User-Preference Based Evolutionary Algorithms for Many-Objective Optimisation

A thesis submitted for the degree of
Doctor of Philosophy

W. R. M. U. K. Wickramasinghe B.Sc. (Hons.), M.Sc.
School of Computer Science and Information Technology,
College of Science, Engineering and Health,
RMIT University,
Melbourne, Victoria, Australia.

18^{th} October, 2010

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

W. R. M. U. K. Wickramasinghe
School of Computer Science and Information Technology
RMIT University
18^{th} October, 2010

Acknowledgements

The compilation 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 thank my senior supervisor Dr. Xiaodong Li, School of Computer Science and I.T.; for his guidance and support given to me throughout the research. I thank him for the time and effort spent in conducting this research and providing valuable guidance all the way.
- I would like to thank my second supervisor Assoc. Prof. Vic Ciesielski, School of Computer Science and I.T.; for his invaluable input and guidance during this research.
- I would like to thank Mr. Robert Carrese, School of Aerospace Mechanical and Manufacturing Engineering; for helping me a lot during aerofoil design phase of this research and also providing valuable input, which was immensely useful for the successful completion of this thesis.
- I would like to thank Mr. Steven Burrows, School of Computer Science and I.T.; for being my *thesis buddy*. The feedback and discussions we had by exchanging chapters were very valuable for a successful compilation of this thesis.
- I would have not been able to do anything if it was not for the encouragement and support of my wonderful wife, Jayani. I thank her for proof reading this thesis and being my moral support throughout my academic career.
- I would like to give a special thank to Optical Research Associates for providing an academic license for the CODE V design tool.
- Last but not least I would like to thank all my fellow colleagues who helped me during this research and I'm deeply grateful for all the support to make this research a success.

I was in receipt of the International Postgraduate Research Scholarship (IPRS) and the RMIT University International Scholarship (RUIS) during the research period.

Dedication

This I dedicate to my wife Jayani...

Credits

Portions of the material in this thesis have previously appeared in the following publications:

- Upali K Wickramasinghe and Xiaodong Li. Integrating User Preferences with Particle Swarms for Multi-objective Optimization. In *Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO)*, Atlanta, July 2008.

- Upali K Wickramasinghe and Xiaodong Li. Choosing leaders for multi-objective PSO algorithms using Differential Evolution. In *Proceedings of the Simulated Evolution And Learning Conference (SEAL)*, Melbourne, December 2008.

- Upali K Wickramasinghe and Xiaodong Li. Using a Distance Metric to Guide PSO Algorithms for Many-Objective Optimization. In *Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO)*, Montreal, July 2009.

- Upali K Wickramasinghe and Xiaodong Li. A Distance Metric for Evolutionary Many-Objective Optimization Algorithms using User-Preferences. In *Proceedings of the Australasian Joint Conference on Artificial Intelligence (AI)*, Melbourne, December 2009.

- Upali K Wickramasinghe, Robert Carrese and Xiaodong Li. Designing Airfoils using a Reference Point based Evolutionary Many-objective Particle Swarm Optimization Algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Barcelona, July 2010.

- Robert Carrese, Upali K Wickramasinghe, Hadi Winarto, and Jon Watmuff. Preference-Based Optimization for Transonic Airfoil Shape Design over a Multi-Operational Spectrum. *Journal of Aircraft from American Institute of Aeronautics and Astronautics (AIAA)*, 2011.

- Upali K Wickramasinghe and Xiaodong Li. Optical Lens Design using a Reference Point Many-objective Particle Swarm Algorithm. In *Proceedings of the ACM Genetic and Evolutionary Computation Conference (GECCO)*, Dublin, July 2011 (under review).

- Upali K Wickramasinghe, Xiaodong Li and Robert Carrese. A Distance Metric based on User-Preferences for Evolutionary Many-Objective Optimisation Algorithms. In *IEEE Transactions on Evolutionary Computation*, 2011 (under review).

Note

Unless otherwise stated, all fractional results have been rounded to the displayed number of decimal figures.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

d–EMO	distance metric based EMO algorithm
AP	Aspiration Point
DE	Differential Evolution
DM	Decision Maker
DST	Distortion
DTLZ	Deb, Thiele, Laumanns and Zitzler test suite
EA	Evolutionary Algorithm
EFL	Effective Focal Length
EMO	Evolutionary Multi-objective Optimisation
ES	Evolution Strategies
FNO	F-Number
GA	Genetic Algorithm
GP	Genetic Programming
HV	Hyper-Volume
LB	Lower Bound
MCDM	Multi-Criteria Decision Making
MCO	Multiple Criteria Optimisation

MDEPSO	The hybrid multi-objective DE and PSO algorithm
MOPSO	Multi-Objective Particle Swarm Optimisation
NLF	Natural Laminar Flow
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NSPSO	Non-dominated Sorting Particle Swarm Optimisation
OAL	Overall Length
PSO	Particle Swarm Optimisation
RP	Reservation Point
SA	Spherical Aberration
SBX	Simulated Binary Crossover
UAV	Unmanned Aerial Vehicles
UB	Upper Bound
WFG	Walking Fish Group test suite
ZDT	Zitzler, Deb and Thiele test suite

Nomenclature

α_{TE}	PARSEC aerofoil parameter for trailing edge direction
β_{TE}	PARSEC aerofoil parameter for wedge angle
χ	Constriction factor in Constriction Type 1″ PSO
δ'	Value to define a volume in a preferred region for HV calculation
δ	Spread value for the distance metric
Δ_{TE}	PARSEC aerofoil parameter for trailing edge thickness
Λ	Lebesgue measure
ϕ	Random numbers in PSO functions
$\prec$	Dominance relation
$\preceq$	Weakly dominance relation
ρ	Augmentation coefficient in light beam search
φ	Constant used in the Constriction Type 1″ PSO
$\vec{\lambda}$	Weight vector in light beam search
$\vec{\bar{z}}$	Reference point
$\vec{p_g}$	Global best of a particle's neighbourhood
$\vec{p_i}$	Personal best of a particle
$\vec{v}$	Velocity of a particle

$\vec{x}$	Solution vector/Position of a particle
$\vec{x}_{nad}$	Nadir point in the objective-space
$\vec{z^c}$	Middle point in light beam search
$\vec{z^r}$	Aspiration point in light beam search
$\vec{z^v}$	Reservation point in light beam search
C_d	Drag coefficient of an aerofoil
C_l	Lift coefficient of an aerofoil
C_m	Moment coefficient of an aerofoil
CR	Crossover ratio in DE/rand/1/bin scheme
CUY_{Si}	Curvature of the i^{th} surface of a lens
$dist()$	distance metric
F	Scaling factor in DE/rand/1/bin scheme
f	Objective function
$f_i^{\max}$	Maximum of i^{th} objective
$f_i^{\min}$	Minimum of i^{th} objective
$hv(B)$	HV metric of the solution front B
k	Number of position variables in the WFG test suite
l	Number of distance variables in the WFG test suite
M	Number of objectives
Ma	Mach number
N	Number of decision variables
n	Population size
P	Parent population

Q Offspring population

r_{LE} PARSEC aerofoil parameter for the leading edge radius

$rand(0, 1)$ Random number generator between [0, 1]

Re Reynolds number

S Outranking relation

t Generation/Iteration number

t_{MAX} Maximum nmber of generations/iterations

THI_{Si} Thickness of the i^{th} surface of a lens

w Weight for an objective

x_{LO}, z_{LO} PARSEC aerofoil parameters for lower crest locations

x_{UP}, z_{UP} PARSEC aerofoil parameters for upper crest locations

xx_{LO} PARSEC aerofoil parameter for lower crest curvature

xx_{UP} PARSEC aerofoil parameter for upper crest curvature

z_{TE} PARSEC aerofoil parameter for trailing edge coordinate

AX Axial colour aberration of a lens

IMGDIST Distance from the last surface of a compound lens to the focal plane

LAT Lateral colour aberration of a lens

PTB Third order Petzval blur aberration of a lens

PTZ Petzval surface curvature of a lens

SAS Third order sagittal astigmatic aberration of a lens

TAS Third order tangential astigmatic blur aberration of a lens

TCO Third order tangential coma aberration of a lens

Abstract

Evolutionary Algorithms (EA) have enjoyed great success in finding solutions for multi-objective problems that have two or three-objectives in the past decade. The majority of these Evolutionary Multi-objective Optimisation (EMO) algorithms explored the decision-space using the selection pressure governed by dominance ranking methods. These ranking methods were based on the dominance relation, which was used to compare solutions in multi-objective problem instances. These EMO algorithms are effective in locating the entire Pareto front of solutions in one execution run. However, they have not been very successful for problem instances having more than three objectives, usually named as *many-objective* problems. The main reason behind this shortcoming is the fact that the dominance comparison becomes ineffective as the number of objectives increases. Therefore, these EMO algorithms can no longer provide the required selection pressure, stagnating the optimisation process.

In this thesis, we present some user-preference methods that were originally developed in multi-criteria decision making. We incorporate these preference methods into EMO algorithms to enhance their ability to handle many-objective problems. To this end, we introduce a distance metric derived from user-preference schemes such as the reference point method and light beam search. This distance metric is used to guide the population of an EMO algorithm to locate solutions within certain areas of the objective-space. These areas indicated by the Decision Maker (DM) are known as preferred regions. In our distance metric approach, the DM is allowed to specify the amount of spread of solutions along the front as well. We name this distance metric based EMO algorithm as d–EMO, which is a generalised framework that can be constructed using any EA. We illustrate using empirical results that EMO algorithms based on d–EMO are effective in locating solutions in these preferred regions on various many-objective problems. This distance metric approach is also computationally less expensive as it does not rely on dominance ranking methods.

Some examples of user-preference EMO algorithms are found in the current literature and they have had variable degrees of success on the many-objective problems. One key issue that remains to be resolved is that there are no suitable metrics for comparing the performance of these user-preference EMO algorithms. Therefore, we introduce a variation of the normalised Hyper-Volume (HV) metric that is suitable for comparing user-preference EMO algorithms. The key feature in our HV calculation process is to consider only the solutions within each preferred region. This methodology favours user-preference EMO algorithms that have converged closely to the Pareto front within a preferred region. We also demonstrate how this HV calculation process can be used to visualise the progress of the algorithms in converging onto the preferred regions.

Particle Swarm Optimisation (PSO) has been a popular search-strategy employed in EMO algorithms because of their fast convergence property. In this thesis, we introduce a multi-objective PSO algorithm named MDEPSO. This PSO algorithm is less prone to getting stuck in local optimal fronts and still retains its fast convergence ability. In MDEPSO, this feature is achieved by generating leader particles using a differential evolution rule rather than picking particles directly from the population or an external archive. We also extend MDEPSO to include the distance metric to handle some real-world many-objective problem instances.

We have identified two real-world engineering design problems in optimising aerofoil and lens designs, and formulated them as many-objective problems. The optimisation process of these many-objective problems is computationally expensive. Hence, we use a reference point MDEPSO algorithm to locate solutions effectively in fewer function evaluations. The main feature of the optimisation process of these aerofoil and lens design problems is the derivation of reference points based on existing designs. We illustrate how these existing designs can be used to either obtain better or new design solutions that correspond to various requirements. This process of deriving reference points based on existing design models, and integrating them into a user-preference EMO framework is a novel approach in the optimisation process of such computationally expensive engineering design problems.

Chapter 1

Introduction

Optimisation is regarded as a branch of *numerical mathematics*, *operational research* or *systems engineering design* [Wilde, 1964]. In numerical optimisation, the task is to find the optimal value (or values) for a given problem instance that reflects a desired output [Schwefel, 1993]. The process of optimisation may involve finding the maximum or minimum values for a mathematical function or the best values for the decision variables that are used to define a problem. The traditional methods used to finding the optimal values for mathematical functions involve solving a set of equations [Schwefel, 1993]. However, the problems found in disciplines like engineering design, economics and operational research has the need to find the decision variable values so that the desired *outcome* is optimal. We refer to these outcomes as *objectives* in this thesis. Therefore, the function that maps the decision variables to an objective is called an *objective function.*

The traditional mathematical process of optimising an objective function involves selecting some approximate values for the decision variables and computing values for the objectives [Boyd and Vandenberghe, 2004]. This process is applied for many iterations changing the decision variable values. The aim is to minimise the deviations of the variables so that the output is optimal. There exist several mathematical algorithms like the *Newton-Raphson iteration* that make this process simpler to some extent [Schwefel, 1993]. However, Evolutionary Algorithms (EAs) have recently become a popular approach in the optimisation process of such parameter-based problems [Eiben and Smith, 2003].

In an EA, a population of individuals is used to represent candidate solutions each encoding some variable values. EAs are built on the theory of evolution by Darwin [2003 (first published in 1859], which follows the *survival of the fittest individual* principle. These individ-

uals are then *evolved* in an optimisation process which spans several *iterations* [Michalewicz, 1994]. In an EA, the individuals which represent the best solutions are used to influence the population to produce better solutions in subsequent iterations. The individuals of a population will generate new candidate solutions at each iteration based on rules that influence the creation of *fitter* individuals than before. Therefore, the population gets fitter over the successive iterations, which corresponds to achieving better solutions. The use of a population based approach like EAs makes the optimisation process more effective, because within a single execution of the algorithm, a large area of the search-space can be explored in parallel. Problem instances where only one objective function is being optimised are usually referred to as *single-objective* problems.

Many examples in the EA literature can be found where single-objective problems have been solved using algorithms mimicking the evolutionary process [Eiben and Smith, 2003; Michalewicz, 1994]. However, there are many problems that can be perceived to have more than one objective function to be optimised; often these objectives are conflicting with each other. These problems are named as *multi-objective* problems [Sawaragi et al., 1985]. In multi-objective optimisation, all the objective functions are optimised simultaneously. This is more challenging than single-objective problems because the optimisation process has to consider trade-offs between the objectives in order to obtain *optimised solutions*. There exist many traditional mathematical methods for optimising multi-objective problems, which are developed in Multiple Criteria Optimisation (MCO) [Ehrgott, 2000; Steuer, 1989]. However, in this thesis we are interested in exploring EAs in multi-objective optimisation. These EA-based approaches are termed Evolutionary Multi-objective Optimisation (EMO) [Deb, 2008].

EMO algorithms have enjoyed great popularity in the last decade where many publications concentrated on two and three-objective problems [Coello Coello et al., 2007; Deb, 2001]. However, there have been relatively little research work in problem instances with more than three objectives because the most popular EMO algorithms perform poorly on such problems. Therefore, in this thesis, we explore optimisation methods suitable for problem instances having more than three objectives, which are usually referred to as *many-objective* problems [Hughes, 2005]. We explore the possibilities of integrating new optimisation procedures into EMO algorithms such that they can find solutions on many-objective problems more effectively than the current state-of-the-art EMO algorithms.

1.1 Motivation

The state-of-the-art EMO algorithms operate based on the *dominance* relationship, which is used to compare candidate solutions to advance the evolutionary process [Deb, 2001]. In the dominance relation, an individual is said to be fitter than another, if its objective values are equal or better than the latter for all objective functions. However, if the individual has better values on some objectives and worst values on the other objectives, then the two individuals are said to have an equal fitness. In many-objective problem instances the probability of the entire population being equally fit to each other becomes very high, deeming the dominance relation ineffective to be used in the optimisation process [Wagner et al., 2007]. Therefore, we are motivated to explore methods that can be used within the optimisation process that are still effective in comparing the individuals in many-objective problem instances.

Another issue in many-objective problems is that, as the number of objectives increases, the size of the objective-space also increases dramatically. EMO algorithms would have to spend a considerably large amount of resources to locate all the possible solutions in this vast area. This can be a waste of resources if all the solutions are not useful for a given application. To address these issues, many methods are proposed in the EMO literature that approach the optimisation process in different forms. One approach is to use dimensionality reduction techniques to reduce the number of objectives [Brockhoff and Zitzler, 2006; Zhang and Li, 2007]. These problems with reduced number of objectives can be solved successfully with an EMO algorithm that uses dominance comparisons. However, there exist some instances where such an approach to reduce the number of objectives itself becomes an optimisation problem. Therefore, we are more interested in EMO algorithms that can handle many-objective problems directly rather than using dimensionality reduction methods.

Some of examples of real-world problems having many conflicting objectives are found in land planning [Stewart et al., 2004], construction engineering [Nakayama et al., 1995], aerospace engineering [Hughes, 2006; Takagi, 2001], optical engineering [Joseph, 2005] and water engineering [di Pierro, 2010]. These applications were another motivation factor for us to explore methods to make EMO algorithms better suited to handle many-objective problems. In this thesis, we also present two many-objective engineering design problems for which we use an EMO process to obtain useful solutions efficiently.

There exist few EMO algorithms in the literature that are able to handle some many-objective problem instances. However, they still depend on *dominance ranking* [Kukkonen and Lampinen, 2007] and *modified dominance relations* [Köppen et al., 2005] based methods.

These dominance ranking and modified dominance relation methods are computationally expensive and ineffective as the number of objectives increases. Therefore, we are interested in applying methods that do not depend on any dominance based relations. One of the recent approaches that has been gaining popularity in handling many-objective problems is to use preference mechanisms found in the Multi-Criteria Decision Making (MCDM) [Ehrgott and Gandibleux, 2002] literature within an EMO algorithm. We are motivated to integrate some of these user-preference methods into EMO algorithms, so that they provide additional guidance to the algorithms to locate solutions in many-objective problems. This guidance will allow the EMO algorithms to concentrate the search effort into preferred regions of the objective-space to locate solutions. To this end, we propose a *distance metric* derived from user-preferences that is used to guide the EMO algorithms. This distance metric based approach is simpler, yet more efficient than the current user-preference algorithms that rely on dominance relations.

Another open issue is that there are no performance metrics, which have been universally agreed upon to compare user-preference EMO algorithms meaningfully. Existing performance metrics for EMO algorithms cannot be directly applied to user-preference EMO algorithms. These metrics measure the effectiveness of the algorithms in locating all solutions on the Pareto front but not the targeted areas as indicated by the user. In the research presented in this thesis, we propose a performance metric more suitable for user-preference EMO algorithms. The proposed metric is a result of the background study of current metrics and their shortcomings with relation to the characteristics of user-preference EMO algorithms.

Among the many EAs in the literature, Particle Swarm Optimisation (PSO) [Kennedy and Eberhart, 2001] algorithms have been very popular due to their simple functionality, requiring only a few execution parameters. EMO algorithms that use PSOs as the search strategy are usually named as Multi-Objective Particle Swarm Optimisation (MOPSO) [Coello Coello and Lechuga, 2002] algorithms. We are interested in coupling user-preferences with MOPSO algorithms for many-objective problems. MOPSO algorithms are also very fast in finding solutions compared with other EMO algorithms. Therefore, we are interested in using a MOPSO algorithm, which is guided by our user-preference based distance metric to find solutions in some real-world engineering design applications. The applications we are interested in, are in the domains of aerodynamic optimisation and optical lens design. Another motivation factor for real-world applications is the exploration of methods to integrate the refined knowledge of the designers to guide the optimisation process.

These motivation factors are the main driving forces of the research presented in this thesis. Next, we outline our research work guided by some research questions and the research methodology.

1.2 Research questions

The research described in this thesis is based on the following questions:

1. *How can we integrate user-preference methods from the MCDM literature to design efficient EMO algorithms that can find solutions for many-objective problems?*
   We use some existing preference methods found in the MCDM literature to build a framework to aid the functionality of EMO algorithms in solving many-objective problems. The proposed distance metric will be an integral part of this framework. The key merits of this framework are its simplicity and the absence of a dominance ranking search method. Not having a dominance ranking method is essential to make this framework computationally less expensive than the state-of-the-art EMO algorithms.

2. *How can we measure the performance of a user-preference EMO algorithm using a suitable performance metric?*
   In the current literature of user-preference EMO algorithms, there is an absence of a suitable performance metric. We explore possible metrics and adapt them in a manner meaningful for user-preference EMO algorithms.

3. *How can we develop a robust PSO algorithm which can be used in many-objective problems?*
   The current PSO algorithms are very fast in converging onto solutions in multi-objective problem instances. However, they can also get stuck in local optimal solutions. We explore methods to retain the fast convergence of PSO algorithms even on many-objective problem instances by improving the guidance with a new leader selection mechanism and user-preferences.

4. *How can real-world problems benefit from user-preference EMO algorithms in the optimisation process of engineering designs?*
   Practitioners of real-world domains have a very refined knowledge about the problem instances. We harness this knowledge to provide guidance to EMO algorithms so that they can be used in the optimisation process of engineering design problems. These

problems often can be formulated to have a large number of objectives. We explore methods to integrate the guidance from practitioners into a framework based on user-preference EMO algorithms.

Each of these research questions is described in detail and examined in the upcoming chapters. In the following section we present several techniques adopted to address the above research questions.

1.3 Methodology

To answer the research questions we first need to have a good understanding of the state-of-the-art EMO algorithms. This is achieved by a comprehensive literature review that describes the current EMO algorithms and their weaknesses in handling problems with a large number of objectives. In particular, we explore existing preference mechanisms in the MCDM literature that have been applied to EMO algorithms. Based on these approaches, we present a framework for EMO algorithms that is more efficient and effective than the state-of-the-art EMO algorithms for solving many-objective problems.

We use some existing test problem suites found in the EMO literature to evaluate the algorithms that we propose in this thesis. These test problem suites include multi-objective problems having a wide range of characteristics. Initially we compare our preference-based algorithms against standard dominance ranking-based algorithms like NSGA-II [Deb et al., 2002] on these problems. We assess the capability of these algorithms to find the global optimal solutions using visual representation of the solutions and analyse them against known properties of the problems.

To analyse our preference-based algorithms more effectively against other preference-based algorithms suitable performance metrics are required. The existing performance metrics are developed to measure how well an EMO algorithm locates solutions on the entire Pareto front. However, in user-preference EMO algorithms the goal is to locate solutions on certain regions of the Pareto front. Therefore, the performance metric has to take this information into account in the measurement process to obtain meaningful results. Most of the existing performance metrics measure the deviation of the solutions against known Pareto optimal values for each problem instance. This feature makes the existing metrics unsuitable because there exists many problems where the global optimal solutions are not known. We explore methods to use a hyper-volume metric [Knowles and Corne, 2002] in a suitable manner for preference-based EMO algorithms. The hyper-volume metric is suitable because it

does not require the knowledge of a known Pareto front. However, the current hyper-volume metric is used to measure how well an EMO algorithm locates the entire Pareto front. Therefore, it has to be modified to better suit the performance measure of user-preference EMO algorithms.

In EMO algorithms, performance metrics are presented as averages to mitigate the random chance in the stochastic process. Therefore, we have to evaluate these average hyper-volume values using some statistical tests. We use the Student's t-test to test the significance of the results because we execute different algorithms on the same test problems to observe which one is better in locating solutions. The null hypothesis is defined as; *the observed mean of each population is different*. The null hypothesis is rejected if the t-test yields a p-value greater than 0.05.

To explore the possibility to use a user-preference EMO algorithm in engineering design problems, we use a robust MOPSO algorithm to take advantage of its fast convergence behaviour. However, the current MOPSO algorithms have to be enhanced to be able to handle many-objective problems with a large number of local optimal solutions. Therefore, we first explore the possibilities of obtaining a PSO algorithm, which is robust in handling difficult multi-objective problems, and then extend it to handle many-objective problems with the use of user-preferences.

We apply this preference-based MOPSO algorithm to two engineering design problems. The problems we present in this thesis are; aerofoil designs for unmanned aircraft and lens designs for still photography. We use existing knowledge of these problem domains to develop some preferences that will be used within the MOPSO algorithms. Finally, we present the optimised designs for each problem instance and evaluate them against existing solutions and other preferred characteristics of the designs. These real-world applications are important for the completeness of this thesis because it demonstrates the practical relevance of the proposed user-preference EMO algorithms and provides insights on utilising the refined domain knowledge of the engineers to guide EMO algorithms.

1.4 Original contributions

This thesis makes a number of novel contributions to the field of EMO, in particular many-objective optimisation. The core of these contributions is on employing user-preference methods to aid EMO algorithms to solve problems having a large number of objectives. The following are the key contributions of this thesis.

- A simple yet effective framework to aid EMO algorithms to function in many-objective problems with the use of user-preferences. This framework is based on a *distance metric* derived from the user-preference, which can be embedded into any EMO algorithm. This distance metric is a scalar value that can be used successfully to compare solutions in many-objective problems to provide the necessary selection pressure for the EMO algorithms to work effectively. This is advantageous over EMO algorithms based on dominance ranking methods because of the ineffectiveness of the dominance relation in many-objective problems.

- A performance metric based on the hyper-volume measurement that is suitable for comparing user-preference EMO algorithms. We also present a discussion of the shortcomings of the current performance metrics if used for user-preference EMO algorithms.

- A robust and effective MOPSO algorithm that can handle difficult multi-objective problems. This MOPSO algorithm is integrated with the distance metric to obtain an effective MOPSO algorithm which is more efficient than the state-of-the-art.

- Applications of this MOPSO algorithm to aerofoil designs for unmanned aircraft and lens designs for still photography. Having user-preferences in such engineering design problems can provide guidance to the EMO algorithm to locate useful solutions reducing valuable computational expenses. Current aerospace engineering or lens design applications do not use user-preference EMO algorithms in their design processes. The real-world engineering applications presented in this thesis are a first attempt of using such optimisation methods.

These contributions are aligned with the research questions and are described within the next chapters.

1.5 Outline of the thesis

This thesis has seven chapters and an Appendix. The chapters are organised as follows:

- A literature review of the state-of-the-art EMO algorithms is given in Chapter 2. Here the basics for multi-objective optimisation are described with example problem instances. Next, we present a review of EAs and how they are used in solving multi-objective problems. We also review preference mechanisms in MCDM and recent work using them in EAs in this chapter.

- We describe the use of user-preferences to derive the distance metric in Chapter 3. This distance metric is used to guide evolutionary algorithms to find solutions in many-objective problem instances, hence remove the need of dominance-ranking methods. We present results of applying this distance metric on many problem instances having a large number of objectives and compare them with algorithms that use dominance-ranking methods. We answer the first research question in this chapter.

- We answer the second research question in Chapter 4 by proposing a better performance metric for user-preference algorithms. We first discuss some existing performance metrics and their shortcomings, and then introduce a modification to the hyper-volume metric as a suitable performance metric for user-preference algorithms.

- In Chapter 5, we present a new MOPSO algorithm that is capable of exploring the search-spaces of many-objective problems without getting stuck in local optimal solutions. We use preferences to guide the algorithm to find solutions efficiently in these problem domains by harnessing the fast converging property of PSO. We present some results of this MOPSO algorithm in various test problem instances in comparison with some other MOPSO algorithms found in the literature. We present answers to the third research question in this chapter.

- In Chapters 6 and 7 we present the two engineering design problems. Within these chapters we aim to answer the fourth research question. The engineering problems are in aerofoil designs for unmanned aircraft and lens designs for still photography. The MOPSO algorithm described in the previous chapter is used in a manner where an engineer can guide the algorithm using their refined knowledge about the problem domain. The preferences will guide the algorithms to find solutions adhering to the design criterion.

- Finally in Chapter 8 we summarise the outcomes of this thesis with conclusions and discussions for future work. We revisit the research questions and summarise the answers with respect to the results in Chapters 3–7.

The Appendix describes the many-objective test problems used in this thesis in detail, followed by the Bibliography of cited publications and other references.

Chapter 2

Literature review

There exist many problems in engineering design, which can be formulated to have several conflicting objectives. Finding good designs satisfying all these objectives is challenging, and many algorithms have been developed to find such design solutions. In this thesis, we propose some EAs that are able to find solutions for such problems. However, before we get into the details of the algorithms it is important to describe the background material related to the research presented in this thesis.

In this chapter we provide a literature review of all the *building blocks* used for the research. This review illustrates the limitations of the state-of-the-art algorithms and provides motivation for this research. First, in Section 2.1, we introduce multi-objective problems using the standard definitions. We also present the definition of many-objective optimisation, which is the crux of this research. Next, in Section 2.2, some existing multi-objective test problems are described. These test problems are used in the literature as means of comparing the effectiveness of multi-objective algorithms. Section 2.3 shows the traditional approach in solving multi-objective problems. We then describe the shortcomings of this mathematical approach and the justification for the use of an EA approach.

We briefly describe the concept of EAs in Section 2.4, with some examples. Next, we describe in detail how EAs have been used to develop EMO algorithms in Section 2.5. This section illustrates the issues in current EMO algorithms when faced with many-objective problems. We also provide some techniques found in the literature that are used to overcome these issues. One such approach is to integrate *preferences* into EMO algorithms. Section 2.6 details these preference methods and integration into EMO algorithms. This chapter also describes some of the existing performance metric for EMO algorithms in Section 2.7.

2.1 Multi-objective optimisation

In multi-objective optimisation, the main aim is to find solutions which have some trade-offs between multiple conflicting objectives. These objectives have to be optimised simultaneously. We can formally define such an optimisation as; there is a set of objective functions which need to be maximised or minimised [Deb, 2001]. Without loss of generality we assume that the objectives are to be minimised. This can be generally defined as a set of functions described by:

$$\text{minimise } f_i(\vec{x}) \text{ where } i = 1, 2, \ldots, M \tag{2.1}$$

Here, $f_i(\vec{x})$ is an objective function and M is the number of objectives. The solution $\vec{x} = [x_1, \ldots, x_N]$ is a vector having N number of *decision variables*. For example, assume the problem of designing of beams that are used to build a structure. Here, the decision variables can be the dimensions of the beam (x_1 = height, x_2 = width and x_3 = depth). We are interested in finding an optimal design for a beam which gives the most strength with the least amount of material. Here, our objectives can be f_1 = strength and f_2 = weight. It is also interesting to note that these two objectives are conflicting with each other.

2.1.1 Variable boundaries

The decision variables can be defined to be valid between some extreme values. In the beam design problem, these extreme values can be the maximum and minimum values allowed for each dimension. We define these extreme values as the Upper Bound (UB) and Lower Bounds (LB) of the decision variables. These boundaries can be expressed as:

$$x_i^{(LB)} \leq x_i \leq x_i^{(UB)} \text{ or } x_i \in \left[x_i^{(LB)}, x_i^{(UB)}\right] \text{ for } i = 1, 2, \ldots, N \tag{2.2}$$

2.1.2 Constraints

Each optimisation problem can have a certain number of constraints. These are to be satisfied in the design process. In the beam design problem, a constraint could be the stress amount of a beam. A constraint is defined as either an inequality function or an equality function. Generally these constraints are written as:

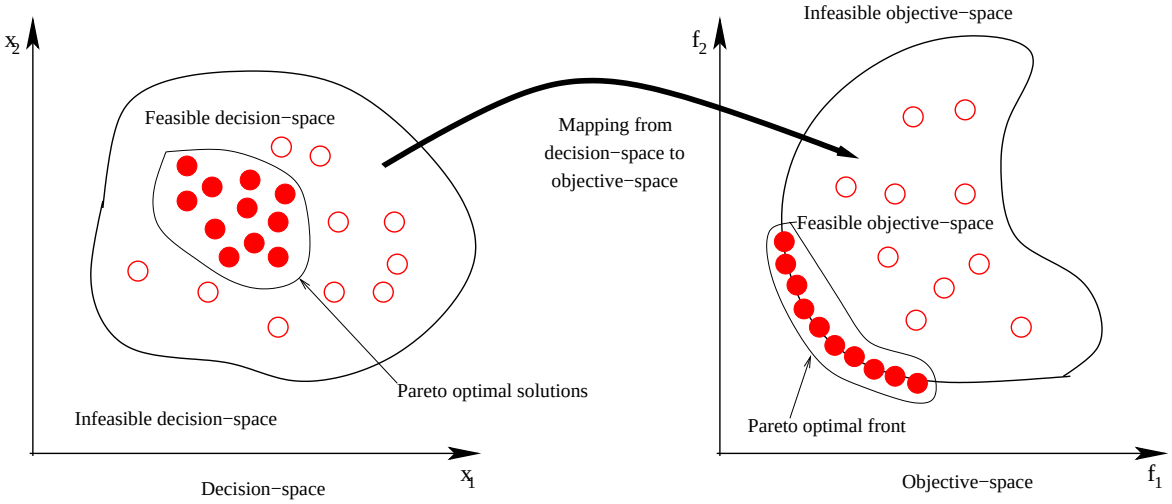


Figure 2.1: Solutions in the decision-space or search-space are mapped to the objective-space using the one-way objective functions.

$$g_j(\vec{x}) \leq 0 \text{ where } j = 1, 2, \ldots, J$$
$$h_l(\vec{x}) = 0 \text{ where } l = 1, 2, \ldots, L \tag{2.3}$$

2.1.3 The mapping from decision-space to objective-space

The decision vectors can be illustrated as points on a coordinate axis plot. The axes represent the decision variables. This plot is then called the *decision-space* (or search-space). The objective functions map each point in this decision-space to a point on a coordinate axis plot called the *objective-space*. The objective-space represents each objective on its axes. Figure 2.1 illustrates this mapping between a two-dimensional decision-space and a two-dimensional objective-space.

Here, solutions within the variable boundaries and adhering to the constraints are referred as the *feasible solutions*. These feasible solutions will be mapped to the feasible objective-space. It is useful to note that in multi-objective problems, the objective functions provide a mapping of the decision variables from the decision-space to the objective-space only. The main task of multi-objective algorithms is to find solutions having suitable values for the decision variables such that they provide the best (or optimal) values for the objective functions. These algorithms function by observing the behaviour of the objective-space, because optimising the objective-space is the main task.

2.1.4 The dominance relation

The *dominance* relation is used to compare two solutions mapped on the objective-space in a multi-objective optimisation problem [Deb, 2001]. For two decision vectors $\vec{a}$ and $\vec{b}$ on the objective-space, the dominance relation $\vec{a}$ *dominates* $\vec{b}$ (denoted by $\vec{a} \prec \vec{b}$) is defined as:

$$\vec{a} \prec \vec{b} \Leftrightarrow \forall i \ f_i(\vec{a}) \leq f_i(\vec{b}) \wedge \exists j \ f_j(\vec{a}) < f_j(\vec{b}) \tag{2.4}$$

The decision vector $\vec{a}$ is said to dominate $\vec{b}$ if and only if, $\vec{a}$ is as good as $\vec{b}$ in all objectives and $\vec{a}$ is strictly better than $\vec{b}$ in at least one objective. If $\vec{a}$ or $\vec{b}$ do not dominate each other, then these vectors are said to be *non-dominant* to each other. The relation: $\vec{a}$ is as good as $\vec{b}$ in all objectives, is a weaker form of the dominance relation. This *weakly dominated* relation (denoted by $\preceq$) is given as:

$$\vec{a} \preceq \vec{b} \Leftrightarrow \forall i \ f_i(\vec{a}) \leq f_i(\vec{b}) \tag{2.5}$$

The set of solutions which are non-dominant to each other is called the *non-dominated set* in the decision-space, and the *non-dominated front* in the objective-space. A decision vector is said to be *Pareto optimal* if and only if there is no other vector in the objective-space that dominates it. This set of Pareto optimal solutions is called the Pareto optimal set in the decision-space and the Pareto optimal front in the objective-space. Figure 2.1 also illustrates these collections.

2.1.5 Many-objective optimisation

Traditionally, most of the multi-objective optimisation problems found in the literature usually have two or three objectives. Problems having more than three objectives are referred to as *many-objective* problems [Fleming et al., 2005; Hughes, 2005]. We provide special attention to these many-objective problems in this thesis, although two or three objectives are used in diagrams for the ease of visualisation.

In the literature, the objective-space of a multi-objective problem is of high interest because the main task of the algorithm is to find solutions that can be mapped onto the Pareto optimal front, which is in the objective-space. In other words, researchers in multi-

Table 2.1: Properties of the ZDT test problems and the characteristics of their Pareto fronts. The number of decision variables N, is constant for each problem.

Problem	N	**Shape of the Pareto front**
ZDT1	30	Convex
ZDT2	30	Concave
ZDT3	30	Convex with disconnected surface
ZDT4	10	Convex
ZDT6	10	Concave

objective optimisation are interested in developing algorithms that can *converge* onto the Pareto optimal front. Patterns emerging on the Pareto front depict various features of a multi-objective problem. These features can also represent the difficulty levels of a problem. A multi-objective algorithm will strive to converge onto a solution front which conforms to those expected features. However, it is difficult to judge if this solution front an algorithm converged to, is Pareto optimal or not without some extra information. To this end, researchers in multi-objective optimisation have developed several multi-objective test problems, where the Pareto fronts are known. These test problems are used to determine if a multi-objective algorithm is capable of converging to the Pareto front or not.

2.2 Multi-objective optimisation test problems

These test problems with known Pareto fronts show various different characteristics on the objective-space for each problem instance. Using these test problems, researchers in multi-objective optimisation are able to develop algorithms that can be used to find solutions to various types of problems, without having the need to modify the algorithm for each problem instance. Here, we present a brief overview of three widely used multi-objective test problem sets. Detailed descriptions of the objective functions are given in Appendix A (page 198).

2.2.1 ZDT test suite

Zitzler et al. [2000] proposed six two-objective test problems. In each of these problems, the objectives are to minimise the given functions. The characteristics of the test problems are given in Table 2.1. The detailed descriptions of the ZDT test problems are given in Appendix A.1 (page 198).

In the ZDT test suite, $x_i \in [0, 1]$ for all the decision variables. However, in ZDT4, $x_1 \in [0, 1]$ and all other $x_i \in [-5, 5]$. ZDT1–3 are simple two-objective problem instances,

Table 2.2: Properties of the DTLZ test problems and the characteristics of their Pareto fronts. The number of decision variables N, depend on the number of objectives M for each problem.

Problem	N	**Shape of the Pareto front**
DTLZ1	$M + 4$	Linear
DTLZ2	$M + 9$	Convex
DTLZ3	$M + 9$	Convex
DTLZ4	$M + 9$	Convex
DTLZ5	$M + 9$	Convex
DTLZ6	$M + 9$	Convex
DTLZ7	$M + 19$	Disconnected regions

each having one global optimal front. ZDT4 has 21^9 local optimal fronts. ZDT4 is designed to test an algorithm's ability to avoid local optimal fronts and locate the global optimal front. Problems with many local optimal and one global optimal front are called *multi-modal* problems. Therefore ZDT4 is a multi-modal multi-objective problem. We have omitted ZDT5 because it is a binary encoded problem, whereas the scope of this thesis is on real-valued problems.

2.2.2 DTLZ test suite

This test problem suite was proposed by Deb et al. [2005b] as an extension to the ZDT problems. The problems in this suite can be scaled to produce any number of objectives. The objectives are to minimise the given functions. Table 2.2 illustrates the properties of the DTLZ problems. The detailed descriptions of the DTLZ test problems are given in Appendix A.2 (page 201).

In the DTLZ test suite, $x_i \in [0, 1]$ for all the decision variables. DTLZ1 is multi-modal having $11^5 - 1$ local optimal fronts. DTLZ3 is multi-modal having $3^{10} - 1$ local optimal fronts. The Pareto fronts of DTLZ5 and DTLZ6 are an area spanning $M - 1$ dimensions in an $M \geq 3$ objective-space. In a three-objective space the Pareto front is a line. In a two-objective space the Pareto front is a line similar to ZDT2. We have omitted DTLZ8 and DTLZ9 because they are constrained problems, which are not in the scope of this thesis.

2.2.3 WFG test suite

This test problem suite was proposed by Huband et al. [2006] addressing some of the limitations found in the ZDT and DTLZ suites. Among these limitations, the notion of *separability* was a key factor. A *separable* problem gives the same Pareto optimal set even if each decision

Table 2.3: Properties of the WFG test problems and the characteristics of their Pareto fronts.

Problem	**Separable**	**Multi-modal**	**Bias**	**Shape of the Pareto front**
WFG1	Yes	No	Yes	Convex with mixed surface
WFG2 (version in the thesis)	No	No	No	Convex with disconnected surface
WFG3	No	No	No	Linear
WFG4	Yes	Yes	No	Concave
WFG5	Yes	No	No	Concave
WFG6	No	No	No	Concave
WFG7	Yes	No	Yes	Concave
WFG8	No	No	Yes	Concave
WFG9	No	Yes and Deceptive	Yes	Concave

variable is optimised separately. Another issue was *bias*. Bias indicates if a problem has a natural favouritism in aiding the search process. If we are to randomly assign values to a large number (about 50,000) of decision vectors and observe that some of them belong to the Pareto set, then this problem is considered to be biased. Another issue identified by Huband et al. [2006] was that the existing multi-objective test problems were not *deceptive*. Basically a deceptive problem is multi-modal and the search-space spans mostly on the local optimal fronts. Such a problem is difficult for multi-objective algorithms because the optimisation process tends to locate a local optimal front. In this thesis, we refer to problems which are non-separable, biased and non-deceptive as *simple* problems. In the ZDT test suite only the ZDT5 problem was a deceptive problem. In our initial experiments, we identified that ZDT1–3 were simple problems because of bias. ZDT4 was not a simple problem, because it was not biased and was multi-modal. In the DTLZ problems, none were deceptive but DTLZ1 and DTLZ3 were multi-modal. We also found that DTLZ2 was biased. Both ZDT and DTLZ test suites did not contain any separable problems [Huband et al., 2006].

The WFG test problem suite addressing these issues is very comprehensive and complex. Here also all objectives are to be minimised. A solution vector $\vec{x}$ consists of k *position variables* and l *distance variables* such that $k + l = N$. All $x_i \in [0, 1]$. In this thesis, we used $M - 1$ position variables and $2(M - 1)$ distance variables giving $N = 3(M - 1)$ number of decision variables, for $M \geq 3$ objectives. For two-objectives we used $k = 2$ and $l = 4$. A brief overview of the test problems are given in Table 2.3. The detailed descriptions of the WFG test problems are given in Appendix A.3 (page 206).

The implementations of the ZDT, DTLZ and WFG test problem suites and the sample Pareto fronts used in this thesis were obtained from the jMetal framework[1] [Durillo et al., 2006].

2.3 Traditional approach in solving multi-objective problems

The traditional mathematical approach to finding solutions in multi-objective problems is to convert the problem into a single objective problem [Miettinen, 2008]. All the objective functions are combined with some weights ($w_i > 0$) to create a single objective function (T) as:

$$T = \sum_{i=1}^{M} w_i f_i(\vec{x}), \text{ where } \sum_{i=1}^{M} w_i = 1 \tag{2.6}$$

Using some algorithm this single objective T function can be optimised. Optimising T will give a single solution. If a user or Decision Maker (DM) is interested in obtaining a set of optimal solutions, which gives the Pareto optimal set, they have to re-run the algorithm with various values for the weights. The main issue in this approach is that finding an optimal set of weight values itself is an optimisation problem. A better method to find solutions for multi-objective problems is to use a population based approach like an EA.

2.4 Evolutionary algorithms

EAs have been used very successfully to find solutions in multi-objective optimisation problems because they use a population of individuals to represent candidate solutions [Deb, 2001]. Before we illustrate some of these algorithms as examples, it is useful to describe the basic functionality of an EA.

2.4.1 General framework of an EA

The basic form of an EA is represented in a cycle, which is shown as in Algorithm 1 [Eiben and Smith, 2003].

[1]The jMetal framework is available at: http://jmetal.sourceforge.net/. This framework also contained the codes for the state-of-the-art evolutionary multi-objective algorithms used in this thesis for comparison. E.g. NSGA-II.

Algorithm 1 Pseudocode of an EA

```
1: INITIALISE a population with random candidate solutions
2: EVALUATE each candidate
3: repeat
4:    SELECT parents
5:    RECOMBINE pairs of parents
6:    MUTATE the resulting offspring
7:    EVALUATE new candidates
8:    SELECT individuals for the next generation
9: until TERMINATION is satisfied
```

The key steps of an EA are *initialise*, *evaluate*, *select*, *recombine*, *mutate* and *termination* [Eiben and Smith, 2003]. In the initialisation stage, candidate solutions are assigned to the individuals by randomly generating values within the variable bounds. These individuals make up the population. A data structure called the *genome* inside an individual represents a candidate solution. The fitness value of an individual is the quality of its genome. In an algorithmic sense this fitness shows how close an individual is to an optimal solution. These candidate solutions may not be the desired solution or the optimal solution, but using recombination (*crossover*) and mutation the fitness values may be improved so that the individuals may reach the desired solution after many iterations.

In recombination, parent individuals create a new individual with a combination of their information. A recombination actually refers to the combination process of these genomes of each parent to generate a new individual (*offspring*). After the recombination, certain decision variables are changed using an operator referred to as *mutation*. This mutates (*changes*) the resulting genome. Mutation is an important step in the EA cycle since this increases the possibility of jumping out of locally optimal solutions. Whereas crossover is a binary operator, mutation is a unary operator.

Within the evaluation stage an individual is assessed (*evaluated*) and assigned a fitness value. This is an important step, since, to select individuals to be parents or to survive, the fitness value is considered. This stage of evaluating an individual's fitness value and selecting the fitter ones is referred to as selection.

As seen in Algorithm 1, these stages are performed on a population in a cyclic manner until a termination criterion is reached. Usually the termination criteria are when a solution is found to the problem, or when a maximum number of evolution cycles (*generations*) are

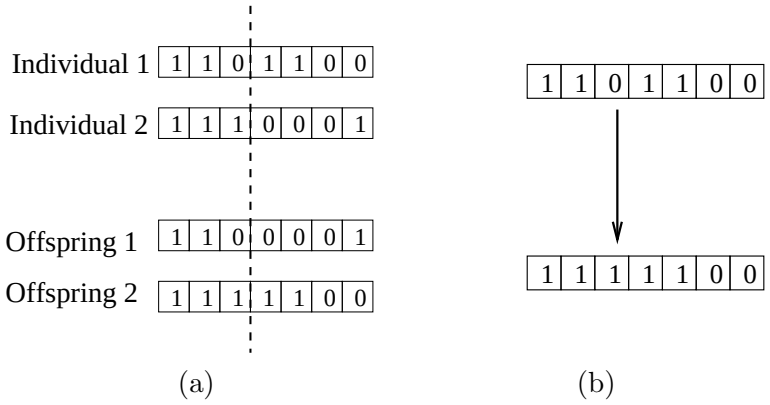


Figure 2.2: Simple crossover and mutation mechanisms of a binary GA: (a) Single point crossover; (b) Single point mutation.

performed. For an EA to function properly several parameters have to be set. These are crossover probability (P_{xor}), mutation probability (P_{mut}) and population size (n), which are problem dependent.

2.4.2 Types of evolutionary algorithms

The literature describes many different types of EAs. We briefly describe some of them illustrating the various crossover and mutation operators.

Genetic Algorithms

The simplest form of EA is in Genetic Algorithms (GA). Holland [1992] described the idea of Darwin's theory of evolution implemented as a form of algorithm, which was later known as GA. In the simplest form of GA, each individual's genome is a binary string comprising of $1s$ and $0s$.

As seen in Figure 2.2, in the crossover operation two individuals (parents) exchange their genome with each other at a common intersection point. This is known as a *single point crossover*. This operation creates two offspring. Depending on a probability, two parents will produce offspring using some crossover mechanism. If the parents are below the probability for crossover, the offspring will simply be a copy of the parents. In mutation, a single binary bit is flipped to mimic a change in the genome. The bit to be flipped is chosen randomly. This is known as a *single point mutation*. As for crossover, a probability is used to decide if an individual is to perform a mutation or not. Other than these simple operations, there

exists other versions of GA with complex crossover and mutation schemes [Eiben and Smith, 2003; Fogel, 1994]. Crossover promotes the obtaining of better solutions using the current information in the population, while mutation provides a mechanism of maintaining diversity in the population. In the simplest form of GA, n number of parents will generate n number of offspring. These offspring will replace the parents forming the next generation. A concept of *elitism* is also available in GAs to promote adding the fittest parents to the next generation, replacing some of the offspring. Extra parameters that have to be set for a GA execution include the P_{xor}, P_{mut}, crossover and mutation schemes, and a percentage for elitism.

Evolution Strategies

In Evolution Strategies (ES), individuals are represented as real-valued vectors in a search-space [Bäck, 1996]. Each vector element is considered as a *variable*. In ES, the main driving force is mutation, whereas in GA the main driving force is crossover. In ES, the mutation is done using a *strategy parameter*. Usually in an ES a Gaussian distribution is used to perturb a parent to generate an offspring. ESs are also able to adapt to the problem by adjusting the strategy parameter.

Genetic Programming

In GAs and ESs, each individual represented a candidate solution in the form of a bit string or vector. With Genetic Programming (GP) each individual represents a *single line program* [Koza, 1992]. The individuals are usually represented in a tree structure. Here, the inner nodes consist of simple mathematical operations or other complex functions. The leaf nodes represent some values or operands. In a GP algorithm, each individual represents a candidate program by combining inner nodes from a set of *functions* and leaf nodes from a set of *terminals*. The fitness of an individual represents the *quality of the program*.

Figure 2.3 illustrates a simple crossover and mutation scheme for GP, similar to GA. However, in GP, the points of crossover and mutation have some restrictions to avoid generating *invalid programs*. For example, crossover can occur only at sub-trees with a function at the root. In mutation, entire sub-trees can be replaced by another sub-tree or a single node and vice-versa. These crossover and mutation operations are carried out by adhering to some problem dependent rules that avoid generating invalid programs. In GP, additional parameters are supplied to have a control on the size of trees other than the standard parameters as found in GA.

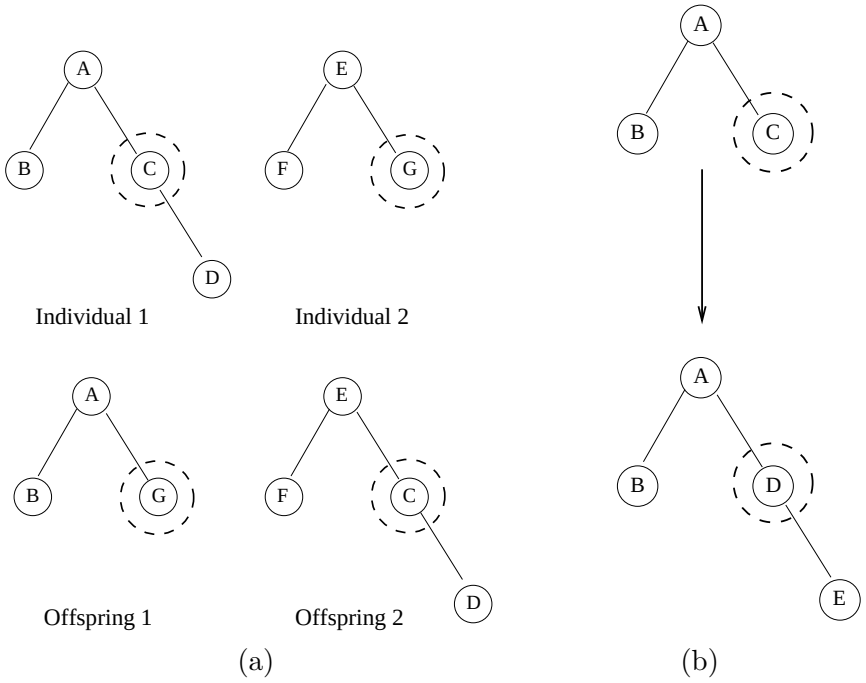


Figure 2.3: Simple crossover and mutation mechanisms of a GP: (a) Single point crossover; (b) Single point mutation.

Particle Swarm Optimisation

In Particle Swarm Optimisation (PSO), each individual in a swarm is referred to as a *particle*. A particle represents a vector in the search-space as in ES. PSOs mimic the behaviour of swarming bees, flocking birds or schooling fish [Kennedy and Eberhart, 2001]. These behaviours are modelled as *update rules* governing the movement of particles in the search-space. Each particle *moves* in the search-space by adjusting its velocity and position, which are influenced by its interaction with other particles in its neighbourhood. In a PSO, particles can be connected with each other in various topologies, which define different neighbourhoods. The default is considered to be the Global Topology, where particles can *interact* with any other particle in the population. In other words, this shows that each particle has a global *view* of the entire population.

In PSOs, the update rules modify the i^{th} particle's velocity and position at time t to time $t+1$ according to two equations respectively. The velocity update rule from the *Constriction Type* $1''$ PSO version [Clerc and Kennedy, 2002] is given by:

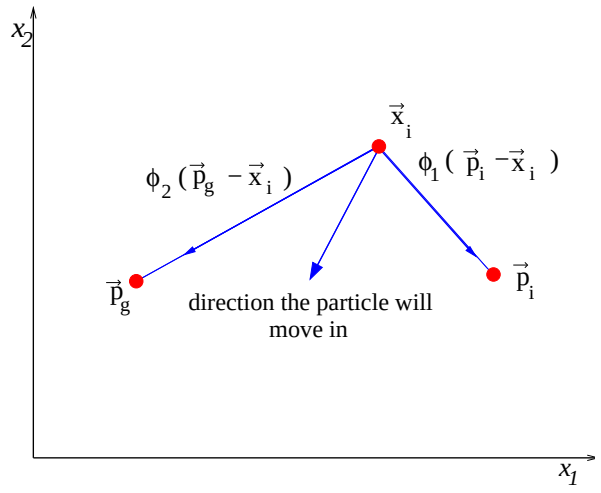


Figure 2.4: Updating a particle's position using the velocity and position update rules.

$$\vec{v}_i(t+1) = \chi\left(\vec{v}_i(t) + \phi_1(\vec{p}_i - \vec{x}_i(t)) + \phi_2(\vec{p}_g - \vec{x}_i(t))\right) \tag{2.7}$$

Here, $\vec{v}_i$ is the velocity of the i^{th} particle and $\vec{x}_i$ is its position. $\vec{p}_i$ is the best position found by the particle (also known as *personal best*); $\vec{p}_g$ is the best position so far (also known as *global best*) found in the particle's neighbourhood. The variables ϕ_1 and ϕ_2 are random numbers generated uniformly between $[0, \frac{\varphi}{2}]$. Here, φ is a constant equal to 4.1. ϕ_1 and ϕ_2 are used to perform point-wise vector multiplication. χ is called the constriction factor, and is used to prevent a particle from exploring too far in the search-space. Here, $\chi = 0.7298$, which is calculated from $\frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$ [Clerc and Kennedy, 2002]. We use this *Constriction Type* $1''$ PSO version later in Chapter 5 when developing our user-preference MOPSO algorithm.

Once the velocity is updated, the position of the particle is updated using a common equation for all PSO algorithms, which is given by:

$$\vec{x_i}(t+1) = \vec{x_i}(t) + \vec{v_i}(t) \tag{2.8}$$

Essentially each particle moves towards somewhere between its personal best and the global best as illustrated in Figure 2.4.

In PSO, the notion of crossover and mutation is implicit in the update rules. However, there are versions of PSO which incorporate explicit mutation schemes. In PSO, particles move across the search-space between time-steps, therefore, the term generation is replaced by *iteration* in the EA literature. PSOs are generally considered simpler than GA, GP or ES

because of the lack of extra parameters. There is no notion of parents or offspring in PSO. However, the position of a particle before movement at any iteration maybe considered as the parent and the updated position as the offspring.

In PSO, the calculation process of the velocities can produce particles having position values which are outside the variable bounds. In such situations, the velocities are adjusted (or *clamped*) so that the positions are within the boundary limits — $x_{j,i} \in \left[x_j^{(LB)}, x_j^{(UB)}\right]$ for every position index j in particle $\vec{x}_i$ — or close to the boundary limits. In the literature, the velocity is clamped to a maximum given by $v_j^{max} = \left|x_j^{(UB)} - x_j^{(LB)}\right|$, for every velocity index j in $\vec{v}_i$. After the velocities are calculated using Equation 2.7, each particle will clamp the velocity indexes according to Algorithm 2 [Kennedy and Eberhart, 2001].

Algorithm 2 Velocity clamping

```
1: for each particle x_i do
2:   for each position j in v_i do
3:     if v_{j,i} < -v_j^max then
4:       v_{j,i} = -v_j^max
5:     end if
6:     if v_{j,i} > v_j^max then
7:       v_{j,i} = v_j^max
8:     end if
9:   end for
10: end for
```

PSOs are very popular in the EA literature because the velocity update rule can be adjusted so that particles will explore the search-space widely or narrowly. Particles *flying* across huge distance is usually known as *exploration*, while narrowing the search is called *exploitation*. In PSO, particles update their velocities and positions depending on the global best particle in their neighbourhood. This best particle is also known as a *leader*. This behaviour of updating the velocities and positions to get close to the leader — guided by the leader — is referred as *following the leader*. This unique feature differs PSOs from the other EAs.

Differential Evolution

Differential Evolution (DE) differs from other EAs in its mechanism of *generating* offspring. In EAs, an individual plays the role of a parent to generate an offspring. In DE, an offspring

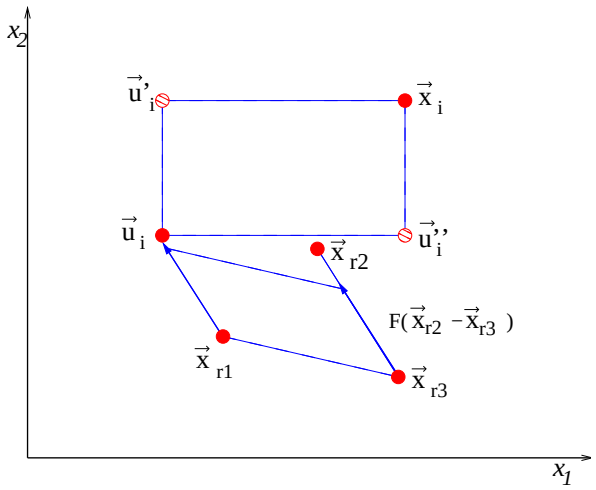


Figure 2.5: Generating an offspring from the DE/rand/1/bin scheme.

is generated by using vector differences among individuals in the population [Price et al., 2005]. The simplest and most popular method to generate offspring is the DE/rand/1/bin scheme. For an individual vector (parent) $\vec{x}_i$, a trial vector (offspring) $\vec{u}_i$ is generated using three other individuals $\vec{x}_{r1}, \vec{x}_{r2}, \vec{x}_{r3}$ from the population such that $i \neq r1 \neq r2 \neq r3$. The variable j of an individual is generated using the following equation:

$$u_{j,i} = \begin{cases} x_{j,r1} + F(x_{j,r2} - x_{j,r3}) \text{ if } (rand_j < CR \text{ or } j = j_{rand}) \\ x_{j,i} \text{ otherwise} \end{cases} \tag{2.9}$$

Here, $j_{rand} \in [1, N]$ where N is the number of dimensions in the search-space. $F \in [0, 1]$ is called the *scaling factor*. CR is the *crossover ratio* and $rand_j$ is randomly generated uniformly between $[0, 1]$. A potential offspring for an individual $\vec{x}_i$ will be located in the area bounded by the vectors $\vec{x}_i$, $\vec{u}_i$, $\vec{u}'_i$ and $\vec{u}''_i$ depending on CR and F (Figure 2.5).

In DE, the mutation is represented in the process of generating a vector using $\vec{x}_{r1} + F(\vec{x}_{r2} - \vec{x}_{r3})$. This process is known as the *differential mutation*. The crossover is the recombination of the parent ($\vec{x}_i$) and the vector generated by the differential mutation operation as described in Equation 2.9. This process is also known as *discrete recombination*. These generated offspring are then evaluated to see if they possess better fitness than the parent. If so, the parent is replaced by the offspring. As in PSO, this updating of an individual's position in the search-space is considered to occur within an iteration. Various versions of DE exists which either immediately update the parents as soon as the offspring are created or is done as a batch process at the end of an iteration.

As in PSO, DE vectors have to be adjusted if the generated offspring violate any variable bounds. Several boundary handling methods can be used as follows:

- Brick wall penalty: If a position value is outside the bounds then the position values are placed on the bounds (Algorithm 3) [Price et al., 2005].

Algorithm 3 Brick wall penalty

for each individual $\vec{x}_i$ **do**
  for each position j in $\vec{x}_i$ **do**
    if $x_{j,i} < x_j^{(LB)}$ **then**
      $x_{j,i} = x_j^{(LB)}$
    end if
    if $x_{j,i} > x_j^{(UB)}$ **then**
      $x_{j,i} = x_j^{(UB)}$
    end if
  end for
end for

- Random re-initialisation: If a position value is outside the bounds then it is initialised again with some randomness. One such approach is given in Algorithm 4 [Price et al., 2005]. Here, $rand(0, 1)$ generates a random number uniformly in the interval $[0, 1]$.

Algorithm 4 Random re-initialisation

for each individual $\vec{x}_i$ **do**
  for each position j in $\vec{x}_i$ **do**
    if $x_{j,i} < x_j^{(LB)}$ **or** $x_{j,i} > x_j^{(UB)}$ **then**
      $x_{j,i} = rand(0, 1) \times \left(x_j^{(UB)} - x_j^{(LB)}\right)$
    end if
  end for
end for

- Adaptive penalty: If a position value is outside the bounds then the fitness is reduced, depending on the number of position violations.
- Bounce-Back: This method takes into consideration the current progress made and adjust the indexes that violate the boundaries. Algorithm 5 [Price et al., 2005] describes this process.

Algorithm 5 Bounce-Back

1: **for** each parent $\vec{x}_i$ and corresponding offspring $\vec{u}_i$ **do**
2: **for** each position j in $\vec{u}_i$ **do**
3: **if** $u_{j,i} < x_j^{(LB)}$ **then**
4: $u_{j,i} = x_{j,i} + rand(0,1) \times \left(x_j^{(LB)} - x_{j,i}\right)$
5: **end if**
6: **if** $u_{j,i} > x_j^{(UB)}$ **then**
7: $u_{j,i} = x_{j,i} + rand(0,1) \times \left(x_j^{(UB)} - x_{j,i}\right)$
8: **end if**
9: **end for**
10: **end for**

The Bounce-Back mechanism given in Algorithm 5, to adjust the boundaries is more suitable for DE algorithms because it preserves the progress of optimisation (to some extent) of an individual rather than abruptly changing the exploration path as in Algorithms 3 and 4.

2.5 Evolutionary multi-objective optimisation algorithms

Among the many EAs available, we describe a more sophisticated hybrid DE and PSO algorithm later in Chapter 5, which is used to obtain solutions for many-objective problem instances. First, we describe the general setup of an EA used in multi-objective optimisation.

In Evolutionary Multi-objective Optimisation (EMO) algorithms, each individual's genome represents the decision variables for the problem instance. Therefore an individual is represented by a vector. The crossover operation between two individuals is done by performing point-wise vector operators. A P_{xor} is used to decide if two parents produce two new offspring by crossover or just copy themselves as offspring. An individual is chosen for mutation, depending on P_{mut}. Then the mutation is performed on one or more decision variables, depending on another P_{mut} for each decision variable. The fitness of an individual is represented as a vector of values, each position indicating the value for one objective function. The basic functionality of an EMO algorithm is shown in pseudocode form in Algorithm 6 [Deb, 2001].

EAs are well suited to find solutions in multi-objective problems, because of the use of a population of candidate solutions. This population can be used to get a good approximation of the Pareto optimal set within one execution. The first EA used for multi-objective optimisation was the Vector Evaluated Genetic Algorithm (VEGA) [Schaffer, 1985]. In VEGA, the

Algorithm 6 Pseudocode of an EMO algorithm

```
1: INITIALISE a population with individuals having random values for decision variables
2: EVALUATE each individual and assign objective values
3: repeat
4:    SELECT parents using a suitable criterion
5:    RECOMBINE parents to create offspring
6:    MUTATE the resulting offspring
7:    EVALUATE each offspring and assign objective values
8:    COMBINE the parent and offspring populations
9:    SELECT individuals from the combined population for the next generation
10: until TERMINATION is satisfied
```

population is divided into several sub-populations such that each one will optimise one objective. Individuals are selected from each sub-population in a proportional selection scheme to obtain the next generation. This same scheme is used to select the parents to generate the offspring. The main disadvantage of VEGA is that the diversity of the population was not sufficient. The population would converge to sub-populations (also known as *speciation*) especially on concave Pareto optimal surfaces. One solution to this issue as proposed by Kursawe [1991] was to maintain diversity by niching. The niching mechanisms maintained diversity by eliminating solutions which are too close to each other. This is also known as the elimination of *crowding solutions*. Maintaining mechanisms to remove solutions that are too close (or *crowded*) to each other gives more diversity to the population of the EMO algorithm.

In Multi-Objective Genetic Algorithm (MOGA), presented by Fonseca and Fleming [1993] the DM's input is used as a consecutive evaluation step provided by a *utility function*. This approach was able to extract non-dominated solutions over the dominated solutions providing an avenue to give equal emphasis on the non-dominated solutions. Thus giving the EA an ability to maintain a population that can converge to the Pareto optimal front.

One of the most popular mechanisms used to preserve diversity and emphasise on dominant solutions is *non-dominated sorting*. This idea was first presented by Srinivas and Deb [1994]. Non-dominated sorting is based on the idea of a ranking selection method to emphasise good solutions and to use a niching mechanism to obtain stable sub-populations. Non-dominated sorting has been refined to improve on the efficiency and computational complexity, which is known now as fast non-dominated sorting. This is the widely used non-dominated sorting procedure as seen in the EMO literature.

2.5.1 Non-dominated sorting

The process of a fast non-dominated sorting procedure as proposed by Deb et al. [2002] on a population P is given in Algorithm 7 [Deb et al., 2002].

Algorithm 7 Fast non-dominated sorting of a population P

```
for each p ∈ P do
   S_p = {}  // set of individuals p dominate
   n_p = 0  // counter for number of individuals that dominate p
   for each q ∈ P do
      if p ≺ q then
         S_p = S_p ∪ {q}
      else if q ≺ p then
         n_p = n_p + 1
      end if
   end for
   if n_p == 0 then  // p is in the first front
      p_rank = 1
      F_1 = F_1 ∪ {p}
   end if
end for
i = 1  // initialise the front counter
while F_i ≠ {} do
   Q = {}  // set of individuals of the next front
   for each p ∈ F_i do
      for each q ∈ S_p do
         n_q = n_q − 1
         if n_q == 0 then  // q is in the next front
            q_rank = i + 1
            Q = Q ∪ {q}
         end if
      end for
   end for
   i = i + 1
   F_i = Q
end while
return F_i
```

In the process of extracting the non-dominated fronts — we use the term *fronts* rather than sets here onwards, because in the EMO literature the objective-space is of more interest — each individual (p) has two entries. They are n_p, the number of individuals that dominate p and S_p, the set of solutions p dominates. p_{rank} indicates the non-dominated front to which p belongs to. In Algorithm 7, Step 11 shows that p belongs to the first front, while Step 22 shows that q belongs to the next front. This sorting process has a computational complexity of $O(Mn^2)$, where M is the number of objectives and n is the size of the population. In the EMO literature there are many EAs which incorporate the process of non-dominated sorting.

Non-dominated sorting is an example of a Pareto dominance ranking method for EMO algorithms. There exist several EMO algorithms in the literature that are based on Pareto dominance methods [Jensen, 2003]. Some examples of these are SPEA2 [Zitzler et al., 2002], PESA-II [Corne et al., 2001], PAES [Knowles and Corne, 2000] and NSGA-II [Deb et al., 2002]. We next describe NSGA-II that is based on non-dominated sorting. NSGA-II is an important algorithm for the research in this thesis because we considered certain features of NSGA-II in developing our user-preference EMO algorithms later in Chapters 3 and 4.

Non-dominated sorting GA (NSGA-II)

In NSGA-II, to maintain diversity of the individuals a *crowding distance* measurement is used along with the non-dominated sorting procedure. The crowding distance measurement gives an indication of how far an individual is from the others. This measurement provides a mechanism to give priority towards individuals that are in regions of the objective-space, which is *less crowded*. An outline of NSGA-II is given in Algorithm 8 [Deb et al., 2002]. Here, t denotes a generation.

NSGA-II is usually executed until a maximum number of generations (t_{MAX}) is reached. The diversity preservation mechanism is seen in steps 13 and 17 of Algorithm 8, where, if the size of next generation (P_{t+1}) is less than n the least crowded individuals — having the highest crowding distance values — of the next non-dominated front is included until the desired population size (n) is reached. Parent selection is done by a *tournament selection of size two*. Here, two individuals are selected at random, and the individual which has the better fitness is selected to be one parent. In NSGA-II, the dominant individual is considered to be the fitter one. If both individuals are non-dominated then the one with the highest crowding distance is selected. The two individuals selected in this process will perform a Simulated Binary Crossover (SBX) [Deb and Agrawal, 1995], to produce two offspring.

Algorithm 8 The NSGA-II algorithm

1: INITIALISE a population of size n, $t = 0$
2: EVALUATE each individual and assign fitness
3: **repeat**
4: SELECT parents (P_t) using tournament selections
5: RECOMBINE parents to create offspring (Q_t)
6: MUTATE the resulting offspring
7: EVALUATE each offspring and assign fitness
8: COMBINE the parent and offspring populations ($R_t = P_t \cup Q_t$)
9: NON-DOMINATED SORT R_t using Algorithm 7 to obtain the non-dominated fronts F_i
10: $P_{t+1} = \{\}$
11: $i = 1$
12: **while** $|P_{t+1}| + |F_i| \leq n$ **do**
13: CALCULATE crowding distance values on F_i
14: $P_{t+1} = P_{t+1} \cup F_i$
15: $i = i + 1$
16: **end while**
17: SORT F_i in descending order of crowing distance
18: ADD the first $(n - |P_{t+1}|)$ individuals from F_i to P_{t+1}
19: $t = t + 1$
20: **until** $t = t_{MAX}$

Algorithm 9 [Deb and Agrawal, 1995] describes the creation of two offspring $\vec{q}_1$ and $\vec{q}_2$ from two parents $\vec{p}_1$ and $\vec{p}_2$ using the SBX crossover mechanism.

Algorithm 9 The SBX operator for decision variables

1: **for** $i = 0$ **to** $N - 1$ **do** // for each corresponding decision variable
2: $q_{1,i} = 0.5(p_{1,i}(1 + \beta) + p_{2,i}(1 - \beta))$
3: $q_{2,i} = 0.5(p_{1,i}(1 - \beta) + p_{2,i}(1 + \beta))$
4: **end for**

In SBX, each real-valued decision variable is used to *simulate* a single point binary crossover (see Section 2.4.2 – page 21). The value β indicates the ratio of the spread of the average values of the decision variables of the offspring with relation to the parents.

- if $\beta < 1$, the average values of the offspring's decision variables will be between the average values of the parents' decision variable.
- if $\beta > 1$, the average values of the parents' decision variables will be between the average values of the offspring's decision variable.
- if $\beta = 1$, the average values will be equal for both parents and offspring.

In NSGA-II, $P_{xor} = 0.9$ and $\beta = 20.0$. The offspring are mutated using a *polynomial mutation* scheme [Deb and Goyal, 1996]. Algorithm 10 [Deb and Goyal, 1996] illustrates the process to mutate individual $\vec{p}$ to individual $\vec{q}$. Here $P_{mut} = \frac{1}{N}$ and $\beta = 20.0$.

Algorithm 10 The Polynomial mutation for decision variables

1: **for** $i = 0$ **to** $N - 1$ **do** // for each corresponding decision variable
2: $\Delta_{MAX} = p_i^{(UB)} - p_i^{(LB)}$
3: $u = rand(0, 1)$
4: **if** $u < 0.5$ **then**
5: $\overline{\delta} = (2u)^{\frac{1}{(\beta+1)}} - 1$
6: **else**
7: $\overline{\delta} = 1 - [2(1 - u)]^{\frac{1}{(\beta+1)}}$
8: **end if**
9: $q_i = p_i + \overline{\delta}\Delta_{MAX}$
10: **end for**

The process of crossover and mutation is done until n number of parents generate n number of offspring. Therefore $|R_t| = 2n$. NSGA-II has been very effective in multi-objective problems having two or three-objectives. The success of NSGA-II was a motivation for EMO researchers to introduce other algorithms based on the concept of non-dominated sorting.

Non-dominated sorting PSO (NSPSO)

NSPSO [Li, 2003] employs the same non-dominated sorting procedure used in NSGA-II. In NSPSO, all particles in the first non-dominated front are sorted according to the crowding distance values. Each particle chooses its leader (global best) from a set of least-crowded particles. This could be the top 10% of the particles in the first non-dominated front sorted in decreasing order of crowding distance values. Consequently, particles of the future iterations are well distributed along the Pareto front. The particles update their velocities using a inertia weight PSO model [Shi and Eberhart, 1998]. The inertia weight w, is used as:

$$\vec{v_i}(t+1) = w\vec{v_i}(t) + \phi_1(\vec{p_i} - \vec{x_i}(t)) + \phi_2(\vec{p_g} - \vec{x_i}(t)) \tag{2.10}$$

Here, w is varied so that in the beginning the PSO will explore a wider area by increasing the momentum (large w values), then towards the end of the total number of iterations to narrow the search (small w values). The weight w controls this exploration and exploitation process. In NSPSO, a population of n parents generate n offspring (the movement of particles in an iteration) to create a population of size $2n$. From this population of size $2n$, n number of particles are selected using the non-dominated sorting procedure. These n number of particles will be carried over to the next iteration. The main advantage of NSPSO over NSGA-II is its fast convergence, which is an inherent property of PSO algorithms.

Non-dominated sorting DE (NSDE)

In NSDE, the crossover operator found in NSGA-II is replaced by the DE rule **DE/current-to-rand/1** [Iorio and Li, 2004]. In this DE scheme, the offspring $\vec{u}_i$ for the parent $\vec{x}_i$ is generated by:

$$\vec{u}_i = \vec{x}_i + K(\vec{x}_{r3} - \vec{x}_i) + F(\vec{x}_{r1} - \vec{x}_{r2}) \tag{2.11}$$

Here, K is the crossover ratio and F is the scaling factor. This modification was provided to give the NSDE algorithm the ability to handle rotated multi-objective problem instances.

In rotated problems, the decision variables have interactions between them. NSGA-II was not successful in handling such problems because the crossover operators were not able to produce offspring that were *rotationally invariant*. The crossover operators produced offspring that were biased in exploring the search-space along the axes of the variables, thus limiting the ability of exploring the entire search-space. In a rotated problem, the Pareto optimal solutions could reside in regions between the axes, therefore better crossover operators are needed to produce individuals that can explore these regions. The DE scheme proposed in NSDE produces individuals which are not biased in exploring the search-space along the coordinate axes. This made the individuals in NSDE rotationally invariant and thus able to locate the Pareto optimal solutions of rotated multi-objective problems.

2.5.2 Non-dominated sorting in many-objective problems

In many-objective optimisation problems — that have more than three objectives — comparing individuals using dominance becomes less effective [Hughes, 2008; Köppen and Vicente-Garcia, 2004; Köppen et al., 2005]. In EMO literature the decrease of dominating individuals as the number of objectives increases is known as the *curse of dimensionality* [Köppen and Yoshida, 2007; Kukkonen and Lampinen, 2007].

We observed that the average percentage of non-dominated solutions in a randomly initialised population of 100 individuals for the DTLZ test problems increased exponentially as the number of objectives increased. This is seen in Figure 2.6 for four DTLZ problem instances. In the two-objective instance, the percentage of non-dominated solutions were on average about 10%, from 50 independent runs. In three and four-objectives instances the percentage of non-dominated individuals were on average 50% and 80% respectively. When the number of objectives were increased to five and above, the percentage of non-dominated individuals were almost 100%. This increase of non-dominated individuals has also been seen in many studies in the EMO literature [Köppen and Yoshida, 2007; Kukkonen and Lampinen, 2007]. From these results we concluded that in many-objective search-spaces major part of the population — usually about 90% — becomes non-dominated to each other. This behaviour severely limits a dominance-ranking based EMO algorithm's ability to compare and search for solutions in many-objective problems. To solve this issue, one of the main ideas that is found in the EMO literature is to modify the dominance relation to better suit many-objective problems.

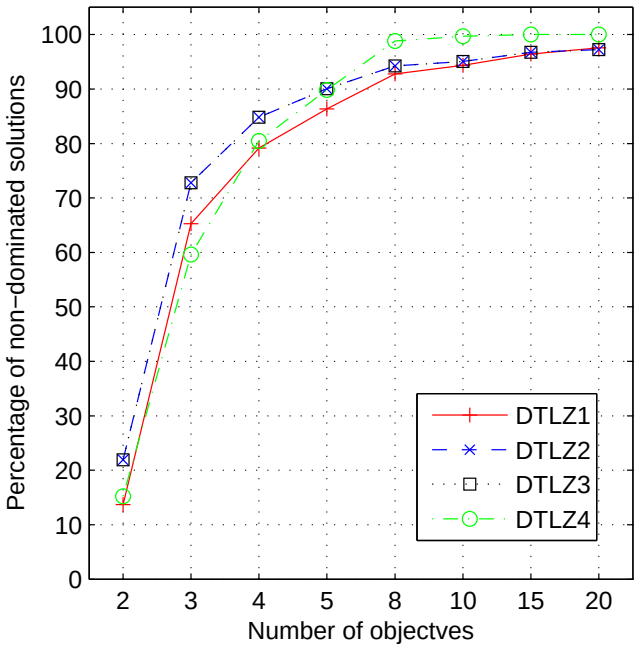


Figure 2.6: The average percentage of non-dominant solutions in the initialisation steps of some many-objective DTLZ test problems.

2.5.3 Modifying the dominance relation

Modifying the dominance relation by making it less strict or introducing new features make it easier for the EMO algorithms to find solutions in many-objective problems. Next, we describe some of these methods.

Ranking dominance

A scheme named *ranking dominance* was introduced by Kukkonen and Lampinen [2007], where solution points are ranked according to each objective. Then an aggregation function is used to obtain a fitness value from all the rank values. For a minimisation problem, if the fitness obtained from the aggregate function for an individual $\vec{x}$ is less than another individual $\vec{y}$, then $\vec{x}$ is said to be *better* or dominant than $\vec{y}$. In this study the ranking dominance concept provided the EMO algorithm the ability to converge to the Pareto front in many-objective problems. However, this method was computationally expensive and as the authors have observed diversity of the population was hard to maintain. Although the EMO algorithm was able to converge, a good spread of solutions was not found.

The ϵ-Preferred relation

Sülflow et al. [2007] introduced a relation named ϵ-Preferred. This was an extension of the original *favour* relation introduced by Drechsler et al. [2001]. The favour relation is a relaxed version of dominance which is given as:

$$\vec{a} \prec_{preferred} \vec{b} \Leftrightarrow \left|\{i : f_i(\vec{a}) < f_i(\vec{b})\}\right| > \left|\{j : f_j(\vec{a}) > f_j(\vec{b})\}\right| \tag{2.12}$$

Here, $\vec{a}$ is said to be *preferred* (or *favoured*) to $\vec{b}$ if and only if $\vec{a}$ is better than $\vec{b}$ in a larger number of objectives. The ϵ-Preferred relation is given as:

$$\vec{a} \prec_{\epsilon-preferred} \vec{b} \Leftrightarrow \vec{a} \prec_{\epsilon-exceed} \vec{b} \vee \left(\vec{a} \not\prec_{\epsilon-exceed} \vec{b} \wedge \vec{a} \prec_{preferred} \vec{b}\right) \tag{2.13}$$

$$\vec{a} \prec_{\epsilon-exceed} \vec{b} \Leftrightarrow \left|\{i : f_i(\vec{a}) < f_i(\vec{b}) \wedge \left|f_i(\vec{a}) - f_i(\vec{b})\right| > \epsilon_i\}\right| > \left|\{j : f_j(\vec{b}) < f_j(\vec{a}) \wedge \left|f_j(\vec{b}) - f_j(\vec{a})\right| > \epsilon_j\}\right|$$

Here, ϵ-exceed counts how many times the solutions exceed the given $\epsilon_i > 0$ limits. It is also interesting to note that this relation is not transitive like dominance. The computational complexity of ϵ-Preferred is equal to the standard dominance relation. The study by Sülflow et al. [2007] showed an example of a many-objective scheduling problem, which was able to find solutions better with ϵ-Preferred relation than with the standard dominance relation.

Gradual dominance

A method to avoid using the standard dominance relation in the leader selection stage of a MOPSO algorithm was introduced by Köppen and Yoshida [2007]. Here, leaders are obtained from the population using a *gradual dominance* relation. The particles are assigned with a *ranking value*, which is the maximum of the *degree of being dominated*. This value is calculated using a *fuzzy scheme* described extensively by Köppen and Vicente-Garcia [2004]. Leaders are chosen from a set of particles with the lowest ranking values. These lowest ranked particles will also be the least crowded. In this study the authors have observed that the PSO algorithms can get stuck in local optima even with gradual dominance.

Using a Maximin approach

maximinPSO [Li, 2004] adopts the maximin strategy [Balling, 2003; Osborne and Rubinstein, 1994] found in game theory. This strategy is incorporated into EMO as; the minimum of the

differences of each objective i between any two decision vectors $\vec{a}$ and $\vec{b}$ in the population. This is given by $\mathbf{min}_{i=1...M}\{f_i(\vec{a}) - f_i(\vec{b})\}$. The fitness of any $\vec{a}$ is given by the maximum of the minimum values between $\vec{a}$ and all other vectors in the population (other than $\vec{a}$):

$$fitness(\vec{a}) = \max_{j=1...n:a\neq b_j} \left\{ \min_{i=1...M} \left\{ f_i(\vec{a}) - f_i(\vec{b_j}) \right\} \right\} \tag{2.14}$$

The particles with fitness values less than 0.0 are considered as the non-dominated solutions. These non-dominated solutions are in the first non-dominated front. The total number of individuals that move onto the next generation will consist of all individuals in the first non-dominated set. If the size of the non-dominated set is less than desired, then dominated individuals are chosen at random to fill the vacant positions. A particle will choose a leader (global best), randomly from the top 10% of least crowded particles in the non-dominated front. The velocity update is done using the inertia weight PSO model. maximinPSO seems to be sensitive to population sizes and maximinPSO can only identify the first non-dominated front, which was a drawback. The upside is that maximin fitness provides information on both non-dominance and diversity.

We see that in the above studies the main aim was to modify the dominance comparisons to suit many-objective problems. Some approaches used a relaxed version of dominance, while others used another ranking scheme on top of the dominance comparisons. Corne and Knowles [2007] have shown that some ranking schemes perform better than modifying the dominance schemes. Unfortunately, the usage of these methods in difficult many-objective problems (multi-modal) is not present in the literature. A different approach to modifying the dominance concept that has been gathering momentum in the recent years is integrating preferences to EMO algorithms. These preference mechanisms were seen originally in the Multi-Criteria Decision Making (MCDM) [Ehrgott and Gandibleux, 2002] literature.

2.6 Integrating preferences into EMO algorithms

Preference methods as described in the MCDM and EMO literature come in three forms based on the stage of the optimisation process the DM is involved in [Branke and Deb, 2004; Coello Coello, 2000].

1. *A Priori* methods: A DM gives preferences first then the algorithm finds solutions considering those preferences.

2. *A Posteriori* methods: After an algorithm provides all possible solutions the DM selects the interesting ones.

3. *Interactive* methods: A DM will be involved within the progress of the algorithm in many stages providing guidance.

Actually the *a posteriori* methods are the most common approaches taken by the current EMO algorithms. Depending on the circumstances, a DM will select suitable solutions from the solution front that the algorithm has converged to [Miettinen, 2008]. In *a priori* methods, the DM is required to first indicate *preferred regions* of the objective-space for an algorithm to find solutions in [Miettinen, 2008]. The algorithm will only concentrate on those preferred regions to find solutions on the Pareto optimal front. In the interactive methods the DM is fully involved in the execution process of the algorithm [Miettinen et al., 2008]. The DM will observe the current solutions and provide guidance to the algorithm based on intermediate feedback. This involvement of the DM during optimisation will eventually guide the algorithm to find useful solutions.

2.6.1 Guidance for EMO algorithms using a priori methods

A survey by Miettinen [2001] shows many different *a priori* and *a posteriori* methods for nonlinear many-objective optimisation problems. However, most existing EMO algorithms are *a posteriori* methods, since they perform a search first before presenting a set of final solutions to the DM. Branke and Deb [2004], and Coello Coello [2000] have presented surveys of how various preference mechanisms can be handled in EMO algorithms. In this section we describe some of the EMO algorithms which incorporate *a priori* methods.

Goal programming

Deb [1999] proposed a *goal programming* approach with GAs as a mechanism to find solutions in multi-objective problems. A modified non-dominated sorting algorithm was used to find a set of points closest to the supplied goal, which is a set of aspiration values for each objective. This approach did not care if the solutions were not found on the Pareto front. If the goal was in the feasible region then the algorithm is directed to that solution point. If the goal was in the infeasible region then the solution point closest to the goal was obtained.

Guided Multi-Objective Evolutionary Algorithm (G-MOEA)

Branke et al. [2001] proposed G-MOEA, where the definition of dominance was modified using the preferences given by the DM. Here the DM can provide accepted trade-off values for each objective in a two-objective optimisation problem. The trade-off values for each objective give a notion of preference between the two objectives.

More specifically, two utility functions called w_{min} and w_{max} are defined on the two objective functions f_1 and f_2 such that they define the boundary of the values a DM is interested in. These maximum and minimum utility functions basically indicate how many units of an objective function the DM is willing to trade-off against the other objective function. For example if the DM decides that f_1 is worth at most a units of f_2, then $w_{max}(f_1, f_2) = af_1 + f_2$. Similarly, the minimum utility function w_{min} can also be derived. Based on these utility functions the dominance relation between two individuals $\vec{x}$ and $\vec{y}$ is modified as:

$$\begin{aligned} \vec{x} \prec \vec{y} \Leftrightarrow w_{max}(f_1(\vec{y}), f_2(\vec{y})) \leq w_{max}(f_1(\vec{y}), f_2(\vec{y})) \ \wedge \\ w_{min}(f_1(\vec{y}), f_2(\vec{y})) \leq w_{min}(f_1(\vec{y}), f_2(\vec{y})) \end{aligned} \tag{2.15}$$

This modified dominance relation provides guidance to the EMO algorithm, which is based on the DM's preferences for the trade-offs between f_1 and f_2.

Interactive EMO algorithm

Thiele et al. [2009] proposed an interactive EMO algorithm based on preferences. First, a rough idea of the Pareto front is presented to the DM, who then specifies some points in the objective-space where the algorithm should concentrate on. This method is a two-step approach since the algorithm must first find an approximate Pareto front and then incorporate the reference points. This process can also be executed in many iterative steps.

Recently, some of the classical MCDM methods have been incorporated into EMO algorithms. These methods are usually known as *user-preferences*. We first provide the classical definitions of the reference point and the light beam search methods that are used in this thesis, followed by some examples found in the EMO literature.

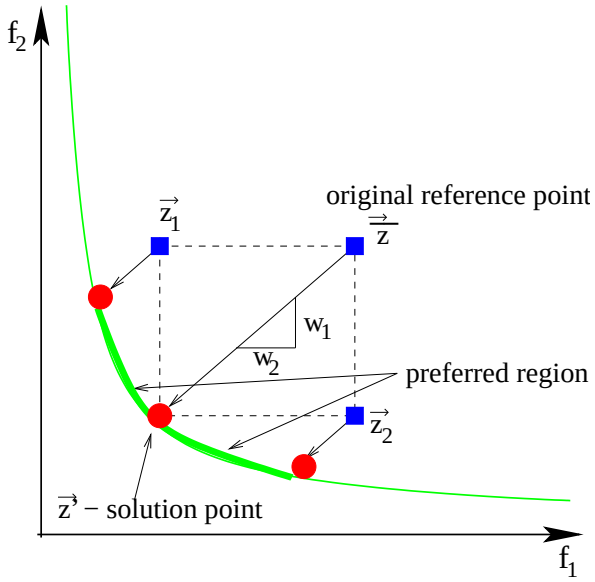


Figure 2.7: The classical definition of the reference point method.

2.6.2 Reference point method

The classical reference point method was first described by Wierzbicki [1980]. A reference point $\vec{\overline{z}}$ for a many-objective problem consists of *aspiration values* for each objective. In the classical MCDM literature this reference point is used to construct a single objective function (given by Equation 2.16), which is to be minimised over the entire search-space.

If $\vec{x}$ is a solution in the search-space,

$$\text{minimise} \max_{i=1,\ldots,M} \{w_i(f_i(\vec{x}) - \overline{z}_i)\} \tag{2.16}$$

Here, $\vec{\overline{z}} = [\overline{z}_1, \ldots, \overline{z}_M]$ is the reference point and $\vec{w} = [w_1, \ldots, w_M]$ is a set of weights, such that $\sum_{i=1}^{M} w_i = 1$ and $\forall i, w_i > 0$. The DM can assign values for weights, which represent any bias toward an objective.

Figure 2.7 illustrates the classical reference point method in a two-objective space. The MCDM literature [Deb et al., 2006; Miettinen, 2001] shows that several other reference points ($\vec{z_1}$ and $\vec{z_2}$) can be derived using the original reference point ($\vec{\overline{z}}$) and a solution point ($\vec{z'}$) by:

$$\vec{z_i} = \vec{\overline{z}} + (\vec{z'} - \vec{\overline{z}}) \cdot e_i \tag{2.17}$$

Here, e_i is the i^{th} coordinate directional vector and $\cdot$ is the dot-product between e_i and

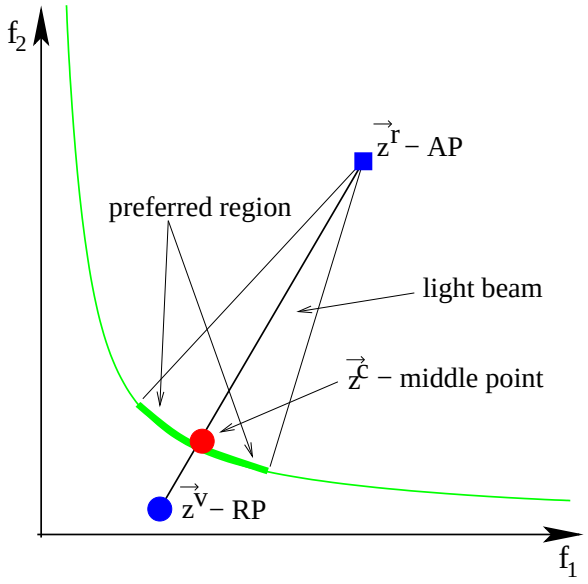


Figure 2.8: The classical definition of the light beam search method.

the vectors. In a recursive manner these new $\vec{z}_i$ reference points can be used to derive more solution points. This traditional approach can be used to define a spread of solutions in the objective-space. The regions where solutions are found, surrounding the reference points, are known as *preferred regions*.

2.6.3 Light beam search method

The light beam search was first introduced by Jaszkiewicz and Słowiński [1999]. The DM first needs to indicate two points in the objective-space, the Aspiration Point (AP), denoted by $\vec{z^r}$ and the Reservation Point (RP), denoted by $\vec{z^v}$. In situations where the AP and RP are not given, some other points like the *nadir point* and *ideal point* can be used instead. The search direction is given from AP to RP. Metaphorically, this illustrates a light beam originating from AP in the direction of RP. Figure 2.8 illustrates the classical light beam search setup in a two-objective space. In the classical MCDM literature the light beam search method uses an *achievement scalarising function* (given by Equation 2.18), which is to be minimised.

If $\vec{x}$ is a solution in the search-space,

$$\text{minimise} \max_{i=1,\ldots,M} \{\lambda_i \left(f_i(\vec{x}) - z_i^r\right)\} + \rho \sum_{i=1}^{M} \left(f_i(\vec{x}) - z_i^r\right) \tag{2.18}$$

Here, $\vec{z^r} = [z_1^r, \ldots, z_M^r]$, $\vec{z^v} = [z_1^v, \ldots, z_M^v]$ and ρ is a sufficiently small positive number

called the *augmentation coefficient* that is usually set to 10^{-6}. $\vec{\lambda} = [\lambda_1, \ldots, \lambda_M]$, where $\lambda_i > 0$ is a weighted vector. This weighted vector is derived from Equation 2.19.

$$\lambda_i = \frac{1}{|z_i^r - z_i^v|} \tag{2.19}$$

The projection of the AP in the direction of the RP will result in a middle point on the non-dominated solution front. In the usual notation, a middle point is given by $\vec{z^c} = [z_1^c, \ldots, z_M^c]$. The DM can then decide on a region surrounding this middle point, which gives the preferred region. This region is obtained by the notion of outranking (S) [Jaszkiewicz and Słowiński, 1999]. $\vec{a}$ outranks $\vec{b}$ (denoted by $\vec{a}S\vec{b}$) if $\vec{a}$ is considered to be at least as good as $\vec{b}$ within some *threshold* value.

In the MCDM literature, outranking is defined by either one of three possible threshold values. They are the *indifference threshold* (m_q), *preference threshold* (m_p) or *veto threshold* (m_v). For example, if the veto threshold values are $\vec{v} = [v_1, \ldots, v_M]$ then $\vec{x}S\vec{z^c}$ if and only if $m_v(\vec{z^c}, \vec{x}) = 0$ where,

$$m_v(\vec{z^c}, \vec{x}) = card\{i : f_i(\vec{x}) - z_i^c \geq v_i, i = 1, \ldots, M\} \tag{2.20}$$

The requirement in Equation 2.20 suggests that if the differences of every corresponding objective i between $\vec{x}$ and $\vec{z^c}$ are within a given threshold value, then $\vec{x}S\vec{z^c}$. All solutions which outrank the middle point indicates the preferred region *illuminated* by the light beam.

2.6.4 Using MCDM user-preferences in EMO algorithms

The reference point and light beam search methods have been integrated into some existing EMO algorithms. These studies have shown that it is possible to integrate user-preferences in EMO algorithms which aids in finding solutions in many-objective problems.

Reference point based NSGA-II

Deb et al. [2006] presented an EMO algorithm incorporating the reference points into NSGA-II. Here, the *crowding distance* was modified to incorporate the reference point information. A weighted normalised Euclidean distance measurement was used here to modify the crowding distance. This weighted normalised Euclidean distance from any vector $\vec{x}$ to the reference point $\vec{\bar{z}}$ is given as:

$$d = \sqrt{\sum_{i=1}^{M} w_i \left(\frac{f_i(\vec{x}) - \overline{z}_i}{f_i^{\max} - f_i^{\min}} \right)^2} \tag{2.21}$$

An individual having the least weighted normalised Euclidean distance values was considered better. The algorithm controlled the spread of solutions using ϵ-dominance [Deb et al., 2005a], while still having the non-dominated sorting as the crux of the algorithm. This approach included many components built around the non-dominated sorting procedure.

Reference direction and Light beam search based NSGA-II

Deb and Kumar [2007b] incorporated the reference direction method [Korhonen and Laakso, 1986] into NSGA-II. Here the crowding distance was also modified to incorporate the preference information and ϵ-dominance controlled the spread. Subsequently, the light beam search method was included into NSGA-II by Deb and Kumar [2007a]. Although, the crowding distance modification was used here to incorporate the preference information the spread of solutions was handled by the veto threshold values.

Using reference points in PSO

Allmendinger [2008] illustrated the use of a *steady-state* PSO algorithm which incorporated the reference point method. This algorithm did not use non-dominated sorting procedures, but compared particles based on dominance in a steady-state manner. However, this PSO algorithm was unable to find solutions in difficult many-objective problems instances.

Using reference points in group decisions

Pfeiffer et al. [2008] introduced a method of combining different reference points to obtain a notion of *global preference*, which can be then used in group decision support systems. Here, several DMs can collectively propose various reference points for a given problem. The EMO algorithm will obtain solutions in a region of the objective-space, which incorporates preferences by all the DMs.

Using reference points to modify dominance relations

Recently, Molina et al. [2009] introduced a new dominance concept named *g-Dominance*. Here, reference points were used to modify the standard dominance concept. The results

however did not indicate that the EMO algorithm based on this idea was able to handle difficult many-objective problem instances.

The studies where EMO algorithms incorporated the traditional MCDM preference methods have shown to be able to handle some simple many-objective problem instances. However, these EMO algorithms were complex in the integration of preferences and computationally expensive because of their dependence on the standard dominance comparisons. They also suffer from the same problem of not being able to distinguish solutions effectively for problems with a large number of objectives, in the non-dominated sorting process. Consequently, these algorithms have been less effective in search and are inclined to converge prematurely to local Pareto-fronts in difficult multi-modal many-objective problems.

2.7 Performance metrics for EMO algorithms

Metrics are required to assess the performance of EMO algorithms. In the EMO literature there exist several such metrics. It is important to note that all the metrics for EMO algorithms presented here represent measurements made in the objective-space. A survey by Okabe et al. [2003], discusses some of the merits and shortcomings of some of these performance metrics, by categorising their functionality in the multi-dimensional objective-space.

2.7.1 Metrics to measure the convergence and distribution of solutions

Knowles and Corne [2002] presents a survey of some popular metrics for EMO algorithms found in the literature. These metrics indicate the performance of an EMO algorithm in the areas of:

- Closeness of convergence to the Pareto front
- Extent of the spread of solutions
- Uniformity of the spread of solutions

Next, we describe in detail how these areas are assessed using several performance metrics. It is important to note that in the following metrics, the solutions found by an EMO algorithm are compared with solutions that exist on a known Pareto front. Although these metrics were often used to compare the final solution set of an EMO algorithm, they can also be used to indicate the intermediate performances of an EMO algorithm during a run.

Average distance to the Pareto front

The convergence of an EMO algorithm is indicated by measuring how far the solution front is from the Pareto front. The metric by Zitzler et al. [2000], given in Equation 2.22 is used to measure the convergence.

$$M_1^* \left(Y'\right) = \frac{1}{|Y'|} \sum_{\vec{p'} \in Y'} \left(\min_{\vec{q} \in \overline{Y}} \sqrt{\sum_{i=1}^{M} \left(f_i(\vec{p'}) - f_i(\vec{q}) \right)^2} \right) \tag{2.22}$$

Here, the solution front obtained from an EMO algorithm is given as Y', the known Pareto optimal front is given as $\overline{Y}$ and M is the number of objectives. The sampling points on $\overline{Y}$ are selected so that they are evenly distributed along the Pareto front. Then the Euclidean distance from each solution $\vec{p'} \in Y'$ to every solution $\vec{q} \in \overline{Y}$ is calculated using $\sqrt{\sum_{i=1}^{M} \left(f_i(\vec{p'}) - f_i(\vec{q})\right)^2}$. The minimums of these Euclidean distances are averaged from all the $\vec{p'} \in Y'$ solutions points to obtain a measurement on how close an EMO algorithm has converged onto the Pareto front. This metric was also named *generational distance* by Van Veldhuizen [1999].

Distribution of solutions

Another important metric for EMO algorithms is the spread of solutions on the solution front. Ideally an EMO algorithm should obtain solutions that are evenly spread across the solution front, rather than having isolated groups of solutions. Equation 2.23 [Zitzler et al., 2000] measures this aspect by grouping solution points.

$$M_2^* \left(Y'\right) = \frac{1}{|Y' - 1|} \sum_{\vec{p'} \in Y'} \left| \vec{q} \in Y' : \left(\sqrt{\sum_{i=1}^{M} \left(f_i(\vec{p'}) - f_i(\vec{q}) \right)^2} \right) > \sigma^* \right| \tag{2.23}$$

Here, $\sigma^* > 0$ is called a niche neighbourhood size. M_2^* will provide a value in the range of $[0, |Y'|]$. This metric measures how widely the solutions are spread across the front. More specifically, this metric measures the number of niches (groups) based on some σ^* value. A higher value for M_2^* indicates a better distribution of solutions.

Extent of the solution front

It is also important for an EMO algorithm to obtain solution points that are at the extremes of the Pareto front. The metric in Equation 2.24 [Zitzler et al., 2000] measures the maximum distance between the solution points to indicate the extent of the spread of solutions.

$$M_3^* \left(Y'\right) = \max_{\vec{p'},\vec{q'} \in Y', \vec{p'} \neq \vec{q'}} \left(\sqrt{\sum_{i=1}^{M} \left(f_i(\vec{p'}) - f_i(\vec{q'}) \right)^2} \right) \tag{2.24}$$

This metric will provide the maximum Euclidean distance between all the solution points on the solution front Y'. The metric is used to determine if the EMO algorithm was able to find the extreme solutions.

Error ratio

This metric basically measures how many vectors of the solution set Y' are in the Pareto optimal set $\overline{Y}$ [Van Veldhuizen, 1999]. For all $\vec{p'} \in Y'$:

$$ER\left(Y'\right) = \frac{1}{|Y'|} \sum_{\vec{p'} \in Y'} c_{\vec{p'}} \tag{2.25}$$

Here $c_{\vec{p'}} = 0$ if $\vec{p'} \in \overline{Y}$ else $c_{\vec{p'}} = 1$. The lower values represent *low error*, which indicates that more solution points are Pareto optimal.

Maximum Pareto front error

This metric measures the largest distance between any solution $\vec{p'} \in Y'$ and the corresponding closest solution $\vec{q} \in \overline{Y}$ [Van Veldhuizen, 1999]. This is calculated from:

$$MPFE\left(Y'\right) = \max_{\vec{q} \in \overline{Y}} \left\{ \min_{\vec{p'} \in Y'} \sqrt{\sum_{i=1}^{M} \left(f_i(\vec{p'}) - f_i(\vec{q}) \right)^2} \right\} \tag{2.26}$$

Here solutions closer to the Pareto front will give a *low maximum error* value. For example, this metric will give a higher value in a situation where nine of the ten solutions are on the Pareto front and one is very far from the Pareto front, compared to ten solutions very close to the Pareto front.

Reversed generational distance

A metric by Li et al. [2007] emphasised the diversity of a solution set Y, while maintaining the measurement for convergence. In this approach the reverse of Equation 2.22 was used to obtain a measurement for diversity, and is defined as:

$$rGD\left(Y'\right) = \frac{1}{\left|\overline{Y}\right|} \sum_{\vec{q} \in \overline{Y}} \left(\min_{\vec{p'} \in Y'} \sqrt{\sum_{i=1}^{M} \left(f_i(\vec{q}) - f_i(\vec{p'}) \right)^2} \right) \tag{2.27}$$

The best possible convergence of a solution front Y' compared to the Pareto front $\overline{Y}$ will yield in $rGD(Y') = 0$. This metric is useful because it can give useful information about both the convergence and spread of solutions in Y'.

Normalised Euclidean distance

Deb and Jain [2002] proposed a simple metric which shows the convergence of a solution front to a known Pareto front using a normalised Euclidean distance measure. For any solution $\vec{p'} \in Y'$ of any given generation, the minimum Euclidean distance between every solution $\vec{q} \in \overline{Y}$ is calculated as:

$$NEucD\left(\vec{p'}\right) = \min_{\vec{q} \in \overline{Y}} \sqrt{\sum_{i=1}^{M} \left(\frac{f_i(\vec{p'}) - f_i(\vec{q})}{f_i^{\max} - f_i^{\min}} \right)^2} \tag{2.28}$$

Here, $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum values respectively for each objective from the solution front in consideration.

Another measurement for diversity was proposed by Li et al. [2005] based on clustering individuals. Initially, each individual is considered a subgroup. The process then involves combining two subgroups so that the distance between their centroids is smaller than a given threshold value. This combination process ends when the maximum number of subgroups is found. The diversity is the maximum number of subgroups divided by the size of the solution front.

2.7.2 Metrics to measure dominance between solution fronts

Knowles [2002] presents the C metric, which gives a cardinal measure of the dominant vectors between two solution fronts. If Y_1 and Y_2 are two solution sets then $C(Y_1, Y_2)$ maps the ordered pair (Y_1, Y_2) to $[0, 1]$ as:

$$C(Y_1, Y_2) = \frac{|\{\forall \vec{q} \in Y_2 \quad \exists \vec{p} \in Y_1 : \vec{p} \preceq \vec{q}\}|}{|Y_2|} \tag{2.29}$$

$C(Y_1, Y_2) = 1$ gives that all solutions in Y_2 is weakly dominated (see Section 2.1.4 – page 15) by all solutions in Y_1. It is also interesting to note that $C(Y_1, Y_2)$ is not necessarily equal to $1 - C(Y_2, Y_1)$.

All these performance metrics that measure the relative distance towards a solution front — to emphasise on the convergence and diversity — can only be used to compare EMO algorithms on problem instances of known Pareto fronts like the ZDT, DTLZ and WFG test suites. This is a major shortcoming when EMO algorithms are compared on real-world problem instances, where usually the Pareto fronts are unknown. We address this issue by describing a better performance metric later in Chapter 4. This performance metric is then extended for comparing user-preference EMO algorithms.

2.8 Summary

In this chapter, we described multi-objective optimisation problems, the dominance relation and non-dominated solutions. In multi-objective problems, there are several conflicting objectives that have to be optimised simultaneously fulfilling some conditions. We presented some existing multi-objective test suites from the literature because they are used frequently by researchers in multi-objective optimisation to compare the performance of algorithms.

To better understand the EMO algorithms it is important to know the basic functionality of EAs. This chapter presented some of the existing EAs, and the process of using them in solving multi-objective problems. The main functionality of the current EMO algorithms is the non-dominating sorting procedure. We described the non-dominated sorting procedure and how it is used with the crowding distance operator in NSGA-II, which was a basis for many EMO algorithms. However, it was also found that the dominance relation became ineffective in many-objective problems, which are usually considered to have more than three conflicting objectives.

Some methods seen in the literature which aid EMO algorithms to function effectively in many-objective problem domains are in the form of modifying the dominance relation and using preferences. We have described in detail the advantages of using preferences over modifying the dominance relation. User-preferences in EMO algorithms are the basis of this thesis. One main motivation of this study is to use traditional user-preference mechanisms found in the MCDM literature to enhance EMO algorithms so that they are more efficient than the current state-of-the-art algorithms.

We also presented an overview of some popular EMO performance metrics in this chapter. These performance metrics have some features not suitable for user-preference EMO algorithm. We address these issues later in Chapter 4. In the next chapter, we introduce a *distance metric* utilising the user-preference information that is provided by the DM. This method removes the need to use dominance comparisons, and is computationally efficient and simpler than modifying the standard dominance concept.

Chapter 3

A framework for integrating user-preferences

In Chapter 2, we presented some existing user-preference EMO algorithms that were primarily based on the non-dominated sorting mechanism. Although these algorithms were successful in locating solutions on some many-objective problems, they were computationally expensive and did not function well on difficult problem instances. To address these issues we propose some optimisation techniques based purely on user-preferences.

In this chapter, we describe how a metric is derived from user-preferences that can be used to guide EMO algorithms to find solutions in many-objective problem instances. We denote this metric as the *distance metric*, which is derived from the reference point and the light beam search methods found in the MCDM literature. Section 3.1 introduces this distance metric. We then describe its functionality in EMO algorithms and how the search process is guided in the objective-space. This section also describes the process of making the best use of a population to derive a range of solutions on the Pareto front guided by the distance metric. We then introduce an EMO framework using this distance metric. Based on this framework we then derive an EA for many-objective optimisation. This EMO algorithm known as d–EMO retains the EA operators in NSGA-II, except for the fact that it does not use any non-dominated sorting mechanism. We then describe how this algorithm is more computationally efficient than NSGA-II and effective in solving many-objective problems.

In Section 3.2, we illustrate the functionality of d–EMO on various test problems of the ZDT, DTLZ and WFG test suites. Here, results are shown for some selected two, three, five and ten-objective problem instances that have various characteristics. This section also

describes a scheme that can be used by a DM to have better control of the solution spread. These include the process of defining preferred regions and also imposing bias on selected objectives.

We describe results on how d–EMO compares against NSGA-II in many-objective problem instances. Here, we specifically concentrate on difficult many-objective problems that are not described in other preference based EMO algorithms in the literature. We emphasise the fact that d–EMO is able to find solutions in various many-objective problems, including the more challenging ones. Finally, this section paves a way to understanding the need for better metrics to compare user-preference EMO algorithms.

3.1 The distance metric

One of the main research questions that was set forth in this thesis was to explore mechanisms to integrate user-preferences to EMO algorithms, which produce more efficient algorithms than the state-of-the-art. In Section 2.6.4 (page 43), we described some of the existing user-preference based EMO algorithms and their shortcomings. In our early experiments, we extended the work by Deb et al. [2006] by introducing reference point based MOPSO algorithms [Wickramasinghe and Li, 2008]. These MOPSO algorithms were based on the underlying NSPSO [Li, 2003] and maximinPSO [Li, 2004] algorithms.

One of the main issues with these algorithms — including the MOPSO versions used by us — was the fact that they still rely on dominance comparisons, which has proven to be ineffective in many-objective problems. Therefore, in this research we have explored methods to avoid using the dominance relation with the use of user-preferences. To this end, we propose a *distance metric* for EMO algorithms that will enable the algorithms to find solutions efficiently in many-objective problems without a need for dominance comparisons. Here onwards we use the term EMO to refer to Evolutionary Many-objective Optimisation, because the main focus of this thesis is on many-objective problems.

3.1.1 Defining the distance metric

In the MCDM literature, classical user-preference methods were used to find a single solution on the Pareto front. This single solution would be the closest point to a reference point (see Section 2.6.2 – page 41) on the Pareto front or the middle point (see Section 2.6.3 – page 42) derived from the light beam search. We introduce a distance metric based on the process of obtaining this solution point.

Let us define the *distance metric* of an individual $\vec{x}$ to a reference point $\vec{\bar{z}}$ using Equation 2.16 as:

$$dist(\vec{x}) = \max_{i=1,\ldots,M} \{w_i(f_i(\vec{x}) - \bar{z}_i)\} \tag{3.1}$$

Similarly in the light beam search, for any individual $\vec{x}$, its distance metric is defined using Equation 2.18 as:

$$dist(\vec{x}) = \max_{i=1,\ldots,M} \{\lambda_i (f_i(\vec{x}) - z_i^r)\} + \rho \sum_{i=1}^{M} (f_i(\vec{x}) - z_i^r) \tag{3.2}$$

We consider a solution $\vec{a}$ to be better than $\vec{b}$ ($\vec{a}$ is fitter than $\vec{b}$) if and only if $dist(\vec{a}) < dist(\vec{b})$. Essentially, we replace the dominance relations in an EMO algorithm with this distance metric $dist()$. We present a generalised EMO algorithm using the distance metric later in Section 3.1.3.

Using this distance metric[1], an EMO algorithm can be guided to converge onto certain regions of the Pareto optimal front. Figure 3.1 illustrates the search directions of an EMO algorithm as represented by the reference point method and light beam search. The search direction is dictated by the distance metric derived from the reference point $\vec{\bar{z}}$ (Figure 3.1(a)). However, the DM also has the ability to change the direction of search by adjusting the AP and RPs in the light beam search (Figure 3.1(b)). The term *intersection point* ($\vec{u}$) is used to identify the solution point ($\vec{z'}$) obtained by the reference point method or the middle point ($\vec{z^c}$) derived by the light beam search.

It is important to realise that the distance metric derived from a reference point does not make the population converge onto the reference point, but onto the Pareto front. The proposed distance metric provides both negative and positive values to each individual, depending on its location in the objective-space. If the individual is above the reference point the distance metric will be positive and when the individual is below the reference point the distance metric will be negative. Fitter individuals in the population have smaller distance metric values. Therefore, the search-pressure provided by the fitter individuals guides the EMO algorithm towards the Pareto front. However, if one uses a Euclidean distance based metric — which is always ≥ 0 — as in the reference point NSGA-II algorithm (see Section 2.6.4 – page 43) the population would converge onto the reference point if the reference

[1]It is important to note that we do not use the term *distance* in the true mathematical sense of linear measurement. We use it as a proper noun to refer to the *dist()* function.

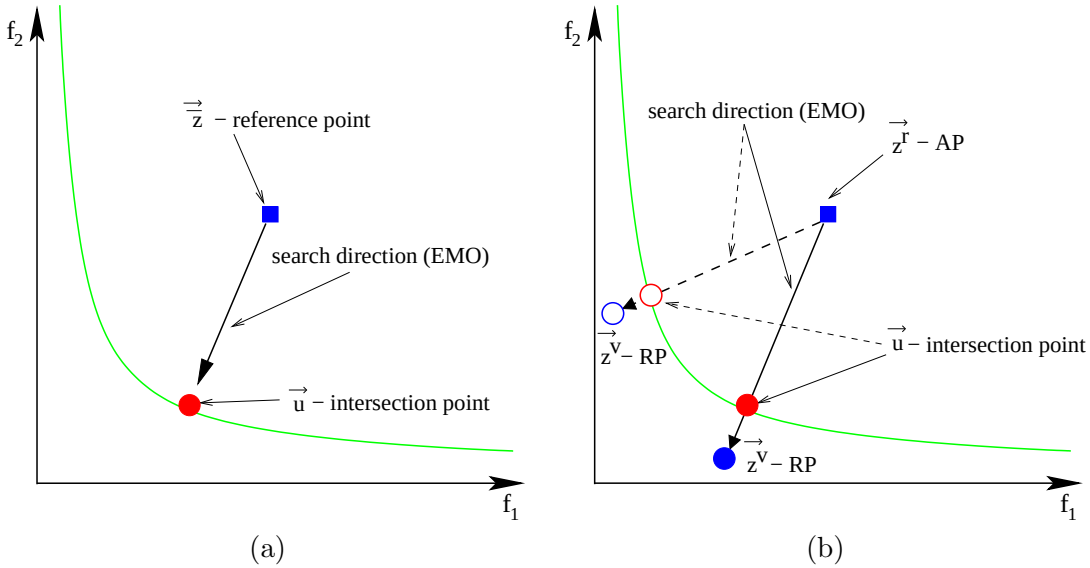


Figure 3.1: The directions of search dictated by the preferences in EMO algorithms: (a) Using the reference point method; (b) Using the light beam search.

point is in the feasible regions of the objective-space. In our distance metric approach such a situation does not arise because of both the positive and negative values. Therefore, the DM has the freedom to select any reference point in the objective-space without having prior knowledge of the location of the Pareto front. The distance metric based EMO algorithm will converge onto the Pareto front regardless of the reference point being in the feasible or infeasible regions of the objective-space.

3.1.2 Controlling the spread of solutions

As seen in Figure 3.1, if there is no control of the spread of solutions, the EMO algorithm will explore the search-space along the given search-direction and converge to the intersection point $\vec{u}$. This is the expected behaviour if one is to integrate a classical user-preference approach into an EMO algorithm. With the advantage of having a population of solutions, a DM would be more interested in obtaining a spread (or set) of solutions. It is essential to obtain a set of solutions rather than a single solution to reflect the user-preferences better. This idea has also been shown to be useful in EMO algorithms previously [Jin and Sendhoff, 2002].

To have control over the spread of solutions we define a threshold value δ (> 0) for the distance metric using the notion of outranking (see Section 2.6.3 – page 42). It is important

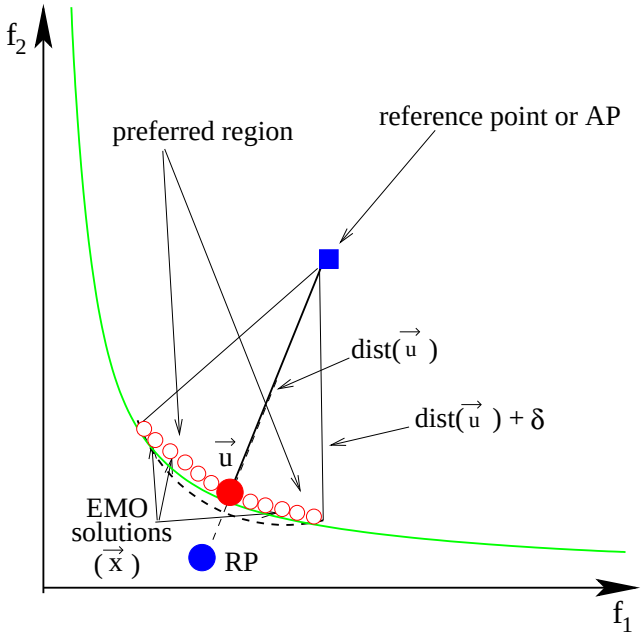


Figure 3.2: Defining the spread of solutions using the δ value.

to note that the formulation of outranking used in our user-preference algorithms are based on the distance metric rather than the threshold values in the classical light beam search.

Here, our aim is to obtain a set of solutions around the intersection point $\vec{u}$. More specifically, to allow the EMO algorithm to converge to an area of solution points rather than a single solution. If $\vec{x}$ is any solution point we define $\vec{x}S\vec{u}$ as:

$$\vec{x}S\vec{u} \Leftrightarrow dist(\vec{x}) < dist(\vec{u}) + \delta \tag{3.3}$$

This relation allows the EMO algorithm to converge not only to the solution point $\vec{u}$, but to any other solution $\vec{x}$ around $\vec{u}$ as long as $\vec{x}S\vec{u}$. In other words, this relation says that $\vec{x}$ is as good as $\vec{u}$ within limits of δ.

All the solutions outranking $\vec{u}$ by a given δ defines a preferred region. Figure 3.2 illustrates the formation of a preferred region. Here, we have combined the preference points of reference point and light beam into one diagram for brevity.

In this approach of using a distance metric for EMO algorithms, a preferred region consists of:

1. The search direction: as given by the preference method (either reference point or light beam search); and

2. The spread of solutions: as defined by the threshold δ.

Using different values for δ, a DM has control over the spread of solutions. A larger value of δ provides a larger spread (or larger preferred region) and a smaller value provides a smaller spread (or smaller preferred region). $\delta = 0$ gives the single solution point $\vec{u}$. Essentially, a very large value for δ ($\rightarrow \infty$) will give the entire solution front. Next, we describe a user-preference EMO algorithm incorporating this distance metric.

3.1.3 A distance metric based EMO algorithm: *d*–EMO

The distance metric integrated into an EMO algorithm is described in Algorithm 11. We refer to this EMO algorithm as d–EMO. Here, we have used the original NSGA-II algorithm and replaced the non-dominated sorting procedure by integrating the proposed distance metric. In fact, the distance metric can be easily integrated into any population based stochastic optimisation method such as multi-objective PSO or DE algorithms. For example, in Chapter 4, we describe a hybrid DE and PSO version based on this distance metric. Therefore, d–EMO can be considered as a general framework for integrating user-preferences into EMO algorithms.

Algorithm 11 The d–EMO framework

1: OBTAIN preferences from the DM
2: INITIALISE a population of size n, $t = 0$
3: EVALUATE each individual and assign fitness, and distance to the preferred region
4: **repeat**
5: SELECT parents (P_t) using tournament selection
6: RECOMBINE parents to create offspring (Q_t)
7: MUTATE the resulting offspring
8: EVALUATE each offspring and assign fitness, and distance to the preferred region
9: COMBINE the parent and offspring populations ($R_t = P_t \cup Q_t$)
10: SORT R_t to obtain the next generation P_{t+1}
11: $t = t + 1$
12: **until** $t = t_{MAX}$

We first describe d–EMO having a single preferred region. As in NSGA-II, the parent population at iteration t is P_t and the offspring population is Q_t. The combined population of parents and offspring at iteration t is R_t ($R_t = P_t \cup Q_t$). Algorithm 11 differs from a traditional EMO algorithm because of the distance metric assignment in Steps 3, 8 and 10.

The main steps of the d–EMO algorithm are:

Step 1: Obtain preferences from the DM (line 1)

The DM will first choose a preference method; either the reference point method or light beam search. Depending on the preference method the DM will provide aspiration values to indicate a reference point or an AP and RP. The DM will next provide a δ value indicating the spread of solutions. The DM has the freedom to indicate any points on the objective-space without considering them to be *feasible* or *infeasible*. In a real-world scenario a DM will have some knowledge on the problem domain, but the Pareto optimal front might not be known for that specific problem. Therefore, a user-preference EMO algorithm should be able to handle the aspiration values regardless of their feasibility.

Step 2: Initialise the population (line 2)

A population (P_0) of size n is first initialised. Here, random values for the decision variables ($x_{j,i}$) for each individual ($\vec{x_i}$) are obtained by:

$$x_{j,i} = rand(0,1) \times \left(x_j^{(UB)} - x_j^{(LB)}\right) + x_j^{(LB)} \tag{3.4}$$

The individuals are then evaluated using the objective functions and fitness is assigned. Next, each individual's distance metric is assigned using Equations 3.1 or 3.2 depending on the preference method.

Step 3: Select parents (line 5) and Step 4: Generating offspring (line 6)

Parents are selected using a tournament selection size of two, to create P_t. Here, individuals closest to the preferred region are given priority. The fittest individuals are considered to have the lowest $dist()$ values. The distance metric replaces the dominance relation as the mechanism to compare individuals. The parents will crossover and mutate to produce offspring (Q_t). d–EMO uses the SBX crossover and polynomial mutation as in NSGA-II. The parent population of size n will create n number of offspring. The offspring are evaluated to assign fitness and distance metric.

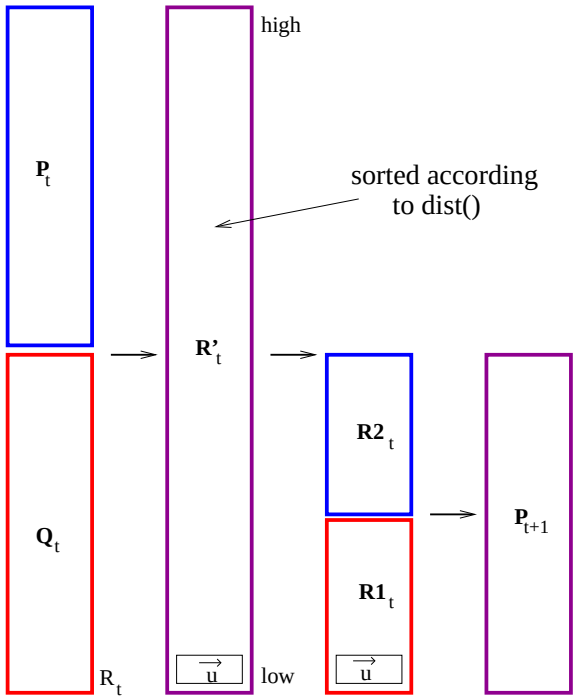


Figure 3.3: The process of obtaining the next generation using the distance metric.

Step 5: Obtain next generation (line 10)

The parent population of size n is combined with the offspring population of size n to create a population of size $2n$ (R_t). From this combined population, n number of individuals (P_{t+1}) are selected to move to the next iteration.

More specifically, as illustrated in Figure 3.3, first, the $2n$ population is sorted in descending order of their distance metric values. This set is referred as R'_t. The individual with the lowest distance metric value is selected as the intersection point $\vec{u}$. Next a maximum of $\frac{n}{2} - 1$ number of individuals which outrank $\vec{u}$ are selected in descending order of the distance metric. This set is given as $R1_t$. If $R1_t$ contains less than $\frac{n}{2}$ individuals, the next individuals with the lowest distance metric is added to $R1_t$. This process is repeated until the size of $R1_t$ is $\frac{n}{2}$. Next a $\frac{n}{2}$ number of random individuals ($R2_t$) are selected from the rest of the population (which are not in $R1_t$) in order to provide more diversity to the population. This n number of individuals move to the next generation ($P_{t+1} = R1_t \cup R2_t$). Essentially we have replaced the non-dominated sorting procedure in NSGA-II with a simplified sorting procedure using the distance metric.

To prevent the algorithm exploring too far from the preferred regions we employ a mechanism of exploration and exploitation. Basically, for a predefined period of generations d–EMO will either perform an exploration or exploitation process in an alternating manner. Towards the end of the execution the exploitation process is used to fine tune the convergence onto the preferred regions. The exploration process is the process we described using Figure 3.3 where $|R1_t| = \frac{n}{2}$. In the exploitation process individuals are selected in a similar manner such that $|R1_t| = n$. In other words, random individuals are not included into the next generation making $|R2_t| = 0$ and essentially making $P_{t+1} = R1_t$.

Similar to NSGA-II, d–EMO terminates when the maximum number of generations (t_{MAX}) is reached. At the end of the execution the final solution set is obtained by extracting the first non-dominated front from the population. It is important to note that this is the only instance where a non-dominated sorting procedure is used.

The distance metric approach guides the population towards the preferred region so that solutions are found on the Pareto front as seen in Figure 3.2. Although Equations 3.1 and 3.2 provide a scalar value, the aim is not to optimise that value, but to use the value as a distance metric to guide the population. The closer the individual is to the preferred region, the better it is. This behaviour differentiates d–EMO from traditional MCDM approaches, where the many-objective problem is converted to a single-objective problem for optimisation.

3.1.4 Providing more than one preferred region

If the DM wishes to include more than one preferred region, only minor changes are needed for d–EMO. The population is divided equally into sub-populations, such that each sub-population is assigned to one preferred region. As in the standard d–EMO algorithm the search process is done within each of these sub-populations.

We have opted for a mechanism that explores each preferred region in parallel within a single execution run, rather than to explore each preferred region separately in different execution runs. The main reason behind this approach is that for computationally expensive optimisation problems (we describe some in Chapters 6 and 7) having multiple runs to explore separate preferred regions is more expensive than exploring the preferred regions in parallel within one run.

Table 3.1: The parameters used for d–EMO on the various many-objective problems.

Parameter	**Value**
Population size (n)	200
Maximum number of generations (T_{MAX})	500
SBX crossover probability	0.9
Polynomial mutation probability	$\frac{1}{N}$
Ratio of spread for offspring — used in the crossover and mutation schemes (β)	20.0

3.1.5 Computational complexity of *d*–EMO

A dominance comparison based EMO algorithm normally has a computational complexity of $O(Mn^2)$ for each iteration because of the non-dominated sorting procedure (see Section 2.5.1 – page 30). However, d–EMO only depends on the sorting of the population using the distance metric. As a result, the computational complexity for the entire population is $O(n \log n)$. This shows that the distance metric based approach is not only simpler but also more computationally efficient than the dominance relation based approaches.

3.2 Experimental results

We used some test problems to illustrate the functionality of d–EMO. The ZDT [Zitzler and Thiele, 1999] test problems were used for two-objective problems and the DTLZ [Deb et al., 2005b] test problems were used for two to ten objective problems. We have also used the WFG [Huband et al., 2006] test problem suite for problem instances having two or more objectives. For each test problem, we used the corresponding number of decision variables for each problem instance as described in Section 2.2 (page 16). Using the various test problems found in each suite we illustrate the convergence of d–EMO on the Pareto optimal front. d–EMO was executed on each problem instance for 30 independent runs. The results we present here are from the best run of each problem instance. The parameters given in Table 3.1 were used for d–EMO regardless of the problem instance.

We noticed from the initial experiments that for simpler problems like ZDT1–3 and DTLZ2 the population was able to converge in about 100–150 generations. However, we have used $T_{MAX} = 500$ to allow d–EMO to converge to the global fronts in all problem instances.

We first illustrate[2] behaviour of the exploration and exploitation phases of d–EMO. Next, the features of the reference point method and light beam search that are available to the DM to define the preferred regions. These define the spread (using δ) in both preference methods and allowing bias (using weights w_i) in the reference point method. We illustrate the control of the spread and bias on two and three-objective problems instances here because of the ease of visualisation. The mechanisms for controlling the algorithm are still valid for any many-objective problem.

3.2.1 Behaviour of the exploration and exploitation phases

Figure 3.4 illustrates an example execution of d–EMO on ZDT1 at different iterations. We illustrate the behaviour of d–EMO using a two-objective problem for ease of visualisation. Here the reference point is $(0.5, 0.5)$ and $\delta = 0.05$. As mentioned before, d–EMO was able to converge to the Pareto front at around iteration 150. The exploration phases can be seen in Figures 3.4(b) and (d), while Figures 3.4(c) and (e) represent the exploitation phase. Although the final solutions (first non-dominated front) given in Figure 3.4(f) include a subset of the population (since ZDT1 is a two-objective problem), in a many-objective problem the final solution set will include the entire population. This is because all solutions will be non-dominated to each other according to the standard dominance relation.

[2]In the figures, the final solution points are in red while the points in green represent the known Pareto front. Aspiration points are in blue. We use this colour scheme throughout this thesis.

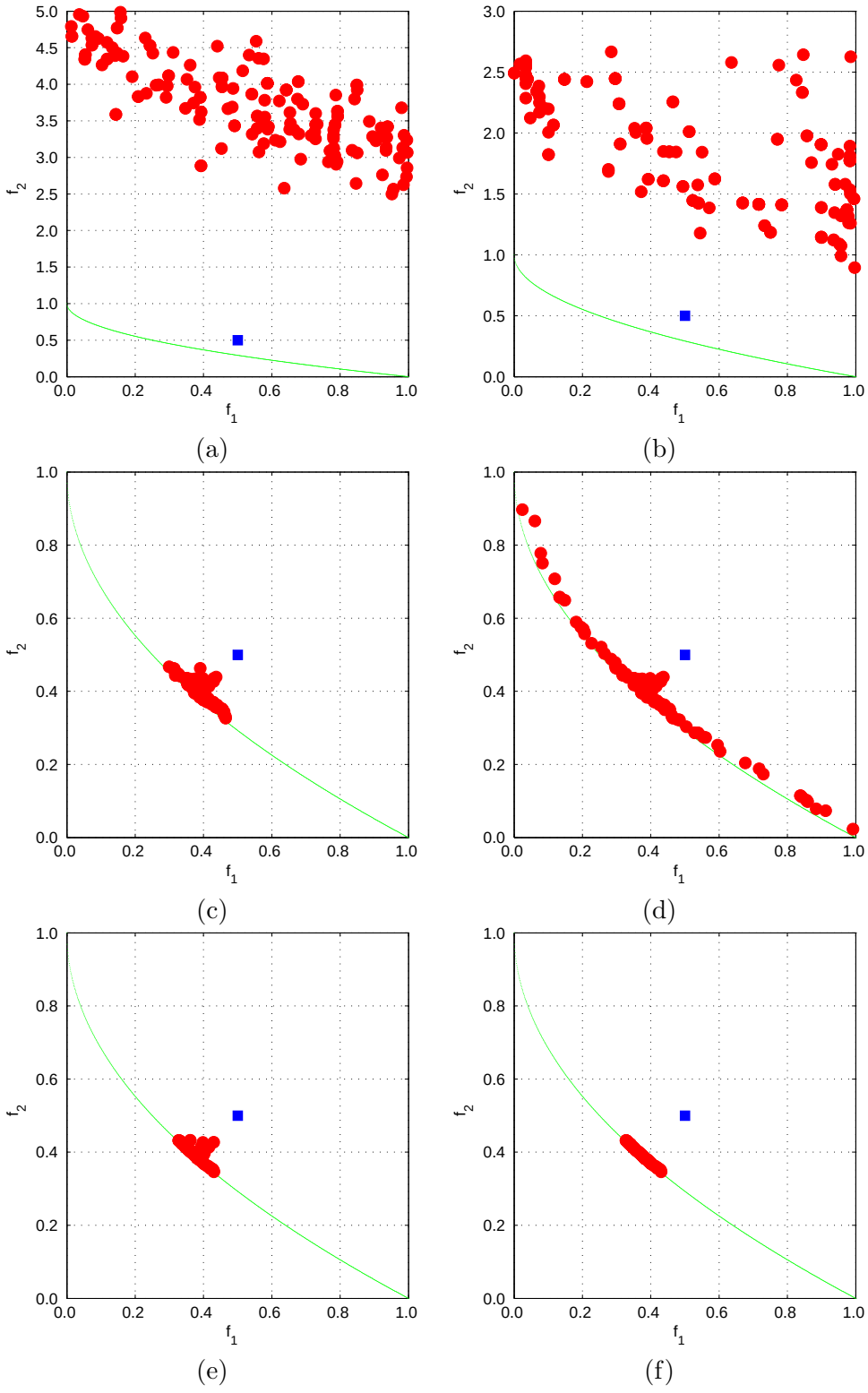


Figure 3.4: Execution of d–EMO on ZDT1: (a) Initialisation; (b) Iteration 20; (c) Iteration 70; (d) Iteration 92; (e) Iteration 150, all solutions; (f) Iteration 150, final solutions.

3.2.2 Varying spread in reference point approaches

Figures 3.5 and 3.6 illustrate how the DM can define the preferred regions by varying δ. These figures illustrate how the population converges on the global front at the end of the execution. A smaller value for δ provides a smaller spread of solutions, while a larger value for δ provides a larger spread. Although ZDT1 ($N = 30$) was one of the simplest problems, DTLZ1 ($N = 7$) is a more challenging multi-modal problem.

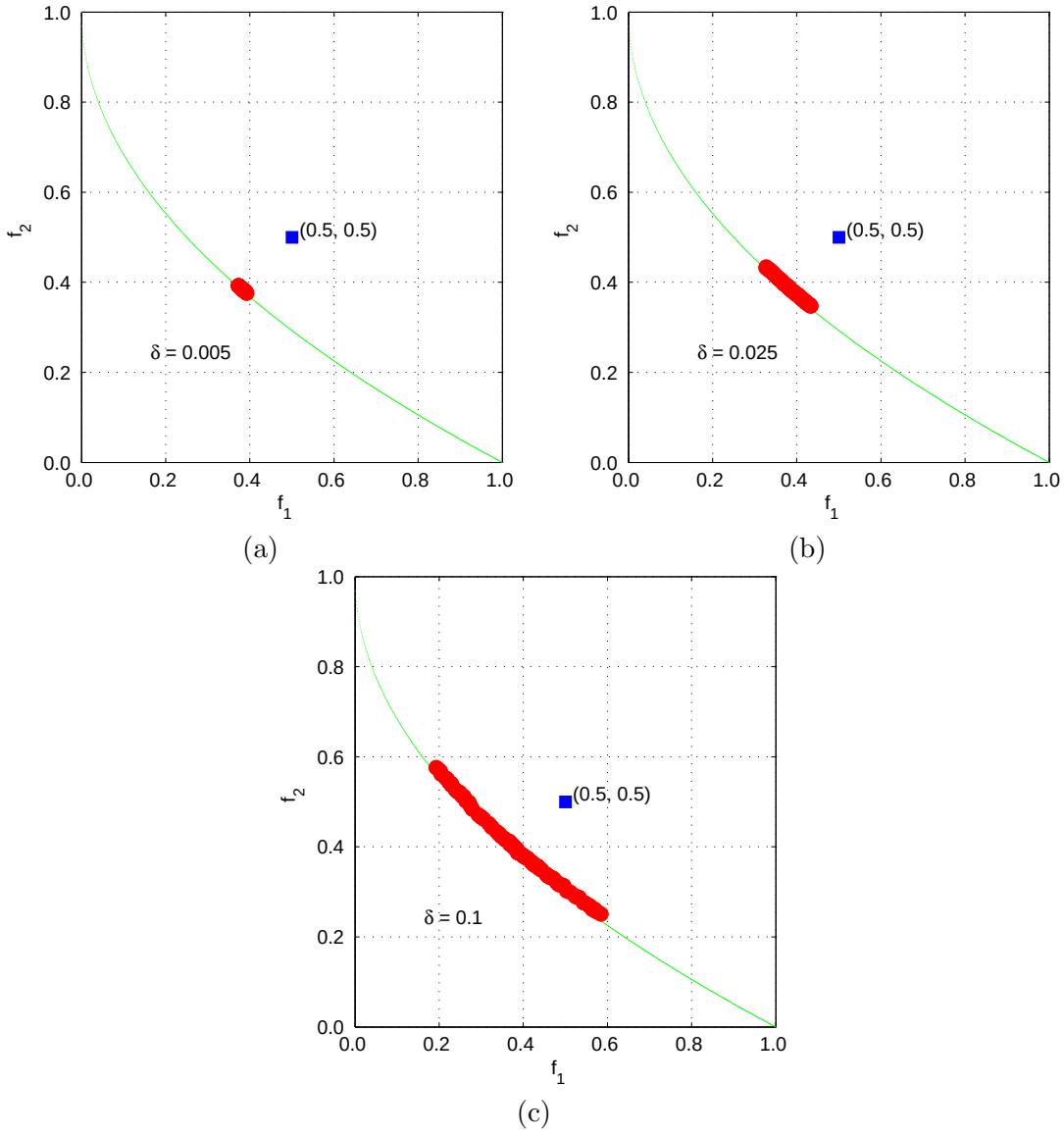


Figure 3.5: ZDT1 with varying spread values for using one reference point: (a) $\delta = 0.005$; (b) $\delta = 0.025$; (c) $\delta = 0.1$.

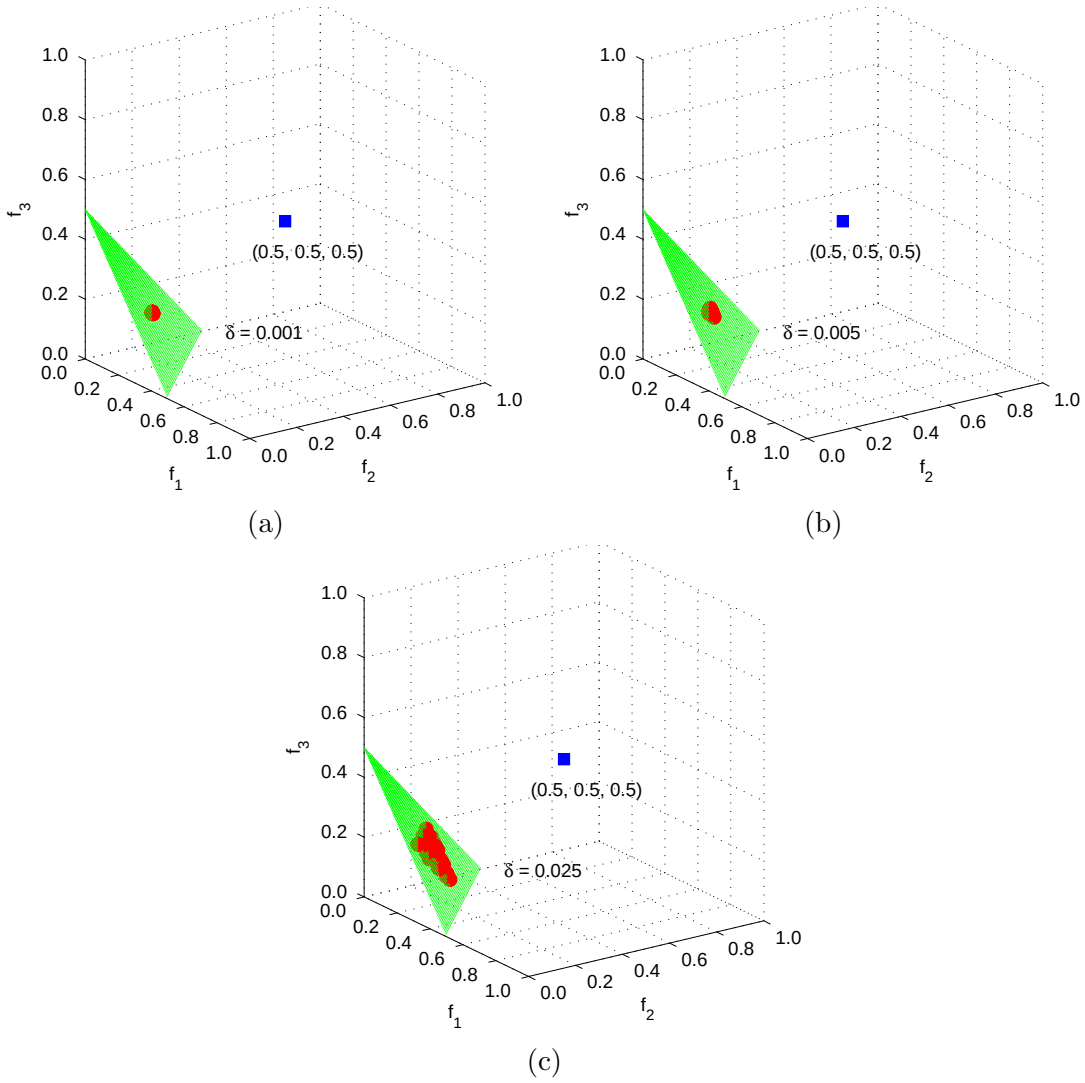


Figure 3.6: DTLZ1 having three-objectives with varying spread values using one reference point: (a) $\delta = 0.001$; (b) $\delta = 0.005$; (c) $\delta = 0.025$.

3.2.3 Introducing bias in reference point approaches

Using different values for the weights in the distance metric (Equation 3.1) the DM can provide a bias towards objectives. Figure 3.7 illustrates examples for bias on ZDT2 ($N = 30$). Here, $\delta = 0.05$ was used for all three problem instances. If the DM is not interested in any bias towards the objectives, then each weight can be set to 1.0 or to some equal values (Figure 3.7(b)) aggregating to 1.0.

3.2.4 Selected results for multi-objective problems using reference points

Among the various problems available in the three test problem suites, we present some of the most interesting results. Each problem instance had two preferred regions consisting of different reference points and spread values. The figures illustrate the solutions at the end of an execution run.

Two-objective problems

Figure 3.8 shows the solution fronts obtained for some selected two-objective problems. All the problem instances had preferred regions defined by some reference points having $\delta = 0.01$ and $\delta = 0.05$.

Figure 3.8(a) illustrates the result for ZDT4 having $N = 10$, which is a multi-modal two-objective problem. The two reference points are in the feasible region of the objective-space. d–EMO was successful in guiding the population across the many local fronts to converge on the global front. In the ZDT6 problem ($N = 10$) illustrated Figure 3.8(b) the two reference points are in the infeasible region of the objective-space. The reference point $(0.2, 0.9)$ is outside the region bounded by the Pareto optimal front. However, d–EMO successfully obtained the solutions near the reference point that is at the extreme end of the Pareto front. Figures 3.8(c), (d) and (e) indicate situations where the DM had specified the reference points far from the Pareto front. However, d–EMO still found solutions closest to the reference points on the Pareto front. The two-objective WFG problems consisted of $N = 6$ decision variables, which comprised of two position variables and four distance variables. The result for the two-objective DTLZ7 problem ($N = 21$) is seen in Figure 3.8(f). Here one reference point is in the feasible region and one is in the infeasible region. The algorithm was still successful in locating solutions on the Pareto fronts.

We used the reference points $(0.5, 0.5)$ and $(0.2, 0.9)$ in most of the test problems to illustrate that the DM has the freedom to choose reference points without any prior knowledge

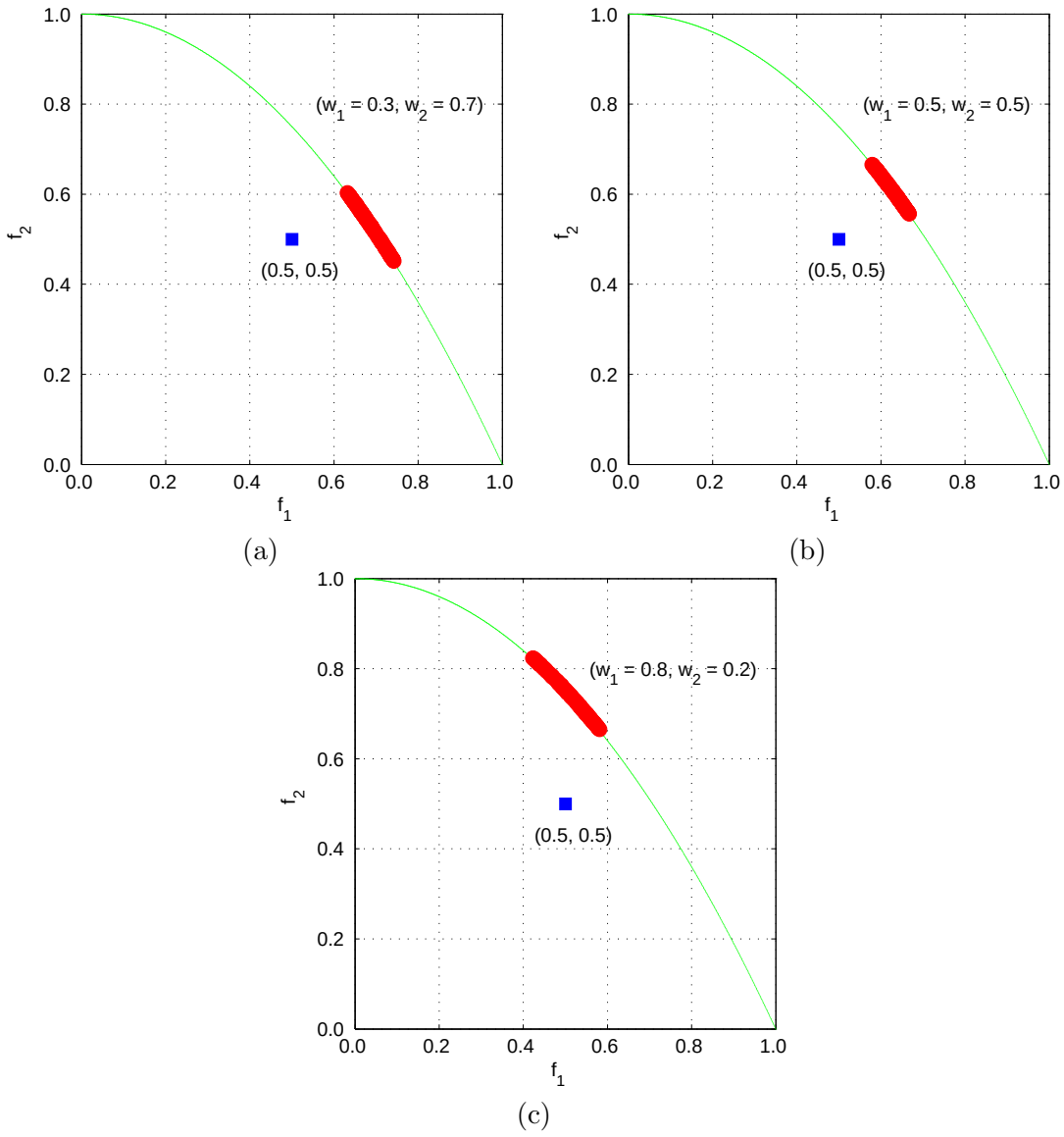


Figure 3.7: ZDT2 with varying bias values using one reference point: (a) $w_1 = 0.3, w_2 = 0.7$*; (b)* $w_1 = 0.5, w_2 = 0.5$*; (c)* $w_1 = 0.8, w_2 = 0.2$.

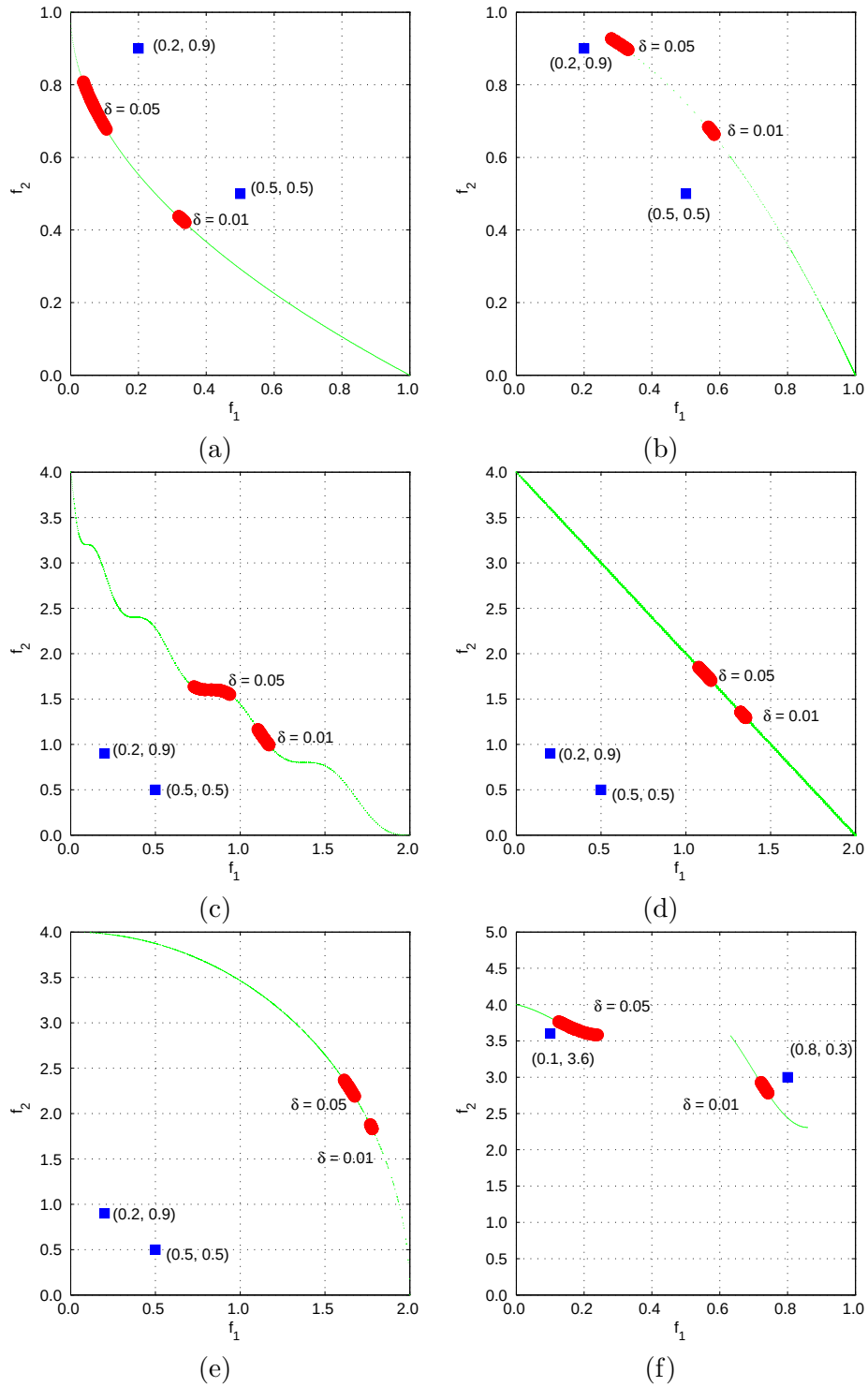


Figure 3.8: Results for two-objective problems using reference points: (a) ZDT4; (b) ZDT6; (c) WFG1; (d) WFG3; (e) WFG4; (f) DTLZ7.

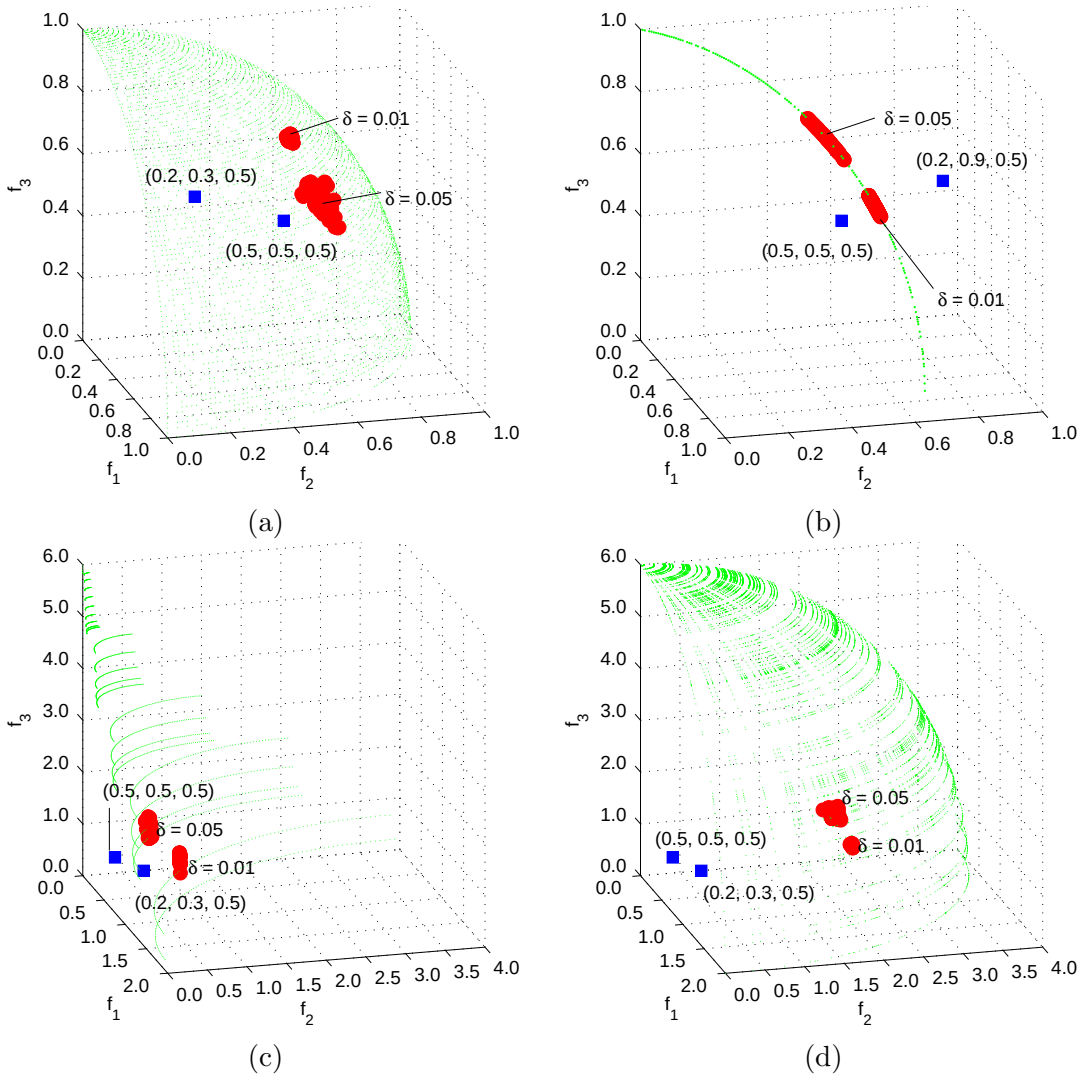


Figure 3.9: Results of the three-objective problems using reference points: (a) DTLZ3; (b) DTLZ6; (c) WFG2; (d) WFG4.

of the location of the Pareto front. In some instances these reference points were feasible, while in others they were infeasible. *d*–EMO was able to handle each of these situations successfully. This property of *d*–EMO is very useful in a real-world problem where the Pareto front maybe unknown and the DM will need to indicate the preference based on desired values for the objectives.

Three-objective problems

Figure 3.9 illustrates some results from the three-objective problems. Here d–EMO was also able to converge onto the global fronts of the preferred regions in each problem instance. The number of decision variables for the DTLZ3 and DTLZ6 problems was set at $N = 12$. The WFG2 and WFG4 problems had $N = 2 + 4$ decision variables (two position variables and four distance variables). In the DTLZ6 instance the spread $\delta = 0.05$ is for the reference point given by $(0.2, 0.9, 0.5)$ and for all the other preferred regions the reference points are in the vicinity of the solution points. DTLZ6 is an unique problem because the Pareto front on a three-objective space is two-dimensional. d–EMO will converge to a line segment, near the region closest to the reference point.

3.2.5 Varying spreads in light beam search approaches

Similar to the reference point approach, the DM can vary δ to obtain a different spread of solutions. Figures 3.10 and 3.11 illustrate the convergence of the population on different preferred regions at the end of the execution. Here also the population has converged onto the global fronts of WFG2 ($N = 2 + 4$) and DTLZ2 ($N = 12$).

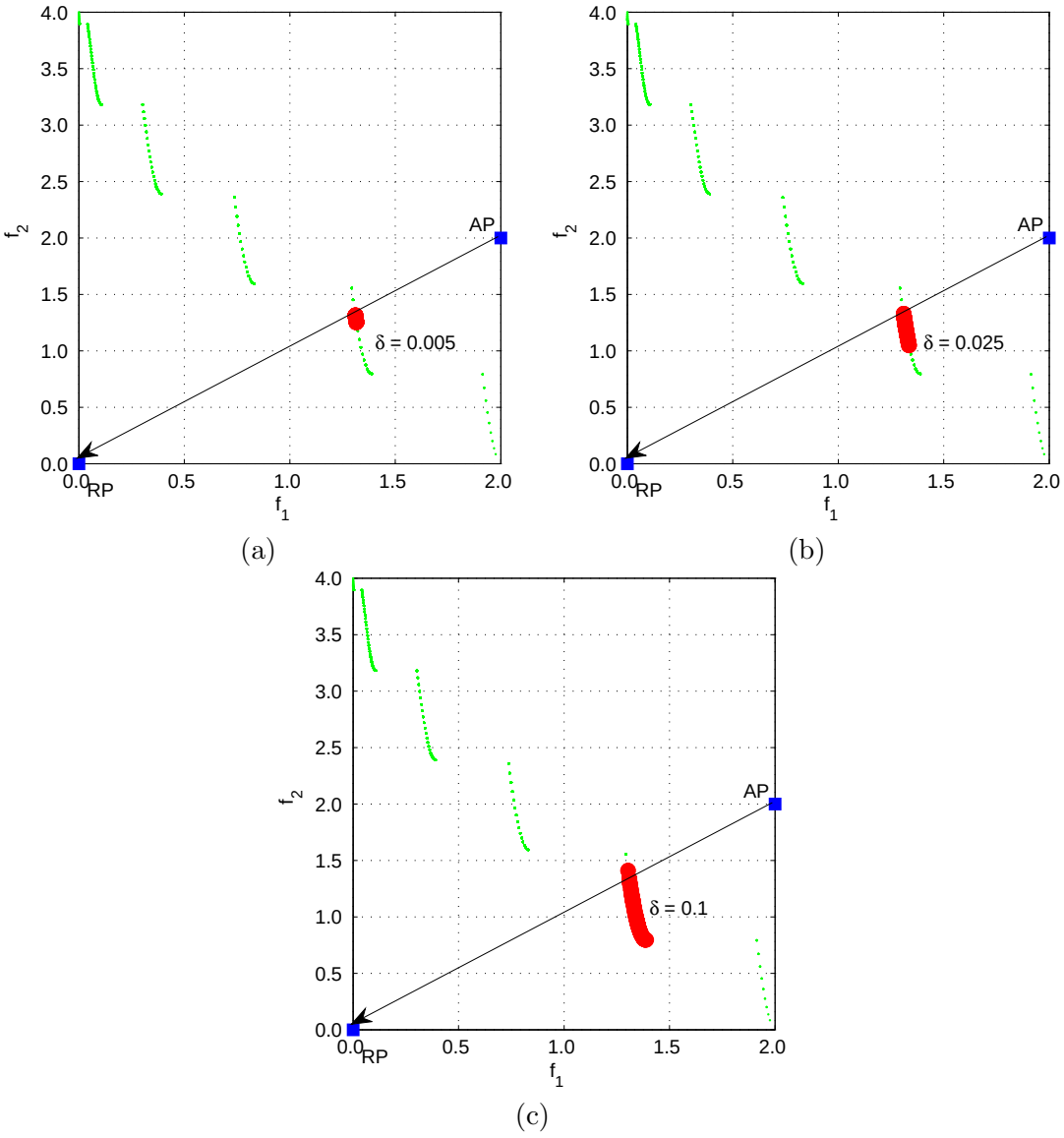


Figure 3.10: WFG2 having two-objectives with varying spread values with a light beam: (a) $\delta = 0.005$; (b) $\delta = 0.025$; (c) $\delta = 0.1$.

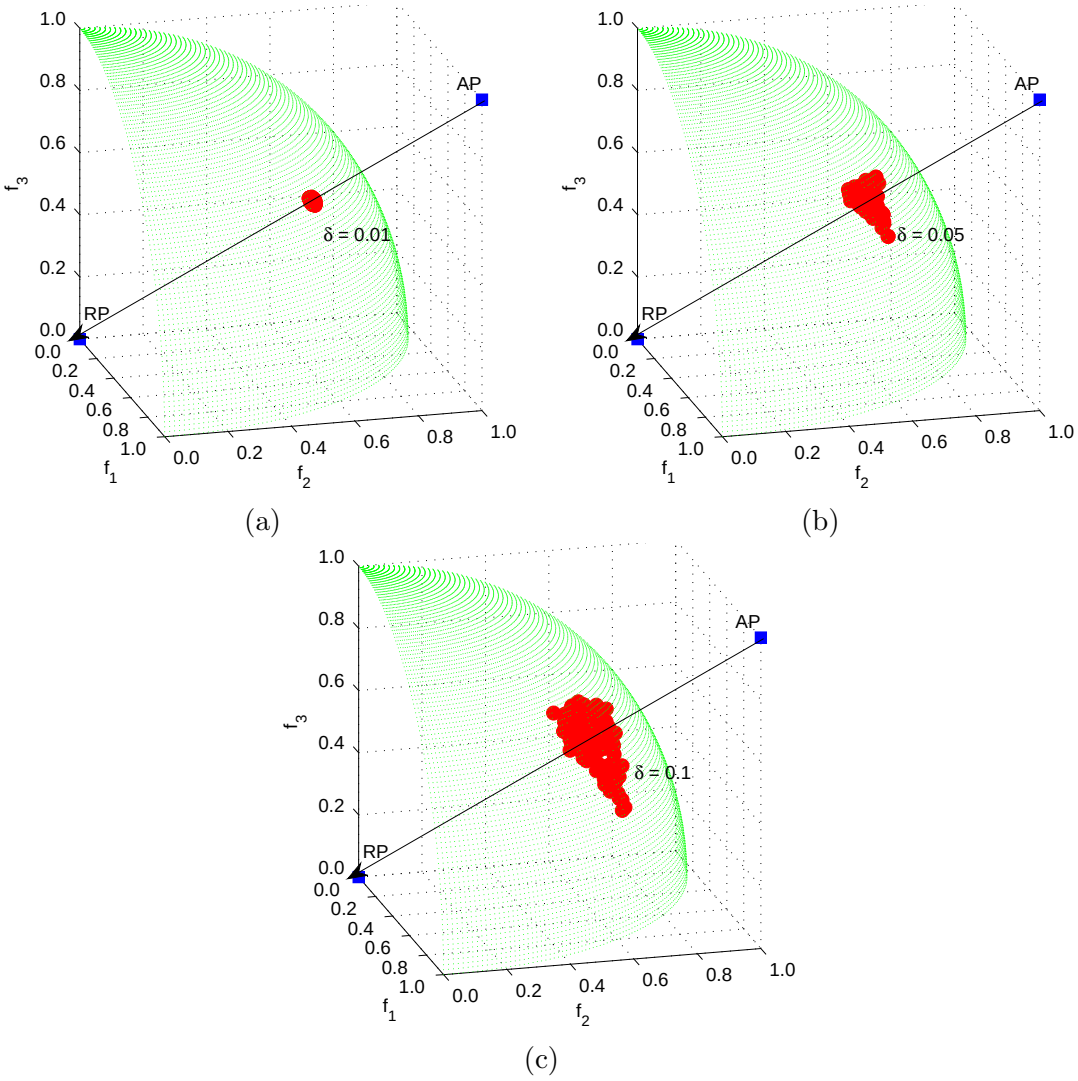


Figure 3.11: DTLZ2 having three-objectives with varying spread values with a light beam: (a) $\delta = 0.01$; (b) $\delta = 0.05$; (c) $\delta = 0.1$.

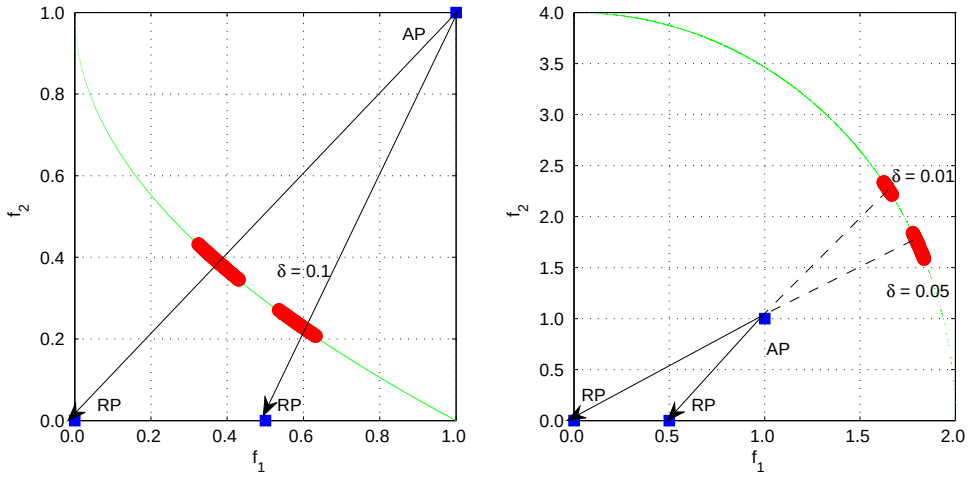


Figure 3.12: Results on two-objective problems using light beams: (a) ZDT4; (b) WFG4.

3.2.6 Selected results for multi-objective problems using light beams

For brevity, among the many test problems used in our experiments we present some selected results. We have used a mixture of one and two light beams with varying spreads. As in the reference point results, these are also the best results found at the end of the execution runs.

Two-objective problems

Figure 3.12 illustrates the results of using two light beams on the multi-modal ZDT4 ($N = 10$) and two-objective WFG4 ($N = 2+4$). Figure 3.12(a) shows two light beams originating from AP at $(1.0, 1.0)$ towards the RP at $(0.0, 0.0)$ and another RP at $(0.5, 0.0)$. ZDT4 is a very challenging problem for EMO algorithms because it has 21^9 local fronts. However, d–EMO is still able to converge onto the global Pareto front. A very interesting result can be seen for the two-objective WFG4 problem in Figure 3.12(b). Here, the light beams are located in the infeasible region of the objective-space, because both AP and RPs are infeasible. However, d–EMO with the light beam search still managed to guide the population in the direction of the light beams until solutions are located on the global Pareto front. This behaviour is present because d–EMO guides the population in the direction of the vector indicated by the AP and the RP. Therefore even if the APs and RPs are infeasible, d–EMO can locate the global optimal solutions on the preferred regions. This feature of d–EMO again shows that the DM has control when defining the preferred regions similar with reference points.

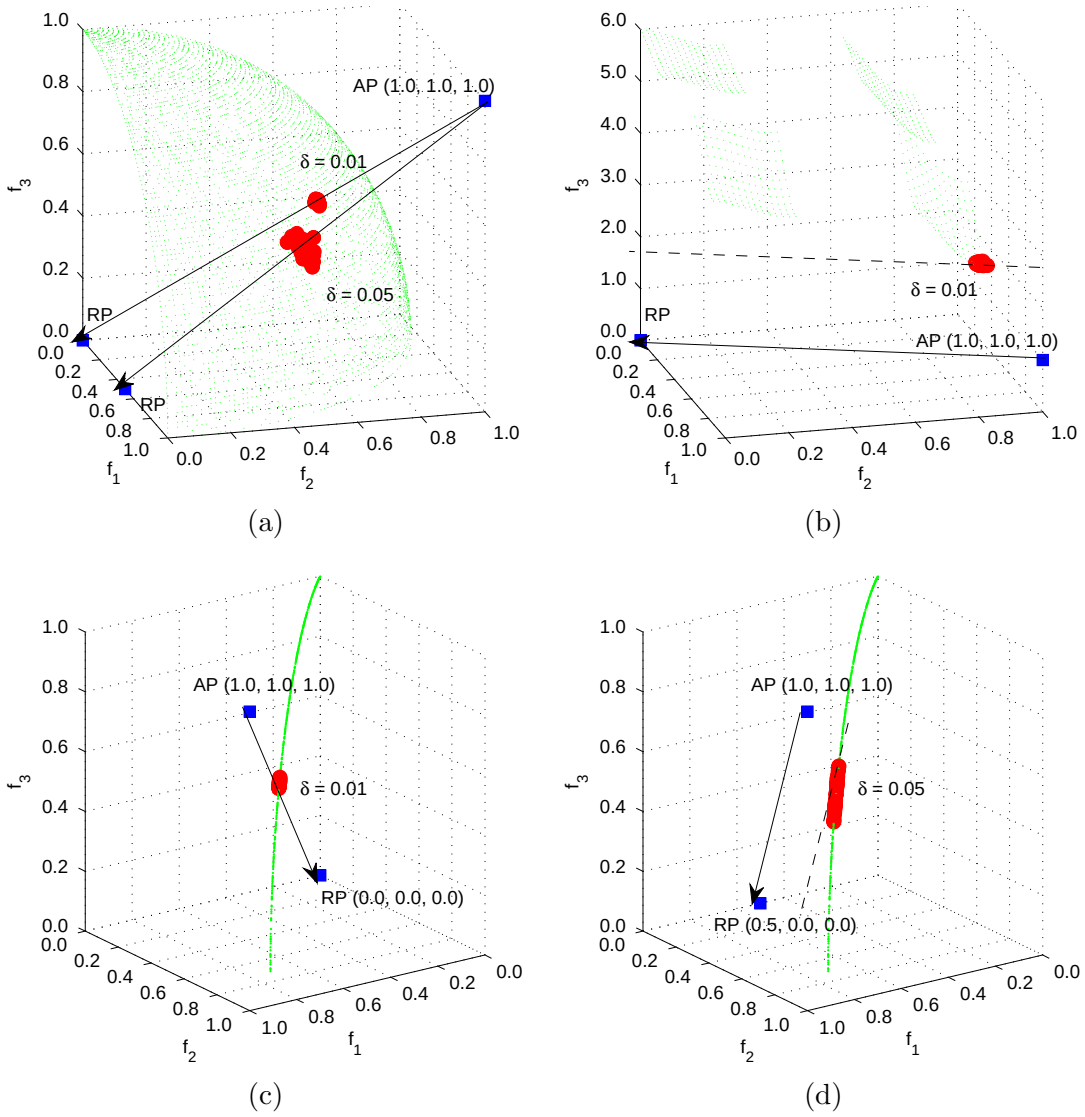


Figure 3.13: Results of three-objective problems using light beams: (a) DTLZ3; (b) DTLZ7; (c) DTLZ5 with a small spread; (d) DTLZ5 with a large spread.

Three-objective problems

DTLZ3 ($N = 12$) is one of the more difficult multi-modal problems having close to 3^{10} number of local Pareto fronts and one global Pareto front. In Figure 3.13(a) results are seen for two light beams originating from the AP at $(1.0, 1.0, 1.0)$ towards the RP at $(0.0, 0.0, 0.0)$ and another RP at $(0.5, 0.0, 0.0)$. We have also specified two spread values with $\delta = 0.01$ and $\delta = 0.05$. d–EMO using the light beam search is able to converge on the Pareto front

successfully. Figure 3.13(b) shows the result obtained for DTLZ7 ($N = 7$) with one light beam from $(1.0, 1.0, 1.0)$ to $(0.0, 0.0, 0.0)$. Although the light beam does not go through Pareto front, d–EMO was still able to locate solutions on the region of the Pareto front which is closest to this light beam. The distance metric guides individuals in the direction given by the vector from AP to RP. d–EMO has the ability to move in parallel along the direction of this vector until a middle point is found on the Pareto front because it is a population base algorithm. The *dashed* line parallel to the original light beam illustrates the vector that intersects the Pareto front. Hence that region illuminated from the *dashed light beam* is considered as the region closest to the original light beam.

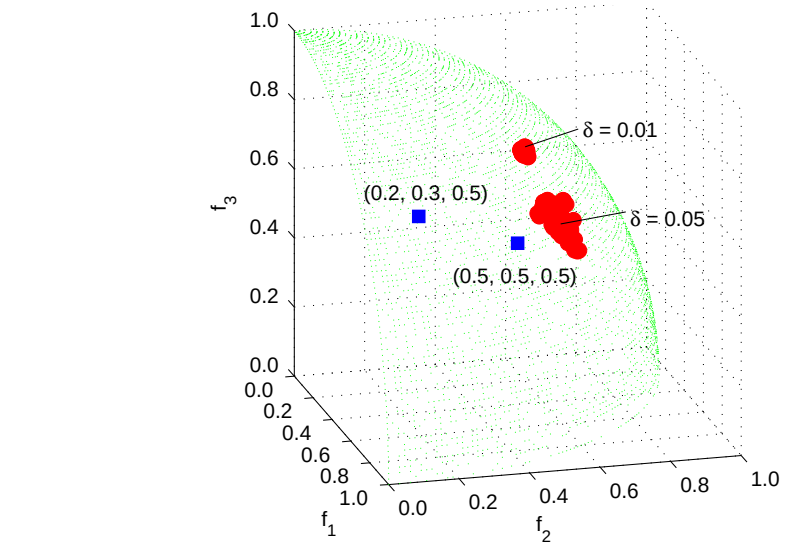
Figure 3.13(c) and (d) show the results obtained for DTLZ5 ($N = 12$), with a spread of $\delta = 0.01$ and $\delta = 0.05$ respectively. The three-objective DTLZ5 is similar to the three-objective DTLZ6, where the Pareto front is a line in the three-objective space. Here also we observe the behaviour of d–EMO when the light beam goes through the Pareto front (AP $(1.0, 1.0, 1.0)$ and RP $(0.0, 0.0, 0.0)$) and otherwise (AP $(1.0, 1.0, 1.0)$ and RP $(0.5, 0.0, 0.0)$). In Figure 3.13(d) we illustrate the region illuminated by a parallel dashed light beam similar to Figure 3.13(b).

3.3 *d*–EMO on many-objective problems

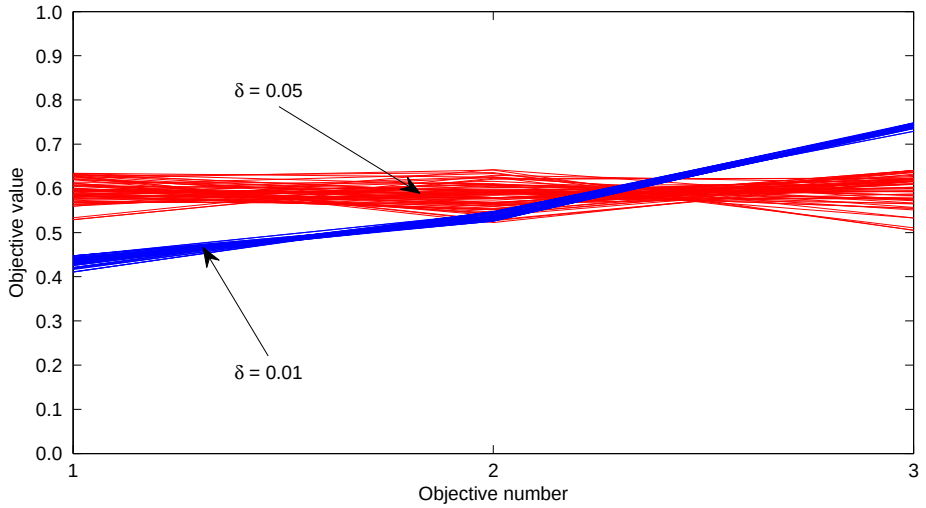
Up until now, we have used two and three-objective problem instances to illustrate the behaviour of the d–EMO framework. However, one of the research questions in this thesis is to use user-preferences in many-objective problems, so that EMO algorithms can efficiently find solutions. In many-objective problem instances, the visualisation of the final solution fronts becomes difficult. Therefore, we use a parallel axis plot to display the solutions instead of the traditional Cartesian coordinate plot.

3.3.1 Visualising a many-objective solution-space

In Figure 3.14 we illustrate the solutions found on a three-objective DTLZ3 problem instance using two reference points. On this parallel axis plot, the horizontal axis represents the objectives of the problem and the vertical axis represents the values for the objectives. Each line represents a solution point, where the intersection at the objectives' axis represents the value for that objective. We have omitted the points of the Pareto front seen in Figure 3.14(a) from the parallel axis plot in Figure 3.14(b) to make the plot clear. Otherwise there will be many lines interleaving the solutions points making them invisible. We use parallel axis plots

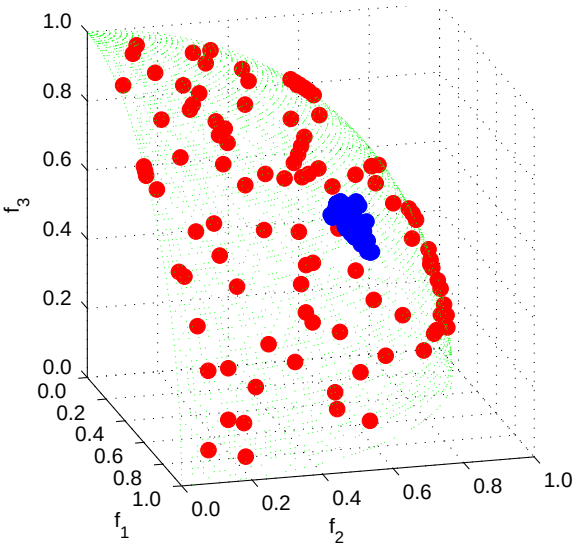


(a)

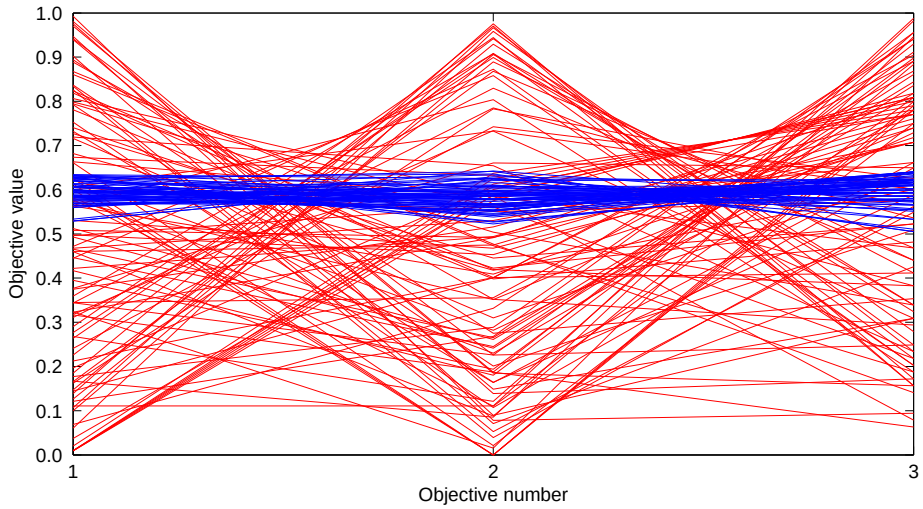


(b)

Figure 3.14: Visualisation of DTLZ3 with three-objectives: (a) On a Cartesian coordinate plot; (b) On a Parallel axis plot.



(a)



(b)

Figure 3.15: Some results of DTLZ3 with and without using user-preferences: (a) On a Cartesian coordinate plot; (b) On a Parallel axis plot. The solutions in blue are from reference point d–EMO.

from here onwards whenever many-objective results are illustrated. The clustering of the lines suggests that the solutions have converged to a preferred region, rather than spreading across the entire solution front. We can see clearly in Figure 3.14 how each preferred region is *visualised* in both forms of plots.

We use Figure 3.15 to clarify the expected differences in the parallel axis plots between solutions obtained on DTLZ3 with and without user-preferences. The solutions in blue are the ones obtained from reference point d–EMO, while the ones in red are from an EMO algorithm that does not have user-preferences. We expect solutions obtained from an user-preference EMO algorithm in many-objective problems to have a similar characteristics of lines clumped together in a parallel axis plot. If the algorithm obtained solutions spread across the objective-space then we would expect lines spread across all possible objective values in the parallel axis plot.

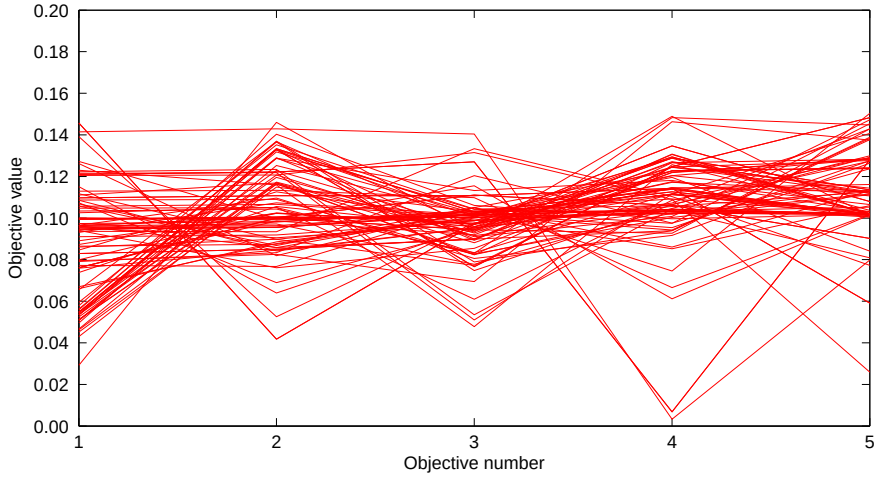
The many-objective parallel axis plots in the next sections will attempt to display the vertical axis in a manner that they enhance the preferred region rather than illustrating all the possible values for the entire Pareto front. We use this approach to give a clear visual representation of the solutions. We also deduce that the solutions are on or very close to the Pareto front using some mathematical relations of each test problem, complementing the plots.

To illustrate the effectiveness of d–EMO in many-objective problem instances we compared it against the standard NSGA-II algorithm. Similar to the d–EMO implementation, NSGA-II was also executed with the same parameter values in Table 3.1. NSGA-II was also executed for 30 runs on each problem instance and the best convergent run is illustrated.

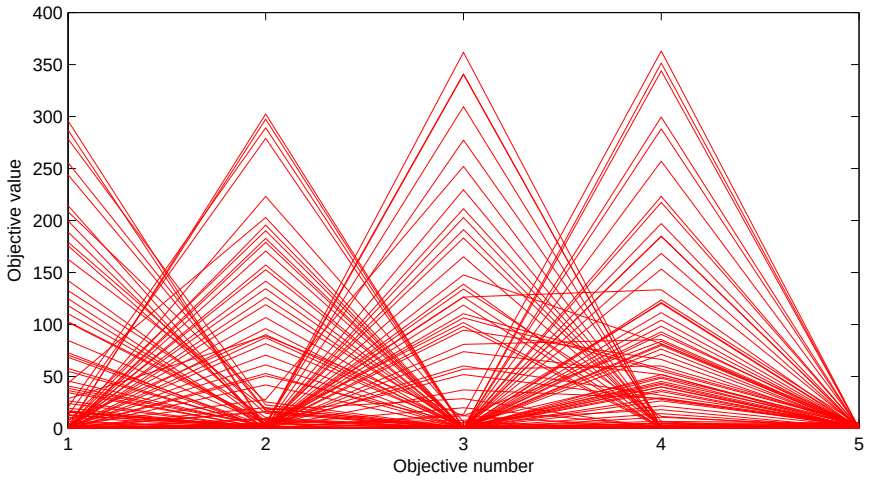
3.3.2 Using reference points

We used 0.5 for each objective to define a reference point and a spread of $\delta = 0.05$ and no bias. Figure 3.16 illustrates the results obtained for the multi-modal DTLZ1 problem having five-objectives and nine decision variables. We observe that d–EMO has been able to minimise the objectives better than NSGA-II and also has converged to the preferred region. It is important to note that the vertical axis of the two parallel axis plots have different ranges of values. We have illustrated the solutions points in such a manner to visualise the objectives values clearly. If both solution sets were included in one parallel axis plot, as seen in Figure 3.17, then the solutions from d–EMO would be represented as a line. This illustration can be misleading to think that all solution points have converged into a single point in the objective-space. Therefore, for clarity we use different ranges of values for the vertical axes of the parallel axis plots from here onwards.

Using the solutions we have also concluded that in d–EMO, the sum of the objective values of each individual was found to be in the range of $[0.5039, 0.5373]$, while for NSGA-II



(a)



(b)

Figure 3.16: Solution points of a five-objective DTLZ1 problem instance: (a) d–EMO with one reference point; (b) NSGA-II.

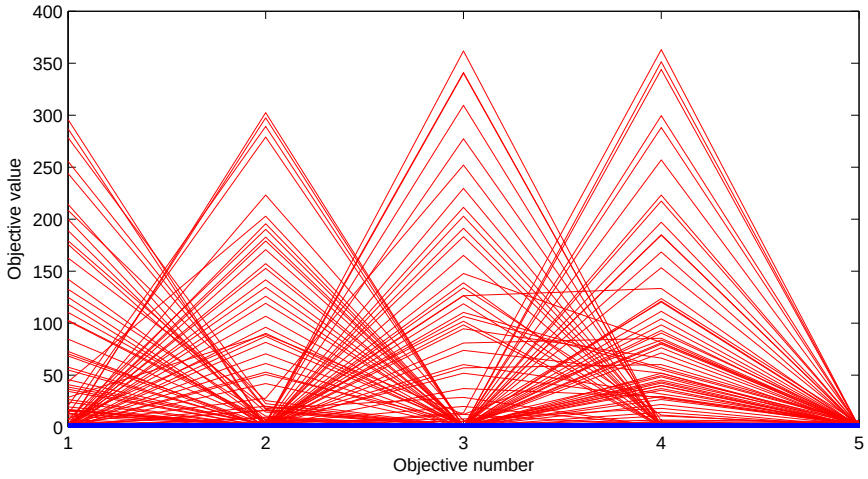
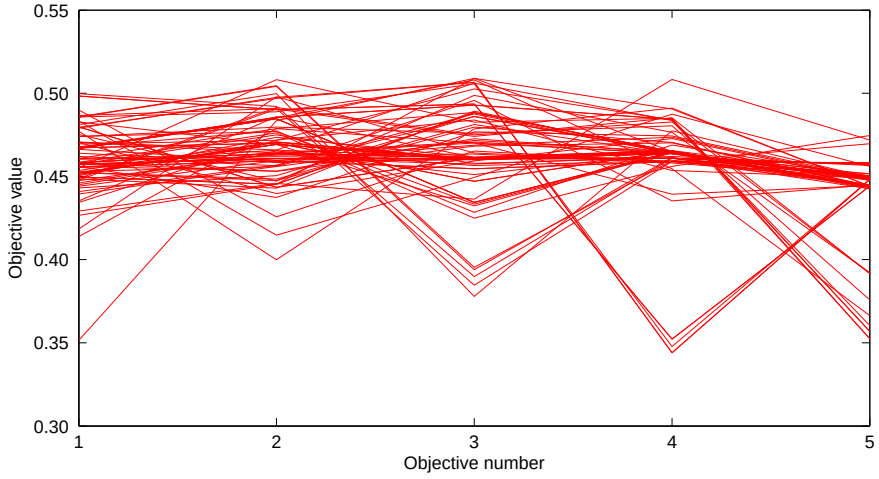


Figure 3.17: Solution points of a five-objective DTLZ1 problem instance obtained by d–EMO and NSGA-II illustrated in one parallel axis plot. The solutions in blue are from reference point d–EMO.

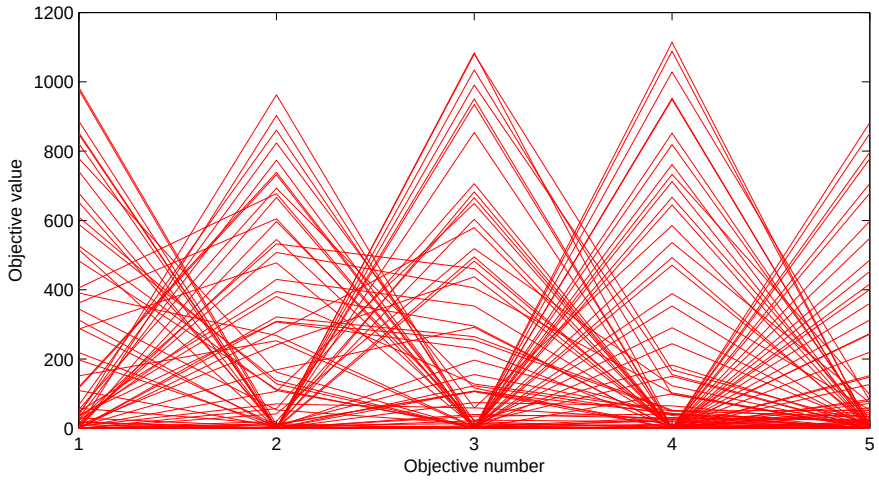
it was $[5.7942, 406.1182]$. This shows that d–EMO converged very close to the Pareto front, because on DTLZ1 a solution $\vec{x}$ is on the Pareto front if and only if $\sum_{i=1}^{M} f_i(\vec{x}) = 0.5$ (see Appendix A.2 – page 201).

Figure 3.18 illustrates the solutions obtained for the multi-modal DTLZ3 problem instance having 14 decision variables. Here also we observe that d–EMO is able to minimise the objectives better than NSGA-II and converge onto the preferred region. We conclude that the solutions on d–EMO has converged very close to the Pareto front, because on DTLZ3 a solution $\vec{x}$ is on the Pareto front if and only if $\sum_{i=1}^{M} (f_i(\vec{x}))^2 = 1$, which is a property of a sphere. The sum of the squares of the objectives values were in the range of $[1.0458, 1.0598]$ for the solutions illustrated in Figure 3.18 for d–EMO. For NSGA-II results were in the range of $[14818.4693, 1240268.1558]$.

Figure 3.19 illustrates the results for the ten-objective DTLZ1 ($N = 14$) problem instance. Here also we observe that d–EMO is able to converge very close to the Pareto front on the preferred region as opposed to NSGA-II. For the solution points in d–EMO $\sum_{i=1}^{M} f_i(\vec{x}) \in [0.5061, 0.5184]$ and for NSGA-II $\sum_{i=1}^{M} f_i(\vec{x}) \in [59.4101, 532.0352]$.

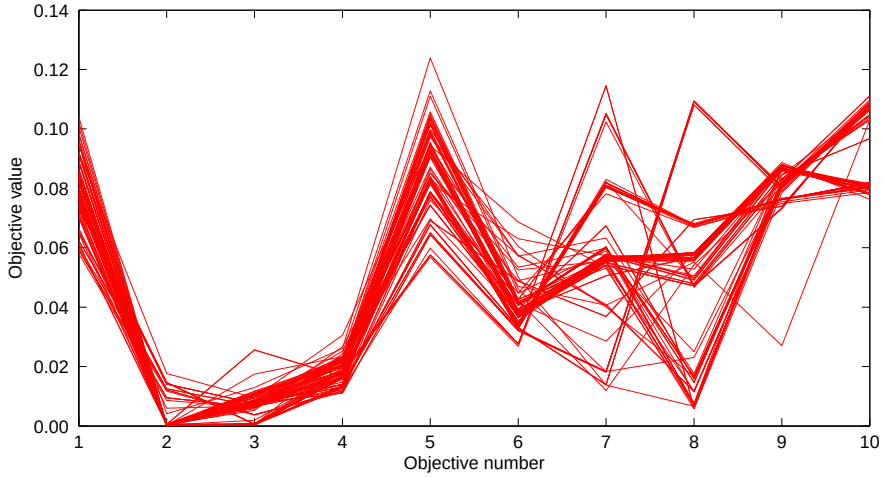


(a)

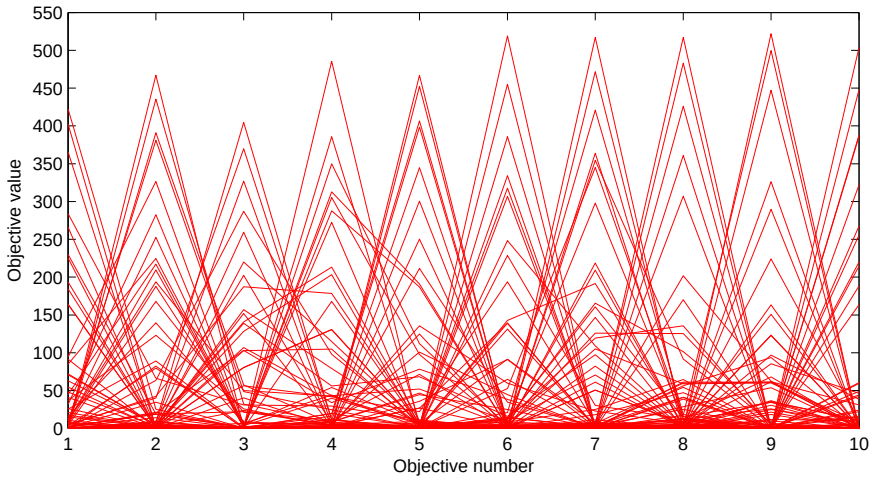


(b)

Figure 3.18: Solution points of a five-objective DTLZ3 problem instance: (a) d–EMO with one reference point; (b) NSGA-II.

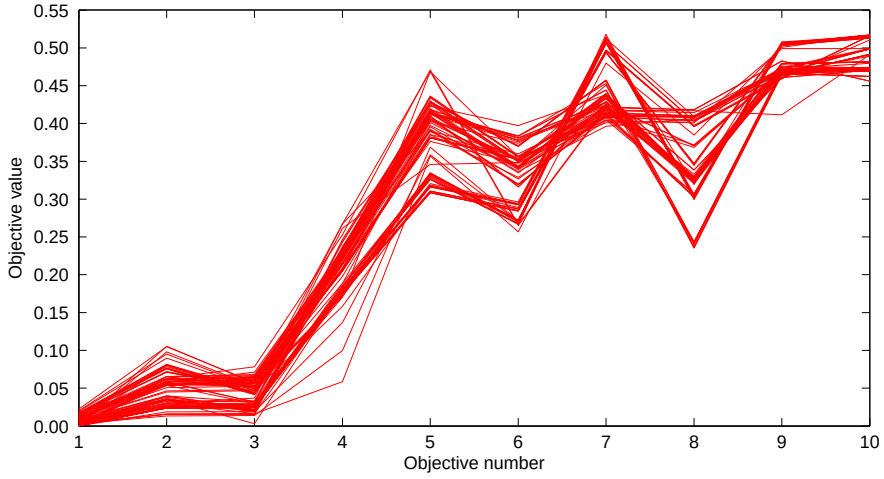


(a)

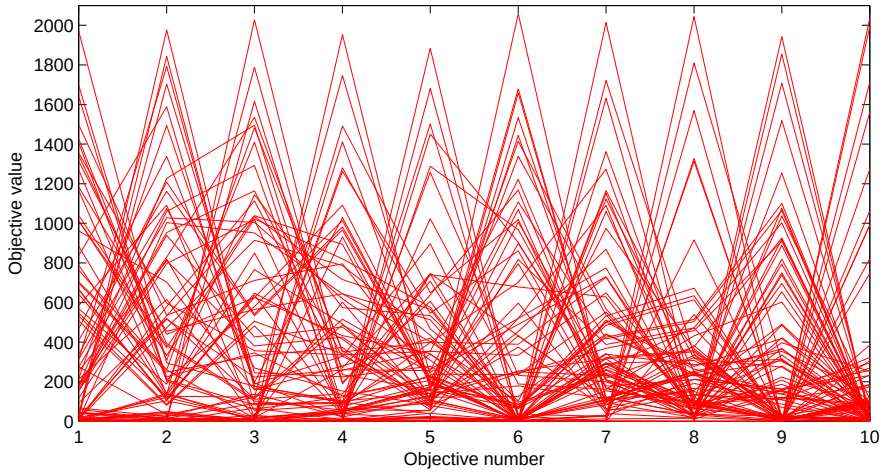


(b)

Figure 3.19: Solution points of a ten-objective DTLZ1 problem instance: (a) d–EMO with one reference point; (b) NSGA-II.



(a)



(b)

Figure 3.20: Solution points of a ten-objective DTLZ3 problem instance: (a) d–EMO with one reference point; (b) NSGA-II.

Similar to the multi-modal DTLZ1 problem, d–EMO was able to converge very close to the Pareto front of the preferred region on the ten-objective DTLZ3 ($N = 19$) problem instance. The solutions are illustrated in Figure 3.20. For the solutions in d–EMO $\sum_{i=1}^{M} (f_i(\vec{x}))^2 \in [1.0837, 1.1322]$ and for NSGA-II $\sum_{i=1}^{M} (f_i(\vec{x}))^2 \in [148910.6824, 4301601.5556]$.

These results show that using a reference point, d–EMO can be guided to obtain solutions very close to the Pareto front effectively on preferred regions rather than trying to find the entire Pareto front on many-objective problems. We have not illustrated the simpler many-objective problems here because we want to emphasise the effectiveness of d–EMO on the more difficult many-objective problems. Our experiments have shown that d–EMO is more effective than NSGA-II on all the many-objective test problems found in the DTLZ and WFG test suites. We present these results later in Chapter 4, using a performance metric. Next, we illustrate some results using the light beam search for completeness of the user-preference methods used in d–EMO.

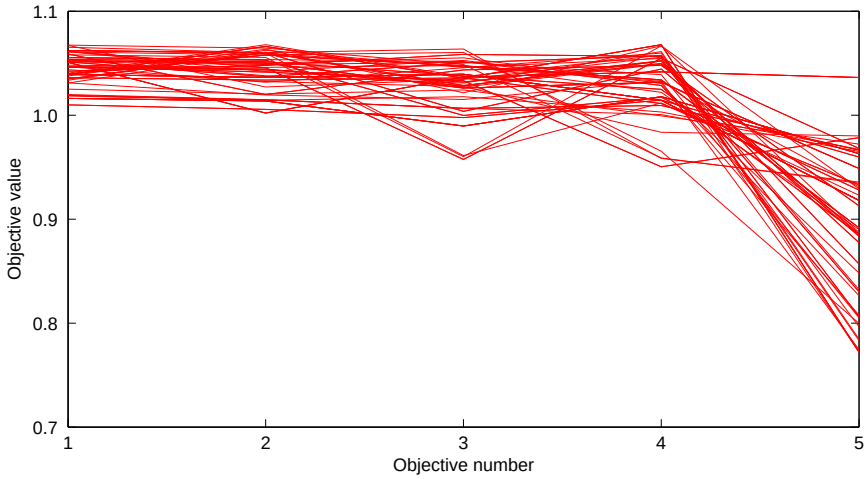
3.3.3 Using light beams

To illustrate the use of light beam search in d–EMO we used one light beam where the AP has 1.0 for all objectives and the RP has 0.0 for all objectives. We used a spread value of $\delta = 0.05$. Here we illustrate some results for the five and ten-objective problem instances of WFG1.

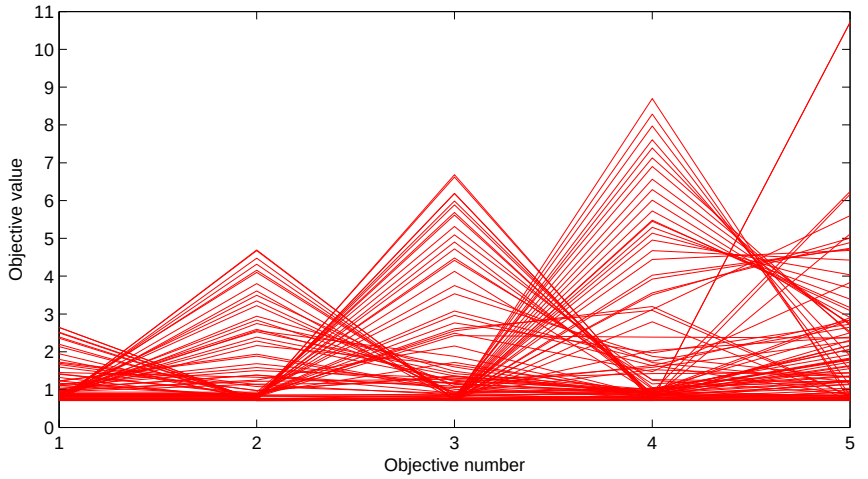
Figure 3.21 illustrates the final solutions obtained from the five-objective WFG1 problem instance. In this instance, we used twelve decision variables comprising of four position variables and eight distance variables. We observe that d–EMO is able to minimise the objectives better than NSGA-II. According to the specifications of the WFG test suite (see Appendix A.3 – page 206) we examined the decision variable to find out if the solutions containing those decision variables, are on or close to the Pareto front. Table 3.2 illustrates the ranges of the values of the distance variables and their *offset* value. The offset column indicates the maximum offset from the minimum or maximum values to the Pareto optimal value of each decision variable. For any decision variable x_i, we define the offset value as:

$$x_i^{\text{offset}} = \max\left\{\left|x_i^{\min} - x_i^*\right|, \left|x_i^{\max} - x_i^*\right|\right\} \tag{3.5}$$

Here, $x_i^{\min}$ and $x_i^{\max}$ are the minimum and maximum values respectively for the decision variables in the population and x_i^* is the value of the decision variable if the solution is Pareto optimal. The values in **bold** in the offset column show the minimum offset from the



(a)



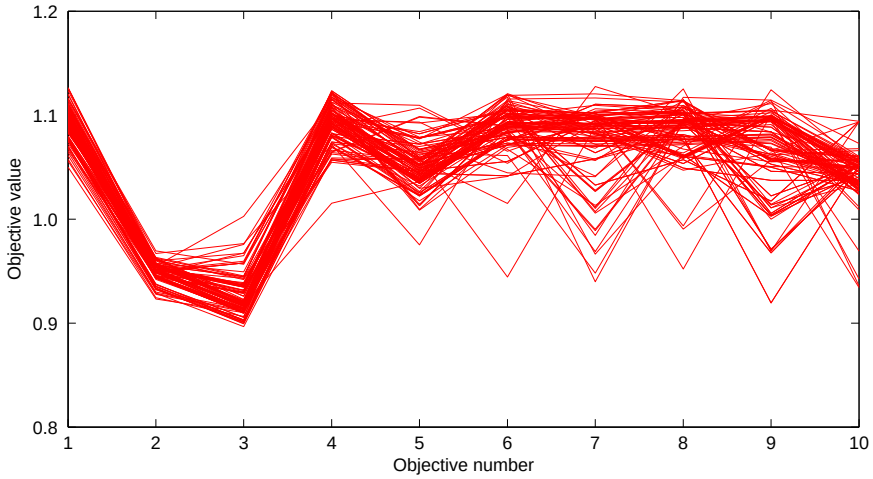
(b)

Figure 3.21: Solution points of a five-objective WFG1 problem instance: (a) d–EMO with one light beam; (b) NSGA-II.

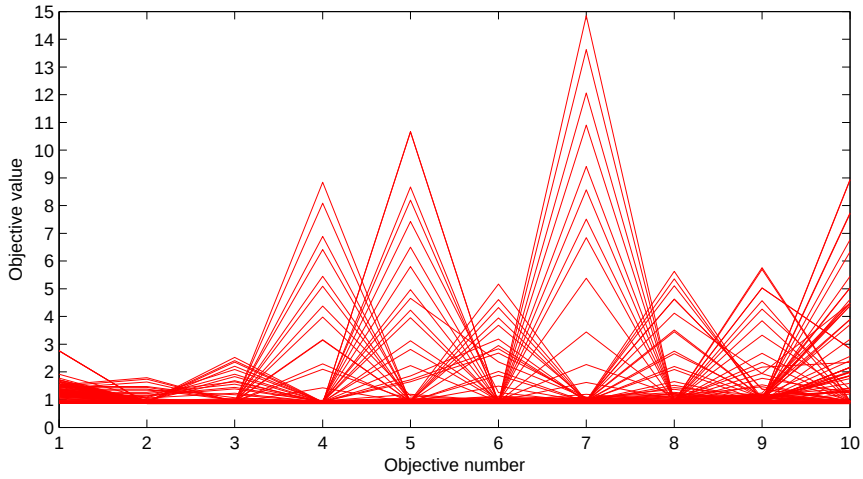
Pareto optimal value for each corresponding decision variable between the two algorithms. We consider the algorithm which has more low offset values to be closer to the Pareto front than the other. The offset values in Table 3.2 show that *d*–EMO has found the distance variables very close to the values of the ones on the Pareto front, much closer than NSGA-II. Using the observation in Table 3.2 and the results of Figure 3.21 we can conclude that *d*–EMO had better convergence on the five-objective WFG1 than NSGA-II.

Table 3.2: Values of the distance variables and their offset values, of d–EMO and NSGA-II for the five-objective WFG1 problem instance. The values in ***bold*** *are the minimum offset values between the algorithms for each decision variable.*

	***d*–EMO**			**NSGA-II**			**Pareto value**
Variable	Min. $\left(x_i^{\min}\right)$	Max. $\left(x_i^{\max}\right)$	Offset $\left(x_i^{\text{offset}}\right)$	Min. $\left(x_i^{\min}\right)$	Max. $\left(x_i^{\max}\right)$	Offset $\left(x_i^{\text{offset}}\right)$	$x_i^* = 2i \times 0.35$ for $i = 5, \ldots, 12$
x_5	3.4997	3.5000	**0.0003**	2.6003	4.9160	1.4160	3.5
x_6	4.2000	4.2000	**0.0000**	2.6125	4.4764	1.5875	4.2
x_7	4.9000	4.9000	**0.0000**	4.8241	7.5943	2.6943	4.9
x_8	5.4793	5.6106	**0.1207**	4.6112	8.0531	2.4531	5.6
x_9	6.3000	6.3000	**0.0000**	3.6261	7.3839	2.6739	6.3
x_{10}	7.0000	7.0000	**0.0000**	4.3506	9.0519	2.6494	7.0
x_{11}	7.7000	7.7000	**0.0000**	5.6686	11.1617	3.4617	7.7
x_{12}	8.4000	8.4000	**0.0000**	6.9208	10.5426	2.1426	8.4



(a)



(b)

Figure 3.22: Solution points of a ten-objective WFG1 problem instance: (a) d–EMO with one light beam; (b) NSGA-II.

Figure 3.22 illustrates the final solutions obtained from *d*–EMO and NSGA-II on the ten-objective WFG1 problem. Here we used 27 decision variables having 9 position variables and 18 distance variables. The results in Figure 3.22 and Table 3.3 show that *d*–EMO found solutions closer to the Pareto front than NSGA-II in the ten-objective WFG1 problem instance. This shows that *d*–EMO using light beam search is also effective as the reference point method in guiding an EMO algorithm to find solutions in many-objective problems.

Table 3.3: Values of the distance variables and their offset values, of d–EMO and NSGA-II for the ten-objective WFG1 problem instance. The values in ***bold*** *are the minimum offset values between the algorithms for each decision variable.*

	***d*–EMO**			**NSGA-II**			**Pareto value**
Variable	Min. $\left(x_i^{\min}\right)$	Max. $\left(x_i^{\max}\right)$	Offset $\left(x_i^{\text{offset}}\right)$	Min. $\left(x_i^{\min}\right)$	Max. $\left(x_i^{\max}\right)$	Offset $\left(x_i^{\text{offset}}\right)$	$2i \times 0.35$ for $i = 10, \ldots, 27$
x_{10}	6.6746	7.2044	**0.3254**	6.6064	11.3753	4.3753	7.0
x_{11}	7.6749	7.7481	**0.0481**	7.4402	17.2907	9.5907	7.7
x_{12}	7.8659	8.1035	**0.5341**	7.0044	9.8337	1.4337	8.4
x_{13}	8.4275	9.1854	**0.6725**	0.4062	25.6923	16.5923	9.1
x_{14}	9.8206	10.0853	**0.2853**	0.2721	8.3093	9.5279	9.8
x_{15}	9.2149	10.5200	**1.2851**	7.3107	15.3685	4.8685	10.5
x_{16}	10.0296	11.2219	**1.1704**	9.2179	13.0066	1.9821	11.2
x_{17}	11.9853	13.2936	**1.3936**	10.1808	17.2627	5.3627	11.9
x_{18}	12.5473	13.8809	**1.2809**	9.3787	19.8562	7.2562	12.6
x_{19}	12.7985	14.0204	**0.7204**	5.8404	19.3245	7.4596	13.3
x_{20}	13.3840	14.0304	**0.6160**	10.9051	15.7161	3.0949	14.0
x_{21}	13.8933	14.7900	**0.8067**	10.8580	19.8428	5.1428	14.7
x_{22}	14.6479	15.5372	**0.7521**	7.9213	18.1551	7.4787	15.4
x_{23}	15.0963	16.9654	**1.0037**	10.8557	24.2741	8.1741	16.1
x_{24}	16.7895	19.1812	**2.3812**	9.4951	23.3103	7.3049	16.8
x_{25}	17.3437	18.2646	**0.7646**	9.5629	30.5167	13.0167	17.5
x_{26}	18.1913	18.7312	**0.5312**	16.1949	40.4054	22.2054	18.2
x_{27}	34.4213	46.9661	28.0661	12.0451	29.9022	**11.0022**	18.9

We have shown that d–EMO is more effective in locating solutions on or very close to the Pareto fronts of many-objective problems than NSGA-II. These results show that the distance metric can be used very effectively to aid EMO algorithms to find solutions in many-objective problems.

It is important to note that regardless of using or not using a preference mechanism does not change the percentage of non-dominated solutions in the final generation. To be precise, all the solutions will be non-dominated to each other if the dominance was derived using the standard dominance relation. This is the main reason why we use a distance metric to differentiate the solution points such that they can provide the necessary search pressure and progress in locating solutions on the Pareto front.

We have drawn our conclusions about this distance metric based approach by visualising the solutions using Cartesian or parallel axis plots and examining the properties of the objective values and decision variables. However, for real-world problems where the Pareto fronts are not known, we need to have a better metric to compare and analyse the effectiveness of the algorithms that are used to solve them. There are some metrics found in the EMO literature [Zitzler et al., 2000] that can be used for analysing EMO algorithms (see Section 2.7 – page 45). However, these standard metrics are not suitable for comparing user-preference EMO algorithms.

3.4 Summary

This chapter aimed in answering the research question: *How can we integrate user-preference methods from the MCDM literature to design efficient EMO algorithms that can find solutions for many-objective problems?* To this end, we introduced a distance metric that was derived from the reference point method and the light beam search. Next, we presented how the distance metric can be used to guide an EMO algorithm to find solutions in many-objective problems. The population of the EMO algorithm can be used to define a range of solutions, which spread across a solution front. We have proposed a method to define this spread using the notion of outranking. Using the spread of solutions and the search direction we defined the term preferred region. A DM will indicate such preferred regions in the objective-space of a many-objective problem and the EMO algorithm will concentrate only on those regions to locate solutions. This is a more efficient approach than using a non-dominated sorting approach to find all the solutions on the Pareto front of a many-objective problem. We have

presented d–EMO, which is an implementation of an EMO algorithm using the distance metric framework. This algorithm uses the same GA operators as NSGA-II except that it does not use any non-dominated sorting procedure, which makes it more computationally efficient than NSGA-II.

To determine the effectiveness of d–EMO we executed it on various test problems from the ZDT, DTLZ and WFG test problem suites. We presented the results of some of the more difficult problems from those test suites, which have not been used in other preference based EMO algorithms found in the literature. We have deduced that d–EMO is able to handle not only the simple problems but also the more difficult many-objective problems effectively.

We have observed that the methods used to analyse the user-preference EMO algorithms were not suitable where the Pareto fronts are unknown. This is a concern when these algorithms are used in real-world problems that have many-objectives. In the next chapter, we address this issue by introducing a better performance metric for evaluating user-preference EMO algorithms.

Chapter 4

Designing a suitable performance metric

So far we have shown results of our proposed algorithms merely based on visual comparisons of the obtained solutions in Chapter 3. However, for a proper comparison between several EMO algorithms one should make use of performance metrics. Although there are several existing performance metrics for EMO algorithms, they are not suitable for user-preference algorithms in their current form. Therefore, we propose a suitable performance metric in this chapter to better compare user-preference algorithms.

In this chapter we first present some background information on the limitations of existing EMO performance metrics and why they are not suitable for user-preference EMO algorithms in Section 4.1. In Section 4.2, we discuss in detail the usefulness of the hyper-volume metric in EMO algorithms and why it can be modified for comparing user-preference EMO algorithms.

In Section 4.3 we define formally a modified hyper-volume metric that is more appropriate for comparing user-preference EMO algorithms. Using the DTLZ and WFG many-objective test suites we show how the newly proposed hyper-volume metric can be used to compare the performance d–EMO and NSGA-II in many-objective problems. We also provide a comparison of performance between d–EMO and two other reference point EMO algorithms found in the literature on various many-objective test problems. These results show that the distance metric based method is more effective than the standard non-dominated sorting methods on many-objective problems.

4.1 The need for a better performance metric

As seen in the previous chapter, visually assessing the performance of an EMO algorithm is rather limiting. Therefore we require a metric to better assess the performance of the algorithm. We presented some existing EMO performance metrics in Section 2.7 (page 45).

All these popular performance metrics that measure the relative differences between given solution fronts can only be used to compare EMO algorithms on problem instances of known Pareto fronts like the ZDT, DTLZ and WFG test suites. This is a major shortcoming when EMO algorithms are compared on problem instances where the Pareto fronts are unknown. These metrics also measure the convergence and the diversity of an EMO algorithm against the entire Pareto front.

These metrics are not suitable for user-preference EMO algorithms because these algorithms do not intend to find the entire solution front. These metrics are also not suitable for problem instances where the true Pareto front is unknown, which is often the case in real-world problems (see Chapters 6 and 7). Therefore, we need to come up with a suitable metric to compare the performance of user-preference EMO algorithms more meaningfully. Designing a suitable metric for user-preference EMO algorithms is an important part of this research.

4.2 The Hyper-Volume metric

One metric found in the EMO literature that does not depend on a known Pareto front is the Hyper-Volume (HV) metric. This idea was first described by Zitzler and Thiele [1998] as the S metric. In Section 4.3 we describe a mechanism to use this HV metric in a manner suitable for user-preference EMO algorithms. However, first, let us illustrate the standard HV metric.

4.2.1 Calculating HV values

The HV metric gives the total volume bounded by the points on a solution front and a selected point in the objective-space. One of the issues with this initial approach was that the volume bounded was dependent on the selected point. Using a point that is dominated by all the points in the solution front can eliminate this shortcoming [Knowles, 2005]. This *nadir point* is dominated by all other solution points. At the nadir point, all objectives are at their worst values simultaneously [Messac and Mattson, 2004].

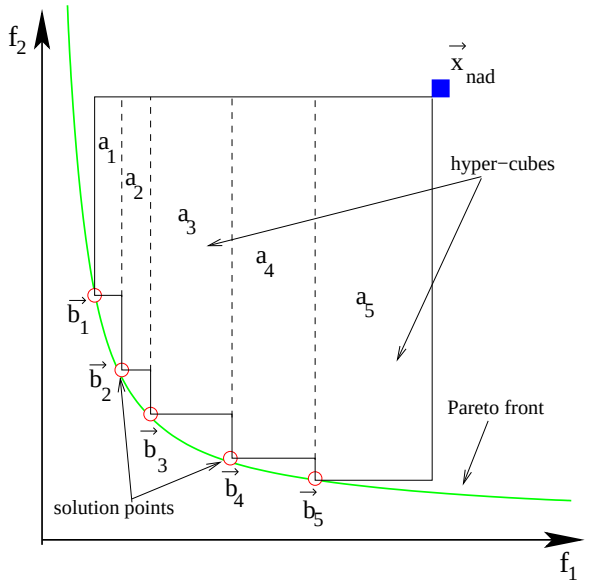


Figure 4.1: The HV calculation process on a two-objective problem instance.

The nadir point $\vec{x}_{nad}$ is derived by:

$$\vec{x}_{nad} = \left[f_1^{nad}(\vec{x}), \ldots, f_M^{nad}(\vec{x}) \right], \text{ where } f_i^{nad}(\vec{x}) = \max_{\vec{x} \in B} \{ f_i(\vec{x}) \} \tag{4.1}$$

Here, B is the solution front in consideration. This solution front can be at any generation of the EMO algorithm. The HV is the area bounded by all the hyper-cubes between the nadir point and the points in the solution front. Mathematically, the HV is the Lebesgue measure (Λ) of the union of all a_i defined by a solution point $\vec{b_i} \in B$ and the nadir point $\vec{x}_{nad}$ [Emmerich et al., 2005]. The HV calculation process is:

$$hv(B) = \Lambda \left(\left\{ \bigcup_i a_i | \vec{b_i} \in B \right\} \right) = \Lambda \left(\bigcup_{\vec{b_i} \in B} \left\{ \vec{x} | \vec{b_i} \prec \vec{x} \prec \vec{x}_{nad} \right\} \right) \tag{4.2}$$

The most important step in the HV calculation is to use a common set of solutions — for the set B — when deriving a nadir point. This ensures that all the algorithms used in a comparison will have their HV values calculated against the same nadir point. In our experiments we achieve this by using all the solutions from the final generations of all the algorithms used for comparison to create the set B.

Figure 4.1 illustrates the calculation of the HV metric in a two-objective problem instance. The example solution front contains five solutions, which are given by $\vec{b}_1$, $\vec{b}_2$, $\vec{b}_3$, $\vec{b}_4$ and $\vec{b}_5$.

The hyper-cubes bounded by each solution point and the nadir point $\vec{x}_{nad}$ are a_1, a_2, a_3, a_4 and a_5. Therefore, the HV value is the sum of total area in all the volumes, which is given as $\Lambda\left(a_1 \cup a_2 \cup a_3 \cup a_4 \cup a_5\right)$. The HV metric is very useful in comparing different EMO algorithms because it provides one numerical value for measuring both convergence and spread of the corresponding solution fronts.

4.2.2 Using the HV metric in EMO algorithms

To mitigate random chance in the stochastic process of EMO algorithms, we execute an EMO algorithm on the same problem instances for many runs — 30 or more runs — and average the results. The performance metric of an EMO algorithm for a given problem instance is the average performance from all of the runs. Therefore, we present the average HV value for an EMO algorithm from multiple runs for any given problem instance. We use the following steps as indicated by Zitzler and Thiele [1999] to define the general steps of calculating the HV values for an EMO algorithm on a given problem instance:

1. For a given problem instance, combine the all solution points of the final generation of each run to obtain a grand population of solutions. This grand population consists of all the solution points in the final generation of all the algorithms in consideration. We use only the final generations' solution points to reduce the number of solution points in the grand population because the computational complexity of the HV calculation process increases exponentially with the number of solution points in consideration and the number of objectives [While et al., 2006]. Therefore, we only use the HV calculation after the optimisation has completed, not for each generation within the execution run.

2. Obtain the extreme solution points from the grand population. These points will be used to normalise the objective values of the solution points.

3. Compute the nadir point (using Equation 4.1) from this grand population. This constant nadir point is used to calculate the HV values.

4. Obtain the HV values for any generation of each run and present the average HV value. The HV values will be in the range of $[0, 1]$ because the solution points have been normalised.

Step 1 will include the solution points from the final generation of all the runs of each EMO algorithm, if we are comparing the performance of two or more EMO algorithms for a

given problem instance. This grand population will be used to compute the nadir point and also normalise the objective values. When comparing two EMO algorithms, the one that gives a larger HV value is considered to have a better convergence and spread of solutions. The volume calculated by this metric gives a single numeric measure on both the spread and the closeness of the solutions to the Pareto front. Theoretical results by Fleischer [2003] show that when a solution set dominates another solution set, the HV metric yields larger values for the dominant set of solutions than the dominated set of solutions. It is important to realise that the normalised HV values are relative metrics rather than absolute values when used to compare EMO algorithms.

This process of comparing EMO algorithms using a normalised HV metric is advantageous than other metrics, because it does not depend on the Pareto front. We have found that so far there are no universally agreed performance measurements for comparing user-preference EMO algorithms in the literature. This was a major motivation for us to explore the feasibility of revising the HV metric as a performance metric for user-preference EMO algorithms. More specifically, the standard HV calculation procedure has to be modified to include only the solution points within the preferred regions, not the entire Pareto front.

4.3 Redesigning the HV metric for user-preference EMO algorithms

The goal of user-preference EMO algorithms is to locate solutions in preferred regions. Therefore, the HV calculation should incorporate this information to make it suitable for comparing such algorithms. To achieve this, we exclude the solution points outside the preferred regions so that the nadir point used for the HV calculation lies within the preferred region. This process will favour the HV calculation towards algorithms that converge onto the preferred regions. We extend the standard HV metric in the following key steps for one preferred region.

Step 1: Obtain the solution point closest to the ideal point

First, for a given problem instance, all the final solution points from every run of each algorithm are combined to make a grand population. Next, the Euclidean distance is calculated for each solution point in this population from the ideal point, as seen in Figure 4.2. The solution point with the lowest Euclidean distance is selected to define a volume.

We use the closest solution point to the ideal point to emphasise convergence. If we selected the solution point closest to the reference point or AP, and if they were in the

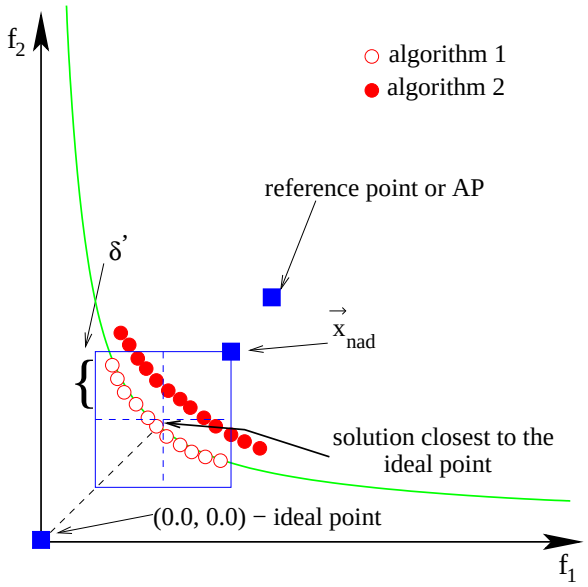


Figure 4.2: A volume is defined around the solution that is closest to the ideal point, so that solutions outside of this volume are excluded from the HV calculation process.

feasible region, the HV calculation will favour an EMO algorithm converging close (or on) to the reference point or AP. This EMO algorithm could give better HV values than an EMO algorithm that was actually able to get close to the Pareto front, because solutions of the better algorithm could be excluded from the HV calculation process. If the reference point or AP were in the infeasible region, such an issue would not arise. However, one cannot make a judgement if the reference point or AP will be feasible or infeasible, a priori, for any given many-objective problem, especially when the Pareto front is unknown. Therefore, to accommodate all these possible conditions we use the ideal point, which will favour the HV calculation on an EMO algorithm that has the better convergence.

Step 2: Define a volume for HV calculation

A volume is defined around the solution point with the lowest Euclidean distance to the ideal point. As seen in Figure 4.2, for a two-objective problem, where the objectives are to be minimised and all the objective values are always positive, the ideal point is $(0.0, 0.0)$. For a different problem instance this ideal point may change depending on the values for the objectives. It is important to note that the DTLZ and WFG test problems we used all had positive objective values that had to be minimised.

The DM can provide value δ' such that a volume is defined having $2\delta'$ for each objective. The DM can adjust this δ' value such that a sufficient number of solution points lie within the defined volume. It is not necessary to define a volume with equal length for all objectives. We used an equal length for simplicity since all objectives are normalised.

Step 3: Filter solutions points and calculate the HV

Solution points from the final set of solutions obtained by all algorithms and runs in consideration are removed if they reside outside this common volume. The remaining solution points are used to obtain the nadir point, which will be located within the defined volume. The calculation of the HV values for each run of each algorithm is done using this common nadir point and solution points within the volume. This process emphasises on the solution points within the preferred region.

If the EMO algorithms have used multiple preferred regions, this process is applied separately to calculate the HV values for each region. We use this modified HV calculation process to compare the performance of d–EMO with NSGA-II and several other user-preference EMO algorithms which use non-dominated sorting. To emphasise the solutions on the preferred regions, only the final solution points of user-preference EMO algorithms are used to obtain the grand population as described in Step 1.

4.3.1 Visualising convergence using the HV metric

In Section 3.3.1 (page 75) we presented the final solution points obtained by d–EMO and the original NSGA-II algorithm in some many-objective problems. The parallel axis plots showed the convergence within a region of a solution front and we used the problem's characteristics to infer that the solutions were close to the Pareto front. The HV metric for user-preference EMO algorithms as described in Section 4.3 can also be used to observe the convergence of the

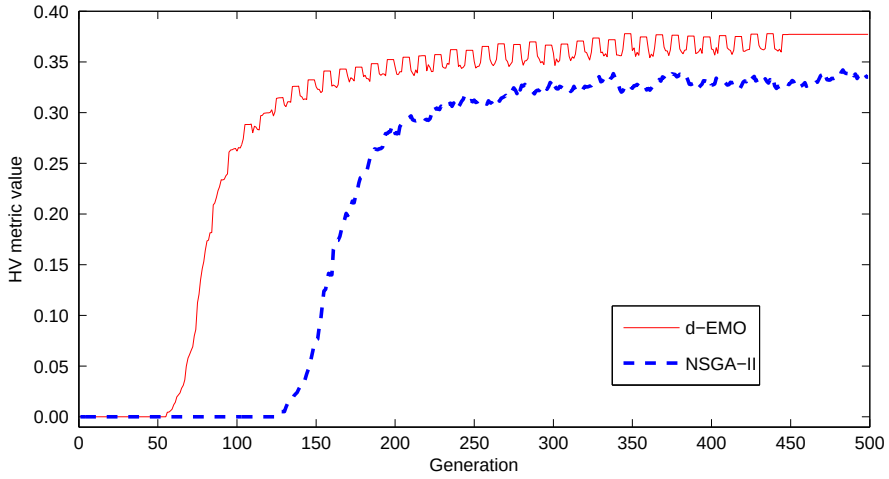


Figure 4.3: HV values across the generations of d–EMO and NSGA-II on the three-objective DTLZ1 problem.

algorithms during the execution process. Here, we calculate the HV metric at each generation using the nadir point calculated from the final solutions to visualise the convergence towards to preferred regions.

Using the parameter settings described in Table 3.1 (page 60), we executed d–EMO and NSGA-II for 30 independent runs on some many-objective problems and observed the HV values at each generation. For d–EMO, we used one preferred region with $\delta = 0.05$ and a reference point having 0.5 for all objectives. The nadir point was calculated from the final solution fronts of all the runs combined for a given problem instance. This process emphasised the convergence and spread onto the preferred region. As described in Section 4.3, a volume was defined using various δ' values such that at least 90% of the solution points of the algorithm closest to the ideal point were within the defined volume. All other solution points were excluded from HV calculation if they were not in the defined volume.

Figures 4.3 and 4.4 illustrate the average HV values for d–EMO and NSGA-II from the 30 runs on the three-objective DTLZ1 and DTLZ2 problems respectively. We used $\delta' = 0.1$ for both instances to define the volume for HV value calculation. On average NSGA-II performs well on the three-objective DTLZ1 and DTLZ2 problems, converging onto the Pareto fronts. However, since we are only interested in the solution points within the defined volume, we observe that NSGA-II gives lower HV values than d–EMO. This is because more individuals were found in the preferred region by using d–EMO than by NSGA-II from the entire population of 200 individuals. In other words, NSGA-II distributed solutions over

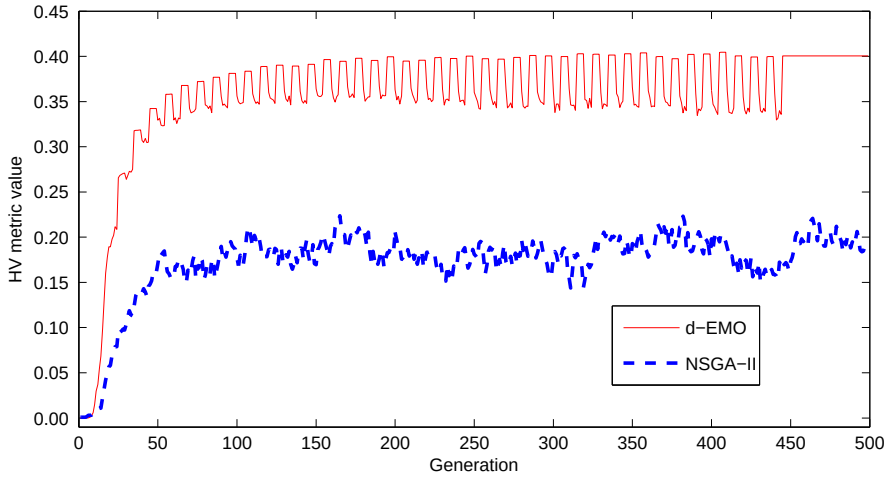


Figure 4.4: HV values across the generations of d–EMO and NSGA-II on the three-objective DTLZ2 problem.

the entire Pareto front, while d–EMO converges to a selected region of the Pareto front. From Figures 4.3 and 4.4, we also observe the exploration and exploitation steps during the execution of d–EMO (see Section 3.1.3 – page 56) when abrupt changes of the HV values occur. Usually a drop in the HV values indicates the exploration process, where more random individuals move into the next generation. These random individuals could lie outside the defined volume, reducing the HV values. It is also observed that the exploitation process occurs towards the end of the execution. This process fine tunes the convergence onto the preferred region. Although d–EMO and NSGA-II were able to converge onto the Pareto fronts of DTLZ1 and DTLZ2, we observe that on average d–EMO reached the Pareto front in fewer numbers of generations (less than 100) than NSGA-II.

Next, we present the results on problem instances having more than three objectives. Figures 4.5 and 4.6 illustrate the average HV values through the execution process of d–EMO and NSGA-II on five-objective DTLZ1 and DTLZ2 problem instances. Here also we used $\delta' = 0.1$ to define the volume. We observe that the reference point based algorithm converges closer to the Pareto fronts than the non-dominated sorting based approach. We deduced this result using mathematical properties of the problem instances as illustrated in Appendix A.2 (page 201).

The HV values clearly show that there are more solutions in the preferred region in d–EMO than for NSGA-II. In the many-objective problem instances, NSGA-II had HV values close to 0.0 which indicates that very few (almost none) of the solutions were within the

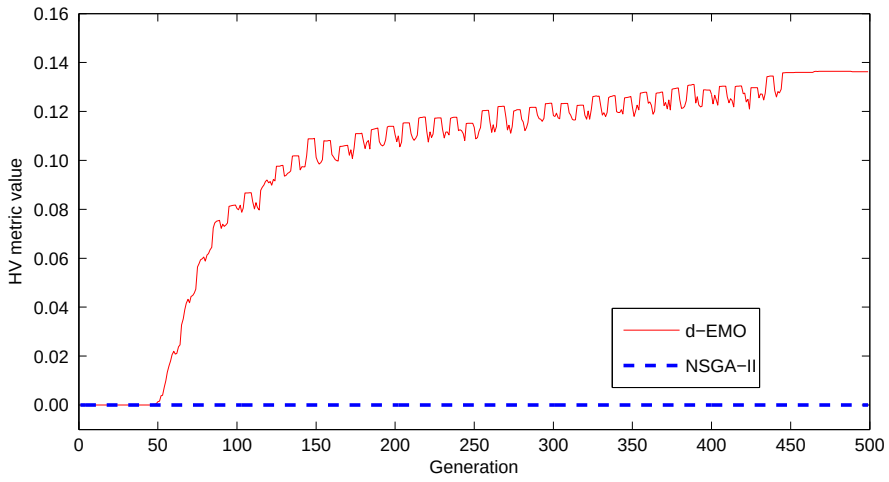


Figure 4.5: HV values across the generations of d–EMO and NSGA-II on the five-objective DTLZ1 problem.

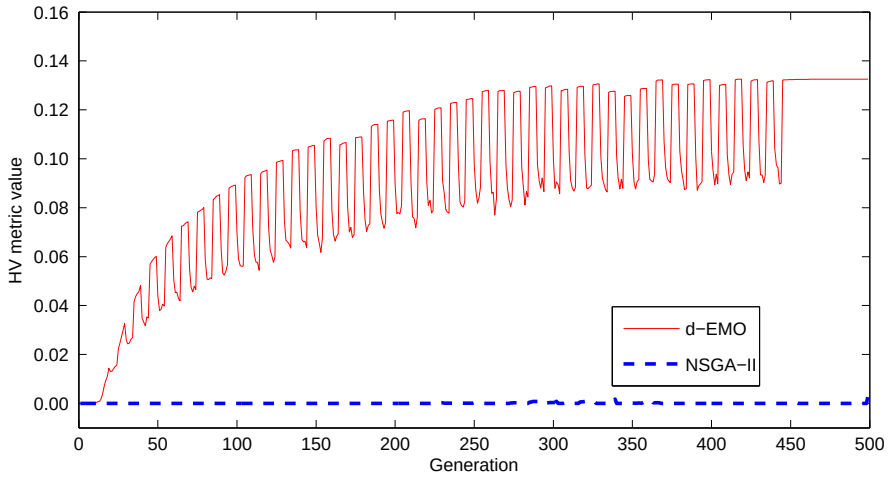


Figure 4.6: HV values across the generations of d–EMO and NSGA-II on the five-objective DTLZ2 problem.

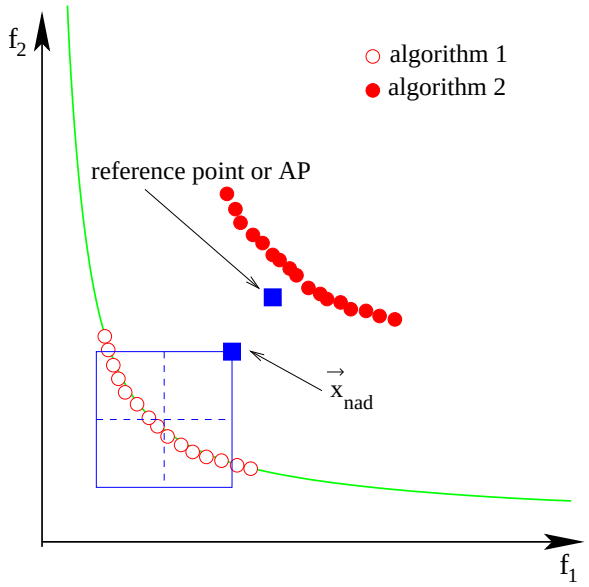


Figure 4.7: A hypothetical scenario on a two-objective problem instance where one algorithm does not have any solutions within the defined volume. In such a case algorithm 2 will have a 0.0 *value for the HV metric.*

defined volume in the preferred regions. Figure 4.7 illustrates this scenario on a hypothetical two-objective problem instance. Here, algorithm 1 has solution points closer to the ideal point than for algorithm 2. Conversely this also illustrates that algorithm 1 converges better than algorithm 2. Therefore, when the volume for the HV calculation is defined all of the solution points in algorithm 2 will be excluded because they are very away far from the ideal point. Thus algorithm 2 will yield 0.0 for the HV metric.

We used $\delta' = 0.1$ and 0.3 to define volumes for the ten-objective DTLZ1 and DTLZ2 problem instances respectively. Figures 4.8 and 4.9 illustrate the average HV values through the execution of d–EMO and NSGA-II on those ten-objective problems. At ten-objectives, we observe that non-dominated sorting is not effective even on a problem instance which is biased like DTLZ2. Also NSGA-II was hardly effective in a multi-modal problem like the ten-objective DTLZ1. However, d–EMO was able to effectively converge onto a solution front close to the Pareto front.

These HV based results complement the solution plots in Section 3.3 (page 75) to show that a user-preference EMO algorithm performs better than a standard non-dominated sorting based EMO algorithm in many-objective problems. However, as described in Section 2.6.4

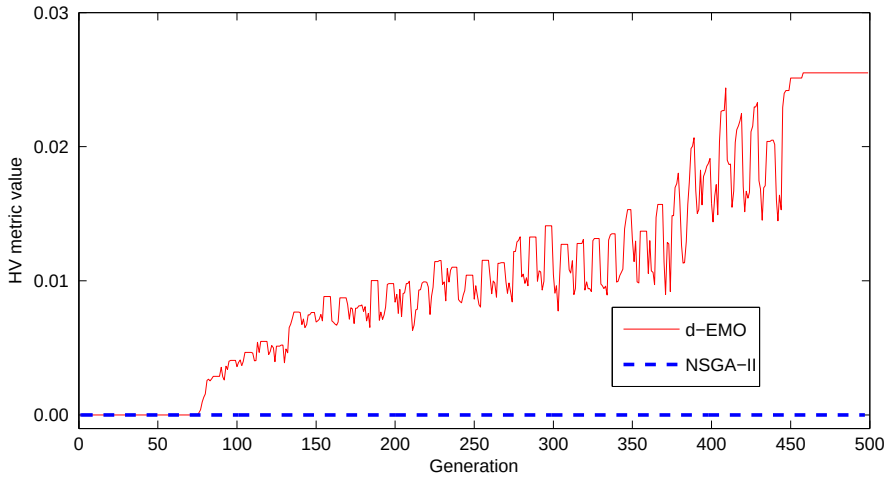


Figure 4.8: HV values across the generations of d–EMO and NSGA-II on the ten-objective DTLZ1 problem.

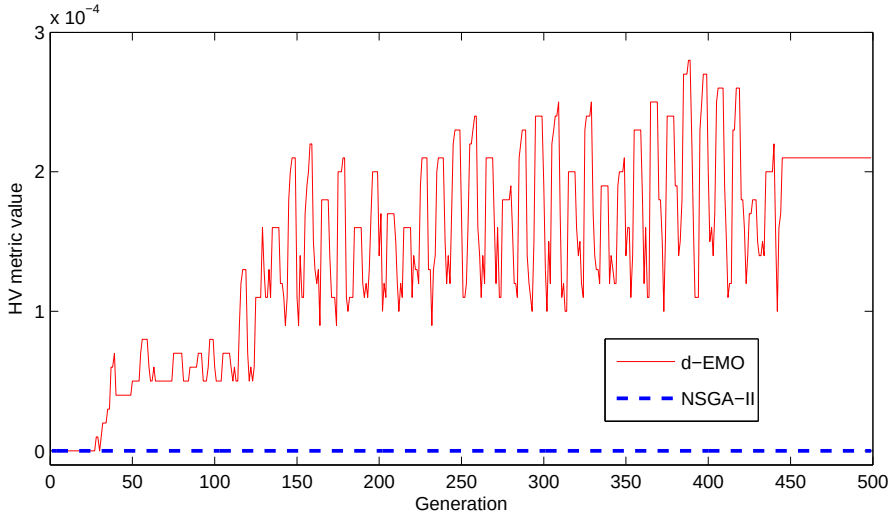


Figure 4.9: HV values across the generations of d–EMO and NSGA-II on the ten-objective DTLZ2 problem.

(page 43) there are several user-preference EMO algorithms based on non-dominated sorting in the literature. The results presented in these studies were on some simple many-objective problems, basically indicating the feasibility of incorporating user-preferences into EMO algorithms. This was a motivation for us to conduct a comprehensive comparison between d–EMO and the other non-dominated sorting based user-preference EMO algorithms on various many-objective problems to identify if a distance metric approach was more efficient and effective than a non-dominated sorting approach.

4.3.2 Comparison of d–EMO and other user-preference EMO algorithms

To compare the performance of user-preference EMO algorithms using the distance metric and non-dominated sorting, we executed several algorithms on all the test problems of the DTLZ and WFG test suites. Here we used the three, five and ten-objective problem instances of each suite. d–EMO was compared with the reference point based NSGA-II algorithm [Deb et al., 2006] and our reference point based NSPSO algorithm [Wickramasinghe and Li, 2008]. NSGA-II was modified to incorporate the reference point as described by Deb et al. [2006]. However, we incorporated the δ value with the notion of outranking, instead of $\epsilon-dominance$ concept, to the normalised Euclidean distance to define the spread of solutions in the preferred region. This ensured an equal spread of solutions in the preferred region across all the EMO algorithms in consideration. Having an equal spread of solutions between all the EMO algorithms is important to interpret the HV values as a fair assessment for convergence. As we described in Section 3.1.2 (page 54), the spread value δ is an integral component to define preferred regions. Having different methods to define the spread would not ideally represent the *same preferred region*, even though they all have the same reference point. Results were obtained by averaging over 50 runs on each algorithm on each problem instance. A population of 200 individuals is used for a maximum of 500 iterations. The standard configuration of NSGA-II as described in Table 3.1 (page 60) was used for the reference point integration. The standard values for the PSO parameters as described by Li [2003] were used for the reference point based NSPSO algorithm.

We have used a reference point which was created by having 0.5 for all objectives in each many-objective problem instance, and a spread value of $\delta = 0.05$. The results in Tables 4.1, 4.2 and 4.3 show the average and standard deviation of the HV values for the last generation of each algorithm on each problem instance. We have also indicated the p–values of the Student's t–test with a 95% confidence interval. This paired two-tail t–test provides the

statistical significance for the mean HV values when comparing different algorithms. The t–test, tests the null hypothesis as:

$$H_0 : \mu_{algorithm1} \neq \mu_{algorithm2} \tag{4.3}$$

Here, $\mu_{algorithm1}$ and $\mu_{algorithm2}$ represent the means of the HV values of the final solution fronts of the two algorithms in consideration. We reject H_0 if the p–value of the t–test is greater than 0.05. In other words, this means that comparing the mean HV values of the two algorithms in consideration does not conclude that either one of them has better convergence and spread of solutions than the other. This is because the true mean values lie in overlapping regions between the observed means and standard deviations. This makes it unclear which true mean value is greater.

We tested the average HV values of d–EMO against the reference point based NSGA-II algorithm and d–EMO against the reference point based NSPSO algorithm. We have used various δ' values in the problem instances to obtain at least 90% of the individuals from the final solution sets from the EMO algorithm closest to the ideal point. This process favours the HV calculation process towards the EMO algorithm that has the best convergence in the preferred region.

Three-objective problems

The results for the HV values of the three-objective problems using one reference point are given in Table 4.1. We used $\delta' = 0.15$ for the DTLZ problems and $\delta' = 1.75$ for the WFG problems to define the volume. The values in **bold** are the best values found for each problem instance. The values in *italics* show the comparisons that are not statistically significant (p–value > 0.05). The HV values of 0.0000 indicate that the solution points were far away from the preferred region, that they were outside the volume and thus excluded from the HV calculation process. Such a scenario occurs when an EMO algorithm has converged to a local optimal front far from the preferred region. The results in Table 4.1 show that on average d–EMO is able to converge better on the preferred regions than the non-dominated sorting based algorithms for various three-objective problems.

Five-objective problems

Table 4.2 illustrates the results for the five-objective problems using one reference point. Here, for the DTLZ problems we used $\delta' = 0.5$ and for the WFG problems we used $\delta' = 1.75$

Table 4.1: The average HV values for reference point based d–EMO, NSGA-II and NSPSO on three-objective test problems using one reference point. The values in ***bold*** *are the best values found for each problem instance and the values in italics show the comparisons which are not statistically significant.*

Problem	*d*–EMO	NSGA-II	NSPSO	*p*-value for *d*–EMO against NSGAII	*p*-value for *d*–EMO against NSPSO
DTLZ1	**0.4361** ± **0.0154**	0.0623 ± 0.0580	0.2428 ± 0.0707	2.6088E-49	8.3497E-36
DTLZ2	**0.2849** ± **0.0101**	0.2715 ± 0.0181	0.1501 ± 0.0162	2.9593E-05	7.5760E-46
DTLZ3	**0.2733** ± **0.0180**	0.1320 ± 0.0954	0.0000 ± 0.0000	1.6833E-15	7.5508E-60
DTLZ4	**0.4135** ± **0.1028**	0.0676 ± 0.0147	0.2471 ± 0.2194	2.0521E-28	4.4814E-11
DTLZ5	**0.2618** ± **0.0147**	0.2386 ± 0.0021	0.1766 ± 0.0302	2.8460E-22	6.4548E-15
DTLZ6	**0.3123** ± **0.0349**	0.1886 ± 0.1008	0.2662 ± 0.0460	1.4297E-10	4.7285E-07
DTLZ7	**0.5719** ± **0.0841**	0.0000 ± 0.0000	0.3146 ± 0.1883	6.8234E-43	4.4542E-13
WFG1	**0.3165** ± **0.2155**	0.0000 ± 0.0000	0.0778 ± 0.0746	5.5993E-14	2.0800E-05
WFG2	**0.2673** ± **0.0063**	0.1309 ± 0.0614	0.0813 ± 0.0466	2.4120E-22	1.0916E-40
WFG3	**0.1169** ± **0.6700**	0.0189 ± 0.0030	0.0638 ± 0.0590	1.9558E-15	0.0389E+00
WFG4	***0.0720*** ± ***0.0521***	0.0349 ± 0.0024	*0.0560* ± *0.0503*	2.8587E-06	*0.3769E+00*
WFG5	**0.3621** ± **0.2635**	0.0443 ± 0.0036	0.0444 ± 0.0435	5.9413E-12	1.3079E-09
WFG6	***0.0694*** ± ***0.0100***	*0.0591* ± *0.0524*	0.0277 ± 0.0223	*0.2414E+00*	6.2176E-16
WFG7	**0.3818** ± **0.1710**	0.0135 ± 0.0025	0.0049 ± 0.0038	9.6559E-21	2.1231E-11
WFG8	*0.1753* ± *0.0827*	***0.1857*** ± ***0.1637***	0.0421 ± 0.0275	*0.7047E+00*	1.2355E-14
WFG9	***0.2248*** ± ***0.1735***	*0.0717* ± *0.0081*	0.1166 ± 0.0787	*0.4533E+00*	0.0124E+00

Table 4.2: The average HV values for reference point d–EMO, NSGA-II and NSPSO on five-objective test problems using one reference point. The values in ***bold*** *are the best values found for each problem instance and the values in italics show the comparisons which are not statistically significant.*

Problem	*d*–EMO	NSGA-II	NSPSO	*p*-value for *d*–EMO against NSGAII	*p*-value for *d*–EMO against NSPSO
DTLZ1	**0.1718 ± 0.0210**	0.0000 ± 0.0000	0.0463 ± 0.0438	9.8271E-47	1.0344E-23
DTLZ2	**0.0941 ± 0.0101**	0.0341 ± 0.0048	0.0460 ± 0.0074	1.4861E-36	1.0331E-30
DTLZ3	**0.0511 ± 0.0203**	0.0189 ± 0.0029	0.0000 ± 0.0000	0.0003E+00	4.6707E-23
DTLZ4	***0.0630 ± 0.0287***	0.0207 ± 0.0033	*0.0505 ± 0.0026*	2.7003E-06	*0.4163E+00*
DTLZ5	**0.1917 ± 0.0131**	0.0296 ± 0.0177	0.1735 ± 0.0277	8.4122E-46	9.2720E-05
DTLZ6	**0.0701 ± 0.0153**	0.0068 ± 0.0033	0.1782 ± 0.0581	2.2716E-16	9.3118E-33
DTLZ7	**0.2215 ± 0.0448**	0.0000 ± 0.0000	0.1020 ± 0.0490	2.5095E-36	4.5918E-16
WFG1	**0.1438 ± 0.1347**	0.0035 ± 0.0026	0.0020 ± 0.0017	5.9484E-11	8.4806E-11
WFG2	**0.3015 ± 0.0121**	0.1630 ± 0.0307	0.1941 ± 0.0432	3.2856E-33	1.0559E-20
WFG3	**0.0667 ± 0.0277**	0.0000 ± 0.0000	0.0290 ± 0.0277	3.0089E-22	7.9911E-13
WFG4	**0.0902 ± 0.0784**	0.0183 ± 0.0165	0.0134 ± 0.0037	7.7177E-10	1.3816E-09
WFG5	**0.0760 ± 0.0560**	0.0039 ± 0.0009	0.0006 ± 0.0006	9.7089E-13	3.8161E-12
WFG6	**0.0447 ± 0.0244**	0.0080 ± 0.0071	0.0000 ± 0.0000	6.2953E-17	1.9853E-17
WFG7	**0.0249 ± 0.0174**	0.0008 ± 0.0007	0.0000 ± 0.0000	8.2592E-15	1.8924E-15
WFG8	**0.0895 ± 0.0787**	0.0000 ± 0.0000	0.0027 ± 0.0006	1.6683E-10	6.1372E-10
WFG9	**0.1480 ± 0.1222**	0.0017 ± 0.0011	0.0030 ± 0.0027	7.0358E-13	5.8618E-13

to define the volumes. Comparing the results of Table 4.1 and 4.2, it is clear that the convergence ability of NSGA-II and NSPSO have become less effective compared to d–EMO. This is observed in the low HV values in those algorithms. It is also interesting to observe that there are fewer statistically insignificant comparisons between d–EMO and the non-dominated sorting based EMO algorithms. In the five-objective problem instances on average d–EMO shows larger HV values than NSGA-II and NSPSO indicating better convergence.

Ten-objective problems

Table 4.3 shows the results for the ten-objective problems. Here, the same values for δ' were used as in the five-objective problems to define the volumes. The decrease of performance in NSGA-II and NSPSO as the number of objectives increases can be seen clearly by comparing the results of Tables 4.1, 4.2 and 4.3. Table 4.3 has more 0.0000s for the non-dominated sorting algorithms compared to Table 4.2. This result indicates that NSGA-II and NSPSO have converged to an area in the objective-space far from the preferred region. All the results are statistically significant, so the average HV values contribute to meaningful comparisons. The reference point based d–EMO significantly outperforms reference point based NSGA-II and NSPSO in ten-objective problems. This shows that using preference information, EMO algorithms can be guided to find solutions more effectively in higher objective problems than using non-dominated sorting.

This process of calculating the HV values can be used to compare other user-preference based EMO algorithms. The key is to exclude solution points which do not lie within a common volume in the preferred region. This will ensure that the nadir point is defined within the preferred region, which makes the performance metric suitable for user-preference based EMO algorithms. It is also paramount to define an equal spread of solutions for each algorithm so that they converge onto the same preferred region.

4.4 Summary

In this chapter, we addressed the research question: *How can we measure the performance of a user-preference EMO algorithm using a suitable performance metric?* First, we described the limitations of the exiting EMO performance metrics that makes them unsuitable for user-preference EMO algorithms. This was the foundation to explore if the HV metric is better suited for user-preference algorithms because it does not require the knowledge of a known Pareto front.

Table 4.3: The average HV values for reference point based d–EMO, NSGA-II and on ten-objective test problems using one reference point. The values in ***bold*** *are the best values found for each problem instance and the values in italics show the comparisons which are not statistically significant.*

Problem	d–EMO	NSGA-II	NSPSO	p-value for d–EMO against NSGAII	p-value for d–EMO against NSPSO
DTLZ1	**0.4304** ± **0.0547**	0.0000 ± 0.0000	0.0000 ± 0.0000	5.9558E-46	5.9558E-46
DTLZ2	**0.0082** ± **0.0071**	0.0001 ± 0.0001	0.0000 ± 0.0000	3.5534E-12	6.3877E-08
DTLZ3	**0.1725** ± **0.0833**	0.0011 ± 0.0011	0.0000 ± 0.0000	3.7133E-19	4.8950E-10
DTLZ4	**0.0536** ± **0.0266**	0.0000 ± 0.0000	0.0000 ± 0.0000	4.7360E-19	4.7360E-19
DTLZ5	**0.0726** ± **0.0046**	0.0016 ± 0.0012	0.0000 ± 0.0000	2.0314E-59	1.2872E-60
DTLZ6	**0.0050** ± **0.0048**	0.0000 ± 0.0000	0.0000 ± 0.0000	3.4786E-10	3.4786E-10
DTLZ7	**0.0035** ± **0.0014**	0.0000 ± 0.0000	0.0000 ± 0.0000	2.1622E-23	2.1622E-23
WFG1	**0.0002** ± **0.0001**	0.0000 ± 0.0000	0.0000 ± 0.0000	2.7230E-18	2.7230E-18
WFG2	**0.0005** ± **0.0002**	0.0000 ± 0.0000	0.0000 ± 0.0000	1.1987E-25	1.1987E-25
WFG3	**0.0021** ± **0.0019**	0.0000 ± 0.0000	0.0000 ± 0.0000	2.1355E-10	2.1355E-10
WFG4	**0.0038** ± **0.0033**	0.0001 ± 0.0001	0.0000 ± 0.0000	4.3762E-08	4.8954E-10
WFG5	**0.0013** ± **0.0009**	0.0002 ± 0.0001	0.0000 ± 0.0000	0.0180E+00	0.0121E+00
WFG6	**0.0010** ± **0.0010**	0.0000 ± 0.0000	0.0000 ± 0.0000	1.6613E-09	1.6613E-09
WFG7	**0.0062** ± **0.0057**	0.0001 ± 0.0001	0.0000 ± 0.0000	1.6735E-11	1.2177E-11
WFG8	**0.0038** ± **0.0031**	0.0000 ± 0.0000	0.0000 ± 0.0000	1.2414E-12	1.2414E-12
WFG9	**0.0018** ± **0.0017**	0.0001 ± 0.0001	0.0000 ± 0.0000	3.8772E-10	2.5658E-10

The HV metric calculates the area between the solution points and a selected point in the objective-spaces, named the nadir point. Using this area bounded by the solutions, we can compare EMO algorithms on their convergence and spread of solutions. However, the standard HV calculation process is not suitable for user-preference EMO algorithms because they converge onto a smaller region of the Pareto front (the preferred region). Therefore, to make the HV metric meaningful for user-preference EMO algorithms, we only consider the solution points within a preferred region.

To exclude the solution points outside the preferred region, we extended the standard normalised HV calculation process by defining a volume in the objective-space. As in the standard HV calculation process, the final solution points of all the EMO algorithms in consideration from all the runs for a given problem are combined to obtain a grand population. From this population, the solution point closest to the ideal point is selected using the Euclidean distance between the solutions and ideal point. We define a volume around this solution point using a preference value defined by δ'. We use the solution point closest to the ideal point, to emphasise the EMO algorithm closest to the preferred regions.

We have illustrated that this extension to the HV calculation process is suitable for user-preference EMO algorithms by comparing d–EMO with NSGA-II on various many-objective problems. By observing the HV values across each generation we were able to visualise the convergence of the algorithms onto the preferred regions. These results clearly showed that the standard non-dominated sorting approach becomes ineffective in many-objective problems, even for simple problem instances.

It is important to note that the HV calculation is a relative measurement between the algorithms in consideration. The same HV values cannot be used as absolute values if we are to add more EMO algorithms optimising the same problem. In such a scenario we have to calculate the nadir point again with the new solution points from the other EMO algorithms added to the comparison. This step ensures a valid comparison of the performance using the HV metric.

In the next chapter, we introduce a user-preference hybrid DE and PSO algorithm, based on the distance metric. This algorithm is used to find solutions on some computationally intensive real-world problems. We chose a PSO version for these problems to take advantage of the faster convergence, which is more suitable for computationally demanding problems.

Chapter 5

A robust and efficient many-objective PSO algorithm

PSO algorithms have been used frequently in various engineering design problems. The fast convergence has been one of the main contributing factors to their popularity. Therefore, we present a PSO algorithm incorporating user-preferences for solving some many-objective real-world problems. First, we discuss the reasons for the popularity of PSO algorithms in multi-objective optimisation in Section 5.1. However, this fast convergence can have negative consequences as particles can get stuck in local optimal solution fronts. One remedy for this behaviour is found in the process of selecting leaders (global best) for the particles to follow. We present several existing leader selection schemes and propose a new leader selection method using DE within a multi-objective PSO algorithm.

In Section 5.2, we describe our hybrid DE and PSO algorithm for multi-objective optimisation. Here we present a mechanism of using DE to generate leaders so that the particles have a better chance to avoid getting stuck in local optimal fronts. In this section, we present some results of selected two and three-objective problem instances for this hybrid algorithm. In particular, these problem instances include the multi-modal problems from the ZDT, DTLZ and WFG test suites.

In Section 5.3 we illustrate how the hybrid DE and PSO algorithm is integrated into the distance metric based user-preference framework. To validate the performance of the hybrid PSO algorithm we present some results of multi-modal five and ten-objective problem instances of DTLZ suite, with the use of reference points and light beams respectively.

5.1 PSO algorithms in EMO

In Section 2.5 (page 28) we presented some instances of PSO algorithms used in multi-objective optimisation. In the EMO literature, PSOs have been very popular as optimisation algorithms because of their fast convergence [Trelea, 2003]. Reyes-Sierra and Coello Coello [2006] present a survey of many MOPSO algorithms characterising their velocity update rules and mutation schemes. The fast convergence of PSO algorithms is due to the particles exploring the search-space by updating their positions according to the *best particle* (or leader) in their neighbourhood. Ideally all the particles will try to move towards this best particle. As we have mentioned before in Section 2.4.2 (page 21), this behaviour is usually referred to as *following a leader*. PSOs are ideal candidates for user-preference EMO algorithms because they have this built-in property of being *guided* by leader particles. In user-preference EMO algorithms the DM guides the population towards the preferred regions. We can use this guidance from the DM to propel the swarm by influencing the leaders to move towards the preferred regions.

However, this movement of following certain particles of a population can become a disadvantage in problems where there are many local optimal fronts. The MOPSO algorithms can prematurely converge onto a local front rather than find the optimal global front [Clerc and Kennedy, 2002; Trelea, 2003]. We observed this down-side in our initial experiments in user-preference PSO algorithms [Wickramasinghe and Li, 2008]. Therefore, we explored mechanisms to avoid the swarm from getting stuck in local optimal fronts. To this end, we developed a hybrid PSO algorithm which uses a DE operator to *generate* leaders, rather than selecting leaders from the population itself. This mechanism of generating particles offers the ability to obtain a diverse range of leaders (a property of DE). Some of these leaders can be placed outside a local optimal front. This becomes an advantage to the MOPSO algorithm because now particles are attracted towards these leaders, which increases the likelihood of moving the population out of a local optimal front.

5.1.1 Leader selection schemes

The EMO literature describes many other leader selection schemes which have been used to address this issue of premature convergence onto a local optimal front. There are several examples of MOPSO algorithms where leaders are chosen by the notion of dominance. In DNPSO [Hu et al., 2003], particles choose a leader from their neighbours. Here, a particle will have a set of dynamically changing neighbours and it will choose the best particle among

them as a leader. Here the best particle is the dominant one from the set of particles that make up the neighbourhood. If several non-dominated particles exist in this set, the least crowded one is selected as the leader.

NSPSO [Li, 2003] adopts the non-dominated sorting mechanism proposed in NSGA-II [Deb et al., 2002] (see Algorithms 7 and 8 – pages 30 and 32). In NSPSO, leaders are chosen from the first non-dominated front of the current population. Here, each particle chooses a leader randomly from the top 10% of least crowded particles of the first non-dominated front, and updates their velocity and positions. The NSPSO algorithm then executes in an identical manner as in NSGA-II.

Li [2004] proposed the maximinPSO, which used the maximin fitness function [Balling, 2003] to derive dominance. In every iteration of the execution process a set of particles was extracted using the maximin fitness. From this set, the top 10% of particles with the least maximin value was chosen as candidates to be leaders. Each particle will create a leader by selecting values from all dimensions from this set of particles. In maximinPSO, the leader of a particle is not present in the population, but created from several candidates for leaders. However, the leaders generated in maximinPSO would still reside in the first non-dominated front.

In other studies of MOPSO algorithms, leaders were selected from an external archive. An elite archive is used by Fieldsend et al. [2003] to store the non-dominated particles from every iteration. The archive had a data structure called a *dominance tree* which was updated with the non-dominated particles found in each run. The algorithms proposed by Reyes-Sierra and Coello Coello [2005], and Mostaghim and Teich [2003] both used an archive to store particles, which are candidates for leaders. In the approach by Reyes-Sierra and Coello Coello [2005], the non-dominated particles are added to the archive in each iteration. This archive is truncated to a fixed size according to the crowding distance of the particles. This ensures that the least crowded particles will be candidates for leaders. The *Sigma method* was used by Mostaghim and Teich [2003] to obtain suitable particles as leaders in each iteration. These leaders are then stored in the archive in each iteration and truncated so that the particles with smallest sigma values are retained.

Allmendinger [2008] introduced a methodology of choosing a random particle from the population as a leader. A particle will adjust its velocity and position using this random particle. Then according to some replacement rules, the algorithm will decide whether to replace this leader or another particle if the particle now is in a dominant position.

In all of these MOPSO algorithms the leaders are extracted from members of the population. In contrast, in our hybrid DE and PSO algorithm, leaders are not necessarily particles of the current population.

5.1.2 Single-objective hybrid DE and PSO algorithms

In the EA literature, there exist several hybrid DE and PSO algorithms that have been used for single-objective optimisation problems. A study by Hendtlass [2001] proposed a PSO algorithm, where from time to time the DE operator is used to create an individual that will replace a particle rather than using the PSO rules to move it. A similar approach was also proposed by Hao et al. [2007] where the movement of the particles is determined partly by a DE operator and partly by PSO update rules. Here, a deterministic procedure was used to change the updating of particles from DE to PSO.

DEPSO [Zhang and Xie, 2003] uses the DE operator to update the personal best position of particles. However, this updating procedure is done once every couple of iterations. Xu et al. [2008] described a scheme where the PSO update rules were modified using a DE operator. Here, every particle underwent this modified PSO rule to move in the search-space.

In all these algorithms the movement of the particles was influenced by a DE operator. However, in our hybrid algorithm the particles move following the PSO rules, while only a leader is created using a DE operator. This mechanism ensures a faster convergence because of the PSO update rules, while retaining a MOPSO's ability to escape from a local optimal front because of a diverse range of leaders generated by the DE operator. Next, we describe our multi-objective hybrid DE and PSO variant.

5.2 The hybrid multi-objective DE and PSO algorithm: MDEPSO

First, we describe MDEPSO in Algorithm 12 which does not use user-preferences. The algorithm is executed until a maximum of t_{MAX} number of iterations is reached.

Step 1: Initialise the particles (line 1)

A swarm of size n is first initialised. For each particle $\vec{x_i}$ the decision variables (positions) $x_{j,i}$ are initialised from Equation 3.4 (see Section 3.1.3 – page 56). The velocity is initialised to a random value in the interval of $\left[0, x_j^{(UB)} - x_j^{(LB)}\right]$. The personal best ($\vec{p_i}$) of an individual is set to the current values of the decision variables ($p_{j,i} = x_{j,i}$). Half of the particles are set

Algorithm 12 The MDEPSO algorithm

1: INITIALISE a swarm (P_t) of size n, $t = 0$
2: EVALUATE each particle and assign fitness
3: $t = 1$
4: **repeat**
5: GENERATE leaders using the *DE/rand/1/bin* scheme
6: UPDATE the velocity using the *Constriction Type 1″* scheme
7: MUTATE particles
8: UPDATE positions and ADJUST boundary violations
9: EVALUATE each particle and UPDATE the personal best
10: COMBINE the swarm at the beginning of the iteration (P_t) with the one that has updated positions (Q_t) such that $R_t = P_t \cup Q_t$
11: NON-DOMINATE SORT R_t to obtain the next generation P_{t+1}
12: $t = t + 1$
13: **until** $t = t_{MAX}$

to the reverse direction by setting the velocity to negative according to a coin toss to better explore the search-space. The particles are then evaluated with the objective functions and fitness is assigned. The fitness values are used to determine the dominance.

Step 2: Generate leaders (line 5)

Each particle $\vec{x}_i$ will choose three other particles $\vec{x}_{r1}, \vec{x}_{r2}, \vec{x}_{r3}$ from the population such that $i \neq r1 \neq r2 \neq r3$. Then a trail vector ($\vec{u}_i$) is derived from the DE operator in Equation 2.9 (see Section 2.4.2 – page 25). In a traditional multi-objective DE algorithm, if $\vec{u}_i$ dominates $\vec{x}_i$ then $\vec{x}_i$ is replaced by $\vec{u}_i$, if not $\vec{u}_i$ is discarded [Abbass et al., 2001]. However, regardless of the domination aspect we use the trial vector $\vec{u}_i$ as the leader $\vec{p}_g$ for the particle $\vec{x}_i$.

The traditional values used for *DE/rand/1/bin* are $CR = 0.9$ and $F = 0.5$. However, we observed that these values for CR and F are unsuitable for multi-objective optimisation problems from our initial experiments. A higher CR value allows less frequent crossover operations. This contributes to placing leaders far from the vicinity of other particles in the search-space. These leaders could be placed in worse positions than the current particles. To overcome this issue, we used $CR = 0.2$ and $F = 0.4$, which locates the leaders near the vicinity of other particles. A lower CR value is also observed as desirable in other examples of multi-objective optimisation algorithms using DE in the EMO literature [Abbass et al., 2001; Huang et al., 2005; Kukkonen and Lampinen, 2005; Robic and Filipic, 2005].

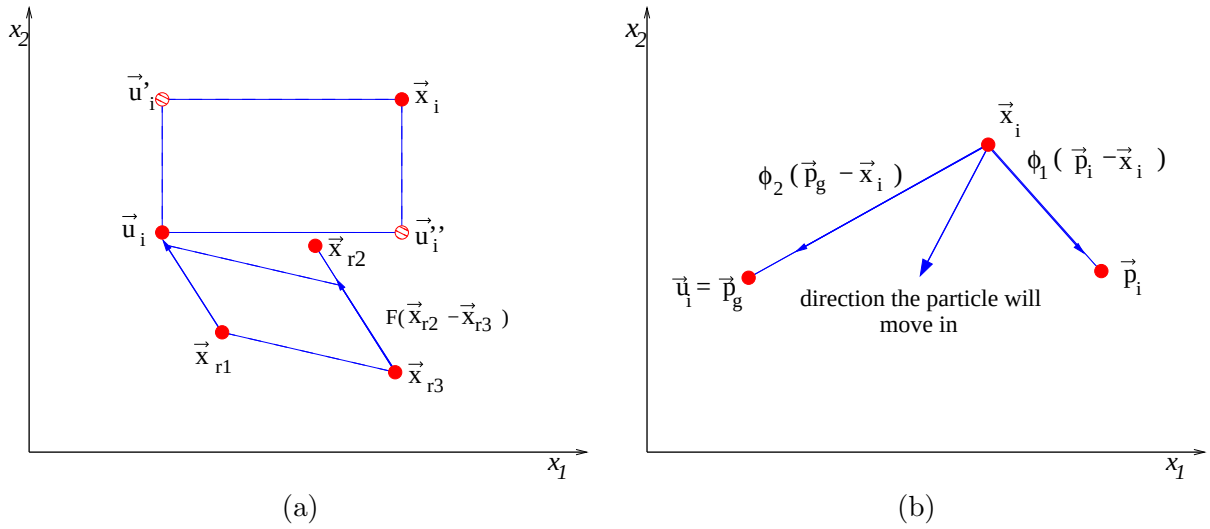


Figure 5.1: Generating a leader and moving a particle in MDEPSO: (a) Generating a leader vector using the DE operator; (b) Applying the PSO rules to move the particle.

Step 3: Update velocities (line 6)

Each particle will move towards their generated leader vectors as seen in Figure 5.1, updating their velocities and positions according to the PSO rules. In MDEPSO, we update the velocity using the *Constriction Type* $1''$ PSO update rules (see Equations 2.7 and 2.8 – pages 24 and 24).

Step 4: Mutate the particles (line 7)

A particle is mutated with a given probability using a PSO mutation rule. A particle is chosen for mutation if the probability is less than some P_{mut}. A particle chosen for mutation will update each decision variable j if the probability is again less than P_{mut}. The decision variable is mutated according to Equation 5.1. We used $P_{mut} = 0.1$ as the mutation probability.

$$x_{j,i} = \frac{\left(x_{j,i} + rand\left(x_j^{(LB)}, x_j^{(UB)}\right)\right)}{2} \tag{5.1}$$

Step 5: Update positions and adjust boundary violations (line 8)

Before evaluation of the particles, each particle is checked to see if the position vectors are within the boundaries of the decision variables. Unlike in standard PSO algorithms where the velocity is clamped (see Algorithm 2 – page 25), in MDEPSO we adjust the positions

using the bounce back scheme described in Algorithm 5 (page 28). We used the bounce back scheme because it preserves the current progress of the swarm.

Step 6: Update the particles' personal bests (line 9)

The particles are evaluated according to the objective functions and fitness is assigned. Next, the particles' personal bests are updated according to their current positions and best positions found so far. For any particle, if the current position $\vec{x}_i(t)$ dominates the best position $\vec{p}_i$ then $\vec{p}_i = \vec{x}_i(t)$.

Step 7: Obtain the particles to move to the next iteration (line 11)

The swarm of n particles at the beginning of the iteration (P_t) is combined with the n number of particles that have changed their positions (Q_t) to create a population (R_t) of size $2n$. The non-dominated sorting process (see Algorithm 7 – page 30) is applied to this $2n$ population to obtain n particles which are carried over to the next iteration (P_{t+1}). The steps 5 to 12 are continued until the maximum number of iterations t_{MAX} is reached.

5.2.1 Advantage of generating leaders using DE

To illustrate the advantage of generating leaders for MOPSO algorithms, we compare the execution processes of MDEPSO and NSPSO [Li, 2003] on the multi-modal two-objective ZDT4 test problem. Although the original NSPSO algorithm did not include a mutation scheme, we have included the mutation scheme in Equation 5.1 to increase the chances of escaping the local optimal fronts.

Figure 5.2(a) shows the generated leaders for ZDT4 test problem within an iteration. The DE operator takes the differences between vectors to generate a leader. This becomes an advantage because there is a chance for particles not in the first non-dominated front to be selected in the process of creating a leader. The vector differences between a particle in the first non-dominated front and one in a different front can be so large that the generated leader will be in a position outside the first non-dominated front. These leaders can now successfully attract other particles to move out of the current local optimal front towards the global optimal front. In other MOPSO algorithms, particles would rarely move outside a local optimal front because all leaders are trapped to the same front, as illustrated in Figure 5.2(b). This feature of generating leaders outside of the current front, gives MDEPSO the ability to escape a local optimal front.

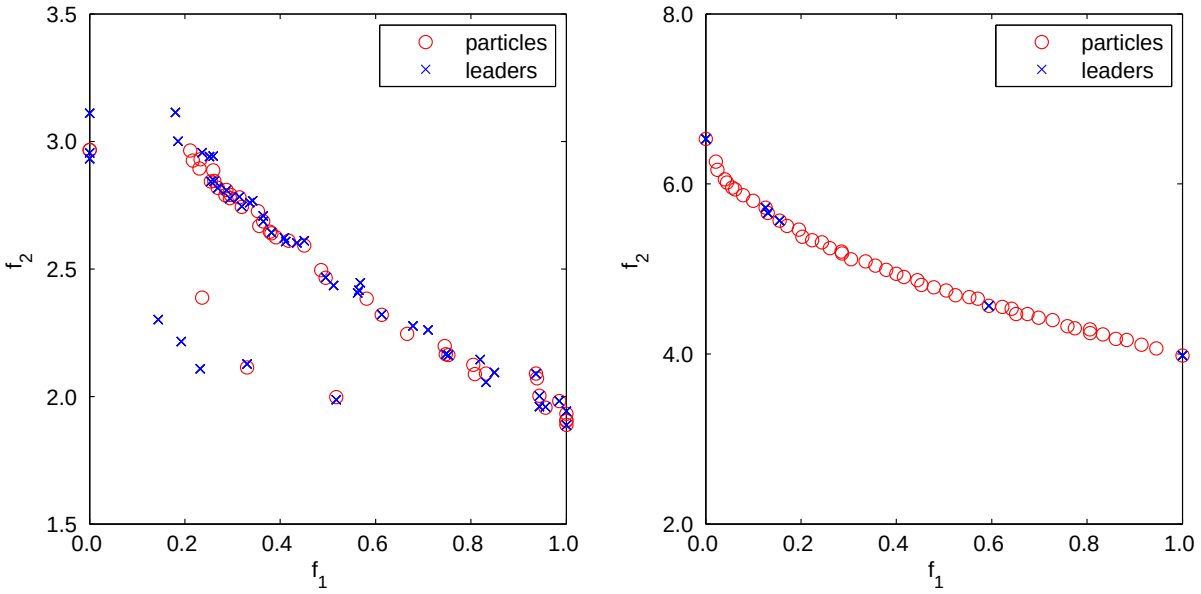


Figure 5.2: Positions of leaders for ZDT4 within an iteration: (a) MDEPSO; (b) NSPSO. These results are from the same iteration of the two EMO algorithms. This iteration was chosen to highlight the advantage of generating leaders over selecting particles as leaders.

Selecting all leaders from the first non-dominated front or an archive (Figure 5.2(b)), can also be a disadvantage for MOPSO algorithms mainly because many particles could follow the same leader. This restrains the PSO algorithm from exploring the search-space effectively. We also observe from Figure 5.2 that MDEPSO has more diverse leaders for the particles to follow than for NSPSO. A similar observation can be made for PSO algorithms picking leaders from an archive. Figure 5.2 illustrates the clear advantage of using the DE operator to generate leaders to give MOPSO algorithms a better chance to locate the global optimal front and still retain the fast convergence ability.

5.2.2 Experimental results

To illustrate the performance of MDEPSO, we executed it alongside some other MOPSO algorithms on various test problems. In this section, our aim is to show that MDEPSO is able to find the entire Pareto front of some two and three-objective problems. We do not intend to show results of many-objective problems, because MDEPSO uses non-dominated sorting and thus ineffective in high-objective problems (see Section 2.5.2 – page 35). Later, in Section 5.3 we extend MDEPSO to use user-preferences and replace the non-dominated

sorting procedure with the distance metric, so that the algorithm can effectively find solutions in many-objective problem instances.

We used the following problems to test the convergence of the PSO algorithms; two-objective ZDT test suite, three-objective DTLZ and WFG test suites. We present the standard normalised HV metric as described in Section 4.2.2 (page 94) to provide a relative measure of the convergence and spread of solutions along the solution fronts. We compare the effectiveness of MDEPSO against NSPSO [Li, 2003], maximinPSO [Li, 2004] and OMOPSO [Reyes-Sierra and Coello Coello, 2005] on all the test problems. NSPSO was chosen because leaders are chosen from the first non-dominated front at each iteration. In maximinPSO, leaders are generated by combining information from particles of the first non-dominated front at each iteration. OMOPSO was chosen because it uses an archive to store the leaders. This archive is updated at every iteration with the least crowded dominant particles, so that the best particles are retained as leaders for the rest of the execution process. We employed the same mutation rule given in Equation 5.1 and $P_{mut} = 0.1$ on NSPSO and maximinPSO also, because the original versions did not contain a mutation rule. OMOPSO already had a mutation rule in its functionality.

For each problem a constant population of 200 individuals was used. All the algorithms were executed for a maximum of 500 iterations on all test problems. For simpler problems like ZDT1–ZDT3, MDEPSO found the Pareto optimal fronts in under 50 iterations, but for consistency the maximum number of iterations was fixed. Each algorithm was executed for 50 runs on each test problem and the HV values were averaged. The parameter values for MDEPSO, NSPSO and maximinPSO were the standard values presented in Constriction Type $1''$ PSO. $CR = 0.2$ and $F = 0.4$ were used for the DE operator in MDEPSO. An archive size of 100 and $P_{mut} = 0.5$ was used in OMOPSO as presented by Reyes-Sierra and Coello Coello [2005].

To show the statistical significance of the average HV values, we employ the Student's t-test between MDEPSO and NSPSO, MDEPSO and maximinPSO, MDEPSO and OMOPSO. This t-test tests the null hypothesis as described in Section 4.3.2 (page 103).

Comparing the results from Tables 5.1, 5.2 and 5.3, it is clear that MDEPSO outperforms the other MOPSO algorithms in the test problems for convergence, especially for the multimodal problems (ZDT4, DTLZ1, DTLZ3 and WFG4). The results in **bold** are the best HV values found for each problem, while the results in *italics* are the statistically insignificant comparisons. MDEPSO gives a higher hyper-volume value than the other PSO algorithms on average.

Table 5.1: Normalised HV values for MDEPSO, NSPSO, maximinPSO and OMOPSO on the two-objective ZDT test suite. The values in ***bold*** *are the best values found for each problem instance and the values in italics show the comparisons that are not statistically significant.*

Problem	MDEPSO	NSPSO	maximinPSO	OMOPSO	p-value for MDEPSO against NSPSO	p-value for MDEPSO against maximinPSO	p-value for MDEPSO against OMOPSO
ZDT1	**0.6637** ± **0.0001**	**0.6637** ± **0.0001**	0.6632 ± 0.0002	0.6620 ± 0.0000	1.6094E-20	1.0719E-40	1.0457E-69
ZDT2	**0.3304** ± **0.0001**	0.3304 ± 0.0001	0.3300 ± 0.0001	0.3287 ± 0.0000	7.4252E-19	9.4895E-38	5.4478E-66
ZDT3	**0.5166** ± **0.0000**	0.5167 ± 0.0000	0.5163 ± 0.0002	0.5159 ± 0.0000	2.8455E-53	2.0271E-12	3.5487E-84
ZDT4	**0.6628** ± **0.0005**	0.5331 ± 0.2053	0.6448 ± 0.0514	0.0022 ± 0.0114	0.0001E+00	0.0006E+00	6.1819E-43
ZDT6	0.4032 ± 0.0154	**0.4033** ± **0.0001**	0.4027 ± 0.0001	0.4014 ± 0.0001	2.0978E-29	1.0014E-65	1.3577E-64

Table 5.2: Normalised HV values for MDEPSO, NSPSO, maximinPSO and OMOPSO on the three-objective DTLZ test suite. The values in ***bold*** *are the best values found for each problem instance and the values in italics show the comparisons that are not statistically significant.*

Problem	MDEPSO	NSPSO	maximinPSO	OMOPSO	p-value for MDEPSO against NSPSO	p-value for MDEPSO against maximinPSO	p-value for MDEPSO against OMOPSO
DTLZ1	**0.7754** ± **0.0036**	0.0840 ± 0.0059	0.0925 ± 0.0008	0.0000 ± 0.0000	5.7701E-12	4.6816E-08	1.8871E-11
DTLZ2	**0.4191** ± **0.0016**	0.4032 ± 0.0029	0.3885 ± 0.0030	0.3721 ± 0.0037	2.1302E-57	1.0507E-58	8.9394E-62
DTLZ3	**0.3628** ± **0.0053**	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0403E+00	0.0403E+00	0.0403E+00
DTLZ4	**0.4132** ± **0.0018**	0.3948 ± 0.0024	0.3797 ± 0.0031	0.3739 ± 0.0030	9.0536E-16	7.2834E-64	1.1793E-42
DTLZ5	***0.0945*** ± ***0.0001***	***0.0945*** ± ***0.0001***	0.0939 ± 0.0001	0.0940 ± 0.0000	*0.2167E+00*	8.9359E-38	7.4458E-44
DTLZ6	**0.0959** ± **0.0000**	0.0950 ± 0.0000	0.0956 ± 0.0001	0.0949 ± 0.0000	3.0098E-14	8.0024E-30	1.3665E-63
DTLZ7	***0.3092*** ± ***0.0014***	*0.3080* ± *0.0017*	0.3076 ± 0.0171	0.2758 ± 0.0030	*0.6369E+00*	0.0089E+00	4.0443E-62

Table 5.3: Normalised HV values for MDEPSO, NSPSO, maximinPSO and OMOPSO on the three-objective WFG test suite. The values in ***bold*** *are the best values found for each problem instance and the values in italics show the comparisons that are not statistically significant.*

Problem	MDEPSO	NSPSO	maximinPSO	OMOPSO	p-value for MDEPSO against NSPSO	p-value for MDEPSO against maximinPSO	p-value for MDEPSO against OMOPSO
WFG1	**0.3276**	0.3255	0.3123	0.3186	8.6082E-14	2.2285E-46	5.7199E-33
	± **0.0006**	± 0.0011	± 0.0019	± 0.0020			
WFG2	**0.9219**	0.9143	0.9163	0.9002	2.6645E-07	2.8720E-27	6.1755E-34
	± **0.0009**	± 0.0017	± 0.0014	± 0.0030			
WFG3	**0.3276**	0.3255	0.3123	0.3186	8.6082E-14	2.2285E-46	5.7199E-33
	± **0.0006**	± 0.0011	± 0.0019	± 0.0020			
WFG4	**0.3980**	0.3691	0.3650	0.3334	2.1163E-17	1.2473E-14	4.8756E-40
	± **0.0034**	± 0.0041	± 0.0074	± 0.0056			
WFG5	**0.3902**	0.3755	0.3668	0.3342	1.5545E-20	1.6575E-38	1.6543E-29
	± **0.0022**	± 0.0029	± 0.0030	± 0.0000			
WFG6	**0.4169**	0.4010	0.4010	0.3703	0.9055E+00	4.4678E-26	2.9219E-34
	± **0.0028**	± 0.0043	± 0.0042	± 0.0052			
WFG7	***0.3884***	*0.3613*	*0.3430*	0.3578	*0.0883E+00*	*0.8539E+00*	1.5011E-34
	± ***0.0058***	± *0.1088*	± *0.1671*	± 0.0052			
WFG8	**0.2965**	0.2781	0.2599	0.2395	4.0327E-13	7.2658E-26	1.1244E-15
	± **0.0028**	± 0.0046	± 0.0122	± 0.0050			
WFG9	**0.4077**	0.3897	0.3854	0.3639	2.7584E-08	3.0282E-36	4.1471E-25
	± **0.0025**	± 0.0028	± 0.0035	± 0.0055			

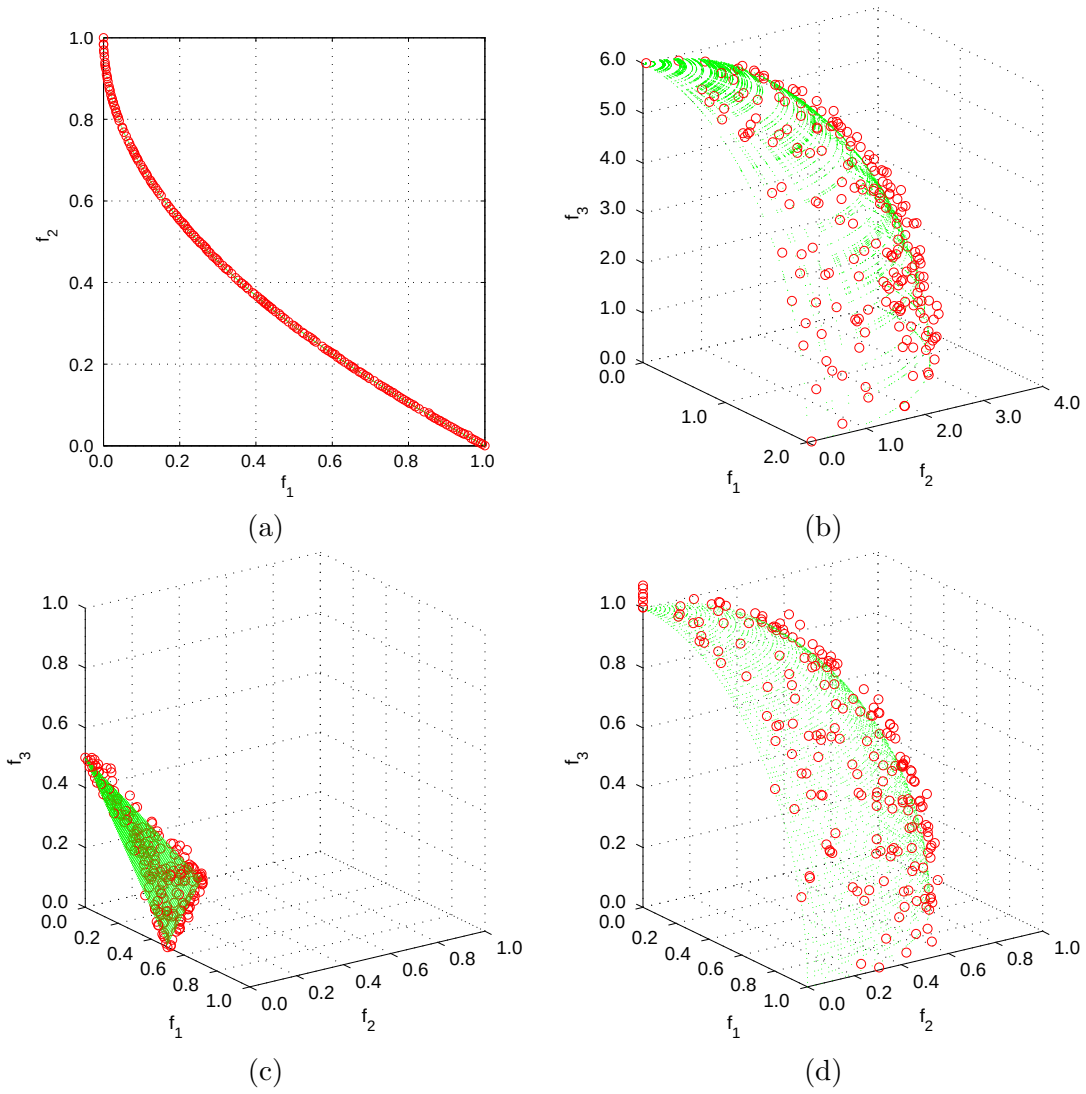


Figure 5.3: Solution fronts obtained by MDEPSO on some multi-modal problems: (a) Two-objective ZDT4; (b) Three-objective WFG4; (d) Three-objective DTLZ1; (e) Three-objective DTLZ3.

Figure 5.3 shows the final non-dominated solution fronts obtained by MDEPSO after 500 iterations for the multi-modal problems. These results show that the final solutions from MDEPSO are very close to the Pareto optimal fronts. The results in Tables 5.1, 5.2 and 5.3, and Figure 5.3 show that MDEPSO is capable of handling difficult multi-modal multi-objective problems having two or three-objectives. However, we are more interested in using MDEPSO for real-world applications having many-objectives. These real-world

applications usually have very noisy (challenging) multi-modal objective-spaces. Therefore, next we describe how the distance metric is integrated into MDEPSO.

5.3 Integrating the distance metric into MDEPSO

To use MDEPSO within the distance metric framework requires some minor modifications in Algorithm 12. We describe these steps in Algorithm 13 for one preferred region.

Algorithm 13 MDEPSO using the distance metric

1: OBTAIN preferences from the DM
2: INITIALISE a swarm (P_t) of size n, $t = 0$
3: EVALUATE each particle and assign fitness, and distance to the preferred region
4: $t = 1$
5: **repeat**
6: GENERATE leaders using the *DE/rand/1/bin* scheme
7: UPDATE the velocity using the *Constriction Type 1″* scheme
8: MUTATE particles
9: UPDATE positions and ADJUST boundary violations
10: EVALUATE each particle and UPDATE the personal best
11: COMBINE the swarm at the beginning of the iteration (P_t) with the one that has updated positions (Q_t) such that $R_t = P_t \cup Q_t$
12: SORT R_t according to Step 10 of Algorithm 11 obtain the next iteration P_{t+1}
13: $t = t + 1$
14: **until** $t = t_{MAX}$

In the user-preference MDEPSO algorithm, first the DM will indicate the preferences as described in Algorithm 11 (page 56). Next, the swarm is initialised according the initialisation steps described in Algorithm 12. Using the preference mechanism the distance metric is derived and each particle is assigned their distance values. A particle $\vec{x}_i$ at iteration t will generate a leader, and update its velocity and position according to the same steps in MDEPSO. The functionality of MDEPSO is unchanged in the user-preference version. However, in this version a particle will update its personal best according to the distance metric values, not by the dominance relation as in Algorithm 12. A particle $\vec{x}_i(t)$ is said to be in a better position if and only if its new distance metric value is less than the distance metric value of the current personal best. This is written as:

$$\vec{p}_i = \vec{x}_i(t) \Leftrightarrow dist(\vec{x}_i(t)) < dist(\vec{p}_i) \tag{5.2}$$

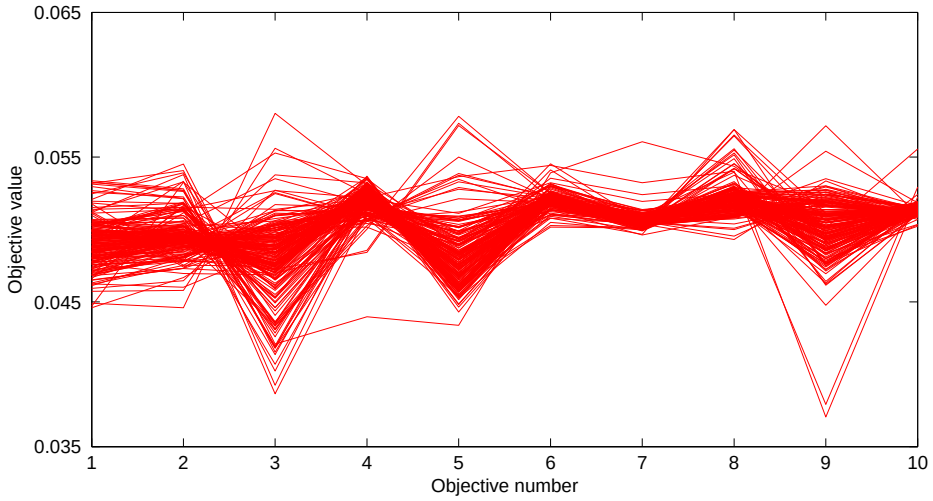
The n number of particles (P_t) at the beginning of iteration t are combined with the particles that have changed positions (Q_t) to create a swarm (R_t) of size $2n$. From R_t, n number of particles are selected according to the process described in Step 10 of Algorithm 11 to move to the next iteration. The user-preference MDEPSO algorithm will terminate once it has reached iteration t_{MAX}.

Although we are more interested in real-world applications for the user-preference MDEPSO algorithm, for completeness we present some results of the difficult multi-modal problems from the ZDT, DTLZ and WFG test suites. The MDEPSO algorithm implementing the reference point method and light beam search both had 200 particles and were executed for a maximum of 500 iterations. Other parameters were the same as per the standard MDEPSO algorithm. We present the final solution fronts from the best converged run from 50 independent runs.

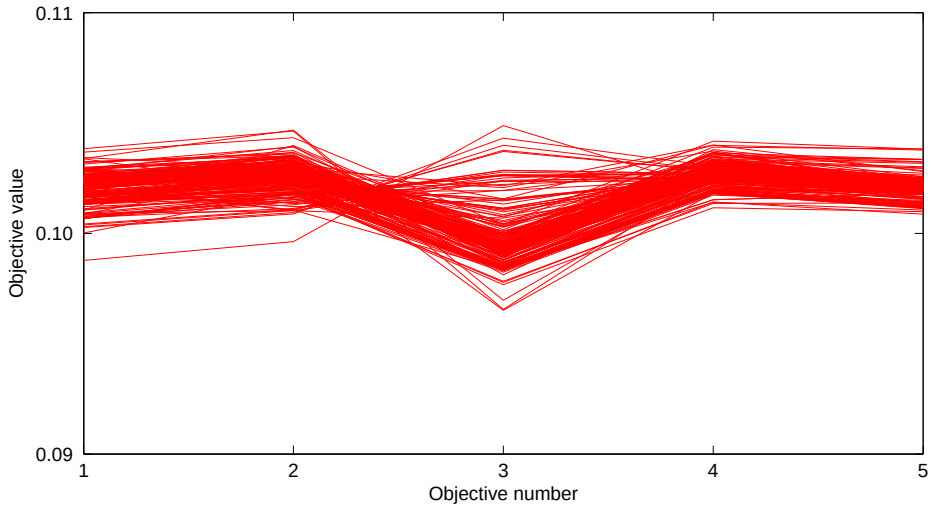
5.3.1 MDEPSO with reference points

Figure 5.4 shows the result obtained for ten and five-objective multi-modal DTLZ1 instances. Here, the reference point was at 0.5 for all of the objectives in the objective-space. There was no bias towards any objective and $\delta = 0.01$. For the ten-objective problem, the sum of the objective values of each particle was found to be in the range $[0.5012, 0.5109]$. For the five-objective problem, the sum of the objective values were in $[0.5019, 0.5164]$. These results suggest that the particles were very close to the true Pareto fronts of DTLZ1 because a particle $\vec{x}$ is on the Pareto front if and only if $\sum_{i=1}^{M} f_i(\vec{x}) = 0.5$.

Figure 5.5(a) shows the solutions fronts obtained for the two-objective multi-modal ZDT4 problem with two reference points on MDEPSO. Here, the two reference points have spread values of $\delta = 0.01$ and $\delta = 0.05$. MDEPSO was able to locate solutions on the global optimal front, without getting stuck on a local optimal front. The results obtained for ZDT6 with two reference points is given in Figure 5.5(b). Here, spread values of $\delta = 0.01$ and $\delta = 0.1$ were used for the two reference points separately. It is also interesting to note that the reference point at $(0.2, 0.9)$ is outside the region bounded by the Pareto optimal front. However, MDEPSO successfully obtained the solutions near the reference point at the extreme end of the Pareto front. Figures 5.5(c) and 5.5(d) illustrates the solution fronts obtained for two-objective multi-modal WFG4 and three-objective DTLZ1 problem instances respectively on MDEPSO. Here, $\delta = 0.01$ and no bias was used in any objective. MDEPSO



(a)



(b)

Figure 5.4: Many-objective DTLZ1 with one reference point on MDEPSO: (a) Ten-objective DTLZ1; (b) Five-objective DTLZ1.

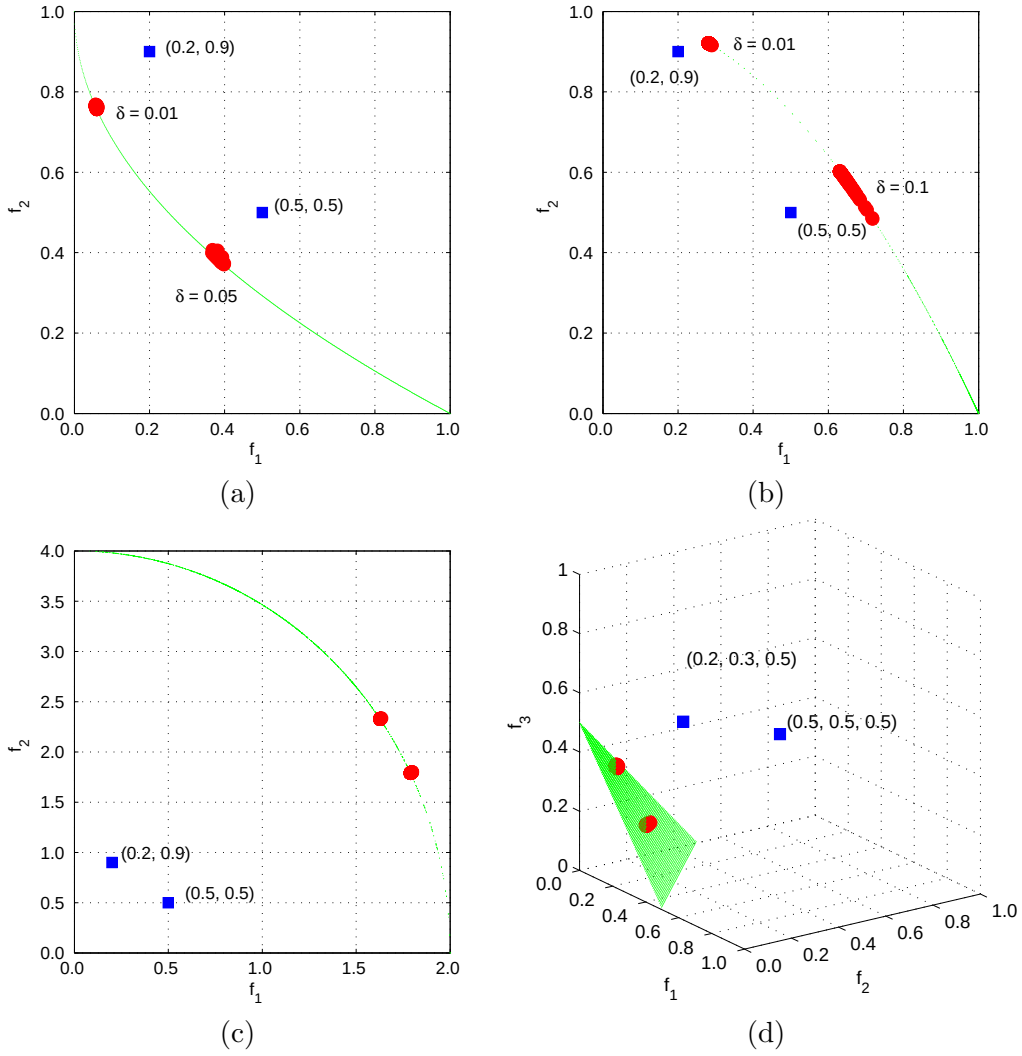


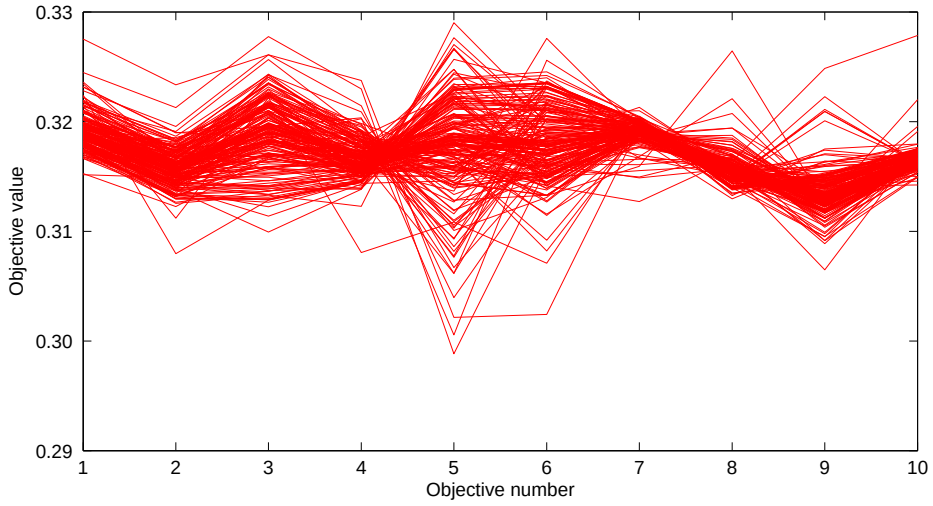
Figure 5.5: Results of MDEPSO having reference points: (a) Two-objective ZDT4; (b) Two-objective ZDT6; (c) Two-objective WFG4; (d) Three-objective DTLZ1.

was again successful in locating solutions on global optimal fronts of the preferred regions of these multi-modal problem instances. These results show that MDEPSO can be guided using reference points to obtain solutions on the Pareto fronts.

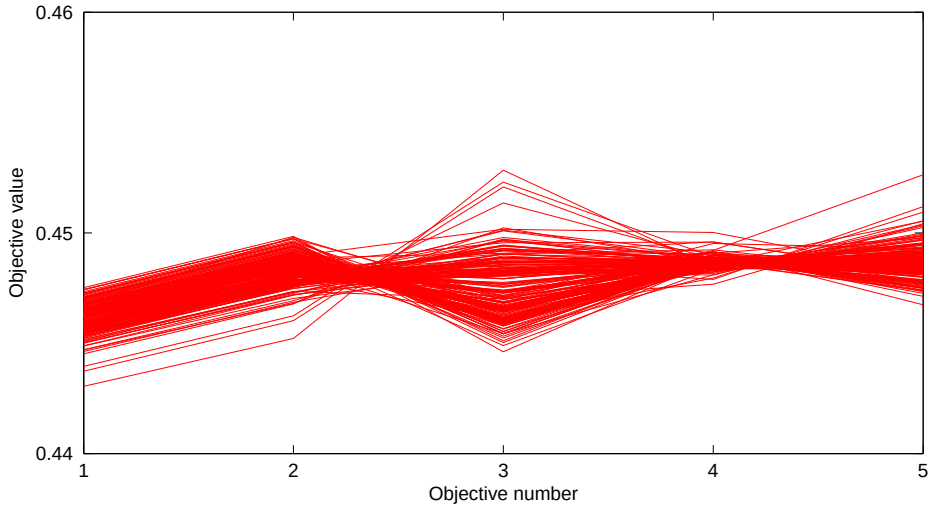
5.3.2 MDEPSO with light beams

In our experiments for the light beam search we used various APs and RPs with different spreads of solutions. In the ten and five-objective DTLZ3 instances, the AP was set to have 1.0 for all objectives and the RP to have 0.0 for all objectives. DTLZ3 is one of the more difficult multi-modal problems having close to 3^{10} local optimal fronts and one global optimal front. The solutions given in Figure 5.6 show that for each particle $\vec{x}$, the sum of its squared objective values $\left(\sum_{i=1}^{M}(f_i(\vec{x}))^2\right)$ gives values in the range $[1.0050, 1.0083]$. In the five-objective problem the values were in the range $[1.0013, 1.0083]$. These results show that the particles are very close to the Pareto fronts. This is deduced from the property of DTLZ3 where each $\vec{x}$ on the true Pareto front gives $\sum_{i=1}^{M}(f_i(\vec{x}))^2 = 1$.

Figure 5.7(a) shows a two-objective WFG2 problem instance with two light beams. It is interesting to observe that although the light beam (with AP $(5.0, 5.0)$ and RP $(0.0, 2.0)$) goes through the disjoint Pareto front, MDEPSO was still able to locate solutions on the region of the Pareto front which is closest to this light beam. The distance metric guides particles in the direction given by the vector from AP to RP. This is possible because the algorithm concentrates its search in the direction of this vector. With a swarm of particles the algorithm has the ability to move in parallel along the direction of this vector until a middle point is found on the Pareto front. A very interesting result can be seen for the two-objective WFG4 in Figure 5.7(b). Here, the light beams are located in the infeasible region of the objective-space. However, MDEPSO using the light beam search still managed to guide the swarm in the direction of the light beams until solutions are located on the global Pareto front. Figure 5.7(c) shows the final solutions obtained for DTLZ3 with three-objectives. Here, two light beams having AP $(1.0, 1.0, 1.0)$ and RPs at $(0.5, 0.0, 0.0)$ and $(0.0, 0.0, 0.0)$ respectively were used. Two spread values were having $\delta = 0.1$ and $\delta = 0.05$. These results show that MDEPSO can be used with the light beam search also to guide the swarm across the many local optimal fronts to obtain solutions on the global optimal fronts.



(a)



(b)

Figure 5.6: Many-objective DTLZ3 with one light beam on MDEPSO: (a) Ten-objective DTLZ3; (b) Five-objective DTLZ3.

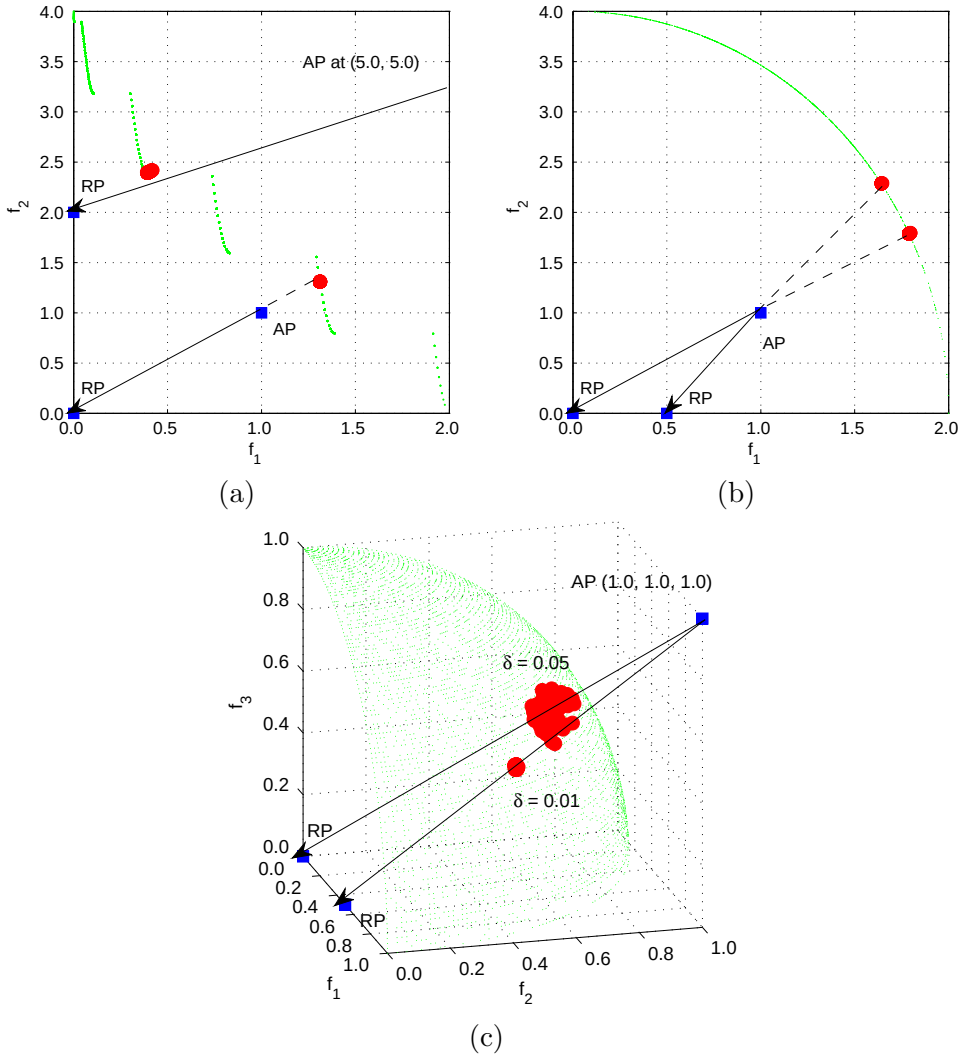


Figure 5.7: Results of MDEPSO having light beams: (a) Two-objective WFG2; (b) Two-objective WFG4; (c) Three-objective DTLZ3.

These various test problems illustrated that MDEPSO is capable of being guided using user-preferences to successfully obtain solutions on many-objective problems. MDEPSO is robust to handle multi-modal problems and can be guided using the refined knowledge of the DM.

5.4 Summary

In this chapter we addressed the research question: *How can we develop a robust PSO algorithm which can be used in many-objective problems?* This discussion started by illustrating the popularity of PSO algorithms in the EMO process. Although the fast convergence of the PSO algorithms was a contributing factor to their popularity, it also had a negative impact on the optimisation process. This is because the swarm can get stuck in a local optimal front. Therefore, we presented some methods for leader selection which can be embedded inside a PSO algorithm. This in turn can provide some resistance towards getting stuck in local optima.

Among the many schemes for selecting leaders, we found the use of DE rules within PSO rules to be most effective. Some studies on single-objective problems have shown that using DE rules to modify the personal best or modify the process of updating the velocities have a positive impact in improving PSO algorithms on multi-modal problems.

Most of the PSO algorithms chose particles as leaders from the existing swarm or an archive. We use the DE/rand/1/bin rule to generate a leader for each particle to follow. This process generates new leaders, which can be positioned outside a local optimal front where the swarm is currently stuck. Therefore, particles following these leaders will be most likely to escape the local optimal front and eventually reach the global optimal front. We named this hybrid DE and PSO multi-objective algorithm as MDEPSO.

Using various two and three-objective test problems from the ZDT, DTLZ and WFG test-suites we illustrated the performance of MDEPSO. We emphasised the multi-modal problems from these test suites, but for completeness all the problems were considered in the comparison studies. MDEPSO was compared with NSPSO, maximinPSO and OMOPSO, which are examples of MOPSO algorithms found in the literature. The convergence ability of MDEPSO was justified with the use of normalised HV values and figures illustrating the convergence on the Pareto front.

We have incorporated the distance metric described in Chapter 3 into MDEPSO so that it is able to better handle many-objective problems. We then provided the convergence of

MDEPSO on some selected problems in the ZDT, DTLZ and WFG test suites with the use of reference points and light beams. These results showed that MDEPSO can be used in many-objective problem instances, where additional guidance is provide by the DM.

The usage of user-preferences was observed as a natural extension to the guidance provided by the leaders in PSO algorithms. Therefore, we apply MDEPSO with user-preferences on two computationally expensive engineering design problems. In the next two chapters, we describe aerofoil designs for unmanned aircraft and camera lens designs used in still photography, which are examples of real-world many-objective problems.

Chapter 6

Real-world application: Aerofoil design

Our research began with the aim of exploring effective EMO algorithms that can find solutions in many-objective problems that are interesting to the DM. In Chapter 3, we described an efficient framework based on user-preferences to guide EMO algorithms in finding solutions on many-objective problem instances. We then presented a suitable performance metric to compare user-preference EMO algorithms in Chapter 4. So far, we have presented results on these two components based on many-objective test problems, which have known Pareto fronts. However, we indicated in Chapter 5 the rationale of the proposed robust MOPSO algorithm named MDEPSO, which was to be used in real-world applications. In this chapter and next chapter, we use this user-preference MDEPSO algorithm in the optimisation process of two real-world applications. Unlike the test problems, the real-world applications discussed do not have known Pareto fronts. This makes the HV calculation process introduced in Chapter 4 very useful in monitoring the optimisation process.

In this chapter we present an application for optimising aerofoils for unmanned low flying aircraft[1]. We begin by providing some basics of aerofoil optimisation referencing existing numerical and evolutionary computing based optimisation methods in Section 6.1. Then, we define the problem instance by introducing the decision variables and their bounds in Section 6.2. The decision variables represent a parameterised scheme to define aerofoils. We

[1]This chapter has been a result of the collaborative work carried out at the School of Computer Science and I.T, and the School of Aerospace Mechanical and Manufacturing Engineering, of RMIT University. This inter-disciplinary collaborative work have resulted in publications by Carrese et al. [2010 (under review] and Wickramasinghe et al. [2010].

define six objective functions that represent flying conditions in Section 6.3. Each objective function conflicts with at least one other objective function, but not necessarily all at once. However, this suffices for the proof of concept of using user-preference EMO algorithms in real-world applications in this thesis. The decision variables and objectives are used to define the aerofoil design problem instance. We also present an exiting aerofoil in Section 6.3, which is used as a reference design. This reference aerofoil is used to derive aspiration values to define a reference point for the optimisation process involving the MDEPSO algorithm. In Section 6.4 we present the aerofoil analysis tool used in our optimisation process. This external *evaluator* is used to analyse the candidate aerofoil shapes and extract data that are used to evaluate the objectives. In Section 6.5, we present the optimisation process. We describe this process as a generalised framework, which is also to be used in Chapter 7 for the lens design application.

The main goal of this chapter is to use the existing aerofoil design to obtain better aerofoils that can operate in the conditions provided by the objective functions. Unlike in the test problems, we analyse the obtained designs with respect to the actual functionality of the objectives comparing them with the reference design models. However, we also present some experimental results comparing the reference point MDEPSO algorithm and the standard MDEPSO algorithm for each many-objective application. These results and analysis are presented in Section 6.6.

The main motivation for these real-world application chapters lies in the investigation of the advantages of user-preference EMO algorithms in problems where the refined knowledge of a DM is integrated into the optimisation process. These chapters also bring all the research components presented in the previous chapters together to fulfil goals in the optimisation process of *unknown* many-objective problem instances.

6.1 Aerofoil designs for unmanned aircraft

Shape optimisation has been studied to a great extent in a wide range of engineering disciplines [Haslinger and Makinen, 2003]. For example, in the automotive industry, shape optimisation is actively pursued in an attempt to increase down-force and handling to maintain high manoeuvring speeds during cornering of the vehicles in restrictive environments. The standard method of optimising a shape has been to start with an initial guess of the intended shape and iteratively *refine* it until an optimal design is reached. This optimisation process usually is carried out using numerical methods, such as gradient based approaches,

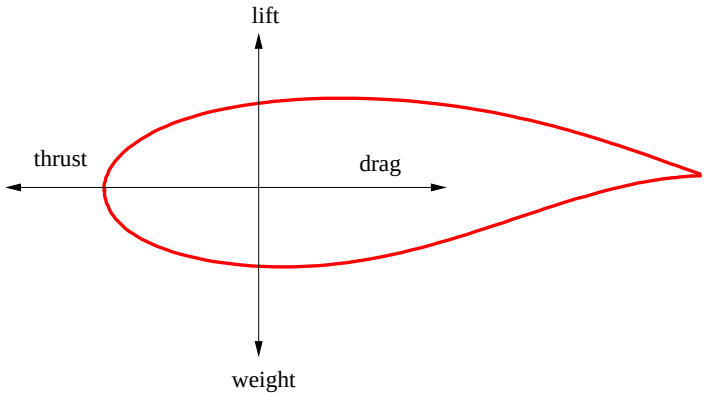


Figure 6.1: Weight: The force caused by the gravitational force acting on the mass of the body. Lift: The component of the force that is perpendicular to the oncoming flow, either air or fluid. Thrust: The force that propels body through the medium. Drag: The force that occurs in the opposite direction of the thrust. Equilibrium of drag, lift, thrust and weight are required to maintain a steady flight through a fluid medium.

until the shape conforms to a given objective or requirement [Haslinger and Makinen, 2003].

In this section, we describe some shape optimisation requirements found in the aerospace industry. More specifically, we are interested in shape optimisation requirements of low speed Unmanned Aerial Vehicles (UAV). In UAVs, the process of aerodynamic shape optimisation is critical during all phases of its design [Khurana et al., 2009a]. UAVs are interesting subjects in aerodynamic design optimisation, because of ratio of the forces acting on them. In such designs, the shapes (*bodies*) where the force, which is parallel and opposite to the direction of motion (*drag*) is significantly smaller in magnitude to the force component acting normal to the direction of motion (*lift*). Figure 6.1 illustrates these forces acting on an aerodynamic body.

A sub-discipline in UAV design that has benefited greatly from optimisation theory in the recent past is aerofoil design [Keane and Nair, 2005]. In this section, we use the reference point MDEPSO algorithm to obtain optimum designs for aerofoils, which satisfy some requirements. However, before we describe this process it is useful to illustrate the components of aerofoils and a method to define their shapes.

6.1.1 Aerofoil design

An aerofoil is the cross-section of any three-dimensional lifting surface, such as a wing. Figure 6.2 illustrates the basic components of an aerofoil. The *chord* is the imaginary line

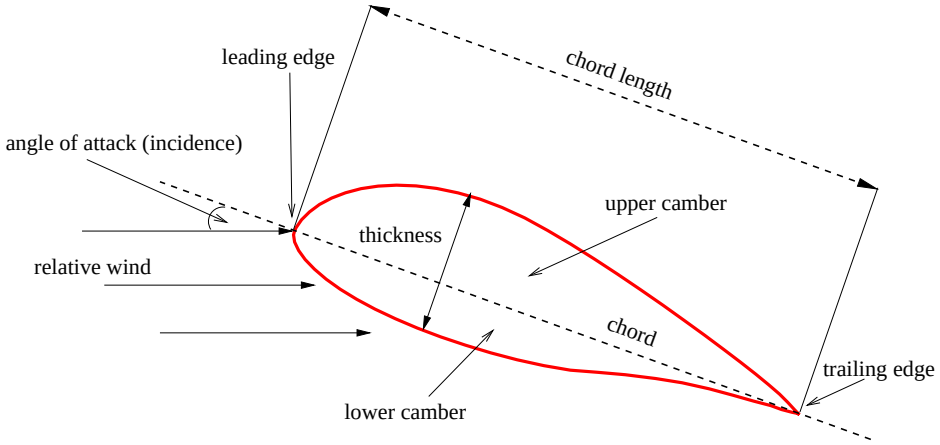


Figure 6.2: Components of an aerofoil. Chord: The imaginary line joining the trailing edge with the centre of curvature of the leading edge. Camber: Asymmetry between the top and bottom curves. Thickness: expressed as a percentage of the chord.

joining the trailing edge of an aerofoil with the centre of curvature of the leading edge. *Camber* denotes the asymmetry between the top and bottom curves of the aerofoil. The *thickness* of an aerofoil is usually expressed as a percentage of the chord. The thickness is also known as the *maximum thickness*. For example, the NLF0416 aerofoil [Somers, 1981] is a 16% thick aerofoil. This aerofoil will be used later as a reference design in our optimisation process.

Subsonic forces on aerofoil sections arise from the surface pressure and air viscosity. The term *subsonic forces* indicate the forces occurring on a body when travelling at speeds lower than the speed of sound. The pressure difference of the upper and lower surfaces produces the resultant lift force. This lift is required to maintain level flight for a given *operating condition*. An operating condition is the different states an aircraft is in during flight. For example, some of the operating conditions are take-off, landing, approach and cruising. The location where the lift force acts generates a moment. This moment governs the magnitude and direction of the pitch inclination. Drag is comprised of both pressure and viscous components; the latter is dominant in subsonic flight. Viscosity of the flow in the immediate vicinity of the surface (*boundary layer*) generates surface shear stress from which the drag is derived.

6.1.2 Aerofoil optimisation

The most popular design methodology in aerodynamic shape optimisation follows an integration of three distinct modules [Jin and Sendhoff, 2009; Khurana et al., 2009b; Quagliarella and Vicini, 2001; Ray and Tsai, 2004]. They are:

- A geometric parameterisation model
- An efficient search engine
- A computational flow solver

A *geometric parameterisation model* is used to express an aerofoil using several parameters, rather than a vast number of coordinates that defines the shape on a two-dimensional Cartesian plot [Song and Keane, 2004]. This strategy makes the optimisation process less computationally intensive than the latter; because the optimisation is carried out on few design (decision) variables. We describe the parameterisation model used in this thesis later in Section 6.2.

For any given operating condition, deviations to the aerofoil geometry will result in variations to the pressure and shear stress distributions. These variations of the pressure and stress will in-turn alter the force values acting on the aerofoil. In aerofoil optimisation, an algorithm — the *search engine* — is used to determine the necessary perturbations to the aerofoil geometry in order to reduce the desired forces. These forces are usually expressed as scalar coefficients. For an aerofoil at a given operating condition with an angle of attack α, the lift C_l, the moment C_m and the drag C_d are derived as:

$$[C_l,\ C_m,\ C_d] = f\left(\alpha,\ Re,\ Ma\right) \tag{6.1}$$

Here, Re is the Reynolds number, which is the ratio of the inertial forces to the viscous forces of the medium. Ma is the Mach number, which is a measure of the air velocity against the speed of sound. An aircraft travelling at the speed of sound or greater will have $Ma \geq 1.0$. These speeds are usually known as *supersonic* speeds. The aerofoils found on a subsonic aircraft will have $Ma < 1.0$. We are interested in optimising such subsonic aerofoils.

There are several methods that are used to compute these coefficients. They vary in accuracy and computational expense. The computational tool used to extract these coefficients is referred as the *computational flow solver*. The flow solver employed in this thesis (described in Section 6.4) follows the work by Drela and Giles [1987]. Depending on the flight

condition, one of these force coefficients (or a combination of forces) is generally treated as the objective function, while providing some constraints on the other values [Khurana et al., 2009b]. For example, in the cruise state an aerofoil is designed in an attempt to reduce the drag coefficient at the required lift coefficient to maintain steady flight for fuel economy. However, during take-off, emphasis is placed on maximising the lift coefficient [Quagliarella and Vicini, 2001].

A good aerofoil design is one that exhibits optimal aerodynamic performance over a range of flight conditions and mission segments. The traditional methods of developing aerofoils for single mission requirements have resulted in a large number of UAVs. These suffer from many difficulties in operation and support, because they have been optimised for only one mission or flight condition [Khurana et al., 2009a]. Therefore, a multi-mission design process is regarded as a viable design concept to address the issues with the current UAVs. The current UAV design processes are plagued with performance restrictions and operations outside the intended designs are not permissible. For example, long endurance designs are optimised for slow speed operations at low Re values. Having one of those designs to perform high g-force manoeuvres would result in sub-optimal performances [Khurana et al., 2009a]. Therefore, the current design trend in the aerospace design focuses on developing UAV aerofoils that exhibit optimal performances in multi-mission scenarios.

There have been many gradient based methods applied for single-mission (or single-objective) aerofoil designs [Anderson and Venkatakrishnan, 1997; Hicks and Henne, 1978; Jameson and Reuther, 1994]. However, these approaches may cause the search process to converge prematurely due to the multi-modal search-space [Khurana et al., 2009b]. Therefore, EAs have become popular in single-objective aerofoil design methodologies because they are less prone to get stuck in local optimum solutions [Keane and Nair, 2005]. Among these many EAs, PSOs have been used extensively in shape optimisation problems, which also include aerofoils [Chandrashekarappa and Duvigneau, 2007; Duvigneau et al., 2006; Fourie and Groenwold, 2002; Venter and Sobieszczanski-Sobieski, 2004]. There also have been several examples of multi-objective aerofoil designs in the literature based on PSO algorithms [Khurana et al., 2009a;b]. However in this thesis, we explore possibilities of employing user-preference EMO algorithms in many-objective aerofoil designs. In this process, a DM will provide reference points derived from existing aerofoil designs to guide the MDEPSO algorithm in the multi-modal many-objective search-space to obtain better aerofoils that give optimal performance in many operating conditions. First, we present the aerofoil shape parameterisation methodology employed in our optimisation framework.

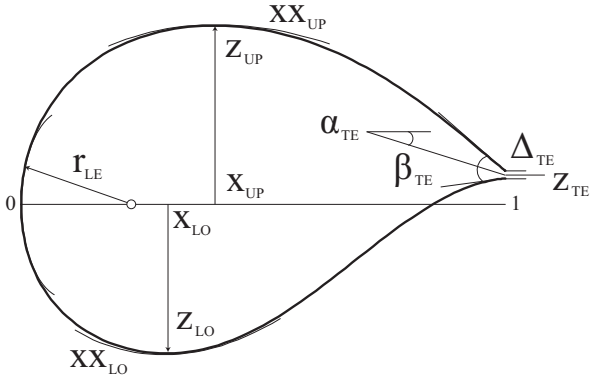


Figure 6.3: PARSEC method for aerofoil parameterisation. The variables: Leading edge radius (r_{LE}); Upper and lower crest locations ($x_{UP}, z_{UP}, x_{LO}, z_{LO}$) and their curvatures ($xx_{UP}, xx_{LO}$); Trailing edge coordinate (z_{TE}) and its thickness (Δ_{TE}); Trailing edge direction (α_{TE}); Wedge angle (β_{TE}).

6.2 Aerofoil shape parameterisation

The shape of an aerofoil can be represented using a variety of parameterisation methods [Samareh, 1999; Song and Keane, 2004]. The selection of a suitable parameterisation scheme is an important contributing factor to the efficiency of the algorithm since it will define the design-space (search-space) [Samareh, 1999; Keane and Nair, 2005; Song and Keane, 2004]. Furthermore, certain parameterisation techniques are suited to specific forms of optimisation. Therefore, the development of efficient parameterisation models has been given significant attention, to increase the flexibility of geometrical control with a minimum number of design variables.

6.2.1 The PARSEC Method

A popular method for aerofoil shape parameterisation is the PARSEC method [Sobieczky, 1998]. It has the advantage of strict control over important aerodynamic features, and also allows independent control over the aerofoil geometry for imposing shape constraints.

Illustrated in Figure 6.3 are the basic eleven parameters that are used to completely define an aerofoil. The parameters include the leading edge radius (r_{LE}), the upper and lower crest locations ($x_{UP}, z_{UP}, x_{LO}, z_{LO}$) and their curvatures ($xx_{UP}, xx_{LO}$), trailing edge coordinate (z_{TE}) and its thickness (Δ_{TE}), trailing edge direction (α_{TE}) and the wedge angle (β_{TE}). In this thesis, the variable $\Delta_{TE} = 0.0$. Thus, blunt trailing edge sections are not considered and the number of search-space (decision-space) dimensions is reduced to ten.

Table 6.1: The decision variables used in the PARSEC method to define an aerofoil and their UBs and LBs.

Variable x_i	$x_i^{(LB)}$	$x_i^{(UB)}$
$x_1 : r_{LE}$	0.0055	0.0215
$x_2 : \alpha_{TE}$	-0.3580	0.0230
$x_3 : \beta_{TE}$	0.0200	0.2600
$x_4 : x_{UP}$	0.2875	0.5345
$x_5 : z_{UP}$	0.0880	0.1195
$x_6 : xx_{UP}$	-1.0300	-0.4200
$x_7 : x_{LO}$	0.3060	0.5075
$x_8 : z_{LO}$	-0.0650	-0.0500
$x_9 : xx_{LO}$	-0.0490	0.8205
$x_{10} : z_{TE}$	-0.0200	0.0200
$x_{11} : \Delta_{TE}$	0.0000	0.0000

The optimiser perturbs the geometrical variables to generate different aerofoil shapes, based on the performance rating provided by the flow solver.

6.2.2 Variable boundaries

Providing feasible boundaries for the PARSEC parameters is yet another contributing factor to the increasing efficiency of the optimisation architecture. A certain class of aerofoils do not provide favourable aerodynamic performance under all flight conditions. With knowledge of the occurring flow, the PARSEC parameters may be restricted to conform to a specific family of aerofoils. This is achieved through inverse mapping of benchmark profiles that have been developed, either by experimental or computational methods, to perform favourably in low-speed flow conditions [Somers, 1981]. These include representative profiles of the Natural Laminar Flow (NLF) series and the National Advisory Committee for Aeronautics/National Aeronautics and Space Administration (NACA/NASA) series [UIUC, 2010]. A *laminar flow* — also known as a streamline flow — is a flow of fluid in parallel layers where there are no disturbances between the layers [Keane and Nair, 2005].

Defining the aerofoil boundaries through inverse mapping as opposed to arbitrarily selecting boundaries is advantageous to omit poorly performing areas of the search-space. The UB and LB values for the parameter boundaries are shown in Table 6.1.

6.3 Objective functions

In this study, the main emphasis was placed on the ability to locate feasible solutions in a many-objective setting. Therefore, six objective functions have been formulated for this optimisation problem. These objectives relate to various operating conditions and requirements. The ten PARSEC parameters constitute the decision variables. The lift constraints are satisfied explicitly, by letting the flow solver determine the angle of incidence that generates the desired lift. Analogously, geometrical constraints are inherent within the defined PARSEC variable boundaries. The formulated objectives cover a range of operating conditions, which would satisfy the typical requirements of a multi-mission UAV. For this application, we formulated all the objectives to be minimised.

Objective f_1

It is essential that the aerofoil exhibits low drag values during cruise flight. Therefore, for the first objective, minimisation of the drag coefficient at an operating lift coefficient $C_l = 0.5$ was required. Re and Ma were fixed at 4×10^6 and 0.3 respectively. This objective is represented as:

$$f_1 = C_d \text{ where } C_l = 0.5,\ Re = 4 \times 10^6,\ Ma = 0.3 \tag{6.2}$$

Objective f_2

It is of equal importance that the aerofoil obtains a high maximum lift-to-drag ratio for ascent, as well as increased flight endurance. Maximum endurance is a prerequisite for UAVs which are expected to perform automated missions for an extensive period of time. For each candidate solution, the incidence angle is floated to determine the minimum drag-to-lift ratio. Therefore, the lift-to-drag objective is given as:

$$f_2 = \frac{C_d}{C_l^{\frac{3}{2}}} \text{ where } \ Re = 4 \times 10^6,\ Ma = 0.3 \tag{6.3}$$

Objective f_3

Providing a reduced drag design during cruise conditions generally occurs at the expense of a highly aft cambered aerofoil section which results in excessive pitching moments. Formulating the objective to minimise the zero-lift pitching moment coefficient is desirable for stability

and control. The incidence angle which corresponds to zero-lift generation (α_0) is determined and the pitching moment at this point C_{m_0} is recorded. Therefore, the objective to reduce the drag is:

$$f_3 = C_{m_0}^2 \text{ where } \ Re = 4 \times 10^6, \ Ma = 0.3 \tag{6.4}$$

Objective f_4

The preceding objectives are deemed as sufficient to address optimal cruise performance. However, the UAVs are also required to perform manoeuvres at cruise without the risk of stall. The objective to maximise the highest possible lift coefficient $C_{l_{\max}}$ before stall occurs is given as:

$$f_4 = \frac{1}{C_{l_{\max}}^2} \text{ where } \ Re = 4 \times 10^6, \ Ma = 0.3 \tag{6.5}$$

Objective f_5

During the descent and approach of a flight, a high lift value is beneficial as it contributes towards an increased lift-induced drag, which is essential for landing. Here the angle of incidence is fixed at 5°, which is regarded as a typical incidence angle during the approach of a flight. Thus, the objective for high lift is given as:

$$f_5 = \frac{1}{C_l^2} \text{ where } \alpha = 5^\circ, \ Re = 2 \times 10^6, \ Ma = 0.15 \tag{6.6}$$

Objective f_6

Providing optimal performance in the approach condition should not be at the expense of massive flow separation or leading-edge boundary layer transitions (x_{tr}). In this case, it was sought to maintain a smooth flow-field during approach flight by maximising the laminar portion of the upper surface of the aerofoil. Therefore, the objective to obtain a maximum boundary layer is given as:

$$f_6 = \frac{1}{x_{tr}} \text{ where } \alpha = 5^\circ, \ Re = 2 \times 10^6, \ Ma = 0.15 \tag{6.7}$$

Table 6.2: These aspiration values for each objective of the NLF0416 aerofoil are used to define the reference point $\vec{\overline{z}}$.

Objective f_i	**Aspiration value $\overline{z}_i$**
f_1	0.0052
f_2	0.0061
f_3	0.0098
f_4	0.3081
f_5	0.9231
f_6	0.6546

With these formulated objectives, mission phases such as cruise (or endurance), approach (or descent), and manoeuvring have been addressed. Further objectives can be formulated which consider take-off or multi-point cruise, but they are beyond the scope of this thesis.

The reference point

To assist the MDEPSO algorithm in locating solutions in this many-objective problem, we define a reference point to guide the particle swarm. The reference aerofoil selected for this application to derive a reference point was the NLF0416 aerofoil [Somers, 1981]. It is considered as a benchmark profile for low-speed applications because experimental data is available. The NLF0416 aerofoil was designed for a moderate lift coefficient in cruise conditions. It has a relatively high upper-surface curvature to maintain favourable pressure gradients for laminar flow control, while providing increased maximum lift values for manoeuvring due to the highly cambered aft section. The NLF family is the product of years of prior research in subsonic aerofoil aerodynamics. The aspiration values derived on the NLF0416 aerofoil for the objective functions we defined are given in Table 6.2.

6.4 The computational flow solver

We have selected XFOIL [Drela and Giles, 1987] as the computational flow solver for the optimisation process. XFOIL is a viscous-inviscid iterative software. The inviscid pressure distribution is modelled using a linear vortex strength distribution. Viscous effects and the development of the laminar-turbulent boundary layers are modelled using the empirical integral boundary layer theory [Schlichting and Gersten, 2002]. XFOIL provides relatively accurate results for subsonic aerofoil analysis. Such software are predominantly used in preliminary design processes as precursors to detailed modelling and wind-tunnel simulations.

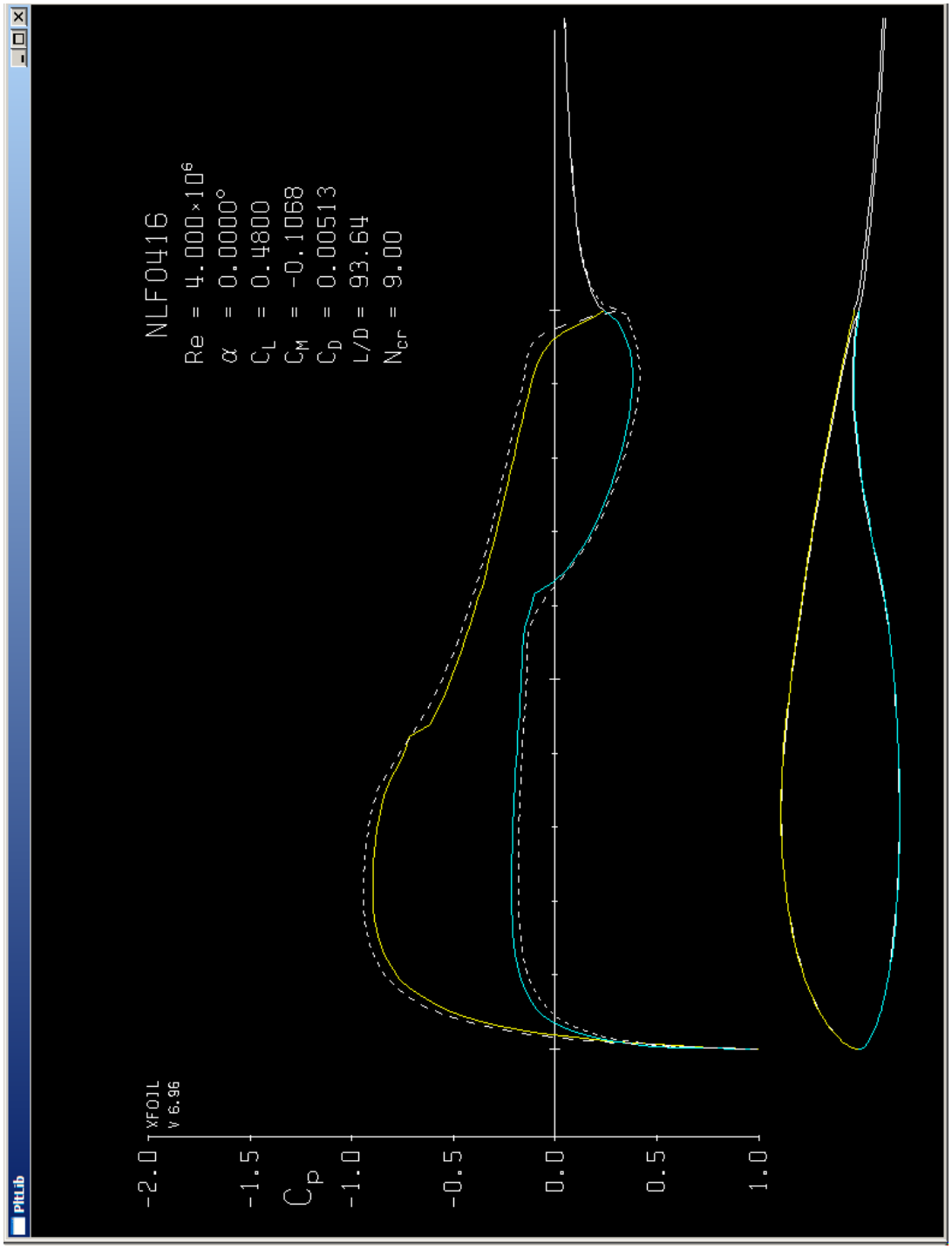


Figure 6.4: NLF0416 in the XFOIL environment. The aerofoil is seen in the bottom, while the pressure differences are illustrated above. C_p is the pressure coefficient at each point of the surface of the aerofoil.

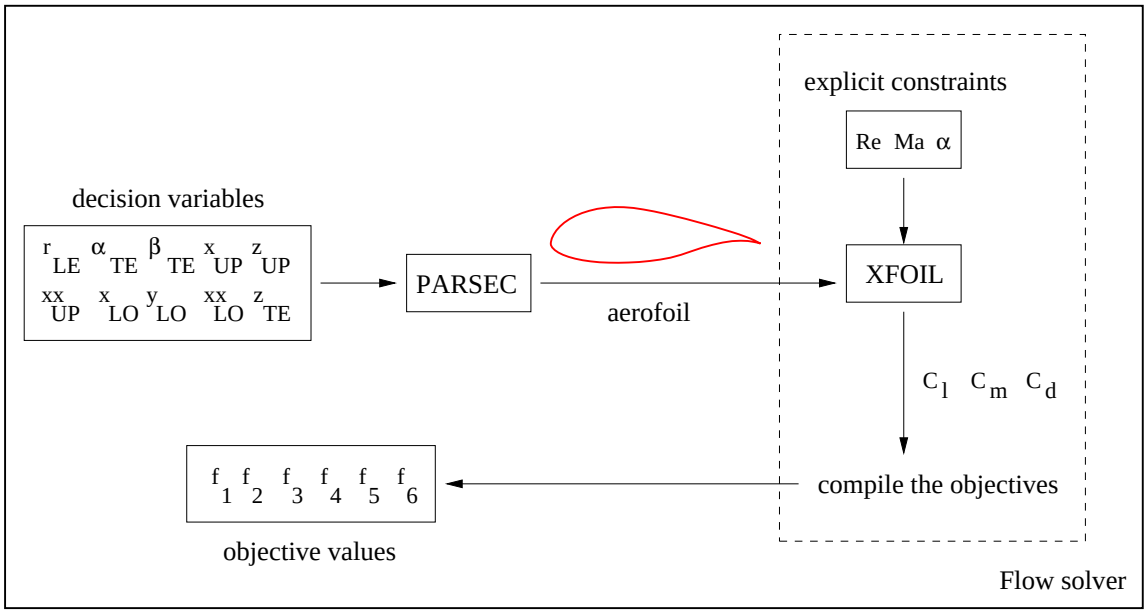


Figure 6.5: The framework for evaluating aerofoils. First, the ten PARSEC decision variables are used to define an aerofoil shape. This aerofoil is analysed in the XFOIL environment by providing the explicit constraints for α, Re and Ma to obtain values for C_d, C_l and C_m. These coefficients are used to calculate the values for the objectives.

Figure 6.4 illustrates NLF0416 aerofoil shape and the pressure coefficient C_p inside the XFOIL environment. We use these pressure coefficients to analyse the optimised aerofoils later in Section 6.6. The figure also indicates the values for C_d, C_l and C_m derived for a given aerofoil shape in a medium defined by values for Re and α. In our optimisation process, we provide α, Re and Ma explicitly to XFOIL for any aerofoil so that the solver provides the C_d, C_l and C_m values required to compose the objectives. Next, we describe this evaluation process within an EMO framework.

6.5 Reference point MDEPSO framework for optimising aerofoils

To optimise the aerofoil designs we use the reference point MDEPSO algorithm (see Algorithm 13 – page 123) described in Section 5.3. The only additional steps required in the reference point MDEPSO algorithm is to create an aerofoil using the ten decision variables and then analyse it using XFOIL to obtain the values for C_d, C_l and C_m to derive the objective values. This process is illustrated in Figure 6.5. However, we present a generalised framework based on the reference point MDEPSO algorithm, for the optimisation process of

any engineering design model in Algorithm 14. We intend to use a similar method for the optimisation of lens designs later in Section 7.1.

Algorithm 14 Reference point MDEPSO framework for optimising engineering designs

1: OBTAIN the reference point $\vec{\bar{z}}$ and spread value δ from the DM
2: INITIALISE a swarm (P_t) of size n, $t = 0$
3: **for** each particle $\vec{x}_i \in P_t$ **do**
4: MODEL the design using the decision variables
5: ANALYSE the design using an external tool
6: OBTAIN the data from the analysis and compose the objective values
7: ASSIGN fitness and distance metric based on the objectives
8: **end for**
9: $t = 1$
10: **repeat**
11: **for** each particle $\vec{x}_i \in P_t$ **do**
12: GENERATE a leader using the *DE/rand/1/bin* scheme
13: UPDATE the velocity using the *Constriction Type 1″* scheme
14: MUTATE the particle
15: UPDATE positions and ADJUST boundary violations
16: MODEL the design using the decision variables
17: ANALYSE the design using an external tool
18: OBTAIN the data from the analysis and compose the objective values
19: ASSIGN fitness and distance metric based on the objectives
20: UPDATE the personal best based on the distance metric
21: **end for**
22: COMBINE the swarm at the beginning of the iteration (P_t) with the one that has updated positions (Q_t) such that $R_t = P_t \cup Q_t$
23: SORT R_t according to Step 10 of Algorithm 11 obtain the next iteration P_{t+1}
24: $t = t + 1$
25: **until** $t = t_{MAX}$

The steps 4–6 and 16–18 in Algorithm 14 represent the evaluation process we illustrate in Figure 6.5. These generalised steps can be used in other engineering design problems, where the evaluation of the candidate models is done using an external program. In the lens design application in Section 7.1 we evaluate our candidate lens designs in a similar manner using an external program. Therefore, this modular framework for optimising and evaluating engineering design problems is scalable for other applications and also for other evolutionary search strategies. Next, we present some results from the aerofoil optimisation process based on the reference point MDEPSO algorithm.

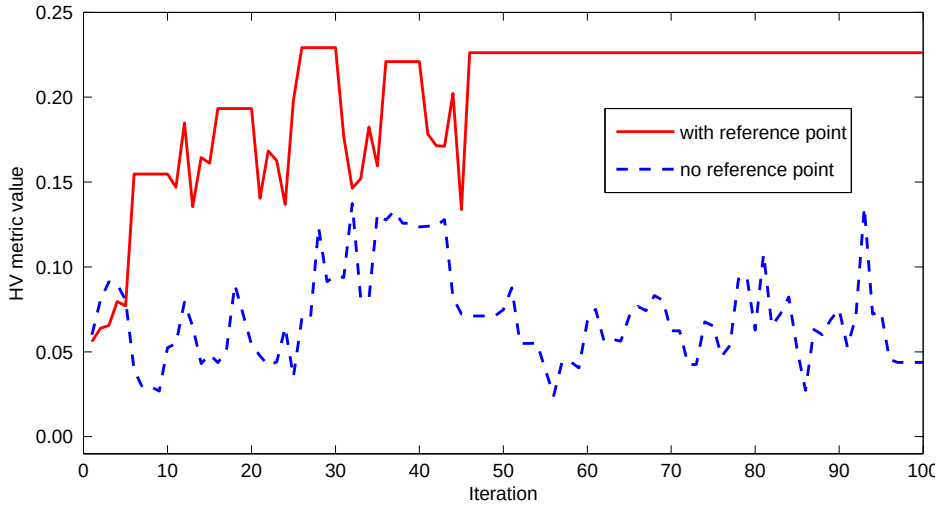


Figure 6.6: The average HV values at each iteration for the reference point and standard MDEPSO algorithms. These HV values are derived from the calculation process described in Section 4.3.

6.6 Experiments

We first analyse the convergence behaviours between the reference point MDEPSO algorithm and the original MDEPSO algorithm that is based on dominance ranking, on the aerofoil application. To this end, we modified the evaluation process of original MDEPSO algorithm in a similar manner to the reference point algorithm described in Algorithm 14. Here, we integrated the evaluation process illustrated in Figure 6.5 into Step 9 of the original MDEPSO algorithm (see Section 5.2 – page 113). Both algorithms used a population size of 100 for 100 iterations. The evaluation process of the individuals was computationally expensive, due to the complex nature of the matrix operations within the analysis process. Therefore, we decided on these parameter values for the EMO algorithm as a trade-off for the heavy time consumption. Initial results indicated that 10,000 evaluations took about 36 hours on a 2.3GHz dual-core CPU machine. We used $CR = 0.2$ and $F = 0.4$ for the DE operator in MDEPSO. We used a spread value of $\delta = 0.05$ for the reference point approach. The results are an average of 10 independent runs.

6.6.1 Convergence and spread of solutions

Figure 6.6 illustrates the average HV values at each iteration of the runs. We used $\delta' = 0.5$ to define a volume in the preferred region. This δ' value was chosen to ensure that at least

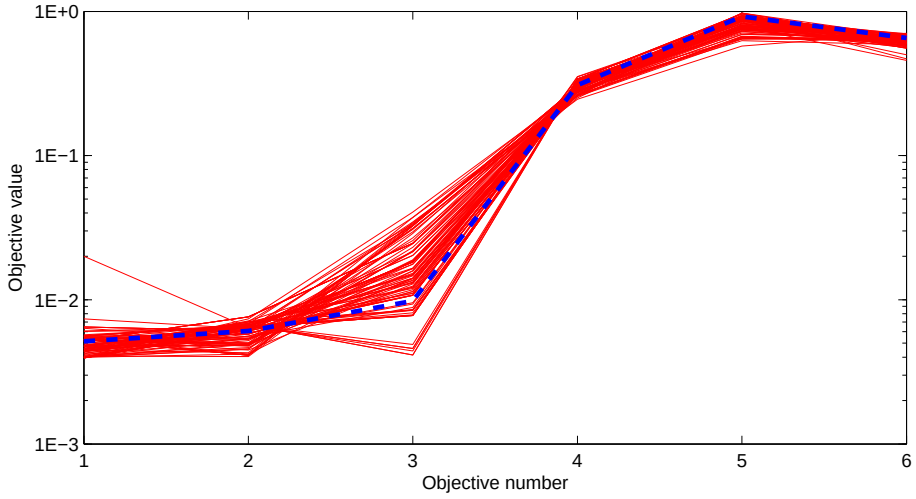
90% of the particles of the algorithm, which has better convergence are selected. Then as described in Section 4.3 (page 95) a nadir point was selected and the HV values at each iteration were calculated. The abrupt changes of the HV values for the reference point based approach are due to the exploration and exploitation process we adopted in the distance metric approach (see Section 3.1.3 – page 56). This is due to the fact that particles might move away from the preferred region to explore the search-space and avoid getting stuck at local optima. It is clear from the results that as the iterations progress the reference point based algorithm converges better than the dominance based algorithm. The results also show that the reference point based algorithm has converged to an optimum at about iteration 50, while the dominance based algorithm fails to converge even after the maximum number of iterations have been reached. These results show that the reference point based MDEPSO algorithm performs much better in a six-objective problem instance as opposed to a standard non-dominance sorting based approach.

Figure 6.7 shows the solution points obtained from the best converged runs of each algorithm. The best convergence was deduced from the run that had the largest HV value in the final iteration. Here, the reference point is given as the *dashed* line, while the solid lines represent the candidate solutions. It is clear that the reference point MDEPSO algorithm converges better in all objectives compared to the original MDEPSO algorithm. We have shown many such results, where the distance metric based algorithms are better than dominance based algorithms in many-objective problems in Chapters 3, 4 and 5. However, in this chapter, we are more interested in analysing the obtained designs with respect to the actual behaviour of the objectives. Therefore, we next present an analysis of the designs with respect to some operating conditions.

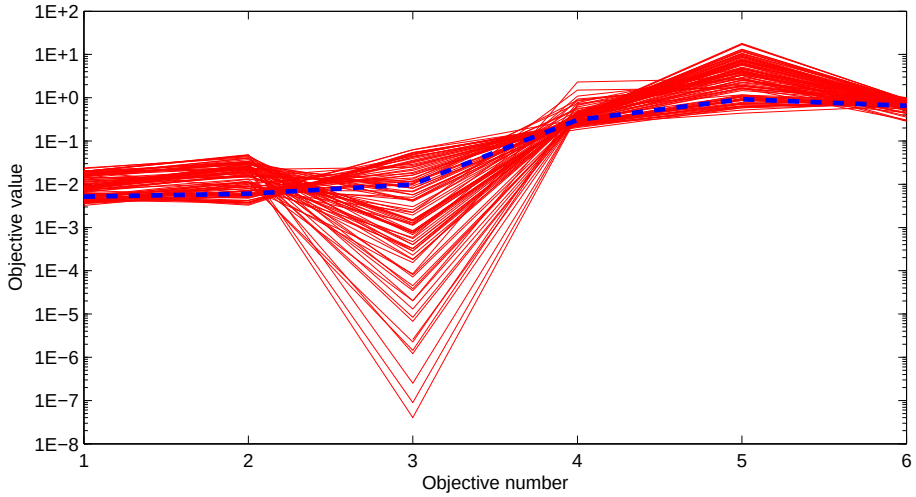
6.6.2 Optimised aerofoil designs

We consider the aerofoil which exhibits the most feasible compromise between all objectives as the *preferred aerofoil*. According to our optimisation method the preferred solution is the one which has the minimum distance metric value. It follows that the preferred design is the closest compromising design to the NLF0416. Therefore, it is the most improved design.

The variations of the aerofoil geometries of the preferred and reference aerofoils are illustrated in Figure 6.8. The preferred solution has a relatively large leading edge radius. This results in an approximately 16.5% thick aerofoil. The upper surface is highly cambered,



(a)



(b)

Figure 6.7: Final solutions obtained on MDEPSO: (a) Using a reference point; (b) Without a reference point. The dashed line represents the reference point.

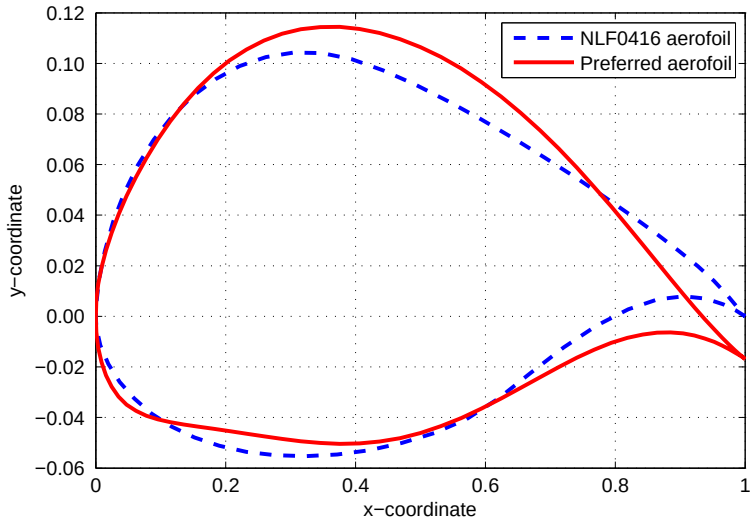


Figure 6.8: Comparison of the preferred aerofoil with the NLF0416 aerofoil. The preferred aerofoil is the one which has the minimum distance metric value. This aerofoil gives the most feasible compromise between all the objectives in consideration.

prolonging laminar flow during cruise. The lower surface is quite flat, which induces a lower moment value and is beneficial for stability. The aerofoil has a highly cambered aft section for lift generation. The trailing edge is drooped and the wedge angle is significantly small. The drooped trailing edge is essential to limit separation of the boundary layer, which results in lower drag values and less chaotic flow when generating increased lift. Further investigation has proven that the trailing edge wedge geometry may be impractical to manufacture, because of a *fish tail* shape. Although this aberration is very small, it can be an issue in the actual manufacturing process. Therefore, we require additional constraints to be imposed on the design variables to restrict such geometries. We have not considered manufacturing constraints in this thesis, because our main aim was to test the applicability of our algorithms on real-world applications. However, we believe that for a complete optimisation and manufacturing process these additional constraints can be imposed in future work extending the research in this thesis. Next, we analyse the preferred solution in each operating condition with respect to the pressure distributions and lift coefficients.

Figure 6.9 illustrates the pressure coefficient distribution of the preferred aerofoil against the reference NLF0416 aerofoil for the cruise operating condition (objective function f_1). In these pressure coefficient diagrams the curve which predominantly has a negative pressure is the upper surface of the aerofoil. Similarly, the lower surface is denoted by the curve which is

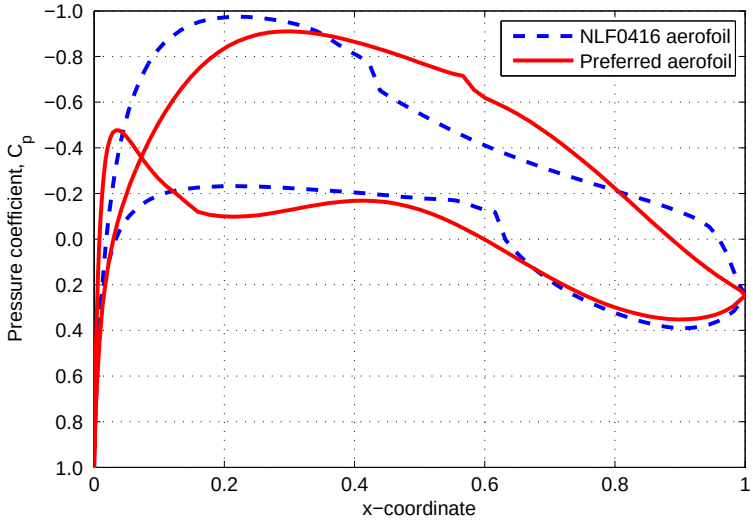


Figure 6.9: Pressure curves of the preferred aerofoil and the NLF0416 aerofoil for objective f_1. Here, $C_l = 0.5$, $Re = 4 \times 10^6$ and $Ma = 0.3$.

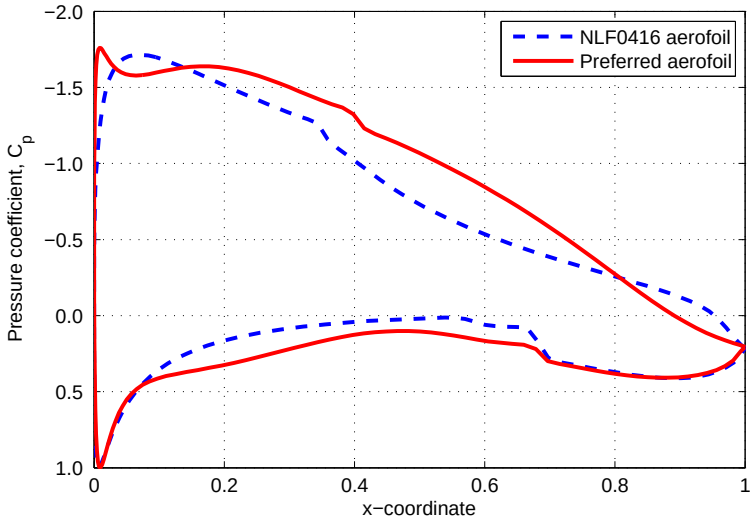


Figure 6.10: Pressure curves of the preferred aerofoil and the NLF0416 aerofoil for objective f_5. Here, $\alpha = 5°$, $Re = 2 \times 10^6$ and $Ma = 0.15$.

predominantly experiencing positive pressure. The difference between these two curves yields the resultant force. In this example it is the lift, because the lower surface of the aerofoil has higher air pressure than the upper surface.

Figure 6.10 illustrates a similar pressure curve to Figure 6.9, which is referring to the objective function f_5. Here, the pressure differences between the highest and lowest points

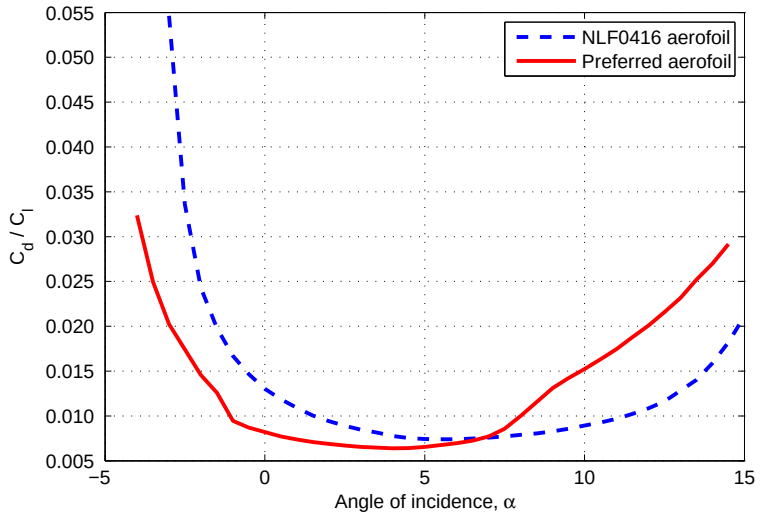


Figure 6.11: Drag to lift ratio polar of the preferred aerofoil and the NLF0416 aerofoil. Here, $Re = 2 \times 10^6$ *and* $Ma = 0.15$.

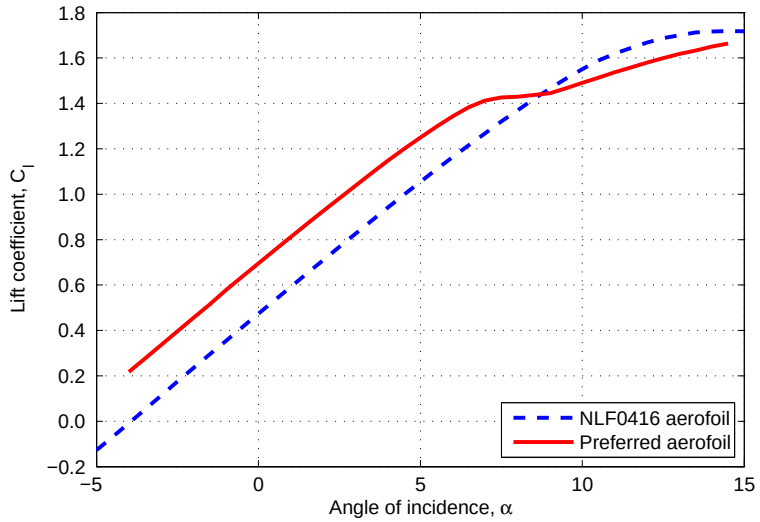


Figure 6.12: Lift coefficient curve of the preferred aerofoil and the NLF0416 aerofoil. Here, $Re = 2 \times 10^6$ *and* $Ma = 0.15$.

of the x-coordinates are on average greater on the preferred aerofoil compared with the NLF0416 aerofoil. This indicates that for the given operating condition — the decent and approach phase — a greater lift is achieved on the aerofoil. This shows that the preferred aerofoil performs better on this phase of flight than the reference NLF0416 aerofoil.

*Table 6.3: The objective values of the solutions which provide the best values for each operating consideration. The preferred aerofoil gives the best trade-off for each conflicting objective. The objective value of the best aerofoil for each operating condition is also presented for comparison. The values in **bold** are the best values for each corresponding objective.*

Aerofoil	f_1 **value**	f_2 **value**	f_3 **value**	f_4 **value**	f_5 **value**	f_6 **value**
NLF0416 aerofoil	0.0052	0.0061	0.0098	0.3081	0.9231	0.6546
Preferred aerofoil	0.0054	0.0054	0.0128	0.3143	0.6729	0.5953
Best aerofoil for f_1	**0.0041**	0.0073	0.0171	0.2874	0.9192	0.6417
Best aerofoil for f_2	0.0041	**0.0040**	0.0288	0.3002	0.9716	0.6261
Best aerofoil for f_3	0.0056	0.0067	**0.0041**	0.3387	0.9610	0.5738
Best aerofoil for f_4	0.0041	0.0047	0.0398	**0.2398**	0.8873	0.7010
Best aerofoil for f_5	0.0060	0.0052	0.0197	0.2761	**0.5864**	0.6436
Best aerofoil for f_6	0.0054	0.0048	0.0115	0.3236	0.9344	**0.4525**

Table 6.4: Improvement over the NLF0416 aerofoil for the respective objectives considering all the solutions obtained from the reference point MDEPSO algorithm.

Aerofoil	**Percentage of improvement for f_i of NLF0416**
Best aerofoil for f_1	21.1%
Best aerofoil for f_2	33.3%
Best aerofoil for f_3	57.9%
Best aerofoil for f_4	22.1%
Best aerofoil for f_5	36.5%
Best aerofoil for f_6	30.9%

Figures 6.11 and 6.12 illustrate the drag polar and lift curves respectively. The drag polar demonstrates that for incidence angles up to $\alpha = 6^\circ$, the preferred aerofoil exhibits greater aerodynamic efficiency than the NLF0416 aerofoil. This attribute is inferred because the drag-to-lift ratio — corresponding to objectives f_2 and f_5 — is lower on the preferred aerofoil than of the reference aerofoil. The lift curve (Figure 6.12) of the preferred aerofoil shows a consistently greater lift which achieved for values up to $\alpha \approx 7^\circ$, after which boundary layer separation on the upper surface results in lower lift values. This phenomenon is validated by objective f_4, which indicates a lower maximum lift (or higher inverse) is achieved compared with the NLF0416 aerofoil. These results clearly show that the preferred aerofoil obtained from the optimisation process has better aerodynamic characteristics compared to the reference NLF0416 aerofoil.

The objective values of the preferred aerofoil and the aerofoils which exhibit the optimal values for each respective objective are given in Table 6.3. We observe a marginal increase in the objectives f_1 and f_4 compared to the NLF0416, and an improvement of approximately 13%, 27% and 9% is obtained for the preferred solution for objectives f_2, f_5 and f_6 respectively. However, despite improvements in these objectives, the NLF0416 aerofoil still exhibits greater stability characteristics with a 30% lower value for objective f_3 against the preferred aerofoil. Table 6.3 also demonstrates the improvement over the NLF0416 aerofoil for solutions which give the minimum values for each corresponding objective. These results are beneficial if the DM was inclined towards a particular objective, rather than a feasible compromise. It is observed that there is one solution which improves on the NLF0416 for at least one objective at a time. The percentage improvements for the respective objectives are shown in Table 6.4. We have illustrated the shapes of the aerofoils that exhibit the best objective values for each objective function with comparison to the NLF0416 aerofoil in Figure 6.13.

In this study, we have found that using a reference point EMO algorithm to optimise such a computationally expensive problem has more merit than simply being able to find a set of the best solutions. The process is also able to find the best solutions for each objective with the one execution process, if the DM is interested in such solutions. We were able to achieve such a set of solutions by using outranking in the optimisation process. If one were to use a traditional MCDM optimisation approach only a single solution would be available at the end of the execution run. Repeating the execution to obtain more solutions would be inefficient, especially for computationally expensive problems. Therefore, we conclude that this user-preference EMO optimisation process of aerofoils is very effective and efficient when compared with traditional mathematical approaches. We also found that better aerofoil designs than the reference aerofoil designs are obtained in fewer function evaluations saving valuable computational resources.

6.7 Summary

We defined the aerofoil design application to consist of six objectives that correspond to six operating conditions. Each aerofoil was represented using a prameterised scheme known as the PARSEC method. This method consisted of 11 decision variables, but we only used 10 of them. One decision variable was set to 0.0 so that blunt tail edges were not generated

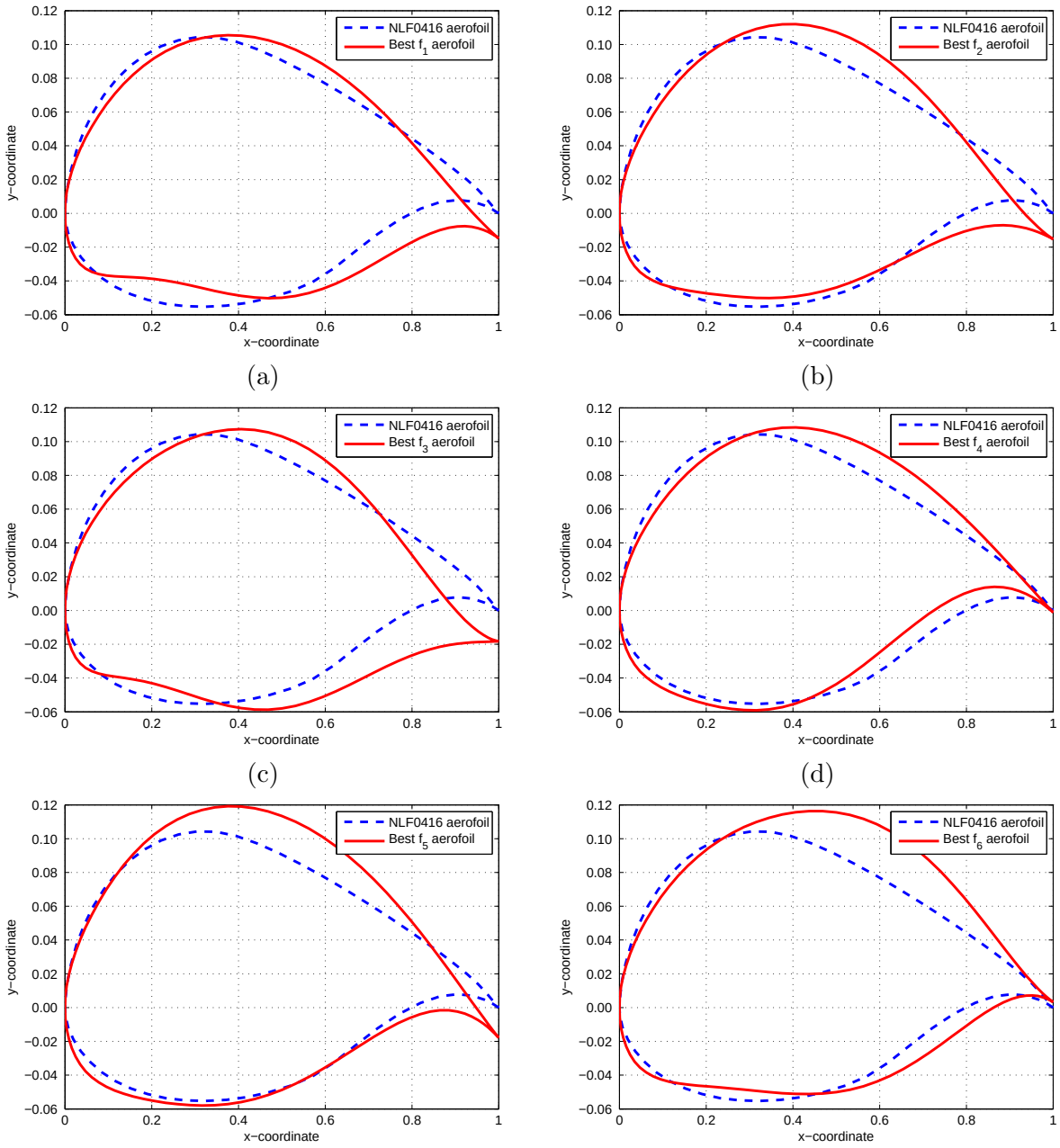


Figure 6.13: Best aerofoil obtained for each objective function compared with the NLF0416 aerofoil: (a) Objective f_1; (b) Objective f_2; (c) Objective f_3; (d) Objective f_4; (e) Objective f_5; (f) Objective f_6.

in the optimisation process. The boundary values for each of the decision variables were set using existing data from the domain. We extended the reference point MDEPSO algorithm introduced in Chapter 5, to include a module for evaluating the designs. This extension was generalised as a framework so that any industrial design application can be included in the MDEPSO algorithm. The main requirement of these applications was to have the generated designs evaluated using an external flow solver. The external program used in this application was XFOIL. We derived a reference point using the NLF0416 reference aerofoil. This aerofoil was chosen because of existing experimental data, which we later used in the analysis of the generated aerofoils.

First we compared the convergence ability of the reference point MDEPSO algorithm with the standard MDEPSO algorithm on the many-objective problem aerofoil optimisation problem. Our aim in this comparison was to observe any unique behaviour of the EMO algorithms, which were not seen in the other test problem instances. We compared the two algorithms using the HV calculation process we introduced in Section 4.3 (page 95). The results showed that the reference point MDEPSO algorithm was able to converge better than the dominance comparison MDEPSO algorithm in the aerofoil application. This output was consistent with the results we presented in Chapters 3, 4 and 5.

In our analysis, we named the candidate design having the lowest distance metric value as the *preferred design*. This preferred aerofoil design was analysed with respect to various aspects of the flying conditions represented by the objectives. We also found that the best designs for each objective can be obtained from one optimisation process, in addition to the preferred design. This proved to be very useful and efficient if the DM was also interested in such aerofoil designs. In a computationally expensive application like aerofoil design, these solutions were very valuable for designing further experiments. We also observed that better aerofoil designs, compared to the reference aerofoil design can be obtained in less number of evaluations. This saves computational resources making our user-preference EMO optimisation process more efficient than existing methods.

In the next chapter, we present the second real-world application, which is on optimising lens designs. We present the concluding remarks including both applications in the next chapter with respect to the research question: *How can real-world problems benefit from user-preference EMO algorithms in the optimisation process of engineering designs?*

Chapter 7

Real-world application: Lens design

We introduce a lens design application in a similar manner as the aerofoil design application covering some of the basic background information, existing optimisation methods and the problem description. First, in Section 7.1, we present some basic background information about lens design and existing optimising methods. These optimisation methods include both numerical and evolutionary approaches. Similar to the aerofoil application, we use an external evaluator for this application. This external tool is presented in Section 7.2. Unlike in the aerofoil application, this tool is used to analyse the lens design and also model it.

Unlike the aerofoil parameterisation scheme, the number of decision variables for the lens design application is not constant for different problem instances. We present a scheme that defines the required decision variables for any a lens design application in Section 7.3. This scheme will be used later to define the problem instance we are interested in. For this problem instance, we define five objective functions in Section 7.4. Similar to the aerofoil application, here also not all objectives conflict with each other at the same time. These objectives are used to derive best quality lenses that can be used for different scenarios in still photography. We derive a reference point in a similar manner to the aerofoil application using an existing lens design. However, the components of this reference lens design are not identical to the problem instance discussed. We use this reference lens design — which is already an optimised design — to provide some guidance for the reference point MDEPSO algorithm to obtain new lens designs suitable for various scenarios. In Section 7.5, we present this reference lens design and define the problem instance we are interested in. Next in Section 7.6, we present the experimental results and analysis of the lens designs that are obtained from the optimisation process.

Our main motivation for this lens design problem is to explore the possibility of using an existing reference lens design to obtain new lens designs suitable for various requirements. In the aerofoil application, we attempted to further improve the reference aerofoil design such that it can be used successfully in different operating conditions. This approach to obtain new designs using the reference design than further optimising the reference design is the major difference between the two real-world problem instances.

7.1 Lens designs for still photography

Designing lenses for still photography has been a field that gathered much interest for many decades [Kidger, 2004]. One of the main issues in this domain is the complexity of the design process, which involves a mixture of existing domain knowledge and suitable optimisation procedures [Vasiljevic, 2002]. Similar to the aerofoil design application we incorporate the refined knowledge of a DM as user-preferences to guide the MDEPSO algorithm to locate useful solutions in the lens application. The traditional approach of lens design is similar to aerofoil design, where an initial design is developed and optimised iteratively using various mathematical methods. There also have been some attempts of using EAs in the optimisation process of lens design.

A survey of such EA approaches in lens design was presented by Thibault et al. [2005]. One of the very first applications of EAs in lens design can be seen in the study by Ono et al. [1998]. This study presented two objective functions named *resolution* and *distortion*. Ono et al. [1998] used a GA to optimise these functions and generate some lens designs. In the work by Nagata [2004], a Co-variance Matrix Adaptation Evolution Strategy (CMA-ES) was used to optimise the two objective functions presented by Ono et al. [1998]. This study showed comparison results with the GA based approach by Ono et al. [1998]. In the work by Beaulieu et al. [2002], a GA based algorithm was described for optimising lenses. This approach used a different formulation of the fitness functions than by Ono et al. [1998], which was derived from the characteristics of the lens system. Another approach using GP for lens design was presented by Koza et al. [2005]. This study illustrated an automated process to re-engineer six existing lens designs. The obtained designs were close in features to the existing lens designs. Recently, another ES approach for lens design was presented by Gagné et al. [2008]. This study presented a comparative analysis with the CMA-ES approach. They also obtained designs very close in competitiveness to some existing classical designs.

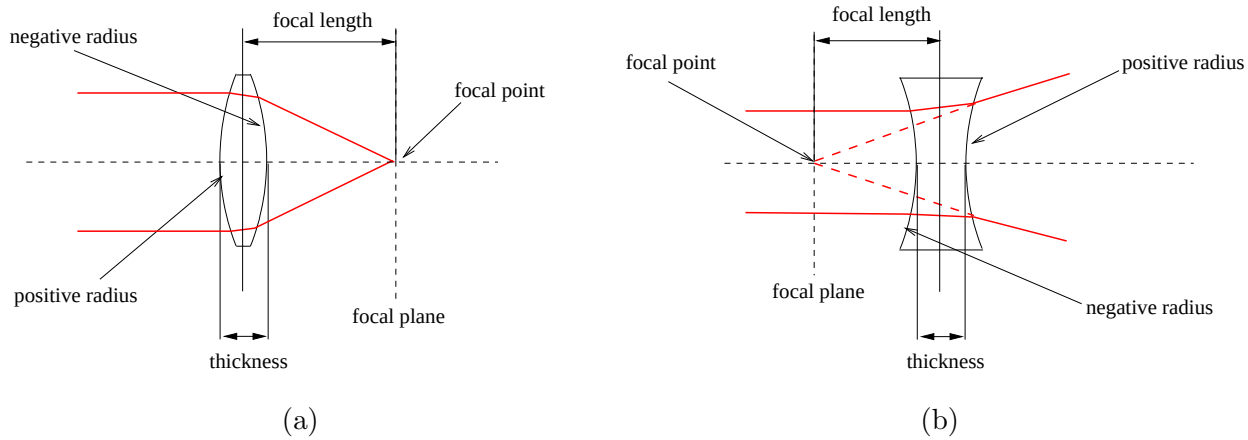


Figure 7.1: Lens designs: (a) Positive (converging) lens; (b) Negative (diverging) lens. The convention used in this thesis reads lens elements from left to right. Reading from left to right, a convex surface has a positive radius, while a concave surface has a negative radius.

Other than these examples, the work presented by Joseph [2005]; Joseph et al. [2006a;b; 2007] introduced several additional objectives extending the work by Ono et al. [1998]. However, these works used an NSGA-II variant in the optimisation process, which we have shown to be inefficient in many-objective problems. Therefore, we are motivated to explore the possibility of using reference points to better explore this many-objective search-space and obtain new or better lens designs. First, we present some background information about lens design and optics.

7.1.1 Lens design

A lens is considered as a symmetric optical device, which is able to converge or diverge a beam of light [Kidger, 2004]. A lens is considered to be symmetric along the horizontal axis, but most lenses are symmetric along all three-dimensional axes. Although lenses can be created using many different materials, we mainly consider lenses made of glass. The design and optimisation process we present here is based on the characteristics of glass. Most lenses found in still photography are made with glass [Bach and Neuroth, 1995].

Figure 7.1 illustrates two simple lens designs. A lens has two *surfaces*. The surface of a lens is considered to be a *slice* of a sphere with a given radius. We measure the *curvature* of a surface as the *inverse of the radius*. As the radius of the sphere increases, the surface of the lens *flattens* out. Therefore, a flat surface is considered to be a slice of a sphere having an infinitely large radius. The convention in lens design is to read the surfaces of the optical

elements from left to right. A convex surface is considered to be a slice of a sphere with a *positive radius*, while a concave surface has a *negative radius*.

The positive lens in Figure 7.1 is also known as a *biconvex* lens because it has two surfaces that are convex, reading from the mediums of glass to air. This lens has the ability to converge a beam of light into a single point. This point is usually referred as the *focal point*. The negative or *biconcave* lens has the ability to diverge a beam of light. This divergence can also be visualised as rays of light emitted from a single point along the horizontal axis. This point is considered as the focal point of the negative lens. The vertical plane intersecting the horizontal axis at the focal point is referred as the *focal plane* and the distance roughly from the middle of the lens to the focal plane is referred as the *focal length*. The focal length of a lens depends on the *thickness*, the *refractive index* of the glass and the curvature of the surfaces of the lens. The thickness is the distance between the surfaces of the lens. The refractive index is a measure of the angle at which a light beam bends as it passes through the glass. For a lens having a thickness of d between the two surfaces defined by R_1 and R_2 radii; and a refractive index of r, positioned in air — having a refractive index $r = 1.0$ — the Effective Focal Length (EFL) is given as:

$$\frac{1}{EFL} = (r-1)\left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(r-1)d}{rR_1R_2}\right] \tag{7.1}$$

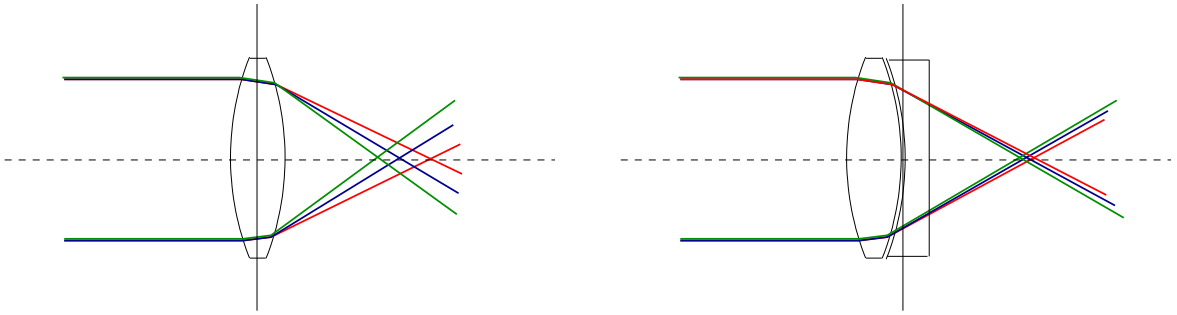
The EFL of a *compound lens* system having two lens elements with EFLs F_1 and F_2 separated by a distance of d in air, can be given as:

$$\frac{1}{EFL} = \frac{1}{F_1} + \frac{1}{F_2} - \frac{d}{F_1F_2} \tag{7.2}$$

The focal length (or EFL) of a lens is also synonymous with its *magnification power*. This indicates that a lens with a larger focal length has a higher magnification. One can use such a lens to get a close-up image of a distant object [Kidger, 2004]. However, creating a single lens element with a large EFL is not possible without considering the many *aberrations* or *optical distortions*. Minimising these aberrations, while obtaining a suitable EFL and other properties are the main challenges in lens optimisation [Vasiljevic, 2002].

7.1.2 Lens optimisation

Glass inherently has a property of being able to split light into its composite colours [Bach and Neuroth, 1995]. This phenomenon is illustrated in Figure 7.2. This basic type of aberration



(a) (b)

Figure 7.2: Light splitting into colours due to refraction: (a) Chromatic aberrations occurring on a lens; (b) Fixing the chromatic aberrations using an achromatic doublet.

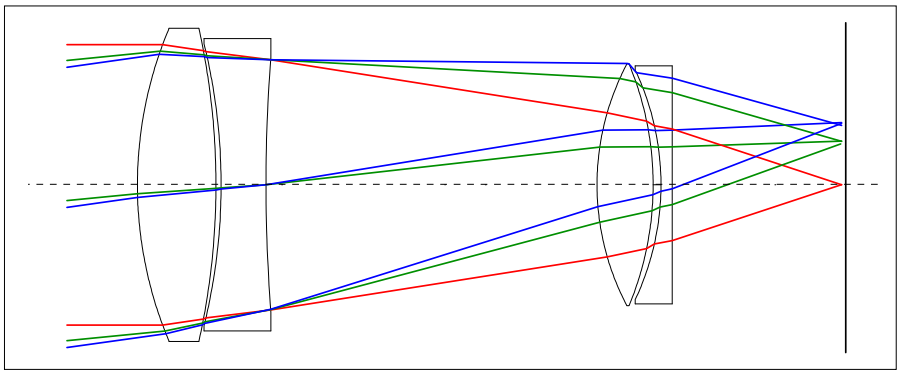


Figure 7.3: A Petzval lens. It is made with four lens elements, which can minimise chromatic aberrations. The light rays illustrate the points of focus on a plane, usually referred as the focal plane.

is known as chromatic aberration. Chromatic aberrations are undesirable because the image obtained from a lens will have colour shifts, distorting the output. Chromatic aberrations can be avoided with the use of several converging and diverging lens elements, as seen in Figure 7.2. A pair of such lens elements is known as an *achromatic doublet*. These compound lenses are one of the key building blocks in lens design. There are many such compound lens designs found in the optical research literature and patents [Kidger, 2004; Vasiljevic, 2002].

Figure 7.3 illustrates one such lens design that is able to minimise CAs. This compound lens having two achromatic doublets is known as a Petzval lens [Optical Research Associates, 2010]. The use of such groups of lens elements is one of the most used lens design techniques found in the industry, especially for *fixed focal* (non-zoom) lenses. We consider only fixed focal

lenses in this thesis, to simplify this proof-of-concept application. This Petzval lens diagram also illustrates the path taken by three rays of light originating far from the first surface though the compound lens towards the focal plane. In lens design application these light rays are considered to be originating from infinity. This process of analysing a lens design using directed light rays is known as *ray-tracing*. The ray-tracing process is an important part of the lens optimisation because the optical properties of a compound lens system can be obtained by simulation rather than building a physical model. Based on these ray-tracing results, further optimisation of the lens system can be performed iteratively until a suitable design is achieved for modelling [Vasiljevic, 2002].

There exist many other aberrations that have to be minimised in a lens optimisation process. We briefly describe some of these aberrations in Figure 7.4. These five aberrations are known as Seidel aberrations or third order aberrations [Bach and Neuroth, 1995; Kidger, 2004; Vasiljevic, 2002].

Spherical aberration

In spherical aberration the light emitted from a single point does not focus on a single point after passing through the lens system. Having the light not focused properly on a single focus point causes the image created to blur or to have a *loss of sharpness*.

Distortion

In distortion, the image created from the lens system will not be as rigid as the original object. This is seen as a set of equally spaced points on a plane vertical to the horizontal axis being focused at unequal distances on the focal plane. If the distances have increased more than the object, it is known as *pincushion distortion*. The negation is known as *barrel distortion*.

Curvature of field

Here although the points are equally spaced, they do not belong to the same focal plane. This causes the original object to have different proportions in the created image.

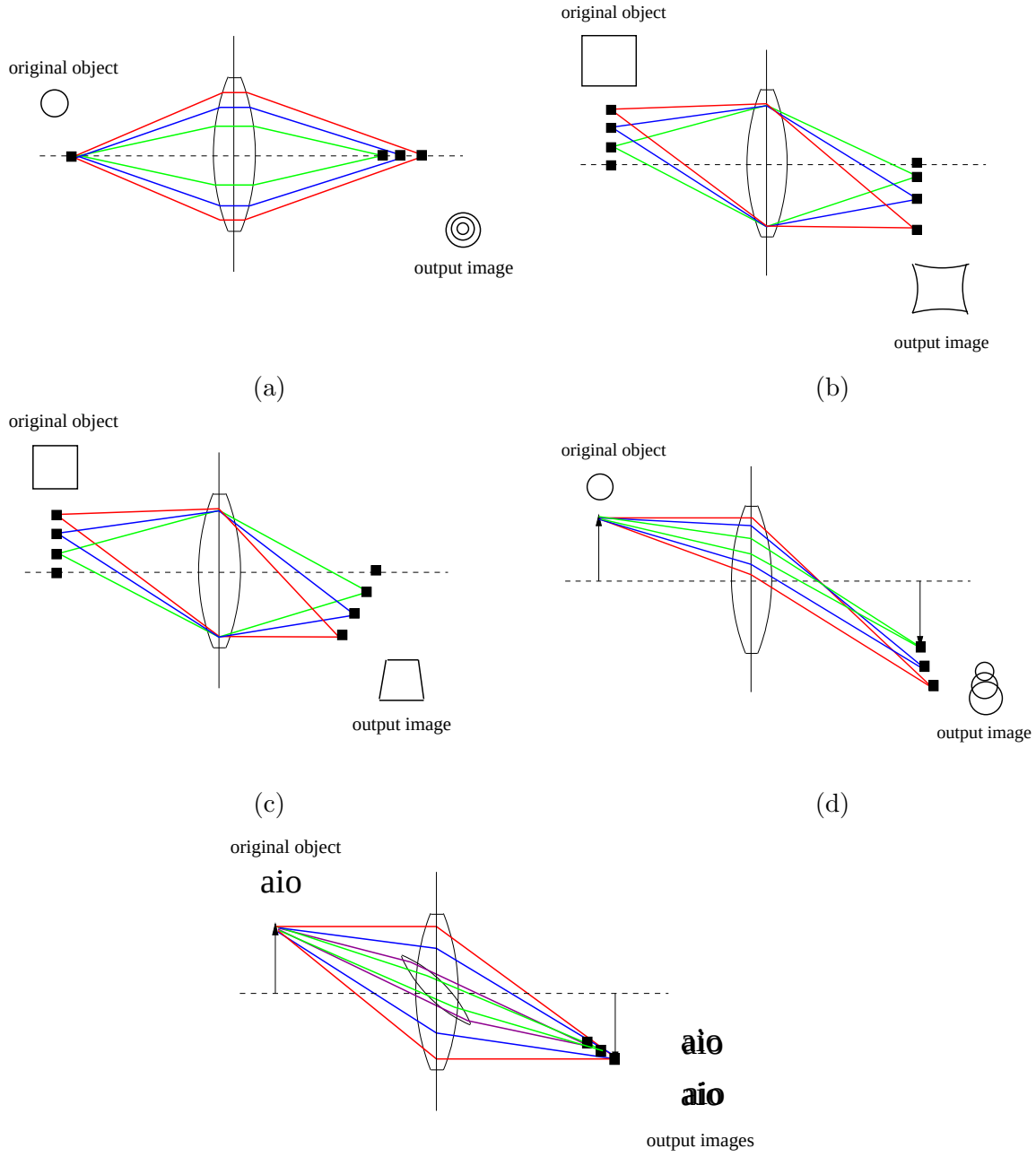


Figure 7.4: The five Seidel aberrations, which are also known as third order aberrations: (a) Spherical aberration; (b) Distortion; (c) Curvature of field; (d) Coma aberration; (e) Astigmatism.

Coma Aberration

Even if the spherical aberrations are eliminated, there is a possibility for the image to have points created on different focal planes. This makes the image have *comet-like flares*. This is known as a Coma Aberration.

Astigmatism

Even if the created images possess no chromatic aberrations or Coma aberrations there can be a *tilt*. This tilt of an image occurs either on the vertical or horizontal axis or even both. This is known as astigmatism. As seen in Figure 7.4(e) an offset is present in the images, especially in characters. This corresponds to a different *profile* or behaviour of a lens within the original lens.

These aberrations can be minimised using compound lenses, while retaining certain useful attributes of the lens system. However, including more glass elements can be undesirable because the increase in elements introduces more obstructions for the light to pass through. The lens system can also become very large and cumbersome to use. Therefore, one has to find a trade-off design to achieve each of these conflicting objectives. Before we describe this many-objective problem, we present some information about the design environment employed in this application.

7.2 Ray-tracing tool

The software tool we used to construct and analyse the candidate lens systems using ray-tracing is CODE V [Optical Research Associates, 2010]. CODE V has been used in many industrial design applications and also contains a library of many patented lens designs. We integrated CODE into our design framework described in Algorithm 14 (page 145) using the its APIs. Optical Research Associates [2010] were kind enough to provide us with an academic license, giving access to an industrial standard design tool.

Figure 7.5 illustrates the Petzval lens (Figure 7.3) in the CODE V design environment. The main windows of interest show the construction of each lens surface, the command window to issue API commands, and the many visualisation tools. One such visualisation tool is the two-dimensional ray-trace output of the lens system. The API provides commands to construct a lens system and extract its attributes after ray-tracing. For the ease of integrat-

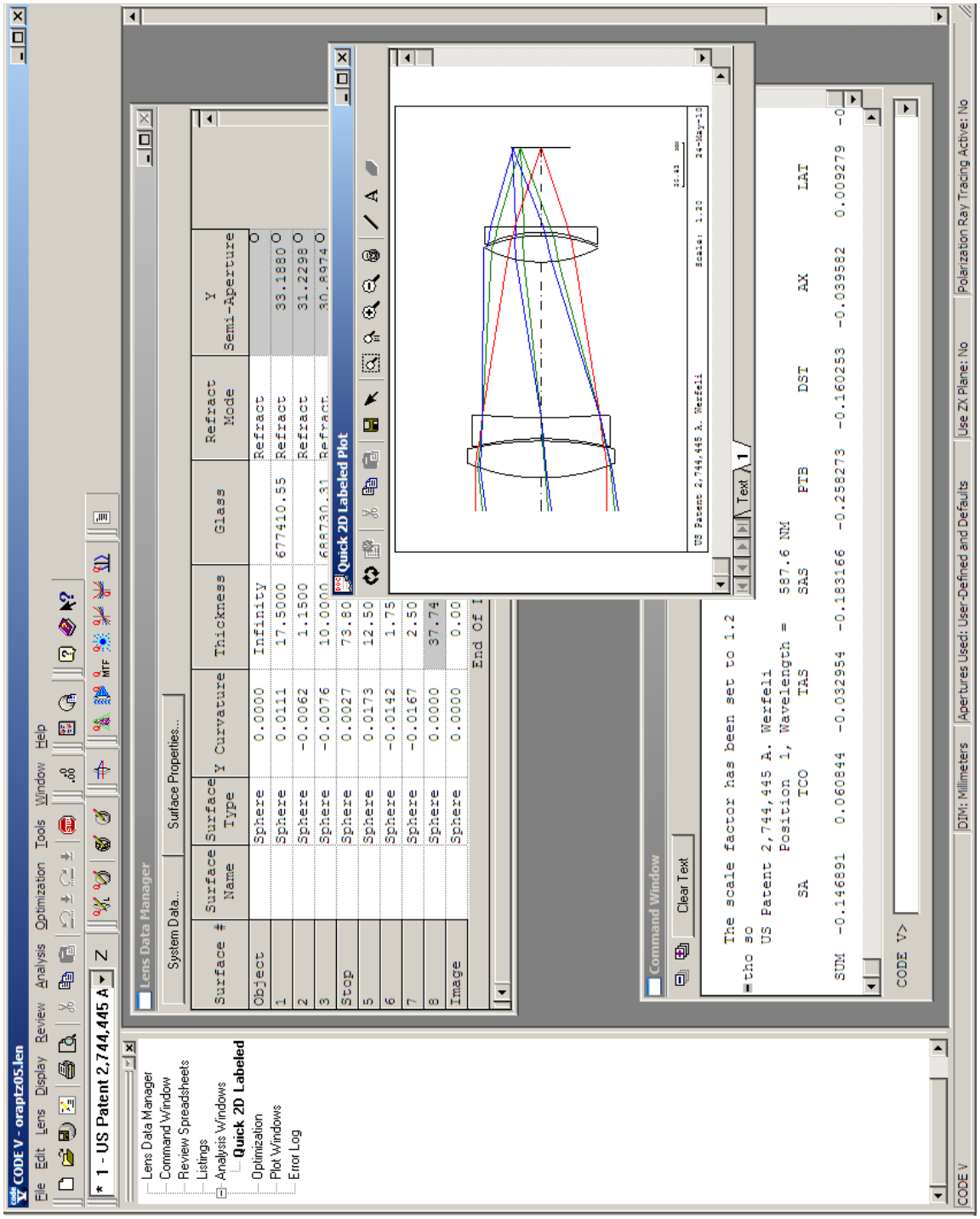


Figure 7.5: The Petzval lens in the CODE V design environment. CODE V provides many interfaces to construct the lens systems and visualise them. It also analyses the lens system by ray-tracing to extract the attributes. CODE V also has an API to interact with other programs.

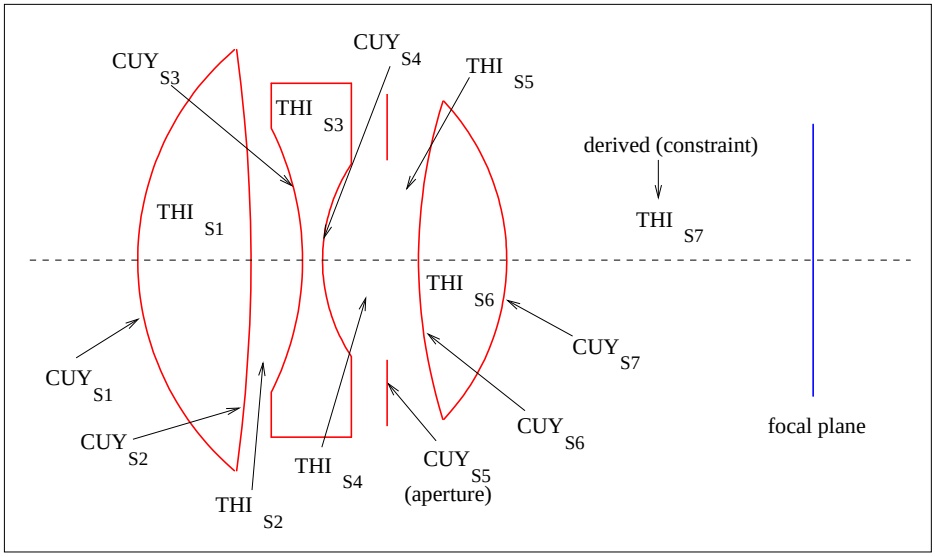


Figure 7.6: The decision variables used to define a lens system. Here, seven curvatures (CUY) and six thicknesses (THI) are used for each surface (S). Therefore, 13 *decision variable are used for this. The air-gap closest to the focal plane is a constraint such that* $THI_{S7} \geq 0$.

ing the API into our design framework, we define the decision variables and the objective functions using the CODE V design commands.

7.3 Decision variables

The decision variables required to construct a lens system include the curvatures of the surfaces of the optical elements and their thicknesses. The lens elements have to be placed within certain distances to minimise the aberrations. Therefore, we indicate the distances between lens elements as the thicknesses of air-gaps. We describe the design (or decision) variables with an example because the number of variables changes depending on the system that is being designed.

Figure 7.6 illustrates a lens system having three lens elements and an aperture. The aperture is an opening in the lens system, where light travels through. This opening determines how *collimated* the incoming rays are [Kidger, 2004]. A collimated beam of light is *nearly parallel* to the horizontal axis, such that it diverges slowly as the beam propagates through the lens system. This is useful to direct the light properly onto the focal plane and thus towards the film or image sensor. This plane is usually referred to as the image plane.

We consider the aperture also as a surface. Therefore, this lens system has seven curvatures. Each i^{th} surface is denoted by S_i and the curvatures are denoted by CUY. The thickness of the lens elements (or air-gaps) are denoted by THI. This system has seven thicknesses, but only six are used to define the decision variables. The last air-gap is derived by ray-tracing to indicate the focal plane. Therefore, the 13 decision variables for this system are CUY_{S1}, THI_{S1}, CUY_{S2}, THI_{S2}, CUY_{S3}, THI_{S3}, CUY_{S4}, THI_{S4}, CUY_{S5}, THI_{S5}, CUY_{S6}, THI_{S6} and CUY_{S7}. The aperture of the lens system is given by CUY_{S5}, while THI_{S7} is derived by ray-tracing. The construction of the lens system requires some additional data in the form of glass types and the surface which denotes the aperture. This data is provided explicitly into CODE V within the optimisation process. Attributes are extracted to evaluate the objectives after the construction of the lens system and ray-tracing is performed.

7.4 Objective functions

We extend and simplify the conflicting objectives defined by Ono et al. [1998] and Joseph [2005] to construct a five-objective lens design application. The objectives are constructed using attributes of the lens system that are extracted from the ray-tracing process. Each of these attributes can be computed manually rather than using a ray-tracing tool, but we have not included those details in this thesis. These mathematical calculations are lengthy and complex. However, they are all based on the refraction properties of glass. Born and Wolf [1999] describes these calculation procedures extensively. However for simplicity of discussion, we construct the objectives using the CODE V commands that are used to extract the attributes required to define the objectives. We formulate all the objectives to be minimised.

Objective f_1

An important requirement of a lens system is its sharpness. In photographic terms this is also known as *resolution* of a lens. We obtain sharpness by reducing the amount of Spherical Aberration (SA) in a lens system. Therefore the objective is defined as:

$$f_1 = SA \tag{7.3}$$

Objective f_2

Another important feature of a lens system is to be able to obtain an image which closely represents the original object. Therefore, reducing the amount of distortion (pincushion or

barrel) and field curvature is important. The objective giving a consolidated value indicating the distortion (DST) is defined as:

$$f_2 = DST \tag{7.4}$$

Objective f_3

The next objective was named the *merit function* by Joseph [2005]. This merit function is defined as the square root of the sum of squares of all the third order aberrations. Therefore, we define an objective function which minimises all the other aberrations in Figure 7.4 (not included in f_1 and f_2), and some that are used in industrial standard lens designs. We define this objective as:

$$f_3 = \sqrt{TCO^2 + TAS^2 + SAS^2 + PTB^2 + AX^2 + LAT^2 + PTZ^2} \tag{7.5}$$

The following aberrations are used to construct the merit function:

- TCO: Third order tangential coma aberrations
- TAS: Third order tangential astigmatic blur aberrations
- SAS: Third order sagittal astigmatic aberrations
- PTB: Third order Petzval blur aberrations
- AX: Axial colour aberrations
- LAT: Lateral colour aberrations
- PTZ: Petzval surface curvature

We used the aberrations related to resolution and distortion as separate objective functions in f_1 and f_2 because in photography these aspects are immediately visible on a lens design. Resolution and distortion are key attributes in the process to determine the *best* lens among a sample of lenses having equal EFLs. Therefore, we did not include them as part of the merit function.

Objective f_4

This objective is related to the aperture. Although the aperture is used to obtain the most parallel beams of light from a lens system, it can also reduce the amount of light falling onto the image plane. The lesser the amount of light falling requires more exposure of the image plane. This reduces the ability to capture fast moving objects without incurring blur. Therefore, the F-Number (FNO) is used to denote how much the aperture can be opened. The greater aperture can be opened the *faster* the lens is. The FNO of a lens is derived as the ratio of the diameter of the aperture against the EFL. Therefore, a small FNO indicates a wide-open aperture. However, opening the aperture increases the chance of aberrations because the light beams become less parallel to the horizontal axis. We define this objective as:

$$f_4 = FNO \tag{7.6}$$

Objective f_5

Increasing lens elements to reduce the aberrations also has another negative aspect by making the lens large. This increases the length and weight of the lens considerably. Therefore, a good fast lens design with minimum aberrations ideally should also have a low Overall Length (OAL). The OAL of a lens is defined as the distance between the first surface and the focal plane. To achieve this requirement, we define our fifth objective as:

$$f_5 = OAL \tag{7.7}$$

Other than these attributes to construct the objective values, we extracted some others to progress the optimisation process and analyse the designs.

Constraints

As we indicated in Section 7.3, the distance from the last surface of the lens system to the focal plane has to be derived from the ray-tracing process. A *real image* is created on the image plane if the lens system is able to focus on a plane on the right side of the last surface. Otherwise, the image will not be projected on the image plane if the focal plane is on the left of the last surface. In such a scenario the image created is known as a *virtual image* [Kidger, 2004] (see Figure 7.1). We consider the thickness of the last surface (THI_{S7}

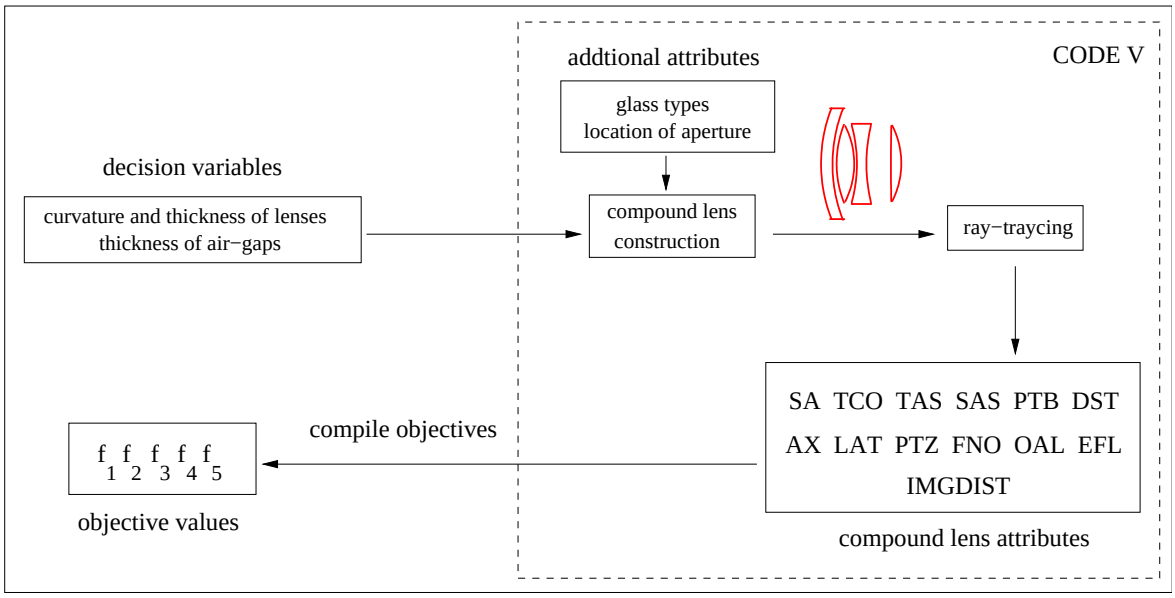


Figure 7.7: The lens evaluation framework. First, the curvatures and the thicknesses of each surface are input to CODE V as the decision variables. The compound lens is constructed with the decision variables and additional information concerning the type of glass and the location of the aperture. CODE V performs the ray-tracing to obtain the attributes of the lens. These attributes are used to construct the values for the objectives and constraints.

in the example in Figure 7.6) as positive if the image is real, otherwise negative if the image is virtual. Therefore, we consider $IMGDIST \geq 0$ as a constraint. Here, $IMGDIST$ will represent the thickness of the last surface of a lens system. For the example in Figure 7.6, $IMGDIST = THI_{S7}$. We consider a lens system adhering to the constraints as a feasible design.

Useful attributes

We are also interested in the magnification of the developed lens system. Therefore, we obtain the EFL from each feasible lens design from the ray-tracing process. Next, we describe the optimisation process using the reference point MDEPSO framework.

7.5 Reference point MDEPSO framework for optimising lenses

Similar to the aerofoil design application, we integrate the steps illustrated in Figure 7.7 into the steps 4–6 and 16–18 of Algorithm 14 (page 145). The evaluation process of the candidate

lens designs includes the building of the compound lens system, ray-tracing and extracting of attributes. The other steps in Algorithm 14 remain the same. Unlike the aerofoil application where the number of decision variables is constant, the optimisation goal of the lens design application is slightly different because the number of decision variables changes depending on the problem instance. In this lens design application the aim is to find good designs that have a given number of lens elements, glass types and a fixed aperture location that optimises the five-objectives. We selected an existing lens design having a similar number of lens elements, glass types and aperture location from the patent database to derive a reference point. However in this application, our aim was not to obtain a lens design having the same EFL as the reference design. The optimisation process would obtain lens design having a similar structure but different EFLs and better objective values than the reference design. Therefore, we next present the chosen problem instance with suitable variable bounds and the reference point used to guide the MDEPSO algorithm.

Problem instance and the reference point

The problem instance in this optimisation process consists of four lens elements and one aperture. The lens design was similar to the Petzval lens in Figure 7.3, with the addition of an aperture between the doublets. We included an aperture control into our problem instance to obtain a fast fixed focal lens design. Therefore, the optimisation process had a total of 17 decision variables. These were made with nine surfaces and eight thicknesses. The variables and their bounds are given in Table 7.1.

We have used the variable bounds given in Table 7.1, based on some existing lens designs from the CODE V patent database and initial experiments. All the lens elements have surface curvatures within $[-0.02, 0.02]$ to allow the creation of both convex and concave lens elements. The first surface (CUY_{S1}) and last surface (CUY_{S9}) have different boundary values to force the generation of converging surfaces. Having the last surface to be a converging lens is important to avoid the generation of a divergent lens that outputs virtual images. The first surface being convergent helps to allow more light rays into the lens system, than from a divergent surface. We also used the constraint $THI_{S9} \geq 0$ to remove any infeasible solutions that creates virtual images. The aperture should be a flat surface, therefore $CUY_{S5} = 0.0$. All the lens elements are set to have thickness values within $[1.0, 30.0]$. However, we have increased the thickness bounds of the aperture and the lens surface in front of it to obtain lens designs with wide-open apertures. We observed a similar placement of the aperture in other

Table 7.1: The decision variables used in the lens design application with their UBs and LBs.

Variable x_i	$x_i^{(LB)}$	$x_i^{(UB)}$	
$x_1 : CUY_{S1}$	0.0000	0.0200	first surface
$x_2 : THI_{S1}$	1.0000	30.0000	
$x_3 : CUY_{S2}$	-0.0200	0.0200	second surface
$x_4 : THI_{S2}$	1.0000	30.0000	
$x_5 : CUY_{S3}$	-0.0200	0.0200	third surface
$x_6 : THI_{S3}$	1.0000	30.0000	
$x_7 : CUY_{S4}$	-0.0200	0.0200	fourth surface
$x_8 : THI_{S4}$	10.0000	60.0000	
$x_9 : CUY_{S5}$	0.0000	0.0000	aperture
$x_{10} : THI_{S5}$	10.0000	60.0000	
$x_{11} : CUY_{S6}$	-0.0200	0.0200	fifth surface
$x_{12} : THI_{S6}$	1.0000	30.0000	
$x_{13} : CUY_{S7}$	-0.0200	0.0200	sixth surface
$x_{14} : THI_{S7}$	1.0000	30.0000	
$x_{15} : CUY_{S8}$	-0.0200	0.0200	seventh surface
$x_{16} : THI_{S8}$	1.0000	30.0000	
$x_{17} : CUY_{S9}$	-0.0200	0.0000	eighth surface

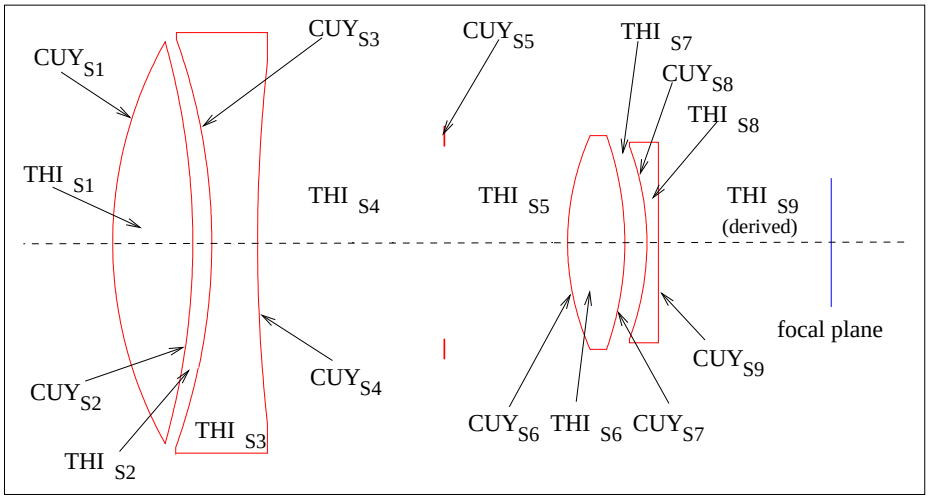


Figure 7.8: A candidate lens design having four lens elements and an aperture. The aperture lies between the two doublets. There are 17 *decision variables in total.*

Table 7.2: These aspiration values for each objective of the reference lens design are used to define the reference point $\vec{\overline{z}}$.

Objective f_i	**Aspiration value** $\overline{z}_i$
f_1	0.1469
f_2	0.1603
f_3	0.1517
f_4	0.1595
f_5	0.1192

patented lens designs, which was also confirmed by our initial experiments. A candidate lens design for the problem instance is illustrated in Figure 7.8.

To guide the MDEPSO algorithm in this five-objective problem instance, we used a Petzval lens design — US Patent 2,744,445 by A. Werfeli [Optical Research Associates, 2010] — to derive a reference point. Figure 7.3 illustrated this lens design. The aspiration values for the five objectives on this lens design are given in Table 7.2. The aspiration values have been normalised to project values within $[0, 1]$. We use this same normalisation scheme within the evaluation process for the extracted attributes from the ray-tracing process. However, when we present the attributes of the lens designs, the normalised values will be converted back to their original values.

We used the same glass materials to construct the lens elements of the reference design in the candidate lens designs also. This reference lens design has an EFL equal to 99.7065mm($\approx$

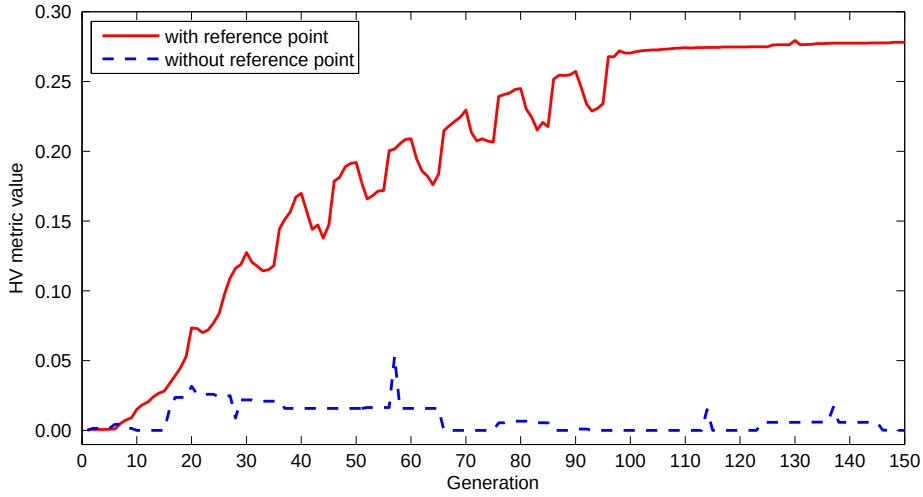


Figure 7.9: The average HV values at each iteration for the reference point and standard MDEPSO algorithms. These HV values are derived from the calculation process described in Section 4.3.

100mm). However, in our optimisation process we are interested in obtaining a fast fixed focal lens design that has four lens elements and an aperture. These lens systems may have different EFL values. Next, we illustrate the results from the optimisation process and analyse them considering the characteristics of each objective.

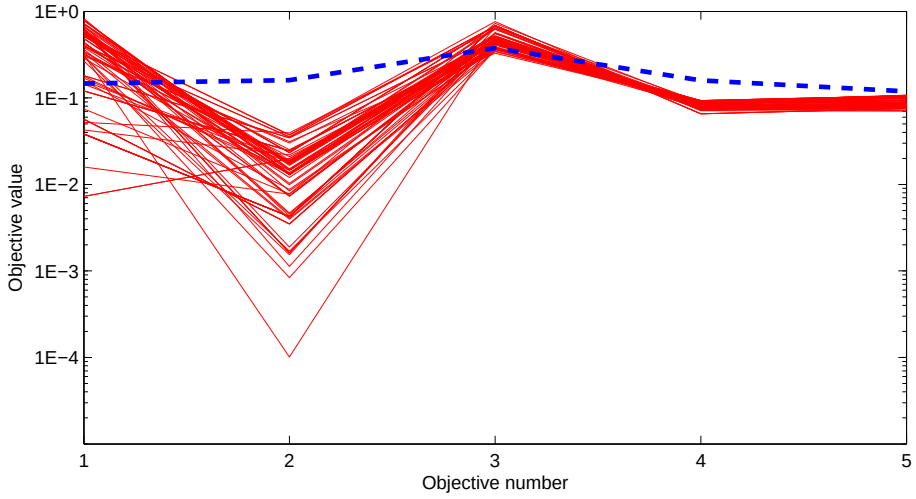
7.6 Experiments

We performed experiments on the lens design application with similar parameter settings as the aerofoil application. For completeness, we used the reference point MDEPSO algorithm compared with the standard MDEPSO algorithm on the five-objective lens design problem. Both MDEPSO algorithms used a population size of 100 for 150 iterations. The evaluation process of the individuals was not as computationally expensive as the aerofoil application. Therefore, we had a total of 15,000 evaluations per run. The results were averaged from 30 independent runs. The same parameter settings of $CR = 0.2$ and $F = 0.4$ were used for the DE operator in MDEPSO. The spread value for the reference point algorithm was set as $\delta = 0.5$.

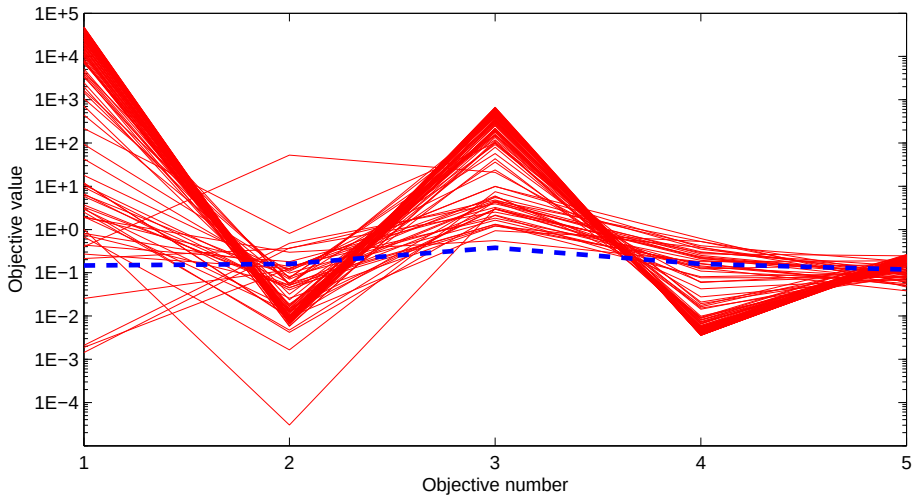
7.6.1 Results for convergence

In Figure 7.9 we illustrate the average HV values at each iteration of the runs of the lens design application. Here, we used $\delta' = 0.75$ to define the volume in the preferred region for calculating the HV as described in Section 4.3 (page 95). This δ' value was used to obtain at least 90% of the particles of the algorithm that has better convergence. The abrupt changes in HV values were due to the exploration and exploitation processes of the distance metric based approach we described in Section 3.1.3 (page 56). Figure 7.9 clearly indicates that the reference point MDEPSO algorithm has better convergence than the dominance ranking MDEPSO version. We also observe that the reference point based algorithm has converged to an optimum around iteration 100. This convergence result again confirms that the distance metric based approach for EMO algorithms is suitable to locate solutions in many-objective problems.

Figure 7.10 illustrates the solution points obtained from best converged runs of each algorithm. The best convergence was deduced from the run that had the largest HV value in the final iteration. Similar to the aerofoil application (Figure 6.7 – page 148), the reference point is given as the dashed line, while the solid lines represent the candidate solutions. We observe from Figure 7.10 that there are solution points which have better values than the reference lens design in at least one objective function. Next, we analyse the preferred lens design with the reference design on various aspects of the objective functions. We made use of the many visualisation tools found in CODE V for this analysis.



(a)



(b)

Figure 7.10: Final solutions obtained on MDEPSO: (a) Using a reference point; (b) Without a reference point. The dashed line represents the reference point.

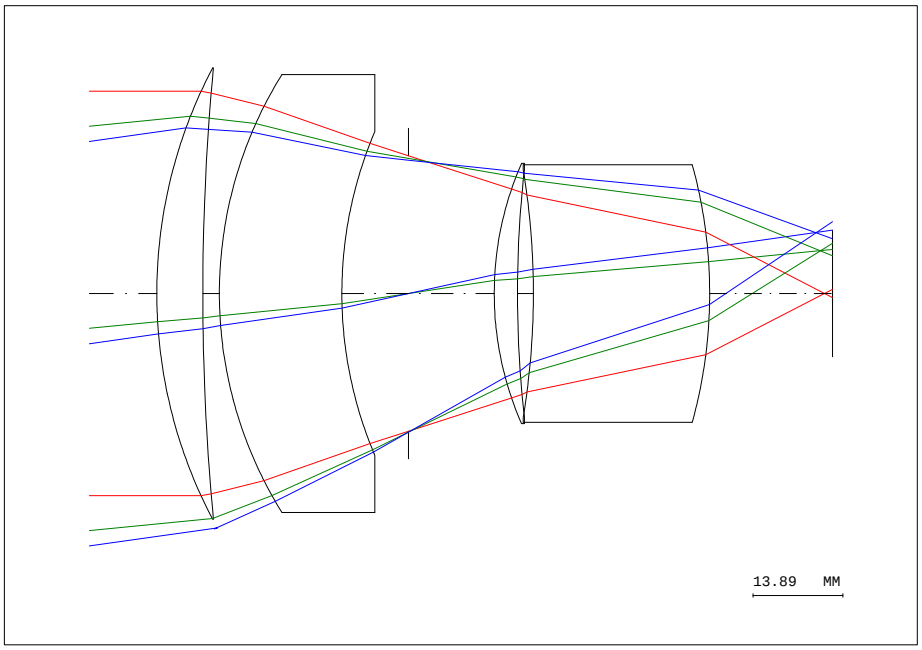
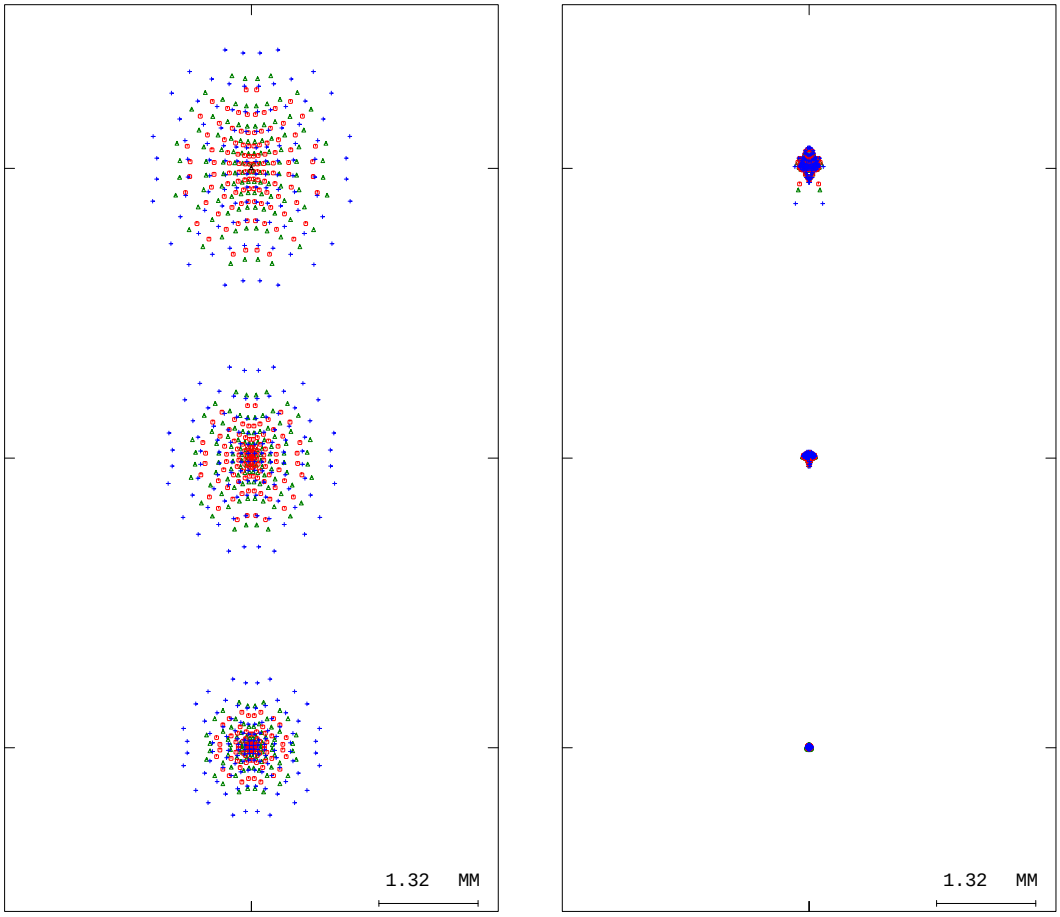


Figure 7.11: The preferred lens design having four lens elements and one aperture.

7.6.2 Analysis of the optimised designs

Figure 7.11 illustrates the preferred solution found in the optimisation process of the lens design application. Unlike the aerofoil application, we do not compare this design with the reference lens design in the same diagram. In this application, our aim was to obtain a lens design with four lens elements and an aperture. However, the reference lens design contained only four lens elements. Therefore, in our analysis we compared the behaviour of light between the candidate design and the reference design than their physical appearances. The lens diagram in Figure 7.11 shows that the light rays focus on points further from the focal plane when compared with the reference design in Figure 7.3. This behaviour can be better visualised from a *spot diagram*.

The spot diagram illustrating the divergence of the light rays onto the focal plane is given in Figure 7.12. The divergence of the light rays indicates the *softness* of the created image. The more divergent the rays are, the softer the image is. The spot diagram illustrates the divergence of the light rays at each point of the focal plane where the rays meet. The bottom of the spot diagram represents the central ray and the top represents the outermost ray. Convergence of rays is indicated by the tight clustering of the points. Therefore, we observe that the reference design has very good central sharpness, while the corners are



(a) (b)

Figure 7.12: Spot diagrams for: (a) Preferred lens design; (b) Reference lens design.

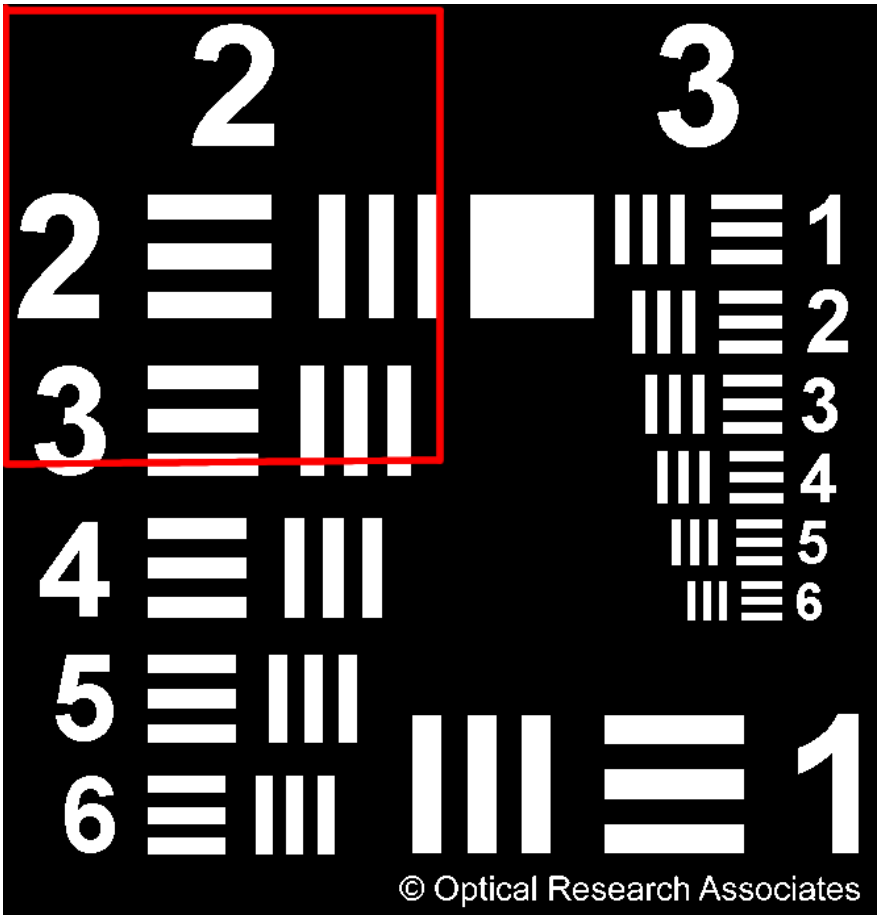
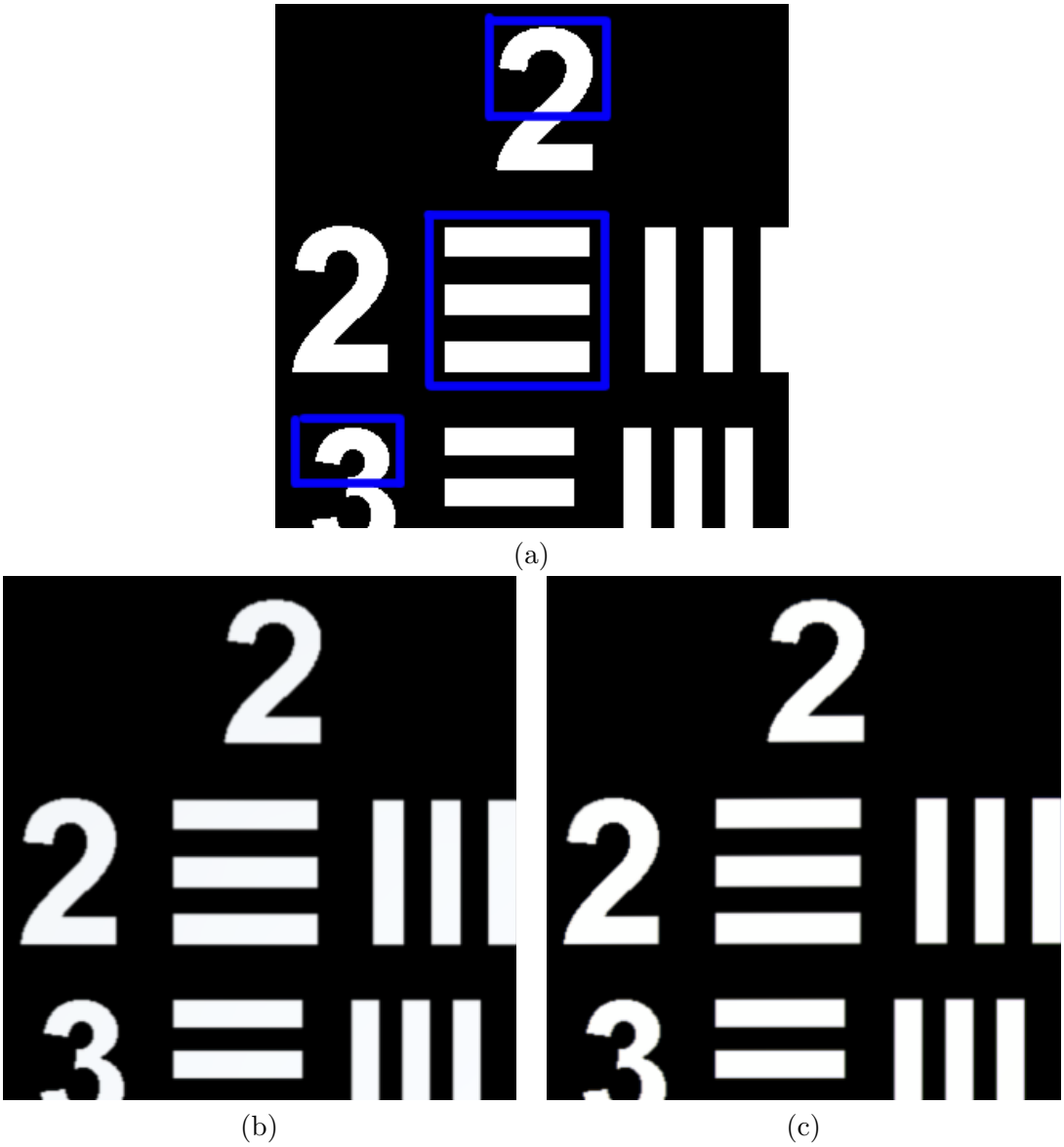


Figure 7.13: The input image used to test the sharpness of the lens designs. The selection in red is enlarged in Figure 7.14.

slightly softer. The preferred design is softer equally on all regions of the focal plane than the reference design. Although one may conclude that the preferred lens is a poor design, it actually is not. This is mainly because in modern still cameras, the image plane can be moved forward and backward by small amounts to fix such issues. We also observe from Figure 7.11 that the divergence occurs very close to the focal plane, which would not make a drastic effect on the final output image. The next test will confirm this claim.

Objective f_1 dealt with the amount of spherical aberrations of the lens design. To illustrate the actual effect of the reduction of spherical aberrations of the light rays (reducing the softness), we present the images created of an object[1] through the preferred and the refer-

[1]All *objects* used in the analysis are images found in the image database of CODE V. These images have been presented in this thesis with the written permission of Optical Research Associates [2010].



(a) (b) (c)

Figure 7.14: Sharpness crops of the lens designs: (a) Input object; (b) Output from the preferred lens design; (c) Output from the reference lens design.

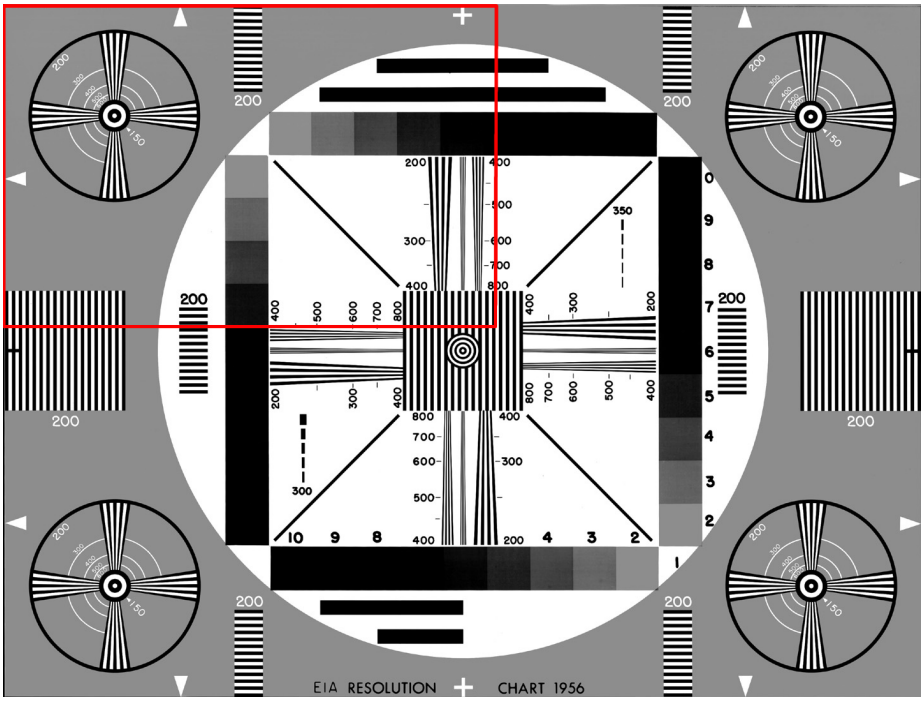
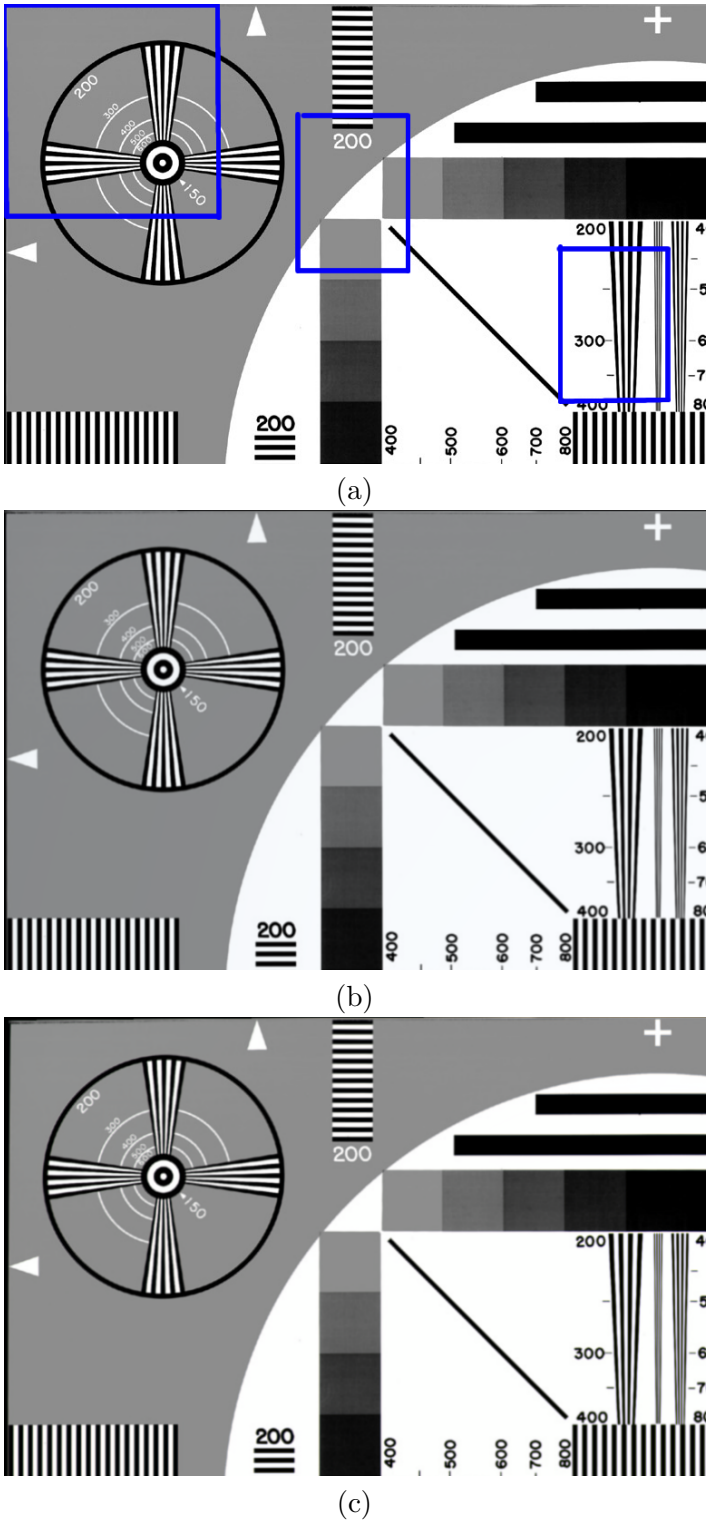


Figure 7.15: The input image used to test the sharpness and distortion of the lens designs. The selection in red is enlarged in Figure 7.16.

ence lens systems. The image in Figure 7.13 was used to test the sharpness of the produced images. We present crops bounded by the red rectangle of Figure 7.13 in Figure 7.14 for better visualisation[2]. The regions indicated in blue in Figure 7.14 represent the regions of interest regarding sharpness of the output. We observe that the difference in the sharpness of both lens designs is very small compared to the output results, where the reference design is slightly sharper. Both lenses have produced *softer* images compared with the sharp edges of the input object. However, between the lenses the final output is almost identical. This indicates that the softness of the preferred lens design is not too drastic when considering the divergence of the light rays alone.

Figure 7.15 illustrates the image used to test the sharpness and distortion of the lens designs. We present the enlarged crop of the region in red from Figure 7.15 in Figure 7.16. Here also we present the regions of interest in blue. These regions contain many small lines, which give a good indication of the resolution. We observe that the reference design is sharper

[2]The subtle differences in colour and clarity of the actual outputs from CODE V have changed during the process of preparing the images for typesetting and printing of this thesis. Therefore, we have uploaded the output images from CODE V at http://www.wicky.me/downloads/lens_analysis.zip for further reference.



(a)

(b)

(c)

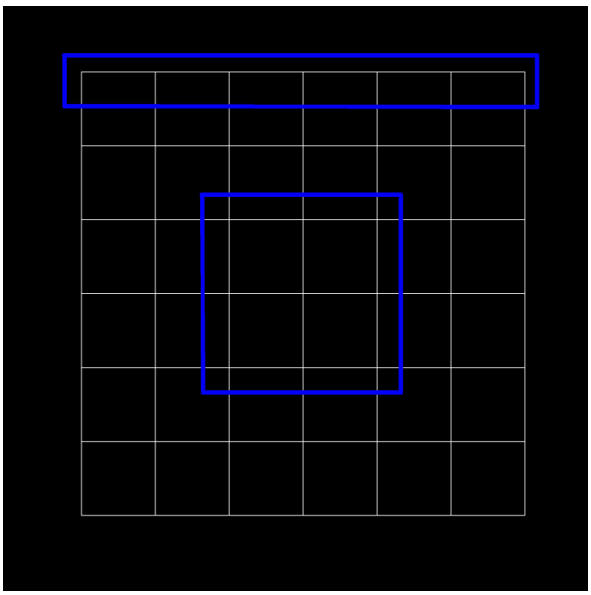
Figure 7.16: Sharpness and distortion crops of the lens designs: (a) Input object; (b) Output from the preferred lens design; (c) Output from the reference lens design.

in the centre of the image than the corners, which was also seen in Figure 7.12. However, the preferred image shows an even sharpness across the entire output. This behaviour is advantageous when the sharpness is increased by moving the image plane in the camera. We also observe more distortion on the reference lens than the preferred lens design. Handling distortions in a lens system was presented in objective f_2. To illustrate this distortion clearly we perform a test on a simpler object, which has less detail in the centre.

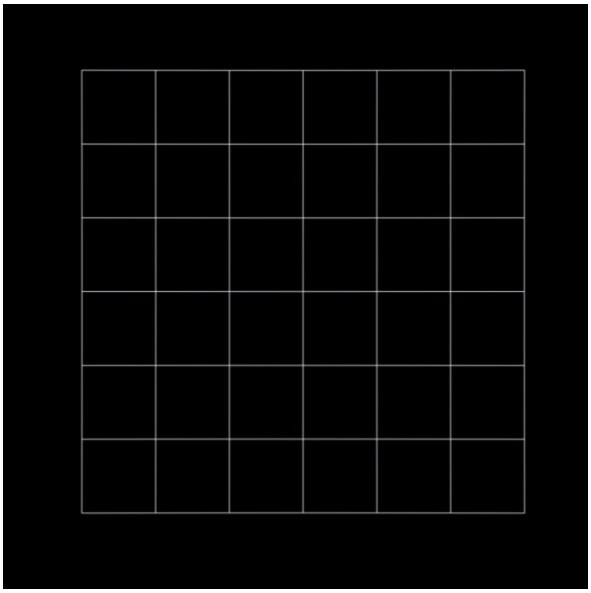
To visualise the distortion handling of the preferred and reference lens designs, the object and output images are illustrated in Figure 7.17. Observing the straight lines of the object — regions in blue — shows that the *sagging outwards* is more pronounced on the reference lens design. We defined this behaviour as barrel distortion earlier in Section 7.1.2. The preferred lens design depicts very little distortion compared with the reference design. This output shows that the preferred lens design has better control of distortion than the reference lens design. We also observe similar barrel distortions in Figure 7.16, where the reference design had more prominent distortions than the preferred design.

We observe some colour shifts occurring in the lines in the centre of the reference design in Figure 7.17. These colour aberrations were considered in objective f_3. There is also a slight shift of the entire object towards the bottom right on the created images. This is more prominent on the reference design than the preferred design. This aberration was known as astigmatism, which was also considered in objective f_3.

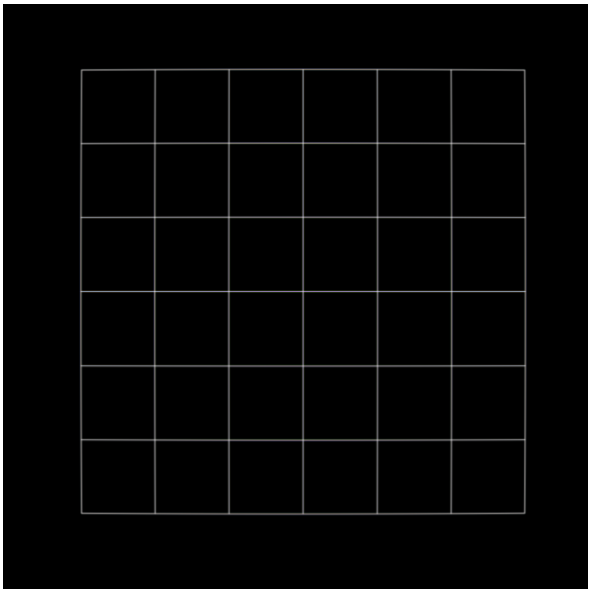
To illustrate the colour reproduction accuracy, we input some primary colours to the lens systems. The results can be seen in Figure 7.18. Both the preferred and reference lens designs were able to reproduce colour accurately. However, we observe the slight loss of sharpness in the output conforming to the results in Figures 7.14 and 7.16.



(a)

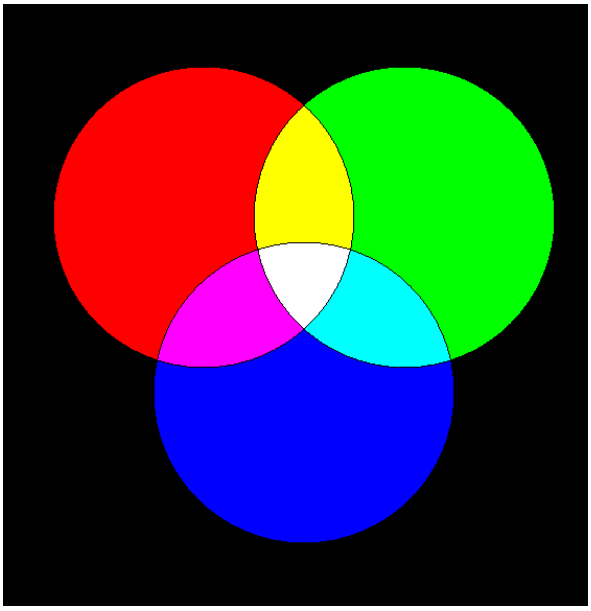


(b)

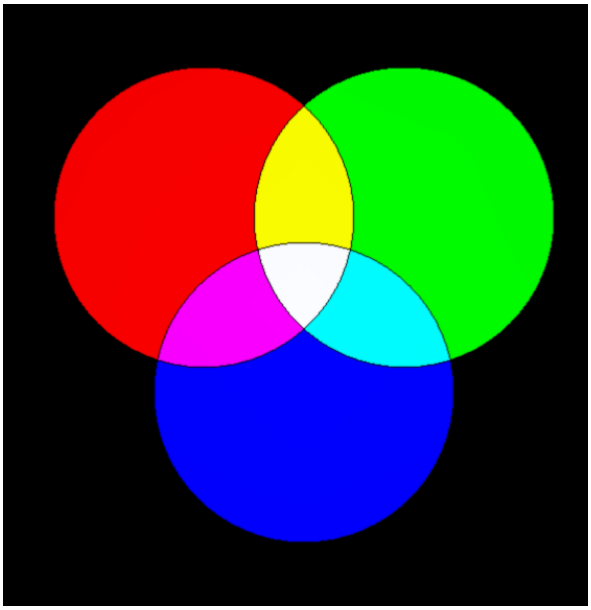


(c)

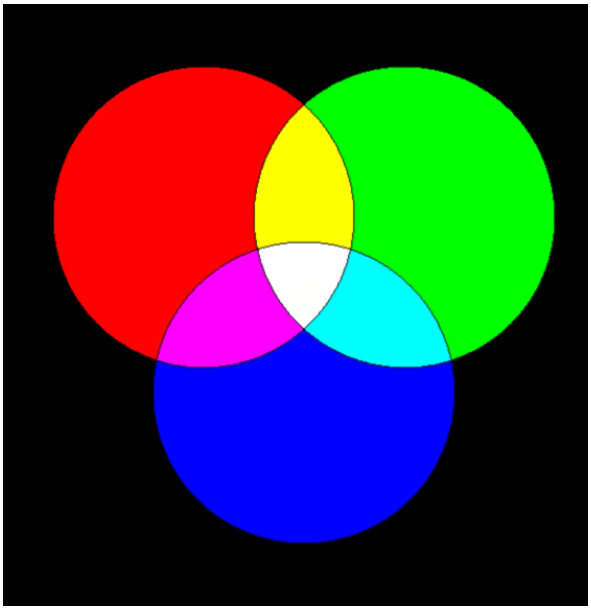
Figure 7.17: Distortion of the lens designs: (a) Input object; (b) Output from the preferred lens design; (c) Output from the reference lens design.



(a)



(b)



(c)

Figure 7.18: Colour reproduction of the lens designs: (a) Input object; (b) Output from the preferred lens design; (c) Output from the reference lens design.

Table 7.3: A summary of the best lens designs for each objective. The values for EFL were rounded to a number divisible by five to conform to industrial design standards. FNO and OAL were rounded to the first decimal place also to conform to industrial standards.

Lens design	**EFL** (mm)	**FNO** (f/#)	**OAL** (mm)
Reference design	100	1.6	119.2
Preferred design	70	1.0	113.4
Best lens for f_1	85	1.2	133.4
Best lens for f_2	55	1.2	103.5
Best lens for f_3	65	1.6	130.4
Best lens for f_4	100	0.7	75.6
Best lens for f_5	55	0.8	61.2

Table 7.4: The normalised objective of the best lens designs for each objective. The values in ***bold*** *are the best values for each corresponding objective.*

Lens design	f_1 **value**	f_2 **value**	f_3 **value**	f_4 **value**	f_5 **value**
Reference design	0.1469E+00	0.1603E+00	0.3775E+00	0.1595E+00	0.1192E+00
Preferred design	0.1329E+00	0.0254E+00	0.3318E+00	0.0978E+00	0.1134E+00
Best lens for f_1	**3.2801E-04**	0.2161E+00	0.7882E+00	0.1216E+00	0.1334E+00
Best lens for f_2	0.2654E+00	**1.4261E-06**	0.5797E+00	0.1209E+00	0.1035E+00
Best lens for f_3	0.2725E+00	0.2272E+00	**0.1663E+00**	0.1706E+00	0.1304E+00
Best lens for f_4	0.5209E+00	0.0233E+00	0.6829E+00	**0.0656E+00**	0.0756E+00
Best lens for f_5	0.3073E+00	0.0213E+00	0.7708E+00	0.0883E+00	**0.0612E+00**

Table 7.3 presents a summary of the best lens designs for each objective function. The results have been converted back from the normalised values and rounded to conform to industrial standards. The values of each objective for each lens design are given in Table 7.4 in their normalised form. Corresponding to objectives f_4 and f_5, the best lens designs having the lowest FNO and OAL are seen from Tables 7.3 and 7.4. It is important to note the different EFL values of the obtained lens designs, because we did not include a constraint to achieve a specified EFL value. Handling many constraints was not in the scope of this thesis. For future work such additional constraints can be included in the application. On average, we observe that the lens designs that were obtained from the optimisation process have low FNO values. These values indicate that fast lenses were generated.

Figure 7.19 illustrates the best lens designs obtained if each objective was considered separately. These lens design corresponds to the objective values presented in Table 7.4. Among the designs, the best lens design for objective f_2 can be considered as an unsuitable design.

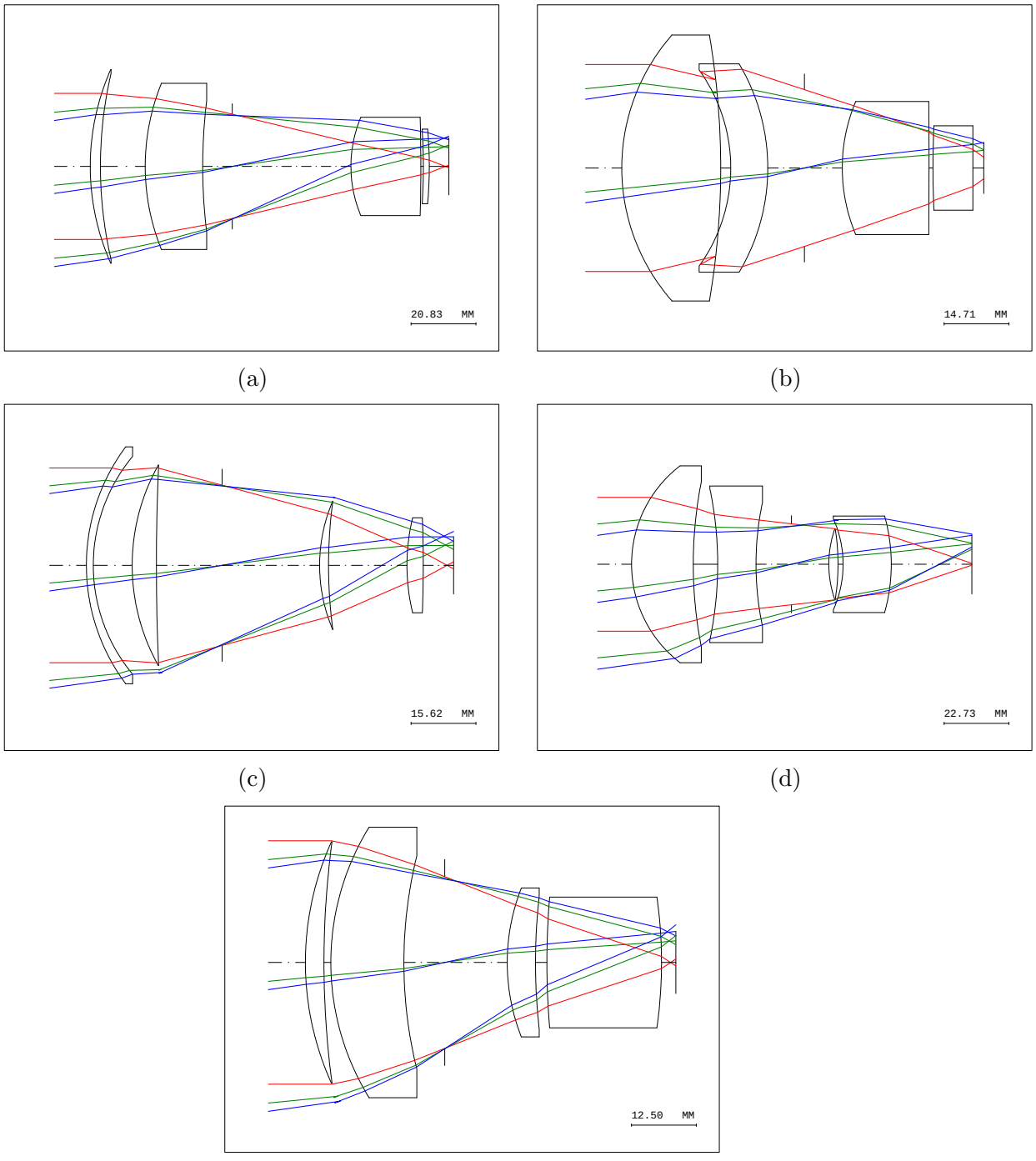


Figure 7.19: The lens diagrams for the best designs obtained for each objective: (a) Objective f_1; (b) Objective f_2; (c) Objective f_3; (d) Objective f_4; (e) Objective f_5.

This design has one lens element inside another, which is not suitable for manufacturing. Such scenarios can be avoided in the optimisation process by including more constraints. However, we did not consider such constraints because the scope of this thesis does not include optimisation of many-objective problems having many constraints. Future work can include such additional constraints within the optimisation process.

7.7 Summary

In Chapters 6 and 7 we aimed to find answers to the research question: *How can real-world problems benefit from user-preference EMO algorithms in the optimisation process of engineering designs?* We developed this final research question with the aim of compiling all the major contributions of this thesis into some real-world applications. We chose the aerofoil design and lens design applications that are many-objective in nature for this purpose. Our main aim was to investigate the feasibility of including the refined knowledge of the DM within an optimisation process rather than simply executing an algorithm and later analysing the solutions. We were able to integrate the existing knowledge of a DM into the optimisation process by deriving reference points. The DM was also involved in setting the boundaries for the decision variables and guiding the optimisation using reference points derived from existing designs.

In the lens design application, we used CODE V as the external evaluating program. The problem instance we considered had five objectives that corresponded to preferred characteristics of a compound lens system. These objectives were formulated based on some existing work on lens designs using EAs. We formulated the lens design problem instance to have 17 decision variables and 1 constraint. These corresponded to a lens system having four lens elements and one aperture. The constraints were used to obtain real images from the lens system. We also set the boundary values for the variables such that more light rays entered the system and converged onto the focal plane, than diverge out of the system. The reference point for this application was derived from a Petzval lens system found in the CODE V patent database. This lens has four lens elements and no aperture. We decided on this reference lens design, so that new lens elements can be generated from the reference point MDEPSO algorithm.

Similar to the aerofoil application we compared the convergence ability of the reference point MDEPSO algorithm with the standard MDEPSO algorithm on the lens design problem. The HV results showed that the reference point MDEPSO algorithm was able to converge

better than the dominance ranking MDEPSO algorithm in the lens design problem. We analysed the preferred lens design with the reference lens design using many tools available in CODE V. This analysis was with respect to the various components of the objective functions. However, the images created by simulating an input object onto the lens systems do not display all the fine details. This was an issue we faced when typesetting and compiling this thesis. Therefore we have included a URL, where these images can be downloaded for further analysis. The optimisation process for the lens design also was able to produce the best lenses for each objective in a similar manner as the aerofoils. We have also presented an analysis of these results also.

One of the major drawbacks we found in our optimisation process for each application is the omission of some constraints. Most of the real-world applications consist of many constraints required to develop a design that can be manufactured properly. Although handling a large number of constraints was outside the scope of this thesis, we included some for a successful optimisation process. Most of the results obtained from the reference point MDEPSO algorithm were satisfactory, but some were not suitable for manufacturing. In fact these additional constraints can be viewed as user-preferences specified in the decision-space instead of the objective-space. In future work, we can include these additional constraints so that the optimisation process is able to generate aerofoil or lens designs that are suitable for manufacturing. In the next chapter, we present some of these possible extensions of the research presented in this thesis, our key findings from the various studies and concluding remarks.

Chapter 8

Conclusions and future work

Our research began with the goal of exploring methods of integrating user-preference into EMO algorithms to aid them in locating solutions in many-objective problems. In many-objective problem instances, comparing candidate solutions using the standard dominance relation becomes ineffective as the number of objectives increases. The main reason is the high probability of individuals of the population being non-dominated to each other. This causes the optimisation process to stagnate.

Many methods have been proposed in the literature that address this problem by modifying the standard dominance concept. However, these methods tend to be very complex and computationally expensive since the standard dominance ranking methods is still employed. One alternative approach that has become popular recently is to integrate existing multi-criteria decision making user-preference methods into EMO algorithms. Our research began at this point; to explore MCDM methods that are useful to obtain more efficient user-preference based EMO algorithms. The final goal was to develop EMO algorithms that are effective in locating useful solutions in preferred regions of the many-objective landscape than spending valuable computational resources trying to locate the entire Pareto front.

Firstly, we have developed a distance metric that can be used efficiently to guide an EMO algorithm in many-objective problems without the need for dominance ranking. This distance metric was according to using the user-preferences provided by the DM. One of the main challenges in the current user-preference EMO literature is that there is no unified approach to measure the performance of such algorithms. We have designed a performance metric based on the HV metric that is suitable for measuring the performance of user-preference EMO algorithms. The key here was to consider only the solutions in a preferred region within

the HV calculation process. This emphasis favoured algorithms that converge on a preferred region than the entire Pareto front.

Among the many stochastic search-strategies found in the EMO literature, we were more interested in PSO algorithms because of their fast convergence characteristic. We also found that guiding PSO algorithms with user-preferences were natural extensions to the PSO update rules, because they explore the search-space by following leading particles.

Utilising the fast convergence property, we deployed our MDEPSO algorithm on some computationally expensive real-world many-objective problems, including optimising aerofoils and lens designs. These real-world applications gave us some new insights into methods of integrating the refined knowledge of a DM into the optimisation process. This methodology was efficient in locating useful solutions rather than having to pick solutions from a set of solutions spread across the objective-space. We present all these results in more detail in the next section with reference to our original research questions.

8.1 Research questions

We restate our research questions here and provide the answers from the findings of the research undertaken in this thesis.

1. *How can we integrate user-preference methods from the MCDM literature to design efficient EMO algorithms that can find solutions for many-objective problems?*

    - We introduced a distance metric in Chapter 3, which can be derived from any MCDM user-preference method. This distance metric is a generic approach that can be integrated into any evolutionary algorithm. In the examples, the reference point method and light beam search were used. This distance metric was used to guide the population by providing the necessary selection and search pressures. Although the classical MCDM approaches were used to obtain a single solution, we were interested in obtaining a set of solutions using an EMO algorithm. Therefore, we introduced a δ parameter that controls the spread of solutions along a solution front. This δ value simply indicates a larger value for a larger spread and a smaller value for a smaller spread. It does not represent an actual length measure of the spread of solutions.

- The example EMO algorithm we introduced using the proposed distance metric was d–EMO. This was essentially NSGA-II with the non-dominated sorting mechanism replaced by the distance metric and δ value. We presented how d–EMO is more efficient than dominance ranking based EMO algorithms because d–EMO has a lower computational complexity. Experimental results using problem instances of the ZDT, DTLZ and WFG test suites indicated that d–EMO is able to converge onto the preferred regions of the Pareto fronts. Visual plots confirmed this claim for the two and three-objective problems, while for higher objectives we used the characteristics of the problems to assess the convergence. Comparative results against the dominance ranking EMO algorithms with and without user-preferences, concluded that d–EMO is effective in locating solutions in many-objective problems.

2. *How can we measure the performance of a user-preference EMO algorithm using a suitable performance metric?*

- The existing performance metrics for EMO algorithms are not suitable for user-preference algorithms because they are designed to compare algorithms that attempt to locate the entire Pareto front. Some of these most popular metrics rely on the knowledge of a known Pareto front, which is not suitable for real-world problems. However, the HV metric does not require the knowledge of a known Pareto front. The HV metric basically measures the area bounded by the solution points and the nadir point, which is derived from the worst values for each objective. The HV metric is also desirable because it gives a single measure for both convergence and spread of solutions. However, the HV metric is not suitable for user-preference EMO algorithms in its original form because it still assumes the convergence to the entire Pareto front. Therefore in Chapter 4, we introduced a version of the HV metric that takes into consideration solutions converged only onto a preferred region of the Pareto front.
- The main steps in the HV metric we introduced are to define a volume in the preferred region and exclude any solutions that lie outside it. To this end we defined a δ' value, which is a preference given by the DM. The DM will provide a δ' such that a volume is defined having $2\delta'$ for each objective. Although the volume does not necessarily have to have an equal length for each objective, we used it

for simplicity. All the solutions of the EMO algorithms that are to be compared are combined to make a combined solution set. These EMO algorithms can be all user-preference algorithms or not. The volume is defined around the solution closest to the ideal point. Then any solution point outside this volume is excluded and the nadir point is derived. This method ensures the nadir point is located within the defined volume inside the preferred region. Once the nadir point is defined, the normalised HV metric is calculated for each EMO algorithm. This HV calculation process is suitable for user-preference EMO algorithms because it favours algorithms converging better onto the preferred regions with a good spread of solutions.

3. *How can we develop a robust PSO algorithm which can be used in many-objective problems?*

   - PSO algorithms are desirable stochastic search strategies for EMO algorithms because of their fast convergence. This fast search process is ideal for computationally expensive applications like the real-world problems we considered in this thesis. However, this fast convergence property can have a negative impact on the search process because the swarm can converge onto a local optimal solution front. Therefore, we explored methods to overcome this issue but still retain the fast convergence ability of a PSO algorithm. We introduced MDEPSO in Chapter 5, which is a multi-objective hybrid DE and PSO algorithm. The unique feature of this PSO variant is that leader particles are generated using the DE/rand/1/bin rule. This differs from other multi-objective PSO algorithms because most of them select leader particles from the population or an external archive. The results from the two and three objective ZDT, DTLZ and WFG test suites showed that MDEPSO is capable of locating the Pareto front of difficult multi-modal problem instances.
   - To give MDEPSO the ability to locate solutions in many-objective problems, we integrated the distance metric that was introduced in Chapter 3. The results from the many-objective problem instances having three, five and ten-objectives from the DTLZ and WFG test suites confirmed that the reference point and light beam search MDEPSO algorithms are capable of locating solutions on the Pareto fronts. Therefore, the distance metric based MDEPSO algorithm was able to

handle difficult many-objective multi-modal problem instances better than other PSO algorithms.

4. *How can real-world problems benefit from user-preference EMO algorithms in the optimisation process of engineering designs?*

   - Real-world problems often have unknown Pareto fronts. Therefore, for an EMO algorithm to find solutions in a many-objective problem, some existing knowledge about the problem instance is required. The DM in this instance can provide the necessary variable boundaries with the aid of existing solutions and later adjust them as desired.
   - We found that the DM can be much more involved in the optimisation process when user-preferences are included. The DM has the ability to introduce some aspiration points that can be used to guide the EMO algorithm in the many-objective problem instances. We found that reference points can be derived using existing engineering design solutions, which can be integrated into our user-preference framework. Integrating this existing refined knowledge of the DM into the optimisation process is very efficient and effective in many-objective real-world problems, especially when they are computationally expensive and have a multi-modal objective-space. We presented example real-world applications describing these steps in Chapters 6 and 7.

8.2 Additional findings

We identified some additional findings from this research including:

- One has to properly define the preferred regions of the objective-space to compare the performance of user-preference EMO algorithms. We defined a preferred region with the following information:

  1. The search direction governed by the aspiration points
  2. The spread of solutions across the solution front

  The challenging criterion is to define the spread of solutions, because all the methods found in the literature and also presented here do not specify an exact measure of

spread. All of the preference methods define the spread as large or small, which is not an exact measure. However, providing an exact measure is not meaningful when we consider an arbitrary unknown problem instance.

- The proposed distance metric approach is more computationally efficient than dominance ranking based approaches. The computational complexity of a dominance ranking based algorithm is usually given as $O(Mn^2)$, while for our proposed approach it is $O(n \log n)$, where M is the number of objectives and n is the size of the population. This is because our proposed approach only has a sorting process of the population based on the distance metric.

- We can use a parallel axis plot to visualise the solution front for a many-objective problem instance. In such a plot, individuals converged to a preferred region is seen as a concentrated clustering of lines within a smaller range of the objective values. If an algorithm was to find the entire Pareto front, this plot would be filled with lines spanning the entire range of the objective values.

- The key to our proposed HV metric calculation process is to exclude solutions that are not within the defined preferred region. This step is useful to compare the performance between any user-preference algorithms, or algorithms that attempt to locate the entire Pareto front.

- It is important to select the solution point that is closest to the ideal point to define the volume for the HV value calculation. Choosing the solution closest to an aspiration point can be erroneous, because the calculation will favour algorithms that converge closely to that aspiration point. If this aspiration point happens to be in the feasible region, the algorithms that managed to locate solutions dominant to the aspiration point will get lower HV values, thus giving the impression that a weaker algorithm is better.

- Although the proposed normalised HV metric is used to assess the final solutions, we found that the HV calculation method is also useful to monitor the progress of the optimisation process. The increase of HV values within each generation indicates that the algorithm is converging onto a solution front in the preferred region and also spreading across the front. We also observed the exploration and exploitation stages of our distance metric approach indicated by the changes of the HV values.

- Generating leader particles using a DE method is more advantageous than selecting candidates from the swarm population or an archive, especially if the problem instance is multi-modal. This strategy gives a better chance to avoid the population from getting stuck in local optimal solution fronts.
- We found that reference points can be derived using existing engineering designs. These reference points can be used to either improve the existing design as in the aerofoil application, or to generate new designs as in the lens design application. The DM is more involved in the optimisation process with the use of user-preferences making best use of their refined knowledge about the problem instance. This is a very important and useful strategy to involve engineers more frequently in an optimisation process that is based on EAs.

8.3 Future work

We presented methods for developing efficient and effective EMO algorithms with the use of user-preferences, a suitable performance metric, a robust MOPSO algorithm and two real-world applications in this thesis. In this section, we point out areas that can be improved or extended in future work:

- The distance metric and EMO algorithms we introduced in this thesis were based on user-preference mechanisms found in the MCDM literature. It may be worth exploring other user-preference mechanisms that can be found in other disciplines. This may help with the design of more efficient interactive EMO algorithms.
- We visualised the many-objective solution-space of an arbitrary problem instance using a parallel axis plot. However, for certain applications a better visualisation method could be used. Especially if each objective comes from different disciplines creating a multi-disciplinary problem instance. An example for such a problem is in aerodynamic design where some objectives represent the structure of a body while others are in areas of acoustics and fuel efficiency. A good visualisation tool for the solution points can be invaluable for interactive EMO algorithms. Korhonen and Wallenius [2008], and Lotov and Miettinen [2008] describe some of these visualisation methods suitable for different application domains.
- The modified HV metric we introduced is very useful for measuring the performance of user-preference EMO algorithms on unknown problems. However, some data is

required about the *best solution front achieved so far* to calculate the HV values at each generation. It would be useful to explore possible heuristics that do not have to rely on such information.

- In our real-world applications, we observed that the user-preference EMO algorithms were able to locate better or new engineering designs than the reference design in each problem instance. However, we did not consider many of the possible constraints in each problem instance in great detail because they were outside our scope. We are very interested in extending these applications by integrating other constraints so that the candidate designs found by the EMO algorithms are close to manufacturing standards. Although this extension is application specific and might yield few advances in EMO research, it could be of practical importance.

- We are also interested in applying the user-preference EMO algorithms and performance metrics to other real-world problems, especially problem instances that can benefit from an interactive optimisation process. The aerofoil and lens design problems we presented were computationally expensive and were not considered in an interactive optimisation process. Stewart et al. [2008] describes several many-objective real-world applications with some levels of interaction. A user-preference EMO algorithm with a suitable visualisation scheme for the many-objective solutions would benefit greatly as an interactive optimisation process for real-world problems.

- We have also found some extensions in the Aerofoil application with the use of surrogate modelling [Queipo et al., 2005]. We look forward to include the distance metric based EMO algorithms within surrogate modelling for expensive optimisation problems.

It is clear from these avenues of further work that a very interesting topic could be interactive EMO algorithms. We believe that interactive EMO algorithms will be a natural and useful extension to the state-of-the-art of user-preference EMO algorithms, which has also received much attention recently [Figueira et al., 2008; Jaszkiewicz and Branke, 2008].

8.4 Summary

In this thesis, we presented an efficient framework to integrate user-preference into EMO algorithms so that they can solve many-objective problems more effectively than the state-of-the-art EMO algorithms. This novel approach (d–EMO) is computationally less expensive

and very effective in locating useful solutions on preferred regions of the objective-space as indicated by the DM. We also presented a variation of the HV metric that is suitable for measuring the performance of such user-preference EMO algorithms. Using a robust MOPSO algorithm we illustrated how this user-preference approach can be used successfully in finding solutions on real-world engineering design problems. The main characteristic of this MOPSO algorithm is that leader particles are generated using a DE rule, rather than picking particles from the swarm as leaders. This methodology enabled the MOPSO algorithm to have better chances of escaping local optimal fronts of a many-objective problem instance having multi-modal objective-spaces. The aerofoil design application and lens design application are two examples of such many-objective problems. Our proposed user-preference EMO algorithm approach to optimise these aerofoil and lens designs proved to be more efficient than existing methods because it uses fewer function evaluations. In summary, the research has made several useful contributions especially in the area of many-objective optimisation.

Appendix A

Multi-objective test problem suites

A.1 ZDT test problem suite

The general format of the test problems can be given as:

$$\begin{aligned} f_1 &= x_1 \\ f_2 &= g.h \end{aligned} \tag{A.1}$$

Where g and h are functions of $\vec{x}$.

- ZDT1

$$\begin{aligned} f_1 &= x_1 \\ g &= 1 + 9\sum_{i=2}^{N} \frac{x_i}{N-1} \\ h &= 1 - \sqrt{\frac{f_1}{g}} \end{aligned} \tag{A.2}$$

Where $N = 30$ and $x_i \in [0, 1]$. This has a convex Pareto front, which is obtained by $g(\vec{x}) = 1$.

- ZDT2

$$\begin{aligned} f_1 &= x_1 \\ g &= 1 + 9\sum_{i=2}^{N} \frac{x_i}{N-1} \\ h &= 1 - \left(\frac{f_1}{g}\right)^2 \end{aligned} \tag{A.3}$$

  Where $N = 30$ and $x_i \in [0, 1]$. This has a concave Pareto front, which is obtained by $g(\vec{x}) = 1$.

- ZDT3

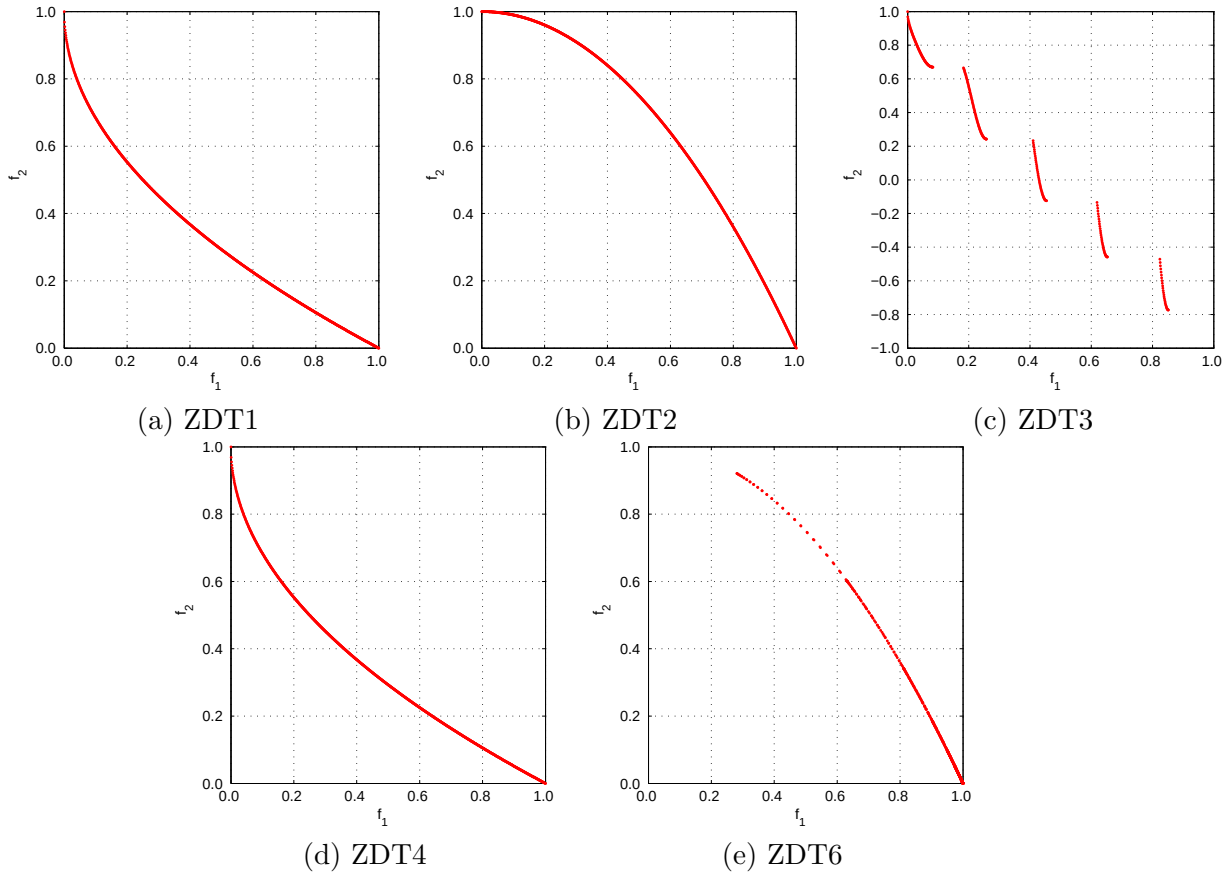
$$\begin{aligned} f_1 &= x_1 \\ g &= 1 + 9\sum_{i=2}^{N} \frac{x_i}{N-1} \\ h &= 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right)\sin\left(10\pi f_1\right) \end{aligned} \tag{A.4}$$

  Where $N = 30$ and $x_i \in [0, 1]$. This has a discontinuous Pareto front, which is obtained by $g(\vec{x}) = 1$.

- ZDT4

$$\begin{aligned} f_1 &= x_1 \\ g &= 1 + 10(N-1) + \sum_{i=2}^{N} \left(x_i^2 - 10\cos\left(4\pi x_i\right)\right) \\ h &= 1 - \sqrt{\frac{f_1}{g}} \end{aligned} \tag{A.5}$$

  Where $N = 10$, $x_0 \in [0, 1]$ and all other $x_i \in [-5, 5]$. This has a convex Pareto front, which is obtained by $g(\vec{x}) = 1$.



(a) ZDT1 (b) ZDT2 (c) ZDT3

(d) ZDT4 (e) ZDT6

Figure A.1: The Pareto fronts of the two-objective ZDT test problems.

- ZDT6

$$
\begin{aligned}
f_1 &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\
g &= 1 + 9\left(\frac{\left(\sum_{i=2}^{N} x_i\right)}{N-1}\right)^{0.25} \\
h &= 1 - \left(\frac{f_1}{g}\right)^2
\end{aligned}
\tag{A.6}
$$

Where $N = 10$ and $x_i \in [0, 1]$. This has a concave Pareto front, which is obtained by $g(\vec{x}) = 1$.

The Pareto fronts for these test problems are illustrated in Figure A.1.

A.2 DTLZ test problem suite

The function g used to generate the objectives is defined on a subset of the decision variables. This subset is given as $\vec{x}_M = [x_M, x_{M+1}, \ldots, x_N]$ and $x_i \in [0, 1]$.

- DTLZ1

$$\begin{aligned} f_1 &= \frac{1}{2} x_1 x_2 \cdots x_{M-1}(1+g) \\ f_2 &= \frac{1}{2} x_1 x_2 \cdots (1-x_{M-1})(1+g) \\ &\vdots \\ f_{M-1} &= \frac{1}{2} x_1 (1-x_2)(1+g) \\ f_M &= \frac{1}{2}(1-x_1)(1+g) \\ g &= 100 \left[|\vec{x}_M| + \sum_{x_i \in \vec{x}_M} (x_i - 0.5)^2 - \cos\left(20\pi(x_i - 0.5)\right) \right] \end{aligned} \tag{A.7}$$

  Where $N = M + 4$ and $|\vec{x}_M| = 5$. The Pareto front is linear having $\vec{x}_M = 0$ and $\sum_{i=1}^{M} f_i = 0.5$. DTLZ1 is multi-modal having $11^5 - 1$ local fronts.

- DTLZ2

$$\begin{aligned} f_1 &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\cdots\cos\left(x_{M-2}\frac{\pi}{2}\right)\cos\left(x_{M-1}\frac{\pi}{2}\right) \\ f_2 &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\cdots\cos\left(x_{M-2}\frac{\pi}{2}\right)\sin\left(x_{M-1}\frac{\pi}{2}\right) \\ f_3 &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\cdots\sin\left(x_{M-2}\frac{\pi}{2}\right) \\ &\vdots \\ f_{M-1} &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\sin\left(x_2\frac{\pi}{2}\right) \\ f_M &= (1+g)\sin\left(x_1\frac{\pi}{2}\right) \\ g &= \sum_{x_i \in \vec{x}_M} (x_i - 0.5)^2 \end{aligned} \tag{A.8}$$

  Where $N = M + 9$ and $|\vec{x}_M| = 10$. The Pareto front is convex having $\vec{x}_M = 0.5$ and $\sum_{i=1}^{M} (f_i)^2 = 1$.

- DTLZ3

$$
\begin{aligned}
f_1 &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\cdots\cos\left(x_{M-2}\frac{\pi}{2}\right)\cos\left(x_{M-1}\frac{\pi}{2}\right) \\
f_2 &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\cdots\cos\left(x_{M-2}\frac{\pi}{2}\right)\sin\left(x_{M-1}\frac{\pi}{2}\right) \\
f_3 &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\cos\left(x_2\frac{\pi}{2}\right)\cdots\sin\left(x_{M-2}\frac{\pi}{2}\right) \\
&\vdots \\
f_{M-1} &= (1+g)\cos\left(x_1\frac{\pi}{2}\right)\sin\left(x_2\frac{\pi}{2}\right) \\
f_M &= (1+g)\sin\left(x_1\frac{\pi}{2}\right) \\
g &= 100\left[|\vec{x}_M| + \sum_{x_i\in\vec{x}_M}(x_i-0.5)^2 - \cos\left(20\pi(x_i-0.5)\right)\right]
\end{aligned}
\tag{A.9}
$$

Where $N = M + 9$ and $|\vec{x}_M| = 10$. The Pareto front is convex having $\vec{x}_M = 0.5$ and $\sum_{i=1}^{M}(f_i)^2 = 1$. DTLZ3 is multi-modal having $3^{10} - 1$ local optimal fronts.

- DTLZ4

$$
\begin{aligned}
f_1 &= (1+g)\cos\left(x_1^\alpha\frac{\pi}{2}\right)\cos\left(x_2^\alpha\frac{\pi}{2}\right)\cdots\cos\left(x_{M-2}^\alpha\frac{\pi}{2}\right)\cos\left(x_{M-1}^\alpha\frac{\pi}{2}\right) \\
f_2 &= (1+g)\cos\left(x_1^\alpha\frac{\pi}{2}\right)\cos\left(x_2^\alpha\frac{\pi}{2}\right)\cdots\cos\left(x_{M-2}^\alpha\frac{\pi}{2}\right)\sin\left(x_{M-1}^\alpha\frac{\pi}{2}\right) \\
f_3 &= (1+g)\cos\left(x_1^\alpha\frac{\pi}{2}\right)\cos\left(x_2^\alpha\frac{\pi}{2}\right)\cdots\sin\left(x_{M-2}^\alpha\frac{\pi}{2}\right) \\
&\vdots \\
f_{M-1} &= (1+g)\cos\left(x_1^\alpha\frac{\pi}{2}\right)\sin\left(x_2^\alpha\frac{\pi}{2}\right) \\
f_M &= (1+g)\sin\left(x_1^\alpha\frac{\pi}{2}\right) \\
g &= \sum_{x_i\in\vec{x}_M}(x_i-0.5)^2
\end{aligned}
\tag{A.10}
$$

Where $\alpha = 100$, $N = M + 9$ and $|\vec{x}_M| = 10$. The Pareto front is convex having $\sum_{i=1}^{M}(f_i)^2 = 1$.

- DTLZ5

$$
\begin{aligned}
f_1 &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\cos\left(\theta_2\frac{\pi}{2}\right)\cdots\cos\left(\theta_{M-2}\frac{\pi}{2}\right)\cos\left(\theta_{M-1}\frac{\pi}{2}\right) \quad \text{(A.11)}\\
f_2 &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\cos\left(\theta_2\frac{\pi}{2}\right)\cdots\cos\left(\theta_{M-2}\frac{\pi}{2}\right)\sin\left(\theta_{M-1}\frac{\pi}{2}\right)\\
f_3 &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\cos\left(\theta_2\frac{\pi}{2}\right)\cdots\sin\left(\theta_{M-2}\frac{\pi}{2}\right)\\
&\vdots\\
f_{M-1} &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\sin\left(\theta_2\frac{\pi}{2}\right)\\
f_M &= (1+g)\sin\left(\theta_1\frac{\pi}{2}\right)\\
g &= \sum_{x_i\in\vec{x}_M}(x_i-0.5)^2\\
\theta_i &= \frac{\pi}{4(1+g)}(1+2gx_i) \text{ for } i=2,\ldots,M-1\\
\theta_1 &= x_1\frac{\pi}{2}
\end{aligned}
$$

Where $N = M+9$ and $|\vec{x}_M| = 10$. The Pareto front is convex having $\sum_{i=1}^{M}(f_i)^2 = 1$. Here, the Pareto front is an area spanning $M-1$ dimensions in an $M \geq 3$ objective-space.

- DTLZ6

$$
\begin{aligned}
f_1 &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\cos\left(\theta_2\frac{\pi}{2}\right)\cdots\cos\left(\theta_{M-2}\frac{\pi}{2}\right)\cos\left(\theta_{M-1}\frac{\pi}{2}\right) \quad \text{(A.12)}\\
f_2 &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\cos\left(\theta_2\frac{\pi}{2}\right)\cdots\cos\left(\theta_{M-2}\frac{\pi}{2}\right)\sin\left(\theta_{M-1}\frac{\pi}{2}\right)\\
f_3 &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\cos\left(\theta_2\frac{\pi}{2}\right)\cdots\sin\left(\theta_{M-2}\frac{\pi}{2}\right)\\
&\vdots\\
f_{M-1} &= (1+g)\cos\left(\theta_1\frac{\pi}{2}\right)\sin\left(\theta_2\frac{\pi}{2}\right)\\
f_M &= (1+g)\sin\left(\theta_1\frac{\pi}{2}\right)\\
g &= \sum_{x_i\in\vec{x}_M}x_i^{0.1}\\
\theta_i &= \frac{\pi}{4(1+g)}(1+2gx_i) \text{ for } i=2,\ldots,M-1\\
\theta_1 &= x_1\frac{\pi}{2}
\end{aligned}
$$

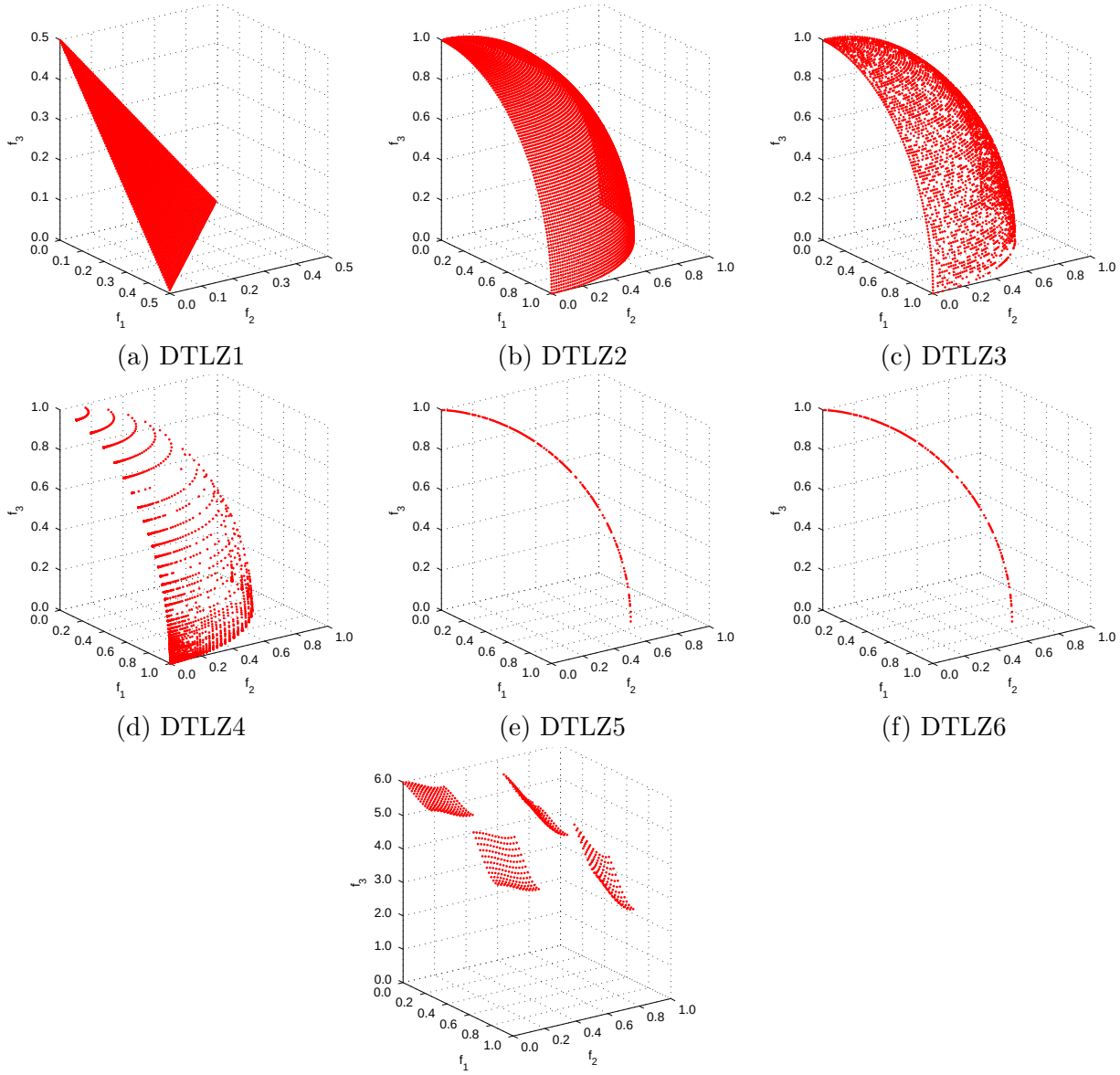
Where $N = M+9$ and $|\vec{x}_M| = 10$. The Pareto front is convex having $\sum_{i=1}^{M}(f_i)^2 = 1$. The Pareto front is similar to the one of DTLZ5, which is a shape having $M-1$ dimensions in a $M \geq 3$ objective-space.

- DTLZ7

$$\begin{aligned} f_1 &= x_1 \\ f_2 &= x_2 \\ &\vdots \\ f_{M-1} &= x_{M-1} \\ f_M &= (1+g)h \\ g &= 1 + \frac{9}{|\vec{x}_M|}\sum_{x_i \in \vec{x}_M} x_i \\ h &= M - \sum_{i=1}^{M-1}\left[\frac{f_i}{1+g}\left(1+\sin\left(3\pi f_i\right)\right)\right] \end{aligned} \tag{A.13}$$

Where $N = M+19$ and $|\vec{x}_M| = 20$. The Pareto front has 2^{M-1} number of disconnected regions having $\vec{x}_M = 0$.

The Pareto fronts for the DTLZ test problems for three-objectives are illustrated in Figure A.2.



(a) DTLZ1 (b) DTLZ2 (c) DTLZ3

(d) DTLZ4 (e) DTLZ5 (f) DTLZ6

(g) DTLZ7

Figure A.2: Pareto fronts of the three-objective DTLZ test problems.

A.3 WFG test problem suite

The general format of the test suite can be give as:

$$\begin{aligned}
\text{given } \vec{x} &= \{x_1, \ldots, x_k, x_{k+1}, \ldots, x_N\} \qquad (A.14)\\
\text{minimise } f_{m=1\ldots M} &= z_M + S_m h_m\left(z_1, \ldots, z_{M-1}\right)\\
\text{where } \vec{z} &= \{z_1, \ldots, z_{M-1}\}\\
&= \left\{\max\left(t^p_M, A_1\right)\left(t^p_1 - 0.5\right) + 0.5, \ldots, \max\left(t^p_{M-1}, A_1\right)\left(t^p_{M-1} - 0.5\right) + 0.5, t^p_M\right\}\\
\vec{t^p} &= \left\{t^p_1, \ldots, t^p_M\right\} \leftarrow\!\shortmid\, \vec{t^{p-1}} \ldots \leftarrow\!\shortmid\, \vec{t^1} \leftarrow\!\shortmid\, \vec{x}_{[0,1]}\\
\vec{x}_{[0,1]} &= \left\{x_{1,[0,1]}, \ldots, x_{N,[0,1]}\right\}\\
&= \left\{\frac{x_1}{x_{1,max}}, \ldots, \frac{x_N}{x_{N,max}}\right\}
\end{aligned}$$

Where $\vec{z}$ is a set of M underlying parameters, where $z_i \in [0, 1]$. z_M is a distance parameter, while $z_{1\ldots M-1}$ are position parameters. $\vec{x}$ is a set of $k + l = N \geq M$ working parameters, where $x_i \in [0, x_{i,max}]$ and $x_{i,max} > 0$. k are the position parameters and l are distance parameters. In this thesis we used these $k + l$ parameters to define the decision variables. $A_{1\ldots M-1} \in \{0, 1\}$ are degeneracy constants. For every $A_i = 0$ the dimensions of the Pareto front is reduced by one. $h_{1\ldots M}$ are shape functions and $S_{1\ldots M}$ are scaling constants. $\vec{t^{1\ldots p}}$ are transition vectors and $\leftarrow\!\shortmid$ shows the transition.

The shape functions

The shape functions defined as below, where A, α and β are constants:

- Linear

$$\begin{aligned}
linear_1(z_1, \ldots, z_{M-1}) &= \prod_{i=1}^{M-1} z_i \qquad (A.15)\\
linear_{m=2\ldots M-1}(z_1, \ldots, z_{M-1}) &= \left(\prod_{i=1}^{M-1} z_i\right)(1 - z_{M-m+1})\\
linear_M(z_1, \ldots, z_{M-1}) &= 1 - z_1
\end{aligned}$$

When $h_{m=1\ldots M} = linear_m$ the Pareto front is a hyper-plane having $\sum_{m=1}^{M} h_m = 1$.

- Convex

$$\begin{aligned} convex_1(z_1,\ldots,z_{M-1}) &= \prod_{i=1}^{M-1}\left(1-\cos\left(z_i\frac{\pi}{2}\right)\right) \\ convex_{m=2\ldots M-1}(z_1,\ldots,z_{M-1}) &= \left(\prod_{i=1}^{M-1}\left(1-\cos\left(z_i\frac{\pi}{2}\right)\right)\right)\left(1-\sin\left(z_{M-m+1}\frac{\pi}{2}\right)\right) \\ convex_M(z_1,\ldots,z_{M-1}) &= 1-\sin\left(z_1\frac{\pi}{2}\right) \end{aligned} \tag{A.16}$$

  When $h_{m=1\ldots M} = convex_m$ the Pareto front is purely convex.

- Concave

$$\begin{aligned} concave_1(z_1,\ldots,z_{M-1}) &= \prod_{i=1}^{M-1}\sin\left(z_i\frac{\pi}{2}\right) \\ concave_{m=2\ldots M-1}(z_1,\ldots,z_{M-1}) &= \left(\prod_{i=1}^{M-1}\sin\left(z_i\frac{\pi}{2}\right)\right)\cos\left(z_{M-m+1}\frac{\pi}{2}\right) \\ concave_M(z_1,\ldots,z_{M-1}) &= \cos\left(z_1\frac{\pi}{2}\right) \end{aligned} \tag{A.17}$$

  When $h_{m=1\ldots M} = convex_m$ the Pareto front is purely concave having $\sum_{m=1}^{M}(h_m)^2 = 1$.

- Mixed convex/concave ($\alpha > 0, A \in \{1,2,\ldots\}$)

$$mixed_M(z_1,\ldots,z_{M-1}) = \left(1-z_1-\frac{\cos\left(2A\pi z_1+\frac{\pi}{2}\right)}{2A\pi}\right)^{\alpha} \tag{A.18}$$

  The number of concave or convex segments is controlled by A. If $\alpha > 1$ overall shape is convex. If $\alpha < 1$ it is concave. If $\alpha = 1$ it is linear.

- Disconnected ($\alpha, \beta > 0, A \in \{1,2,\ldots\}$)

$$disc_M(z_1,\ldots,z_{M-1}) = 1-(z_1)^{\alpha}\cos^2\left(A(z_1)^{\beta}\pi\right) \tag{A.19}$$

  The number of disconnected segments is controlled by A. If $\alpha > 1$ overall shape is convex. If $\alpha < 1$ it is concave. If $\alpha = 1$ it is linear. β influences the locations of disconnection.

The transformation functions

Associated with these shape functions are some transformation functions, which are used to build the transition vectors. The primary parameters are given as $\vec{y} = [y_1, \ldots, y_{|\vec{y}|}]$, where $y_i \in [0, 1]$ and $y \in [0, 1]$. A, B, C, α and β are constants. The transformation functions are:

- Bias: Polynomial $(\alpha > 0, \alpha \neq 1)$

$$b\_poly(y, \alpha) = y^{\alpha} \tag{A.20}$$

  When $\alpha > 1$, y is bias towards zero else towards one.

- Bias: Flat region $(A, B, C \in [0, 1], B < C, B = 0 \Rightarrow A = 0 \wedge C \neq 1, C = 1 \Rightarrow A = 1 \wedge B \neq 0)$

$$b\_flat(y, A, B, C) = A + \min(0, \lfloor y - B \rfloor) \frac{A(B-y)}{B} - \min(0, \lfloor C - y \rfloor) \frac{(1_A)(y-C)}{1-C} \tag{A.21}$$

  The values of y between B and C are the flat regions.

- Bias: Parameter dependant $(A \in (0, 1), 0 < B < C)$

$$\begin{aligned} b\_param(y, \vec{y'}, A, B, C) &= y^{B+(C-B)v(u(\vec{y'}))} \\ v(u(\vec{y'})) &= A - (1 - 2u(\vec{y'})) \left| \left\lfloor 0.5 - u(\vec{y'}) \right\rfloor + A \right| \end{aligned} \tag{A.22}$$

  $\vec{y'}$ is a secondary parameter vector in $[0, 1]$, u is a reduction function, where $u(\vec{y'}) \in [0, 0.5]$ are mapped to $[B, B + (C - B)A]$ and $u(\vec{y'}) \in [0.5, 1]$ are mapped to $[B + (C - B)A, C]$.

- Shift: Linear $(A \in (0, 1))$

$$s\_linear(y, A) = \frac{|y - A|}{|\lfloor A - y \rfloor + A|} \tag{A.23}$$

  A is a value for which y is mapped to zero.

- Shift: Deceptive ($A \in (0,1), 0 < B \ll 1, 0 < C \ll 1, A - B > 0, A + B < 1$)

$$s\_decept(y, A, B, C) = 1 + (|y - A| - B)\times \left(\frac{\lfloor y - A + B \rfloor \left(1 - C + \frac{A-B}{B}\right)}{A - B} + \frac{\lfloor A + B - y \rfloor \left(1 - C + \frac{1-A-B}{B}\right)}{1 - A - B} + \frac{1}{B} \right) \tag{A.24}$$

A is a value for which y is mapped to zero, and B is the aperture size of the leading to the global minimum A. C is the deceptive minimum.

- Shift: Multi-modal ($A \in \{1, 2, \ldots\}, B \geq 0, (4A + 2)\pi \geq 4B, C \in (0, 1)$)

$$s\_multi(y, A, B, C) = \frac{1 + \cos\left[(4A + 2)\pi \left(0.5 - \frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)\right] + 4B \left(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)^2}{B + 2} \tag{A.25}$$

A controls the number of minimum, B controls the magnitude of the multi-modality and C the value for y is mapped to zero. When $B = 0$, $2A + 1$ values of y are mapped to zero, and when $B \neq 0$, $2A$ values are local minimum and one global minimum at C.

- Reduction: Weighted sum ($|\vec{w}| = |\vec{y}|, w_1, \ldots, w_{|\vec{y}|} > 0$)

$$r\_sum(\vec{y}, \vec{w}) = \frac{\sum_{i=1}^{|\vec{y}|} w_i y_i}{\sum_{i=1}^{|\vec{y}|} w_i} \tag{A.26}$$

- Reduction: Non-separable ($A \in \{1, \ldots, |\vec{y}|\}, |\vec{y}| \bmod A = 0$)

$$r\_nonsep(\vec{y}, A) = \frac{\sum_{i=1}^{|\vec{y}|} \left(y_i + \sum_{j=0}^{A-2} \left| y_i + y_{1+(i+j) \bmod |\vec{y}|} \right| \right)}{\frac{|\vec{y}|}{A} \left\lceil \frac{A}{2} \right\rceil \left(1 + 2A + -2 \left\lceil \frac{A}{2} \right\rceil\right)} \tag{A.27}$$

$r\_nonsep(\vec{y}, 1) = r\_sum(\vec{y}, \{1, \ldots, 1\})$.

The test suite

The WFG problems are then constructed by having for any $\vec{t^i}$, $\vec{y} = \vec{t^{i-1}}$ and for $\vec{t^1}$, $\vec{y} = \vec{x}_{[0,1]} = \left\{\frac{x_1}{2}, \ldots, \frac{x_n}{2n}\right\}$. Also $x_{i=1\ldots N,max} = 2i$, $S_{m=1\ldots M} = 2m$, $A_1 = 1$ and $A_{2\ldots M-1} = 0$ for WFG3 and 1 for others. The test suite is given as:

- WFG1

$$\begin{aligned}
h_{m=1\ldots M-1} &= convex_m \\
h_M &= mixed_M \quad \alpha = 1 \quad A = 5 \\
t^1_{i=1\ldots k} &= y_i \\
t^1_{i=k+1\ldots N} &= s_linear(y_i, 0.35) \\
t^2_{i=1\ldots k} &= y_i \\
t^2_{i=k+1\ldots N} &= b_flat(y_i, 0.8, 0.75, 0.85) \\
t^3_{i=1\ldots N} &= b_poly(y_i, 0.02) \\
t^4_{i=1\ldots k} &= r_sum\left(\left\{\frac{y_{(i-1)k}}{(M-1)+1}, \ldots, \frac{y_{ik}}{(M-1)}\right\}, \left\{\frac{2((i-1)k)}{(M-1)+1}, \ldots, \frac{2ik}{(M-1)}\right\}\right) \\
t^4_{i=k+1\ldots N} &= r_sum(\{y_{k+1}, \ldots, y_N\}, \{2(k+1), \ldots, 2N\})
\end{aligned} \tag{A.28}$$

- WFG2

$$\begin{aligned}
h_{m=1\ldots M-1} &= convex_m \\
h_M &= disc_M \quad \alpha = \beta = 1 \quad A = 5 \\
t^1_i &= \text{same as WFG1 } t^1_i \\
t^2_{i=1\ldots k} &= y_i \\
t^2_{i=k+1\ldots k+\frac{l}{2}} &= r_nonsep(\{y_{k+2(i-k)-1}, y_{k+2(i-k)}\}, 2) \\
t^3_{i=1\ldots M-1} &= r_sum\left(\left\{\frac{y_{(i-1)k}}{(M-1)+1}, \ldots, \frac{y_{ik}}{(M-1)}\right\}, \{1, \ldots, 1\}\right) \\
t^3_M &= r_sum\left(\left\{y_{k+1}, \ldots, y_{k+\frac{l}{2}}\right\}, \{1, \ldots, 1\}\right)
\end{aligned} \tag{A.29}$$

- WFG3

$$\begin{aligned}
h_{m=1\ldots M} &= linear_m \\
t^{1,2,3}_i &= \text{same as WFG2 } t^{1,2,3}_i
\end{aligned} \tag{A.30}$$

- WFG4

$$\begin{aligned} h_{m=1\ldots M} &= concave_m \\ t^1_{i=1\ldots N} &= s\_multi(y_i, 30, 10, 0.35) \\ t^2_{i=1\ldots M-1} &= r\_sum\left(\left\{\frac{y_{(i-1)k}}{(M-1)+1}, \ldots, \frac{y_{ik}}{(M-1)}\right\}, \{1, \ldots, 1\}\right) \\ t^2_M &= r\_sum\left(\{y_{k+1}, \ldots, y_N\}, \{1, \ldots, 1\}\right) \end{aligned} \tag{A.31}$$

- WFG5

$$\begin{aligned} h_{m=1\ldots M} &= concave_m \\ t^1_{i=1\ldots N} &= s\_decept(y_i, 0.35, 0.001, 0.05) \\ t^2_i &= \text{same as WFG4 } t^2_i \end{aligned} \tag{A.32}$$

- WFG6

$$\begin{aligned} h_{m=1\ldots M} &= concave_m \\ t^1_i &= \text{same as WFG1 } t^1_i \\ t^2_{i=1\ldots M-1} &= r\_nonsep\left(\left\{\frac{y_{(i-1)k}}{(M-1)+1}, \ldots, \frac{y_{ik}}{(M-1)}\right\}, \frac{k}{(M-1)}\right) \\ t^2_M &= r\_nonsep\left(\{y_{k+1}, \ldots, y_N\}, l\right) \end{aligned} \tag{A.33}$$

- WFG7

$$\begin{aligned} h_{m=1\ldots M} &= concave_m \\ t^1_{i=1\ldots k} &= b\_param\left(y_i, r\_sum\left(\{y_{i+1}, \ldots, y_N\}, \{1, \ldots, 1\}\right), \frac{0.98}{49.98}, 0.02, 50\right) \\ t^1_{i=k+1\ldots N} &= y_i \\ t^2_i &= \text{same as WFG1 } t^1_i \\ t^3_i &= \text{same as WFG4 } t^2_i \end{aligned} \tag{A.34}$$

- WFG8

$$
\begin{aligned}
h_{m=1\ldots M} &= concave_m \qquad \text{(A.35)}\\
t^1_{i=1\ldots k} &= y_i\\
t^1_{i=k+1\ldots N} &= b\_param\left(y_i, r\_sum\left(\{y_1,\ldots,y_{i-1}\},\{1,\ldots,1\}\right), \frac{0.98}{49.98}, 0.02, 50\right)\\
t^2_i &= \text{same as WFG1 } t^1_i\\
t^3_i &= \text{same as WFG4 } t^2_i
\end{aligned}
$$

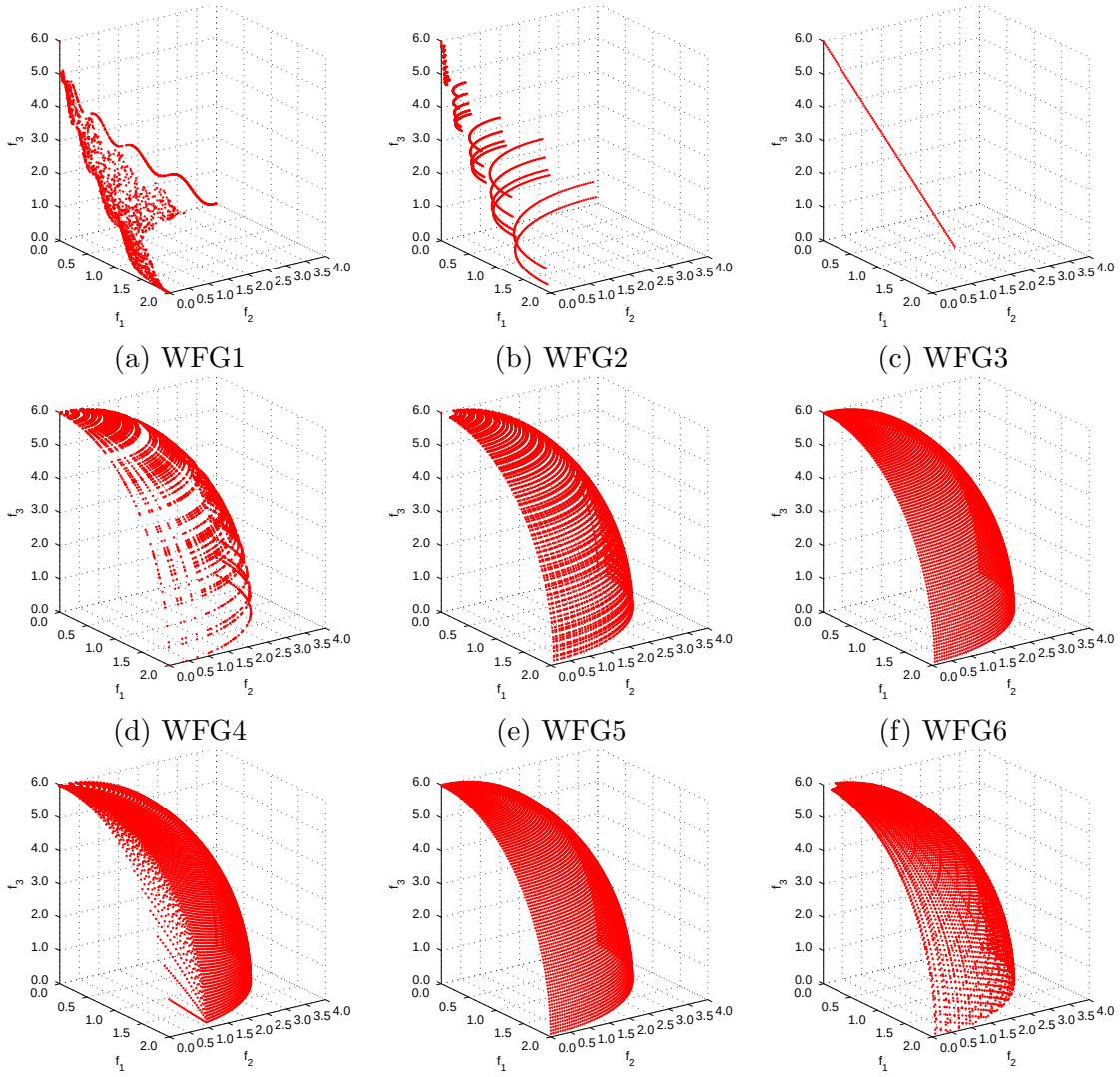
- WFG9

$$
\begin{aligned}
h_{m=1\ldots M} &= concave_m \qquad \text{(A.36)}\\
t^1_{i=1\ldots N-1} &= b\_param\left(y_i, r\_sum\left(\{y_{i+1},\ldots,y_N\},\{1,\ldots,1\}\right), \frac{0.98}{49.98}, 0.02, 50\right)\\
t^1_N &= y_N\\
t^2_{i=1\ldots k} &= s\_decept(y_i, 0.35, 0.001, 0.05)\\
t^2_{i=k+1\ldots N} &= s\_multi(y_i, 30, 95, 0.35)\\
t^3_i &= \text{same as WFG6 } t^2_i
\end{aligned}
$$

A decision vector is Pareto optimal on,

- WFG1-7 if and only if $x_i = 2i \times 0.35$
- WFG8 if and only if $x_i = 2i \times 0.35^{\left(\frac{1}{0.02+49.98\left(\frac{0.98}{49.98}-(1-2u)\left\lfloor 0.5-u\right\rfloor+\frac{0.98}{49.98}\right\rfloor\right)}\right)}$
- WFG9 if and only if $x_i = 2i \times \begin{cases} 0.35^{\left(\frac{1}{0.02+1.96u}\right)}, & \text{where } i \neq N \\ 0.35, & \text{where } i = N \end{cases}$

Where $i = k{+}1,\ldots,N$ (the distance variables) and $u = r\_sum\left(\{x_1,\ldots,x_{i-1}\},\{1,\ldots,1\}\right)$. The Pareto fronts of the three-objective WFG problems are illustrated in Figure A.3.



(a) WFG1 (b) WFG2 (c) WFG3

(d) WFG4 (e) WFG5 (f) WFG6

(g) WFG7 (e) WFG8 (f) WFG9

Figure A.3: Pareto fronts of the three-objective WFG test problems.

Bibliography

H. Abbass, R. Sarker, and C. Newton. PDE: a Pareto-frontier Differential Evolution approach for multi-objective optimization problems. In *Congress on Evolutionary Computation (CEC)*, pages 971–978, Seoul , South Korea, 2001. IEEE Computer Society Press.

R. Allmendinger. Reference Point-based Particle Swarm Optimization Using a Steady-State Approach. Master's thesis, University of Karlsruhe, Institute AIFB, Karlsruhe, Germany, 2008.

W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. Technical Report TR-97-9, Institute for Computer Applications in Science and Engineering (ICASE), 1997.

H. Bach and N. Neuroth. *The Properties of Optical Glass*. Springer-Verlag Telos, 1995. ISBN 978-0387583570.

T. Bäck. *Evolutionary algorithms in theory and practise: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996. ISBN 978-0195099713.

R. Balling. The Maximin Fitness Function; Multi-objective City and Regional Planning. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 1–15, Faro, Portugal, 2003. Springer.

J. Beaulieu, C. Gagné, and M. Parizeau. Lens System Design And Re-engineering With Evolutionary Algorithms. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 155–162, New York, NY, USA, 2002. Morgan Kaufmann Publishers Inc.

M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 1999. ISBN 978-0521642224.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0521833783.

J. Branke and K. Deb. Integrating user preferences into evolutionary multi-objective optimization. In *Knowledge Incorporation in Evolutionary Computation*, pages 461–478. Springer, 2004.

J. Branke, T. Kaußler, and H. Schmeck. Guidance in Evolutionary Multi-Objective Optimization. *Advances in Engineering Software*, 32:499–507, 2001.

D. Brockhoff and E. Zitzler. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In *Parallel Problem Solving from Nature (PPSN)*, pages 533–542, Reykjavik, Iceland, 2006. Springer.

R. Carrese, U. K. Wickramasinghe, H. Winarto, and J. Watmuff. Preference-Based Optimization for Transonic Airfoil Shape Design over a Multi-Operational Spectrum. *Journal of Aircraft from American Institute of Aeronautics and Astronautics (AIAA)*, 2010 (under review).

P. Chandrashekarappa and R. Duvigneau. Metamodel-assisted particle swarm optimization and application to aerodynamic shape optimization. Technical Report RR-6397, OPALE - INRIA Sophia Antipolis / INRIA Rhône-Alpes - INRIA - CNRS : UMR6621 - Université de Nice Sophia-Antipolis, 2007.

M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6: 58–73, 2002.

C. Coello Coello and M. Lechuga. MOPSO: a proposal for multiple objective particle swarm optimization. In *World Congress on Computational Intelligence (WCCI)*, pages 1051–1056, Honolulu, HI, USA, 2002. IEEE Computer Society Press.

C. A. Coello Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *Congress on Evolutionary Computation (CEC)*, pages 30–37, San Diego, CA, USA, 2000. IEEE Computer Society Press.

C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007. ISBN 978-0387332543.

D. W. Corne and J. D. Knowles. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 773–780, London, UK, 2007. ACM Press.

D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 283–290, San Francisco, CA, USA, 2001. Morgan Kaufmann.

C. Darwin. *The Origin of Species*. Signet Classics, 2003 (first published in 1859). ISBN 978-0451529060.

K. Deb. Solving goal programming problems using multi-objective genetic algorithms. In *Congress on Evolutionary Computation (CEC)*, pages 77–84. IEEE Computer Society Press, 1999.

K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., 2001. ISBN 978-0470743614.

K. Deb. Introduction to Evolutionary Multiobjective Optimization. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 3, pages 59–96. Springer-Verlag, 2008.

K. Deb and R. B. Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9:115–148, 1995.

K. Deb and M. Goyal. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics*, 26:30–45, 1996.

K. Deb and S. Jain. Running performance Metrics for evolutionary multi-objective optimization. Technical Report 2002004, Kanpur Genetic Algorithm Laboratory, Indian Institute of Technology Kanpur, 2002.

K. Deb and A. Kumar. Light Beam Search Based Multi-Objective Optimization Using Evolutionary Algorithms. In *Congress on Evolutionary Computation (CEC)*, pages 2125–2132, Singapore, 2007a. IEEE Computer Society Press.

K. Deb and A. Kumar. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 781–788, London, UK, 2007b. ACM Press.

K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

K. Deb, M. Mohan, and S. Mishra. Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501–525, 2005a.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In *Evolutionary Multiobjective Optimization (EMO): Theoretical Advances and Applications*, chapter 6, pages 105–145. Springer, 2005b.

K. Deb, J. Sundar, U. B. R. N., and S. Chaudhuri. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. *International Journal of Computational Intelligence Research*, 2(3):635–642, 2006.

F. di Pierro. *Many-Objective Optimization and Applications to Water Engineering: Theory and Practice*. LAP Lambert Academic Publishing, 2010. ISBN 978-3838303178.

N. Drechsler, R. Drechsler, and B. Becker. Multi-objective Optimisation Based on Relation Favour. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 154–166, Zurich, Switzerland, 2001. Springer.

M. Drela and M. B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *American Institute of Aeronautics and Astronautics (AIAA)*, 25(10):1347–1355, 1987.

J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba. jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical Report ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, 2006.

R. Duvigneau, B. Chaigne, and J.-A. Désidéri. Multi-Level Parameterization for Shape Optimization in Aerodynamics and Electromagnetics using a Particle Swarm Optimization Algorithm. Technical Report RR-6003, OPALE - INRIA Sophia Antipolis / INRIA Rhône-Alpes - INRIA - CNRS : UMR6621 - Université de Nice Sophia-Antipolis, 2006.

M. Ehrgott. *Lecture Notes in Economics and Mathematical Systems: Multicriteria optimization*. Springer-Verlag, 2000. ISBN 978-3642059759.

M. Ehrgott and X. Gandibleux. *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Survey*, volume 52 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, 2002.

A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003. ISBN 978-3642072857.

M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 62–76, Guanajuato, Mexico, 2005. Springer.

J. Fieldsend, R. Everson, and S. Singh. Using unconstrained elite archives for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 7:305–323, 2003.

J. R. Figueira, S. Greco, V. Mousseau, and R. Słowiński. Interactive Multiobjective Optimization Using a Set of Additive Value Functions. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 4, pages 97–119. Springer-Verlag, 2008.

M. Fleischer. The Measure of Pareto Optima Applications to Multi-objective Metaheuristics. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 519–533, Faro, Portugal, 2003. Springer.

P. J. Fleming, R. C. Purshouse, and R. J. Lygoe. Many-Objective Optimization: An Engineering Design Perspective. *Evolutionary Multi-Criterion Optimization*, pages 14–32, 2005.

D. Fogel. An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5(1):3–14, 1994.

C. Fonseca and P. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *International Conference on Genetic Algorithms*, pages 416–423, Urbana-Champaign, IL, USA, 1993. Morgan Kaufmann Publishers Inc.

P. C. Fourie and A. A. Groenwold. Structural and multidisciplinary optimization. *The particle swarm optimization algorithm in size and shape optimization*, 23(4):259–267, 2002.

C. Gagné, J. Beaulieu, M. Parizeau, and S. Thibault. Human-competitive lens system design with evolution strategies. *Applied Soft Computing*, 8(4):1439–1452, 2008.

Z.-F. Hao, G.-H. Guo, and H. Huang. A Particle Swarm Optimization Algorithm with Differential Evolution. *International Conference on Machine Learning and Cybernetics (ICMLC)*, 2:1031–1035, 2007.

J. Haslinger and R. A. E. Makinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation*. Society for Industrial and Applied Mathematics, 2003. ISBN 978-0898715361.

T. Hendtlass. A Combined Swarm Differential Evolution Algorithm for Optimization Problems. In *International conference on Industrial and engineering applications of artificial intelligence and expert systems (IEA/AIE)*, pages 11–18, Budapest, Hungary, 2001. Springer-Verlag.

R. M. Hicks and P. A. Henne. Wing Design By Numerical Optimisation. *Journal of Aircraft from American Institute of Aeronautics and Astronautics (AIAA)*, 15(7):407–412, 1978.

J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, 1992. ISBN 978-0262581110.

X. Hu, R. Eberhart, and Y. Shi. Particle swarm with extended memory for multiobjective optimization. *Swarm Intelligence Symposium (SIS)*, pages 193–197, 2003.

V. L. Huang, P. N. Suganthan, A. K. Qin, and S. Baskar. Multiobjective Differential Evolution with External Archive and Harmonic Distance-Based Diversity Measure. Technical Report 2005, School of Electrical and Electronic Engineering, Nanyang Technological University, 2005.

S. Huband, P. Hingston, L. Barone, and R. L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10: 477–506, 2006.

E. J. Hughes. Evolutionary many-objective optimisation: many once or one many? In *Congress on Evolutionary Computation (CEC)*, pages 222–227, Edinburgh, UK, 2005. IEEE Computer Society Press.

E. J. Hughes. Radar Waveform Optimisation as a Many-Objective Application Benchmark. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 700–714, Matsushima, Japan, 2006. Springer.

E. J. Hughes. Fitness Assignment Methods for Many-Objective Problems. In *Multi-Objective Problem Solving from Nature: From Concepts to Applications*, chapter 4, pages 307–329. Springer, 2008.

A. W. Iorio and X. Li. Solving Rotated Multi-objective Optimization Problems Using Differential Evolution. In *Australian Conference on Artificial Intelligence (AI)*, pages 861–872, Cairns, Australia, 2004. Springer.

A. Jameson and J. Reuther. Control theory based airfoil design using the Euler equations. In *American Institute of Aeronautics and Astronautics (AIAA)*, pages 1–17, Seattle, WA, USA, 1994. AIAA.

A. Jaszkiewicz and J. Branke. Interactive Multiobjective Evolutionary Algorithms. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 7, pages 179–193. Springer-Verlag, 2008.

A. Jaszkiewicz and R. Słowiński. The Light Beam Search approach -an overview of methodology and applications. *European Journal of Operational Research*, 113:300–314, 1999.

M. T. Jensen. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):503–515, 2003.

Y. Jin and B. Sendhoff. Incorporating of Fuzzy Preferences into Evolutionary Multiobjective Optimization. In *Asia-Pacific Conference on Simulated Evolution and Learning (SEAL)*, pages 26–30, Singapore, 2002. Springer.

Y. Jin and B. Sendhoff. A systems approach to evolutionary multiobjective structural optimization and beyond. *IEEE Computational Intelligence Magazin*, 4(3):62–76, 2009.

S. Joseph. *Lens design and optimization using multi-objective evolutionary algorithms*. PhD thesis, Department of Mathematics and Computer Science, University of Missouri, St. Louis, USA, 2005.

S. Joseph, H. Kang, and U. Chakraborty. Lens Optimization in a Multi-objective Framework. In *International Conference on Recent Advances in Soft Computing*, pages 309–314, Canterbury, UK, 2006a. Springer.

S. Joseph, H. Kang, and U. Chakraborty. Lens Optimization in a Classical-Evolutionary Hybrid Framework. In *International Mendel Conference on Soft Computing (MENDEL)*, pages 45–50, Brno, Czech Republic, 2006b. Mendel.

S. Joseph, H. W. Kang, and U. K. Chakraborty. Optical Design with Epsilon-Dominated Multi-objective Evolutionary Algorithm. In *International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)*, pages 77–84, Warsaw, Poland, 2007. Springer.

A. Keane and P. Nair. *Computational Approaches for Aerospace Design: The Pursuit of Excellence*. John Wiley Sons, Ltd., 2005. ISBN 978-0470855409.

J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., 2001. ISBN 978-1558605954.

M. S. Khurana, A. K. Sinha, and H. Winarto. Multi-mission re-configurable UAV - Airfoil optimization through swarm approach and low fidelity solver. In *Bristol International Unmanned Air Vehicle Systems*, Bristol, UK, 2009a.

M. S. Khurana, H. Winarto, and A. K. Sinha. Airfoil optimization by swarm algorithm with mutation and artificial neural networks. In *AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, pages 1–15, Orlando, FL, USA, 2009b. AIAA.

M. J. Kidger. *Intermediate Optical Design*. SPIE, 2004. ISBN 978-0819452177.

J. Knowles. *Local-search and hybrid evolutionary algorithms for Pareto Optimization*. PhD thesis, Department of Computer Science, University of Reading, Reading, UK, 2002.

J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2005.

J. Knowles and D. Corne. On metrics for comparing non-dominated sets. In *Congress on Evolutionary Computation (CEC)*, pages 711–716, Honolulu, HI, USA, 2002. IEEE Computer Society Press.

J. D. Knowles and D. W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

M. Köppen and R. Vicente-Garcia. A fuzzy scheme for the ranking of multivariate data and its application. In *North American Fuzzy Information Processing Society (NAFIPS)*, pages 140–145, Alberta, Canada, 2004. IEEE Computer Society Press.

M. Köppen and K. Yoshida. Many-Objective Particle Swarm Optimization by Gradual Leader Selection. In *International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)*, pages 323–331, Warsaw, Poland, 2007. Springer.

M. Köppen, R. Vicente-Garcia, and B. Nickolay. Fuzzy-Pareto-Dominance and its Application in Evolutionary Multi-objective Optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 399–412, Guanajuato, Mexico, 2005. Springer.

P. Korhonen and J. Laakso. A Visual Interactive Method for Solving the Multiple Criteria Problem. *European Journal of Operational Research*, 24:277–287, 1986.

P. Korhonen and J. Wallenius. Visualization in the Multiple Objective Decision-Making Framework. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 8, pages 195–212. Springer-Verlag, 2008.

J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992. ISBN 978-0262111706.

J. R. Koza, S. H. Al-Sakran, and L. W. Jones. Automated re-invention of six patented optical lens systems using genetic programming. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 1953–1960, Washington, DC, USA, 2005. ACM Press.

S. Kukkonen and J. Lampinen. GDE3: the third evolution step of generalized differential evolution. In *Congress on Evolutionary Computation (CEC)*, pages 443–450, Edinburgh, UK, 2005. IEEE Computer Society Press.

S. Kukkonen and J. Lampinen. Ranking-Dominance and Many-Objective Optimization. In *Congress on Evolutionary Computation (CEC)*, pages 3983–3990, Singapore, 2007. IEEE Computer Society Press.

F. Kursawe. A Variant of Evolution Strategies for Vector Optimization. In *Workshop on Parallel Problem Solving from Nature (PPSN)*, pages 193–197, Dortmund, Germany, 1991. Springer-Verlag.

X. Li. A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 37–48, Chicago, IL, USA, 2003. Springer.

X. Li. Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 117–128, Seattle, WA, USA, 2004. Springer.

X. Li, J. Zheng, and J. Xue. A Diversity Metric for Multi-objective Evolutionary Algorithms. In *International Conference of Advances in Natural Computation (ICNC)*, pages 68–73, Changsha, China, 2005. Springer.

X. Li, J. Branke, and M. Kirley. On performance metrics and particle swarm methods for dynamic multiobjective optimization problems. In *Congress on Evolutionary Computation (CEC)*, pages 576–583, Singapore, 2007. IEEE Computer Society Press.

A. Lotov and K. Miettinen. Visualizing the Pareto Frontier. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 9, pages 213–243. Springer-Verlag, 2008.

A. Messac and C. A. Mattson. Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. *American Institute of Aeronautics and Astronautic (AIAA) Journal*, pages 1–1, 2004.

Z. Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag Inc., 1994. ISBN 978-3540606765.

K. Miettinen. Some Methods for Nonlinear Multi-objective Optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 1–20, London, UK, 2001. Springer-Verlag.

K. Miettinen. Introduction to Multiobjective Optimization: Noninteractive Approaches. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 1, pages 1–26. Springer-Verlag, 2008.

K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to Multiobjective Optimization: Interactive Approaches. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 2, pages 27–57. Springer-Verlag, 2008.

J. Molina, L. V. Santana, A. G. Hernndez-Daz, C. A. C. Coello, and R. Caballero. g-Dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, 197(2):685–692, 2009.

S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization. *Swarm Intelligence Symposium (SIS)*, pages 26–33, 2003.

Y. Nagata. The Lens Design Using the CMA-ES Algorithm. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 1189–1200, Seattle, WA, USA, 2004. Springer.

H. Nakayama, K. Kaneshige, S. Takemoto, and Y. Watada. An application of a multi-objective programming technique to construction accuracy control of cable-stayed bridges. *European Journal of Operational Research*, 87(3):731–738, 1995.

T. Okabe, Y. Jin, and B. Sendhoff. A Critical Survey of Performance Indices for Multi-objective Optimisation. In *Congress on Evolutionary Computation (CEC)*, pages 878–885, Canberra, Australia, 2003. IEEE Press.

I. Ono, S. Kobayashi, and K. Yoshida. Global and multi-objective optimization for lens design by real-coded genetic algorithms. In *International Optical Design Conference (IODC)*, pages 110–121, Kona, HI, USA, 1998. SPIE.

Optical Research Associates. CODE V: Optical Design, Fabrication, and Analysis Software. http://www.opticalres.com/cv, 2010.

M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994. ISBN 978-0262650403.

J. Pfeiffer, U. Golle, and F. Rothlauf. Reference point based multi-objective evolutionary algorithms for group decisions. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 697–704, Atlanta, GA, USA, 2008. ACM Press.

K. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag, 2005. ISBN 978-3540209508.

D. Quagliarella and A. Vicini. Viscous single and multicomponent airfoil design with genetic algorithms. *Finite Elements in Analysis and Design*, 37(5):365–380, 2001.

N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Tucker. Surrogate-Based Analysis and Optimization. *Progress in Aerospace Sciences*, 41:1–25, 2005.

T. Ray and H. M. Tsai. Swarm algorithm for single and multiobjective airfoil design optimization. *American Institute of Aeronautics and Astronautics (AIAA)*, 42(2):366–373, 2004.

M. Reyes-Sierra and C. A. Coello Coello. Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and epsilon-Dominance. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 505–519, Guanajuato, Mexico, 2005. Springer.

M. Reyes-Sierra and C. A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2:287–308, 2006.

T. Robic and B. Filipic. DEMO: Differential Evolution for Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 520–533, Guanajuato, Mexico, 2005. Springer.

J. A. Samareh. A Survey of Shape Parameterization Techniques. Technical Report 1999-209136, NASA Langley Research Center, 1999.

Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of multiobjective optimization*. Academic Press, 1985. ISBN 0126203709.

J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *International Conference on Genetic Algorithms*, pages 93–100, Pittsburgh, PA, USA, 1985. L. Erlbaum Associates Inc.

H. Schlichting and K. Gersten. *Boundary-Layer Theory*. Springer, 2002. ISBN 978-3540662709.

H.-P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., 1993. ISBN 978-0471571483.

Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Congress on Evolutionary Computation (CEC)*, pages 69–73, Anchorage, AK, USA, 1998. IEEE Computer Society Press.

H. Sobieczky. Parametric Airfoils and Wings. In *Notes on Numerical Fluid Mechanic*, pages 71–88. Vieweg, 1998.

D. S. Somers. Design and Experimental Results for a Natural-Laminar-Flow Airfoil for General Aviation Applications. Technical Report 1861, Langley Research Center, 1981.

W. Song and A. J. Keane. A Study of Shape Parameterisation Methods for Airfoil Optimisation. In *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1–8, Albany, NY, USA, 2004. AIAA.

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

R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application.* Krieger Publishing Company, 1989. ISBN 978-0894643934.

T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Gbelt, Y. Jin, H. Nakayama, S. Poles, and D. D. Stefano. Real-World Applications of Multiobjective Optimization. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, chapter 11, pages 285–327. Springer-Verlag, 2008.

T. J. Stewart, R. Janssen, and M. van Herwijnen. A genetic algorithm approach to multiobjective land use planning. *Computers and Operations Research*, 31(14):2293–2313, 2004.

A. Sülflow, N. Drechsler, and R. Drechsler. Robust Multi-Objective Optimization in High Dimensional Spaces. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 715–726, Matsushima, Japan, 2007. Springer.

H. Takagi. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

S. Thibault, C. Gagné, J. Beaulieu, and M. Parizeau. Evolutionary Algorithms Applied to Lens Design: Case Study and Analysis. In *SPIE International Symposium on Optical Systems Design (EOD)*, volume 5962, Jena, Germany, 2005. SPIE.

L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3):411–436, 2009.

I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325, 2003.

UIUC. University of Illinois at Urbana-Champaign: Airfoils coordinates database. http://www.ae.illinois.edu/m-selig/, 2010.

D. A. Van Veldhuizen. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations.* PhD thesis, Air Force Institute of Technology, Wright Patterson AFB, Ohio, USA, 1999.

D. Vasiljevic. *Classical and Evolutionary Algorithms in the Optimization of Optical Systems.* Springer, 2002. ISBN 978-1402071409.

G. Venter and J. Sobieszczanski-Sobieski. Structural and multidisciplinary optimization. *Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization*, 26(1–2):121–131, 2004.

T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 742–756, Matsushima, Japan, 2007. Springer-Verlag.

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

U. K. Wickramasinghe and X. Li. Integrating user preferences with particle swarms for multi-objective optimization. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 745–752, Atlanta, GA, USA, 2008. ACM Press.

U. K. Wickramasinghe, R. Carrese, and X. Li. Designing Airfoils using a Reference Point based Evolutionary Many-objective Particle Swarm Optimization Algorithm. In *Congress on Evolutionary Computation (CEC)*, pages 1857–1864, Barcelona, Spain, 2010. IEEE Computer Society Press.

A. Wierzbicki. The use of reference objectives in multiobjective optimisation. In *Lecture notes in Economics and Mathematical systems: MCDM Theory and Application*, pages 468–486. Springer Verlag, 1980.

D. J. Wilde. *Optimum Seeking Methods.* Prentice Hall, 1964. ISBN 978-0136382393.

X. Xu, Y. Li, S. Fang, Y. Wu, and F. Wang. A Novel Differential Evolution Scheme Combined with Particle Swarm Intelligence. In *Congress on Evolutionary Computation (CEC)*, pages 1057–1062, Hong Kong, China, 2008. IEEE Computer Society Press.

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

W.-J. Zhang and X.-F. Xie. DEPSO: Hybrid Particle Swarm with Differential Evolution Operator. In *International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 3816–3821, Xi'an, China, 2003. IEEE Computer Society Press.

E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Parallel Problem Solving from Nature (PPSN)*, pages 292–304, Amsterdam, The Netherlands, 1998. Springer-Verlag.

E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3: 257–271, 1999.

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

E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.