughes, E., 2003. Multiple single objective pareto sampling. In: Evolutionary Computation (CEC), 2003 IEEE Congress on. IEEE, pp. 2678–2684.

Hughes, E., 2005. Evolutionary many-objective optimisation: many once or one many? In: Evolutionary Computation (CEC), 2005 IEEE Congress on. IEEE, pp. 222–227.

Hughes, E. J., 2007. MSOPS-II: A general-purpose Many-Objective optimiser. In: Evolutionary Computation (CEC), 2007 IEEE Congress on. IEEE, pp. 3944–3951.

Ikeda, K., Kita, H., Kobayashi, S., 2001. Failure of pareto-based moeas: Does non-dominated really mean near to optimal? In: Evolutionary Computation (CEC), 2001 IEEE Congress on. Vol. 2. IEEE, pp. 957–962.

Inselberg, A., 1985. The plane with parallel coordinates. The Visual Computer 1 (2), 69–91.

Iorio, A. W., Li, X., 2004. A cooperative coevolutionary multiobjective algorithm using non-dominated sorting. In: GECCO 2004: Proceedings of the Genetic and Evolutionary Computation Conference. Springer, pp. 537–548.

Ishibuchi, H., Hitotsuyanagi, Y., Ohyanagi, H., Nojima, Y., 2011. Effects of the Existence of Highly Correlated Objectives on the Behavior of MOEA/D. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 166–181.

Ishibuchi, H., Murata, T., 1998. A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 28 (3), 392–403.

Ishibuchi, H., Narukawa, K., Tsukamoto, N., Nojima, Y., 2008a. An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. European Journal of Operational Research 188 (1), 57–75.

Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y., 2009a. Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 438–452.

Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y., May 2009b. Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization. In: Evolutionary Computation (CEC), 2009 IEEE Congress on. IEEE, pp. 2508–2515.

Ishibuchi, H., Sakane, Y., Tsukamoto, N., Nojima, Y., May 2009c. Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. In: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics. Vol. 1. Ieee, San Antonio, TX, USA, pp. 1820–1825.

Ishibuchi, H., Shibata, Y., 2003. A similarity-based mating scheme for evolutionary multiobjective optimization. In: GECCO 2003: Proceedings of the Genetic and Evolutionary Computation Conference. Springer, Chicago, USA, pp. 201–201.

Ishibuchi, H., Shibata, Y., 2004. Mating scheme for controlling the diversity-convergence balance for multiobjective optimization. In: GECCO 2004: Proceedings of the Genetic and Evolutionary Computation Conference. Springer, Seattle, Washington, USA, pp. 1259–1271.

Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008b. Evolutionary many-objective optimization: A short review. In: Evolutionary Computation (CEC), 2008 IEEE Congress on. IEEE, pp. 2419–2426.

Ishibuchi, H., Tsukamoto, N., Sakane, Y., Nojima, Y., 2009d. Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization. In: Evolutionary Computation (CEC), 2009 IEEE Congress on. IEEE, pp. 530–537.

Ishibuchi, H., Tsukamoto, N., Sakane, Y., Nojima, Y., 2010. Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In: GECCO 2010: Proceedings of the Genetic and Evolutionary ComputationConference. ACM, pp. 527–534.

Ishibuchi, H., Yoshida, T., Murata, T., Apr. 2003. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation 7 (2), 204–223.

Jaszkiewicz, A., 2002. Genetic local search for multi-objective combinatorial optimization. European journal of operational research 137 (1), 50–71.

Jiang, S., Cai, Z., Zhang, J., Ong, Y., 2011. Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In: Natural Computation (ICNC), 2011 Seventh International Conference on. Vol. 3. IEEE, pp. 1260–1264.

Jin, Y., Branke, J., 2005. Evolutionary optimization in uncertain environments-a survey. IEEE Transactions on Evolutionary Computation 9 (3), 303–317.

Jin, Y., Okabe, T., 2001. Adapting weighted aggregation for multiobjective evolution strategies. In: Evolutionary Multi-Criterion Optimization. pp. 96–110.

Jin, Y., Sendhoff, B., 2002. Incorporation of fuzzy preferences into evolutionary multiobjective optimization. In: Proceedings of the 4th Asia Pacific Conference on Simulated Evolution and Learning. Vol. 1. pp. 26–30.

Justel, A., Peña, D., Zamar, R., 1997. A multivariate kolmogorov-smirnov test of goodness of fit. Statistics & Probability Letters 35 (3), 251–259.

Kandogan, E., 2001. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 107–116.

Keerativuttitumrong, N., Chaiyaratana, N., Varavithya, V., 2002. Multi-objective cooperative co-evolutionary genetic algorithm. In: Parallel Problem Solving from Nature-PPSN VII. Springer, pp. 288–297.

Kim, I. Y., De Weck, O., 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. Structural and multidisciplinary optimization 29 (2), 149–158.

Kim, I. Y., De Weck, O., 2006. Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. Structural and Multidisciplinary Optimization 31 (2), 105–116.

Kleeman, M., Lamont, G., 2006. Coevolutionary Multi-Objective EAs: The Next Frontier? In: Evolutionary Computation (CEC), 2006 IEEE Congress on. IEEE, pp. 1726–1735.

Knowles, J., 2002. Local-search and hybrid evolutionary algorithms for pareto optimization. Ph.D. thesis, Department of Computer Science, University of Reading.

Knowles, J., Corne, D., 1999. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In: Evolutionary Computation (CEC), 1999 IEEE Congress on. Vol. 1. IEEE.

Knowles, J., Corne, D., 2002. On metrics for comparing nondominated sets. In: Evolutionary Computation (CEC), 2002 IEEE Congress on. Vol. 1. IEEE, pp. 711–716.

Knowles, J., Corne, D., 2003. Properties of an adaptive archiving algorithm for storing nondominated vectors. IEEE Transactions on Evolutionary Computation 7 (2), 100–116.

Knowles, J., Corne, D., 2007. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 757–771.

Knowles, J. D., Corne, D. W., 2000. Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary computation 8 (2), 149–172.

Kukkonen, S., Member, S., Lampinen, J., Sep. 2007. Ranking-Dominance and Many-Objective Optimization. In: Evolutionary Computation (CEC), 2007 IEEE Congress on. IEEE, pp. 3983–3990.

Laumanns, M., Rudolph, G., Schwefel, H.-P., 1998. A spatial predator-prey approach to multi-objective optimization: A preliminary study. In: Parallel Problem Solving from NaturePPSN V. Springer, pp. 241–249.

Laumanns, M., Thiele, L., Deb, K., Zitzler, E., Jan. 2002. Combining convergence and diversity in evolutionary multiobjective optimization. Evolutionary computation 10 (3), 263–82.

Lee, M., Esbensen, H., Lemaitre, L., 1996. The design of hybrid fuzzy/evolutionary multiobjective optimization algorithms. Fuzzy Logic, Neural Networks, and Evolutionary Computation, 1–20.

Lee, M. A., Esbensen, H., 1997. Fuzzy/multiobjective genetic systems for intelligent systems design tools and components. In: Fuzzy evolutionary computation. Kluwer Academic Publishers, pp. 57–78.

Li, H., Landa-Silva, D., 2011. An adaptive evolutionary multi-objective approach based on simulated annealing. Evolutionary Computation 19 (4), 561–595.

Li, H., Zhang, Q., 2009. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13 (2), 284–302.

Lobo, F. G., 2011. Idealized dynamic population sizing for uniformly scaled problems. In: GECCO 2011: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, pp. 917–924.

Lohn, J., Kraus, W., Haith, G., 2002. Comparing a coevolutionary genetic algorithm for multiobjective optimization. In: Evolutionary Computation (CEC), 2002 IEEE Congress on. Vol. 2. IEEE, pp. 1157–1162.

López Jaimes, A., Coello, C., Urías Barrientos, J., 2009. Online objective reduction to deal with many-objective problems. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 423–437.

López Jaimes, A., Coello Coello, C., Aguirre, H., Tanaka, K., 2011. Adaptive objective space partitioning using conflict information for many-objective optimization. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 151–165.

López Jaimes, A., Coello Coello, C. A., Chakraborty, D., 2008. Objective reduction using a feature selection technique. In: GECCO 2008: Proceedings of the Genetic and Evolutionary ComputationConference. ACM, pp. 673–680.

Lygoe, R. J., Cary, M., Fleming, P. J., 2010. A many-objective optimisation decision-making process applied to automotive diesel engine calibration. In: Simulated Evolution and Learning. Springer, pp. 638–646.

Mallipeddi, R., Suganthan, P. N., 2010. Ensemble of constraint handling techniques. IEEE Transactions on Evolutionary Computation 14 (4), 561–579.

Maneeratana, K., Boonlong, K., Chaiyaratana, N., 2006. Compressed-objective genetic algorithm. Parallel Problem Solving from Nature – PPSN IX, 473–482.

Massebeuf, S., Fonteix, C., Kiss, L. N., Marc, I., Pla, F., Zaras, K., 1999. Multicriteria optimization and decision engineering of an extrusion process aided by a diploid genetic algorithm. In: Evolutionary Computation (CEC), 1999 IEEE Congress on. Vol. 1. IEEE.

Mezura-Montes, E., Coello Coello, C. A., 2011. Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm and Evolutionary Computation 1 (4), 173–194.

Miettinen, K., 1999. Nonlinear multiobjective optimization. Vol. 12. Springer.

Miettinen, K., Mäkelä, M., 2002. On scalarizing functions in multiobjective optimization. OR spectrum 24 (2), 193–213.

Molina, J., Santana, L. V., Hernández-Díaz, A. G., Coello Coello, C. a., Caballero, R., 2009. g-dominance: Reference point based dominance for multiobjective metaheuristics. European Journal of Operational Research 197 (2), 685–692.

Murata, T., Ishibuchi, H., Gen, M., 2000. Cellular genetic local search for multi-objective optimization. In: GECCO. pp. 307–314.

Murata, T., Ishibuchi, H., Gen, M., 2001. Specification of genetic search directions in cellular multi-objective genetic algorithms. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 82–95.

Murata, T., Nozawa, H., Ishibuchi, H., Gen, M., 2003. Modification of local search directions for non-dominated solutions in cellular multiobjective genetic algorithms for pattern classification problems. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 593–607.

Murata, T., Nozawa, H., Tsujimura, Y., Gen, M., Ishibuchi, H., 2002. Effect of local search on the performance of cellular multiobjective genetic algorithms for designing fuzzy rule-based classification systems. In: Evolutionary Computation (CEC), 2002 IEEE Congress on. Vol. 1. IEEE, pp. 663–668.

Nguyen, T. T., Yang, S., Branke, J., 2012. Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation 6, 1–24.

Parreiras, R. O., Vasconcelos, J. A., 2005. Decision Making in Multiobjective Optimization Problems, n. nedjah and l. de macedo mourelle Edition. Real-World Multi-Objective System Engineering. Nova Science Publishers, New York.

Pereira, A. C. M. G., July 1995. Generating alternative routes using genetic algorithms and multi-criteria analysis techniques. In: Wyatt, R., Hossain, H. (Eds.), Fourth International Conference on Computers in Urban Planning and Urban Management. Melbourne, Australia, pp. 547–?60.

Pereira, A. C. M. G., 1997. Extending environmental impact assessment processes: Generation of alternatives for siting and routing infraestructural references facilities by multi-criteria evaluation and genetic algorithms. Ph.D. thesis, New University of Lisbon, Lisbon, Portugal,.

Pirjanian, P., 2000. Multiple objective behavior-based control. Robotics and Autonomous Systems 31 (1), 53–60.

Pirjanian, P., Mataric, M., 2000. Multi-robot target acquisition using multiple objective behavior coordination. In: Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. Vol. 3. IEEE, pp. 2696–2702.

Potter, M. A., Jong, K. A. D., Jan. 2000. Cooperative coevolution: an architecture for evolving coadapted subcomponents. Evolutionary Computation 8 (1), 1–29.

Purshouse, R., Fleming, P., 2003a. An adaptive divide-and-conquer methodology for evolutionary multi-criterion optimisation. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 72–72.

Purshouse, R., Fleming, P., 2003b. Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 67–67.

Purshouse, R., Fleming, P., 2003c. Evolutionary many-objective optimisation: an exploratory analysis. In: Evolutionary Computation (CEC), 2003 IEEE Congress on. Vol. 3. IEEE, pp. 2066–2073.

Purshouse, R., Jalbă, C., Fleming, P., 2011. Preference-driven co-evolutionary algorithms show promise for many-objective optimisation. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 136–150.

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

Rachmawati, L., Srinivasan, D., 2006. Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. In: Evolutionary Computation (CEC), 2006 IEEE Congress on. IEEE, pp. 962–968.

Rachmawati, L., Srinivasan, D., 2010. Incorporation of imprecise goal vectors into evolutionary multi-objective optimization. In: Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, pp. 1–8.

Rekiek., B., December 2000. Assembly line design (multiple objective grouping genetic algorithm and the balancing of mixed-model hybrid assembly line). Ph.D. thesis, CAD/CAM Department, Free University of Brussels, Brussels, Belgium.

Rekiek, B., De Lit, P., Pellichero, F., L'Eglise, T., Falkenauer, E., Delchambre, A., 2000. Dealing with user's preferences in hybrid assembly lines design. In: Binder, Z. (Ed.), Proceedings of the MCPL2000 Conference. Pergamon, Grenoble, France, pp. 989–994.

Rosin, C., Belew, R., 1997. New methods for competitive coevolution. Evolutionary Computation 5 (1), 1–29.

Saltelli, A., 2002. Making best use of model evaluations to compute sensitivity indices. Computer Physics Communications 145 (2), 280–297.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S., 2010. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. Computer Physics Communications 181 (2), 259–270.

Sato, H., Aguirre, H., Tanaka, K., 2007a. Controlling dominance area of solutions and its impact on the performance of MOEAs. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 5–20.

Sato, H., Aguirre, H., Tanaka, K., Sep. 2007b. Local dominance and local recombination in MOEAs on 0/1 multiobjective knapsack problems. European Journal of Operational Research 181 (3), 1708–1723.

Saxena, D. K., Deb, K., 2008. Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In: Evolutionary Computation (CEC), 2008 IEEE Congress on. IEEE, pp. 3204–3211.

Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., Zhang, Q., 2010. Objective reduction in many-objective optimization: Linear and nonlinear algorithms. IEEE Transactions on Evolutionary Computation, 1–22.

Schaffer, 1985. Multiple objective optimization with vector evaluated genetic algorithms. In: The 1st International Conference of Genetic Algorithms. Pittsburgh, PA, pp. 93–100.

Schott, J., 1995. Fault tolerant design using single and multicriteria genetic algorithm optimization. Ph.D. thesis, Department of Aeronautics and Alstronautics, Massachusetts Institute of Technology, Boston.

Schrocker, H., 2008. Uniqueness results for minimal enclosing ellipsoids. Computer Aided Geometric Design 25 (9), 756–762.

Schütze, O., Lara, A., Coello Coello, C. A., 2011. 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.

Serafini, P., 1994. Simulated annealing for multi objective optimization problems. In: Multiple criteria decision making. Springer, pp. 283–292.

Sheridan, R., 2008. Alternative global goodness metrics and sensitivity analysis: heuristics to check the robustness of conclusions from studies comparing virtual screening methods. Journal of chemical information and modeling 48 (2), 426–433.

Sinha, A., 2011. Progressively interactive evolutionary multiobjective optimization. Ph.D. thesis, Department of Business Technology, School of Economics, Aalto University.

Sobol, I., 2001. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Mathematics and computers in simulation 55 (1-3), 271–280.

Sobol, I., Kucherenko, S., 2005. On global sensitivity analysis of quasi-monte carlo algorithms. Monte Carlo Methods Appl 11 (1), 83–92.

Sobol, I. M., 1993. Sensitivity analysis for non-linear mathematical models. Mathematical Modelling and Computational Experiment 1 (1), 407–414.

Srinivas, M., Patnaik, L. M., 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics 24 (4), 656–667.

Srinivas, N., Deb, K., 1994. Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary computation 2 (3), 221–248.

Sülflow, A., Drechsler, N., Drechsler, R., 2007. Robust multi-objective optimization in high dimensional spaces. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 715–726.

Tabak, D., Schy, A., Giesy, D., Johnson, K., 1979. Application of multiobjective optimization in aircraft control systems design. Automatica 15 (5), 595–600.

Tan, K. C., Khor, E. F., Lee, T. H., Sathikannan, R., 2003. An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization. Journal of Artificial Intelligence Research 18, 183–215.

Tan, Y., Jiao, Y., Li, H., Wang, X., 2012. A modification to moea/d-de for multiobjective optimization problems with complicated pareto sets. Information Sciences 213 (2012), 14–38.

Tanaka, M., Tanino, T., 1992. Global optimization by the genetic algorithm in a multiobjective decision support system. In: Proceedings of the Tenth International Conference on Multiple Criteria Decision Making. Vol. 2. pp. 261–270.

Tang, Y., Reed, P., Wagener, T., Van Werkhoven, K., et al., 2007. Comparing sensitivity analysis methods to advance lumped watershed model identification and evaluation. Hydrology and Earth System Sciences 11 (2), 793–817.

Thiele, L., Miettinen, K., Korhonen, P., Molina, J., 2009. A preference-based evolutionary algorithm for multi-objective optimization. Evolutionary Computation 17 (3), 411–436.

Touchette, P. E., MacDonald, R. F., Langer, S. N., 1985. A scatter plot for identifying stimulus control of problem behavior. Journal of Applied Behavior Analysis 18 (4), 343.

Ulungu, E., Teghem, J., Fortemps, P., Tuyttens, D., 1999. Mosa method: a tool for solving multiobjective combinatorial optimization problems. Journal of Multi-Criteria Decision Analysis 8 (4), 221–236.

Van der Maaten, L., Postma, E., Van Den Herik, H., 2009. Dimensionality reduction: A comparative review. Journal of Machine Learning Research 10, 1–41.

Van Veldhuizen, D., 1999. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Tech. Rep. AFIT/DS/ENG/99-01, Air Force Institute of Technilogy.

Veldhuizen, D., Lamont, G., 2000. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. Evolutionary computation 8 (2), 125–147.

Voget, S., Kolonko, M., 1998. Multidimensional optimization with a fuzzy genetic algorithm. Journal of Heuristics 4 (3), 221–244.

Wagner, T., Beume, N., Naujoks, B., 2007. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 742–756.

Wang, J., Terpenny, J., 2003. Interactive evolutionary solution synthesis in fuzzy set-based preliminary engineering design. Journal of intelligent manufacturing 14 (2), 153–167.

Wang, R., Fleming, P., Purshouse, R., 2014. General framework for localised multi-objective evolutionary algorithms. Information Sciences 258 (2), 29–53.

Wang, R., Purshouse, R., Fleming, P., 2013. Preference-inspired Co-evolutionary Algorithms for Many-objective Optimisation. IEEE Transactions on Evolutionary Computation 17 (4), 474–494.

Wickramasinghe, U., Li, X., 2009. Using a distance metric to guide pso algorithms for many-objective optimization. In: GECCO 2009: Proceedings of the Genetic and Evolutionary Computation Conference. ACM, Montreal,Canada, pp. 667–674.

Wickramasinghe, W., Li, X., 2008. Integrating user preferences with particle swarms for multi-objective optimization. In: GECCO 2008: Proceedings of the Genetic and Evolutionary Computation Conference. Altanta, Georgia, USA, pp. 745–752.

Yang, Q., Ding, S., 2007. Novel algorithm to calculate hypervolume indicator of pareto approximation set. In: Advanced Intelligent Computing Theories and Applications: With Aspects of Contemporary Intelligent Computing Techniques: Third International Conference on Intelligent Computing, ICIC 2007. Vol. 2. Qingdao, China, p. 235.

Yaochu Jin, M. O., Sendhoff, B., 2001. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? In: Proceedings of Genetic and Evolutionary Computation Conference. San Francisco, USA, pp. 1042–1049.

Zhang, Q., Li, H., Dec. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation 11 (6), 712–731.

Zhang, Q., Li, H., Maringer, D., Tsang, E., 2010. Moea/d with nbi-style tchebycheff approach for portfolio management. In: Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, pp. 1–8.

Zhang, Q., Liu, W., Li, H., 2009. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In: Evolutionary Computation (CEC), 2009 IEEE Congress on. IEEE, pp. 203–208.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., Zhang, Q., Mar. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1 (1), 32–49.

Zitzler, E., Brockhoff, D., Thiele, L., 2007. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: Evolutionary Multi-Criterion Optimization. Springer, pp. 862–876.

Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation 8 (2), 173–195.

Zitzler, E., Künzli, S., 2004. Indicator-based selection in multiobjective search. In: Parallel Problem Solving from Nature – PPSN VIII. Springer, pp. 832–842.

Zitzler, E., Laumanns, M., Thiele, L., 2002. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems EUROGEN 2001 3242 (103), 95–100.

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

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V., 2003. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 7 (2), 117–132.