Improving Convergence, Diversity and Pertinency in Multiobjective Optimisation

Thesis submitted for candidature

for the degree of Doctor of Philosophy

Salem Fawaz Adra

October 2007

Department of Automatic Control and Systems Engineering

The University of Sheffield

Summary

Real-world problems commonly require the simultaneous consideration of multiple, often conflicting, objectives. Solving a multiobjective optimisation problem (MOP) is concerned with finding an ideal set of tradeoff solutions which are close to and uniformly distributed across the optimal tradeoff surface. Convergence and diversity are thus essential requirements of multiobjective optimisers, which are sometimes also required to focus on pertinent areas of the search space. Evolutionary computation (EC) techniques are stochastic, population-based, global search techniques well suited for solving MOPs. However, EC techniques can often involve a large number of objective function calculations which can make the convergence towards optimal tradeoff surfaces computationally expensive. Additionally, in the evolutionary multiobjective optimization community, the bi-objective case is the most heavily studied. Conclusions drawn from such low-dimensional frameworks used to be generalized for all problems' dimensions. Research, however, has shown that high-dimensional problems (> 3 objectives) can possess different characteristics. One of the most important challenges faced in such optimisation scenarios is the conflict between convergence and diversity of solutions.

In this study, new approaches are proposed for enhancing the convergence and diversification capacities of some of the best multiobjective evolutionary optimisers (MOEAs). The inclusion of quality metrics as indicators is implemented as an approach for solving the conflict between solutions' convergence and diversity in high-dimensional optimisation problems. Moreover, a convergence acceleration technique for MOEAs which exploits the objective space, where the goal and objectives lies, is devised and assessed. In the final part of the study, some established progressive preference articulation techniques are examined, and their utility for tackling MOPs is discussed from the viewpoint of the decision maker. Progressive preference articulation techniques are effective methods for supporting the decision maker in guiding the search into pertinent regions of interest and coping with the curse of dimensionality.

Acknowledgments

I would like to thank my great supervisor Professor Peter J. Fleming for the opportunity to do postgraduate research, his expert advice and the very much appreciated patience and excellent guidance throughout the past few years. I would also like to thank Dr Ian Griffin for his valued friendship, expert supervision and advice, and without whom none of this work would have been possible. Many thanks also go to Dr Tony Dodd for his expertise and instructive advice on neural networks and his open and friendly manner. Also I would like to extend my appreciation to my PhD examiners, Dr Joshua Knowles and Dr Robert Harrison, for all their expert advice and for a very stimulating and interesting examination.

I would also like to extend my thanks to all my friends and colleagues at Sheffield for their most valuable friendship, patience, and encouragement. Without you guys, it would have been a much more difficult journey. In particular, and I hope I haven't forgotten anyone, many and special thanks go to Abdulla Al Hasan, Mohamad Tahlak, Ahmad Hamadeh, Dr Sherif Arafa, Ahmad Hamody, Mohamad El Tayeb, Waleed El Harabi, Shahid Mahmoud, Ahmad Nassef, Mohamad AlKarouri, Alex Shenfield, Andy Mills, Alejandro Torres, Luka Eciolaza, Robin Purshouse, Robert Bracey, Clare F Clarke, Dr Des Ryan and Dr Samer Kharoubi.

Finally, I would like to thank my beloved father, mother and two sisters for all their sacrifice, encouragement and support. The thesis page limit does not allow me to thank you enough! Without you, I wouldn't be the person I am today. And last but not least, I would like to thank my beloved wife and sweetest addiction in life, Fadwa. Thanks for your patience, encouragement, endurance, love and being there in my life.

Statement of Originality

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

Candidate: ______________________________
Salem F. Adra

Supervisor: ______________________________
Peter J. Fleming

To my beloved Fadwa

Contents

Chapter 1

Introduction

1.1. Motivation

Solving real-world problems commonly require the simultaneous consideration of multiple, and often, conflicting, performance measures. Examples of such real world problems include automotive, aerospace and biological applications and disciplines. Such applications generally provide illustrations of some typical design challenges and the considerable number of objectives and constraints usually involved in the problem solving process. Unfortunately, when tackling optimisation problems with many conflicting objectives, no single "Utopian" solution, which gives the best achievable performance across all the competing objectives, can be found. This is due to the fact that such "Utopian" solution would most likely be favouring a certain objective or subset of objectives whilst presenting poor performances in terms of the remaining competing objectives. Solving such problems is therefore concerned with finding an ideal set of tradeoff solutions that satisfies the decision maker's (DM) preferences, and meets the goal values set for the problem objectives without violating any imposed constraints. Moreover, *proximity*, *diversity* and *pertinence* are three essential requirements sought when optimising a multiobjective problem. *Proximity* (or convergence) denotes the requirement for producing a set of tradeoff solutions with maximum closeness[1] to the optimal tradeoff surface of a multiobjective optimisation problem (MOP). *Diversity* is the second requirement and denotes the requisite for a uniformly distributed set of solutions across the optimal tradeoff surface of a MOP. The third requirement for solution's *pertinency* (Purshouse 2004), consists of producing solutions that reside in the DM's region(s) of interest (ROI). At the end of the optimisation process, the DM can then decide on a single solution to be implemented based on preferences and application dependent high-level information.

Evolutionary Algorithms (EAs) are stochastic, global search techniques well tuned for solving MOPs due to their ability to efficiently explore vast hyperspaces for valuable solutions and promising regions of interest. EAs symbolize a metaphor of evolutionary

[1] Closeness is usually the Euclidean distance in most real-coded optimisation problems

biology and are built upon the concept of natural selection and the survival of the fittest. The inspiration of these optimisation and search techniques from a real world concept solidify their grasp and understanding and makes them attractive to, among others, engineers, evolutionary biologists and computer scientists. Nevertheless, traditional EAs usually consist of an explorative, stochastic and iterative set of procedures in the search space of a multiobjective problem. As a result, despite their well-established utility for solving MOPs, EAs usually require an extensive number of objective function evaluations in order for the search to converge to the Pareto optimal front in the objective space.

In the Evolutionary Multiobjective Optimization (EMO) community the bi-objective case has been the most heavily studied. Conclusions drawn from such low dimensional multiobjective frameworks have been used to generalize the multiobjective branch of evolutionary optimisation problems. However, research (Khare, Yao and Deb 2003, Purshouse 2004) has showed that the case of high-dimensional optimisation problems (more than 3 objectives) termed as *many*-objective optimisation[2] is a special case of evolutionary multiobjective problems that needs further investigation and an essential discrimination from the "2-3" objective cases. A whole set of difficulties and challenges arise in the field of evolutionary *many* objectives optimisation. Most importantly, the unambiguous conflict between solutions' convergence and diversity -the two major requirements of EMO- can be noted in such high dimensional scenarios.

In this thesis, a closer look at each of the EMO requirements -solutions' *convergence* towards and *diversity* across the Pareto optimal front as well as their *pertinence* to the DM- is realised in a many objective optimisation framework. Research is produced for investigating and devising innovative approaches for efficiently meeting the convergence and the diversity requirements, and enhancing the state of the art in EMO. A remedial measure for solving the revealed conflict between solutions' convergence and diversity in the *many*-objective optimisation case (Khare, Yao and Deb 2003, Purshouse and Fleming 2003b) is introduced and carefully assessed. Additionally, major progressive preference articulation (PPA) techniques are investigated and analyzed as an attempt to establish the advantages and disadvantages of each of these techniques and promote research into this highly beneficial but somehow overlooked area of EMO. PPA techniques are well-established remedial measures for coping with the curse of dimensionality and satisfying the *pertinence* requirement in EMO.

[2] *Many* objective is an expression introduced in the OR community (Farina and Amato 2002) to denote optimisation problems with more than two or three objectives.

1.2. Outline of the Thesis

In Chapter 2 of the thesis, a review of multiobjective optimisation is presented. The essential requirements for a multiobjective optimisation problem are illustrated. Moreover, the concept of Pareto dominance and its utility for multiobjective optimisation is described. The discipline of evolutionary computation (EC) is described and introduced with a particular emphasis on evolutionary algorithms, one of its major fields. Fundamental and advanced concepts of evolutionary algorithms are illustrated. The utility of evolutionary algorithms for solving multiobjective optimisation problems is then described and contrasted with the utility of classical approaches for solving such problems. The major developments that the EMO community has witnessed over the last 20 years are presented emphasizing the different features suggested and implemented over the years for addressing MOPs' requirements.

In Chapter 3, an innovative approach for addressing the requirement for solutions convergence to the Pareto front is presented. The approach makes use of a beneficial local search in the objective space and the predictive capacities of neural networks for accelerating the convergence of an evolutionary search towards the Pareto front. The approach is implemented and tested on a number of optimisation problems with a various number of objectives in the range [2, 12]. The suggested convergence acceleration approach is meant to be a portable operator which can be hybridized with any population based multiobjective optimiser. In this work, the introduced approach was hybridized with two well-established and one of the most cited multiobjective evolutionary algorithms (MOEAs). For all the optimisation problems and objective dimensionality investigated, the integration of the convergence acceleration technique improved the performance of the MOEAs by producing better results which are closer to the Pareto front.

In Chapter 4, the diversity requirement in EMO is further investigated in a *many* objective optimisation framework. An adaptive strategy for controlling and promoting the diversity requirement in EMO is presented and tested. The strategy uses a diversity indicator to detect undesirable diversity levels such as the over dispersal of solutions in the objective hyperspace, or otherwise the contraction of the handled solutions to certain regions of the search space. The diversity requirement is hence measured, evaluated and consequently controlled in a way which preserves the required level of diversity without sacrificing the convergence process towards Pareto optimal tradeoff surfaces. In Chapter 4, optimisation problems with a number of objectives ranging between 5 and 20 are deployed. Very good results are reported highlighting the utility of the diversity management strategy introduced.

In Chapter 5, the pertinence requirement in EMO is addressed. A comparative study of some of the most reputed and most popular preference articulation techniques are contrasted on multiple optimisation scenarios. In particular, progressive preference articulation is investigated. Well-established PPA techniques and new suggested techniques for progressively articulating preferences in EMO are contrasted and their major advantages and disadvantages are discussed. The aim of this Chapter is to promote research into progressive preference articulation, an overlooked but very beneficial area of research in EMO.

Finally, in Chapter 6, conclusions, future work directions and suggestions are presented.

1.3. Contributions

During the course of this study, a number of contributions in terms of the three multiobjective optimisation requirements for convergence, diversity and pertinency were produced. In the following, the main contributions of the thesis are briefly described.

Contributions in terms of Convergence in Evolutionary Multiobjective Optimisation:

A study by Adra et al (2005b) introduced a hybrid multiobjective evolutionary algorithm (MOEA) and was used to optimise a classical 8 objective problem of aircraft control system design. This work has pioneered one of the first approaches for manipulating objective values directly in the objective space as opposed to the EC traditional continuous search in the decision variable space, and decision space to objective space mapping.
The framework of this research consisted of the Multiobjective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993). The strategy suggested was composed of two main stages. The first stage consisted of training an artificial neural network (NN) with objective values as inputs and decision variables as outputs to model an approximation of the mapping function from objective space to decision variable space. The second stage consisted of a local improvement phase in the objective space preserving objectives relationships, and a mapping process to decision variables using the trained NN. Both the hybrid MOEA and the original MOEA were applied to an 8 objective optimisation problem of aircraft control system design application for assessment. In a decision-making framework integrating preference articulation, the objective space local improvement process played the role of a deterministic translator of the DM preferences by steering the search in the desired directions towards goal values and regions of interests. The results achieved by the introduced hybrid MOGA were significantly better than the results achieved by the standalone MOGA. The binary ε-Indicator (Zitzler *et al* 2003), a popular

binary performance indicator was deployed to test the performance of the introduced optimiser.

Extending the work presented in Adra *et al* (2005b), a novel convergence accelerator using neural network predictions and objective space direct manipulation strategies is introduced in Adra, Griffin and Fleming (2007b). The convergence accelerator is meant to be a portable component that can be plugged into any population based, stochastic optimisation algorithm, such as genetic algorithms. The purpose of the operator is to enhance the search capability, convergence extent and the speed of convergence of the hosting stochastic global optimisation technique. The work presented in Adra, Griffin and Fleming (2007b) is further extended and improved in Chapter 3. The extended work presented in Chapter 3 is currently in the process of submission to the IEEE transactions on Evolutionary Computation.

Moreover, in Adra, Griffin and Fleming (2007c), an operator termed as the Informed Convergence Accelerator (ICA) is introduced. The technique suggested in Adra, Griffin and Fleming (2007c) was inspired by, and derived from the work presented in Chapter 3. The major difference was that the training process of the NN took place offline during an entire execution of a multiobjective optimiser in order to enhance the predictive capabilities of the NN and eliminate the correction step described in Chapter 3. The ICA is a specialised operator which was implemented and intended for use in multiobjective optimisation scenarios under change and incertitude or for tackling robust optimisation problems where the results precision may vary within a certain range of tolerance.

Contribution in terms of Diversity in Evolutionary Multiobjective Optimisation:

In Adra, Griffin and Fleming (2005a), the authors presented a work which is concerned with memetic algorithms, a specific brand of evolutionary algorithms. A new local search technique with an adaptive neighbourhood setting process was introduced and hybridized with the Multiobjective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993). The resulting memetic algorithm was executed to optimise a set of bi-objective test functions presenting different challenges such as multimodality, deception, discontinuity and convexity. Two performance criteria were assessed: the convergence of the achieved results towards the true Pareto fronts of the test functions used and the results diversity. The strategy deployed a performance indicator to steer the search into the right direction by promoting solutions diversity, one of the desired characteristics of an ideal MOEA. Simulation results showed that the algorithm introduced improvements for some of the test functions and had promising search capabilities.

Improving on the results achieved in the work published in the proceedings of GECCO 2005, in Adra, Griffin and Fleming (2006), an enhanced local search technique with an adaptive neighbourhood setting process was introduced, integrated within the cycle of the global search process of MOGA and tested against a diverse set of test functions. The neighbourhood setting process was based on a measure of the local density of the locally best solutions and the extent of diversity of the population.

Contribution in terms of Pertinency in Evolutionary Multiobjective Optimisation:

Preference articulation techniques, especially progressive and interactive ones, are effective methods for supporting the decision maker and guiding the search towards pertinent regions of the search space. In Adra, Griffin and Fleming (2007a), some of the most recent and most established preference articulation techniques were examined, and their utility for tackling multiobjective optimisation problems is discussed and assessed from the viewpoint of the decision maker. The aim of the study presented in Adra, Griffin and Fleming (2007a) consisted of encouraging and promoting the research of incorporating progressive preference articulation techniques into evolutionary multiobjective optimisation. In this work, some of the most recent preference articulation techniques are discussed and upgraded to their progressive versions for incorporation into evolutionary multiobjective optimisation processes. Their major strengths and weaknesses for tackling multiobjective optimisation problems are discussed and illustrated on a straightforward bi-objective scenario for simplicity. Although the deployed scenarios consisted of 2 and 4 dimensional scenarios, the strengths, weaknesses, and therefore the efficiency and suitability of these PPA techniques for the many-objective optimisation were apparent.

Chapter 2

Review of Multiobjective Optimisation Methods

2.1. Multiobjective Optimisation

2.1.1. Introduction

Problem solving is a cognitive process and one of the most complicated intellectual activities of the human brain. Regardless of the problem's nature, the process of problem solving is concerned with finding solutions to certain predicaments or transitions from certain reviled states to desired states, such as the transition from unstable to stable or incorrect to correct. When a problem exists, a solution is sought and an objective is established. Unfortunately, finding a solution to certain problems or satisfying certain objectives can be a tricky and complicated process. An exact solution to some problems might simply be infeasible, especially if the problem consists of multiple conflicting and constrained objectives. In many applications, a good approximation or alternatively an *optimised* estimate to a solution might be deemed very good and sufficient.

Real-world problems commonly require the simultaneous consideration of multiple performance measures. Buying a new car is a simple illustration of such real-world multiobjective tasks. Comfort, price, depreciation factor, safety features, road tax and running costs such as fuel, servicing and repairs are all criteria that usual car buyers consider and look to optimise when buying a new car. Most often, the multiple objectives are in conflict and compete with each other. Ultimately, the decision maker (DM) has to decide on an individual solution based on certain preferences and objectives' priorities. As an example, a DM might decide that the safety features in a car are prioritised over the running costs of its fuel consumption while on the other side the car's depreciation factor can be traded for its luxurious comfort.

Whether trying to figure out the best road combination to take for reaching work before 9:00 am and avoid getting stuck in crowded neighbourhoods or planning a nice vacation that suits both your expectations and finances, multiobjective optimisation is a common task that we frequently encounter in our daily lives.

Different from single objective optimisation which aims to maximize, minimize or achieve a certain goal value for a single objective, multiobjective optimisation consists of multiple criteria that need to be optimised simultaneously. These criteria can manifest pair-wise relationships such as independence, harmony or conflict. In the former two relationships, an improvement or, alternatively a deterioration in terms of a certain objective, will either have no influence on the performance of the remaining independent objectives or alternatively, an impact of a similar nature. Such multiobjective optimisation (MO) scenarios can be ultimately divided into a set of different single objective optimisation problems in the case of complete independence, or reduced to a single optimisation problem of one representative objective in the case of complete harmony. Optimising multiple competing objectives is by far the most complicated multiobjective scenario, as in such scenarios no single Utopian solution can be found.

Without any loss of generality, a multiobjective optimisation problem (MOP) can be formally presented as a minimization problem of a certain vector function **Z**.

Definition 2. 1: Multiobjective optimisation problem

$$\min \mathbf{Z}(\mathbf{x}) = \{z_1(\mathbf{x}), z_2(\mathbf{x}) \ldots\ldots z_n(\mathbf{x})\} \qquad \mathbf{x} = (x_1, x_2 \ldots\ldots x_m)$$

In Definition 2.1, **Z(x)** is a vector of objective functions, 'n' is the number of objectives to be optimised and '**x**' is a vector of decision variables defined over a universe U.

Single objective optimisation are usually solved using numerical analysis methods such as the classical gradient descent, Newton-Fourier and Levenberg-Marquardt methods which operate in a single search space, termed as the decision variable space. For a thorough literature about numerical analysis and optimisation techniques in the operations research (OR) community, the interested reader is directed to Luenberger (1984) and Hillier and Lieberman (2001). OR is an interdisciplinary science which deals with decision making, optimisation, planning and coordinating activities of complex nature from the real-world.

In single objective optimisation, a solution explored in the decision variable space replaces the current best solution only if it presents a superior objective function value. Operating in a single search space is yet another major difference with the simultaneous optimisation of multiple objectives. Multiobjective optimisation consists of finding the set of vectors in the decision variable space which produce the best set of solutions in the objective space. Usually, the search in the decision variable space is steered and influenced by the information that becomes available in the objective space.

Edgeworth (1932) states that in order to compare alternative solutions for a multiobjective optimisation problem or favour a single solution or set of solutions out of a collection of candidate solutions, a consistent line of preference for each objective is required. These lines of preference constitute the basis for comparison and evaluation of solutions. Having a consistent basis for comparing candidate solutions and favouring a certain solution over another makes it appropriate to employ *Pareto* concepts for measuring optimality (Coello, Veldhuizen and Lamont 2002).

In Figure 2.1 an optimisation problem consisting of two objectives and three decision variables is illustrated. Note that the dimensionality of the objective space and the decision variable space can be any positive integer. The grey bounded area in the decision variable space (Figure 2.1) denotes the feasible region of the space which is defined by certain application specific constraints. **'Z'** is the objective vector function which maps a certain solution **'x'** in the decision variable space to its corresponding objective vector. It is only through the objective function mapping that the performance of a certain candidate solution can be assessed.

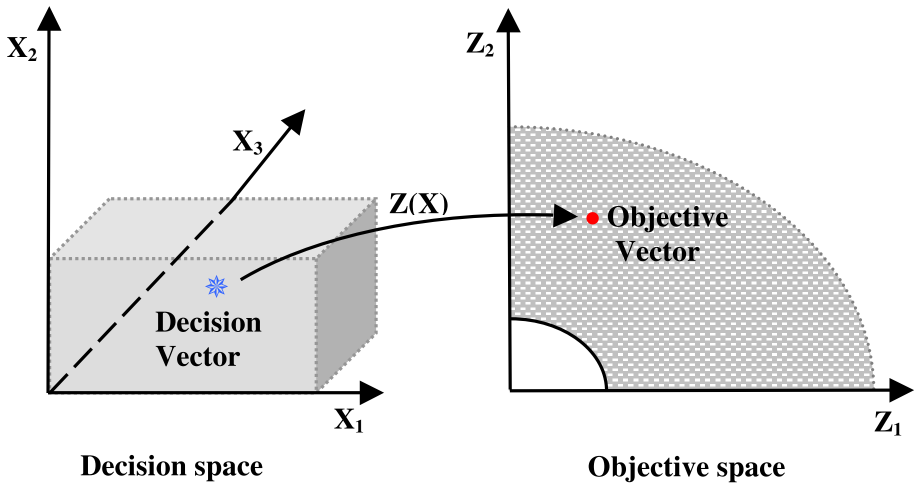


Figure 2. 1 The multiobjective problem domain

Pareto Dominance

"We will say that the members of a collectivity enjoy maximium ophelimity in a certain position when it is impossible to find a way of moving from that position very slightly in such a manner that the ophelimity enjoyed by each of the individuals of that collectivity increases or decreases. That is to say, any small displacement in departing from that position necessarily has the effect of increasing the ophelimity which certain individuals enjoy, and decreasing that which others enjoy, of being agreeable to some, and disagreeable to others."

V. Pareto (1906, p: 261)

[3] Economic Satisfaction

The idea of 'optimal' in MO can be traced back to the period between 1870 and 1900 with the work and philosophies of Ysidro Francis Edgeworth (1845-1926) and Vilfredo Pareto (1948-1923), some of the most brilliant economists of the 19th century. Edgeworth main interests revolved around the utilitarian philosophy whose ultimate aim consisted of maximising society's happiness by optimising the problem of resource allocation. Vilfredo Pareto, on the other hand, concentrated on the use of classical programming techniques such as differential calculus and Lagrangian multipliers for the analysis of general equilibrium theories and the optimisation of market efficiency. His work and theories constituted the foundation of the Pareto optimality concept which comprise the core of most multiobjective optimisers.

A certain solution **'A'** in the decision space of a multiobjective problem is superior to another solution **'B'** *if and only if* **Z(A)** is at least as good as **Z(B)** in terms of all the objectives and strictly better than **Z(B)** in terms of at least a single objective. Solution **'A'** is also said to strictly dominate solution **'B'**. The Pareto dominance concept is illustrated in Figure 2.2 in the objective space of a simple bi-objective scenario.

Definition 2. 2 Pareto Dominance

$$\mathbf{Z(A)} \prec \mathbf{Z(B)} \quad \mathit{iff} \quad \forall i \in \{1,...,n\}, Z_i(\mathbf{A}) \leq Z_i(\mathbf{B}) \wedge \exists i \in \{1,...,n\} : Z_i(\mathbf{A}) < Z_i(\mathbf{B})$$

In the case where no feasible solution '**B**' can be found dominating solution '**A**', solution '**A**' is said to be a *non dominated* solution or a *Pareto optimal* solution. In another words, a solution '**A**' for a multiobjective optimisation problem is a Pareto optimal solution if and only if:

Definition 2. 3 Pareto Optimal Solution

$$\neg \exists \mathbf{B} \in U : \mathbf{Z(B)} \prec \mathbf{Z}(A)$$

A subset of the decision vectors produced will be therefore characterised by the fact that no other solution within the set of candidate solutions, also known as the approximation set (Zitzler et al 2003), offers better objective function values across all objectives. The corresponding objective vectors for such subset constitute the tradeoff surface also called the *Pareto front.*

Other forms of solutions' dominance exist and their use as an alternative to the standard Pareto dominance scheme is becoming more popular especially in the Evolutionary Multiobjective Optimisation (EMO) community. One of the most reputed alternatives of Pareto dominance is the relaxed ε-Pareto dominance (Helbig and Pateva 1994) which was established in the mid-1980s in the operations research community and refined in 2002 by Laumanns et al (2002a).

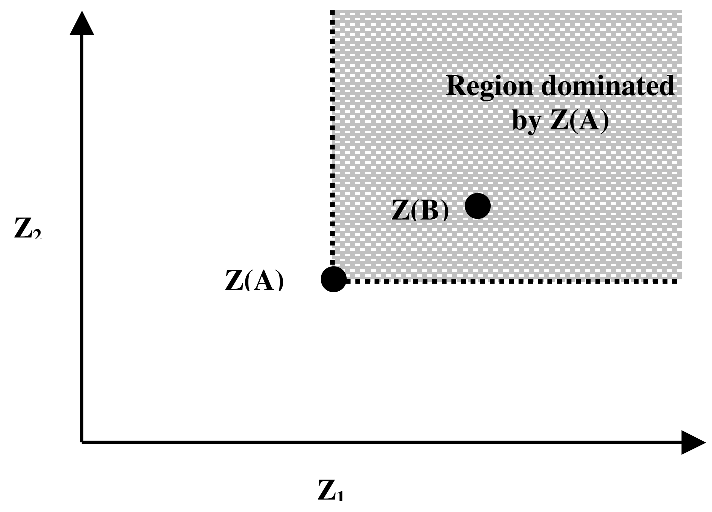


Figure 2. 2 Pareto Dominance Illustration in a 2-dimensional objective space

The use of ε-Pareto dominance is beneficial to tune the granularity of the optimisation process and the convergence speed. Additive and multiplicative ε-Pareto dominance are two well-known forms of this modified Pareto dominance scheme. α-dominance (Ikeda *et al* 2001) and k-dominance (Farina and Amato 2004) are among other alternative forms of Pareto dominance.

Definition 2. 4 Multiplicative ε -Pareto Dominance

A certain solution '**A**' for a multiobjective problem is said to ε-dominate another solution '**B**' for a certain ε > 0 if and only if:

$$\mathbf{Z(A)} \prec_{\varepsilon} \mathbf{Z(B)} \quad \textit{iff} \quad \forall i \in \{1,...,n\}, (1-\varepsilon_i)\, Z_i(\mathbf{A}) \leq Z_i(\mathbf{B})$$

In the Definition 2.4, the multiplicative form of ε-Pareto dominance, where the ε tolerance or deviation in terms of a certain objective is relative to the values of each criterion of Z(A), is illustrated. In Figure 2.3, an illustration of the multiplicative ε - Pareto dominance is presented using a two objective scenario. The shaded area corresponds to the region of the objective space which is *ε–dominated* by solution '**A**'. Solution **'B'** which is normally considered equally good to solution **'A'** and non-dominated by **'A'** using the standard Pareto dominance, is now considered ε–dominated by **'A'**. Each solution therefore dominates larger regions of the objective space.

The additive ε-Pareto dominance, presented in Definition 2.5, is another form of ε - dominance where the tolerance value (ε_i) in terms of a certain objective is absolute.

Definition 2. 5 Additive ε -Pareto Dominance

$$\mathbf{Z(A)} \prec_{\varepsilon} \mathbf{Z(B)} \; \textit{iff} \; \forall i \in \{1,...,n\}, Z_i(\mathbf{A}) - \varepsilon_i \leq Z_i(\mathbf{B})$$

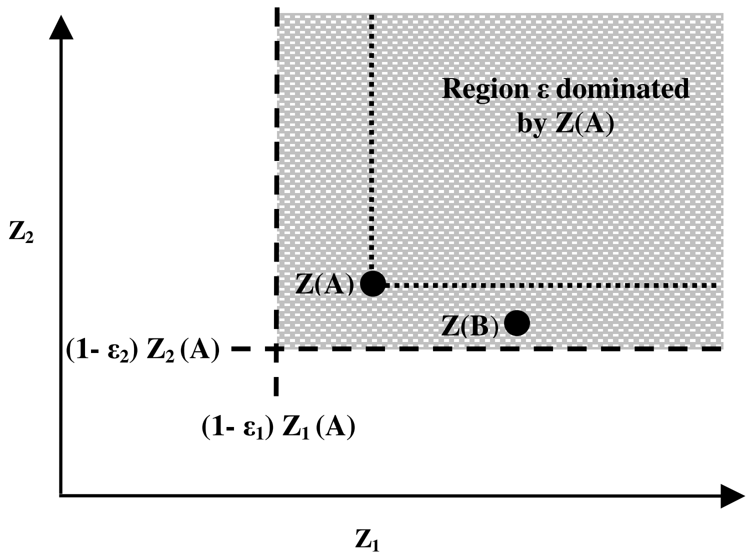


Figure 2. 3 Pareto Dominance Illustration in a 2-dimensional objective space

2.1.2. Requirements of a multiobjective optimiser

Three main requirements for multiobjective optimisers are usually sought and desired.

- **Convergence**: The approximation set achieved for a multiobjective optimisation problem is required to as close as possible to the true Pareto front.
- **Diversity**: Because of the non-existence of an ideal single solution in multiobjective optimisation frameworks with many competing objectives, and due to the fact that the global trade-off surface can potentially present an infinite number of solutions, the set of Pareto optimal solutions is also required to be well spread and uniformly covering wide areas of the Pareto front. Solutions diversity is conventionally preferred in the objective space as to present the DM with a well-distributed set of solutions to choose from, based on certain preferences such as objective priorities or region of interest (ROI). Solutions' diversity is however not restricted to the objective space, and can be a desired requirement in the decision space of some applications.
- **Pertinence** (Purshouse 2004): As the dimensionality of the problem increases, the visualization of the optimisation process becomes a problem. The DM is usually interested in sub-regions of the search space which makes the decision-making process and the optimisation process more practical and efficient. Therefore, the convergence and the diversity of the solutions are particularly required in the pertinent areas of the space, or regions of interest (ROIs).

Hence, convergence, diversity and pertinence are all desired and essential requirements of multiobjective optimisers (Purshouse 2004) and constitute their assessment basis. In

Figure 2.4, the ideal solution for a multiobjective optimisation problem is illustrated for a discontinuous problem.

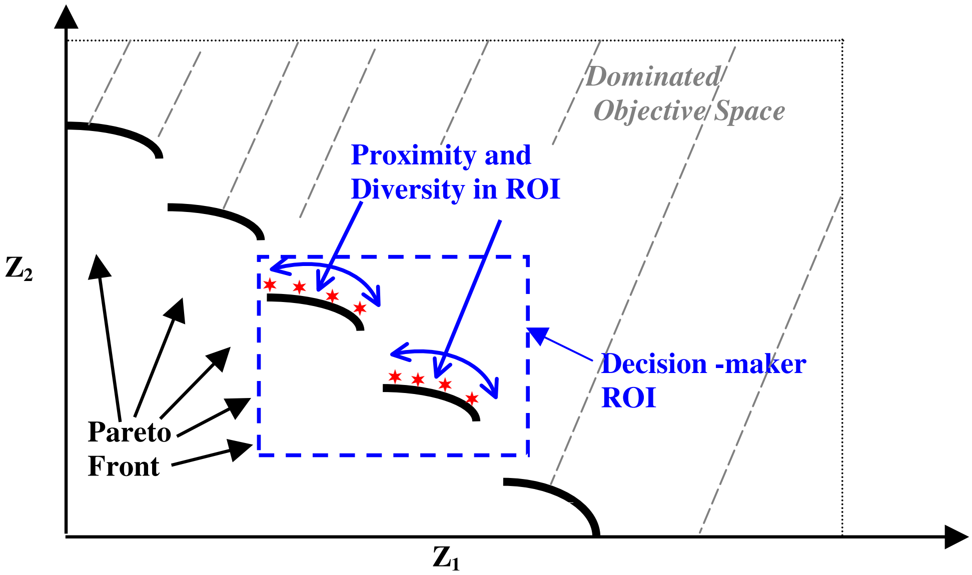


Figure 2. 4 The ideal solution to a multiobjective optimisation problem

2.2. Evolutionary Algorithms

2.2.1. Introduction

According to Darwinism theories, the human's physical and mental status with which we are privileged is the chronological result of successive biological evolutions. For example, prehistorically, within primate populations, the best individuals were able to survive biological and environmental crises such as climatic breakdowns, adapting to new environments and slowly undergoing minor advantageous mutations. This is "*survival of the fittest*".

Evolutionary Computation (EC) is a search discipline which is inspired by the "survival of the fittest" concept in evolutionary biology. More specifically, EC is a sub field of the ever-growing field of *soft computing* which is widely assimilated with the computational artificial intelligence (CI) discipline (Craenen and Eiben 2006). Traditional computing, also known as *hard computing*, is widely used in the OR community. Traditional computing techniques consist of finding hard and exact solutions to well-defined problems based on well-structured proofs and theorems. Soft computing is a discipline inspired by natural processes and has its main resemblance with the human way of reasoning. Soft computing differs from hard computing in the way it is dedicated to finding efficient solutions for complex problems characterised with uncertainties and tolerance. Soft computing techniques are usually considered a practical alternative to exact and specialized computing methods which are known to face difficulties and fail to tackle such problems. EC has it roots in the natural selection concept by Darwin (1859) and the population genetics by Fisher (1930). In Figure 2.5 an illustration of the roots and the different branches of EC is presented. The highlighted nodes in Figure 2.5 represent the relevant topics that are covered in this study.

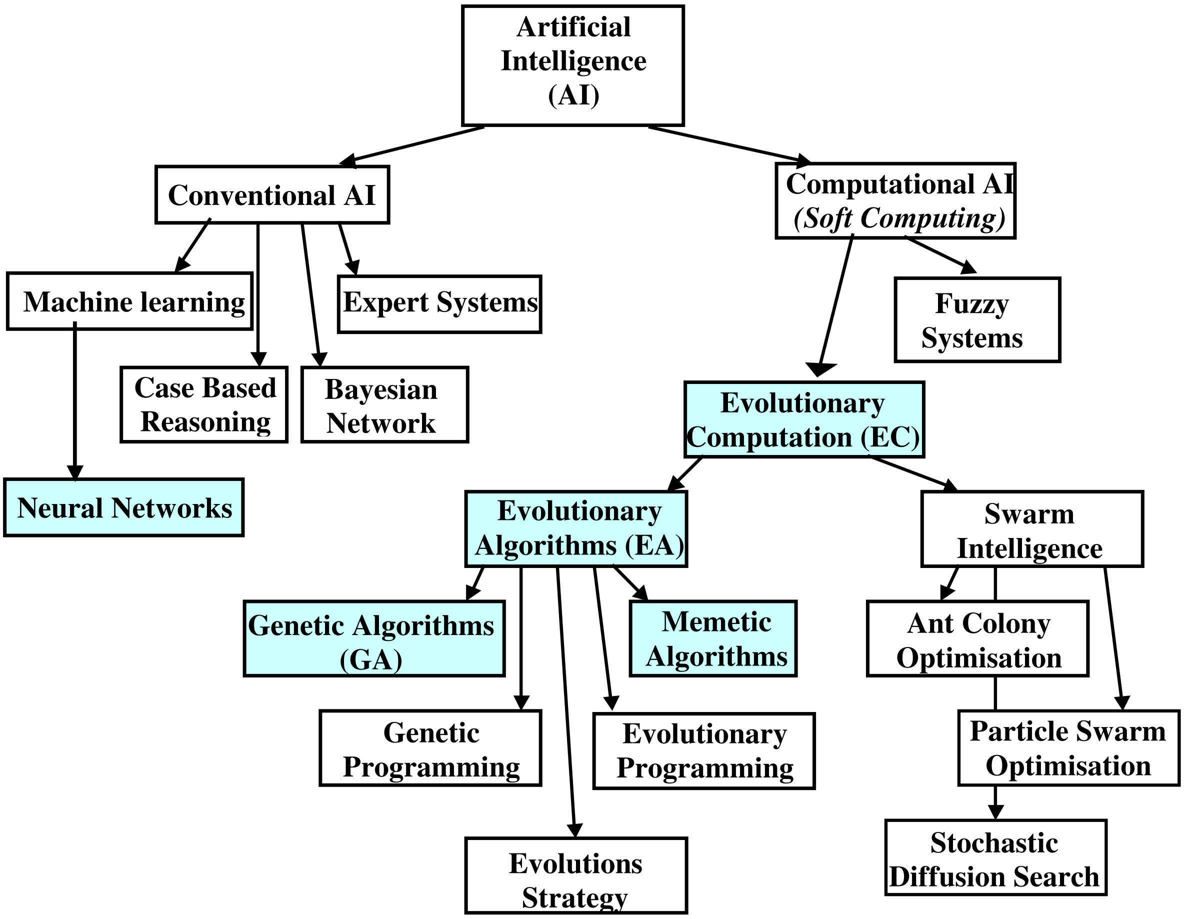


Figure 2. 5 Roots and Branches of Evolutionary Computation

Evolutionary Computation goes back to the late 1950s and had its early derivations in Friedberg's works (Friedberg 1958, Friedberg, Dunham and North 1959) with the growth of interest into machine learning and AI. With the superb advancements in information technology and computational frameworks, more contemporary subfields of EC were innovated. The German *Evolutionstrategian* (Evolution Strategies (ESs)) (Rechenberg 1973), the American Evolutionary Programming (EP) (Fogel, Owens and Walsh 1966) and the American Genetic Algorithm (GA) (Holland 1975) are the three founding approaches and independent schools of thought that defined the field of EC. The differences between ES, EP and GA approaches has diminished with the different communities regularly interchanging ideas over the years, and EC researchers and practitioners are nowadays encouraged to think in broader terms (Michalewicz and Fogel 2000). As a result the term

'*Evolutionary Algorithm'* (EA) was adopted to denote a generic evolutionary search approach.

2.2.2. Generic Evolutionary algorithm

An evolutionary algorithm is a stochastic global search method that imitates the process of natural biological evolution (Goldberg 1989) by operating on "*populations*" of potential solutions and applying the law of the jungle where the survival is for the fittest, hopefully producing better approximations to a given application's solution. Until a stopping criterion is reached (e.g. involving a certain number of generations or a mean deviation in the population), a new set of approximations is created at each generation by selecting "*individuals*" or solutions for 'reproduction'. The general concept of evolutionary search can therefore be described as an iterative application of variation (v) and selection (s) operators to a population of solutions (P). This is illustrated in Equation 2.1 where P[t] is the population at time t (Michalewicz and Fogel 2000).

$$P[t+1] = s(v(P[t])) \qquad \textbf{(2.1)}$$

The selection process usually reflects a solution's fitness for survival. In other words, a fitter solution -in terms of its performance in its application domain- has a better chance of being selected for recombination and producing new solutions. Figure 2.6 illustrates a generic evolutionary algorithm.

2.2.3. Population-Based Strategies

The operative functionality of EAs on a family of candidate solutions rather than just a single point is a key advantage of these optimisation and search techniques. Starting from a certain population of potential solutions, an EA seeks to improve these solutions by filtering out relatively bad ones and exploring the search space for better approximations by applying operators borrowed from natural genetics.

EAs usually operate on a population of solutions which are encoded as strings and composed over some "alphabet". The encoded solutions are assimilated to "*chromosomes*", the basic unit of genetics. The chromosomes' values or "*genotypes*" are mapped onto the decision variable domain or "*phenotype*".

```
Procedure EA
  Begin
        Gen=0
         Initialise P(Gen)
        Evaluate P (Gen)
        While not finished do
         Begin
                Gen = Gen+1
                Select P (Gen) from P (Gen-1)
                Reproduce pairs in P (Gen)
                Evaluate P (t)
          End
End
```

Figure 2. 6 Generic Evolutionary Algorithm

Classical EA approaches, in particular the Genetic Algorithm (GA) approach, widely focused on binary representations (Holland 1975, Goldberg 1989) whereby chromosomes are constituted by the concatenation of binary strings. This reputed choice of representation was endorsed by theoretical work suggesting that the use of low cardinality alphabets results in more efficient schema processing (Goldberg 1991). Despite its classical popularity, binary representation suffers from several deficiencies such as its unsuitability for numerical search spaces and high-precision applications (Michalewicz 1996).

Other chromosome representations, such as integer representations and real-valued representations, are widely used. These representations are increasingly achieving usage interests, as they can result in several advantageous effects in many applications. Particularly, the use of real-value representation usually produces more efficient EAs (Wright 1991), as the adoption of this representation eliminates the need for converting chromosomes into their phenotypic values. Other benefits of such a representation is the reduced memory requirement needed by the whole optimisation process and the avoidance of the loss of precision that can result from the discretisation of phenotypic real values to binary encoding.

Nevertheless, choosing a suitable representation for the population of solutions operated by an EA is an application-dependent design choice. In other words, if a real-valued optimisation problem is tackled, a real-valued representation is the straightforward choice (Herrara, Lozano and Verdegay 1998). In some applications, such as the design of a specialized cantilever beam, a mixed chromosome representation is deployed (Deb and Goyal 1996).

After selecting the type of representation that best suits the application, the next step usually consists of creating an initial population of potential solutions. Alternative methods for this process are widely adopted. One way of initialising the population consists of creating a random generation of the required number of individuals using a random number generator that uniformly distributes numbers in the desired range of definition. An alternative way is the “extended random initialisation” procedure (Bramlette 1991) whereby a given number of random initialisation trials is processed for each individual which will be initialised to the best performance trial. In situations where the nature of the application is well understood in advance, or where EAs are used in conjunction with knowledge based systems, the initialisation can be held in the vicinity of previously known good solutions.

2.2.4. Evaluation

The objective functions reflect the raw performance of each individual in the problem domain. Minimisation and maximisation tasks are the two approaches usually adopted. In the case of minimisation, the fittest individual is allocated the smallest numerical value resulting from the objective function evaluation. After calculating the objective function for all the candidate solutions, a fitness function is then used to transform the objective function’s values into measures of relative fitness. These relative fitness values are then used by an EA for selection and breeding purposes (described later). Whilst the objective function is a domain-specific parameter, the fitness function can be used in several environments and problem domains. Many versions are adopted for the fitness function, nonetheless, the *proportional fitness assignment* (Holland 1975) which assigns fitness values computed using a certain function of the objective function values, and the *rank-based fitness assignment* (Baker 1985) which is based on the relative rank of a certain candidate solution in its population, are the most widely used versions. While the proportional fitness function ensures that each individual is allocated a probability for reproduction proportional to its relative fitness, it is susceptible to the problem of “premature convergence”, whereby certain locally fit solutions dominate the population leading the EA to converge to suboptimal regions of the search space.

The rank-based fitness assignment was suggested as a solution to this problem. This fitness assignment method consists of adding constraints on the reproduction range by limiting the number of offspring an individual can produce to a certain maximum so that no individuals will generate an excessive number of offspring, and thus preventing premature convergence.

2.2.5. Mating Selection

Having assigned the fitness values for every candidate solution, the next step will usually consist of a probabilistic selection of solutions. Based on their assigned fitness values denoting a reproduction expectation, solutions are selected for recombination, a process leading to the production of new solutions. Although the mating selection process, also termed as the *selection for variation*, is a stochastic process, solutions with high fitness values have higher chances for being selected for contributing to the next generation.

Roulette wheel selection, also called stochastic sampling with replacement (SSR) (Goldberg 1989), *Stochastic Universal selection (SUS)* (Baker 1987) and *Tournament selection* are different kinds of selection methods commonly used in a wide range of application domains. Other selection schemes have been recurrently proposed in the literature. A meticulous review of selection schemes can be found in Blickle and Thiele (1995) or Goldberg and Deb (1991).

Deploying a roulette wheel selection, the candidate solutions are allocated contiguous intervals whose length is in the range [0, sum], where "sum" denotes the sum of the expected selection probability of each individual. A random number (pointer) is then generated in the range [0, sum] and the individual whose segment spans the random number is selected. As a result, 'fitter' individuals occupy larger intervals and will have higher probabilities for being selected to breed and propagate to the next generations. The process is repeated iteratively until the desired number of individuals is obtained.

In the roulette wheel selection method, the segment size and thus the selection probability remain invariant through the whole process. Indeed any individual with a segment size bigger than zero could entirely fill the next population. In order to avoid this situation and decrease the chances of premature convergence, Stochastic Sampling with Partial Replacement (SSPR), an extension of SSR reduces the interval's size of an individual once selected.

Stochastic Universal Sampling is generally more efficient than SSR as it consists of *multi* pointers allowing the selection of the required number of solutions in a single run. Selection methods are numerous and possess different advantages and drawbacks. Tournament Selection, for example, is usually considered particularly suitable for optimisation problems involving noisy objective functions (Miller and Goldberg 1995).

2.2.6. Variation

In evolutionary computation, variation denotes the explorative processes responsible for introducing new individuals into a population of candidate solutions. It is only by applying the variation operators that the exploration of good solutions in the search space takes place and the optimisation process evolves. In the absence of variation, the search process converges to the best solution contained in the initial population. *Recombination* and *mutation* are the two operators that generally compose the variation process. Their notion is essentially borrowed from the counterpart recombination and mutation processes in natural biology. Similar to the selection process, numerous techniques and forms for the variation operators (recombination and mutation) are widely used in the literature. Choosing the right variation operators highly depends on the chromosome's encoding used for optimising a certain application.

<u>The Recombination Operator</u>

The recombination operator (also called the *crossover* operator) operates on the solutions' encodings (chromosomes or decision variables) and commonly requires two 'Parent' solutions. Nevertheless, multi-parent recombination operators exist and have been frequently used in the literature (Eiben, Raué and Ruttkay 1994). A comprehensive review of numerous recombination operators existing in the *genetic algorithm* (GA) literature can be found in Spears (1998). Based on a certain (usually high) recombination probability, the selected chromosomes recombine by exchanging genetic information and creating new individuals using the assumption that certain parts of the individuals' genes produce on average fitter individuals. The *single-point binary crossover,* whereby two parent solutions exchange their genotypic material after a certain -randomly picked- crossover point on their chromosomes, is one of the simplest recombination operators. The single-point binary crossover, illustrated in Figure 2.7, was originally implemented and deployed in the classical GA (Holland 1975, Goldberg 1989). An extension of the single point crossover is the *Multi-point Crossover* (DeJong 1975) whereby multi-crossover points are chosen on the chromosomes. The chromosomes' bits spanning consecutive crossover positions are then exchanged between the parents producing offspring sharing specific parts of both parents.

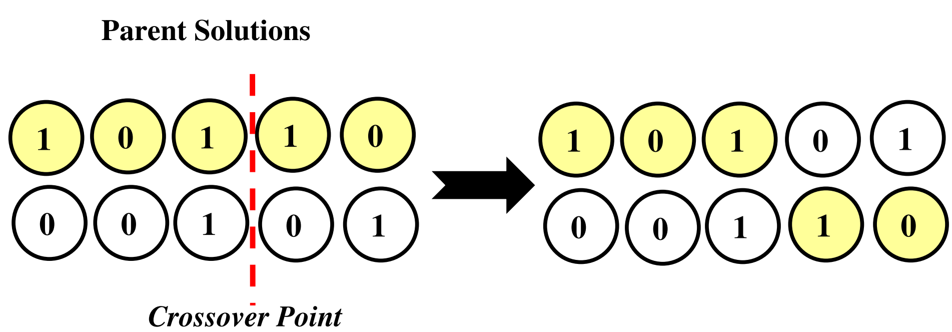


Figure 2. 7 Single-point Binary Crossover

Other crossover operators are widely known and used in several varieties of problems. The "*Uniform crossover*" (Syswerda 1989), the "*Shuffle operator*" (Caruana, Eshelman and Schaffer 1989) and the "*Reduced Surrogate*" (Booker 1987) are some of the other crossover operators which are used with binary chromosomes' representation. Alternatively, real valued chromosome representations possess a whole class of crossovers operators such as the "*Intermediate Recombination*" and the "*Line Recombination*" (Mühlenbein and Schlierkamp-Voosen 1993) which produce offspring phenotypes intermediate to the parents' numerical values. The simulated binary crossover (SBX) (Deb and Agrawal 1995) is another well-reputed crossover operator for real coded applications. Using the SBX operator, the offspring value for a certain decision variable is determined using a probability distribution with a standard deviation based on the distance (usually Euclidean) between the parents' values for that decision variable. A comprehensive review of real-coded operators can be found in (Herrara, Lozano, and Verdegay 1998).

The Mutation Operator

Early ES approaches used to consist of the selection and mutation operators only (Bäck, Hoffmeister and Schwefel 1991). Nevertheless, in most EC strategies, after a crossover is performed, mutation usually takes place. Unlike the recombination operator, mutation is a random process that operates on a single parent solution by modifying the value of a certain decision variable resulting in a new chromosome. Mutation is a backup operator ensuring the capability of escaping a certain local optimum. It is therefore usually considered as a safety operator which guarantees that the probability of searching a specific subspace is not zero. Mutation randomly changes the offspring resulting from the recombination process. Typically, mutation is applied with low probabilities in the range 0.001 and 0.01 (Goldberg 1989). The "bit-flipping" mutation is one of the simplest operators used with binary encodings, and consists of switching a certain randomly chosen bit from 1 to 0 (or 0 to 1). The bit-flipping mutation is illustrated in Figure 2.8.

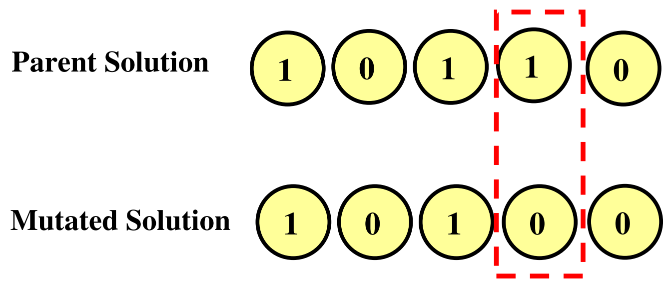


Figure 2. 8 Bit Flipping Mutation

On the other hand, real and integer encoded applications usually apply one of two conventional mutation approaches. The first approach consists of perturbing the chromosomes values in their domain of definition, while the second approach consists of randomly selecting new values for the decision variables within a specific allowed range. For these kinds of representations (Integer and Real value), it is shown that higher mutation rates can be more desirable and enhances the explorative capabilities of EAs (Tate and Smith 1993).

Differential Evolution

Differential Evolution (DE) (Price and Storn 1995) is an evolutionary variation operator which is particularly suited for real coded applications. DE can be assimilated to a variation operator combining both recombination and mutation processes. DE uses a self-adaptive strategy for determining the probability distribution parameters required for the recombination/mutation operators. For a good introduction about DE, the interested reader is directed to Lampinen and Storn (2004).

2.2.7. Selection for survival

The selection for survival process, also termed as the *environmental selection* or *reinsertion operator,* is the process of determining the subset of candidate solutions that will propagate to the next generation. Generally, the population of solutions handled by an EA has a predetermined size that has to be maintained all along the optimisation process. Nevertheless, at a certain generation '*gen*' of the optimisation process, the variation operator produces a new offspring population of solutions *P'(gen)* from the parent population *P(gen).* As a result, the total number of candidate solutions at a certain generation exceeds the bounded size of the population. A strategy for choosing a subset of the solutions produced at a certain generation is therefore required to control the complexity of the algorithm and maintain a fixed population size. Two major schemes for the selection for survival process are widely adopted, and are referred to in ES by the notations (μ+λ) and (μ, λ) (Beyer and Schwefel 2002) where μ denotes the parent population of solutions and λ is the offspring population of solutions resulting from μ. In the (μ+λ) selection for survival scheme, both populations (parent and offspring) compete for inclusion in the population which will survive to the next generation. Applying the concept of "Survival for the fittest", the best solutions usually get chosen for survival. However, other probabilistic selection strategies exist for selection for survival using the (μ+λ) scheme (Bäck 1996). The (μ, λ) selection scheme, on the other hand, only considers the offspring solutions for inclusion in the population *P (gen+1).* However, in some scenarios, the variation operators might be requested to produce fewer (or more) offspring solutions than the parent population. In such cases, the fraction difference between the initial and the new number of individuals is called a *generation gap* (De Jong and Sarma

1993). In order to maintain the number of individuals in a population, a reinsertion process of some of the parent (or alternatively a truncation process of some of the offspring) solutions is required. The common GAs' reinsertion strategy consists of filling the generational gap with the best-fit individuals from the parent population. Despite its simplicity, it has been shown that no significant convergence differences are noted when using such reinsertion schemes compared to other schemes such as the random insertion of solutions (Fogarty 1989). Another strategy consists of reinserting the oldest members of a population, i.e. the individuals that were able to survive several generations, consequently demonstrating good performances.

2.2.8. Exploration versus Exploitation

A good explorative search strategy should possess the capability of efficiently exploring the search space for new potentially good solutions and reaching any possible sub-region of the space. An exploitative strategy on the other hand, should be well adjusted for mining and making the most of use of a certain local region of the space with good potentials. Exploitation is hence usually required in the good 'neighbourhoods' of a search space, where the good solutions reside and can be identified.
The good performance of an EA highly depends on the right balance or tradeoff between the exploration process for new regions in the search space and the exploitation of already identified regions with good solutions (Holland 1975). For a good *e-e balance*, Bäck (1996) and Deb and Agrawal (1995) suggested automatic strategies for controlling the tradeoff between exploration and exploitation. Traditionally, the variation operators are considered as explorative processes in the EA community (Eiben and Schippers 1998). The selection process on the other hand, is widely regarded as an exploitative operator. Despite the direct metaphor from evolutionary biology and which can be inferred from this general classification, EA's theory is more complicated than that. Indeed, a single operator such as the 'blend crossover' (Eshelman and Schaffer 1993) can be used to control both exploration and exploitation processes.

Taking the requirement for a good *e-e* tradeoff into consideration, the impact of the exploitation process is commonly required to gradually increase, and exceed the impact of the exploration process as the search converges (Bäck 1992). This has been shown to be implicitly realised through the automatic reduction of the effect of the variation process (Eiben and Schippers 1998) when the search starts to converge.

2.2.9. Advanced Concepts in Evolutionary Computation

Probabilistic models

Another EA related search methodology, based on the probabilistic modelling of good areas of the search space, is also used in the community and known as '*estimation of distribution algorithms*' or EDAs. EDAs are beneficial in the way they try to decipher the genetic material (decision variables) and highlight the specific parts of the chromosomes which can be responsible for a certain performance. They differ from classical EAs in terms of the variation process and hence the way new solutions are produced. They operate by generating a new solution using the probabilistic model they built for certain good regions of the search space. Based on the performance of the new solution, the probabilistic model gets then updated accordingly (Pelikan, Goldberg and Lobo 1999). Despite their utility of studying and exploiting the relationships between the decision variables and the objective values, EDA's major drawback is their high computational requirement. The interested reader is directed to Pelikan, Goldberg and Lobo (1999) for a thorough review of EDAs.

Niche Formation: Fighting Genetic Drift

The need for *niche formation* is to prevent premature convergence towards sub-optimal regions of the objective space. Multimodal search landscapes are particularly susceptible for the premature convergence. Niche formation is a scheme that aims to maintain multiple subpopulations within a single global population operated by an EA. The multiple subpopulations are usually required to be maintained at distinct areas of the decision maker's region of interest. Commonly the objective space is the space where the niche formation is targeted, although this might be required in the decision variable space or both spaces simultaneously. Premature convergence is a detrimental consequence usually caused by the finite size of the operated population which can infer cumulative sampling errors on expected selection rates. This phenomenon, which also occurs in biological population genetics, is termed *genetic drift* and originates from the impact of the finite population sizes on the selection process (De Jong 1975). Several genetic drift countermeasures were suggested in the EA literature. De Jong (1975) suggested that increasing the size of the evolutionary population or, alternatively, increasing the mutation rate, can help reduce the effect genetic drift. More specialised techniques for addressing the genetic drift problem, such as *crowding* (De Jong 1975) and *fitness sharing* (Goldberg and Richardson 1987), were also suggested and widely used. The main idea behind crowding or fitness sharing is to promote less dense areas of the search space and decrease the selection pressure in the dense areas. The crowding approach consists of deterministically replacing a certain parent solution by its offspring solution in the case of high similarities between the two candidate solutions. Despite its simplicity, the crowding approach is not very efficient and well adjusted for tackling multimodal search spaces (Deb and Goldberg 1989). Fitness sharing on the other hand is an approach which seeks to decrease the fitness of the solutions lying in dense areas of the space. Moreover, the fitness value for a solution lying in a certain niche is decreased according to the number of neighbouring solutions which reside in the same niche. The function 'sh', illustrated in Equation 2.2, is the most commonly used sharing function.

$$sh(d) = \left\{ \begin{array}{ll} 1 - \left(\dfrac{d}{\sigma_{share}} \right)^{\alpha} & if\ d < \sigma_{share}, \\ 0 & otherwise \end{array} \right\} \quad \textbf{(2. 2)}$$

In Equation 2.2, 'd' denotes the distance[4] between two solutions[5], 'σ_{share}' represents the size of the niche and 'α' is a parameter used to shape the sharing function. After calculating the sharing value for all the solutions in the population 'P', the original fitness value of each candidate solution is then divided by its corresponding niche count (the sum of sharing values with respect to all the candidate solutions in the population, including itself).

$$f_i^{'} = \frac{f_i}{\Sigma_{j=0}^{N} \quad sh(d(i, j))} \quad \textbf{(2. 3)}$$

The modified fitness value for a certain solution 'i' is presented in Equation 2.3, where f_i denotes the original fitness value, 'N' is the total number of solutions and 'd (i, j)' is the distance between the solutions 'i' and 'j'.

Fitness sharing is a practical approach for niche formation, but its success is highly sensitive to the non-trivial choice of the niche size 'σ_{share}'. The choice of a niche size is however preferred to be relevant to the –usually unknown- landscape of the basin of attractions where the optimal solutions reside (Deb 2001). In the EC literature, several methods for estimating the niche size were proposed. The major techniques for estimating the niche size include the methods proposed by Deb and Goldberg (1989), Fonseca and Fleming (1993), Tan, Lee and Khor (1999), and Ray, Kang and Chye (2001). A rather notable approach for automatically selecting the niche size was suggested by Fonseca and Fleming (1995). The technique was based on the exploitation of the similarity between the kernel density estimation (specifically, Silverman's (1986) approach for the *Epanechnikov* estimator) and the fitness sharing function.

---

[4] The notion of Distance is usually an application dependent parameter, and is usually the Euclidean distance in most real coded applications.

[5] The distance between two candidate solutions is usually calculated in the objective space, although in some applications this might be equally required in the decision variable space.

Constraint Handling

A solution for a certain problem can either be feasible or infeasible. A feasible solution satisfies all the constraints that might be imposed on it. The constraints' complexity can vary from a simple logical statement such as "Price is less than £10000" to very complex functions that define a valid search space. Solving real-world problems most often requires the simultaneous consideration of multiple constraints. As a result, the process would usually consist of finding a feasible solution that does not violate such constraints. Unless equipped with a specific constraint handling strategy, stand-alone EAs do not automatically allocate any consideration to the constraints. In Coello (2002) and Michalewicz and Schoenauer (1996) a comprehensive survey of the different constraint handling techniques can be found. In the following, a brief illustration of four of the most commonly used techniques for handling constraints in EAs is illustrated.

- **Special representations and operators**

This class of constraint handling techniques constitute a rather specialised approach for dealing with constraints in specific problems. This approach can be briefly described as the custom made design of specialised representation schemes and variation operators that only generate feasible solutions to certain specific problems. Examples of some special representations and operators can be found in Davis (1991).

- **Penalty Functions**

Penalty functions are classical approaches for dealing with constrained optimisation problems and date back to the early 1940s (Courant 1943). When incorporated in an EA, the penalty function scheme operates by straightforwardly not allocating any direct consideration to the constraints, but reducing the fitness of the solutions that violates any constraint instead. The fitness value of a certain solution that violates some constraints is therefore penalised by usually using a certain function of the level of constraint violation. As a result, the search process is biased towards the areas of the search space that satisfy the constraints.

- **Repairing infeasible solutions**

Instead of penalising infeasible solutions, this constraint handling approach seeks to compensate and rectify any constraint violation in a certain infeasible solution. This is usually done by reflecting any out of bounds decision variables back to their domain of definition or deploying a certain local search approach in the neighbourhood of the infeasible solution in order to move such solution to the next feasible solution. There is no generalised repair technique, but instead, the process is usually designed to address a certain specific application. Nevertheless, repairing infeasible solutions is a practical approach (Liepins and Potter 1991) and is especially suitable for combinatorial optimisation problems (Coello 2002).

- **Multiobjective formulation**

An approach for dealing with constraints is their formulation as objectives to be optimised alongside the 'real' objectives in a multiobjective optimisation scenario (Coello 2000b). Consequently, the constructed multiobjective optimisation problem can be approached using evolutionary multiobjective techniques such as Pareto optimality, non-dominance, the preferability operator (Fonseca and Fleming 1998) or the constrained domination (Deb, Pratap, Agrawel and Meyarivan 2002).

Population Topology

Three different topologies for the population of candidate solutions handled by an EA are usually used in the EC literature. These are the global topology, the island topology and the diffusion topology (Chipperfield and Fleming 1995). Whilst the global topology is the simplest and the most commonly used approach, the island and the diffusion models ensure parallelism and the ability of running an optimisation process on multi processors. For more information about the different population topologies, the interested reader is directed to Chipperfield and Fleming (1995).

Fitness Approximation in Evolutionary Computation

The population-based nature of EAs usually trails a considerable computational burden, which is especially emphasised when optimising expensive objective functions. At every generation of an optimisation tackled by an EA, multiple evaluations of the objective function are simultaneously required. Fitness approximation is a well-established remedial measure for reducing the expensive costs usually imposed by EAs. A good review of fitness approximation can be found in Jin (2005). The main idea of fitness approximation is the substitution of the objective function evaluations with the evaluation of a cheaper and simplified model which approximates the exact objective function and captures it as close as possible. The cheaper models are widely known as the meta-models and are usually deployed in expensive optimisation problems (e.g. C.Varcol and Emmerich 2005, Emmerich and Noujoks 2004) or noisy optimisation problems (Regis and Shoemaker 2004). In order to design a meta-model for a certain application, a metamodelling technique is required. Metamodelling is an entire research area, and many well-established techniques for producing meta-models have been proposed and used in a variety of applications. Neural networks (NN) (Hornik, Stinchcombe and White 1989), Kriging methods (Jin, Chen and Simpson 2001), polynomial regression (Chen *et al* 1996 and Simpson *et al* 1998), Gaussian processes (Ulmer, Streichert and Zell 2003) and support vector machines (Abboud and Schoenauer 2002) have all been used as metamodelling techniques. A comprehensive survey and analysis of some of the most established metamodelling techniques can be found in Jin, Chen and Simpson (2001) or Rasheed, Ni and Vattam (2003). An essential matter that should be noted when deploying metamodelling techniques is that the solutions achieved for a meta-model should be cautiously analysed before being considered as solutions to the exact model which is usually of higher fidelity (Michalewicz and Fogel 2000). Interchangeably deploying the exact- and the meta-model, a practice known as *controlled evolution*, is a strategy suggested by Jin *et al* (2000) to avoid misleading the search process through the sole usage of a meta-model. In the controlled evolution, the evaluations of the exact objective function can also be actively used for enhancing the fidelity of the metamodel (Bull 1999).

Memetic Algorithms: A Hybrid Approach

"Memetic Algorithm" is a concept first introduced in 1989 by Moscato (1989). The term "Memetic" has its roots in the word "meme" introduced by Dawkins (1976) and which denoted the "unit of imitation" in cultural transmission. Memetic algorithms, also called hybrid evolutionary algorithms, are increasingly thriving metaheursitics for solving multiobjective optimisation problems. The essential idea behind Memetic Algorithms is the hybridization of local search refinement techniques within a population-based strategy, such as genetic algorithms. Memetic Algorithms share most of GAs' characteristics although they introduce a new improvement procedure based on local search in the neighbourhood of newly generated individuals usually resulting from the recombination operators. The main conceptual difference between genetic algorithms and memetic algorithms is the approach and view of the information's transmission techniques. Whereas genetic information carried by genes is usually transmitted intact to the offspring (e.g. genetic algorithms), "memes" the base unit of memetic algorithms are typically adapted by the individual transmitting them. These hybrid algorithms were applied to a wide variety of problems such as image segmentation (Bhanu, Lee and Das 1995), multiobjective optimization of space allocation problems (Burke *et al* 2001), radiotherapy treatment planning (Haas, Burnham and Mills 1998) and molecular geometry optimisation (Hodgson 2000). They have proved to be highly effective, outperforming similar approaches such as pure evolutionary algorithms in several application domains in terms of convergence towards Pareto optimal solutions.

While genetic algorithms are good at exploring the solution space due to their search from a set of candidate solutions and not just from a single point, they are not well suited for fine-tuning structures that are close to optimal solutions. As stated by Davis (1991) and re-illustrated by Knowles (2002), for improving optimization results achieved by genetic algorithms one should: “Hybridize where possible”. The hybridization of local improvement operators among the evolutionary steps of an EA is essential to deal with near optimal situations. This has been shown (Knowles (2002)) in several application domains to bring improvements to the standard results achieved by stand alone genetic algorithms in terms of results quality and speed of convergence. The combination of global and local search is a strategy used by many successful global optimization approaches, and has in fact been recognized as a powerful algorithmic paradigm for evolutionary computing (Goldberg 1989).

Advantages and Disadvantages of EC

The motivation for using EC as a search and optimisation technique is partly dependent upon the nature of the application under consideration. In other words, if the optimisation problem under investigation is mathematically “well-behaved”, then the appropriate choice would be the use of conventional, deterministic and specialized techniques (Michalewicz and Fogel 2000). Examples of such techniques are the classical *linear* and *nonlinear programming* techniques (Luenberger 1984), the deterministic gradient-based techniques such as *Newton* method and its derivatives and the *simplex* method (Dantzig 1963). However, these conventional optimisation techniques face major difficulties in several scenarios. For example, when the objective function is discontinuous or characterized by many local optima and points at which gradients are undefined, or when the estimation problem involves many parameters that interact in highly non-linear ways, classical optimisation techniques usually fail. In these situations, heuristic methods like EAs are powerful alternatives for exploring search spaces and finding good solutions that cannot be detected by conventional numerical techniques.

Another major attraction of EC techniques is their flexibility and thus, their applicability to a vast variety of applications. This is an essential advantage of EC when compared with alternative optimisation and search methodologies. Classical optimisation techniques usually impose restrictions and require a lot of initial conditions and hard constraints, such as the linearity of an objective function or the shape of a tradeoff surface. EC techniques, such as EAs, are direct search methodologies whose functionalities only involve the calculation of the objective function (exact or metamodel), and do not require any other derivative or theoretical information. A decisive assessment and a choice of a solution is usually achieved through human interaction or a computerised simulation. EC techniques can also operate on any problem encoding and accept a wide variety of data structures while ensuring the possibility of devising suitable operators.

Nevertheless, the iterative and population based nature of EAs can be quite computationally expensive and should be taken into consideration, especially when tackling expensive objective functions. In addition, in order to achieve ‘good’ results using an EA, several design choices need to be cautiously decided. These design choices, among others, include the population size, the mutation probability, the recombination probability and the e-e tradeoff. Choosing the right parameters for an EA is far from being a straightforward process, and requires practical EA expertise. Moreover, the success of certain EA parameters and operators cannot be generalised (Purshouse and Fleming 2003b), instead these parameters should be domain specific and tailored to fit a certain application or search landscape. Some research was dedicated for designing methodical approaches, also known as *competent EAs*, for supporting the user in choosing the right parameters and designing the appropriate EA that accommodates the application at hand (Reed, Minsker and Goldberg 2000 and Lobo and Goldberg 2001). This research area is critical and requires more dedications.

2.3. Evolutionary Algorithms for Multiobjective Optimisation

2.3.1. EAs’ Convenience for *Multiobjective* Optimisation

Classical approaches for solving multiobjective optimisation chiefly consist of multiple-start strategies whose core is the transformation of a MOP into a single objective counterpart. Once converted into a single objective optimisation problem, it is possible to deploy a large variety of conventional optimisation techniques for solving the problem. Using the classical *weighted sum* approach (Jakob, Gorges-Schleuter and Blume 1992) for example, different weights are required to be assigned to the multiple objectives prior to the start of the optimisation problem. The weights’ values should be carefully chosen to reflect the importance and the priority of each objective. The optimisation problem is then converted, by aggregating the weighted objectives, into an optimisation of a weighted sum, therefore converting the multiobjective problem into a single objective problem. The success of the weighted sum approach is critically linked to the *a priori* choice of objectives’ weights, which is generally not a straightforward process from the DM point of view.

Other widely used multiple-start strategies include the ε-constraint method and the goal programming approach. The *ε-constraint* method (Haimes, Lasdon and Wismer 1971) consists of transforming all but one of the objectives into constraints. The standard optimisation problem (illustrated in Figure 2.1) is consequently converted into a constrained single objective optimisation problem (Definition 2.6). The ε values used to define the constraints are usually provided by the DM.

Definition 2. 6 ε-constraint Method

$$\begin{aligned} &\min z_i(\mathbf{x}) \\ &subject\ \ to \quad z_j(\mathbf{x}) \leq \varepsilon_j \qquad for\ j = (1,2,....n) \quad j \neq i \end{aligned}$$

Goal programming, on the other hand, is a term which was officially first introduced in 1961 by Charnes and Cooper (1961). The method reduces a multiobjective optimisation problem into a single objective problem by transforming it into a minimization task of a weighted sum of deviations of the objectives from a certain defined goal vectors (Aouni and Kettani 2001).

The major drawback of these classical MO approaches is that they can only produce a single solution on the tradeoff surface at a certain execution. In order to produce a diverse set of solutions on the tradeoff surface, these classical strategies need to be re-executed many times while changing the weight values and the optimiser's configuration. In the case of the weighted sum approach, it was geometrically demonstrated that the method cannot produce solutions on non-convex regions of the tradeoff surface (Fleming and Pashkevich 1985). In fact, it was shown that the convexity of the tradeoff surface is a hard constraint that should be met in order to produce all the Pareto optimal solutions (Censor 1977) using the weighted sum approach. The multiple-start strategy makes these techniques computationally expensive and requiring a considerable amount of objective function evaluations. In addition, these classical strategies do not make any use of the online interaction of the objectives and requires the aggregation of usually non-commensurate objectives.

On the other hand, in addition to their advantages for single objective optimisation (described in Section 2.4.9), EAs are particularly suitable techniques for solving multiobjective optimisation problems. The population-based nature of EAs and their flexible selection mechanisms have proved to be extremely successful for solving multiobjective optimisation problems. The latter two factors make these optimisation strategies very practical in revealing a satisfactory approximation set to the desired globally optimal set of solutions in a single execution of the algorithm. Moreover, EAs' simultaneous operation on multiple solutions makes the search for optimal solutions a more cooperative process and hence a faster process. Moreover, the Pareto dominance scheme that governs the majority of EC strategies makes it possible to tackle MOPs and assess candidate solutions to such problems without requiring the aggregation of non-commensurate objectives. This is an essential advantage of EAs for tackling multiobjective optimisation problems which allows the preservation of domain specific knowledge and precision as well as it promotes the better understanding of the search landscape and the objective tradeoffs all along the search progress. In Table 1, an illustration of the differences between EAs and classical optimisation techniques is presented.

Table 1. 1: EAs versus Traditional Methods

Evolutionary Algorithms	Classical Methods
Search a population of points in parallel	Operate on a single point
Do not require derivative or auxiliary information	Generally requires derivative information
Use probabilistic transition rules	Generally use deterministic transition rules
Do not require the aggregation of non-commensurate objectives	Generally require the aggregation of non-commensurate objectives
Produce multiple solutions per execution	Produce a single solution per execution

2.3.2. A Chronological Illustration of MOEAs Development

Non Pareto-Based Approaches

In Figure 2.9 a chronological chart illustrating the different MOEAs devised over the last 20 years and their major additions and contributions to the EMO community is presented. Despite the early promotion of GAs as a potential approach for solving MOPs (Rosenberg 1967) in the late 60s, and despite an early attempt for deploying GAs to solve a MOP (Ito, Akagi and Nishikawa 1983), it was only in 1985 that the first actual implementation of a multiobjective evolutionary optimiser, aimed at producing an approximation set, was introduced. Moreover, it is widely accepted in the evolutionary multiobjective optimisation (EMO) community that Schaffer's Vector Evaluated Genetic Algorithm (VEGA) (Schaffer 1985) represents the first implementation of a multiobjective evolutionary algorithm (MOEA).

VEGA

The Vector Evaluated Genetic Algorithm (VEGA) consists of a simple non-Pareto based genetic algorithm. Schaffer's VEGA was inspired by Grefenstette's GENESIS (Grefenstette 1984) program which was designed for solving single objective optimisations. VEGA operates by dividing the handled population of solutions into '*N*' equally sized subpopulations at every generation of the process. Each subpopulation was designated to one of the '*N*' objectives that constituted the optimisation problem.

A proportional selection mechanism was applied to define the membership of a certain solution to a particular subpopulation and variation (crossover and mutation) operators were then applied to the whole population. The major drawback of VEGA lies in its selection procedure. A solution with no particularly remarkable performance in terms of a certain objective is biased against and its chances for survival to successive generations are limited. In other words, the selection scheme used in VEGA prevents the preservation of compromise solutions on the tradeoff surface whilst only keeping solutions that excel in terms of a certain objective. This last problem is commonly known in genetics as the *speciation problem*.

Other early MOEAs included Fourman's (1985) *lexicographic ordering GA* (described below), the *target vector optimisation* (Wienke, Lucasius and Kateman 1992) which is based on the goal programming approach and Kursawe's (1991) *Vector Optimised ES* (VOES). Kursawe's VOES deploys a chromosome encoding which is more closely inspired from natural biology. Using VOES, each chromosome comprises a dormant and a recessive part and consequently possessed two objective vectors (one for each part of the chromosome). Despite its sophistication, VOES uses a selection scheme whose core is similar to the one used in VEGA, and hence presents the same speciation problem.

Lexicographic ordering GA

The lexicographic ordering approach consists of ordering the objectives in terms of their importance and priority. This objective ordering is performed before the start of the optimisation process. The optimisation task then consists of optimising a single objective function which corresponds to the objective with the highest priority. The next steps then consist of optimising the remaining objective functions, one at a time, and in a descending order in terms of the objectives' priorities. Moreover, each optimisation task takes consideration of the objectives that precedes it in terms of priority. For example, when optimising the second objective function (which corresponds to the second prioritised objective), the only solutions kept are the ones that do not deteriorate the values of the solutions previously found for the preceding objective. Determining the objective's priorities and ordering them is the major drawback of this technique. The lexicographic ordering approach suffers from the same problems as those of the weighted sum approach.

Pareto-Based Approaches

In order to overcome the drawbacks of VEGA, mainly linked to its selection mechanism, Goldberg (1989) proposed the use of Pareto dominance for assigning fitness scores to candidate solutions and consequently selecting solutions for recombination and survival. Goldberg also suggested the fitness sharing approach to promote diversity in the approximation set. Most of the modern MOEA approaches are influenced by Goldberg's work (Coello et al 2002), and can be categorized into two consecutive generations.

First Generation MOEAs

The Multiple Objective Genetic Algorithm (MOGA) (Fonseca and Fleming 1993), the Niched Pareto Genetic Algorithm (NPGA) (Horn, Nafpliotis and Goldberg 1994) and the Non-Dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb 1994) are the most reputed and cited MOEAs of the first generation. The major difference between these three MOEAs is their approach for deriving fitness scores from the Pareto dominance scheme and assigning them to the alternative candidate solutions. In addition, most of the first generation MOEAs incorporate fitness sharing among their processes (Goldberg and Richardson 1987) to encourage the production of well-distributed approximation sets. These MOEAs lack the concept of elitism, a critical concept ensuring the preservation of good solutions all along the optimisation process. The emphasis of this first generation of MOEAs is the simplicity of the approaches and the lack of methodology for assessing and contrasting the performance of MOEAs.

MOGA and its Contributions

Following a comparative study by Van Veldhuizen (1999), the Multiple Objective Genetic Algorithm (MOGA) was found to outperform NPGA and NSGA. The strategy used in MOGA (Fonseca and Fleming 1993) for assigning ranks, known as the *Pareto-based ranking*, corresponds to a slight alteration of Goldberg's suggested ranking technique. In MOGA, a candidate solution is ranked according to the number of solutions in the current population that dominates it. In other words, a *non-dominated* solution would have been assigned the rank *zero* denoting that there is no solution in the population dominating it. Compared to the non-dominated sorting approach for ranking solutions (Goldberg 1989, Sirnivas and Deb 1994), the Pareto-based ranking used in MOGA presents a more efficient and a higher resolution ranking. Pareto-based ranking was later used in the multi-objective mixture-based iterated density estimation evolutionary algorithm (MIDEA) by Thierens and Bosman (2001). Moreover, also in the context of Fonseca and Fleming's Pareto-based ranking (1993), Hughes (2001) implemented a method for calculating the probability of correctly deciding on the dominance of a certain solution over another in the presence of Gaussian noise and uncertainty.

In MOGA, fitness sharing is also implemented and applied in the objective space in order to promote diversity and combat genetic drifts.

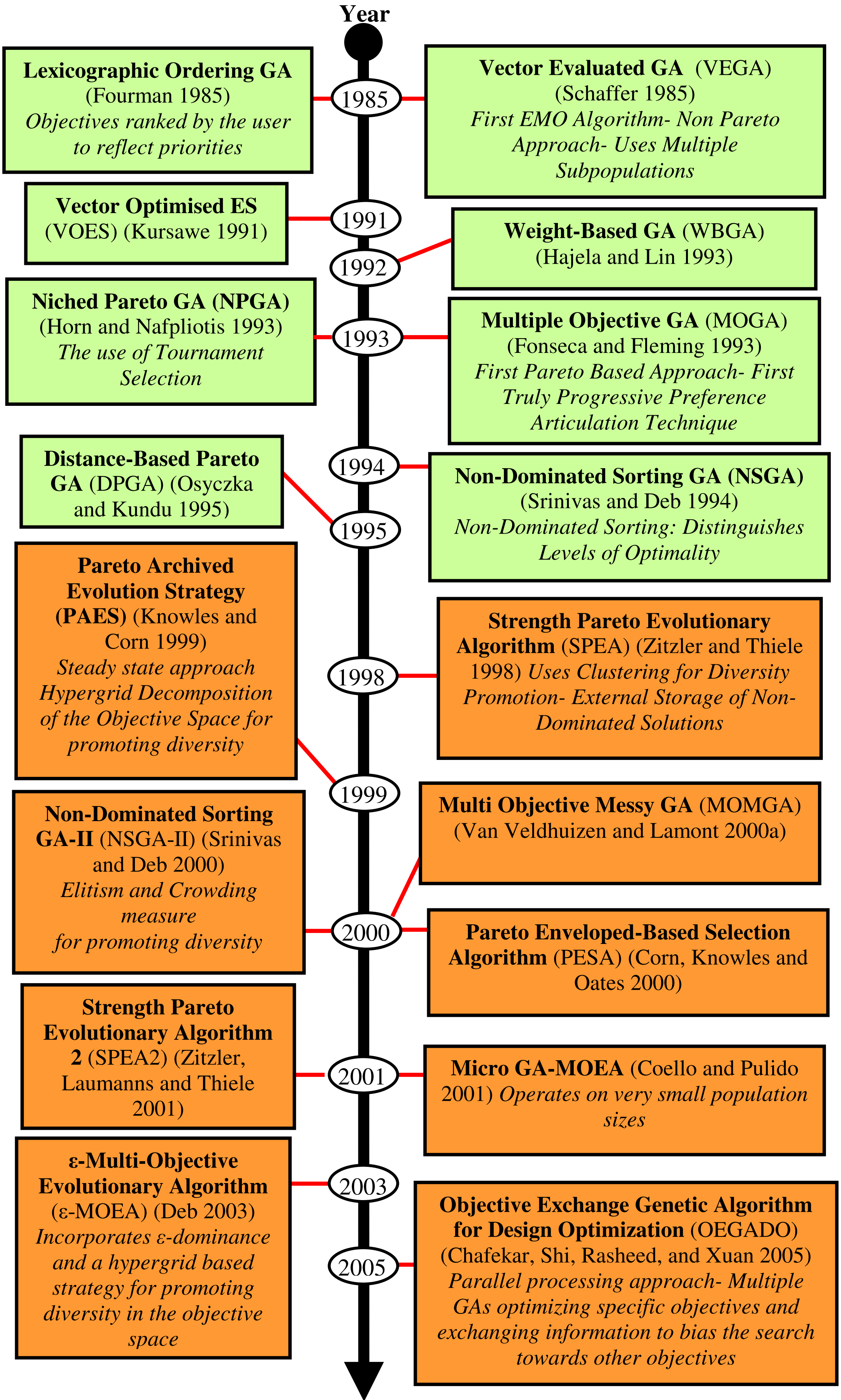


Figure 2. 9 Different MOEAs (1st generation – 2nd generation) proposed over the years

<u>NPGA and its Contributions</u>

The major contribution of the Niched Pareto Genetic Algorithm (Horn and Nafpliotis 1993, Horn, Nafpliotis and Goldberg 1994) is its *Pareto domination tournament* scheme of selection. First, two candidate solutions are chosen at random from the population. A certain number ‘n’ of solutions is then selected randomly from the population to constitute the benchmark comparison set.The two selected solutions are compared with the comparison set of solutions. If one of the two solutions is non-dominated by the comparison set, then it is selected for variation. In the case of a tie (i.e. both selected solutions are non-dominated by the comparison set or alternatively dominated by at least one of the solutions in the comparison set), the two solutions are considered equivalent, and an *equivalence class sharing* is performed. Based on fitness sharing approach, the solution residing in the less dense area of the search space is selected.

<u>NSGA and its Contributions</u>

The Non-Dominated Sorting Genetic Algorithm (NSGA) was based on the *non-dominated sorting strategy* which was originally proposed by Goldberg (1989). A non-domination check is first performed on the whole population of solutions, and the non-dominated solutions are assigned the rank ‘zero’ denoting their membership with the first (and highest) level of non-dominance. The non-domination check is then applied on the population of solutions while excluding the solutions belonging to the rank zero. The process continues until all the solutions in the population are assigned a rank denoting their belonging to a certain non-dominance level. After ranking all the candidate solutions, fitness sharing is applied to every class of dominance. This is done in order to degrade the fitness values of the solutions, based on their density estimate in the objective space, within their class of dominance and hence without degrading their class of dominance itself. The multiple non-dominance checks at every generation of the process and the high sensitivity to σ_{share} resulted in a less efficient performance of NSGA when compared to MOGA.

Second Generation MOEAs

The second generation of MOEAs incorporate *elitism* and deploy more sophisticated methods for promoting solution diversity. An increased emphasis and dedicated research was also given to improve the efficiency of these evolutionary approaches at the end of the 1990s which resulted in the establishment of the second generation of MOEAs. The most representative and widely cited MOEAs of the second generation include, among others, the Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler and Thiele 1999), the Pareto Archived Evolution Strategy (PAES) (Knowles and Corne 2000), the Non-Dominated

Sorting Genetic Algorithm II (NSGA-II) (Deb *et al* 2002) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler, Laumanns and Thiele 2001).

SPEA and its Contributions

The Strength Pareto Evolutionary Algorithm (SPEA) operates on two active populations of solutions. The usual population of solutions and an *external population* or online archive which stores the non-dominated solutions and participates in the variation processes. Before the implementation of SPEA, the majority of MOEAs, such as MOGA, NPGA and NSGA, used an offline archive with an assumed infinite memory to store all the non-dominated solutions achieved all along the optimisation process. The solutions stored in the offline archive were generally passive in that they did not interfere or participate in the evolutionary search process. The fitness assignment process in SPEA is more fine-grained compared to MOGA's Pareto based-ranking. This is mainly due to the strength-based approach implemented in SPEA which takes into consideration the number of solutions that are dominated by a certain solution 'S' (also called *Dominance count*) and the number of solutions that dominate 'S' (also called *Dominance rank*). Dominance rank, dominance count and dominance depth are terminologies suggested by Zitzler (2002) and denote the methodologies that MOEAs use to rank candidate solutions while favouring locally non-dominated ones. *Dominance depth* is the methodology used to rank a certain candidate solution based on the non-dominated sorting approach (e.g. NSGA). In a similar manner to MOGA's ranking procedure, each solution in the external population is first assigned a *strength* value which is proportional to the number of solutions (in the population) that it dominates. Each solution in the population is then assigned a fitness value which is proportional to the strength of the solutions (in the external population) that dominate it. In order to maintain a diverse approximation set, SPEA ensures that the solutions in the external archive are evenly distributed using a *clustering* approach.

The *clustering* approach seeks to distribute the solutions in the external archive into N different clusters. Initially, each of the N' ($>N$) solutions in the population is considered as a separate cluster. A pair-wise measurement of the Euclidean distances between each of the N' clusters and the remaining $N'-1$ clusters is calculated, and the two closest clusters are merged to form a single cluster. This procedure reduces the number of clusters from N' to $N'-1$. The whole process is then executed on the reduced number of clusters until the required number of clusters N is achieved. When multiple solutions reside in a cluster, the Euclidean distance between two clusters is measured as the average Euclidean distance between all pairs of solutions, one from each cluster. This is illustrated in Figure 2.10 where the clustering process is highlighted. The clustering approach for promoting diversity is usually considered a more fine-grained technique compared to the crowding approach (described later) used in NSGA-II (Deb, Mohan and Mishra 2003) but is generally more computationally expensive because of its required measurements and pair-wise comparisons which are repeated at every iteration of the process. In Deb, Mohan and Mishra (2003), a substitution of the crowding measure in NSGA-II with SPEA's clustering technique was attempted. Better diversity was achieved only at the expense of a more computationally expensive process.

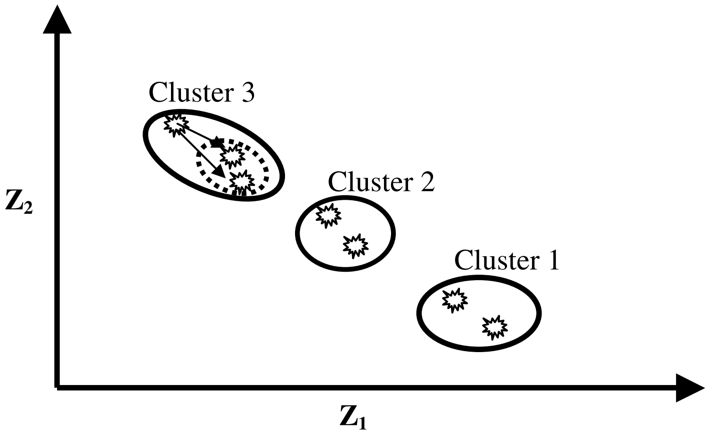


Figure 2. 10 Clustering Technique for Diversity Promotion

Moreover, despite some earlier considerations of elitism for MO (Tamaki, Mori, Araki, Mishima and Ogai 1994 and Husbands 1994), the publication of SPEA (Zitzler and Thiele 1999) in the *IEEE transactions on Evolutionary Computation* earned the authors the credit of formally introducing the concept of elitism.

PAES and its Contributions

The Pareto Archived Evolution Strategy (PAES) (Knowles and Corne 2000) is a simple (1+1) evolution strategy whereby a single parent solution produces a single offspring. PAES uses an archive (with an upper bound on its size) that contains all the non-dominated solutions which were previously found. This archive implements the elitism concept and plays the role of a reference set. The performance of the mutated solution (offspring solution) is assessed by comparing it to the performance of the solutions in the reference set. However, the major feature of PAES is its strategy for promoting diversity in the approximation set. PAES uses an adaptive (hyper-) gridding system in the objective space to divide it into several non-overlapping (hyper-) boxes. The belonging of a certain solution to a certain region in the (hyper-) box is determined by the objectives' values which define the solution's coordinates. In the case where an offspring solution is non-dominated by the reference set, a crowding measure based on the number of solutions residing in a certain grid location is applied to determine whether the offspring solution is accepted or not. The major advantage of this diversity maintenance technique is that it does not require setting any additional parameters such as the niche size parameter σ_{share} in the fitness sharing approaches. The adaptive grid archiving (AGA) concept used in PAES later inspired several researchers, and was altered and deployed in multiple MOEAs such as the Pareto Envelope-based Selection Algorithm (PESA) (a population based version of PAES) (Corne, Knowles and Oates 2000), Coello and Pulido's (2001) Micro Genetic Algorithm and Deb, Mohan and Mishra's (2003) ε-Domination Based Multi-Objective Evolutionary Algorithm (ε-MOEA).

<u>NSGA-II and its Contributions</u>

NSGA-II was introduced in Deb et al (2000) (and Deb et al 2002)) as an enhancement to NSGA. In this enhanced approach, a candidate solution is ranked using the non-dominated sorting scheme but taking into consideration the dominance rank and the dominance count for a better dominance resolution. Moreover, an efficient density estimate approach (*crowding*) in the objective space is deployed for promoting diversity at the selection for variation and survival processes. The density estimate for a certain candidate solution 'S', known as the *crowding distance*, is measured by calculating the sum of the Euclidean distances (in the objective space) between the two neighbouring solutions from either side of 'S' along each of the objectives (i.e. in an objective-wise fashion). The crowding technique is illustrated in Figure 2.11 on a bi-objective problem. More generally, all the solutions are first sorted in descending order in terms of the first objective. Each solution is then assigned an objective-wise crowding measure equal to the normalised difference between its two neighbouring solutions. The crowding distance for a certain solution on a non-dominated front is then calculated as the sum of all such objective-wise crowding measures. Unlike SPEA and SPEA2 (described below) which use an external archive, NSGA-II uses a simple $(\mu+\lambda)$ scheme for the selection for survival.

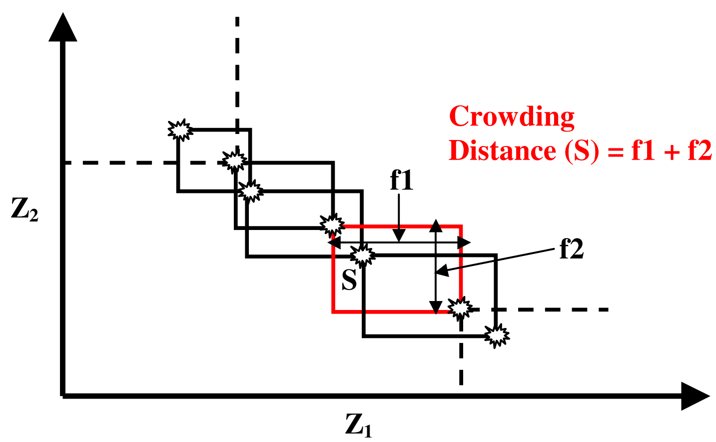


Figure 2. 11 Crowding Technique for Promoting Diversity

Hence, after sorting the combined population of solutions (Archive + Current Population) into different classes of dominance, the crowding distance is calculated for the whole combined population. A Pseudocode of NSGA-II will be given in Chapter 3. In the EMO community, NSGA-II is widely regarded as an efficient optimiser against which news optimisers should be assessed.

SPEA2 and its Contributions

Again, SPEA2 was aimed at improving the performance of its predecessor SPEA which suffered from some weaknesses. These weaknesses were highlighted and criticized in the EMO community at more than one occasion (Corne et al. 2000, and Deb, Pratap, Agarwel and Meyarivan 2000). Three new features were implemented in SPEA2 and differentiated the latter optimiser from its predecessor. The first improvement consisted of ameliorating the granularity of the ranking technique used in SPEA by taking into consideration the count of the dominating and the dominated solutions when ranking *every* solution in the population. The second improved feature implemented in SPEA2 consisted of deploying the k-th nearest neighbour strategy as a density estimate around every individual. The last modification introduced in SPEA2 altered the truncation method, which maintains a fixed size for the external population, ensuring that boundary solutions do not get filtered out.

For an alternative survey about the features and the development of the major MOEAs introduced over the years, the interested reader is directed to Coello (2006) and Ghosh and Dehuri (2004).

Performance Metrics and Test Functions

In addition to the algorithmic enhancements and the various efficient and sophisticated techniques suggested for promoting diversity and proximity in EMO, the second generation of EMO witnessed many other major contributions. Notably, among these contributions are the formulation of structured performance criteria and the introduction of several assessment techniques and test functions for a better analytical and quantitative evaluation of MOEAs. Two main performance criteria for MOEAs were delineated. More specifically, the distance between the approximation set and the true Pareto front should be minimised, and the uniform diversity of the approximation set should be maximised along the Pareto front.

In order to assess the performance of a MOEA in terms of the above criteria, several unary performance metrics were proposed. The *Error Ratio* (Van Veldhuizen 1999) and the *Generational Distance* metric (Veldhuizen 1999) are examples of unary metrics suggested for assessing the closeness of an approximation set to the Pareto optimal front. On the other hand, the *Spread metric* (Deb et al 2000a), the *Maximum Spread* (Zitzler 1999) and the *Chi-Square-Like Deviation Measure* (Deb 2001) are examples of unary metrics that evaluate the diversity of an approximation set. For more information about performance metrics in EMO, the interested reader is directed to Deb (2001), Van Veldhuizen and Lamont (2000b), and Deb *et al.* (2000).

Nevertheless, a theoretical study by Zitzler et al (2003) showed that there is no combination of unary metrics that can conclude the out-performance of a certain MOEA over another. As a remedial measure, the authors proposed the use of binary metrics which contrast the relative performance of two MOEAs simultaneously. Examples of binary metrics include the *Set Coverage metric* (Zitzler 1999) and the *Binary ε-metric* (Zitzler *et al* 2003).

Moreover, in order to challenge the search capacities of MOEAs, a set of bi-objective test functions (Zitzler, Deb and Thiele 2000) as well as a set of seven scalable test functions (in terms of the number of objectives and decision variables) (Deb, Thiele, Laumanns and Zitzler 2002) were introduced. These test functions encapsulate characteristics, such as multi-modality and discontinuity, which are known to generally cause difficulties to most MOEAs. Also, the scalable test functions were intended for the evaluation of the performance of MOEAs when dealing with an increased number of competing objectives. These test functions are widely used as benchmark problems in the EMO community and were designed based on the suggested methodology by Deb (1999b), which is a blue-print for constructing test functions with challenging features to MOEAs. Huband et al (2005) also introduced a practical toolkit which allows the designer to construct scalable multiobjective test functions with well-defined Pareto fronts and desired characteristics. Using their toolkit, Huband et al (2005) also proposed a test suite of 9 scalable multiobjective problems featuring important characteristics such as multimodality and non-separability[6].

2.3.3. Obtaining Good Proximity

The proximity (or otherwise the convergence) of an approximation set to the Pareto front is the primary requirement of evolutionary multiobjective optimisation. In order to achieve proximity, the search process should be steered in the right direction towards the Pareto optimal front of a certain multiobjective problem. This steering is more accurately achieved through the selection processes that govern MOEAs. As a result, fitter solutions, hence closer to the Pareto front, have higher chances for being selected for contributing to the next generations through the variation operators. Additionally, at the environmental

[6] A non-separable MOP is a problem characterised with variable dependencies. Non-separability is a desired feature which is common in real-life applications.

selection process, fitter solutions would equally have higher chances for being selected for survival to the next generations.

Based on the Pareto dominance scheme, several techniques for ranking candidate solutions and assigning them fitness values were proposed in the EMO community. The major ranking approaches devised over the two generations of EMO were described in Section 2.3.2. The interested reader is also referred to Zitzler, Laumanns and Bleuler (2004) for more information about ranking and fitness assignment in EMO. The fitness values assigned to alternative candidate solutions to a MOP usually reflect their relative performance, and hence their preferability in terms of closeness to the Pareto front. These fitness values are then used as the primary selection criteria for variation and contribution to the next generation.

A major approach for improving the convergence of the population of solutions handled by a MOEA towards the Pareto front is the use of *elitism*. Elitism is a strategy which aims to ensure that any good solutions found during the optimisation do not get filtered out and lost (Zitzler *et al.* 2004). The implementation of an elitist strategy can be achieved by means such as deploying an active selection for survival strategy (similar to the (**μ+λ**)-ES), or using an external archive of non-dominated solutions. The latter elitist approach can be implemented in two different ways. An archive can be offline (or inactive) or otherwise online (or active). The first generation of MOEAs (e.g. MOGA) used to deploy offline archives to store all the non-dominated solutions achieved all along the optimisation process. The offline archive however did not have any impact on, or interaction with, the evolutionary search. On the other hand, some of the second generation of MOEAs (e.g. SPEA2) deploy an online archive as an elitist strategy. In addition to storing the best 'representative' solutions, the content of the online archive is used to steer the search by participating in the mating procedures and contributing to the next generations. This last strategy of elitism is generally more efficient then its counterpart (deploying offline archives) and results in an accelerated convergence towards the Pareto front (Deb 2001). Examples of elitist strategies were described in Section 2.3.2. Moreover, it is worth mentioning that the majority of the first generation of MOEAs used to deploy a (**μ, λ**) strategy at the selection for survival stage. All the offspring solutions deterministically replaced their parents regardless of their performances. In other words, the selection for survival process can be described as inactive in the majority of the early EMO approaches and hence elitism was absent.

Deploying elitism in the form of an external archive, the inclusion of candidate solutions in the archive is performed in *en bloc* or *incremental* fashion. The latter two terminologies were suggested by Zitzler (1999) and correspondingly denoted the strategy of simultaneously including a set of solutions into the external archive (e.g. SPEA2) or otherwise the inclusion of one solution at a time (e.g. PAES, PESA). While in the latter approach, the order of including the solutions in the archive is essential, in en bloc strategies the order of inclusions is irrelevant.

The importance of elitism was particularly highlighted after the outcome of several critical studies (Zitzler and Thiele 1999, Zitzler, Deb and Thiele 2000). These studies were conducted in the aim of contrasting the performance of multiple MOEAs on a set of bi-objective optimisation problems. Some of the MOEAs included in the comparative studies deployed elitist strategies, while other MOEAs did not. The studies' outcomes illustrated that the elitist MOEAs were generally outperforming their counterparts with no active elitist strategies. Indeed, study has shown that in order to ensure the convergence of the population of solutions handled by a MOEA in the limit, elitism is an essential and a theoretical requirement (Rudolph and Agapie 2000). The convergence requirement is further investigated in Chapter 3 where enhancements to this primary EMO criterion are devised.

2.3.4. Obtaining Good Diversity

Fitness sharing, a concept first introduced in 1989 by Goldberg (1989), is one of the earliest attempts for promoting diversity as a requirement in EMO. Fitness sharing is based on a density estimation approach and was originally motivated by the need for *niche formation* to prevent premature convergence towards sub-optimal regions of the objective space. As its name reveals, the 'fitness sharing' method forces the sharing and therefore the degradation of the fitness values corresponding to solutions lying within a certain distance from each other. The notion of distance is an application dependent variable, and is usually the Euclidean distance in most real-coded applications. The process of sharing fitness values commonly penalises solutions populating dense areas of the search space without violating the Pareto dominance notion. In other words, fitness sharing discriminates each set of solutions belonging to a certain level of performance -or rank- in terms of diversity without degrading their membership with the rank they occupy. Hence, a non-dominated solution lying in a poorly populated area of the objective –or decision-space will always have the highest probability of selection for variation or survival. The major disadvantage of the fitness sharing method is that its success is highly dependent on the chosen niche size parameter σ_{share} which was previously described in Section 2.2.9.

Several alternatives to the fitness sharing technique for density estimation were proposed in EMO. An overview of diversity promotion strategies in EMO was given in Section 2.3.2. The most widespread alternatives include the *Nearest Neighbour* approach and the *histogram* based techniques. In a framework aiming to improve the performance of SPEA, Zitzler et al (2001) suggested a density estimate based on the use of the k^{th} nearest neighbour measure in the Euclidean objective-space. A statistical heuristic was deployed to determine the value of the critical parameter k based on the square root of the population size. Other density estimation approaches based on the nearest neighbour included the method suggested by Abbass, Sarker and Newton (2001) who replaced the use of the k^{th} nearest neighbour measure as a density estimate with the mean Euclidean distances of the *two* nearest solutions. Sarker, Liang and Newton (2002) later on extended the technique by Abbass *et al* to incorporate the mean Euclidean distances of the M nearest solutions as the density estimate. The two most popular diversity-preserving operators based on the nearest neighbour density estimation are the clustering technique (Zitzler and Thiele (1999)) and the crowding technique (Deb *et al* 2002). These two techniques were previously introduced in Section 2.3.2.

Despite their conceptual simplicity and their relatively low computational requirements, the nearest neighbour diversity promotion mechanisms suffer from a fundamental disadvantage due to their requisite for the consistency and scalability of potentially non-commensurable objectives. Another disadvantage of the nearest neighbour approaches is their requirement for a sensitive choice of the parameter k which is essential to the success of the technique. This disadvantage is vaguely similar to the choice of σ_{share} in the fitness sharing approach for promoting diversity.

On the other hand, histogram based techniques are an alternative density estimation procedure that overcome the disadvantage of requiring distance measurements and the concatenation of non-commensurable objectives in the nearest neighbour schemes. These alternative techniques operate by partitioning the objective space into a grid of different hyperboxes. The density estimation is then based on the count of the number of solutions residing in a certain hyperbox in the objective space. Several objective space gridding systems where introduced in the EMO community. In the context of the Pareto archived evolution strategy (PAES) by Knowles and Corne (1999), the user specifies the number of bisections in terms of each objective range, and therefore specifies the spacing between the objective space hyperboxes. These user-specified spacing ranges between partitions of the objective space are then used in an adaptive grid spacing system defined by the locally non-dominated solutions. Deb *et al* (2003) suggested a steady state approach (ε-MOEA) based on a combination of the ε-dominance concept introduced by Laumanns *et al* (2002a) and an adaptive grid archiving (ADA) similar to PAES. Laumanns *et al* (2002a) used the ε-dominance concept to implement a histogram-based diversity promotion strategy based on the proposition by Papadimitriou and Yannakakis (2000). ε-MOEA maintains diversity in the archive by allowing only one solution in each pre-assigned hyperbox in the objective space. No specific upper limit on the archive size is needed to be predetermined as the archive gets bounded according to the chosen ε vector in terms of each objective.

Despite their increasing popularity in the EMO community, one of the major drawbacks of histogram-based diversity promotion techniques is their exponential computational complexity in the number of objectives. The gridding system applied in the objective space is yet another sensitive design parameter which can be inappropriate to some Pareto front structures. Choosing the right gridding system is particularly a hard design choice when the dimensionality of the objective space increases.

Other miscellaneous approaches for promoting diversity in EMO include *mating restrictions*, *lateral diversity* and *target vector* approaches. Mating restriction was suggested by Deb and Goldberg (1989) after a study on multimodality. They have noticed that when solutions from remote areas of the search space (objective or decision space) recombine, they most often produce week offspring known as *lethals*. Imposing restrictions on the mating process was first introduced by Booker (1982) for promoting diversity in the approximation set (Deb and Goldberg 1989). Similar to fitness sharing's parameter σ_{share}, a parameter σ_{mate} is calculated to determine whether two selected solutions belong to different regions of the space and should be restricted from mating. Lateral diversity on the other hand, suggests maintaining diversity in the dominated regions of the space for obtaining an overall better diversity in the population. This is interpreted as a requirement for keeping a balanced tradeoff between diversity and proximity to the Pareto front (Bosman and Thierens 2003). Examples of studies deploying lateral diversity for maintaining a tradeoff between proximity and diversity include Deb and Goel (2001), Laumanns et al. (2001), Laumanns and Ocenasek (2002b), and Bosman and Thierens (2003).

Moreover, target vector approaches, originally suggested in the OR community and used to assess the performance of EAs, are deployed as a diversity promotion techniques in EMO. The approach consists of suggesting a target point in the search space, and then seeks to minimize the distance between candidate solutions and the suggested target point (Wienke, Lucasius and Kateman 1992). Later on in the EMO community, the target vector approach was extended to allow the suggestion of multiple target objective vectors (Lohn, Kraus and Haith 2002). In the extended approach, better fitness scores get allocated to solutions close to certain target vectors. Nevertheless, fitness sharing takes place when multiple solutions reside in the neighborhood of a target objective vector in the aim of promoting diversity and avoid the genetic drift towards a single target vector.

Chapter 4 will consist of a closer look at the diversity requirement in EMO. New approaches for addressing the requirement for a tradeoff between the two essential requirements (diversity and proximity) will be investigated in a *many* objective optimisation scenario.

2.3.5. Preference articulation techniques

In most scenarios, a decision-making process is equally essential and required as the multiobjective optimisation task itself. In real-world multiobjective optimisation, a single solution is usually sought for implementation among several alternative candidate solutions. Providing the DM with a well-diversified approximation set along the Pareto optimal front is generally half the way towards solving the problem at hand. EAs are optimisation and search techniques with proven suitability for finding optimal approximation sets for multiobjective problems. However, having provided the DM with a well-distributed approximation set, a single solution is usually needed to be chosen. A psychology driven decision making requirement is hence providing the DM with an approximation set of representative solutions and reasonable size for him/her to choose from (Miller 1956). The selected solution is then selected, based on certain preferences and priorities and might undergo certain further analysis and testing (e.g. in terms of robustness through *sensitivity analysis*), before it is implemented. Decision Maker's preferences are very important information and can affect the multiobjective optimisation process and delimit the search process to certain regions of interest if these were incorporated in the optimisation process.

Multi-Criteria Decision Making (MCDM) in EMO is classified in *three* main categories reflecting '*how*' and '*when'* the preferences are articulated (Horn 1997). The three categories are listed below:

1. *A Priori* preference articulation: This category of preference articulation denotes the process of introducing and incorporating the preferences before the search process.
2. *A Posteriori* preference articulation: This category of preference articulation denotes the process of introducing and incorporating the preferences at the end of the search process.
3. *Progressive* preference articulation: This category of preference articulation denotes the process of introducing, incorporating and modifying the preferences in an interactive and progressive way at any time during the search process.

Purshouse (2004) has shown that evolutionary *many*-objective optimisation usually faces the unambiguous conflict of solutions convergence and diversity, also referred to as the curse of dimensionality which is mostly caused by the *dominance resistance* phenomenon. Dominance resistance is a problem first identified by Ikeda *et al* (2001) and denotes the inability of producing offspring solutions that dominate their parent solutions. Deploying preference articulation to delimit the high dimensional search space to certain regions of interests is a remedial measure for overcoming the problem of dominance resistance and diminishing the curse of dimensionality. Nevertheless, compared to other aspects of EMO, the research into preference articulation has been neglected to some extent in EMO. Incorporating preferences in the evolutionary search can provide potential advantages over the use of pure Pareto-optimality, which is unfettered in its search and is liable to produce solutions outside the ROI as well as within it. Coello (1999) produced a comprehensive survey about handling preferences in EMO. Another more recent survey about preference articulation in EMO can be found in Rachmawati, L. and Srinivasan (2006). In the following, an overview of the three categories of preference articulation (a priori, progressive and a posteriori) used in EMO is presented.

A Priori Preference Articulation

In a priori preference articulation techniques, the DM is usually requested to know and articulate his/her preferences prior to the optimisation process. The majority of classical approaches for multiobjective optimisation were built upon the *a priori* preference articulation scheme. This category mostly consisted of aggregating functions which allocate weights to the different objectives considered and transform the optimisation problem into a single objective optimisation problem. Examples of classical multiobjective optimisation techniques that incorporate *a priori* preference articulation to transform the multiobjective problem into a single objective counterpart were presented in Section 2.3.2.

In addition to its common use in multiple start classical optimisation techniques, a priori preference articulation is widely used in MOEAs. For example, in Parmee, Cvetkovic, Watson and Bonham (2000) a modified version of Pareto dominance was suggested for use in a MOEA framework. In the modified version of Pareto dominance, the dominance relation was altered to reflect the multiple weights assigned (a priori) to each objective, in a way favouring the solutions in the decision maker's ROI. In order to facilitate the process of assigning different weight values to the objectives, (Cvetkovic and Parmee 1999) introduced a technique for converting vague and qualitative preferences into quantitative weights using pairwise fuzzy comparisons of the objectives. As the number of objectives increases, the number of pairwise comparisons becomes a tedious task for the DM (Cvetkovic and Parmee 2002). The use of transitive relations was therefore proposed to reduce the number of pairwise comparisons required from the DM.

Another technique -the *Guided MOEA (G-MOEA)*- based on the underlying concept of modifying the Pareto dominance scheme was proposed by Branke *et al* (2001). Instead of assigning weights to the different objectives, pairwise tradeoff values (between each two objectives) were required to define the modified version of Pareto dominance, termed as the *Guided dominance*. These tradeoff values corresponded to *all* the maximally (or otherwise minimally) acceptable units of degradation in terms of each objective ‘i’ that can be deemed compensable by a single unit of improvement in terms of each of the remaining objectives. G-MOEA, its advantages and disadvantages are fully described in Chapter 5 which present a comparative study of some of the most interesting preference articulation techniques in EMO.

Another approach for manifesting *a priori* preference articulations consists of modifying the fitness sharing used in many MOEAs for promoting diversity in the approximation set. This *biased sharing* approach was proposed by Deb (1999a), and suggested assigning weights to the different components (in terms of each objective) of the distance metric used for fitness sharing. The fitness sharing, and hence the diversity promotion process, was consequently biased towards the ROIs to the DM. In order to overcome the shortcoming of the biased sharing approach, presented by its inability of focusing on a certain intermediate region of the Pareto front (Deb, Sundar, N and Chaudhuri 2006), Branke and Deb (2004) introduced a fine-grained approach based on the biased sharing technique. The fine-grained approach, termed as *biased crowding*, provides the DM with facility of focusing on any part of a Pareto optimal front while controlling the intensity of the bias. The biased crowding technique is further explored and described in Chapter 5.

Furthermore, inspired by the goal programming technique (Aouni and Kettani 2001) (previously described in Section 2.3.1), Deb (2001b) suggested the use of a multiobjective formulation of the goal programming approach within a MOEA (NSGA) framework. In order to overcome the classical difficulties of the weighted sum of deviations from unmet goal values, Deb (1999a) formulated the problem as a multiobjective task aiming to minimise the different deviations (each considered as an objective) from the unmet goal values for each objective. The goal values had to be articulated before the start of the optimisation process, and once met the optimiser did not seek to optimise past the pre-determined goal values.

A more recent approach for incorporating the DM preference prior to the optimisation process is also suggested by Deb et al (2006). This time the proposed technique was inspired by the classical approach of using reference points (Wierzbicki 1980). Unlike its original use for single objective optimisation, the technique proposed by Deb et al (2006) suggests the use of multiple reference points in the objective space to steer the search process of a MOEA in their directions.

Finally, the operational research community produced a lot of valuable contributions that are successfully implemented as a priori preference articulation techniques in EMO. Examples of such contributions are the prominent outranking techniques PROMETHEE (Preference Ranking Organisation METHod for Enrichment Evaluations) (Brans, Vincke and Mareschal 1986) and ELECTRE (Benayoun and Sussman 1966). Examples of studies incorporating such OR contributions into MOEAs include the use of ELECTRE (Xunxue Cui and Chuang Lin 2005) and PROMETHEE-II (Rekiek *et al* 2000) for guiding the search based on a priori preferences. It is worth noting that outranking techniques can be equally incorporated in MOEAs for a posteriori decision-making and preference articulation.

The main disadvantage of *a priori* preference articulation is the assumption that the DM knows a priori his/her *static* preferences. In other words, in addition to the difficult requirement of knowing the preferences before the optimisation process, the use of a priori preference articulation does not provide the DM with the flexibility of changing these preferences as new information becomes available. In a lot of real world applications, especially when solid grasp of the application is lacking, the DM might be uncertain about his/her preferences. In such scenarios, a priori preference articulation might be very unsuitable.

A Posteriori Preference Articulation

A posteriori techniques require the DM to articulate his/her preferences and select a single preferred solution, or subset of solutions, from a family of solutions produced by an MOEA. This category of preference articulation is by far the most widely used in EMO. A good optimiser should provide the decision maker with an approximation set presenting well-distributed solutions across the Pareto optimal front. Having exposed a variety of optimal solutions to the DM, the next step would then consist of a subjective process of choosing a certain solution based on certain preferences. Such preferences might be endorsed by a potential better understanding of the application inferred from or highlighted by the search results and any revealed higher-level information.

Three post-optimal techniques for assisting the DM in choosing a certain compromise solution were suggested by Deb (2001). The first technique consists of deploying a *compromise programming* approach (Yu 1973) for selecting the candidate solution whose Euclidean distance towards a certain ideal reference solution in the objective space is minimal. The second approach consists of choosing the solution whose *marginal rate of substitution* (Miettinen 1999) is maximal. The marginal rate of substitution denotes the amount of improvement in terms of a certain objective which can be gained at the expense of a unit of deterioration in terms of another objective. The third approach consists of calculating pseudo-weight vectors for each candidate solution and choosing the solution with the closest weight vector to the DM's preferences.

Alternatively, MCDM techniques borrowed from the OR community, such as PROMOTHEE II or ELECTRE, can be used as a posteriori preference articulation for deciding on a certain particular solution from the approximation set. An example of such a posteriori approaches is the work by Massebeuf, Fonteix, Kiss, Marc, Pla and Zaras (1999) where PROMOTHEE II is used to select a specific solution from an approximation set produced by an MOEA.

Compared to its a priori counterpart, a posteriori preference articulation allows a better understanding of the application at hand as well as its objective space. This should ultimately assist the DM in making an informed and better decision. Nevertheless, a posteriori preference articulation requires the multiobjective optimiser to produce the entire Pareto optimal front. This can be computationally expensive and might introduce problems such as the conflict between the EMO requirements for solutions' convergence and diversity (this is explored in Chapter 4) as well as the difficulty of visualising the objective space as the number of objectives increase. Moreover, providing the DM with a large amount of data to choose from can be very confusing and complicates the a posteriori decision making process.

Progressive Preference Articulation

In progressive preference articulation (PPA) schemes, the DM articulates his/her preferences progressively throughout the optimisation process. Despite their preference in the OR community over their a priori and a posteriori counterparts (Cohon and Marks 1975), progressive preference articulation is the least used approach for articulating preferences in the EMO community. For this purpose, Chapter 5 of this study is entirely dedicated for studying, comparing and analysing some of the most potential PPA techniques with the aim of promoting research in this overlooked direction. Among other benefits, progressive preference articulation is a useful technique for reducing the computational effort required for converging to the whole global tradeoff surface. This is especially beneficial in high-dimensional search spaces when tackling evolutionary many-objective optimisation problems. Moreover, deploying PPA techniques, the DM can modify his/her preferences at any time of the optimisation process and make use of any tradeoff or beneficial information (such as objectives' relationships or goal attainments). Interested readers are referred to (Hwang 1979, Coello (1999), Coello (2000a), Fleming, Purshouse and Lygoe 2005, and Andersson 2000) for a comprehensive literature about MCDM.

The first truly progressive preference articulation was introduced in 1998 in the context of MOGA (Fonseca and Fleming 1998). Earlier, in 1993 a PPA technique suggested by Fonseca and Fleming (1993) allowed the user to interactively articulate goal information for the multiple objectives under optimisation. The goal information was then used to interactively modify the fitness assignment process to reflect the DM's preferences based on the classical goal attainment method (Aouni and Kettani 2001). The PPA technique introduced by Fonseca and Fleming in 1993 was later extended in 1998 to include goal and priority information about the multiple objectives (Fonseca and Fleming 1998). The goal and priority information were fed into a transitive relational operator termed as the preferability operator to modify accordingly the fitness assignment process. More details concerning the preferability operator will be presented in Chapter 5.

Other PPA techniques were reported in the literature and presented a lot of similarities with Fonseca and Fleming's preferability operator. Examples of such PPA techniques include the very similar scheme (know as *favour*) proposed by Drechsler, Drechsler and Becker (2001) and the constrained-domination approach suggested by Deb, Pratap, Agarwal and Meyarivan (2002). The latter approach differed from the preferability operator by seeking to optimise the overall goal violation in terms of the objectives. The constrained-domination approach ensured more information to the optimiser but required the forced cohesion of the different objectives. In the framework of preferability schemes, (Tan, Khor, Lee and Sathikannan 2003) also suggested a method for articulating alternative preference scenarios for the optimisation problem at hand. The technique allowed the articulation of hard and soft preferences such as priorities and constraints using logical ("AND" - "OR") connectives.

Abbass (2004) suggested a generic interactive framework which enables the DM to refine the results achieved by a MOEA. This is achieved by determining bliss points (desired and ideal points) from the actual approximation set produced by a MOEA. The identified bliss points might be infeasible (mapping to infeasible regions of the decision variable space), however the task would consist of enhancing the quality of the actual approximation set by minimising its distance to the infeasible bliss points and converging to the best feasible bliss points.

Despite their utility, progressive preference articulation schemes can be quite demanding in terms of the interactions required from the decision maker. In order to overcome this burden which can require in some scenarios the interaction of the DM at every generation of the optimisation process (Takagi 2001), some methods for simulating and automating the DM interactions were suggested in the literature. Fonseca and Fleming (1993) suggested the use of an automated DM such as expert systems. Later on, Todd and Sen (1999) used an artificial neural network to implement such automated DM by training the NN with exact data (decisions) occasionally collected from the DM at certain intervals of the process to model his/her preferences.

2.4. Summary

In this chapter, a review of multiobjective optimisation methods was introduced. The classical approaches for solving multiobjective problems were described alongside side their limitations. Moreover, the benefits of deploying evolutionary algorithms to solve such problems were illustrated. The discipline of evolutionary computation was presented and the popularity of one of its major branches -evolutionary algorithms- in solving multiobjective optimisation was described. The concept of Pareto dominance and its utility for multiobjective optimisation was presented. The three essential requirements for multiobjective optimisers were described. These are correspondingly the convergence of the approximation set towards the tradeoff surface (proximity), the diversity of the approximation set across the tradeoff surface, and the pertinence of approximation set to the decision maker. Evolutionary algorithms were then comprehensively reviewed, and their features for tackling each of the three requirements were illustrated. In the remainder of this thesis, a closer look into EMO and its three requirements is given and innovative, beneficial and "remedial" approaches for achieving these three requirements are presented.

Chapter 3

Evolutionary Multiobjective Optimisation: Enhancing the Convergence Perspective

3.1. Introduction

As discussed in Chapter 1, real-world problems commonly require the simultaneous consideration of multiple, competing, and generally non-commensurable performance measures. The use of population based optimisation techniques, such as evolutionary algorithms, is a suitable approach for addressing such real-world optimisation problems (see Section 2.3). When solving a MOP, the approximation set produced by a MOEA is required to be well-spread across the objective space and as close as possible to the true Pareto front, presenting the decision maker with a well-distributed set of solutions within the region(s) of interest (ROI). Moreover, to be of practical use for tackling real-world applications, which can be very computationally expensive, a multiobjective optimisation algorithm must produce an approximation set with acceptable proximity and diversity within limited computational resource (most importantly, a fixed and limited budget of objective function evaluations). The time taken by an algorithm to perform a given number of search iterations for a particular problem is dependent upon the available computing power. Nevertheless, the efficiency of a MO optimiser can be determined by the quality of the results achieved within a fixed budget of objective function evaluations. The performance of a stochastic multiobjective optimisation algorithm may then be determined by the proximity and diversity of the approximation sets produced from a given number of iterations and objective function evaluations over multiple runs of the algorithm.

In this Chapter, two essential requirements for multiobjective optimisers are addressed and approached from a new perspective:

- The convergence (or proximity) of the solutions to a multiobjective problem towards the true Pareto front, and
- The speed of convergence of such solutions towards the true Pareto front and hence the efficiency of multiobjective evolutionary algorithms.

As a result, innovative enhancements to the second generation and well established MOEAs are sought. Under-exploited and unaddressed aspects of the evolutionary multiobjective optimisation problems are investigated and implemented, with the aim of introducing a new generation of optimisers which are better suited and equipped for addressing well-known difficulties and issues that are most often encountered in the ever growing field of evolutionary multiobjective optimisation.

In Section 3.2, the conceptual framework and the motivational points for the study presented in this Chapter are introduced. This is aimed at highlighting potentially beneficial areas of research which will be investigated in the following sections. Section 3.2 also serves as a preliminary sketch or conceptual outline for a new generation of MOEAs which addresses the requirements of evolutionary multiobjective optimisers from a new perspective. In Section 3.3 an innovative convergence enhancement strategy is introduced and described in details. The different experimental frameworks adopted for the convergence acceleration strategy are presented alongside carefully chosen performance metrics. The results produced by the introduced strategy for the different optimisation problems investigated in this work are presented and contrasted with the results achieved by some of the most established second generation MOEAs.

3.2. Framework and Motivation

3.2.1. Enhancing Evolutionary Algorithms: Motivations

Traditional evolutionary computation techniques usually consist of an explorative set of procedures operating in decision variable space. These explorative procedures are usually represented by the recombination and mutation operators. Starting from a random population of candidate solutions or from a previously known set of solutions in decision variable space, EAs calculate the corresponding objective function values, assign them fitness scores reflecting their utility in the application domain and bias the search towards high-potential areas of the space by forcing the survival of the fittest solutions (see Section 2.2). In other words, EAs operate in decision space and perform decision space to objective space mapping but fail to exploit direct use of the objective space - this is a lost opportunity. As a result, despite their utility for solving MO problems, the use of EAs often result in a large number of objective function calculations which can be computationally expensive especially when the objective functions themselves are expensive to evaluate.

The stochastic nature of EAs' explorative processes is a major asset and a fundamental reason for the successful reputation of these algorithms. Indeed, EAs' stochastic nature addresses many of the shortcomings that alternative deterministic and more classical optimisation techniques are known to confront. Examples of such challenges and difficulties include the increase of dimensionality, multimodality *–local optima entrapments-* and complexity of the landscape of some optimisation problems. The lack of derivative information and knowledge about the landscape of the solution spaces in some scenarios, two indispensable requirements for the functionality of most deterministic classical techniques, are other examples of frequent difficulties that can be overcome by deploying stochastic EC strategies such as EAs.

However, given the stochastic nature of its operators, an evolutionary algorithm offers no guarantee of finding optimal solutions within a single run or more. Through the variation operators operating in the decision variable space, new solutions are produced with the assumption that "good" parents are more likely to produce "good" offspring and hence should contribute more to the next generations. The addition of some straightforward determinism to such stochastic strategies is sought as a remedial measure which is believed to enhance the performance and the efficiency of EAs. This motivates further investigations and experimentations within a framework hybridizing EAs strength and structure with some innovative deterministic components; a framework commonly known as a *'Hybrid EA'* or *'Memetic Algorithm'* (see Section 2.2.9).

3.2.2. Objectives

The aim of the study presented in this Chapter is to incorporate the direct exploitation of the objective space as an active component of a MOEA. This is designed to assist the stochastic exploration process and guide the search efficiently towards goal values and regions of interests in the objective space in a more or less deterministic way. Ultimately, the hybridization of deterministic components which directly exploit and explore the objective space within the cycle of an EA should introduce several benefits to the optimisation process. The main expected benefits should include the acceleration of the convergence of the handled solutions towards the Pareto front of a MOP and the reduction of the computational effort of these optimisation techniques. More specifically, the reduction in the computational effort is sought in terms of the number of objective function evaluations, which is the main computational burden in most real-world applications. In the following Section, a strategy implementing the targeted conceptual evolutionary multiobjective optimiser described in Section 3.2.1 is introduced.

3.3. Obtaining Better and Faster Proximity

3.3.1. Introduction

In many application domains, calculating the true objective function may be computationally expensive. Given their generational, population-based approach, EAs require a significant number of objective function calculations to be performed. The use of approximated models using Neural Networks (NN), or other metamodelling techniques such as Kriging-based approximations, or response surface models (Farina 2002, El-Beltagy, Nair and Keane 1999) provides low computational burden alternatives to full objective function evaluation (Adra *et al* 2005b, Nariman-Zadeh *et al* 2005). An overview of fitness approximation techniques used in EMO was given in Section 2.2.9. Moreover, a comprehensive survey about fitness approximation is presented in Jin (2005). Fitness inheritance (Sastry, Goldberg and Pelikan 2001, Chen *et al* 2002) is another type of approximation technique which belongs to the class of Evolutionary Approximations. Fitness inheritance, a specific approximation technique which can solely be used in evolutionary algorithms, is an approach where expensive objective function evaluations are reduced by replacing the fitness evaluation (or objective function evaluation) of certain individuals with the interpolation of the fitness values of their parent solutions. Study has highlighted that fitness inheritance should be limited to convex and continuous problems, and is not well suited for addressing complex real-world applications (Ducheyne, Baets and Wulf 2003).

In this Section, a Convergence Acceleration Operator (CAO) is introduced, which maps from objective space to decision variable space (in the reverse direction to a meta-modelling technique). This operator is meant to be a portable component that can be hybridized with any population-based stochastic optimisation algorithm, such as evolutionary algorithms. The portability of the CAO is assessed by hybridizing it with two widely used EAs, the *Non-Dominated Sorting Genetic Algorithm (NSGA-II)* (Deb *et al* 2000) and the *Strength Pareto Evolutionary Algorithm (SPEA2)* (Zitzler, Laumanns and Thiele 2001). The purpose of the CAO is to enhance the performance of the host stochastic global optimisation technique in terms of the proximity of the approximation set for a given number of objective function calculations without impeding the active diversification mechanisms of these search strategies. In contrast to EAs' failure to exploit the direct use of the objective space, the CAO features a direct search in objective space and then a prediction mechanism to map from objective space to decision space. In this work, neural networks are deployed for the prediction process from the objective space to the decision space. Nevertheless, other prediction processes, such as Kriging or response surface models, can be equally used within the CAO and will be investigated in a future work.

The idea of performing local search in the objective space and seeking to map a certain objective vector back to its corresponding decision vector was first introduced in Gaspar-Cunha and Vieira (2004) and Gaspar-Cunha, Vieira and Fonseca (2004). These two papers were then extended in Gaspar-Cunha and Vieira (2005) where the suggested technique was additionally assessed on a real world application problem. In these papers, the authors proposed a method to accelerate the search of a MOEA by approximating the function that maps from the objective space to the decision space using NN techniques. More specifically, Gaspar-Cunha et al's method, which is hybridised with the Reduced Pareto Set Genetic Algorithm (RPSGA) (Gaspar-Cunha and Covas 2002), used a multi-layer perceptron (MLP) approach (Bishop 1995) to map in the reverse direction (i.e. objective vectors as inputs and decision vectors as outputs). The trained MLP is then deployed to predict the approximate vectors of decision variables which should correspond to the objective vectors introduced by a local search around the non-dominated solutions arising from the previous generation. The local search suggested by Gaspar-Cunha *et al* attempts to improve the locally non-dominated solutions by minimising their objective values (normalised in the range [0, 1]) directly in the objective space. Each objective value is minimised by an absolute and fixed step size, h, whose optimal value for the experiments carried in Gaspar-Cunha and Vieira (2004) was found in the range [0.2, 0.3]. Gaspar-Cunha *et al* tested their technique on a set of bi-objective test functions (Zitzler et al 2000) and reported an accelerated convergence on these test functions compared to the standalone RPSGA. Moreover, Gaspar-Cunha *et al* also contrasted the performance of their inverse NN technique with the performance of RPSGA coupled with a standard metamodelling technique (i.e. with a forward NN). The metamodelling technique coupled with RPSGA consisted of introducing a metamodel of the actual objective function used. This was achieved by training a NN with exact data during the first K generations of the optimisation, then replacing (or interchanging) the evaluation of the actual objective function with the trained NN for the remaining generations of the process. The comparison of the inverse NN technique and the metamodelling technique in Gaspar-Cunha and Vieira (2004) is an interesting aspect of their work which highlighted a better performance for the inverse NN approach.

Adra *et al* (2005b) extended Gaspar-Cunha *et al*'s approach and applied it to an 8 objective problem of aircraft control system design. The objective was to investigate the utility of deploying direct local search in the objective space and inverse neural network predictions on a *many* objective optimisation problem. In their experiments, Adra *et al* (2005b) used the *Multiobjective Genetic Algorithm (MOGA)* (Fonseca and Fleming 1993) which integrated Fonseca and Fleming's (1998) *preferability* operator for incorporating DM's preferences for search space reduction. A comparative study of some of the most reputed preference articulation techniques will be presented in Chapter 5.

The local search suggested in Adra *et al* (2005b) differs from the local search used in Gaspar-Cunha and Vieira (2004) with respect to the step size of improvement *h*. Adra *et al* (2005b) observed that when using a fixed and absolute value for the step size, the effectiveness of the local search, and hence the effectiveness of the convergence acceleration strategy, decreases quickly as the search starts to converge. It is to be expected that, as the search progresses, the probability of introducing infeasible solutions residing in the infeasible regions of the objective space becomes higher when using a fixed step size of improvement. As a result, the practical usage and the efficiency of the convergence accelerator become restricted to the early iterations of a MOEA. Moreover, the use of an absolute step size for the local search requires the normalisation of most likely non-commensurable objectives and limits the portability and the applicability of the convergence acceleration technique to other applications.

In Adra *et al* (2005b), the step size, h, is an adaptive parameter which varies throughout the iterations of the host MOEA and whose value is uniquely defined for each objective value chosen to undertake the local search. Moreover, Adra *et al* (2005b) promoted the online detection (i.e. as the NN is trained) of the different relationships that the objectives might exhibit (harmony, conflict or independence) and the consideration of any such relationships when performing the local search in the objective space. The technique was also promoted as a potential progressive preference articulation technique which allows the direct manipulation of the objective vectors in an informed way reflecting the DM's preferences and priorities (Adra *et al* 2005b). This work of Adra *et al* (2005b) has been further enhanced and is described below.

3.3.2. Overview of the Convergence Acceleration Operator

The CAO is a 2–step process, which is illustrated in Figure 3.1 When the CAO is launched, it starts by deterministically improving the best solutions achieved; these solutions are the elite solutions stored in the online archive of the host algorithm.

This improvement takes place in objective space and produces an enhanced version of the archive. The CAO then uses a trained neural network mapping procedure to predict the corresponding decision vectors for the enhancements to the archive. A check of these new decision vectors is made, aimed at reflecting any out-of-bounds decision variables arising from the mapping back into their allowed domain. A correction step is then applied, whereby the true objective values corresponding to all of these new decision vectors are calculated. This is performed in order to maintain the exact fidelity of the optimisation problem and the produced solutions, since, as stated by Michalewicz and Fogel (2000), a solution is only a solution with respect to the model used. The correction step of the CAO is another enhancement to the technique suggested in Gaspar-Cunha and Vieira (2004) and Gaspar-Cunha at al (2004).

The approach suggested in Gaspar-Cunha and Vieira (2004) and Gaspar-Cunha et al (2004) did not attempt to check the feasibility of the introduced solutions or rectify any predictions inaccuracy that might be introduced by the NN.

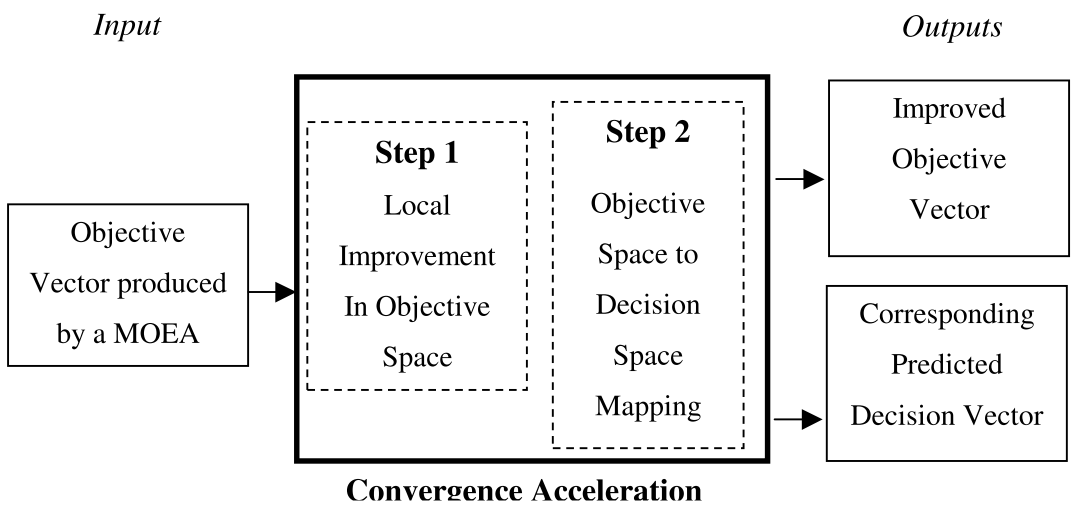


Figure 3.1: The Convergence Acceleration Operator in Context

After the correction step, the enhanced and the original archive of solutions compete to populate the new archive for the next generation, which will represent the pool from which solutions are selected and recombined. The two components of the CAO are described in detail in the following sections.

3.3.3. Local Improvement in Objective Space

The CAO takes place after the recombination and mutation processes and operates on the elitist solutions which would normally propagate to the following generation (or get presented to the DM). The CAO is an auxiliary local improvement operator which does not replace the variation operators in EAs. The first CAO step is a deterministic local improvement procedure in the objective space. This is the component responsible for speeding up convergence and hence reducing computational effort by producing better results within a fixed budget of objective function evaluations. It achieves this by steering objective values obtained by the MOEA towards an improved Pareto front. The objective space local improvement process is implemented in this work for any number, n, of objectives, and is illustrated in Figure 3.2 on a bi-objective problem (n=2) for simplicity. In general, non-boundary solutions in terms of any specific objective ***(solutions B, C and D in Figure 3.2)*** will be improved in terms of all the performance measures by steering their objective values into a region of improved objective function values. The new "improved" values for the objectives (of each non-boundary solution) are determined by linearly interpolating a new value for each objective, between its current value and the next best value(s) achieved for that objective within the population[7]. This is described by:

$$Z_{D'} = Z(x_D + h(x_D - x_C), y_D + h(y_D - y_E)), \quad \textbf{(3.1)}$$

where Z(x,y) represents a point in the bi-objective space, $Z_{D'}$ is the "improved" objective value and h is the interpolation step factor. This process is annotated for ***solution D*** in Figure 3.2. Compared to ***solution D***, ***solution C*** has the next best value in terms of objective 1 while ***solution E*** possesses the next best value in terms of objective 2. Boundary solutions in terms of a certain objective or axis of performance ***(solutions A and E in Figure 3.2)*** are improved in terms of the remaining objectives.

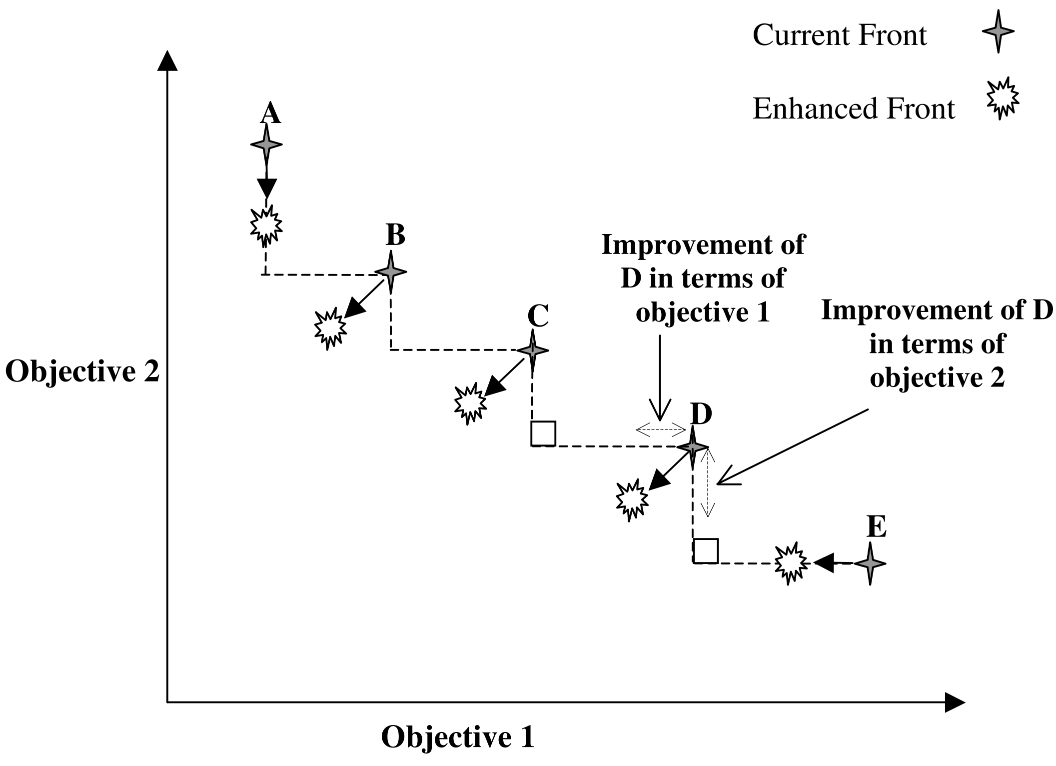


Figure 3. 2: Deterministic improvement of the trade-off surface in objective space

[7] The non-dominated solutions are improved in an objective wise-order, each time sorting the archive of locally non-dominated solutions in terms of a certain objective.

In other words, solution *A* will be improved in the *"Objective 2"* axis direction, thereby enhancing its overall quality by improving it in terms of objective 2, and solution *D* will be improved in the *"Objective 1"* axis direction, consequently improving its overall worth by enhancing it in terms of objective 1. The step size, *h*, is an application dependent parameter and should be carefully chosen; ideally it should depend on the stage of the optimisation, the decision maker's preferences, the regions of interests and the proximity of the population to the expected Pareto front. A larger *h* value is recommended for early generations of the optimisation, with its value gradually decreasing. Moreover, since the decision vectors of the improved front of solutions are to be predicted by the NN (a process described in the following section), it is essential that the new introduced solutions in the objective space reside within the neural network's reliable zone of prediction. Neural networks are very practical tools for regression problems and data fitting, but in common with other curve fitting and data modelling techniques, they are known to be unsuitable for extrapolation tasks. As a result, the step factor *h* should be chosen in a way that maximises the *local* improvement step in the objective space, while preventing the introduction of solutions which reside outside the neural network local region of training. In this study, the value of *h* is manipulated based on the rate of success of the CAO. The rate of success is measured by counting the number of "effective" solutions which are introduced by the CAO, corrected at the correction step, and successfully chosen to propagate to the following generation. The CAO rate of success decreases in one of two cases, which are linked to the large size of *h*:

1- The CAO introduces solutions beyond the Pareto front or the feasible region, or
2- The CAO introduces solutions outside the NN reliable region of prediction, and thus the NN is making extrapolation predictions.

When the CAO rate of success decreases, the step factor *h* is decreased simultaneously (by a factor of 1.5). For the optimisation problems used in this study, the manipulation of the step factor *h* based on the CAO rate of success was deemed sufficient. However, an additional technique for detecting NN extrapolation was investigated and implemented as an optional function whose execution can be automated or controlled by a DM. The technique is based on the approach described in Nabney (2001) (p. 110-113) whereby the prevention of NN extrapolation is performed by deploying a novelty detection step which uses *Gaussian Mixture Models (GMM)*. GMM are very flexible semi-parametric estimation methods widely used for density estimation, clustering and classification problems. GMM are composed of a number of Gaussian function components. The Gaussian functions within a GMM are linearly combined and used to express a certain probability density function (pdf). The number of components that compose a GMM can be freely manipulated, which makes these methods very flexible. Given the right number of components, GMM can approximate any density function within a certain precision. In this study, GMM are efficiently[8] used to model the unconditional probability density function (pdf) of each new introduced solution in the objective space and contrast its likelihood with the range of pdfs corresponding to the neural network's training data (Nabney 2001 and Bishop 1995). If the pdf of a certain new solution is very different from, and does not overlap with any of the pdf values of the training data, the step size is reduced (by a factor of 1.5) as an attempt to avoid extrapolation. Kernel density estimation (also known as Parzen windows) can alternatively be used for detecting novelty and extrapolation (Leonard *et al* 1992), and hence manipulating the step factor *h*.

3.3.4. Objective Space to Decision Space mapping

The description of the mapping method in Gaspar-Cunha and Vieira (2005) is very brief and differs from the local mapping approach described in this section. The second component of the CAO consists of a neural network trained to map the new solutions thus generated in objective space by the first phase of the convergence accelerator back to the corresponding decision variable vectors. Neural networks (NN) are a powerful approach for modelling stochastic and noisy patterns of data in order to produce predicted values of unknown systems. The NN needs to be trained to achieve desirable predictions and to model complex functions as closely as possible. The process of training the NN consists of providing it with samples of input-output data and manipulating weighting variables by adjusting their values and minimizing prediction errors. For more information about neural networks, the interested reader is directed to Bishop (1995).

The second component of the CAO only aims to build a *local model* of the function which maps from the objective space to the decision space at a certain iteration of the optimisation process. This is achieved by training a NN, using exact objective vectors as inputs and their corresponding decision variable vectors as outputs. The training data is the exact data resulting from the objective function values derived within a single iteration of a MOEA such as NSGA-II or SPEA2. More specifically, at every iteration of the optimisation process (or alternatively, when the CAO is called if the CAO is interruptedly executed), a new local model is built based on the locally non-dominated solutions (objective vectors and corresponding decision vectors). The local model is then solely used within the same iteration to predict the decision variables of the new objective values introduced by the first component of the CAO which locally steers the local Pareto front towards an enhanced Pareto front.

---

[8] Since the goal was to get an approximate model of the data density rather than fitting an exact model to the data, only 5 iterations of the algorithm optimising the GMM parameters were executed.

When training a NN, it is vital to ensure well-spread and problem defining data. Abundance of data is an essential point for achieving well-trained NN and high-fidelity models but can be a problem in some computationally expensive applications.

Nevertheless, the CAO is designed for accelerating population-based optimisation strategies such as evolutionary algorithms, where data abundance is usually an essential requirement for the success of such techniques. If an application is very computationally expensive, the requirement for data abundance can be addressed by using a cheaper and acceptable meta-model which approximates the objective function. Alternatively, since the prediction process within the CAO is promoted as a concept rather than an exact method, other substitutes to neural networks, which overcome the requirement of data abundance, can be deployed as the core of the prediction process within the CAO. In the context of this study, the investigation will confine to the use of neural networks within the framework of the CAO due to their flexibility and good reputation for universal approximation given a sufficient number of hidden units and a suitable choice of parameters (Hornik et al 1989 and Nabney 2001).

In this work, a specific type of neural networks, the *radial basis function* (RBF) (Nabney 2001 and Bishop 1995) (Figure 3.3), is used to build the local models of the local Pareto fronts and for predicting the decision variables of the solutions introduced by the local search (Section 3.3.3).

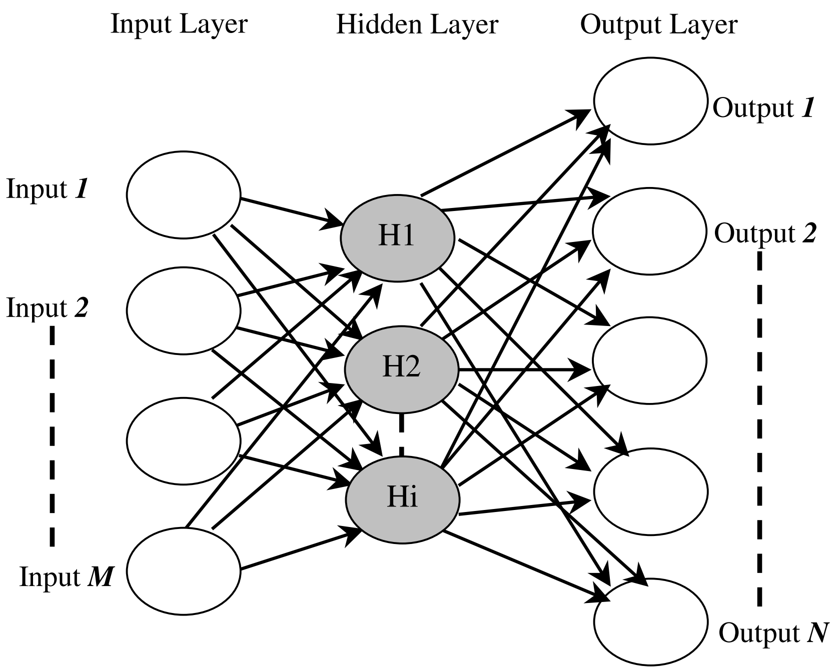


Figure 3. 3: Radial Basis Function Neural Network

In Gaspar-Cunha and Vieira (2004) and Gaspar-Cunha *et al* (2004), the authors used a Multilayer perceptron (MLP) to learn the function that maps from the objective space to the decision space. MLPs are feedforward neural networks generally trained with the standard backpropagation algorithm (Bishop 1995). MLPs are widely used in the field of pattern classification and recognition. The use of MLPs within an acceleration operator such as the CAO is a component that works against the purpose of an accelerator. This is due to the fact that, training a MLP is an iterative process which can be time-consuming. In the neural network literature, the back-propagation algorithm is one of the most studied and used algorithm for training MLPs (Bishop 1995 and Rojas 1996). When MLPs are trained using the back-propagation learning algorithm (Bishop 1995), the output results of the MLP and the exact results are compared and an error value is calculated and fed back through the network. The parameters (weights) of the MLP hidden units are then adjusted and optimised using a non-linear optimiser, usually the gradient descent algorithm. Two major drawbacks of MLPs when trained with the standard[9] back-propagation algorithm and used within a convergence accelerator. These drawbacks are the slow convergence and the susceptibility of getting stuck at local minima in terms of the error functions (and hence sub-optimal weights for the MLP units). It should be noted however that nowadays many alternatives and modifications to the back-propagation algorithm and the gradient descent optimiser are commonly used (Bishop 1995). The conjugate gradient descent, the Levenberg-Marquardt algorithm, Quasi-Newton methods and Delta-bar-Delta (Patterson 1996) are examples of such alternatives and usually perform significantly better than the back-propagation algorithm. RBF neural networks, on the other hand, are two-layered NNs and well known to be practical alternatives to MLPs due to their much faster, two-stage, training process (Bishop 1995 and Nabney 2001). In RBF neural networks, the activation functions of the hidden layer consist of radial basis functions, most commonly a Gaussian, which replace the non-linear activation functions (sigmoidal) used in MLPs. Unlike MLP training process whereby the activation of the hidden units consists of non-linear computation of the scalar product of the input vectors and the weight vectors of the hidden neurons, the hidden neurons of a RBF network are activated by calculating a non-linear function of the distance between the input and the RBF centres. The RBF network mapping to the output layer is described in Equation 3.2, where x is an input data, k is the number of output units, M is the number of radial basis functions Ø and w_{kj} are the output layer weights.

$$y_k(x) = \sum_{j=0}^{M} w_{kj} \phi_j(x) \quad \textbf{(3.2)}$$

The input data is passed through the input layer then processed by the radial basis functions of the hidden layer. The outputs of the hidden layer units are then linearly

[9] Using the gradient descent optimiser

combined and processed at the output layer of the NN. The linear mapping of the hidden layer's values into the output layer of an RBF network is an advantageous feature compared to MLPs. This advantage is due to the RBF training process which consists of adjustments to the linear mapping from the hidden layer to the output layer. As a result, the manipulation of a linear error function in terms of the RBF weights makes it straightforward to efficiently use linear algebra techniques to find the global optima in the parameter space of the error function and hence the optimal values for the RBF weights. Hence, RBF neural networks do not suffer from the problem of getting stuck at local minima in the parameter space because of their quadratic error function whose global minima can be easily found. In other words, in contrast to the MLP the RBF does not have the problem of getting stuck at local minima in the parameter space (i.e., NN weights). The parameter estimates are guaranteed to correspond to the global minimum for a given RBF structure. However, it should be noted that similar to the MLP, the choice of a RBF structure (number of units, the position and widths of the basis functions) remains a nonlinear optimisation problem[10]. As a result, using RBF NN makes the training process a bit easier in that the parameter estimates will be optimal. The considerably faster learning process that RBF neural networks enjoy compared to MLPs make them a suitable choice for deployment within the CAO. This would make it possible to initialise and train a different RBF to model a local model at every iteration the CAO is executed. The ability to map objective vectors to decision variables will make it possible to search directly in objective space for desired combinations of objective values or to devise points of attractions to guide the search.

Two possible approaches to training the NN component of the CAO hybridized with a MOEA are proposed: online and offline training modes. In this study, the online training mode is further elaborated and investigated. However, the interested reader is directed to Adra *et al* (2007c) where the offline training mode is explored and investigated.

Neural Network Training Modes

- *Online Training Mode:*

The online mode consists of concurrently training and validating the NN during the execution of the MOEA. In the online mode, and at every generation of a MOEA, the training data is sequentially collected and instantly used for training a NN and hence building a *local model* of the mapping function from the objective space to the decision space. Many strategies for controlling the use of the CAO in this mode might be devised.

[10] In this study, an unsupervised training process (i.e. without performing a non-linear optimisation) is used to set the number of RBF units, widths and centres (this is described in the next section)

In this mode, the CAO is a performance accelerator that can be launched upon the request of the decision maker (DM) during the execution of the optimisation process.

- *Offline Training Mode:*

An alternative use of the NN is to train it with a collection of data resulting from previous evaluations of the objective functions arising from complete executions of a MOEA. The NN is then incorporated in subsequent runs of a MOEA when used in conjunction with the CAO. In this training mode, the NN is trained on a richer data set and is used for building a *global model* of the mapping function from the objective space to the decision space. However, in the offline training mode, training the NN with conflicting data (caused by one-to-many mapping from the objective space to the decision space) can frequently occur and decrease the NN prediction reliability. The offline training mode also restricts the usage of the CAO to specific applications and optimisation scenarios where the re-execution of a MOEA is necessary. In such applications, the CAO, with its previously trained NN, is subsequently hybridized with any optimiser attempting to solve the same problem. Thus, the CAO will benefit successive executions of the same or other optimisers solving the same problem by speeding up the search and has the potential to offer other benefit.

3.3.5. CAO Summary

Figure 3.4 illustrates the actions of the hybridised MOEA which includes the CAO. Trajectories **2-5** describe the specific actions of the CAO.

- ***Trajectory 1:*** the mapping between a decision variables vector realised by a MOEA and its corresponding computed objective values vector.
- ***Trajectory 2:*** the resulting objective vector – a member of the approximation set at generation **n** - is improved in the objective space.
- ***Trajectory 3:*** a prediction of the decision variables vector corresponding to the improved objective vector is made using the neural network trained with the exact data resulting from earlier evaluations of objective functions –at the same generation **n**- during the MOEA search.
- ***Trajectory 4:*** any invalid decision variable vector introduced by the NN mapping is rectified by reflecting out-of-bounds values of the produced decision variables to their nearest values in their domain of definition.
- ***Trajectory 5:*** finally, the exact objective values vector for the proposed decision variables vector is calculated in the normal way. These candidate solutions will then

compete for archive update and insertion with the best solutions currently stored in the online archive.

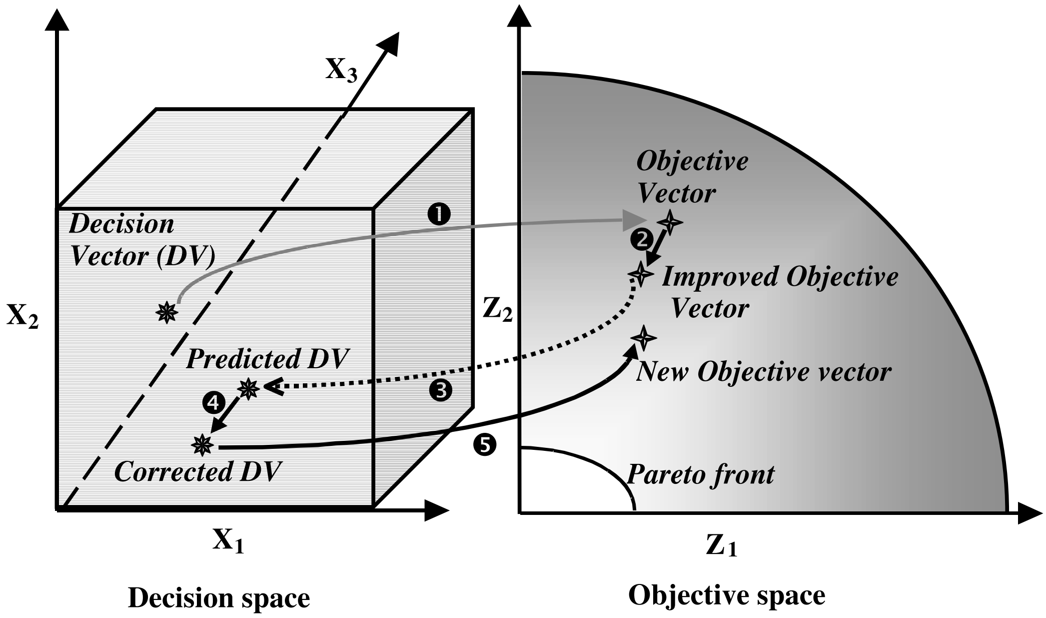


Figure 3.4: CAO steps used in generating a single candidate solution

3.3.6. Experimental Framework

In this section the different experiments investigated for testing the CAO are introduced. The experiments are divided into 2 different categories reflecting the dimensionality of the optimisation problem in the objective space. The two categories are: the *bi*-objective optimisation problems and the *many*-objective optimisation problems. Note, that without any loss of generality, all the optimisation problems used in this study consist of minimization problems. For each of the two categories considered, a description of the optimisation problems deployed will be given. Suitable and reliable performance metrics are also presented for each optimisation problem. Algorithmic illustrations of the CAO-coupled MOEAs used in this study are then described alongside the parametrical configurations of the experiments. The different MOEAs investigated in this work are benchmarked in a way which is similar to the approach used in Bosman and Thierens (2003). In other words, the number of objective function evaluations is fixed beforehand and the best performances over multiple runs of the MOEAs are determined and compared. As a result, a MOEA "A" is deemed more competent than another MOEA "B" if its average performance over multiple runs is superior to the performance of B. This approach of benchmarking MOEAs is more efficient than the benchmarking approach where resources are determined for achieving the optimal results, known *a priori*, for a certain optimisation problem. Bosman and Thierens (2003) state that this way of benchmarking "represents a more practical situation, since we usually do not assume that an unlimited number of function evaluations is available".

Bi-Objective Optimisation Scenarios

In this study a well-established set of optimisation problems is first investigated and used to test the performance of the introduced convergence accelerator. These optimisation problems represent a subset of test functions that belong to a test suite of bi-objective problems presented in Zitzler, Deb and Thiele (2000), and which will be referred to as the ZDT test functions.

The ZDT suite is comprised of six equations, each one of them presenting a specific characteristic and feature that generally cause difficulties to major evolutionary optimisation strategies.

Minimize **F (x) =** $(f_1(x_1), f_2(x))$ Subject to $f_2(x) = g(x_2,\ldots, x_m)\, h(f_1(x_1), g(x_2,\ldots, x_m))$ Where **x =** $(x_1,\ldots, x_m)$			(3.2)
Test function 1: Convex Pareto front formed with g(x) =1, m = 30 and $x_i \in [0,1]$			(3.3)
$f_1(x_1)$:	$g(x_2,\ldots, x_m)$:	$h(f_1, g)$:	
x_1	$1 + 9\sum_{i=2}^{m} x_i /(m-1)$	$1-\sqrt{f_1/g}$	
Test function 3: Discrete Pareto front formed with g(x) =1, m = 30 and $x_i \in [0,1]$			(3.4)
$f_1(x_1)$:	$g(x_2,\ldots, x_m)$:	$h(f_1, g)$:	
x_1	$1+9\sum_{i=2}^{m} x_i /(m-1)$	$1-\sqrt{f_1/g}-(f_1/g)^2\sin(10\pi f_1)$	
Test function 6: Non uniform distribution across a non convex Pareto front formed with g(x) =1, m = 10 and $x_i \in [0,1]$			(3.5)
$f_1(x_1)$:	$g(x_2,\ldots, x_m)$:	$h(f_1, g)$:	
$1-e^{-4x_1}\sin^6(6\pi x_1)$	$1+9((\sum_{i=2}^{m} x_i)/9)^{\frac{1}{4}}$	$1-(f_1/g)$	

Deb (1999b) has recognized numerous features that are most likely to cause difficulties and present challenges to multiobjective EAs in term of their ability to converge towards the Pareto-optimal front while conserving a well-spread distribution of solutions. Isolated optima, deception and multimodality are well known challenges tackled by evolutionary algorithms when converging towards Pareto optimal fronts. On the other hand, features like convexity, non-convexity, discreteness and non-uniformity are other well known difficulties that prevent evolutionary algorithms from achieving a good distribution of solutions and exploring “difficult to reach” regions of the search space. The ZDT test suite encapsulates these difficulties. Each of the ZDT problems consists of a minimisation problem of two competing objectives and comprise 3 distinct functions $\mathbf{f_1}$, $\mathbf{g}$ and $\mathbf{h}$ presented in Equation 3.2, where $\mathbf{f_1}$ is a function of the first decision variable $\mathbf{x_1}$ and $\mathbf{g}$ is a function of the remaining **m-1** decision variables. The bi-objective test functions used to examine the effect of the introduced CAO are three of the most challenging problems, the ZDT1 (convex test function), the ZDT3 (discontinuous test function) and the ZDT6 (non-uniform test function) presented in Equations 3.3, 3.4 and 3.5.

Many Objective Optimisation Scenarios

Test Functions

Studies have shown that conclusions drawn from bi-objective optimisation frameworks cannot be generalized to the *many*-objective optimisation frameworks with more than 3 competing objectives (Purshouse 2004). In order to rigorously investigate the effect of the convergence acceleration operator introduced, deploying optimisation scenarios with more than 2 objectives and various objective relationships were investigated. Hence, 3-objective, 5-objective and 8-objective versions of DTLZ2, a real-parameter scalable test function introduced in Deb *et al* (2002) to test the effectiveness of MOEAs in dealing with increasing number of objectives, were used. DTLZ2 is defined in Equation 3.6.

In Equation 3.6, M presents the number of objectives, $n = M + K - 1$ is the number of decision variables, and K is a “difficulty parameter” ($K = 10$ in this study). DTLZ2 (M) denotes an M-objective instance of DTLZ2. DTLZ2 possesses a continuous and non-convex global Pareto front and comprises two types of decision variables responsible for controlling the solutions convergence towards the global Pareto front and the solutions distribution in the objective space respectively. The first m-1 decision variables ($x_1,\ldots x_{m-1}$) control the proximity of the solutions to the true Pareto front via a k-dimensional quadratic bowl, g, with global minimum $x_{m,\ldots,n} = 0.5$. The decision variables ($x_m,\ldots x_n$) are responsible for controlling the diversity of the solutions and their location on the positive quadrant of the unit sphere.

The scalable DTLZ2 test function belongs to the DTLZ test suite which covers many problem characteristics, such as discontinuity and multimodality. DTLZ2 has been previously demonstrated in Purshouse (2003b) as a challenge for MOEAs especially as the number of objectives increases. Since the CAO does not require any assumptions about the nature of the optimisation problem, the DTLZ2 was deemed sufficient to test the performance of the CAO on optimisation problems with an increasing number of objectives.

$$
\begin{array}{ll}
\min. & z_1(x) = [1+g(x_M)]\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{M-2}\pi/2)\cos(x_{M-1}\pi/2), \\
\min. & z_2(x) = [1+g(x_M)]\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\cos(x_{M-2}\pi/2)\sin(x_{M-1}\pi/2), \\
\min. & z_3(x) = [1+g(x_M)]\cos(x_1\pi/2)\cos(x_2\pi/2)\ldots\sin(x_{M-2}\pi/2), \\
\vdots & \vdots \qquad \vdots \\
\min. & z_{M-1}(x) = [1+g(x_M)]\cos(x_1\pi/2)\sin(x_2\pi/2), \\
\min. & z_M(x) = [1+g(x_M)]\sin(x_1\pi/2), \\
w.r.t & x = [x_1,\ldots\ldots,x_n], \\
where & g(x_M) = \sum x_i \in x_M (x_i - 0.5)^2, with \quad x_M = [x_M,\ldots\ldots,x_n], \\
and & 0 \le x_i \le 1, \quad for\ i = 1,2,\ldots\ldots,n, with\ n = M + \kappa - 1
\end{array} \tag{3.6}
$$

Real-world application: Aircraft Control System Design

A real world engineering optimisation problem was used to investigate the utility of the CAO. The problem consisted of an 8-objective optimisation problem of the classical control system design of an aircraft. The 8-objective optimisation problem encapsulated different pair-wise objective relationships and was previously deployed in a recent study to demonstrate the utility of progressive preference articulation techniques (Fleming, Purshouse and Lygoe 2005). The importance, standardization, and abundant number of criteria to be optimised included in this classical problem were the main reasons for choosing this real world application as a benchmark for testing the proposed CAO-hybrid MOEAs. In this section a simplified illustration of an aircraft dynamical model is shown and a common understanding of the multiobjective optimisation problem is illustrated.

Figure 3.5 illustrates the aircraft body in 3D Cartesian space. The motion of an aircraft in the air is described in terms of 3 main axes: the longitudinal, or *Roll*, axis, the lateral, or *Pitch*, axis and the vertical, or *Yaw*, axis. During its motion, the aircraft makes a combination of changes in both angles and rates of angular velocities; therefore its dynamical model can be represented by an equation combining the main objectives involved in the motion of the aircraft. This equation is highly non-linear across the operating envelope of the aircraft, but it can be linearized for a small deviation around the equilibrium trajectory.

A simplified dynamical model of an aircraft motion can be represented by a fourth order linear equation (Tabak, Schy, Giesy and Johnson 1979). This is presented in Appendix A.

In the context of this study, 8 essential objectives constituted the basis of the optimisation process. The actual dynamical model represents additional characteristics, but a simplified model describing the major issues in controlling aircraft stability is adopted in this work.

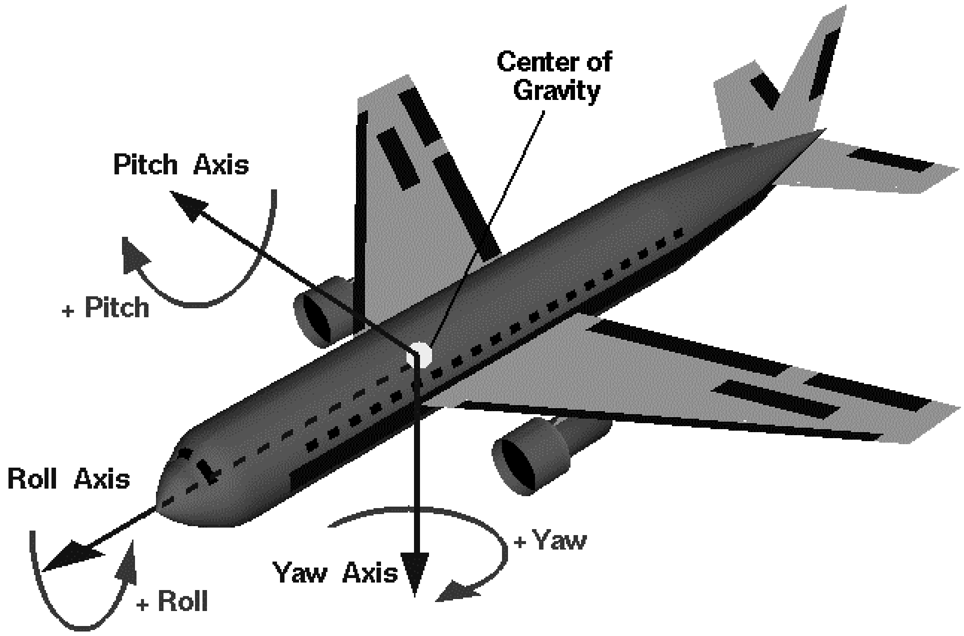


Figure 3.5: Three main axes of the body of an Aircraft (*source: www.grc.nasa.gov*)

The 8 objectives to be minimised consisted of the following:

1. The control effort (sum of squares of gain vector).
2. The spiral root λ_8
3. The damping in the roll root λ_R
4. The dutch-roll damping ζ_d
5. The dutch-roll frequency ω_d.
6. The bank angle at 1.0 seconds ($\phi(1s)$)
7. The bank angle at 2.8 seconds ($\phi(2.8s)$)
8. The sideslip deviation (β).

The characteristic roots of the matrix *(A+BK)* presented in Appendix A

The required bank angle for the fighter aircraft (*According to the military Specifications 1969*)

For further details about the aircraft dynamic model and the variables mentioned the interested reader is referred to Blakelock (1995) and Etkin (1972).

CAO-hybrid MOEAs

The elitist Non-dominated Sorting Genetic Algorithm *NSGA-II* (Deb *et al* 2000) and the Strength Pareto Evolutionary Algorithm *SPEA2* (Zitzler, Laumanns and Thiele 2001), two well-established MOEAs and highly cited second generation optimisers in the EMO community, were chosen as the comparison benchmark optimisers for the problems used in this study. Each was also hybridised - NSGA-II/CAO, SPEA2/CAO - with the introduction of the CAO into their cycles to test its effect. Pseudocode descriptions of SPEA2 and NSGA-II are given in Figures 3.6 and 3.7 respectively. In Figures 3.8 and 3.9, the hybridisation interface of the CAO into SPEA2 and NSGA-II is illustrated within the pseudocode descriptions of the hybrid versions of the two optimisers: SPEA2/CAO and NSGA-II/CAO.

Optimiser configurations used in the experiments involving these 4 optimisers –NSGA-II, SPEA2, NSGA-II/CAO and SPEA2/CAO are given in Table 3.1. In this study, the CAO was operating continuously from the 1st to the 50th (last) generation. At every generation, an RBF neural network is trained then used within the CAO for local improvement and predictions. Training the RBF NN with local and limited data and using it as a local model at a specific generation can help overcome the problem of training the NN with conflicting data resulting from possible one-to-many mappings from objective space to the decision space.

```
-Initialize Population
-Generate initial random population P0 – size Nind and an initial Archive A0
-Evaluate objective values
-Calculate fitness values of individuals in P0 and make A0 = P0
For i=1 to Gen
    -Copy all non-dominated individuals in Pi-1 and Ai-1 to Ai.
    If size of Ai > Nind then reduce Ai (clustering and truncation operators)
    Else fill Ai with dominated individuals in Pi-1 and Ai-1
    Generate new population Pi from Ai – size Nind
                -Binary tournament selection
                -Recombination
                -Mutation
                -Evaluate objective values for the offspring population Pi
                -Calculate fitness values of individuals in Pi and Ai based on the
                -objective vectors of Pi and Ai combined
End loop
```

Figure 3. 6: SPEA2 Pseudocode

As a result, and due to their faster training processes compared to MLPs[11], using RBF NNs is more beneficial for deployment within a convergence accelerator for MOEAs. On the other hand, training a MLP with the standard back-propagation algorithm, requires much more time and effort and makes it impractical to initialize a new NN within the CAO, train it and use it as a local model at each generation of a MOEA. A comparative study assessing the performance of RBF NN with other types of neural networks (e.g. MLP with

[11] When trained with the standard back-propagation algorithm

fast training algorithms) when coupled with the CAO is an interesting task that will be investigated in a future work. Because of the CAO correction step, the number of objective function evaluations per generation in NSGA-II/CAO and SPEA2/CAO is twice the number of objective function evaluations per generation in SPEA2 and NSGA-II.

-Initialize Population and Generate random population **P**– size **Nind**

-Evaluate objective values

For i=1 to Gen

-Assign rank based on Pareto Dominance using non-dominated sorting strategy

-Determine crowding distance between points on each front of solutions (same rank)

-Generate offspring population **Q** – size **Nind**

-Binary tournament selection
-Recombination
-Mutation
-Evaluate objective values for the offspring population **Q**

-Combine parent population **P** and offspring Population **Q** – size: **2*Nind**

-Assign rank based on Pareto Dominance using non-dominated sorting strategy

-Determine crowding distance between points on each front of solutions (same rank)

-Select **Nind** solutions to propagate to the next generation *(1st: elitist -biased towards lower ranks- 2nd: crowding distance - bias less crowded solutions)*

End loop

Figure 3. 7: NSGA-II Pseudocode

In order to compare the algorithms for the same prefixed number of objective function evaluations (10000 evaluations), the CAO-coupled MOEAs were executed for 50 generations per execution, while NSGA-II and SPEA2 were executed for 100 generations. The larger level of exploration and global search afforded NSGA-II and SPEA2 can be seen as an advantage in their favour but constitutes a considerable challenge for assessing the utility of the CAO. The configuration of the optimisers presented in Table 3.1 is a standard configuration usually used in the EMO community when using NSGA-II, SPEA2 or other MOEAs for optimising most of the problems previously presented. The major difference was the number of generations used in this study for the CAO-coupled MOEAs, which was relatively smaller than the standard number of generations *–around 150-* usually used in comparative studies, such as the pioneering study by Zitzler, Deb and Thiele (2000). This choice of configuration was intended to study the effect of the convergence acceleration and any benefits that might be introduced by the CAO. Concatenation of real number decision variables was the convenient choice for encoding the problems under investigation. Due to the stochastic nature of the evolutionary strategies, a well-based judgment concerning the performance of a specific algorithm cannot be stated unless the whole optimisation process is repeated a number of times. In the case of this work, each algorithm was subjected to 10 executions, each running for 100 generations (for SPEA2 and NSGA-II) and 50 generations (for the CAO-coupled MOEAs).

```
-Initialize Population
-Generate initial random population P_0 of size Nind and an initial Archive A_0
-Evaluate objective values
-Calculate fitness values of individuals in P_0 and make A_0 = P_0
For i=1 to Gen
    -Copy all non-dominated individuals in P_i-1 and A_i-1 to A_i.
    If size of A_i > Nind then reduce A_i (clustering and truncation operators)
    Else fill A_i with dominated individuals in P_i-1 and A_i-1

  Apply CAO - Component: Artificial Neural Network Training
     -Initialize an RBF NN and train it with A_i
     -Input: Objective Vectors of A_i – Output: Decision Vectors of A_i

  Apply CAO
           – Component: Objective Space local improvement - on A_i
           – Component: Objective Space to Decision Space Predictions
           – Component: Correction Step
           – Update A_i

                        OR (Alternative Usage)

  If CAO is executed by the DM
     Apply CAO
           – Component: Objective Space local improvement - on A_i
           – Component: Objective Space to Decision Space Predictions
           – Component: Correction Step
           – Update A_i
  End
    -Generate new population P_i from A_i – size Nind
                  -Binary tournament selection
                  -Recombination
                  -Mutation
                  -Evaluate objective values for the offspring population P_i
                  -Calculate fitness values of individuals in P_i and A_i based on the
                …-objective vectors of P_i and A_i combined
End loop
```

Figure 3. 8: SPEA2/CAO Pseudocode

Moreover, the parameters of the RBF networks for each optimisation problem solved are investigated within a set of initial experiments and are set to the best parameters achieved. One of the drawbacks of NNs is the lack of standardization in choosing the number of hidden layers and hidden neurons per layer, which constitutes the architecture of a NN. It is common practice to choose the NN architecture based on previous practise and expertise or based on trial-and-error experiments.

-Initialize Population

-Generate random population $\mathbf{P_0}$– size **Nind**

-Evaluate objective values

For i=1 to Gen

-Assign rank to $\mathbf{P_{i-1}}$ *(non-dominated sorting strategy)*

-Determine crowding distance between points on each front of solutions in $\mathbf{P_{i-1}}$

-Generate offspring population **Q** – size **Nind**
- -Binary tournament selection
- -Recombination
- -Mutation
- -Evaluate objective values for the offspring population **Q**

-Combine parent population $\mathbf{P_{i-1}}$ and offspring Population **Q** – size: **2*Nind**

-Assign rank based on Pareto Dominance using non-dominated sorting strategy

-Determine crowding distance between points on each front of solutions (same rank)

-Select **Nind** solutions to form $\mathbf{P_i}$ *(1st: elitist -biased towards lower ranks- 2nd: crowding distance - bias less crowded solutions)*

Apply CAO - *Component: Artificial Neural Network Training*
-Initialize an RBF NN and train it with $\mathbf{P_i}$
-Input: Objective Vectors of $\mathbf{P_i}$ *– Output: Decision Vectors of* $\mathbf{P_i}$

Apply CAO
– *Component: Objective Space local improvement* **- on** $\mathbf{P_i}$
– *Component: Objective Space to Decision Space Predictions*
– *Component: Correction Step*
– *Update* $\mathbf{P_i}$

OR *(Alternative Usage)*

If CAO is executed by the DM
Apply CAO
– *Component: Objective Space local improvement* **- on** $\mathbf{P_i}$
– *Component: Objective Space to Decision Space Predictions*
– *Component: Correction Step*
– *Update* $\mathbf{P_i}$
End

End loop

Figure 3.9: NSGA-II/CAO Pseudocode

The training of the RBF NN is a two-stage process. The first stage consists of setting the parameters (centres and widths) of the radial functions (Gaussian functions) so that the network models the unconditional density of the training data. In the context of the CAO, the training data consists of the elite population of candidate solutions at a certain generation of the MOEA. As stated by Nabney (2001) (p. 199): "*One of the main advantages of RBF networks, as compared to MLP, is that it is possible to choose good (though possibly not optimal) parameters for the hidden units without having to perform a full non-linear optimisation of all the network parameters*".

Table 3. 1: Optimiser configurations

Optimiser Configuration	
Size of Population	100
Crossover operator	Simulated Binary Crossover (SBX) (Deb and Agrawal 1995) Probability: 0.8
Mutation Operator	Polynomial Mutation Probability: 1/(number of Decision Variables)
Number of generations	NSGA-II: 100 NSGA-II/CAO: 50 SPEA2: 100 SPEA2/CAO: 50
Number of Runs	10
Starting Population	Same Random Population (Different at each run)

In this work, 80 percent of the population of candidate solutions (objective vectors) are chosen at random and set as the centres of the RBF basis functions. The widths of the radial basis functions which compose the units of the RBF network, are an application dependent design choice, and should be chosen in way which allows sufficient overlap between the units.

Table 3. 2 Neural Network and Step Size (h) Configuration

RBF Neural Network	
ZDT1- ZDT3-ZDT6	**DTLZ2 (3) – (8) – (12)**
No of Neurons = 80 **RBF Widths = 1** **Step size h = 1.5**	**No of Neurons = 80** **RBF Widths = 5** **Step size h = 0.05**

In the context of this work, fixed widths values were chosen for each problem based on trial-and-error experiments and set to the values presented in Table 3.2. Alternatively, the values of the RBF widths can be optimised using special algorithms or informingly set by clustering, or calculating the average distances between, the training data. However, it should be noted that this latter approach for training the parameters of a RBF NN has the same shortcomings as the back-propagation algorithm when used with MLPs and does not solve the problem of getting stuck at a local optima of the error function. The second stage of the training of the RBF network consists of finding the weights of the output layer by efficiently using linear algebra[12] to solve a quadratic error function. In Table 3.2, the number of neurons used for the RBF neural networks and the initial values used for the step size h are illustrated for each of optimisation problem tackled. The number of neurons and the initial step size for each optimisation problem were determined based on trial-and-error and after experimenting with a range of different values for each of the two parameters.

Performance Metrics

In Zitzler *et al* (2003) it was shown that there is no finite combination of unary metrics which can determine whether an approximation set 'A' outperforms another approximation set 'B'. Zitzler et al (2003) also showed that binary indicators which compare the quality of one approximation set with another in terms of a certain criterion are suitable metrics for concluding that an approximation set is better than another in terms of the inspected criterion. The effectiveness of the CAO when tackling the bi-objective test functions (ZDT1, ZDT2 and ZDT3), the DTLZ test functions with 3, 8 and 12 objectives and the 8-objective real world optimisation problem of aircraft control system design is assessed by using two well-established binary metrics which simultaneously consider the convergence and the diversity requirements:

- The *dominated distance metric* (DD-Metric), which has its roots in Zitzler (1999), computes the dominated distance between two sets of objective vectors in the objective space. More closely, the DD-metric calculates the difference of dominated distances between two approximation sets produced by MOEAs 'A' and 'B' in the objective space. The dominated distance between an approximation set 'A' and an approximation set 'B' (ddAB) is the sum of Euclidean distances between each solution A_i in 'A' and the closest solution B_i which belongs to the subset of 'B' that dominates A_i. The dominated distances ddAB and ddBA are calculated respectively, and their difference forms the value of DD-Metric (A, B).
- The coverage metric (*C-metric*) of Zitzler (1999), which calculates the percentage of solutions in a certain approximation set that are dominated or equal to any solution in another competing approximation set.

Moreover, the *hypervolume* metric (Zitzler 1999) is used to analyze the performance of the optimisers and their CAO-hybridized versions on the bi-objective functions. The values of the hypervolume metric are plotted against the total time (s) spent at each of the 10 executions of the MOEAs to illustrate their convergence extent versus their efficiency.

---

12 The weights of the output layer are efficiently optimised by calculating the pseudo-inverse of the matrix of hidden unit activations.

This approach was previously adopted for the bi-objective test functions in Gaspar-Cunha and Vieira (2004) within a different benchmarking approach which assumes infinite number of objective function evaluations.

The hypervolume metric, also know as the S-metric or the Lebesgue integral, is a high quality unary metric which illustrates the relative quality of an approximation set in terms of both desired criteria –convergence and diversity- by measuring the amount of objective space that the approximation set dominates. Unlike other metrics requiring some prior knowledge about the Pareto front or the targeted tradeoff surface, the computation of the hypervolume metric requires the proposal of an anti-ideal solution to act as a reference point. The values of the hypervolume metric can than be normalised in terms of the hypervolume measure of the ideal solution. In Figure 3.10, the hypervolume metric is illustrated on a bi-objective optimisation problem for visualisation convenience.

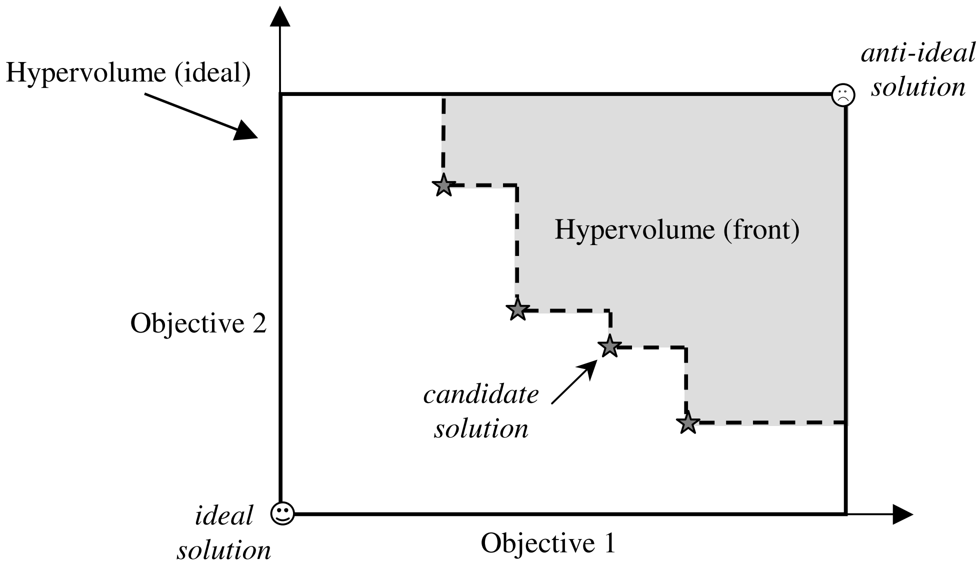


Figure 3. 10 Hypervolume Metric (Minimisation problem assumed)

The hypervolume metric was not deployed as a performance metric on the *many*-objective optimisation problems because of its well-known limitation presented by the metric's computational complexity which is exponential in the number of objectives (Knowles 2002).

3.3.7. Results

The performance and utility of the CAO is investigated in this section. The effect of the introduced operator is examined by deploying the specific performance metrics presented in the previous section and comparing the results achieved by NSGA-II and SPEA2 with the results achieved by their hybridised versions, NSGA-II/CAO and SPEA2/CAO. Moreover, a modified version of the CAO is also implemented for comparison reasons. It is similar to the approach described by Gaspar-Cunha and Vieira (2004) and uses a MLP NN to replace the RBF NN. The modified version of the CAO will be termed as CAO-MLP while the promoted acceleration technique will be termed as CAO-RBF or simply CAO. The MLP configurations used when optimising ZDT1, ZDT2 and ZDT3 were based on trial-and-error and were set to the values used by Gaspar-Cunha and Vieira (2005). The number of hidden neurons and the learning rate of the MLP are (10, 0.2) for ZDT1, (20, 0.3) for ZDT3 and (10, 0.2) for ZDT6. At every generation of the MOEAs, 50 iterations of the standard backpropagation algorithm (Bishop 1995), with a gradient descent optimisation process, is executed for training the MLP neural network and calculating its weights values. The number of hidden neurons and the learning rate of the MLP used with DTLZ2 (3), (8) and (12) are respectively (20, 0.3), (30, 0.3) and (40, 0.3). The same initial values used for the step size h in the CAO-RBF are used with the CAO-MLP.

Bi-objective Test Functions: Results

In Figure 3.11, the values achieved for the S-metric at each of the 10 executions of NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP are illustrated. The three MOEAs were optimising the convex test function ZDT1. The S-metric values achieved at each execution of the algorithms are plotted against the total time spent by each algorithm at the designated execution. The reference point used for calculating the S-metric consisted of the point whose coordinates corresponded to the worst values achieved for each objective by the algorithms combined and within 10 executions.

From Figure 3.11 it can be deduced that NSGA-II/CAO-RBF was consistently, in 9 out of 10 executions, achieving larger values for the S-metric compared to the S-metric values achieved by NSGA-II. Within just 50 generations per execution and a fixed budget of objective function evaluations, the S-metric values achieved by NSGA-II/CAO-RBF were closer to the solid red line which represents the S-metric value of the true Pareto front. Moreover, it was observed that, despite optimising a straightforward and computationally cheap problem (ZDT1), the time spent by NSGA-II/CAO-RBF at each of the 10 executions was comparable to the time spent by NSGA-II. This observation indicates that the CAO-RBF was improving the results achieved by NSGA-II for very little additional cost. Thus the use of CAO-RBF is practical for addressing a wide variety of problems and not just restricted to computationally expensive optimisation problems. On the other hand, NSGA-II/CAO-MLP requires much more time (5 times longer) per execution while presenting some improved results and, at few executions, some remarkable and near optimal values for the S-metric.

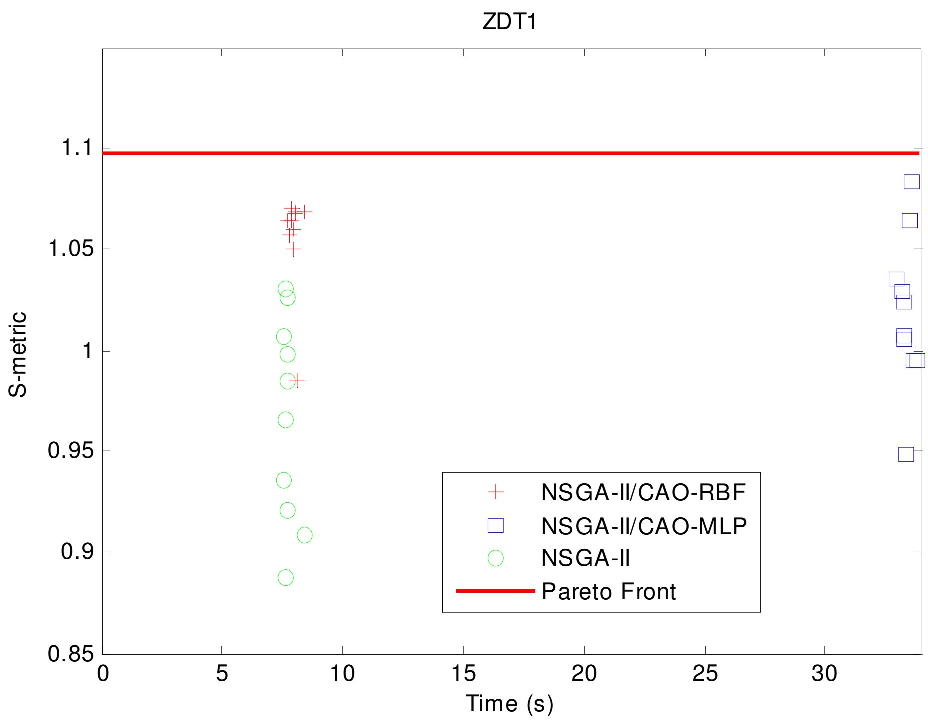


Figure 3. 11: S-metric Values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT1 at each of the 10 executions

A more consistent behaviour for NSGA-II/CAO-MLP can be achieved by optimising the efficiency of the MLP and training it within more iterations (epochs) or using complex training algorithms. An improved behaviour, however, can only be achieved at the expense of increasing the computational complexity of the algorithm. Such tradeoff might be impractical and unacceptable within the context of straightforward optimisation problems such as the ZDTs but desirable when dealing with computationally expensive problems. Nevertheless, from Figure 3.11 it was clear that on a straightforward and computationally cheap problem such as ZDT1, and within the same budget of objective function evaluations, NSGA-II/CAO-RBF was accelerating the convergence of NSGA-II without requiring any significant increase in the computational efforts.

The S-metric values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP for the discontinuous and the non-uniform test functions ZDT3 and ZDT6 are illustrated in Figure 3.14 and 3.17 and similar results are observed. The same observations which highlighted the utility of the CAO (with a MLP or RBF NN in particular) on ZDT1, are observed for ZDT3 and ZDT6. In Figure 3.12, the C-Metric values achieved by NSGA-II and NSGA-II/CAO-RBF are illustrated. For 7 out of 10 executions, the results produced by NSGA-II/CAO-RBF were achieving 100% coverage of the results achieved by NSGA-II. The best coverage achieved by NSGA-II consisted of 21% coverage of the results produced by NSGA-II/CAO-RBF. This was achieved at the 10th execution. Nevertheless, NSGA-II/CAO-RBF achieved 38% coverage of NSGA-II results at the same execution. The remarkably higher coverage achieved by NSGA-II/CAO-RBF (Figure 3.12) reflected a positive contribution by the CAO-RBF.

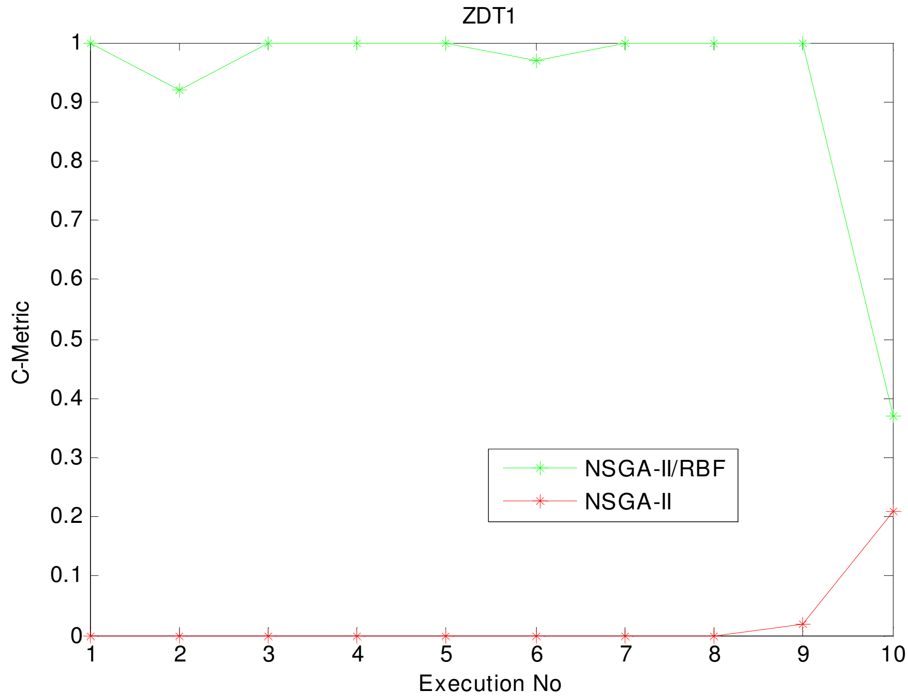


Figure 3. 12 C-Metric Values achieved by NSGA-II and NSGA-II/CAO-RBF on ZDT1 at each of the 10 executions

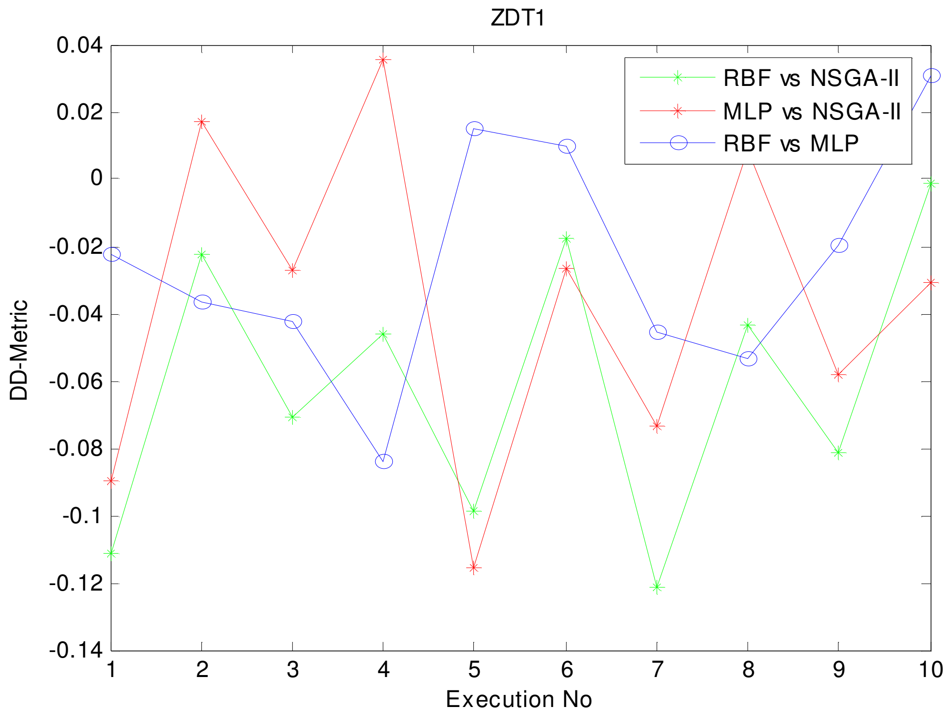


Figure 3.13 DD-Metric Values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT1 at each of the 10 executions

Similarly, the C-metric values achieved by NSGA-II and NSGA-II/CAO-RBF for the discontinuous test function ZDT3 and the non-uniform test function ZDT6 are illustrated in Figures 3.15 and 3.18. The same observations made for ZDT1 concerning the C-metric are again observed for ZDT3 and ZDT6. Moreover, in Appendix B, the C-metric values achieved when comparing the results produced by NSGA-II/CAO-RBF and NSGA-II/CAO-MLP are presented for the convex, discontinuous and non-uniform test functions (ZDT1, 2 and 3). For 7 out of 10 executions, NSGA-II/CAO-RBF was covering and outperforming the results produced by NSGA-II/CAO-MLP when the convex test function was optimised. NSGA-II/CAO-MLP, in its turn, outperformed NSGA-II/CAO-RBF on 3 different occasions, which illustrates some competition between the two CAO approaches. Nonetheless, within limited time and resources, the competition between CAO-RBF and CAO-MLP tends to decrease, highlighting the efficiency of CAO-RBF.

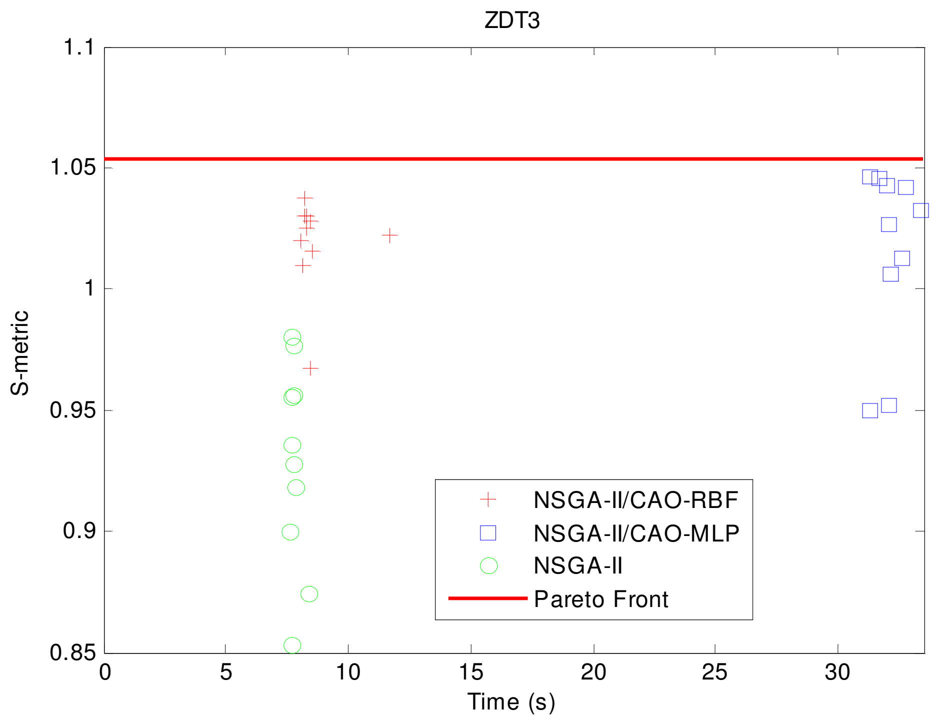


Figure 3. 14: S-metric values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT3 at each of the 10 executions

The competitiveness between CAO-RBF and CAO-MLP seemed to increase when ZDT3 and ZDT6 were optimised with some evidence which favours CAO-RBF in terms of its consistency, robustness and lower computational time. In Figure 3.13, the dominated distance metric (DD-Metric) is computed for ZDT1 and its results are shown for each run of the algorithms. Similar to the C-metric, the DD-metric is a binary metric that highlights whether an approximation set resulting from an algorithm A is better than another approximation set resulting from an algorithm B. A negative DD-metric value denotes that the first input of the metric (e.g. Algorithm A in DD-Metric (A, B)) is better than and dominates most or part of its second input (e.g. Algorithm B)[13]. From the DD-metric results presented in Figure 3.13, it can be observed that NSGA-II/CAO-RBF was outperforming NSGA-II in terms of the DD-metric for all 10 executions of the algorithms. This is highlighted by the green line joining the negative DD-metric values achieved at each execution. On the other hand, when the DD-metric is computed for NSGA-II/CAO-MLP and NSGA-II, NSGA-II/CAO-MLP outperformed NSGA-II in 7 out of 10 executions.

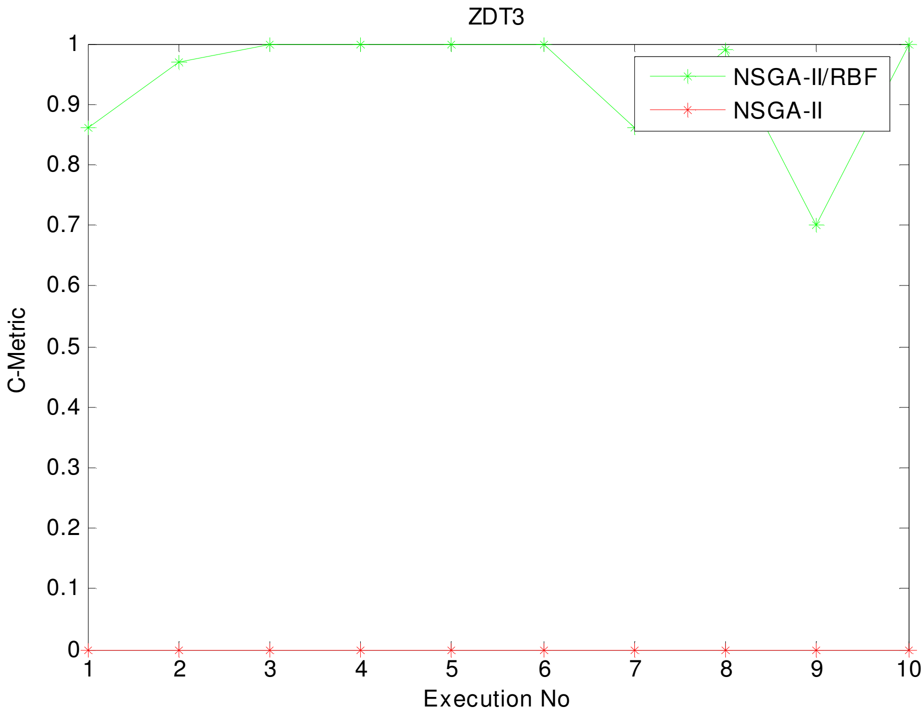


Figure 3. 15 C-Metric Values achieved by NSGA-II and NSGA-II/CAO-RBF on ZDT3 at each of the 10 executions

This is highlighted by the red line in Figure 3.13. Finally, the DD-metric of NSGA-II/CAO-RBF and NSGA-II/CAO-MLP is computed. The blue line in Figure 3.13 indicated that for 7 out of 10 executions NSGA-II/CAO-RBF presented a better performance compared to NSGA-II/CAO-MLP. The DD-metric values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP for the discontinuous test function ZDT3 and the non-uniform test function ZDT6 are illustrated in Figures 3.16 and 3.19. The results presented in Figures 3.16 and 3.19 depict the same DD-metric observations made for ZDT1. CAO-RBF and CAO-MLP both proved to be competent, introducing improvements to the results achieved by NSGA-II. The use of a RBF NN within the CAO is shown to be

[13] In Figure 3.14, 3.18 and 3.22, RBF vs MLP, for example, denotes that Algorithm A is NSGA-II/CAO-RBF and Algorithm B is NSGA-II/CAO-MLP

more practical than a MLP NN mainly due its much faster training process which makes it efficient for deployment within a convergence acceleration technique.

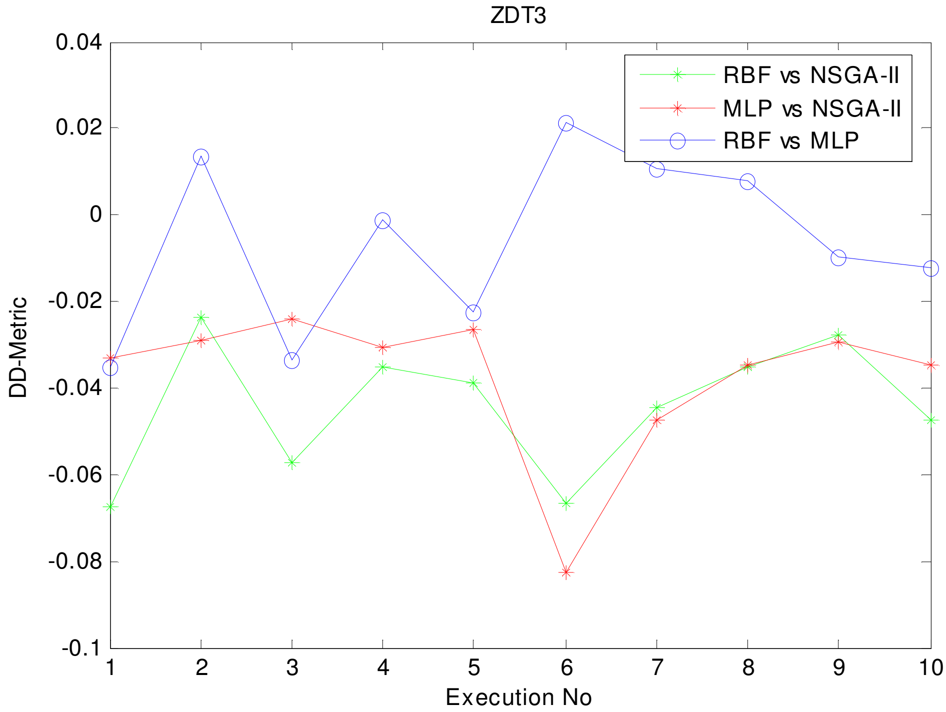


Figure 3.16 DD-Metric Values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT3 at each of the 10 executions

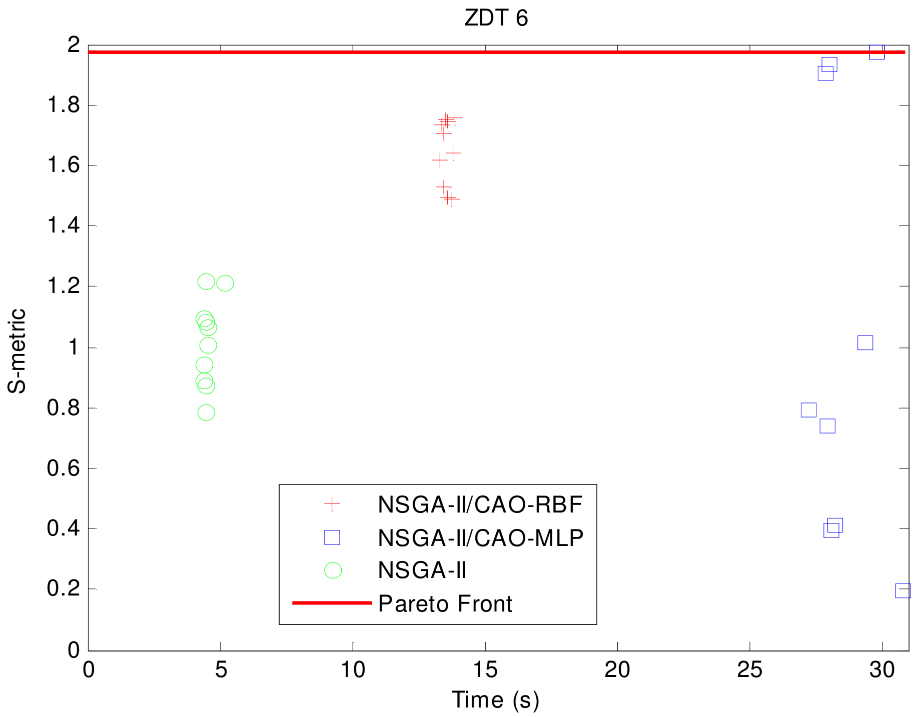


Figure 3.17: S-metric values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT6 at each of the 10 executions

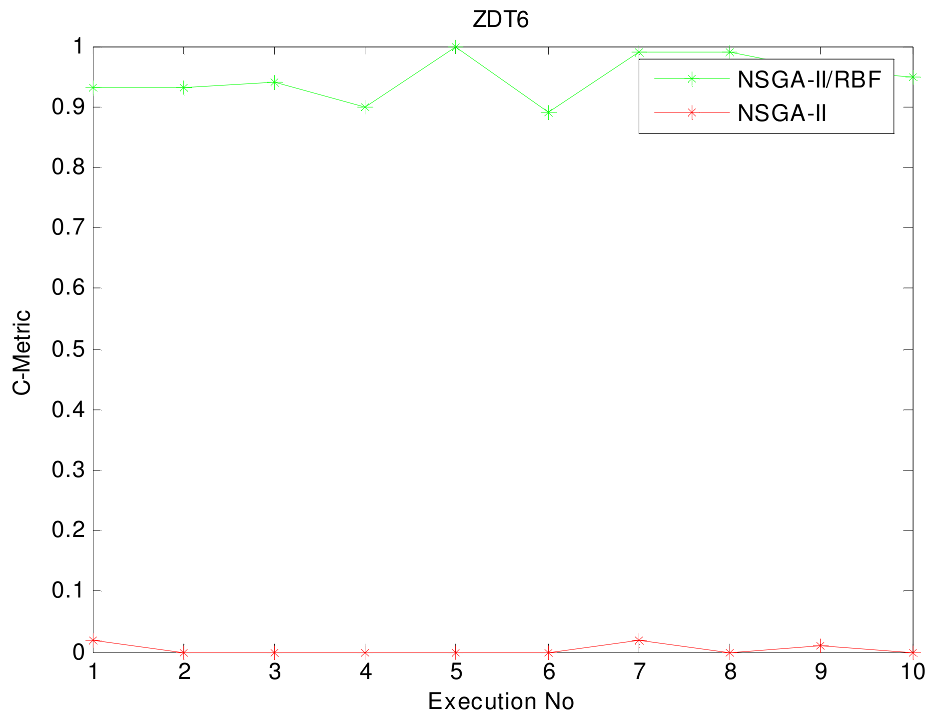


Figure 3.18: C-Metric Values achieved by NSGA-II and NSGA-II/CAO-RBF NN on ZDT6 at each of the 10 executions

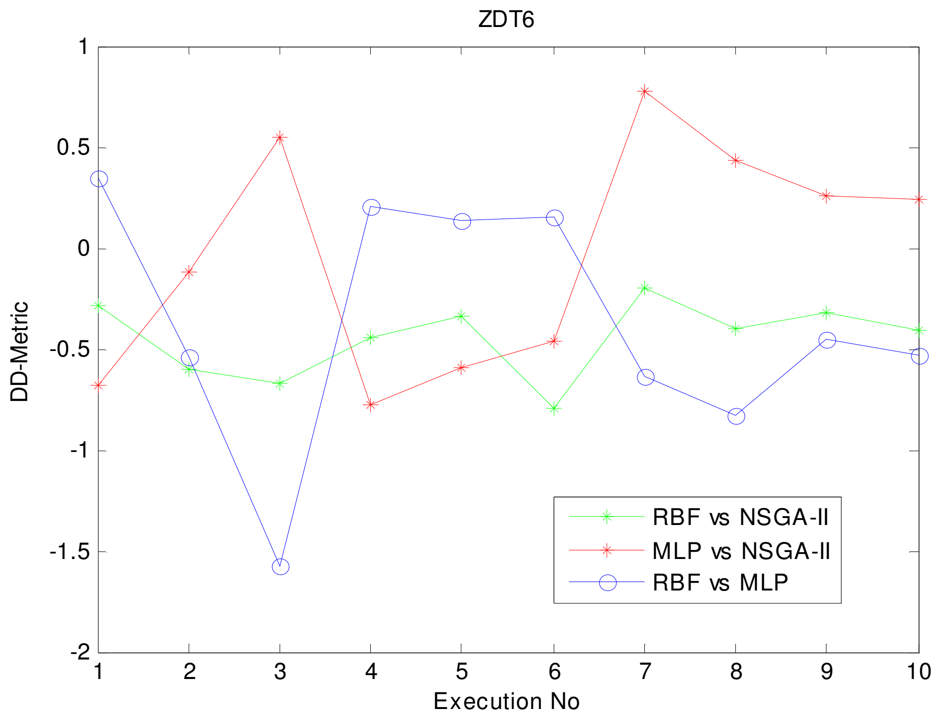


Figure 3.19: DD-Metric Values achieved by NSGA-II, NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT6 at each of the 10 executions

Similar observations are made when the CAO is hybridised with SPEA2. In fact, the CAO seemed to introduce even more benefits to the performance of SPEA2. The results achieved for the S-metric, C-metric and DD-metric when the CAO is hybridized with SPEA2 are illustrated in Appendix C, and, again, demonstrate the impact of the CAO on one of the best-performing MOEAs.

Scalable test Function DTLZ2: Results

Tables 3.2 – 3.7 illustrate the results highlighting the effect of the CAO on optimisation problems with a larger number of objectives. The scalable test function DTLZ2, with 3, 8 and 12 objectives, was chosen to investigate the performance of the CAO. In a similar manner to the experimentations carried on the bi-objective problems, the effect of the CAO was underlined by contrasting NSGA-II with its CAO hybridized counterparts (NSGA-II/CAO-RBF and NSGA-II/CAO-MLP).

Table 3.2: C-metric results for DTLZ2 (3)

Run No:	DTLZ2 - 3 Objectives (A = NSGA-II/CAO-RBF, B = NSGA-II/CAO-MLP and C = NSGA-II)			
	C-Metric (A, C)	C-Metric (C, A)	C-Metric (B, C)	C-Metric (C, B)
1	7%	1%	3%	1%
2	7%	2%	5%	1%
3	7%	0%	7%	0%
4	9%	1%	1%	3%
5	4%	4%	0%	8%
6	7%	3%	5%	0%
7	4%	3%	5%	3%
8	4%	1%	3%	2%
9	4%	4%	1%	3%
10	7%	1%	1%	1%
Mean Value:	6%	2%	4%	2.2%

Table 3.3: DD-metric results for DTLZ2 (3)

Run No:	DTLZ2 - 3 Objectives (A = NSGA-II/CAO-RBF, B = NSGA-II/CAO-MLP and C = NSGA-II)	
	DD-Metric (A, C) $.10^{-3}$	DD-Metric (B, C) $.10^{-3}$
1	-0.838	1.025
2	-1.140	-1.010
3	-2.311	-1.560
4	-1.683	0.0811
5	1.324	1.780
6	-2.105	-1.245
7	2.574	-0.163
8	-1.293	-0.672
9	0.752	-0.351
10	-1.297	-2.746
Mean Value:	-0.601	-0.486

The experiments have shown that the fronts achieved by the CAO-hybridised versions of NSGA-II most frequently achieve a higher coverage compared to the coverage achieved by NSGA-II (Tables 3.2, 3.4 and 3.6). Over the 10 executions of the algorithms, NSGA-II/CAO-RBF produced an average of 6%, 18.5% and 3.10% coverage of the results achieved by NSGA-II for the 3, 8 and 12 objective versions of DTLZ2 respectively.
On the other hand, NSGA-II only achieved an average of 2%, 0.09% and 0.09% coverage of the results achieved by NSGA-II/CAO-RBF for DTLZ2 (3-8 and 12) including several runs with 0% coverage. NSGA-II/CAO-MLP has similarly produced a coverage of NSGA-II results which is higher than the coverage achieved by NSGA-II on all three versions of DTLZ2. On average, NSGA-II/CAO-MLP covered 4%, 15.5% and 2.8% of the results produced by NSGA-II for DTLZ2 (3), (8) and (12) respectively while NSGA-II only achieved an average coverage of 2% for DTLZ2 (3) and 0.001% for DTLZ2 (8) and (12). Based on the C-metric results highlighted in Tables 3.2, 3.4 and 3.6, it was observed that when hybridized with NSGA-II, the 2 versions of CAO were producing higher C-metric values compared to the standalone NSGA-II.

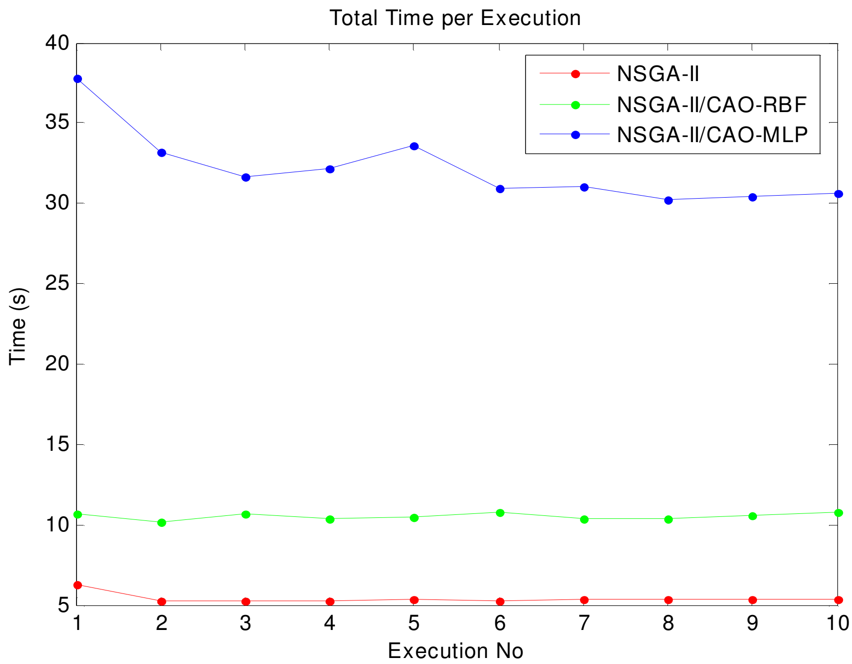


Figure 3.20 Computational Time per execution for DTLZ2 (3)

The same observations highlighted for the C-metric were observed for the DD-metric (Tables 3.3, 3.5 and 3.7). The DD-metric has consistently produced results (< 0) which favoured NSGA-II/CAO (RBF and MLP) over NSGA-II for all dimensions of the problems investigated. Nevertheless, it was remarkable that the RBF version of CAO was introducing the best C-metric and DD-metric values compared to NSGA-II and NSGA-II/CAO-MLP. In addition, the use of the RBF neural network was fulfilling its intended purpose by making NSGA-II/CAO-RBF more efficient and requiring less computational effort compared to NSGA-II/CAO-MLP.

Table 3.4: C-metric results for DTLZ2 (8)

Run No:	DTLZ2 - 8 Objectives (A = NSGA-II/CAO-RBF, B = NSGA-II/CAO-MLP and C = NSGA-II)			
	C-Metric (A, C)	C-Metric (C, A)	C-Metric (B, C)	C-Metric (C, B)
1	37%	0%	5%	0%
2	24%	0%	8%	0%
3	16%	2%	24%	0%
4	12%	0%	19%	0%
5	16%	0%	4%	0%
6	23%	0%	19%	0%
7	9%	0%	21%	0%
8	22%	2%	18%	0%
9	9%	4%	5%	1%
10	17%	1%	32%	0%
Mean Value:	18.5%	0.09%	15.5%	0.001%

In Figures 3.20, 3.21 and 3.22, the computational time spent at each execution of the algorithms optimising DTLZ2 (3) (8) and (12) is illustrated. The time complexity of NSGA-II/CAO-MLP was on average 3 times larger than the time complexity of NSGA-II/CAO-RBF. The use of simple and computationally cheap test functions (ZDTs and DTLZ2) for assessing the CAO has helped emphasize the efficiency of CAO-RBF over CAO-MLP by putting their performance into perspective.

Table 3.5: DD-metric results for DTLZ2 (8)

Run No:	DTLZ2 - 8 Objectives (A = NSGA-II/CAO-RBF, B = NSGA-II/CAO-MLP and C = NSGA-II)	
	DD-Metric (A, C) $.10^{-3}$	DD-Metric (B, C) $.10^{-3}$
1	-720.44	-67.361
2	-404.82	-99.053
3	-264.05	-402.810
4	-145.32	-315.900
5	-236.01	-65.649
6	-360.64	-315.530
7	-133.75	-438.230
8	-307.27	-427.580
9	-110.53	-40.731
10	-192.23	-579.870
Mean Value:	-287.500	-275.300

From the results presented in this section, it was observed that despite the improvements introduced by the CAO, the DD-metric and the C-metric results were highlighting a closer competition between NSGA-II/CAO and NSGA-II on the many-objective problems. This observation makes it intriguing to assess the reliability and accuracy of the C-metric and the DD-metric as the dimensionality of the objective space increases. This issue will be investigated closely in Chapter 4. From the results illustrated in this section, it is also remarkable that in the 8-objectives version of DTLZ, the CAO exhibits the most significant improvement in coverage and dominated distance measures. This feature deserves further study in order to understand why the performance on the 8-objectives version might be significant for this dimension of problem.

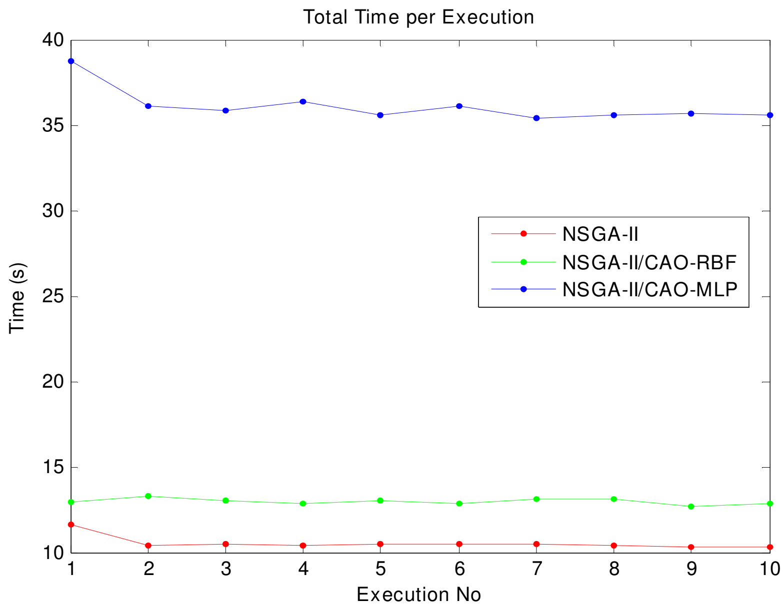


Figure 3.21 Computational Time per execution for DTLZ2 (8)

Table 3.6: C-metric results for DTLZ2 (12)

Run No:	DTLZ2 - 12 Objectives (A = NSGA-II/CAO-RBF, B = NSGA-II/CAO-MLP and C = NSGA-II)			
	C-Metric (A, C)	C-Metric (C, A)	C-Metric (B, C)	C-Metric (C, B)
1	0%	0%	3%	0%
2	0%	0%	1%	0%
3	2%	0%	0%	0%
4	3%	0%	3%	0%
5	2%	0%	4%	0%
6	3%	0%	7%	0%
7	5%	0%	5%	0%
8	8%	0%	1%	1%
9	3%	0%	3%	0%
10	5%	0%	1%	0%
Mean Value:	3.10%	0%	2.80%	0.001%

Studying the interactions between the decision variables and the objectives, through the use of heatmaps (Pryke, Mostaghim and Nazemi 2007) or estimation of distribution algorithms (EDAs) (Pelikan, Goldberg and Lobo 1999) for example, is a suggested approach for investigating the behaviour of CAO on DTLZ2 (5).

Table 3.7: DD-metric results for DTLZ2 (12)

Run No:	DTLZ2 – 12 Objectives (A = NSGA-II/CAO-RBF, B = NSGA-II/CAO-MLP and C = NSGA-II)	
	DD-Metric (A, C) $.10^{-3}$	DD-Metric (B, C) $.10^{-3}$
1	0	-38.451
2	0	-11.933
3	-28.295	0
4	-24.469	-21.525
5	-6.104	-65.927
6	-54.574	-108.940
7	-92.507	-76.071
8	-184.270	-4.692
9	-15.873	-27.631
10	-97.144	-20.298
Mean Value:	-50.300	-37.500

Similar to NSGA-II/CAO, when the CAO is hybridised with SPEA2, SPEA2/CAO has out-performed SPEA2 on all three versions of DTLZ. The C-metric and DD-metric results achieved for the set of experiments contrasting SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP are presented in Appendix C. Further experiments were undertaken in an attempt to quantify the extent of superiority of the CAO hybridized optimisers. It was noted that, on average, the population size of NSGA-II and SPEA2 must be increased to a minimum of 150 individuals (1.5x the population size of NSGA-II/CAO and SPEA2/CAO) in order to match the quality of the fronts achieved by their hybridized counterparts.

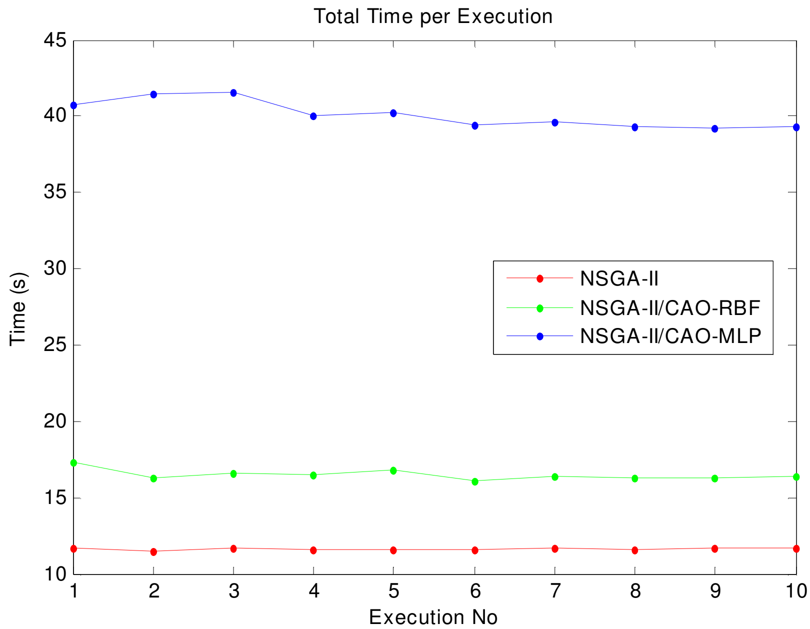


Figure 3.22 Computational Time per execution for DTLZ2 (12)

Thus, SPEA2 and NSGA-II require more objective function evaluations (around 5000 more evaluations) to match the performance of their CAO hybridised equivalent optimiser. This conclusion holds for all the test functions used in this work. The set of experiments conducted in this section highlights the benefits of the CAO in general and the CAO-RBF in particular and demonstrate the improvement it confers to two of the most established MOEAs.

Aircraft Control System Design Problem: Results

The NSGA-II/CAO (RBF) was also tested against the multiobjective optimisation problem of aircraft control system design (ACSD). The intention was to contrast the performances of NSGA-II and NSGA-II/CAO on a real world application. In Table 3.8, the dominated distance metric (DD-Metric) and the Coverage metric (C-metric) are computed and their results are shown for each run of the algorithms. These experiments have shown that the fronts achieved by NSGA-II were constantly outperformed by their counterparts deploying the CAO.

The DD-metric consistently has produced negative results favouring NSGA-II/CAO over NSGA-II for the ACSD problem. On the other hand, the solutions achieved by NSGA-II/CAO over the 10 executions of the algorithms have covered an average of 4.6% of the solutions achieved by NSGA-II for the 8-objectives problem. NSGA-II only scored an average of 0.8% coverage of the results achieved by NSGA-II/CAO, including several runs with 0% coverage.

Table 3.8: DD-metric and C-metric results for the ACSD problem

Run No:	Aircraft Control System Design (8 Objectives, A = NSGA-II/CAO and B = NSGA-II)		
	DD-Metric (A, B) .10-3	**C-Metric (A, B)**	**C-Metric (B, A)**
1	-844.7	2%	0%
2	-433.9	5%	3%
3	-2599.5	3%	0%
4	-935.1	9%	0%
5	-588.7	4%	1%
6	-354.02	0%	0%
7	-3519.2	12%	1%
8	-3454.4	7%	2%
9	-6221.1	4%	0%
10	-2892.0	0%	1%
Mean Value:	-2184.3	4.6%	0.8%

Computational Effort

In this section, the additional computational effort required by the CAO (the RBF NN learning process in particular) is reported. It should be noted that such computational effort measurements depend on the hardware/software resources available. In this study, *MATLAB*[14] was used for implementing, executing and testing all the optimisation frameworks presented in this work. Furthermore, all the experiments were undertaken on a *Pentium 4* machine with 512 megabyte of *Random Access Memory* (RAM). Netlab (Nabney 2001), an open source neural network toolbox for use with Matlab, was used for implementing, training and validating the ANN used in the CAO context.

The neural network training process is a structural and arithmetical process which deals with numerical data (inputs and outputs). Hence, the computational effort required for training the neural network is highly influenced by the number of inputs, outputs, weights and parameters (especially if the last two are being optimised), rather than in terms of the complexity of the objective function being solved by the hosting MOEA. The neural network architecture, configuration and training algorithms are thus the variables that affect the computational effort required for training a NN. These variables form an optimisation problem in themselves. In this work, these variables were carefully chosen after a considerable number of trial-and-error experimentations. In order to formulate an idea about the computational effort of the CAO deployed in this study, the portable component was hybridized with NSGA-II in an optimisation framework attempting to solve the *Gasifier* problem (Griffin *et al* 2000), a relatively expensive (computationally) problem. Gasifiers are increasingly used reactors for power generation from coal, due to their cleanliness and environment friendliness. The gasifier problem described in Griffin *et al* (2000) is a design and optimisation problem of a control system for a linear model of the gasifier. The gasifier problem was purely used in this section as a computationally expensive -14-objective- benchmark problem for contrasting the computational effort required by the CAO (in particular the training and validation of the RBF neural network) and the computational effort required for computing the gasifier's objective function. For more information about the gasifier problem, the interested reader is directed to Griffin *et al* (2000). In Figure 3.23, the values achieved by NSGA-II/CAO and NSGA-II at each of the 10 executions are shown. NSGA-II/CAO was covering an average of 34% of the results achieved by NSGA-II, while NSGA-II was only covering an average of 5% of the results achieved by its CAO-hybridized counterpart.

---

[14] Matlab is a software package for technical computing, developed by The MathWorks, Inc.

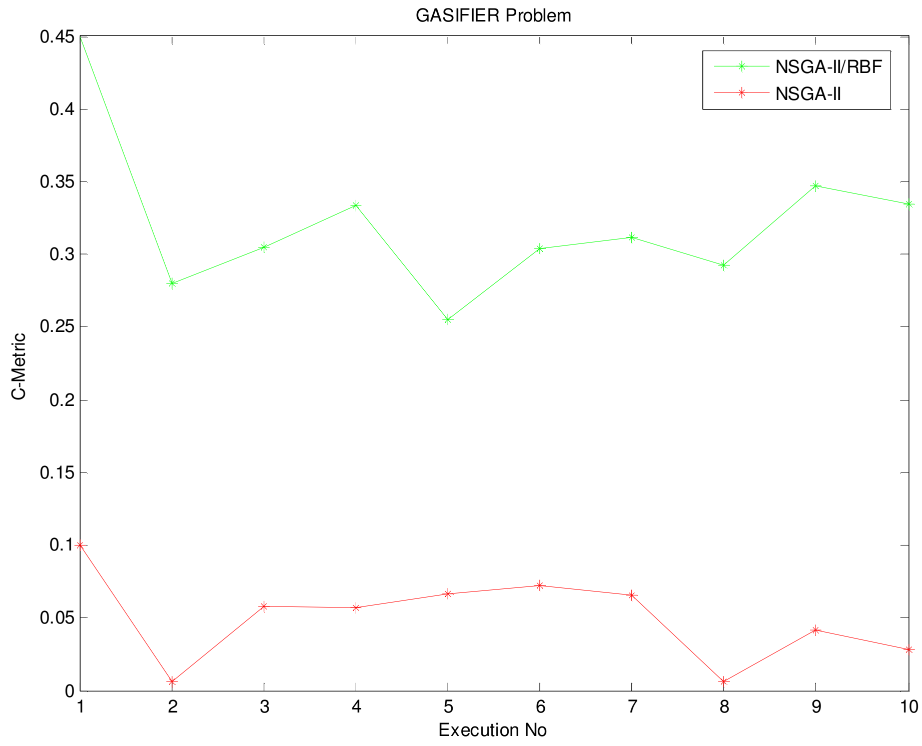


Figure 3. 23 C-Metric Values achieved by NSGA-II/CAO-RBF and NSGA-II on the Gasifier problem at each of the 10 executions

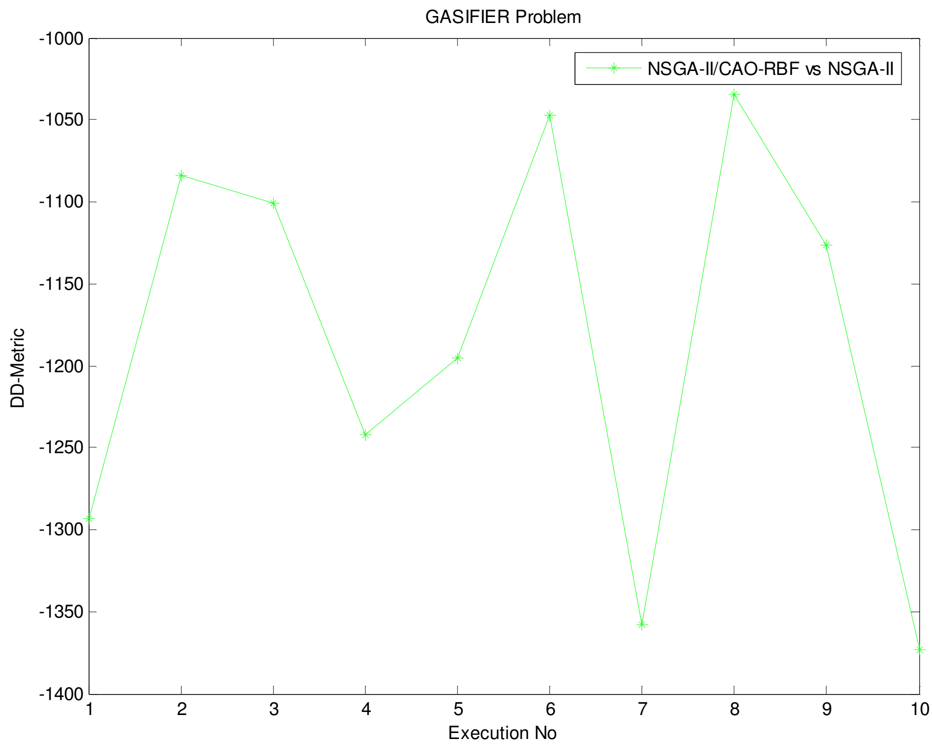
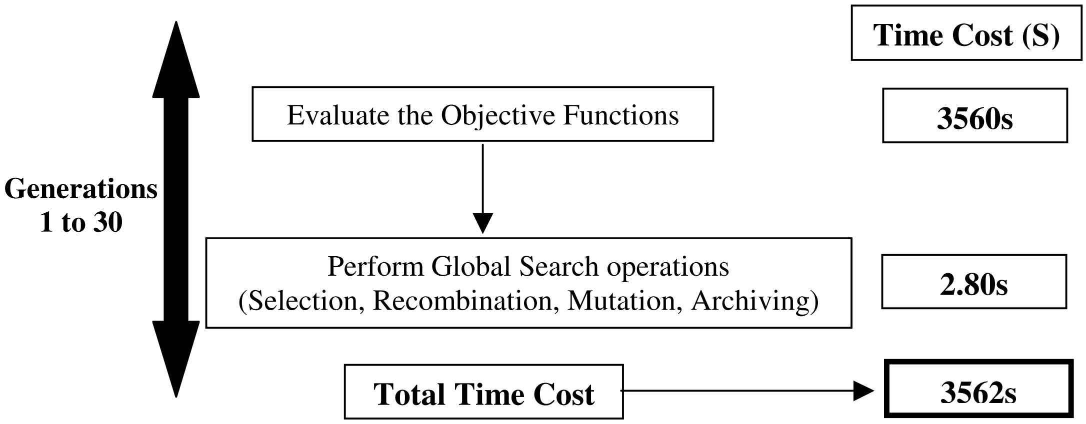
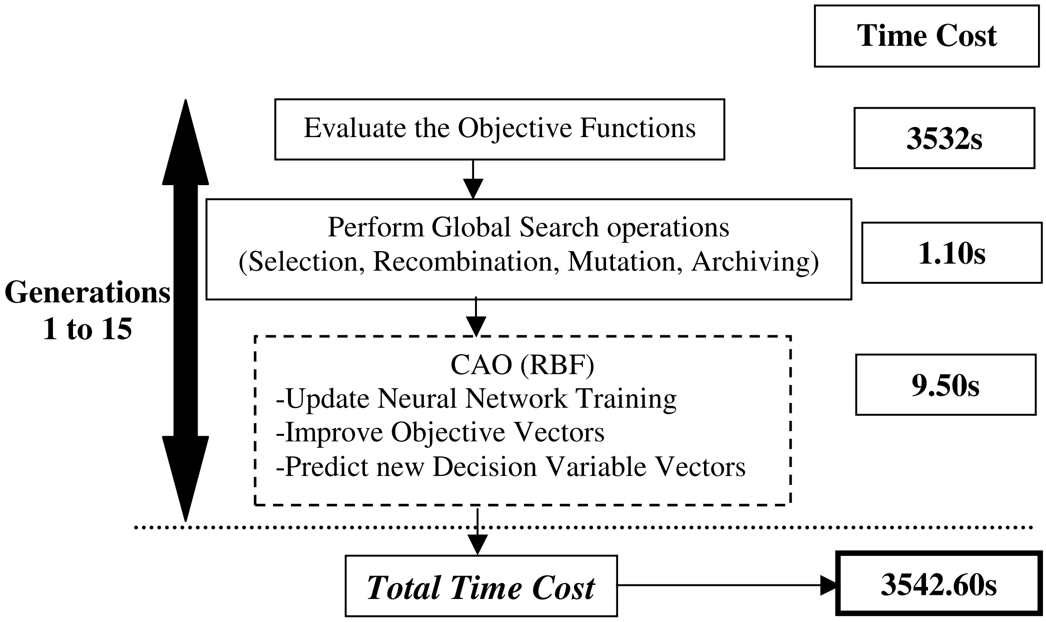


Figure 3. 24 DD-Metric Values achieved by NSGA-II/CAO-RBF and NSGA-II on the Gasifier problem at each of the 10 executions



(Gasifier application, Pop Size: 20, Number of generations: 30)

Figure 3. 25: NSGA-II Computational Effort



(Gasifier application, Pop Size: 20, Number of generations: 15)

Figure 3. 26: NSGA-II/CAO Computational Effort

In a similar way, the DD-metric values presented in Figure 3.24 consisted of negative values which highlighted the outperformance of NSGA-II/CAO over NSGA-II at all 10 executions. The computational effort measurements of the major components of NSGA-II and NSGA-II/CAO optimising the gasifier problem for 30 and 15 generations respectively are presented in Figures 3.25 and 3.26. Except for the reduced number of generations (15 for NSGA-II/CAO and 30 for NSGA-II), population size (= 20) and the initial value of the step size h (= 20), the same configuration used for NSGA-II and NSGA-II/CAO in the previous sections was deployed. Note that in Griffin et al (2000), a larger population size and number of generations were deployed for optimising the gasifier problem. However, the configuration used in this section was deemed sufficient since the goal of the presented experiments was to contrast the efficiency of NSGA-II and NSGA-II/CAO within a limited budget of objective evaluations rather than solving the gasifier problem itself.

From the results illustrated in Figures 3.25 and 3.26, it was clear that the total computational time required by NSGA-II and NSGA-II/CAO were comparable. In fact, NSGA-II/CAO has required 3542.60 seconds to perform the optimisation process (Figure 3.26), saving approximately 18 seconds when compared to the computational effort required by NSGA-II (3562 seconds) to perform the equivalent optimisation process (Figure 3.35). Moreover, compared to the total computational time used for calculating the gasifier's objective function (3532 seconds - ≈ 59 min- in Figure 3.26), the total computational time spent for training and validating the RBF neural network within the CAO was very negligible (9.5 seconds → ≈ 0.27 % of the total time required for calculating the gasifier's objective function). In computationally expensive problems such as the gasifier application, it is worth noting that the computational time required for training and using a MLP NN instead of a RBF NN can also be considered as negligible compared to the time needed for computing the objective functions. As a result, despite the preferability for using the CAO with a RBF neural network, in expensive optimisation applications, the use of a MLP NN within a convergence accelerator such as the CAO can be considered as equally beneficial and efficient.

3.3.8. Conclusion

A portable convergence accelerator operator has been proposed for incorporation in existing algorithms for evolutionary multiobjective optimisation. This operator works by suggesting improved solutions in objective space and using neural network mapping schemes to predict the corresponding solution points in decision variable space. Two leading MOEAs have been hybridised through introduction of the CAO and tested on a variety of recognised test problems. These test problems consisted of convex, discontinuous and non-uniform test functions, with numbers of objectives ranging from 2 to 12. In all cases introduction of the CAO led to improved results for comparable numbers of function evaluations.

When deploying an active strategy for promoting diversity within a slowly converging process to the Pareto front, the convergence process of a MOEA can be hampered and delayed, especially in optimisation problems with many competing objectives. This is an important issue which will be addressed in Chapter 4. Due to the convergence acceleration caused by the CAO, the active diversification mechanisms of the hosting MOEAs get an increasing emphasis. However, the increasing emphasis of the active diversification mechanisms is manifested at converged and near optimal regions of the search space rather than at remote and suboptimal regions. In other words, the CAO not only enhances the convergence towards the Pareto fronts, but should also help enhancing the diversity at regions of interests and promising areas of the search space which are closer to the Pareto front.

It is important to recognise that the CAO introduces additional computational effort through the requirement to train the neural network. When using a RBF NN, this computational effort is negligible even when compared with the execution time associated with computing a ZDT function, for example, since these functions are trivially simple to compute. However, using a MLP within the CAO increases the computational effort and makes it substantial when compared with the execution time associated with the computation of a ZDT function. For example, the MLP training time proved to be approximately 1500 times that of computing the ZDT functions. Clearly, one would not advocate use of CAO-MLP in such situations. However, in a real-world problem such as the ALSTOM gasifier problem (Griffin *et al* 2000), it was found that MLP training time proved to be approximately one-hundredth of the time required to compute the ALSTOM gasifier problem objectives. Moreover, here we have not sought to optimise performance of the NN mapping methodology. Nonetheless, it was demonstrated that while the CAO with a RBF NN is a practical and efficient approach for accelerating the convergence of MOEAs on a variety of problems (ZDT(1, 3, 6), DTLZ2 (3, 8, 12), the ACSD problem and the Gasifier problem), the CAO with a MLP is better restricted for use in real-world problems where objective function computation is non-trivial.

Thus, a portable operator has been described that can be incorporated into any MOEA to improve its convergence. However, it should be noted that the ZDT and DTLZ2 test functions used in this study presented common similarities in the decision space. In particular, the different versions of the DTLZ2 test function used in this study are characterised by the fact that the last k decision variables of any Pareto optimal solution presented the same value. This last feature is rarely present in real life applications and can be seen as simplifying the NN prediction process deployed within the CAO. As a result, investigating the performance of the CAO on challenging test functions, such as the test functions suggested by Huband et al (2005), or other real problems with non-separable variables and deceptive/multimodal decision spaces should be undertaken in future work.

3.4. Summary

In this Chapter, a strategy for addressing the MO requirement for solutions convergence towards the Pareto front of MOPs was introduced. The suggested strategy consisted of the convergence acceleration operator (CAO). The CAO is a portable component that can be integrated within the cycle of any multiobjective evolutionary optimiser. CAO operates by performing direct manipulations in the objective space to guide the evolutionary search towards good areas of solutions and adding a profitable deterministic process to the EAs stochastic approach. CAO also integrates machine-learning strategies in the form of neural networks for acquiring knowledge about the mapping function from the objective space to the decision variable space. The acquired knowledge is used efficiently to accelerate the convergence of MOEAs. CAO was hybridized with two of the best MOEAs and tested on a variety of optimisation problems and number of objectives. The introduced operators proved successful and the hybrid algorithms significantly outperformed the original algorithms.

Having addressed the convergence requirement for multiobjective optimisation problems, the diversity requirement is studied in the following Chapter.

Chapter 4

Evolutionary Multiobjective Optimisation: Enhancing the Diversity Perspective

4.1. Introduction

Finding a "good" set of solutions to a multiobjective optimisation problem consisting of *m* objectives can be more accurately thought of as an optimisation scenario with *m+2* objectives. These *m+2* objectives are divided into *m* tangible and application specific objectives and an additional 2, general and theoretical, objectives. The latter 2 objectives are the required *convergence* of the solutions to a MOP towards the Pareto front and their *diversity* across the tradeoff surface in the objective space. In certain scenarios, diversity might be required in the decision space as well. When solving multiobjective optimisation problems, the existence of objective preferences and priorities, and their incorporation in the search process is an optional, and application dependent scenario. Nevertheless, one thing is explicit and common: the solutions' *convergence* criteria is usually biased and prioritised over the *diversity* criteria. As a result, diversity promotion is usually deployed as a second consideration to proximity promotion in most MOEAs. This is well justified since, as stated by Bosman and Thierens (2003):

> *... the goal is to preserve diversity along an approximation set that is as close as possible to the Pareto optimal front, rather than to preserve diversity in general, the exploitation of diversity should not precede the exploitation of proximity.*

In fact, the diversity requirement is mainly sought in multiobjective optimisation scenarios with competing objectives where no single optimal solution can be found. In such scenarios, diversity is requested to provide the decision maker with a diversified set of solutions, with varying tradeoff performances across the objectives, to choose from. This prioritization of the convergence requirement over the diversity can be observed in the selection for variation and the selection for survival procedures of most MOEAs. In the context of NSGA-II, the selection for variation used is a type of binary tournament selection (Brindle 1981) termed as the *crowded-comparison operator* and defined by Deb *et al* (Deb, Pratap, Agarwel, and Meyarivan 2002). When selecting solutions for inclusion in the mating pool, two solutions are first chosen randomly from the population and compared primarily in terms of their non-dominated ranks. The solution with the higher –Pareto dominance based- rank is selected for inclusion in the mating pool. In the case where the two selected solutions share the same rank, and hence are equally good in terms of Pareto dominance, the secondary criterion for the selection for variation would consists of their crowding measure and therefore the diversity requirement. The solution lying in the less dense area of the space would be selected for inclusion in the mating pool, thus promoting diversity. In the case where the two selected solutions shared the same rank and populated equally dense areas of the objective space, one of the solutions get selected at random for inclusion in the mating pool.

Similarly, the selection for survival process in NSGA-II uses the same hierarchy of selection criteria. Deploying a strategy that maintains a fixed size for the online archive, NSGA-II uses a selection for survival procedure that starts by filling the online archive with the highest ranked solutions following a non-dominated sorting strategy. These solutions are selected from the set of solutions combining the newly produced solutions at a certain generation of the optimisation process and the solutions populating the online archive at the same generation. Only in the case where filling the empty –remaining- slots of the archive necessitates the selection of a subset of solutions from a certain non-domination level or rank does the diversity requirement intervene as a selection criterion. The selection for survival procedure used in NSGA-II is illustrated in Figure 4.1.

Having described the diversity requirement and its priority in EMO, in this Chapter the requirement for solutions' diversity in multiobjective optimisation, with an emphasis on optimisation problems consisting of *many* conflicting objectives, is explored.

In section 4.2, the motivation for promoting and introducing new methods for diversity promotion and maintenance in EMO, especially as the number of conflicting objectives increases, is justified. In section 4.3, a new approach for tackling the diversity requirement from a new perspective is introduced. The suggested approach is embodied in an adaptive local strategy that incorporates well-established diversity indicators within the cycle of MOEAs for guiding and manipulating the optimisation and search process.

In Section 4.4, the experimental results achieved by the introduced strategy for a set of multiobjective optimisation problems with increasing number of conflicting objectives are illustrated. Performance evaluation and conclusions will be based on an analytical comparison of the results produced by the suggested optimisation strategy and another well-established and representative MOEA. A summary of the chapter is presented in Section 4.5.

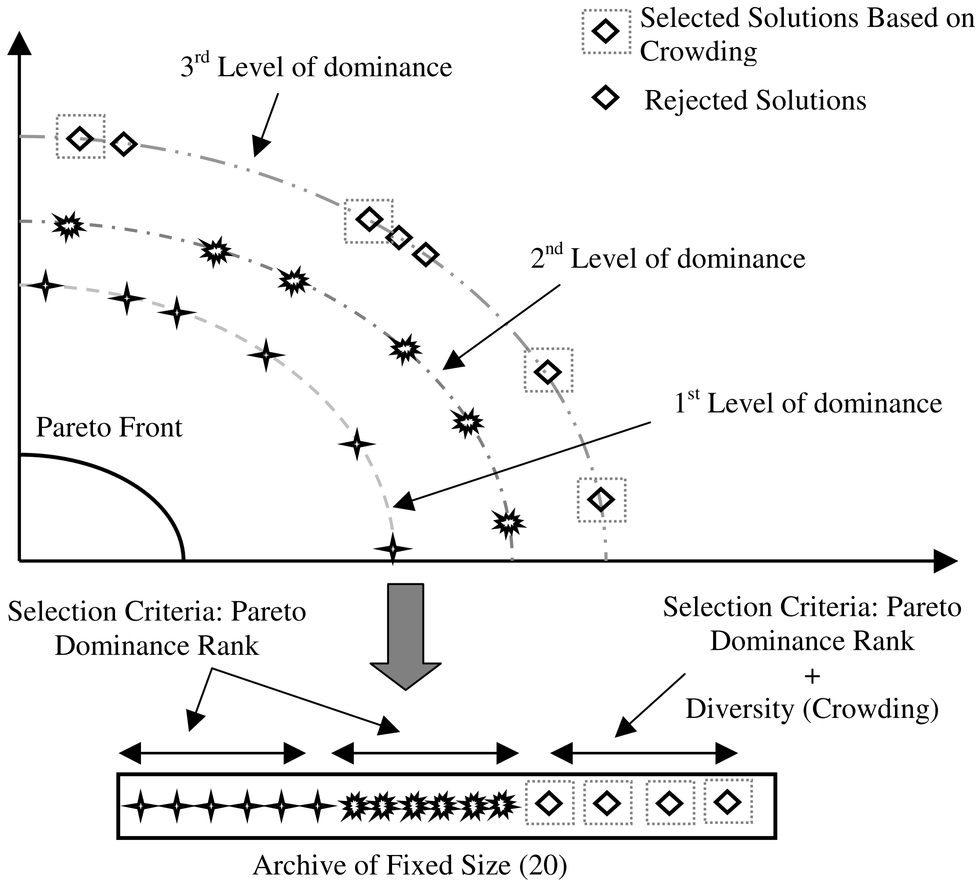


Figure 4. 1: NSGA-II Selection for Survival

4.2. Study Framework and Motivation

Since the late 1980s, many diversity promotion strategies were introduced in EMO. In Chapter 2, an overview of the major milestones in the development of diversity promotion methods for EMO is illustrated. Their known benefits and limitation were also stated. Most of the early approaches for obtaining good distribution of solutions in an approximation set required some density estimation of the population around every single solution. Although some of the density estimators used in the diversity promotion strategies in EMO were borrowed from other scientific disciplines, mostly statistics, other techniques used for density estimation were specifically devised for EMO. The motivation for further research for this particular requirement of EMO is subsequently described.

The Motivation for New Diversity Promotion Techniques

The objective of the study presented in this Chapter is to investigate new approaches and strategies for promoting diversity in evolutionary *many* objective optimisation. This is initially motivated by the outcome of the studies reported in Section 2.3.4 that highlighted the shortcomings of the different diversity promotion techniques (see Chapter 2). Moreover, Purshouse and Fleming (2003b) and Purshouse (2004) showed that classical settings for a representative set of recombination and mutation operators might be suitable for optimisation problems with a small number of objectives, but are inappropriate for optimisation frameworks with a large number of conflicting objectives. The experimental results of their study highlighted the conflict between the primary MOP requirement for convergence towards the Pareto front, and the secondary requirement for maintaining diversity in the approximation set. This conflict between the convergence and diversity requirements in multiobjective optimisation has a detrimental impact on the optimisation process and is particularly aggravated and detected in the *many*-objective optimisation frameworks.

In fact, the size of the feasible objective space for a certain MOP increases with the increase in the dimensionality of the optimisation problem -in terms of the number of objectives. Consequently, the probabilities for the handled solutions to become locally non-dominated (Pareto dominance wise) increase. In fact, no matter how far it is from the Pareto front, any solution close to a certain axis of performance in the objective hyperspace will have a high chance of being preferred in terms of Pareto dominance *and* the diversity criterion (being considered as an extreme solution) for selection for variation and survival. The increasing proportion of non-dominated solutions explored in large objective hyperspaces leads to the diversity promotion mechanisms becoming more emphasized and hence, diversity gradually becomes the primary selection criterion. This tends to overemphasize the diversification process at the expense of the convergence requirement. As a result, when performing selection for variation, solutions from various distant areas of the hyperspace will have greater chances for recombining and producing lower performance offspring known as *lethals* (Deb and Goldberg 1989). The superfluous production of lethals, known as *dominance resistance*, will consequently spoil the optimisation process and the convergence towards the Pareto front. Ikeda *et al* (2001) were the pioneers to recognize the problem of dominance resistance followed by Deb, Thiele, Laumanns and Zitzler (2002). Another impact of the increasing number of redundant non-dominated solutions, the emphasis of diversity over convergence, and the dominance resistance problem, is the combined effect of the aforementioned observations on the selection for survival process. When attempting to downscale the size of the active archive to its pre-determined size by such means as the truncation procedure in SPEA2, *good* locally non-dominated solutions in terms of proximity towards the Pareto front might be filtered out at the expense of keeping *good* solutions in terms of diversity, but which may be distant from the Pareto front. Consequently the search process becomes dispersed in the wide objective space and evolves away from the Pareto front producing an 'over-diversified' approximation set with poor proximity to the tradeoff surface. This latter observation introduces an oscillatory behavior to the evolutionary search by repeatedly discovering and filtering out good solutions that might have been previously explored. This oscillatory behavior delays the convergence process towards the optimal tradeoff surface. As a result, as the number of competing objectives increase, the occurrence of the problem of *speciation* increases due to the combined effect of Pareto dominance based selection and the active diversity promotion mechanisms. At the end of the optimisation process, the probability of exclusively producing an approximation set with solutions excelling in a certain particular objective increases. This problem was originally present in Schaffer's VEGA, widely considered as the first approach for using EAs to solve multiobjective optimisation.

Having explained the motivations for the study, a new approach for diversity promotion in the *many*-objective optimisation frameworks is suggested, investigated and discussed in the next section.

4.3. An Adaptive Strategy for Diversity Promotion

4.3.1. The 'Memetic' Approach

In this Chapter, the requirement for promoting diversity in MOEAs is envisaged as a local, adaptive and varying requirement rather than a global necessity and is therefore investigated within a '*Memetic algorithm*' philosophy which connotes the acquisition of new proficiencies and the cultural and local adaptation to certain environments or communities.

The incorporation of exact local search techniques such as Hill Climbing, Simulated Annealing (Kirkpatrick, Gelatt and Vecchi 1983) or Tabu Search (Glover and Laguna 1997), as well as application specific and new local searcher, within MOEAs has been heavily investigated and this research area is still attracting increasing interest. However, the suggested local search techniques usually consist of iterative procedures that require additional objective function evaluations.

Adaptability, evolution and "self generation" within local search techniques have been proposed in (Krasnogor 2000, Krasnogor and Gustafson 2002, Krasnogor and Gustafson 2004, Krasnogor and Smith 2000 and Adra, Griffin and Fleming 2005a) and demonstrated to be beneficial for several domains. In Krasnogor and Smith (2000) an adaptive, two-purpose, Monte Carlo (MC) (Robert and Casella 2004) based local search was introduced and hybridized with a genetic algorithm. The local search adaptively modified the temperature variable; an essential parameter in simulated annealing methods used to control the acceptance process of newly discovered solutions based on a thermodynamics metaphor. This adaptive setting of the temperature was deployed to reflect the extent of the solutions' diversity in terms of their corresponding fitness values. The LS served two main purposes by operating in one of two modes; fine tuner mode when the current solutions from the GA are well distributed or diversifier mode when the distribution is poor.

In Adra, Griffin and Fleming (2005a), experiments were carried out to investigate new ways of addressing the diversity requirement in evolutionary multiobjective optimisation. In the first part of the study, three well-known techniques were hybridised as a local search with the Multi Objective Genetic Algorithm (Fonseca and Fleming 1993), a simple non-elitist evolutionary algorithm that uses fitness sharing for promoting diversity (Goldberd and Richardson 1987). The three local search algorithms consisted of a simulated annealing technique, a hill-climbing technique and a Tabu search technique. The three resulting hybrid algorithms were applied to a suite of bi-objective test functions introduced by (Zitler, Deb and Thiele 2000). The suite of test functions, commonly known as the ZDTs, comprises six test functions, each one of them presenting specific features that generally cause difficulties to major evolutionary optimisation strategies. The aim of the hybridizations was to investigate the effect of such local search techniques on the evolutionary multiobjective optimisation process in terms of solutions' convergence to the Pareto front, and particularly in terms of the solutions' diversity across the tradeoff surfaces.

The results achieved by the hybrid algorithms were compared with the results achieved by the standard MOGA. In order to provide a well-based comparison, these algorithms were balanced in terms of the number of objective function evaluations that were performed. Careful consideration should be taken when hybridizing an evolutionary algorithm with a local search technique. Excessive emphasis upon the local search at the expense of the evolutionary operators may result in the algorithm underachieving and even deteriorating the quality of the results. The experimental results showed that the hybrid "MOGA/Hill climbing local search" algorithm was particularly suitable for the ZDT problems and was outperforming the standalone MOGA and the MOGA hybridized with Simulated annealing or Tabu local search. These results were intuitive in the context of the ZDT problems due to the greedy behaviour of the hill climbing local search, which was the main cause behind the performance of that hybrid algorithm. On the other hand, despite their well recognized benefits, the extra functionality of simulated annealing (*escaping local optima*) and the Tabu search (*keeping a record of previously visited solutions and fighting oscillatory* *exploration behaviour*) were unnecessary adds-ons in the continuous domain of most of these bi-objective problems and consequently redundant computational effort was allocated to these local searches at the expense of the evolutionary global search.

The second part of the study (Adra, Griffin and Fleming 2005a) introduced a specialised local search technique which was hybridised with MOGA and used to optimise the ZDT test functions as well. The specialised local search incorporated the use of diversity indicators to set the size of the local search neighbourhood around the local front of solutions. When the spread of the solutions is damaged, i.e. the distribution of the solutions gets worse due to effects such as the genetic drift and the premature convergence leading to high concentration of solutions in certain areas of the search space and poor or null concentrations in other areas, the local search process extends the size of the current solutions' neighbourhood, thereby extending the search range in order to explore solutions in a wider neighbourhood. The process is reversed when the distributions of the solutions is good in order to fine tune the solutions in a more tightly constrained neighbourhood without detriment to the spread of solutions. Compared with the traditional results achieved by MOGA, the experimental results produced by the hybrid approach have shown an improved optimisation performance for some of the ZDT test functions and have demonstrated that an optimisation strategy might cope well with some features and might not be well tuned to deal with other features such as the shape of Pareto fronts.

The local search introduced in Adra, Griffin and Fleming (2005a) was later on extended to adapt the size of the local search neighbourhood for each individual candidate solution based on the current distribution of the solutions in objective space and the density estimate around the current individual under consideration (Adra, Griffin and Fleming 2006).

Simulations' results demonstrated further improvements of the performance of the hybrid algorithm relative to the MOGA. The improvements were observed in terms of the closeness of the solutions achieved by the hybrid optimisation technique to the true Pareto fronts of the ZDT test functions and in terms of distribution for most of the test functions. In particular, the adaptive hybrid algorithm coped well with challenging features such as multimodality and deception. The adaptive local search was also shown to provide higher suitability compared to the fitness sharing technique (Goldberg and Richardson 1987) used in that study (Adra, Griffin and Fleming 2006) for promoting solutions diversity for some of the test functions.

4.3.2. The Proposed Diversity Management Operator

Introduction

In this study, a diversity management operator (DMO) for controlling and promoting diversity in the *many*-objective (> 3 objectives) optimisation framework is introduced and hybridized with NSGA-II. NSGA-II is an evolutionary multiobjective optimiser which can be broadly regarded as a representative for a larger family of EMO optimisers. The DMO resembles the local search introduced in Krasnogor and Smith (2000) by serving the two essential purposes of fine-tuning and diversifying the population of solutions conveniently. However, the adaptive strategy introduced in this work drastically differs from Krasnogor and Smith technique in more than one aspect. The DMO is not based on MC or other exact methodologies. It is an adaptive strategy that promotes the integration of very effective performance and diversity indicators, such as the *hypervolume* metric (Zitzler 1999) and the *maximum spread* metric (Zitzler 1999) to efficiently guide the search process of an MOEA towards the tradeoff surface of a MOP while controlling the diversity requirement.

Despite the recognised utility of the hypervolume metric (Zitzler *et al* 2003) and its potential as a selection for survival strategy (Fleischer 2003, and Knowles and Corne 2003b) and as a solution ranking strategy replacing the Pareto dominance concept whose practicality is particularly debatable in the *many*-objective frameworks, the hypervolume metric suffers from some limitations (Zitzler et al 2003, and Knowles and Corne 2002). These limitations include the requirement for a sensitive choice of non-trivial reference parameters, the multiplication of potentially non-commensurable objectives and importantly the computational complexity of calculating the hypervolume metric, which is exponential in the number of objectives (Knowles 2002). The latter limitation has been believed to be addressed by Fleischer (2003) who has put forward an algorithm that should reduce the computational complexity of measuring the hypervolume metric to a polynomial magnitude. However, further research has later showed that the complexity of Fleischer's algorithm was actually exponential in the number of objectives (While 2005).

In this study which promotes the integration of performance metrics in the optimisation processes of MOEAs, the DMO has integrated a particular, computationally efficient, diversity metric which is based on the *maximum spread indicator* introduced by Zitzler (1999) and defined in Equation 4.1. Purshouse and Fleming (2003b) previously used the spread indicator in an exploratory study aimed at investigating the suitability of some classical settings of MOEA parameters for optimisation problems with more than 3 objectives.

In Equation 4.1, *D* represent the measure of the diagonal of the hypercube formed by the extreme objective values attained in a certain approximation set Z_A. *M* denotes the number of objectives and z_A is a candidate objective vector solution which belongs to the approximation set Z_A.

$$D = \left[\sum_{m=1}^{M} \left(\max_{z_A \in Z_A} \{ z_{A_m} \} - \min_{z_A \in Z_A} \{ z_{A_m} \} \right)^2 \right]^{1/2} \tag{4.1}$$

In a similar way to the study by Purshouse and Fleming (2003b), this investigative study experiments with different versions of the scalable DTLZ2 test function introduced by Deb et al (2002) and described in Section 3.3.6. The different versions of DTLZ2 deployed to test the performance of the suggested adaptive strategy vary in terms of the number of competing objectives to be optimised. DTLZ2, with its well-defined Pareto fronts, is a suitable test function for the theoretical analysis and the examination of the performance of new optimisation strategies and their interactive behaviour in the many optimisation contexts.
In order to track the diversity quality of the manipulated set of solutions, the value of the spread indicator presented in Equation 4.1 is normalised with respect to the optimal spread corresponding to the set of solutions representing the Pareto front of these test functions, or alternatively, representing a targeted reference front of solutions. It is only by knowing the normal and the desired condition, that the abnormal and the undesired conditions, such as the dispersal of solutions in suboptimal regions of the objective space or alternatively the convergence to contracted regions of a Pareto optimal space, can be defined and avoided. In other words, an application dependent scale defining the approximate notion of a *low*, *ideal*, *high* and *average* quality of diversity is required to overcome the "clash of the requirements" (convergence and diversity), which is specially evident in high dimensional problems (objective dimensionality).

In the context of the suggested DMO, the DM, usually and preferably an application expert, is only required to suggest an approximate estimate of the defining extremities of the *desired* tradeoff surface. These extremities will then serve as the vertices of the hypercube containing the ideally sought Pareto front. In an optimisation problem consisting of two conflicting objectives, these extremities will correspond to the coordinates (approximate objective values) of the two solutions presenting the best-expected performance in terms of one of the two objectives alongside the worst performance in terms of the competing objective.

Equation 4.1 will then be normalised with respect to the length of the diagonal of such a hypercube and the normalised diversity indicator will be defined by Equation 4.2.

$$I_S = D \Big/ \left[\sum_{m=1}^{M} \left(\max_{z_* \in Z_*} \left\{ z_{*_m} \right\} - \min_{z_* \in Z_*} \left\{ z_{*_m} \right\} \right)^2 \right]^{1/2} \tag{4.2}$$

The spread indicator (I_S) can take any positive real value. Ideally an indicator value close to unity ($I_S = 1$) is sought. Indicator values smaller than one ($I_S < 1$) point out a lack of diversity among the solutions manipulated compared to the desired spread of solutions. This is most likely due to a convergence towards a contracted area - potentially a Pareto optimal sub region - of the solution space. On the other hand, indicator values larger than one ($I_S > 1$) highlight an excessive dispersal of the solutions in the objective space. This kind of excessive dispersal in the hyperspace most likely causes the divergence of the solutions from the Pareto optimal front and hampers the optimisation process by introducing an oscillatory behaviour forcing the MOEA to repeatedly explore previously visited regions of the space.

DMO Functionality

A schematic presentation of the diversity management operator (DMO) is illustrated respectively in Figure 4.2 within the context of NSGA-II. The DMO is composed of two main steps:

Step 1:

Step 1 of DMO is illustrated below:

1. **Calculate the spread indicator I_s for the current approximation set at generation i**
   a. **If** $I_s < 1\text{-}\varepsilon$
      Activate the diversity promotion mechanism in the selection for variation and the selection for survival process
   b. **Else If** $I_s \geq 1\text{-}\varepsilon$
      Deactivate the diversity promotion mechanism in the selection for variation and the selection for survival process

The first step consists of calculating the spread indicator defined in Equation 4.2 for the local front of non-dominated solutions achieved. The calculation of the spread indicator takes place at every generation of the optimisation process prior to the execution of the genetic operators (selection for variation and selection for survival).

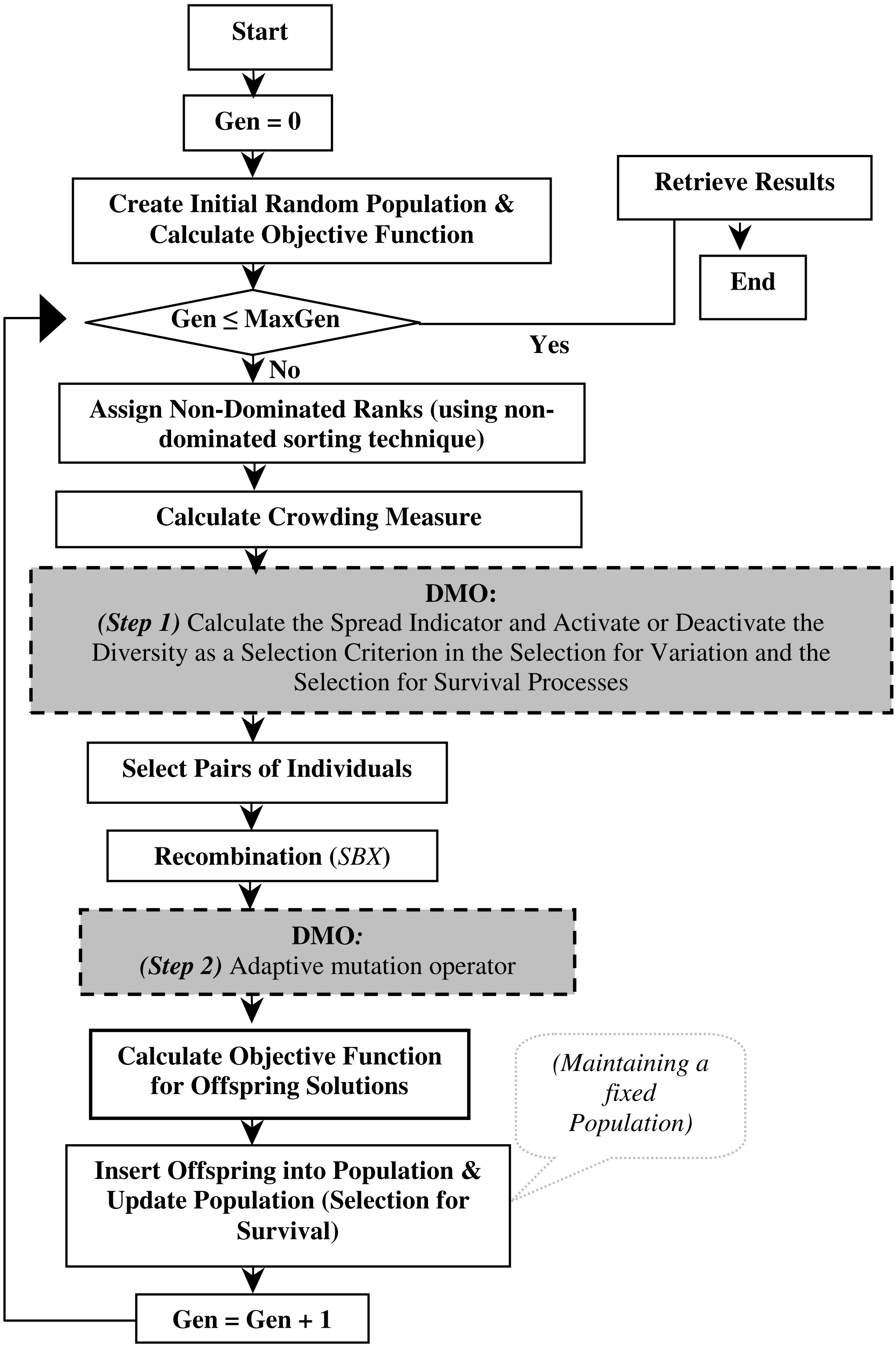


Figure 4. 2 NSGA-II with the addition of the DMO steps

The DMO then adjusts and controls the global search processes of the MOEA in an informative way based on the local level of spread optimality. In other words, if the spread indicator reports an excessive dispersal of the local front of solutions in the objective space (i.e. $I_s \geq 1\text{-}\varepsilon$, ε being an optional –application dependent- tolerance value defined by the

DM), the DMO switches off the diversification mechanisms within the subsequent selection for variation and selection for survival procedures. The goal is to maintain the optimal tradeoff between the convergence and diversity requirements. In NSGA-II context, at the selection for variation stage, deploying the binary tournament selection procedure, two candidate solutions are picked randomly and compared in terms of their Pareto dominance rank. The solution with the highest Pareto dominance rank is inserted in the mating pool. In the case where the two solutions share the same Pareto dominance rank, one of the two solutions is chosen randomly and included in the mating pool, disregarding the NSGA-II crowding measure which usually constitutes the secondary criterion for selection for variation. The selection for variation process continues until the mating pool is filled.

At the selection for survival stage, the diversity measure is again eliminated as a discriminatory criterion for selection. In the situation where the number of locally non-dominated solutions exceeds the prefixed size of the active archive, the solutions are selected randomly for survival and propagation to the succeeding generation of the optimisation process.

On the other hand, when required (i.e. $I_s < 1\text{-}\varepsilon$), the diversity promotion mechanisms are automatically activated in the selection for variation and the selection for survival procedures based on the diversity indicator monitoring the diversity of the locally non-dominated solutions.

Step 2:

The second step of the DMO consists of a novel mutation operator designed to ensure the usual explorative and diversity preserving benefits of the mutation process in evolutionary algorithms. Moreover, the new mutation operator is designed to take into consideration the diversity requirement to minimise the problem of having the mutation process causing a deterioration by uncovering, *in a high dimensional search space*, a non-dominated solution which is remote from the Pareto front and which can hinder the search process. The introduced mutation operator can be compared with Deb and Goyal’s (1996) polynomial mutation to some extent. The polynomial mutation enjoys a good reputation tackling real-valued optimisation problems and has been deployed within several studies solving the DTLZ2 test function used in this study (Deb, Thiele, Laumanns and Zitzler 2002, Khare et al 2003 and Purshouse and Fleming 2003b).

The *polynomial mutation* is presented in Equation 4.3, where $\mathbf{O_i}$ is the mutated (or the offspring) value of the i^{th} decision variable of a certain candidate solution $\mathbf{S_A}$ in an approximation set of decision vectors **A**. In Equation 4.3, $\mathbf{P_i}$ is the original (or the parent) value of the i^{th} decision variable of $\mathbf{S_A}$. $\mathbf{U_i}$ and $\mathbf{L_i}$ denote respectively the upper and the lower bounds of the i^{th} decision variable, $\boldsymbol{\eta_m}$ is the mutation distribution parameter and $\mathbf{r_i}$ is a random number generated uniformly from the range [0, 1].

$$\left.\begin{aligned} O_i &= \begin{cases} P_i + (P_i - L_i)\delta_i & \text{if } r_i < 0.5 \\ P_i + (U_i - P_i)\delta_i & \text{otherwise} \end{cases} \\ \delta_i &= \begin{cases} (2r_i)^{1/(\eta_m+1)-1} & \text{if } r_i < 0.5 \\ 1 - [2(1-r_i)]^{1/(\eta_m+1)} & \text{otherwise} \end{cases} \end{aligned}\right\} \quad (4.3)$$

Using the polynomial mutation operator, the magnitude of the normalised variation of a certain decision variable is inversely proportional to the mutation distribution parameter $\boldsymbol{\eta_m}$ whose value is set at the start of the optimisation process. A common setting for the mutation distribution parameter is $\boldsymbol{\eta_m}$ = 20 alongside a mutation probability of $\mathbf{p_m = 1/n}$ for each individual decision variable, where **n** is the total number of decision variables. The previous settings for the polynomial mutation were deployed in studies such as Deb, Thiele, Laumanns and Ziztler (2002) and Khare, Xao and Deb (2003).

The degree of similarity between the DMO's mutation operator and the polynomial mutation is that they both integrate a control process for the mutation magnitude of a certain candidate solution. However, in a manner which is different from the polynomial mutation which requires a fixed mutation distribution parameter ($\boldsymbol{\eta_m}$) to control the magnitude of the expected mutation of a certain decision variable, the mutation operator of the DMO adapts the magnitude of the expected mutation for a certain decision variable based on two complementary criteria:

(1) the value of the spread indicator (I_s) measuring the relative '*global*' diversity worth of the locally non-dominated set of solutions and
(2) the NSGA-II 'crowding' measure which highlights the '*local*' diversity worth of each single solution separately.

DMO Mutation Operator

The mutation operator of the DMO, illustrated in Figure 4.3, receives its input in the form of the decision variable vectors of the whole population of solutions. These decision variables vector are produced by the selection for variation process from the active archive, containing the set of locally non-dominated solutions, followed by the recombination process. In this study the *simulated binary crossover* (SBX) (Deb and Agrawal, 1995), a popular recombination operator for real coded optimisation tasks, is used.

The mutation operator then adaptively determines the range of mutation for each solution in the population. More precisely, the boundaries of the mutation range for each decision variable are locally, and adaptively, determined in a two-step process and centred over the value of each decision variable chosen to undertake mutation. After determining the local range of mutation for each decision variable of each candidate solutions, the third step then consists of perturbing the decision variables chosen for mutation within their pre-determined local range of mutation.

Determining the global range of mutation for the whole current approximation set

1. At a certain generation '**gen**' of the evolutionary optimisation process, a *global* mutation range, '$\mathbf{R_{gen}}$', for the whole set of locally non-dominated solutions is first determined based on the value of the spread indicator presented in Equation 4.2. This process is illustrated by the first step in Figure 4.3. The spread indicator $\mathbf{I_s}$ is a positive real number whose minimum value is zero and optimal value is 1. At a certain generation of the optimisation process, the value of the spread indicator is normalised in the range [**0.01, 1**] to represent the global mutation range ($\mathbf{R_{gen}}$) for current approximation set of solutions. $\mathbf{R_{gen}}$ defines a certain percentage of the total range of definition [$\mathbf{L_i, U_i}$] of every decision variable $\mathbf{dv_i}$.
    a. A value of $\mathbf{I_s}$ tending to *zero* highlights a very poor diversity quality, and will correspond to the value **1** in the normalisation range [**0.01, 1**], therefore articulating the request for the highest level of mutation possible.
    b. On the other hand, for $\mathbf{I_s}$ values increasing from **0.9** (i.e. $< 1\text{-}\varepsilon$, in here $\varepsilon = 0.1$) (to any positive number), the global mutation range $\mathbf{R_{gen}}$ will tend to the value **0.01** to reduce the amplitude of the mutation process.

Refining the global range of mutation by separately determining local ranges of mutation for each candidate solution in the current approximation set

2. After having determined the length of the global range of mutation for the set of locally non-dominated solutions, customised local ranges of mutation $\mathbf{R_{gen}(z_A)}$ for each single solution $\mathbf{z_A}$ are then set based on the crowding measure around each solution (the second step in Figure 4.3).
    a. In the finite set of locally non-dominated solutions, the crowding measure for each solution is calculated.
    b. These crowding values are then normalised in the range [$\mathbf{0.01, R_{gen}}$], with the minimum local crowding measure mapped to $\mathbf{R_{gen}}$ and the maximum local crowding measure mapped to **0.01**.

This is conveying the request for higher mutation magnitudes for the crowded solutions and vice versa. The normalised values of the crowding measures define the amplitude of the mutation range for each decision variable of each candidate solution. The determined local mutation ranges are then centred over the values of the pre-mutated decision variables for each solution. This is illustrated by the third step in Figure 4.3.

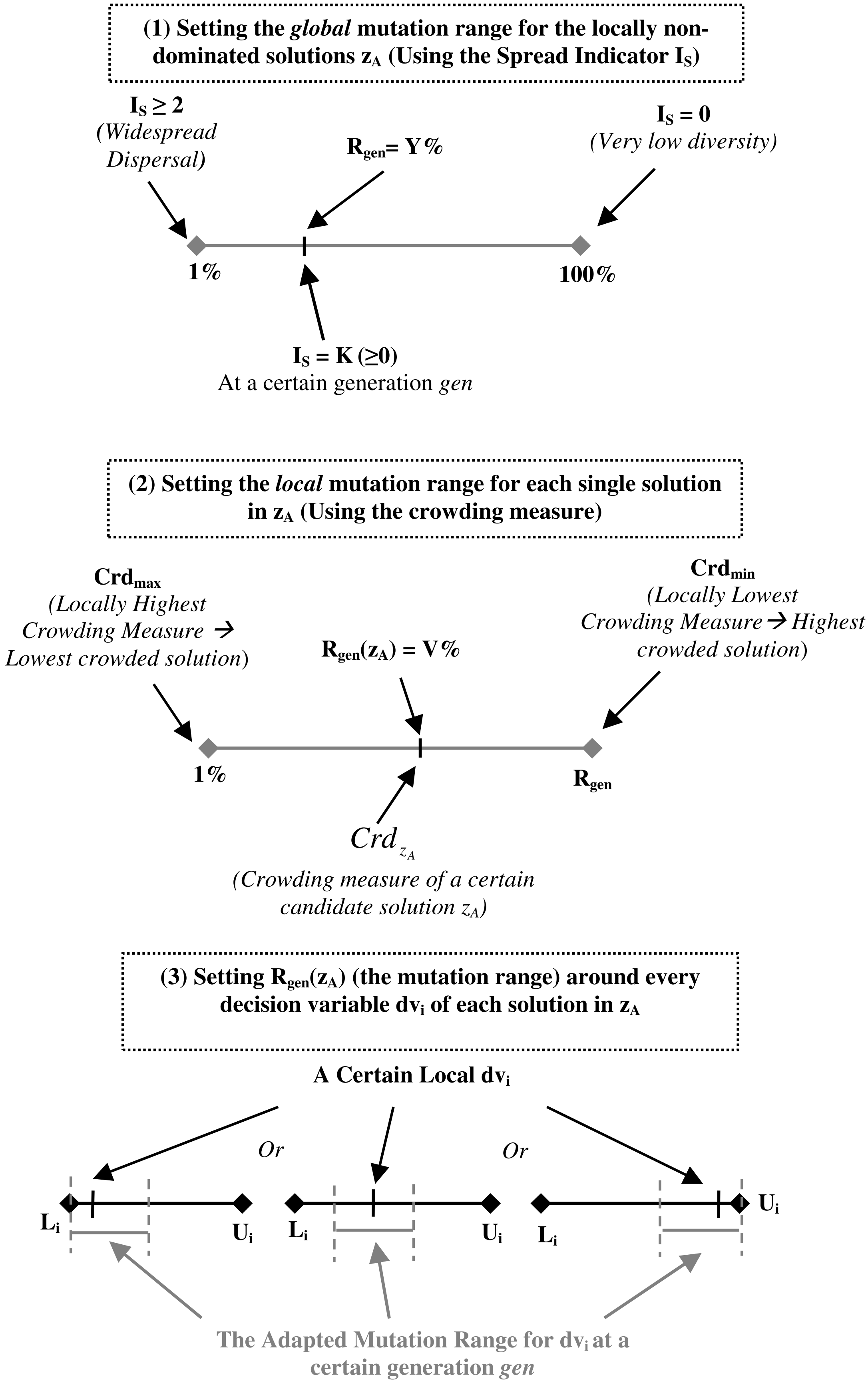


Figure 4. 3 DMO: The Adaptive Mutation

Mutating the decision variables in their defined local mutation range

3. After defining the local mutation range for each decision variable of all the candidate solutions, the value of a certain decision variable chosen to undertake mutation is perturbed randomly within its determined local mutation range. Based on a uniform two-sided coin flip probability, the new value for a mutation subjected decision variable is randomly interpolated in one of the 2 ranges:

[local_lower_mutation_boundary**,** current_value) or

(current_value, local_upper_mutation_boundary]

The DMO mutation operator is applied uniformly to every decision variable with a probability $\mathbf{p_m=1/n}$, where **n** is the total number of decision variables. This is a standard probability for uniform mutation operators and is used in studies deploying the polynomial mutation for solving real-coded multiobjective optimisation problems such as Deb, Thiele, Laumanns and Zitzler (2002) and Khare et al (2003). The setting $\mathbf{p_m=1/n}$ in the context of the DMO, which uses its own adaptive distribution parameter, conforms to the findings by Purshouse and Fleming (2003b). Purshouse and Fleming (2003b) showed that despite being unsuitable for optimisation problems with many competing objectives when combined with the prevalent settings of the distribution parameter in the polynomial mutation context, the setting $\mathbf{p_m=1/n}$ was a good choice of mutation probability (in the context of the DTLZ2 test functions) for a wide range of distribution parameters (in terms of solutions convergence and distribution).

In Figures 4.4 a pseudocode description of the proposed DMO within the context of NSGA-II is presented. A pseudocode description of the DMO mutation operator (i.e. step 2) is also given in Figure 4.5.

In the following section, the experimental framework used to test the performance of the DMO is presented. This will include a summary of the different optimisation problems used, the configuration of the optimisers and the different performance metrics utilised for assessing the utility of the DMO.

Test Functions, Configurations and Performance Metrics

Test Functions and Optimiser Configurations

The DMO is incorporated in the context of NSGA-II. The resulting optimisation strategy will be referred to as NSGA-II/DMO. Four different versions of the scalable DTLZ2 test function, featuring 6, 8, 12 and 20-objective optimisation problems, are used to assess the performance of NSGA-II/DMO.

-Initialize random population **P** [15]

-Evaluate the objective values of **P** and store them in **A**[16]

For gen = 1 to Max_Gen

-Assign ranks to the solutions in **P** using non-dominated sorting strategy

-Determine the crowding distance Crd_{Z_A} for each solution Z_A in **P**

-Generate offspring population **Q** from **P**– size **Nind**

-DMO Step 1:

-Calculate $\mathbf{I_s}$ (Equation 4.2) for the local Pareto front [17] presented in **A**

-If $\mathbf{I_s}$ < 1-ε [18]

-Binary tournament selection (Standard Tie Breaking – diversity promotion active)

-Else

-Binary tournament selection (Random Tie Breaking – diversity promotion inactive)

-Recombination

-**DMO Step 2:**

Adaptive Mutation Operator *(described in Figure 4.4)*

-Evaluate objective values for the offspring population **Q**

-Combine parent population **P** and offspring Population **Q** – size: **2*Nind**

-Assign ranks to the combined population using non-dominated sorting strategy

-Determine the crowding distance for each solution in the combined population

-DMO Step 1 continued:

-If $\mathbf{I_s}$ < 1-ε

Select **Nind** solutions to propagate to the next generation (Selection Criteria: 1st: elitist -biased towards lower ranks- 2nd: crowding distance - bias less crowded solutions – diversity promotion active)

-Else

Select **Nind** solutions to propagate to the next generation (Selection Criteria: 1st: elitist -biased towards lower ranks- 2nd: random – diversity promotion inactive)

End loop

Figure 4. 4 Pseudocode of NSGA-II with the addition of the DMO steps

[15] 'P' is a 2D array of decision variables and its dimension is Nind x Nvar, where Nind is the size of the population of candidate solutions and Nvar is the number of decision variables.

[16] 'A' is a 2D array of objective values and its dimension is Nind x Nobj, where Nobj is the number of objectives being optimised.

[17] For the DTLZ2 test function used in this study, Is is normalized with respect to the true Pareto front, i.e. the minimum and maximum values for each objective (z_m^* in equation 4.2) are respectively 0 and 1.

[18] ε = 0.1 in this study.

/* Calculate the global mutation range R_{gen}: */[19]

If $I_s > 2$

$R_{gen} = 0.01$

Else

R_{gen}[20] $= I_s \times \frac{0.01-1}{2-0} + \frac{1\times 2 - 0.01\times 0}{2-0}$

/* Find the minimum and maximum crowding values Crd */

/* Crd_{min} corresponds to the most crowded solution in A at generation 'gen' */

/* Crd_{max} corresponds to the least crowded solution in A at generation 'gen' */

$Crd_{\min} = \min Crd_{Z_A} : Z_A \in A_{gen}$,

$Crd_{\max} = \max Crd_{Z_A} : Z_A \in A_{gen}$,

For each solution $\mathbf{Z_A}$ in **P**

$$R_{gen}(Z_A) = Crd_{Z_A} \times \frac{0.01 - R_{gen}}{Crd_{\min} - Crd_{\max}} + \frac{R_{gen} \times Crd_{\min} - 0.01 \times Crd_{\max}}{Crd_{\min} - Crd_{\max}}$$

For each decision variable $dv_{i=1}^{\mathrm{Nvar}}$ in $\mathbf{Z_A}$

If $r_1 \leq 1/\mathrm{Nvar}$[21]

/* Set the magnitude of the local mutation range R_{mut} */

$R_{mut} = R_{gen}(Z_A) \times (U_i - L_i)$

/* Set the lower boundary of the local mutation range */

If $(dv_i - R_{mut}/2) < L_i$

$L_{mut}(dv_i) = L_i$

Else

$L_{mut}(dv_i) = dv_i - R_{mut}/2$

/* Set the lower boundary of the local mutation range */

If $(dv_i + R_{mut}/2) > U_i$

$U_{mut}(dv_i) = U_i$

Else

$U_{mut}(dv_i) = dv_i + R_{mut}/2$

/* Mutate the decision variable dv_i */

$dv_{i_{mut}} = L_{mut}(dv_i) + (U_{mut}(dv_i) - L_{mut}(dv_i)) \times r_2$

End If

End For

End For

Figure 4. 5 Pseudocode of the DMO step 2 (The Adaptive Mutation)

[19] /*This is a comment*/

[20] R_{gen} and R_{gen} (Z_A) are calculated using the data scaling formula:

$$Y = X \times \frac{Y_{\max} - Y_{\min}}{X_{\max} - X_{\min}} + \frac{X_{\min} \times Y_{\max} - X_{\max} \times Y_{\min}}{X_{\max} - X_{\min}}$$

where Ymax and Ymin are the boundaries of the new normalisation range, and Xmax and Xmin are the boundaries of the range of the original data. Note that the values of Is and Crd_{Z_A} are normalised in the reverse order within the ranges [0.01, 1] and [0.01, R_{gen}] respectively.

[21] r_1 and r_2 are random numbers drawn from a uniform distribution on the unit interval.

These four instances of DTLZ2 will be referred to as DTLZ2 (6), DTLZ2 (8), DTLZ2 (12) and DTLZ (20) respectively. Some of the DTLZ2 versions used in this chapter (DTLZ2 (6) and DTLZ2 (12)) were investigated by Purshouse and Fleming (2003b) in their study which identified that despite being suitable for multiobjective optimisation problems with a reduced number of objectives (no more than 3), the classical and widespread parameter settings of the simulated binary crossover (p_{ic} and η_c) were not suitable for a higher number of objectives.

Similar to Pusrhouse and Fleming's study, the *simulated binary crossover (SBX)*, a two-parent crossover operator that produces two new solutions, is used in this study as well. SBX is illustrated in Equation 4.4, where $\mathbf{O_{i,1}}$ and $\mathbf{O_{i,2}}$ are the two offspring values for the i^{th} decision variable whose parent values are $\mathbf{P_{i,1}}$ and $\mathbf{P_{i,2}}$, $\boldsymbol{\eta_c}$ is a distribution parameter and $\mathbf{r_i}$ is a random number generated uniformly from [0, 1]. Similar to the study by Purshouse and Fleming, in this work, each decision variable dv_i is independently considered for undertaking the variation operator. The probability of uniformly applying the variation operator on a certain decision variable, $\mathbf{p_{ic}}$, is commonly set to a value of 0.5 alongside a distribution parameter value $\boldsymbol{\eta_c} = \mathbf{15}$ (Deb, Thiele, Laumanns and Zitzler 2002) and a probability of applying variation to a certain pair of solutions $\mathbf{p_c = 1}$.

In Purshouse (2004) and Purshouse and Fleming (2003b) these settings were shown to be convenient for optimising the DTLZ2 (3) but inappropriate for the DTLZ2 (6) and DTLZ2 (12) problems, in terms of the resulting convergence of the produced results towards the Pareto fronts of these test functions, as well as in terms of the solutions diversity.
In this study, two scenarios deploying two different settings for the SBX operator are investigated to assess the performance of the DMO when operating in the NSGA-II framework and optimising each version of the DTLZ2 test function. The first scenario consisted of the SBX operator configured with standard parameters usually used in the EMO community ($\boldsymbol{\eta_c} = \mathbf{15}$, $\mathbf{p_{ic}= 0.5}$ and $\mathbf{p_c = 1}$) (Deb, Thiele, Laumanns and Zitzler 2002, Khare et al 2003). Additionally, based on the findings by Purshouse and Fleming (2003b) and Purshouse (2004), the second scenario for each of the DTLZ2 functions consisted of well-chosen parameters for the SBX operator and which were expected to be more suitable for each of the DTLZ2 functions investigated. These two scenarios were intended to investigate the impact of the DMO and quantify the level of improvement or deterioration that the strategy might introduce when operating in the standard or the informed (optimised) configurations. In the NSGA-II context, the polynomial mutation operator was used and configured with the standard parameters for each of the DTLZ2 functions.

$$\left.\begin{aligned} O_{1,\,i} &= 0.5\ \left[(1+\beta_i)P_{1,i} + (1-\beta_i)P_{2,i}\right] \\ O_{2,\,i} &= 0.5\ \left[(1-\beta_i)P_{1,i} + (1+\beta_i)P_{2,i}\right] \\ \beta_i &= \begin{cases} (2r_i)^{1/(\eta_c+1)} & \text{if } r_i < 0.5 \\ [1/(1-r_i)]^{1/(\eta_c+1)} & \text{otherwise} \end{cases} \end{aligned}\right\} \quad \textbf{(4.4)}$$

The configuration of NSGA-II and NSGA-II/DMO is presented in Table 4.1 below. The two algorithms were balanced in terms of the number of objective function evaluations.

Performance Metrics

The performance of the NSGA-II/DMO is compared with the results achieved by NSGA-II for each of the four versions of DTLZ2 deployed. In order to make a rigorous comparison of the two optimisers and an accurate judgment over their performances, NSGA-II/DMO and NSGA-II were each executed 10 times and their produced results were compared at every execution. The DD-metric (Zitzler 1999), which computes the difference between the dominated distances of two approximation sets, is one of the two binary performance metrics used to assess the quality of the approximation sets achieved by NSGA-II/DMO and NSGA-II. The other binary metric deployed is the C-metric, previously described in Chapter 4. The previous two metrics were intended to highlight the quality of the approximation sets achieved by NSGA-II/DMO and NSGA-II in terms of the Pareto dominance concept.

However, in the *many*-objective optimisation scenarios, the sole usage of the Pareto dominance concept to determine whether an optimiser A is better than optimiser B can be very misleading. In fact, the approximation sets achieved by both such optimisers A and B might be deemed non-dominated to each other and equally good if one of the approximation sets uniquely included non-dominated solutions lying on a contracted region close to the Pareto front, while the other approximation set exclusively contained non-dominated solutions lying around single axes of performances (in a high dimensional objective space) but which are remote from the Pareto front. It is therefore necessary to assess the proximity of the solutions achieved by an optimisation strategy towards the Pareto front and their diversity across the region of interest (in this study the ROI is the whole Pareto front) simultaneously. Hence, the normalised maximum spread metric (Equation 4.2) was used to measure the performance of the two optimisers in terms of the diversity quality of their produced results.

Table 4. 1 Optimiser Configuration

<table>
<tr><td>Size of Population</td><td colspan="7">NSGA-II: 100
NSGA-II/DMO: 100</td></tr>
<tr><td rowspan="7">Crossover operator</td><td colspan="7">NSGA-II: SBX
NSGA-II/DMO: SBX</td></tr>
<tr><td rowspan="2"></td><td colspan="3">Standard Configuration</td><td colspan="3">Informed Configuration</td></tr>
<tr><td>η_c</td><td>p_c</td><td>P_{ic}</td><td>η_c</td><td>p_c</td><td>P_{ic}</td></tr>
<tr><td>DTLZ2 (6)</td><td>15</td><td>1</td><td>0.5</td><td>15</td><td>1</td><td>0.1</td></tr>
<tr><td>DTLZ2 (8)</td><td>15</td><td>1</td><td>0.5</td><td>10</td><td>1</td><td>0.1</td></tr>
<tr><td>DTLZ2 (12)</td><td>15</td><td>1</td><td>0.5</td><td>10</td><td>1</td><td>0.05</td></tr>
<tr><td>DTLZ2 (20)</td><td>15</td><td>1</td><td>0.5</td><td>5</td><td>1</td><td>0.01</td></tr>
<tr><td rowspan="7">Mutation Operator</td><td colspan="7">NSGA-II: Polynomial Mutation
Probability: 1/(number of Decision Variables)</td></tr>
<tr><td></td><td colspan="3">η_m</td><td colspan="3">p_m</td></tr>
<tr><td>DTLZ2 (6)</td><td colspan="3">20</td><td colspan="3">≈ 0.06</td></tr>
<tr><td>DTLZ2 (8)</td><td colspan="3">20</td><td colspan="3">≈ 0.05</td></tr>
<tr><td>DTLZ2 (12)</td><td colspan="3">20</td><td colspan="3">≈ 0.04</td></tr>
<tr><td>DTLZ2 (20)</td><td colspan="3">20</td><td colspan="3">≈ 0.03</td></tr>
<tr><td colspan="7">NSGA-II/DMO:
DMO Adaptive Mutation
Probability: 1/(number of Decision Variables)</td></tr>
<tr><td>Number of generations/ Run</td><td colspan="7">200</td></tr>
<tr><td>Number of Runs</td><td colspan="7">10</td></tr>
</table>

The ideal diversity measure sought was $\mathbf{I_s}$ =1 and which represents an intermediate, uniform, spread measure between the two extreme situations:

(1) Dispersal of solutions in sub-optimal regions of the objective space, and

(2) Contracted diversity in a possible sub region of the Pareto front.

On the other hand, the convergence quality of the achieved approximation sets is assessed in terms of their proximity to the well-defined Pareto fronts (k-dimensional quadratic bowl, k is defined in Section 3.3.6) of the DTLZ2 test function. A specialised proximity metric for DTLZ2 is used to measure the median proximity of the achieved approximation sets (Z_A) to the Pareto fronts of each of the DTLZ2 versions investigated. The proximity metric, presented in Equations 4.6, is none other the generational distance (GD) metric (Veldhuizen 1999) for the case of a continuous Pareto optimal reference set Z_*.

$$1 \leq \left[\sum_{m=1}^{M} (z_m)^2 \right]^{1/2} \qquad \textbf{(4.5)}$$

$$GD = median_{z_A \in Z_A} \left\{ \left[\sum_{m=1}^{M} (z_{A_m})^2 \right]^{1/2} - 1 \right\} \qquad \textbf{(4.6)}$$

The equality in Equation 4.5 (where M is the number of objectives) only holds when the objective vector **z** is a Pareto optimal solution.

Finally, in order to illustrate the performance of the two optimisers in terms of the desired requirements –convergence and diversity- the *Non-Dominated Evaluation* metric (Deb 2001) was used to simultaneously visualise the performance of the optimisers in terms of proximity to the Pareto front and in terms of diversity. The spread metric and the generational distance metric were posed as two objective functions evaluating two competing objectives: *Objective 1: Convergence* and *Objective 2: Diversity*. The problem can then be formulated as a two-objective optimisation scenario optimising (minimising) these two objectives. As a result, the performance of an optimiser **A** would be confidently deemed superior to the performance of another optimiser **B** if its approximation set to the posed bi-objective optimisation problem dominates the approximation set achieved by **B**. The Non-dominated evaluation metric is illustrated in Figure 4.6 where it can be inferred that optimiser **A** outperforms optimiser **B** in terms of convergence and diversity but it cannot be concluded that **A** outperforms **C**.

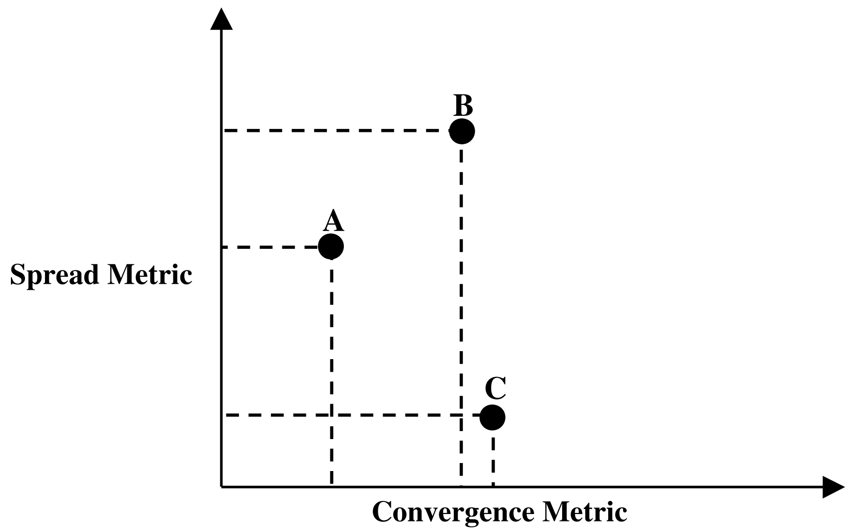


Figure 4. 6. The Non-Dominated Evaluation Metric

In the following section, the results achieved by NSGA-II/DMO are illustrated and compared with the results achieved by NSGA-II for each of the test functions investigated. Conclusions and a summary of the findings are given in the final section of the chapter.

4.4. Results

The adaptive strategy for controlling and promoting diversity (DMO) is composed of two components previously illustrated in Section 4.3.2. The DMO is assessed within two sets of experiments consisting of *many*-objective optimisation scenarios.

The first set of experiments was aimed at evaluating the performance of the suggested strategy (DMO) by measuring its impact on the optimisation process. The final approximation sets achieved by NSGA-II/DMO and NSGA-II for each of the DTLZ2 test functions after 200 generations are contrasted. Each optimiser was executed 10 times in order to assess the significance of the observed results and to make sure that the observations have not arisen by chance.

The second set of experiments was aimed at assessing the second component of the DMO in particular. The second component consisted of an adaptive mutation operator replacing the polynomial mutation. Consequently, a standard NSGA-II optimiser employing the polynomial mutation operator was executed 10 times (200 generations per execution) to optimise the DTLZ2 test functions used in this study. The quality (convergence and diversity) of the approximation sets achieved by NSGA-II for each test function was compared with the quality of the approximation sets achieved by a slightly modified version of NSGA-II using the DMO's mutation operator instead of the polynomial mutation.

In Table 4.2, the DD-metric results and the C-metric results achieved by NSGA-II/DMO and NSGA-II for DTLZ2 (6) are illustrated. The two optimisers were similarly configured with the standard SBX parameters usually used in the EMO community. The C-metric is not a symmetric indicator, and therefore in order to have an informative idea about the relative quality of two approximation sets, the metric had to be executed twice, switching the order of its input.

Table 4. 2. DD-metric and C-metric results for DTLZ2 (6)
A = NSGA-II/DMO and B = NSGA-II (Standard SBX configuration)

	DTLZ2 (6)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1	0.34	0	-0.65711
2	0.35	0	-0.58455
3	0.5	0	-0.85758
4	0.66	0	-1.3793
5	0.54	0	-1.017
6	0.47	0	-0.73012
7	0.44	0	-0.70773
8	0.46	0	-0.88246
9	0.47	0	-0.88319
10	0.51	0	-0.81869
Mean Value:	0.4740	0	-0.8518

Form Table 4.2, it can deduced that the approximation sets achieved by NSGA-II/DMO were covering an average of 47% of the solutions achieved by NSGA-II over the 10 runs.

On the other hand, NSGA-II was repeatedly achieving nil coverage of the solutions achieved by its DMO hybridized counterpart.

The dominated distance metric was uniformly producing results that highlight the superior quality of the approximation sets achieved by NSGA-II/DMO. A negative DD-metric value denotes that the first input of the metric (*e.g. Algorithm A in DD-Metric (A, B)*) produced an approximation set which is overall better than and dominates most or part of the approximation set produced by its second input (*Algorithm B*). Note that the results of the two metrics highlighted in Table 4.2 are measured based on the Pareto dominance concept which constitutes the underlying quality criterion for both binary metrics. In Figure 4.7, the black circles, whose (x, y) coordinates are the values of the GD-metric and the maximum spread indicator respectively, represent the values of the non-dominated evaluation metric achieved by NSGA-II/DMO at each of the 10 executions of the optimiser solving DTLZ2 (6) and using a standard SBX configuration. The values of the non-dominated evaluation metric achieved by NSGA-II for the same optimisation scenario are represented by the red squares. The values of the GD metric achieved by NSGA-II/DMO over the 10 runs were constantly lower than 0.1 and were accompanied by spread measures with an approximate ceiling value of 2.

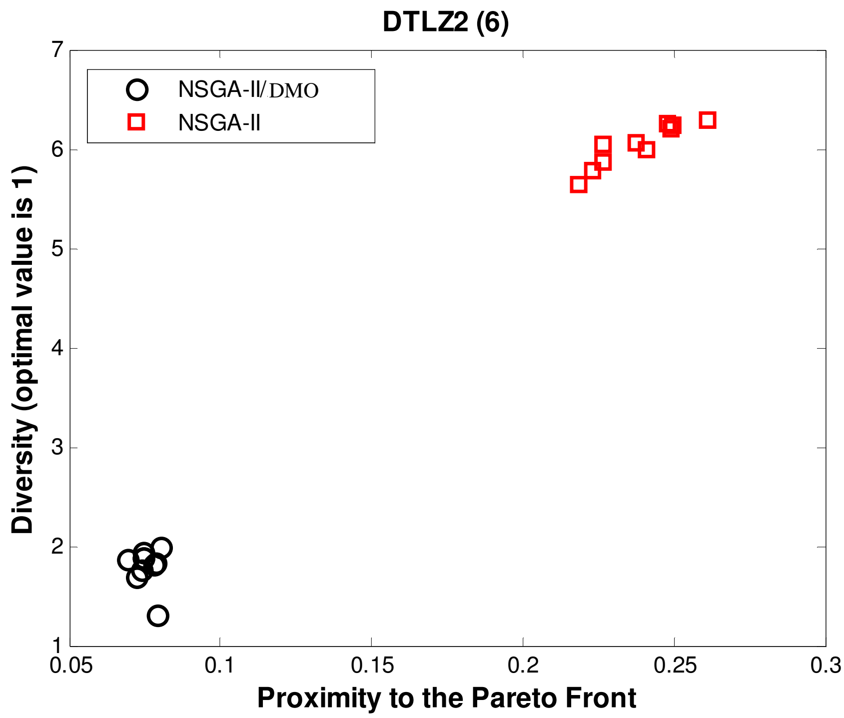


Figure 4. 7. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (6) (Standard SBX configuration)

The observed results highlight good proximity to the Pareto front alongside a simultaneous near-optimal diversity[22]. On the other hand, the GD values achieved by NSGA-II over the 10 runs constantly exceeded the value 0.2 alongside an average diversity measure of 6. From the results illustrated in Figure 4.7, it was very clear that the performance of NSGA-II/DMO was superior to the performance of NSGA-II in terms of both requirements (convergence to the Pareto front and desired diversity).

Table 4. 3. DD-metric and C-metric results for DTLZ2 (6)

A = NSGA-II/DMO and B = NSGA-II (Informed SBX configuration)

	DTLZ2 (6)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1	0.03	0	-0.0090581
2	0.04	0	-0.016966
3	0	0	0
4	0.04	0	-0.018131
5	0.03	0	-0.0089042
6	0	0	0
7	0.04	0	-0.018709
8	0.02	0	-0.010254
9	0.02	0	-0.0072117
10	0.03	0	-0.013414
Mean Value:	0.0250	0	-0.0103

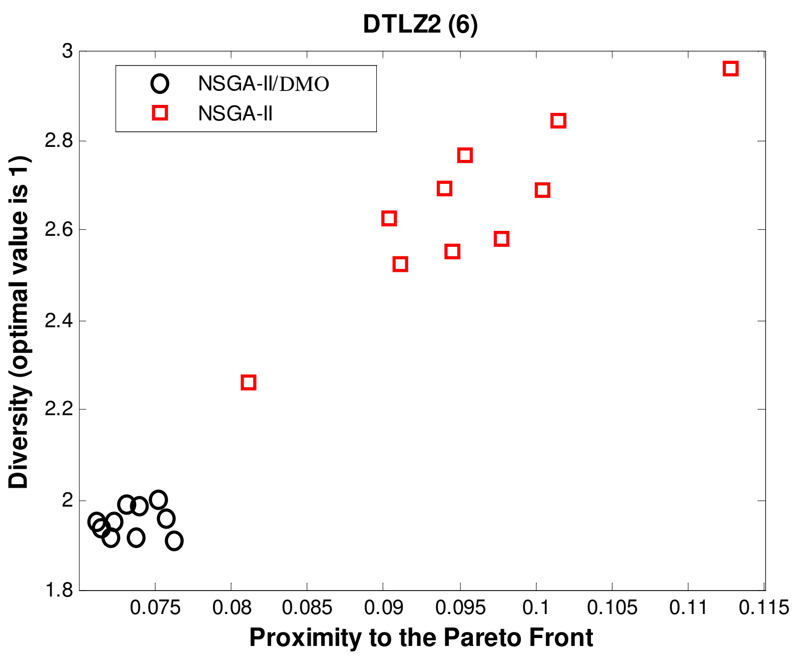


Figure 4. 8. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (6) (Informed SBX configuration)

---

22 Optimal in terms of diversity extent. Future research will look at optimising the uniformity of the diversity (e.g. using entropy based metrics) while controlling the diversity extent using the DMO.

The results achieved by NSGA-II were much more diverse in terms of dispersal in the 6-dimensional objective space. Nevertheless, *absolute* diversity, which can be achieved using a completely random search process, is undesirable in many-objective optimisation scenarios and once again led to the deterioration of the convergence of the search process towards the optimal regions of the space.

In Table 4.3, the results of the C-metric and the DD-metric are highlighted for the same DTLZ2 (6) test function. However, an informed setting for the SBX operator was used, based on the findings in (Purshouse 2004). The observed results showed an improved performance of the NSGA-II conforming to the findings by Purshouse. This is highlighted by the reduction of the average value of the C-metric and the DD-metric achieved by NSGA-II/DMO over the 10 runs. Nevertheless, the binary metric values were still favouring NSGA-II/DMO. In Figure 4.8, a dramatic improvement in the performance of NSGA-II for the 6-objective problem was observed. The enhanced performance of NSGA-II was due to the decrease in the level of exploration manifested by the optimised setting of the SBX operator ($\mathbf{p_e}$ is reduced from 0.5 to 0.1). NSGA-II/DMO on the other hand was less susceptible to the SBX settings for the same test function, but retained a significantly superior performance in terms of both criteria (Figure 4.8). The immunity of the NSGA-II/DMO against the setting modification of the SBX operator, in terms of the convergence and the diversity criteria, was quite understandable and is due to the diversity management operator.

Similar to the results achieved for DTLZ2 (6), the results achieved for the 8-objective version of DTLZ2 highlight a significantly superior performance of the NSGA-II/DMO when compared with the performance of NSGA-II.

Table 4. 4. DD-metric and C-metric results for DTLZ2 (8)

A = NSGA-II/DMO and B = NSGA-II (Standard SBX configuration)

	DTLZ2 (8)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1	0.45	0	-1.028
2	0.25	0	-0.55142
3	0.57	0	-1.3388
4	0.59	0	-1.3333
5	0.31	0	-0.70815
6	0.35	0	-0.77246
7	0.4	0	-0.84258
8	0.37	0	-0.76733
9	0.42	0	-0.90902
10	0.19	0	-0.42073
Mean Value:	0.39	0	-0.8672

The values of the C-metric and the DD-metric achieved by NSGA-II/DMO and NSGA-II, each optimising the DTLZ2 (8), are illustrated in Table 4.4 (standard SBX) and Table 4.5 (optimised SBX). The same observations highlight the superior performance of NSGA-II/DMO (higher coverage and lower dominated distances). Once more, the mean values of the coverage extent and the dominated distances seemed to reduce when a suitable setting for the SBX operator was used. This was only indicating an increased number of relatively non-dominated solutions when the approximation sets produced by A and B were compared.

However, Figures 4.9 and 4.10 demonstrated again the superiority of the NSGA-II/DMO over NSGA-II when a fine-grained analysis of the convergence and the diversity requirements is performed simultaneously.

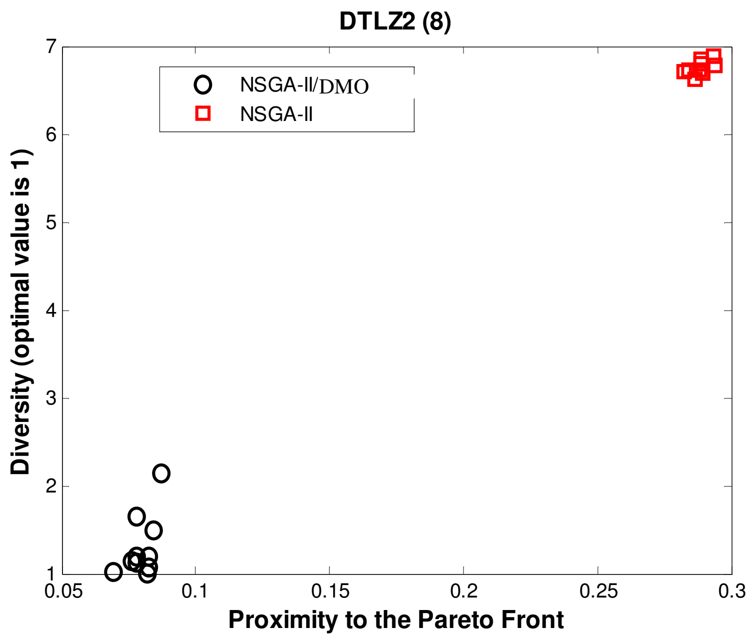


Figure 4. 9. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (8) (Standard SBX configuration)

From Figure 4.10, it can be observed that when equipped with an optimised SBX operator, NSGA-II/DMO was actually able to achieve an enhanced (compared to the scenarios deploying a standard SBX) and near optimal convergence to the Pareto fronts at certain runs of the algorithm for the 8-objective version of DTLZ2. Nevertheless, the enhanced convergence seemed to cause a minor deterioration of the diversity measure by increasing the average value of the spread metric achieved over the 10 runs to a value of 2 (the average value of the spread metric in Figure 4.9 was ≈ 1.3). This observation once again endorses the conflict between the convergence and the diversity requirements in EMO. Nonetheless, the results achieved by the NSGA-II/DMO were significantly better than the results achieved by NSGA-II for the 8-objective optimisation problem, highlighting the beneficial impact and utility of the DMO.

Table 4. 5. DD-metric and C-metric results for DTLZ2 (8)
A = NSGA-II/DMO and B = NSGA-II (Informed SBX configuration)

	DTLZ2 (8)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1	0.02	0	-0.0062699
2	0.05	0	-0.020835
3	0	0	0
4	0.02	0	-0.0058416
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
Mean Value:	0.0090	0	-0.0033

In Tables 4.6 and 4.7 the values of the C-metric and the DD-metric achieved at each execution of A and B, when standard and optimised settings for the SBX operator were used, are respectively presented. This time the optimisers were solving a 12-objective version of the DTLZ2 test function. The same conclusions drawn from the 6- and 8-objective scenarios are achieved for the 12-objective scenario.

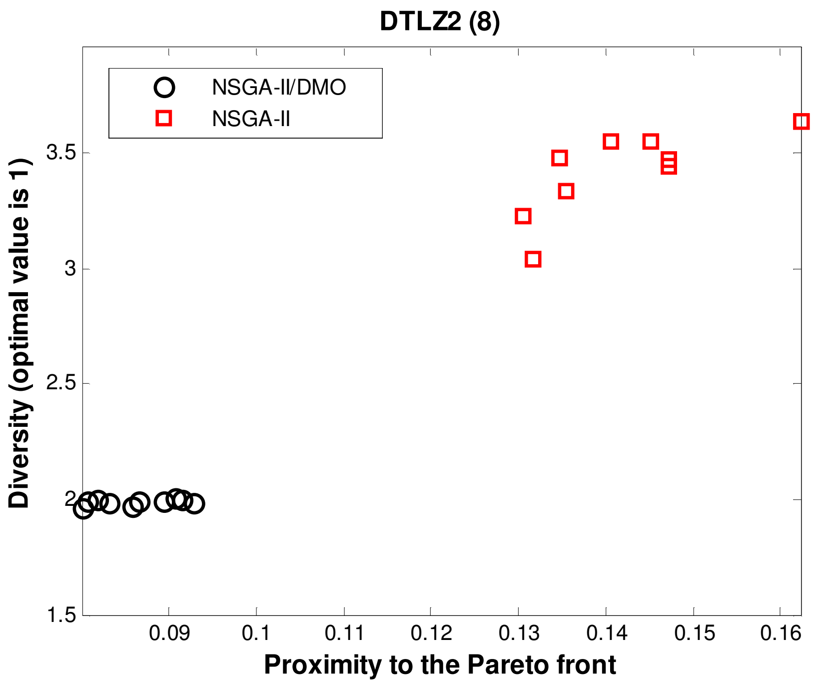


Figure 4. 10. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (8) (Informed SBX configuration)

The results of the non-dominated evaluation metric for the 12-objective scenarios (incorporating standard and optimised SBX settings) are presented in Figures 4.11 and 4.12 respectively. The significantly superior performance of the NSGA-II/DMO in terms of convergence and diversity was very apparent.

Finally, a 20-objective version of the DTLZ2 test function was tackled. In Tables 4.8 and 4.9 the values of the C-metric and the DD-metric achieved at each execution of A and B, when standard and optimised settings for the SBX operator were used, are respectively presented. In both scenarios (standard and informed SBX settings), the results achieved by NSGA-II/DMO were not covering any of the solutions achieved by NSGA-II and vice versa.

Table 4. 6. DD-metric and C-metric results for DTLZ2 (12)
A = NSGA-II/DMO and B = NSGA-II (Standard SBX configuration)

	DTLZ2 (12)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1	0.08	0	-0.18421
2	0.17	0	-0.42636
3	0.07	0	-0.17411
4	0.2	0	-0.4973
5	0.03	0	-0.070199
6	0.06	0	-0.15524
7	0.12	0	-0.3052
8	0.19	0	-0.4975
9	0.17	0	-0.44757
10	0.12	0	-0.29128
Mean Value:	0.1210	0	-0.3049

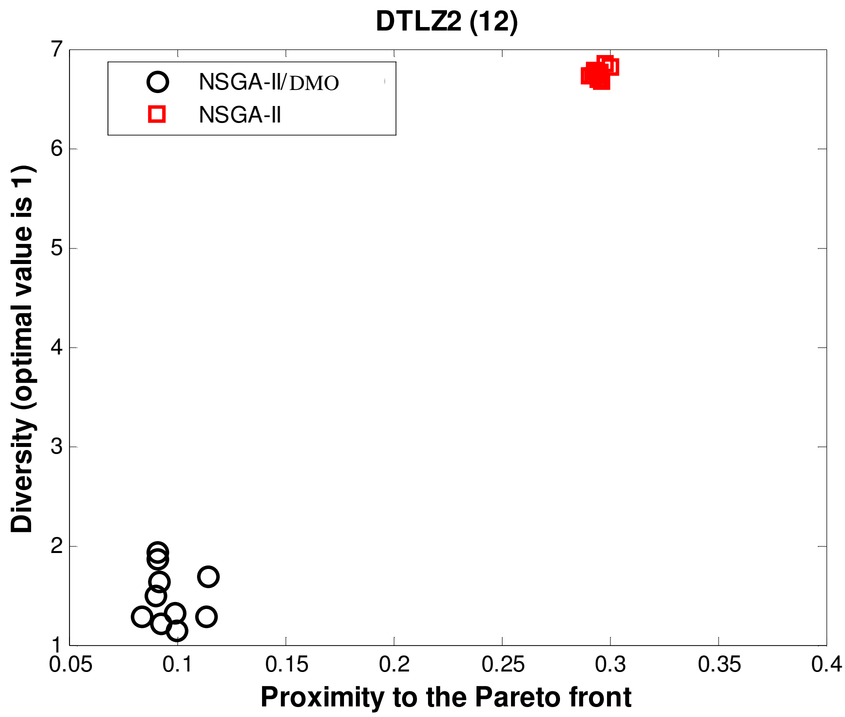


Figure 4. 11. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (12) (Standard SBX configuration)

The dominated distance metric measuring the difference of any dominated distances between the 2 approximations sets in the objective space was constantly producing a zero difference. These observations were actually expected in a 20 dimensional objective space. As the dimensionality of the objective space increases, a certain solution has a higher chance of becoming non-dominated.

Table 4. 7. DD-metric and C-metric results for DTLZ2 (12)
A = NSGA-II/DMO and B = NSGA-II (Informed SBX configuration)

	DTLZ2 (12)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1	0.01	0	-0.011096
2	0.01	0	-0.016474
3	0.04	0	-0.04734
4	0.06	0	-0.088352
5	0	0	0
6	0.04	0	-0.044164
7	0.01	0	-0.013294
8	0	0	0
9	0	0	0
10	0.06	0	-0.085266
Mean Value:	0.0230	0	-0.0306

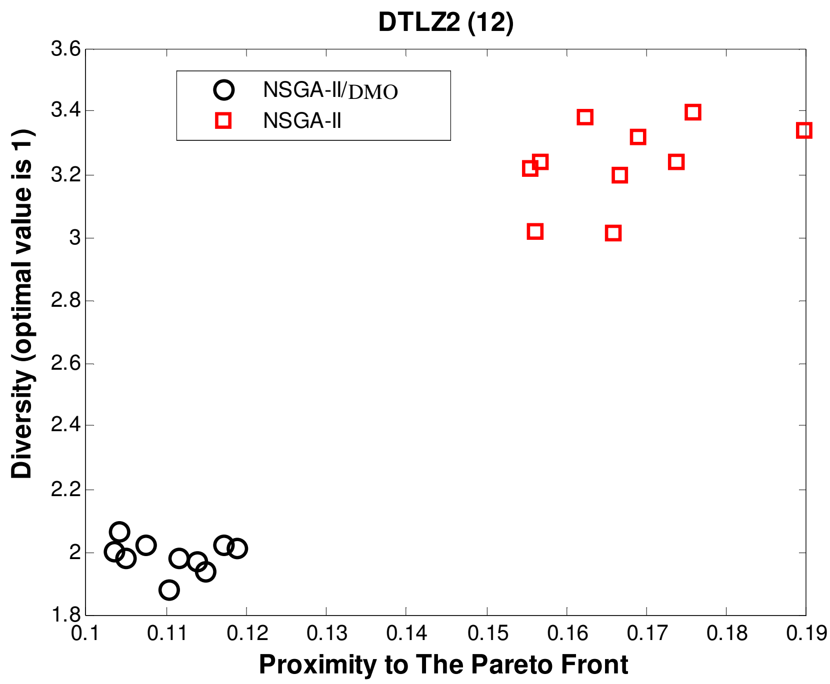


Figure 4. 12. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (12) (Informed SBX configuration)

In the context of the 2 scenarios optimising DTLZ2 (20), basically all the solutions achieved by the optimisers A and B over the 10 runs were considered non-dominated with respect to each other in terms of pure Pareto optimality. The results of the experiments undertaken in this study have highlighted the fact that as the number of objectives increases, the C-metric and the DD-metric become increasingly useless and uninformative. This is an interesting finding which, to the author's knowledge, has not been investigated in a proof of principle context and within a *many*-objective framework. The increasingly misleading outcomes of the C-metric and the DD-metric as the number of objectives increases is due to the fact that these metrics assess the relative performance of two approximation sets by finding pairs of vectors that dominate each other. However, as the dimensionality of the objective space increases, finding pairs of objective vectors that dominate each other becomes much less likely and leads the metrics to conclude equivalence between 2 approximation sets. More information on performance metrics used in the EMO can be found in Zitzler (1999), Deb (2001) and Knowles (2002).

Table 4. 8. DD-metric and C-metric results for DTLZ2 (20)
A = NSGA-II/DMO and B = NSGA-II (Standard SBX configuration)

	DTLZ2 (20)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1-10	0	0	0
Mean Value:	0	0	0

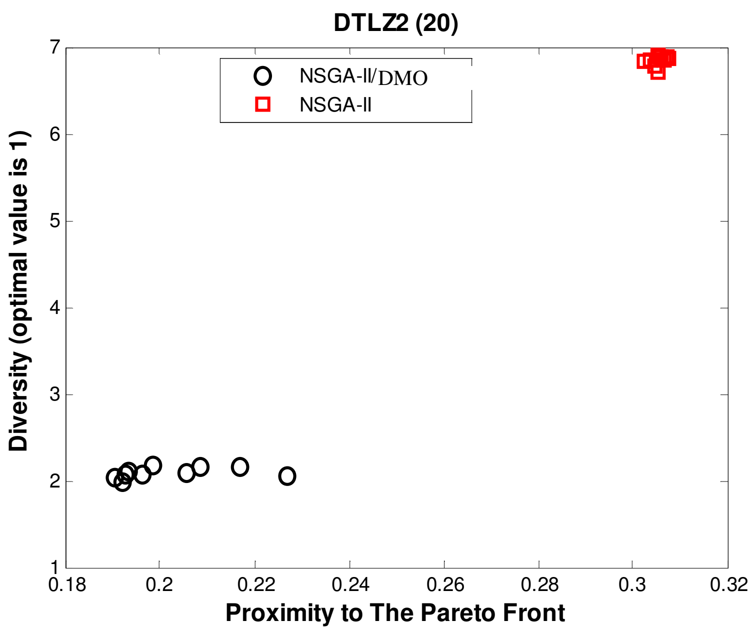


Figure 4. 13. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (20) (Standard SBX configuration)

Stopping the assessment process at this step would have concluded that the 2 optimisers are equally good and efficient. However the results of the non-dominated evaluation metric presented in Figures 4.13 and 4.14 revealed a different outcome. From Figures 4.13 and 4.14 it was very clear that the NSGA-II/DMO was more suitable for the optimisation problem consisting of 20 competing objectives. NSGA-II/DMO was achieving approximation sets much closer to the Pareto front while maintaining a much ‘better’ diversity in the required rather than the global sense. The use of an optimised configuration for the SBX operator introduced a lot of improvements to the results achieved by the 2 optimisers. NSGA-II results were improved in terms of convergence to the Pareto front and in terms of reducing the dispersal of solutions in non-optimal regions of the objective space. On the other hand, the use of an optimised SBX operator has specifically improved the quality of the results achieved by NSGA-II in terms of their convergence to the Pareto optimal front (Figure 4.14). Despite these performance improvements caused by the use of a well-tuned SBX operator, NSGA-II/DMO was still significantly outperforming NSGA-II by producing overall better quality approximation sets in terms of convergence and ‘good’ desired diversity.

Table 4. 9. DD-metric and C-metric results for DTLZ2 (20)
A = NSGA-II/DMO and B = NSGA-II (Informed SBX configuration)

	DTLZ2 (20)		
Execution Number:	**C-Metric (A, B)**	**C-Metric (B, A)**	**DD-Metric (A, B)**
1-10	0	0	0
Mean Value:	0	0	0

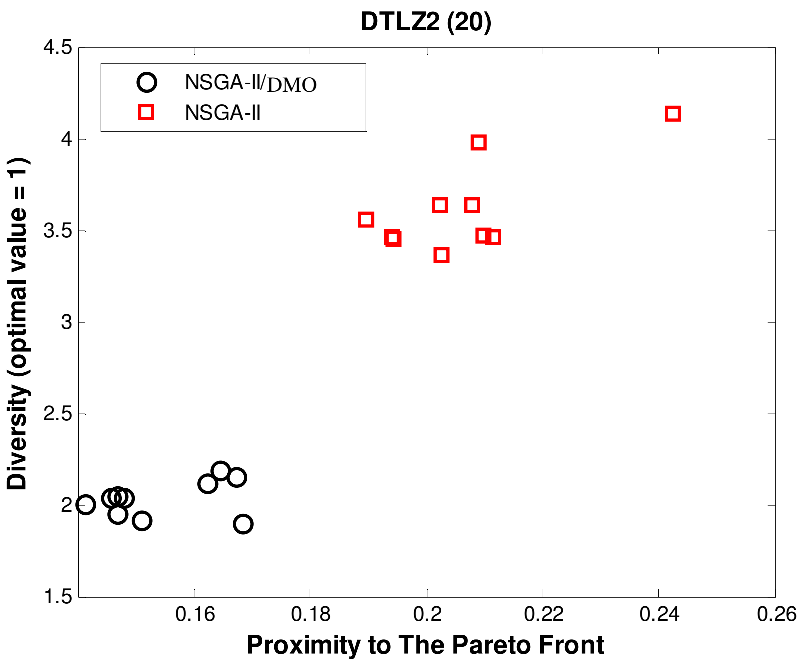


Figure 4. 14. NSGA-II/DMO VS NSGA-II in the context of DTLZ2 (20) (Informed SBX configuration)

The second set of experiments was intended to assess the performance of the second component of the DMO in particular. The second component of the DMO consists of an adaptive mutation operator, and was previously described in section 4.3.2. The effect of the DMO mutation operator was compared with the effect of the polynomial mutation operator. In order to evaluate the effect of the polynomial mutation and the effect of the DMO mutation operator on the optimisation process, similar experiments to the ones undertaken to assess the performance of the DMO are produced. However, the 2 contrasted optimisers (**A** and **B**) were respectively:

- **(A)** Standard NSGA-II optimiser using a simulated binary crossover ($\eta_c = 15$, $p_c = 1$ and $p_{ic} = 0.5$) and the DMO mutation operator ($p_m = 1/n$, where n is the number of decision variables)
- **(B)** Standard NSGA-II optimiser using a simulated binary crossover ($\eta_c = 15$, $p_c = 1$ and $p_{ic} = 0.5$) and a polynomial mutation ($\eta_m = 20$ and $p_m = 1/n$).

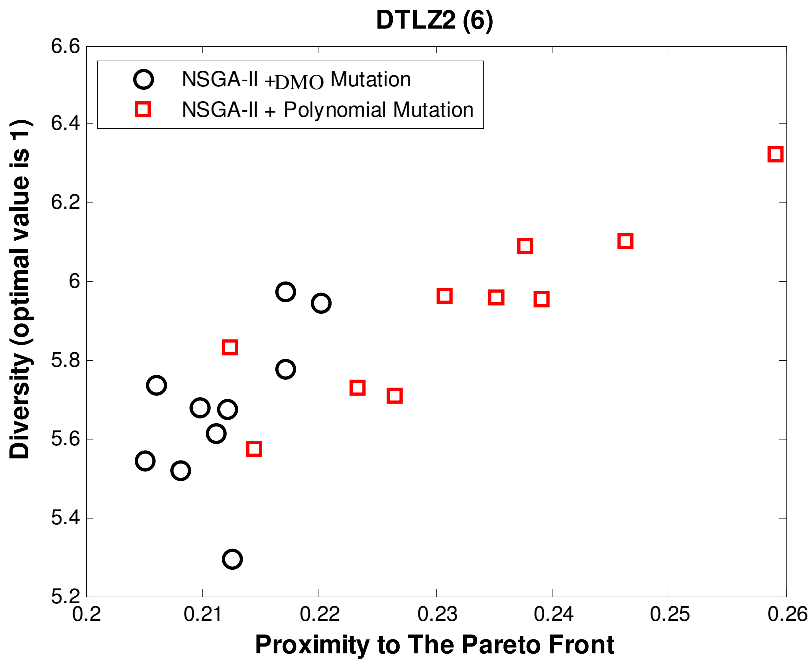


Figure 4. 15. Polynomial Mutation VS DMO Mutation in the DTLZ2 (6) context

Each of the 2 optimisers were optimising the same DTLZ2 test functions used in the first set of experiments and were executed 10 times (200 generations per execution). In Figures 4.15, 4.16, 4.17 and 4.18, the results of the non-dominated evaluation metric for each of the DTLZ2 (6) –(8) –(12) and (20) achieved at each run of the two optimisers are presented respectively. In the 6-objective optimisation context, it was noticed that despite showing some competitiveness, the effect of the DMO mutation was generally more beneficial to the optimisation process compared to the effect of the polynomial mutation. NSGA-II using the DMO mutation operator was constantly achieving convergence values larger than 0.2 but smaller than 0.22 for the DTLZ2 (6). These values were much more superior to the values achieved when the same SBX configuration was deployed alongside the whole DMO (Figure 4.17). On the other hand, NSGA-II only achieved a convergence value in the range [0.21, 0.22] in 2 out 10 executions. In terms of 'good' diversity, the optimiser 'A' using the DMO mutation operator was shown to generally achieve better values over the 10 executions (worst value achieved < 6) compared to the values achieved by the optimiser 'B' (worst value achieved > 6.3).

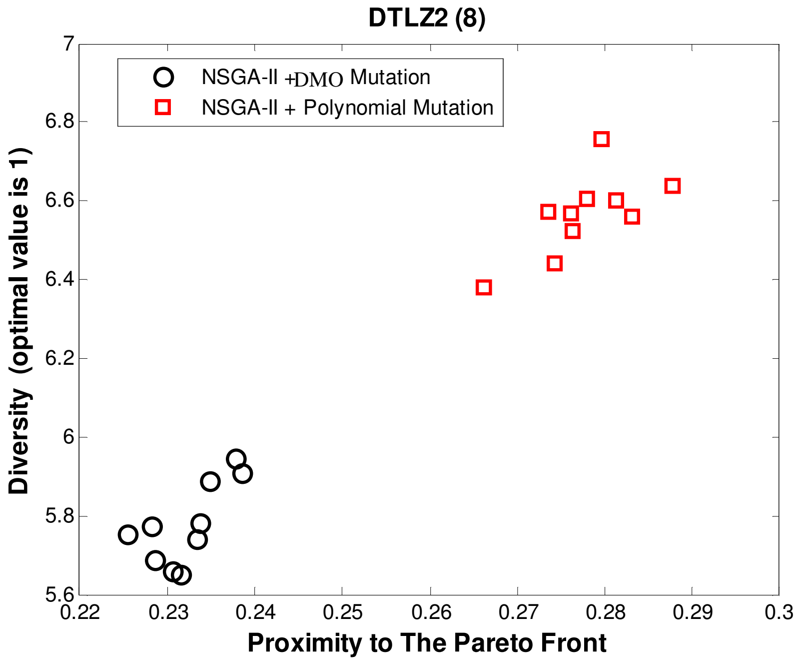


Figure 4. 16. Polynomial Mutation vs DMO Mutation in the DTLZ2 (8) context

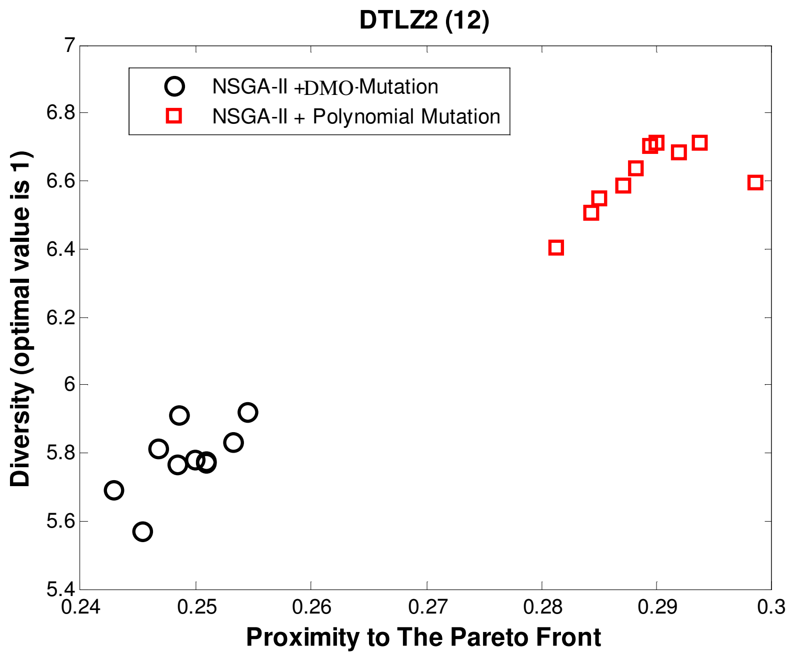


Figure 4. 17. Polynomial Mutation vs DMO Mutation in the DTLZ2 (12) context

As the dimensionality of the optimisation problem increased (in both spaces: objective and decision variable space) the competitiveness between the 2 optimisers ('A' and 'B') seemed to totally vanish (Figures 4.16, 4.17 and 4.18).

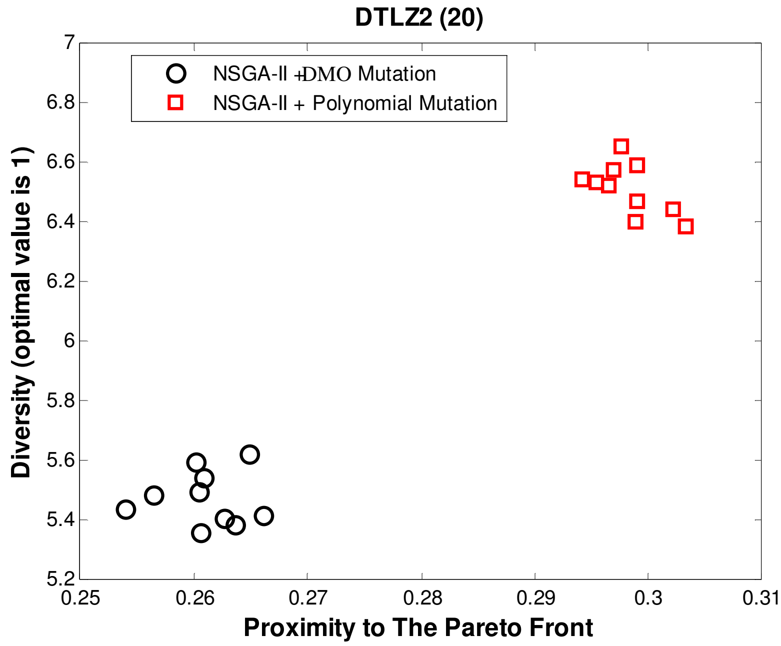


Figure 4. 18. Polynomial Mutation vs DMO Mutation in the DTLZ2 (20) context

NSGA-II using the DMO mutation operator was clearly outperforming its identical counterpart optimiser using the standard polynomial mutation (but not outperforming NSGA-II deploying the whole DMO) in terms of solutions convergence to the Pareto front and in terms of the 'good' diversity of the solutions achieved.

4.5. Conclusion

From the set of experiments presented in this Chapter, the success of the diversity management operator was established on a set of *many*-objective optimisation problems. The DMO is a beneficial strategy that addresses the conflict between the EMO requirement for a good proximity towards the Pareto front and the requirement for maintaining a diverse set of solutions. Purshouse (2004) and Purshouse and Fleming (2003b) highlighted that a successful configuration for the recombination and the mutation operators might be suitable for a certain multiobjective optimisation problem but unsuitable for another. As a result, the parameters of the recombination and mutation operators should be adequately tailored to fit a certain multiobjective optimisation problem rather than being standardised. Despite using a set of test functions with a simultaneously varying dimensionality in the objective and the decision variable space, Purshouse and Fleming (2003b) have suggested that the increase of the number of competing objectives is most likely the main influence on the success of certain recombination and mutation settings. However, using a similar experimental framework to the one used in Purshouse and Fleming (2003b) and Purshouse (2004), it was shown that when NSGA-II was equipped with the DMO, the resulting optimiser was less susceptible to the change of the parameter settings of the genetic operators. This is a very beneficial contribution of the DMO by itself. Additionally, the DMO was demonstrated to be highly beneficial for controlling the diversity requirement which usually hampers the search process and therefore the convergence of the manipulated solutions to the Pareto front of a MOP with many conflicting objectives.

NSGA-II/DMO was significantly and repeatedly outperforming NSGA-II by producing solutions closer to the Pareto front and maintaining a near optimal and desired diversity among the solutions, for a set of *many*-objective optimisation problems (6 to 20 objectives). DMO was tested on set of test functions with well-defined Pareto fronts. Nevertheless, the strategy can be used to solve any multiobjective optimisation problem which is tackled by a MOEA. The DMO is an efficient strategy that does not require high computational efforts. The decision maker/operator is only required to provide an approximate, targeted or desired, value for the extreme solutions (in terms of each objective) in the objective space. These solutions will serve to define an approximation to the vertices of the hypercube which contains the desired ROI, and therefore to define the notion of a 'good' diversity. Note that these suggested vertices will solely play a role defining the notion of a desired diversity measure in order to efficiently control the diversity promotion mechanisms in a MOEA and guide the search towards the Pareto optimal front. The notion of a 'desired' or 'good' diversity can then be progressively and appropriately modified using a progressive preference articulation technique such as the technique by Fonseca and Fleming (1998) or Branke and Deb (2004).

The use of progressive preference articulation techniques to manipulate the definition of dominance and reduce the dimensionality of the search spaces is highly commendable for solving evolutionary multiobjective optimisation problems and will be addressed in Chapter 5.

4.6. Summary

The Proximity of an approximation set towards the Pareto optimal front of a multiobjective optimisation problem, and the diversity of the solutions within the approximation set are two essential requirements in EMO. These two requirements are found to be conflicting with each other in the *many*-objective optimisation scenarios. This conflict is hindering the optimisation process of some of the most established MOEAs that uses Pareto dominance as a primary selection criterion alongside the diversity measure as a secondary discriminator. In this chapter an adaptive local strategy for controlling and promoting diversity in the *many*-objective optimisation scenarios is introduced and tested on a set of test functions with varying number of objectives. The results achieved by the introduced strategy outperformed the results achieved by a reputed and representative MOEA in terms of both criteria: convergence and diversity. The strategy is very promising and future work should include its testing on a real world optimisation problem.

Having addressed the convergence requirement in Chapter 3 and the diversity requirement in Chapter 4, in the following chapter progressive preference articulation techniques, a commendable approach for dealing with both requirements in evolutionary multiobjective optimisation will be explored.

Chapter 5

Progressive Preference Articulation:

The Practical Approach for EMO

'You must have an aim, a vision, a goal. For the man sailing through life with no destination or "port-of- call", every wind is the wrong wind'.

Tracy Brinkmann

5.1. Introduction

As the number of competing objectives increases the optimisation becomes more complicated. This is due to the introduction of new difficulties such as the obvious dimensionality increase of the Pareto front, and the difficulty of visualizing such scenarios. Reducing the dimensionality and therefore the complexity of an optimisation task is a straightforward way for dealing with the high-dimensional problems. Early approaches such as the weighted sum or the Tchebyshev method (Coello, Veldhuizen and Lamont 2002) consisted of scaling techniques to convert multiobjective problems into a single objective counterpart. Such approaches presented several shortcomings, mainly the absence of the desired parallel search capacity. More recent techniques of dimensionality reduction for dealing with multiobjective optimisation problems consist of techniques to identify objectives redundancy and eliminate it. Principal Component Analysis has been used as an example of such a technique (see for example Purshouse (2004), Purshouse and Fleming (2003a) or Deb and Saxena (2005)). Its aim is to identify redundant objectives, whose absence has no substantial effect on the optimisation process, thereby simplifying the complexity of certain high dimensional problems and reducing the hyperspace of solutions. More recent research into dimensionality reduction in EMO includes the work of Brockhoff and Zitzler (see Brockhoff and Zitzler (2006a) and Brockhoff and Zitzler (2006b)). Brockhoff and Zitzler have investigated the problem of finding the minimum subset of objectives which are essential to the optimisation problem. They proposed an exact algorithm for solving the *minimum objective subset problem* (MOSS). The complexity of the proposed algorithm is polynomial in the number of decision variables and exponential in the number of objectives. While dimensionality reduction is a remedial measure to tackle multiobjective optimisation problems, it can only be deployed in reducible scenarios where redundancy or objective relationships such as independence or harmony exist and are detectable.

In scenarios, where insufficient redundancy can be detected in a high-dimensional problem, progressive preference articulation (PPA) is a proven useful alternative remedial measure. The incorporation of DM preference into evolutionary multiobjective optimisation algorithms is very useful for guiding the search into pertinent regions of interest (ROI), which are relevant to the decision maker. Coello (2000a) has produced a comprehensive survey about handling preferences in EMO. More recent surveys about preference articulation techniques in evolutionary multiobjective optimisation can also be found in Rachmawati and Srinivasan (2006). PPA can also provide advantages over the use of pure Pareto-optimality, which is unfettered in its search and is liable to produce solutions outside the ROI as well as within it.

Until recently, most EMO research has focused on bi-objective problems where the need for incorporating the decision maker’s preferences is less apparent. The aim of the study presented in this Chapter is to encourage and promote the research of incorporating progressive preference articulation techniques into evolutionary multiobjective optimisation. In this chapter, some of the most established and most recent preference articulation techniques are discussed and upgraded to their progressive versions for incorporation into evolutionary multiobjective optimisation processes. The use of the increasingly popular ε-dominance concept as a potential PPA technique is also implemented and investigated. The major strengths and limitations of the investigated PPA techniques for tackling multiobjective optimisation problems are discussed from a decision maker’s point of view. Their utilities are evaluated by assessing the pertinence of their achieved results to the DM’s preferences.

The preference articulation techniques investigated in this work include Branke’s “guided dominance principle” Branke *et al* (2001), Deb’s “biased crowding technique” (Branke and Deb 2004), a suggested technique based on a simple modification of the ε-dominance concept within the framework of Deb’s steady state ε–MOEA (Deb, Mohan and Mishra 2003) and the preferability based approach operator (FF-PPA) (Fonseca and Fleming 1998), one of the first truly PPA techniques for EMO.

In Section 5.2, a description of the preference articulation techniques inspected in this study is given. In Section 5.3, the usefulness and practicality of the studied progressive preference articulation techniques are illustrated on simple bi-objective and 4-objectives scenarios. In some cases, a progressive capability is introduced into existing preference articulation techniques. The strengths, weaknesses, user-friendliness and efficiency of these PPA techniques, in a multiobjective optimisation context, are also discussed from the viewpoint of the decision maker. Lastly, in Section 5.4, some concluding remarks are presented.

5.2. The Investigated Preference Articulation Techniques

In this section a brief description of some of the most recent preference articulation techniques is provided:

5.2.1. Guided Dominance for EMO

Branke *et al* (2001) introduced the guided dominance principle within the context of a novel optimiser, termed as the Guided Multi-Objective Evolutionary Algorithm (G-MOEA). The principle of guided dominance manifested the DM's preferences through a modification of the definition of dominance. The user has to determine all maximally acceptable tradeoffs between all pairs of objectives.

To illustrate this concept, consider an optimisation problem consisting of two competing objectives. In order to use the guided dominance scheme, the DM has to decide *a priori* the maximum acceptable amount of degradation in terms of 'objective 2' which can be deemed worthy to be recompensed by a single unit of improvement in terms of 'objective 1', and vice versa. Applying the guided dominance principle modifies the standard Pareto dominance concept which generally governs the selection processes in EAs.

The modified dominance principle, affected by the guided dominance scheme, is presented in Equation 5.1. The parameters m_{12} and m_{21} denote correspondingly the maximum acceptable amount of degradation in terms of objective 1 and 2 which are compensated by a single unit of improvement in terms of objective 2 and 1 respectively. Using the guided dominance principle, a solution x is said to dominate an alternative solution y following the definition presented in Equation 5.1 with an inequality in at least one case. The articulated tradeoff values between pairs of objectives relax the standard Pareto dominance principle. As a result, a certain solution x will then dominates a larger region in the objective space.

$$
\begin{aligned}
x \succ y \Leftrightarrow \quad & (f_1(x) + m_{12} f_2(x) \le f_1(y) + m_{12} f_2(y)) \wedge \\
& (m_{21} f_1(x) + f_2(x) \le m_{21} f_1(y) + f_2(y))
\end{aligned}
\qquad \textbf{5.1}
$$

From a geometrical point of view, the tradeoff values (m_{12} and m_{21}) defined by the decision maker represent the slope values of the borders demarcating the regions of the objective space which are dominated by each candidate solution. By carefully setting suitable trade-off values, the decision maker can articulate a particular region of interest on a convex Pareto front. Such ROI will be bounded by the solutions possessing certain tradeoff functions which are tangent to the convex Pareto front. However, because of the linear utility assumed in the guided dominance principle, it might not be possible to articulate a certain ROI on a concave or multimodal Pareto front.

The guided dominance principle is illustrated in Figure 5.1 on a simple bi-objective optimisation problem. Suppose that the two candidate solutions **A** and **B** in Figure 5.1 had the following values in terms of objectives 1 and 2 respectively: A = (2, 1), B = (1, 2). From a pure Pareto dominance point of view, these two solutions are considered non-dominated with respect to each other.

In a different scenario, deploying the guided dominance principle, the DM might decide that 'Objective 2' is more important than 'Objective 1'. For example, if the DM decides that 'one' unit of improvement in terms of 'Objective 1' is worth a maximum of 'one' unit of deterioration in terms of 'Objective 2', then m_{21} = 1. On the other hand, if the DM decides that 'one' unit of improvement in terms of 'Objective 2' can be traded with a maximum of 'two' units of deterioration in the performance of 'Objective 1', then m_{12} = 2.

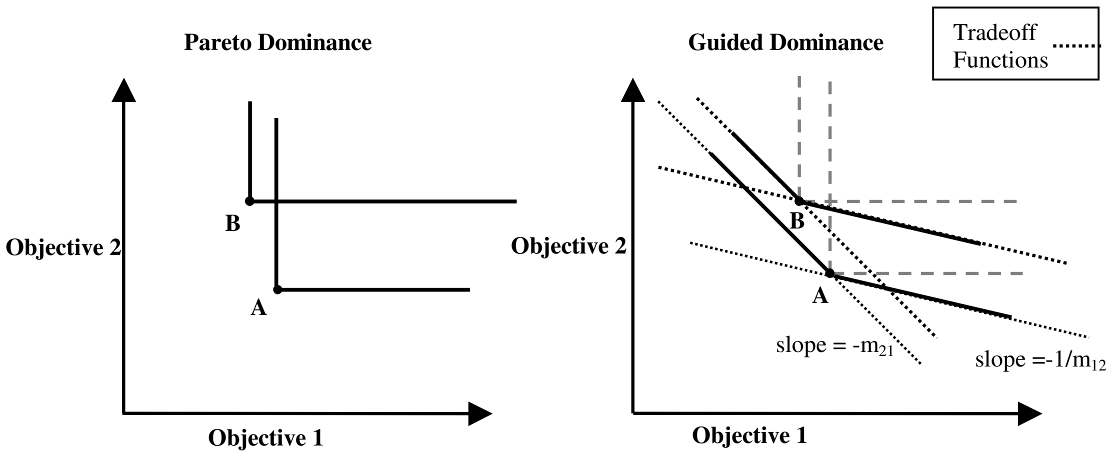


Figure 5. 1 Pareto Dominance versus Guided Dominance

Manifesting the previous decision maker preferences via the guided dominance principle highlights that the candidate solution **A** is now preferred and therefore dominates the solution **B.** In other words, applying Equation 5.1 indicates that solution **A** dominates solution **B** despite being non-dominated with respect to solution **B** using the standard Pareto dominance principle ($[A \succ B \rightarrow (2+2\times1) < (1+2\times2)] \wedge [(1+1\times2) \leq (2+1\times1)]$).

The guided dominance principle can be assimilated with the standard Pareto dominance principle applied to a simple transformation of the objective space. This makes its incorporation into dominance-based evolutionary algorithms straightforward and practical.

5.2.2. Biased Crowding Distance

The biased crowding distance approach is one of the state-of-the-art preference articulation techniques that allows the user to efficiently focus on certain regions of interest on a convex or concave Pareto front. This technique has its roots in the biased fitness sharing approach (Deb 1999a) which was employed in the Non Dominated Sorting Genetic Algorithm (NSGA) (Deb *et al* 2002). In Deb (1999a), the sum of Euclidean distances between a certain candidate solution and its nearest-neighbours, which are measured in terms of each dimension of the objective space and used as a density estimate for sharing fitness values between solutions residing densely populated areas of the objective space and hence for promoting diversity, were affected by weighting values defining certain DM preferences and objective priorities. The transformation of the sum of distance measures into a weighted sum of measures was termed the biased fitness sharing approach and allowed the DM to focus on a certain specific objective from a set of objectives. However, the biased fitness sharing approach did not provide the utility of focusing on an intermediate ROI. The biased crowding measure is inspired by the biased sharing approach, and addresses the shortcomings of the latter approach by providing the DM with the capability of focusing on a certain region of interest on the Pareto optimal front.

The biased crowding measure is defined for any solution *k* on any particular front (in the objective space) all along the optimisation process as follows:

$$D_k = d_k \left(\frac{d_k^{'}}{d_k} \right)^{\alpha} \qquad \textbf{(5. 2)}$$

In Equation 5.2, d_k is the original crowding measure for the solution *k* based on its neighbouring solutions, and $d_{k'}$ is the crowding measure of the projected solutions on the plane whose direction is specified by the user to express a certain region of interest on the Pareto front. The parameter α is responsible for controlling the bias intensity. As a result, solutions located on the region of the front which is tangent to the DM's devised projection plane, which reflect a certain preference of a ROI, will be biased and favourite to be retained because the ratio $d_{k'}/\ d_k$ will be close to unity, and therefore D_k will be approximately the same measure as the original crowding value for such solutions. The crowding measure applied is the nearest-neighbour density estimate used in NSGA-II and which was presented in Section 2.3.2.

Figure 5.2 illustrates the concept of the biased crowding distance for three different ROIs on a convex front. In the first part of Figure 5.2 **(a)**, the DM specifies a direction vector $\boldsymbol{\eta}$ indicating his/her 'vague' preference of ROI in the 2-dimensional objective space. The user-specified direction vector $\boldsymbol{\eta}$ denotes the direction of the solution projections needed to compute the biased crowding measure. In Figure 5.2a, the DM has articulated an interest in the region of the convex Pareto front which bias 'objective 2' (f2). The projection plane **'P'** can then be any line or plane whose '*normal*' vector is $\boldsymbol{\eta}$**.**

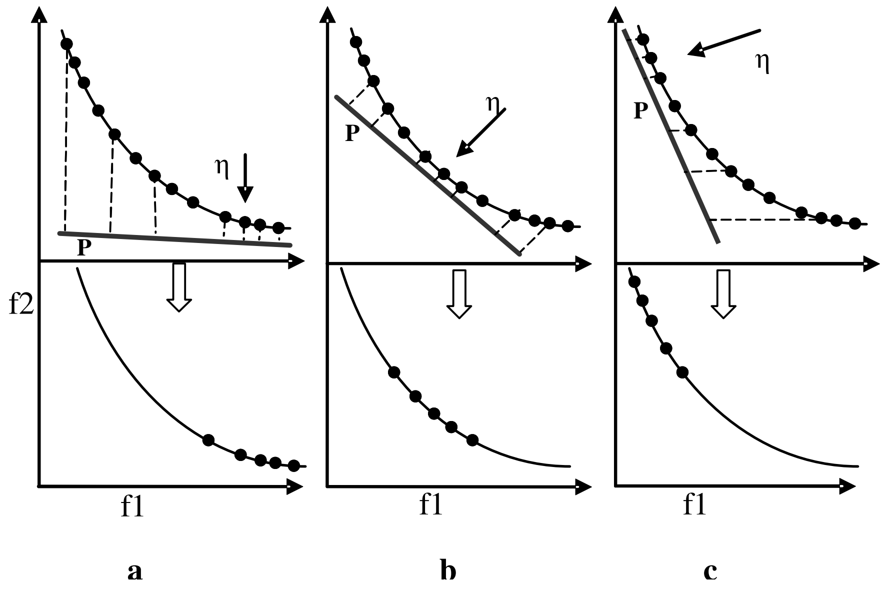


Figure 5. 2 Biased Crowding Distance (P = Projection Plane, ή = Projection Direction)

By projecting the local front of solutions achieved by a MOEA on the projection plane P and calculating the biased crowding distance defined in Equation 5.3, the solutions lying around the lower part of the Pareto front which is tangent to the projection plane P will be favored for selection for variation and survival. In Figures 5.2b and 5.2c, two other preference scenarios are illustrated using the biased crowding distance.

5.2.3. ε –MOEA: Manipulating the ε-dominance

Laumanns, Thiele, Deb, and Zitzler (2000) proposed the ε-MOEA, one of the state-of-the-art multiobjective evolutionary optimisers. At a later stage, Deb, Mohan and Mishra (2003) introduced further improvements to the ε-MOEA. It is a steady-state algorithm composed of two populations of solutions, which co-evolve simultaneously, but independently. A solution can only be included in the archive, which eventually should contain a representative bounded set of solutions which form the Pareto front, if it is not ε-dominated (see Section 2.1.1) by any of the other members of the archive. ε-MOEA uses a grid-like strategy similar to PAES (Knowles and Corne 2000), but more sophisticated, to divide the objective space into hyperboxes and promote solution diversity without setting an upper limit on the archive size prior to the approximation. Instead the strategy used in ε-MOEA ensures that the archive will eventually be bounded with a well-distributed and limited number of solutions, which represent the Pareto front. Despite the sophistication and usefulness of ε-MOEA, the deployed ε-dominance concept is the reason behind choosing this optimisation technique as a preference articulation technique to be investigated along with the other techniques used in this work. By setting a vector of progressively articulated ε-values, instead of a single fixed value, to form the basis for solutions selection and inclusion in the archive, ε-MOEA is upgraded to a PPA technique which enhances its overall performance, at least from a decision maker's point of view within a *many*-objective optimisation context. Each single ε value will correspond to the accuracy or tolerance in terms of a certain specific dimension or objective. The motivation behind this upgrade is to investigate and exploit the efficacy of the ε-dominance concept as a PPA technique.

The ε-MOEA procedure as described by its original authors is presented in Figure 5.3 and is thoroughly illustrated in the following.

At a certain iteration **t** of the optimisation process, two solutions *p* and *e* are first selected respectively from the population of solutions P and the online archive A for recombination. The solution ***p*** is selected from the population P(t) following the procedure *pop_selection* described below.

pop_selection

The selection process starts by picking two random solutions from P. The dominating solution, using standard Pareto dominance, will constitute the solution *p*. In the case where the two randomly picked solutions from P(t) were non-dominated with respect to each other, one of them is picked at random to constitute the solution *p*.

On the other hand, using the *archive_selection* procedure, one solution is randomly chosen from the archive A(t) to constitute the other parent solution *e*. The two picked solutions *p* and *e* are then recombined and one single offspring solution *c* is created and used to update both parent and archive populations. When updating the population *P(t),* the usual Pareto dominance check is applied between the offspring solution *c* and all the solutions in P. *c* is accepted for inclusion in the population following the *pop_acceptance* procedure described below.

pop_acceptance

If ***c*** dominates one or more solutions in P**,** then it replaces the solution (or a random solution out of the subset of solutions) it dominates. On the other hand, if any of the solutions in P dominate *c* then the offspring solution is rejected. In the last case where the two previous checks fail indicating that *c* is non-dominated with respect to all of the solutions in P, *c* is accepted in the population P replacing a randomly picked solution from P for elimination. A fixed population size is therefore maintained throughout the optimisation process.

The most important and innovative process of the ε-MOEA procedure is the archive acceptance step (*archive_acceptance)*. The archive population is updated based on the ε-dominance concept illustrated in Figure 5.4.

Each solution **'y'** in the archive is assigned a so-called '*identification array*' **B (y)** = (B_1, B_2, ….B_n) (where n is the number of objectives). The identification array represents the coordinates of the main corner point (**A** in Figure 5.4) in the objective space which defines the region ε-dominated by the solution '**y**'. The value of the identification array in terms of a certain objective 'i' is defined in Equation 5.3 for a minimization problem (where f_i^{min} and ε_i are respectively the minimum possible value and the tolerance for the i^{th} objective).

$$B_i = \left\lfloor (f_i - f_i^{\min}) / \varepsilon_i \right\rfloor \qquad \textbf{(5.3)}$$

Step 1 `Randomly initialize a population` $P(0)$.
`The non-dominated solutions of` $P(0)$ `are copied to an archive population` $A(0)$. `Set the iteration counter` $t=0$.

Step 2 `One solution` p `is chosen from the population` $P(t)$ `using the pop_selection procedure.`

Step 3 `One solution` e `is chosen from the archive population` $A(t)$ `using an archive_selection procedure.`

Step 4 `One offspring solution` c `is created using` p `and` e.

Step 5 `Solution` c `is included in` $P(t)$ `using a pop_acceptance procedure.`

Step 6 `Solution` c `is included in` $A(t)$ `using an archive_acceptance procedure.`

Step 7 `If termination criterion is not satisfied, set` $t = t + 1$ `and go to Step 2, else report` $A(t)$.

Figure 5. 3 The ε-MOEA procedure

This scheme divides the whole objective space into different hyperboxes whose sizes correspond to the vector of ε –values representing the permissible tolerance in terms of each objective.

In Deb, Mohan and Mishra (2003), the solved optimisation problems consisted of scenarios with equally weighted objectives, hence no objective preferences or discriminations were articulated using the vector of ε-values. As previously mentioned, ε-

MOEA was reported to show superior performance on a set of test functions with varying number of objectives (2 to 4) when compared to some of the most established MOEAs.

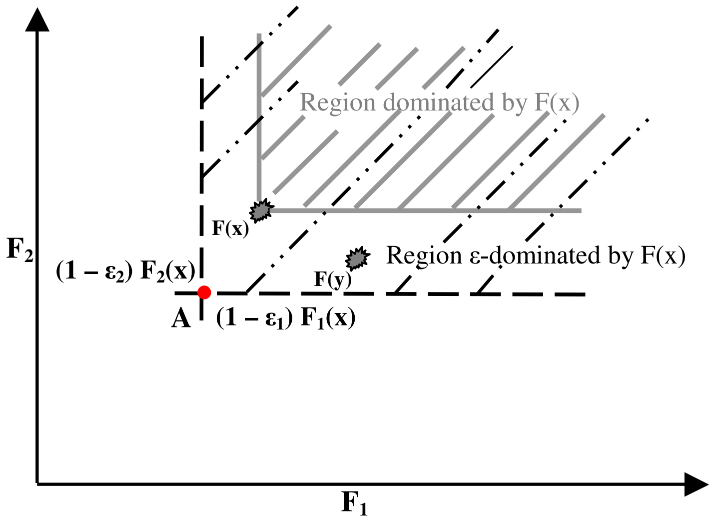


Figure 5. 4 ε-Pareto dominance for 2 objectives

Table 5. 1 ***archive_acceptance***

<table>
<tr><th colspan="2">Case</th><th>Action</th></tr>
<tr><td colspan="2">The identification array of the offspring solution c is dominated by one or more identification arrays which correspond to certain solutions in A(t) indicating that c is ε-dominated</td><td>Reject c</td></tr>
<tr><td colspan="2">The identification array of the offspring c dominates an identification array of a certain solution in A(t)</td><td>Accept c for inclusion in the archive replacing the solution that it ε-dominates</td></tr>
<tr><td rowspan="6">The identification array of the offspring solution c is ε non-dominated with respect to all of the solutions in A(t).</td><td rowspan="2">Sub-Case 1</td><td>Dominance Check: Using the standard Pareto dominance concept:</td></tr>
<tr><td>(1) <u>The offspring c dominates s</u>
Accept c and reject s.</td></tr>
<tr><td rowspan="2">The offspring solution c share the same identification array with a certain solution s in the archive A(t)</td><td>(2) <u>s dominates the offspring c</u>
Reject c</td></tr>
<tr><td>(3) <u>s and c are non-dominated</u>
Keep the solution with the smallest Euclidean distance to the shared identification array.</td></tr>
<tr><td>Sub-Case 2</td><td rowspan="2">Accept c for inclusion in the archive</td></tr>
<tr><td>The offspring solution c does not share the same identification array with any solution in the archive A(t)</td></tr>
</table>

However, the main motivation for including the ε-MOEA procedure in this comparative study is investigating the deployment of the ε –dominance concept that it incorporates in its beneficial strategy for updating the archive as a potential PPA technique. The identification array for the new offspring solution c and all of the solutions in A(t) are calculated using the vector of ε values defined *a Priori* by the DM. The offspring solution c is then either rejected or accepted for inclusion in the archive based on one of four possible cases following a standard Pareto dominance check between the identification arrays corresponding to the offspring solution c and all of the solutions in the archive P(t). The four cases are presented in Table 5.1.

5.2.4. FF-PPA Technique

Formulated and implemented in 1998 as an extension to the rank-based fitness assignment method (Fonseca and Fleming 1993), Fonseca and Fleming's (1998) PPA technique remains an important approach to progressive preference articulation. This technique is based on a combination of concepts such as Pareto optimality, constraint optimisation and satisfaction, the lexicographic method (Fourman 1985) and goal programming (Kursawe 1991). The core of this PPA technique is based on the *Preferability operator*, which is a transitive relational operator that incorporates goal and priority information about the objectives and which consequently modifies the dominance definition.

Using FF-PPA, two alternative solutions **A** and **B** are first compared in terms of their objectives with the highest priority while disregarding the objectives of this priority class that meets their goal values. In the case where the objectives, belonging to the same priority class, of solutions **A** and **B** meet all their goal values or contrarily violate some or all of their goal values in an exact similar way, the next priority class will be considered. This process continues until reaching the lowest priority class, where solutions are compared based on the usual Pareto optimality concept. Using the preferability operator as a decision strategy reduces the usual Pareto dominance based optimisation to the particular scenario where all the objectives possess the same priority class and no targeted goal values are set up. On the other hand, in the case where goal values are articulated within an optimisation problem consisting of equally weighted objectives that belong to the same priority class, the fitness assignment process allocates higher importance to the objectives not yet satisfying their goals. A candidate solution **A** is then said to be preferred to an alternative candidate solution **B** if one of the three following conditions is valid:

(1) Solution **A** meets the goal values for a certain subset of objectives and is better (in terms of Pareto dominance) than **B** in terms of the remaining objectives (not meeting their goals in **A**).

(2) The value of the objectives not meeting their goals in **A** are exactly equal to the values of the same objectives in **B**, but **A** is better (in terms of Pareto dominance) than **B** in terms of the remaining objectives (meeting their goals in **A**).

(3) The value of the objectives not meeting their goals in **A** are exactly equal to the values of the same objectives in **B**, but **B** does not meet the goal values for the remaining objectives (meeting their goals in **A**).

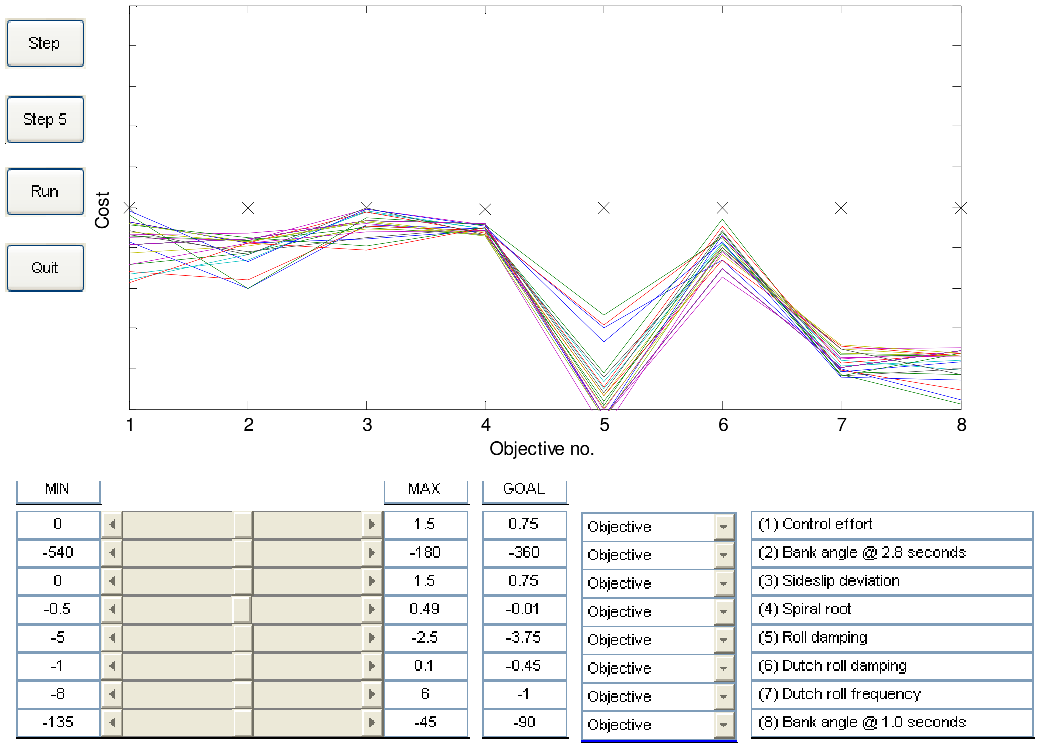


Figure 5. 5 The user interface of Fonseca and Fleming's PPA technique (Optimising an 8 objective problem of Aircraft control system design)

Through a user-friendly interface, the DM can set goal values for the objectives being optimised and can change the priorities of the objectives in a progressive fashion at any time during the optimisation process; the dominance concept gets updated accordingly. In other words, using this technique the DM has full control of the optimisation process and can efficiently focus on any region of interest and reduce the dimensionality of the search space at any time and upon request. Figure 5.5 illustrates the user interface of this PPA technique which includes the parallel coordinates graph (Inselberg 1985), an efficient, FF-PPA independent, visualization technique for any problem dimension. Here, using parallel coordinates, each line in the graph connects the performance objectives achieved by an individual member of the population and represents a potential solution to the design problem. This is in contrast to the usual Cartesian method of representation and has the advantage of being able to handle representations where the number of objectives exceeds three.

In the next section, a graphical demonstration of the above mentioned PPA techniques will be presented illustrating the accuracy, efficiency and usefulness of these techniques from a DM point of view.

5.3. PPA Techniques in Practice

A major benefit for incorporating the decision maker's preferences into the evolutionary multiobjective optimisation process is the manipulation of the Pareto dominance concept which widely governs the selection mechanisms of most MOEAs. Tailoring a pertinent definition of solutions' dominance that fits the DM's preferences is very beneficial for reducing the dimensionality of the search space and dealing with the conflict between solutions' convergence towards the Pareto front and their required diverse distribution. This last conflict is especially apparent in optimisation frameworks with many competing objectives and has been linked to the active diversity promotion techniques deployed in most MOEAs. Being able to progressively articulate preferences, a DM can then make use of any information that becomes available during the optimisation process to modify or refine his/her preferences and steer the search process of a MOEA in the right direction.

In the absence of preference articulation, a certain remote solution **'A'** diverging from the Pareto front but presenting remarkable performance in terms of one of many competing objectives can be explored and deemed non-dominated. Such a solution can potentially spoil the convergence process towards the Pareto front. The latter observation was previously described in Section 4.2 and a diversity management operator addressing this issue was introduced in Chapter 4. This highlights a drawback of the standard Pareto dominance when used as a primary selection criterion alongside an active diversity promotion mechanism in the many objective optimisation scenarios. On the other hand, in an optimisation scenario incorporating the DM's preferences, the same solution '**A**' is only considered as a non-dominated solution if it lies in the decision maker's ROI or when the objective in which '**A**' is excelling is prioritised.

In this section, the performance of the guided dominance concept, the biased crowding measure, the use of ε-dominance and FF-PPA technique will be assessed. These techniques allow the decision makers to articulate their subjective preferences and therefore assist them in the decision making process. Assessing the utility of PPA techniques from a DM point of view involves many psychological criteria (e.g. ease of use, effort to be skilful, ease of learning and effectiveness) and many assessment strategies can be used (e.g.

making surveys and collecting feedback from different DMs). In this study, the assessment and comparison of these techniques is confined to a subjective yet logical appreciation of each technique's utility based on their responses to different type of DM preferences (mainly exact and vague preferences). In other words, a 'Black Box Testing Strategy' will be adopted in this section. Black box testing is a strategy used in software engineering to test software applications without requiring the knowledge of their implementation codes and internal designs. Moreover, two essential factors are examined in order to assess the quality of the investigated PPA techniques. Since PPA techniques require DM interaction and are meant to be helper tools for decision makers, it is hence very crucial to investigate whether these techniques do actually meet the DM requirements (i.e. preferences). The first assessment factor therefore consists of checking whether a certain PPA technique meets the essential user requirements. Assessing this feature is very essential for evaluating new processes and systems, and is widely known as "User Acceptance Testing" (Davis 1989) in the fields of software engineering, cognitive sciences and human-machine interactions. In this study, the main principle of "User Acceptance Testing" is deployed to assess each PPA technique. This is performed by simulating well-defined PPA scenarios and analysing the level of preference satisfaction provided by each PPA technique. Alternatively, user acceptance testing could be realised by letting the user (DM) experiment with the different PPA techniques and report about his/her satisfaction with the utility meeting his/her expectation. However, the second approach for user acceptance testing is better deployed when a fairly similar (ideally, the same) interface (e.g. GUIs) between the DM and each PPA technique is provided. The second examined factor consists of checking whether a certain PPA technique meets its intended purpose as described by its original author(s). The broad concept of assessing whether a certain system or process meets its intended purpose is widely known as 'Usability Testing' (Lindgaard 1994).

The efficiency and practicality of the investigated techniques will therefore be examined from the decision maker's point of view, assuming that their expertise in evolutionary computation might be very limited or nil. In addition to the previously described testing strategy, the goal is to highlight the utility of these PPA techniques to a DM in terms of reducing the search space, and focusing on a desired ROI. Several bi-objective scenarios, convenient for graphical illustrations, will be deployed to highlight the strengths and weaknesses of these techniques, and will permit the inference of well-based conclusions for high-dimensional cases. Additionally, further progressive PPA scenarios will be undertaken on a 4-objective optimisation problem. Note that NSGA-II was chosen to be the underlying optimiser for hybridizing the biased crowding technique and the FF-PPA.

All of the optimisers used similar configurations and were balanced in terms of the number of objective function evaluations.

5.3.1. Demonstration of the Guided Dominance principle as a PPA technique

Similar to the frameworks which will be applied for assessing the other PPA techniques (in Sections 5.3.2, 5.3.3 and 5.3.4), in this Section, the goal was to assess the utility of the guided dominance principle as a PPA technique and promote the use of such techniques in the evolutionary multiobjective optimisation community. The guided dominance method can be assimilated with the Pareto dominance concept operating on a suitably transformed objective space. Note, that the objective space transformation is a straightforward process when dealing with bi-objective optimisation scenarios, but becomes more complicated as the number of objectives increases. The intention was to upgrade the guided dominance principle to a progressive preference articulation technique which can be used interactively by a DM. In Figure 5.6 the suggested user interface for the guided dominance based PPA technique is illustrated.

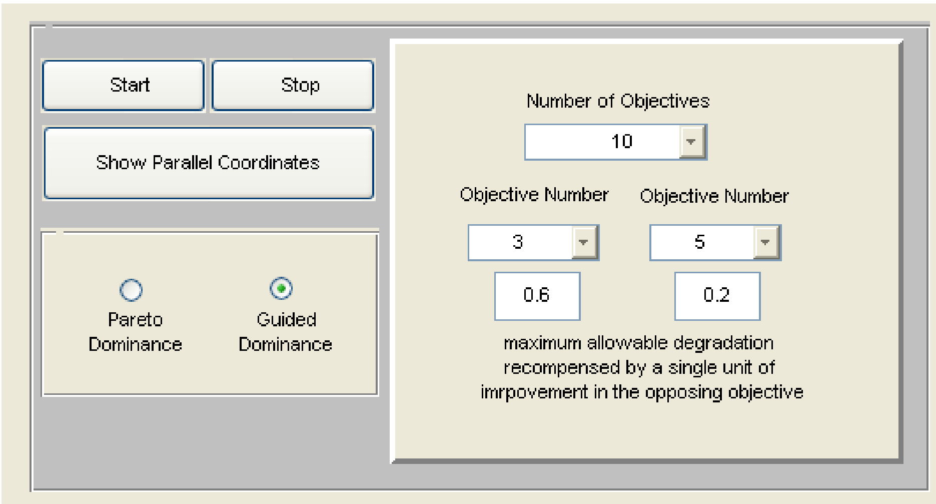


Figure 5. 6 GUI of the Guided dominance based PPA technique

Figures 5.7-5.10 illustrate the results achieved by G-MOEA for the convex test functions ZDT1 for four different consecutive preferences. In Figure 5.7, the amount of degradation in terms of objective 2 that merits a unit of improvement in terms of objective 1 was set to the value 0.8 (m_{21}= 0.8), i.e. four times bigger than the amount of degradation in terms of objective 1 which was deemed worthy to be compensated by a single unit of improvement in terms of objective 2 (m_{12}= 0.2). The articulated preferences were therefore favouring objective 1. Within 200 generations of the search process, the bias in terms of objective 1 was observed. The achieved region of the Pareto optimal front was bounded by the solutions whose tradeoff functions are tangent to the Pareto front.

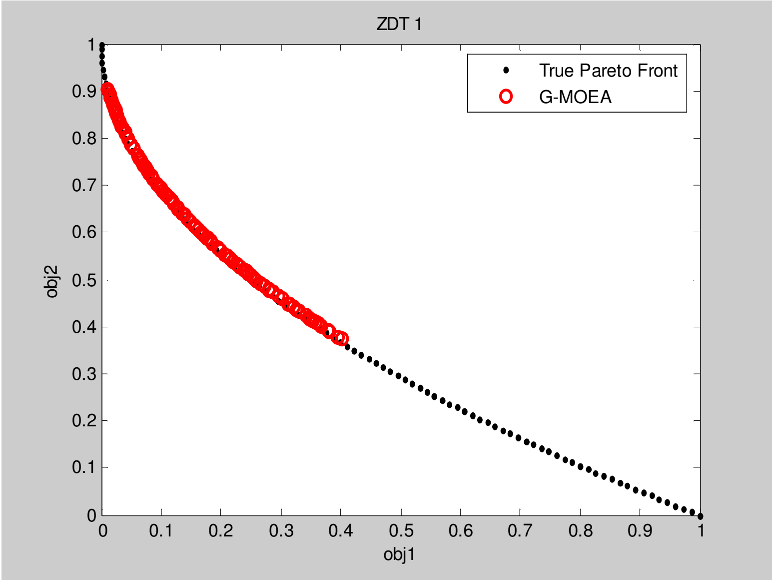


Figure 5. 7 G-MOEA Running on ZDT1 (m_{12} = 0.2, m_{21}= 0.8), 200th generation

In Figure 5.8, the preferences previously articulated were reversed. The new intention was to prioritize objective 2. The amount of degradation in terms of objective 1 that merits a unit improvement in terms of objective 2 was increased to the value 1.4, therefore favouring objective 2. On the other hand, the permissible amount of degradation in terms of objective 2 was reduced to the value 0.1. Within 50 more generations, the new preferences were manifested by the achieved results which populated the lower part of the convex Pareto front and which favoured objective 2.

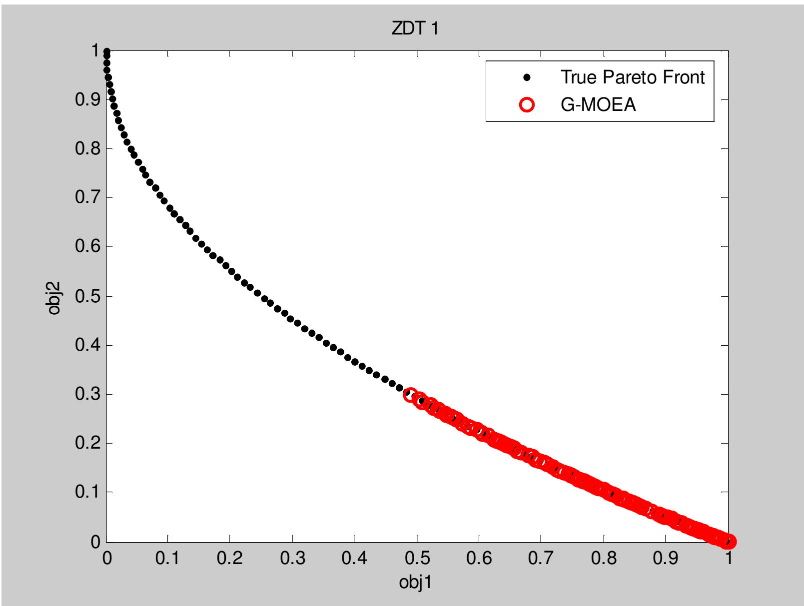


Figure 5. 8 G-MOEA Running on ZDT1 (m_{12} = 1.4, m_{21}= 0.1), 250th generation

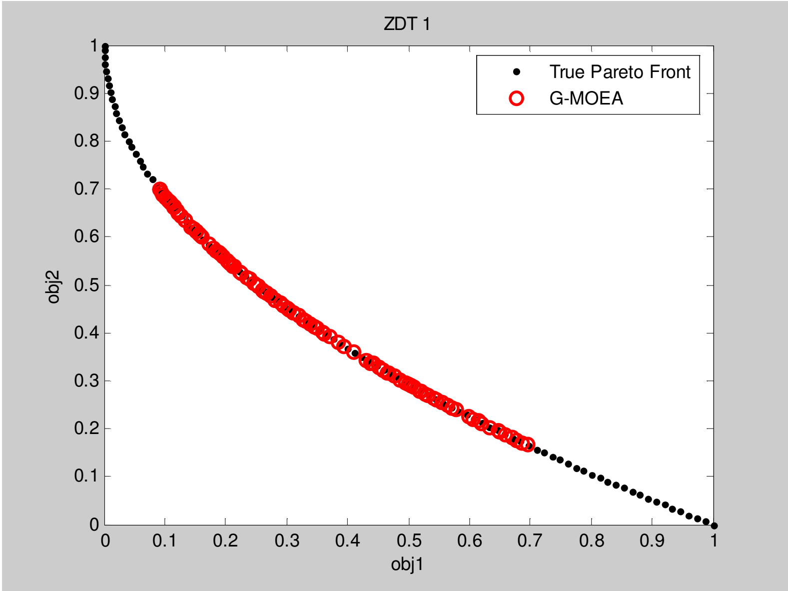


Figure 5. 9 G-MOEA Running on ZDT1 (m_{12} = 0.7, m_{21}= 0.7), 300th generation

In Figure 5.9, there were no preferences between the 2 objectives. Instead, the middle part of the Pareto optimal front was sought. The ROI was therefore decided vaguely by choosing an equal maximal amount of acceptable degradation for the 2 objectives when the other objective improves by a single unit (m_{12} = m_{21}= 0.7). The bounds of the decision maker's ROI could not be simply expressed; instead there was a need for an intermediate translation of the DM preferences into line slopes that delimit the desired ROI. Nonetheless, the desired -vague- ROI was efficiently emphasized at the 300th generation of the optimisation process. Finally, the amount of degradation in terms of the two objectives, which can be redeemed by a single unit of improvement in terms of the other objective, was increased furthermore at the 300th generation (m12 = m21= 0.9 in Figure 5.10). There was still no priority preference between the two objectives, but the aim was now to achieve a smaller ROI on the middle part of the Pareto front. At the 350th generation, the produced results were conforming to the vague preference of achieving a smaller part of the middle region of the Pareto front, compared to the precedent articulated preference.

Overall, it was not a straightforward method from the DM's point of view to execute a specific optimisation and a detailed search scenario. The guided dominance was better suited for efficiently articulating vague preferences. Moreover, a simple PPA scenario based on the guided dominance technique was performed on an optimisation problem with 4 competing objectives. The optimisation problem consisted of a 4-objective version of the scalable test function DTLZ2. In Figures 5.11 and 5.12, the results achieved by G-MOEA at the 200th and the 300th generation are illustrated respectively.

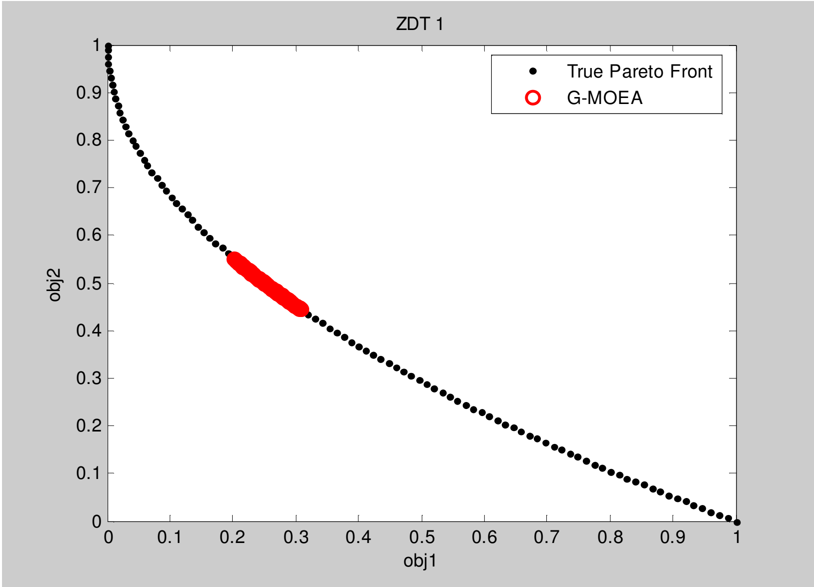


Figure 5. 10 G-MOEA Running on ZDT1 (m_{12} = 0.9, m_{21}= 0.9), 350th generation

Each of the four quadrants in Figures 5.11 and 5.12 illustrated the values achieved for a certain objective. The red and the black horizontal lines in each quadrant correspondingly presented the median and the mean value of a certain objective. Moreover, as an alternative to the results' illustration technique adopted and presented in Figures 5.11 and 5.12 (and later used in Sections 5.3.2, 5.3.3 and 5.3.4 to illustrate the results achieved for DTLZ2 (4) by the remaining PPA techniques discussed in this Chapter), parallel coordinates were also used for presenting the results achieved for the 4 objectives DTLZ2 test function. The parallel coordinates graphs are presented in Appendix D. Using the guided dominance, it is only possible to focus on a certain ROI by articulating desired quantitative tradeoffs between pairs of objectives. The results achieved in Figure 5.11 were affected by the pair wise tradeoffs presented in Table 5.2. The values of the articulated tradeoffs denoted the maximum value of deterioration in terms of a certain objective 'i' (row index), which can be deemed worthy of a single unit of improvement in terms of objective 'j' (column index).

Table 5. 2 Maximum deterioration tradeoffs between Objective 'i' and Objective 'j'

	j			
i	Objective 1	Objective 2	Objective 3	Objective 4
Objective 1	0	0.1	0.1	0.2
Objective 2	0.7	0	0.3	0.7
Objective 3	0.7	0.3	0	0.7
Objective 4	0.2	0.1	0.1	0

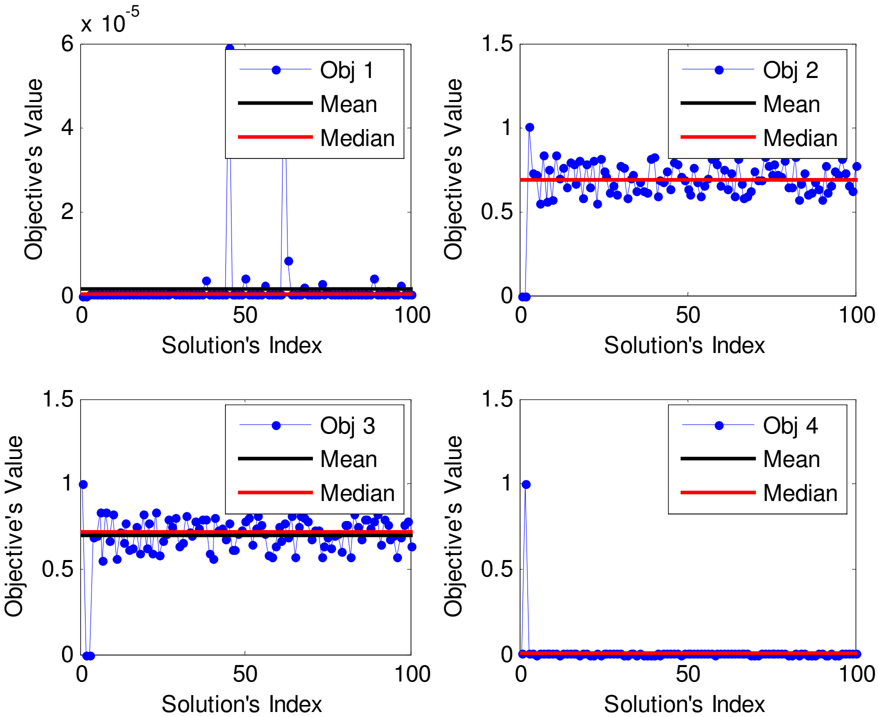


Figure 5. 11 The achieved results in terms of each objective (200th generation)

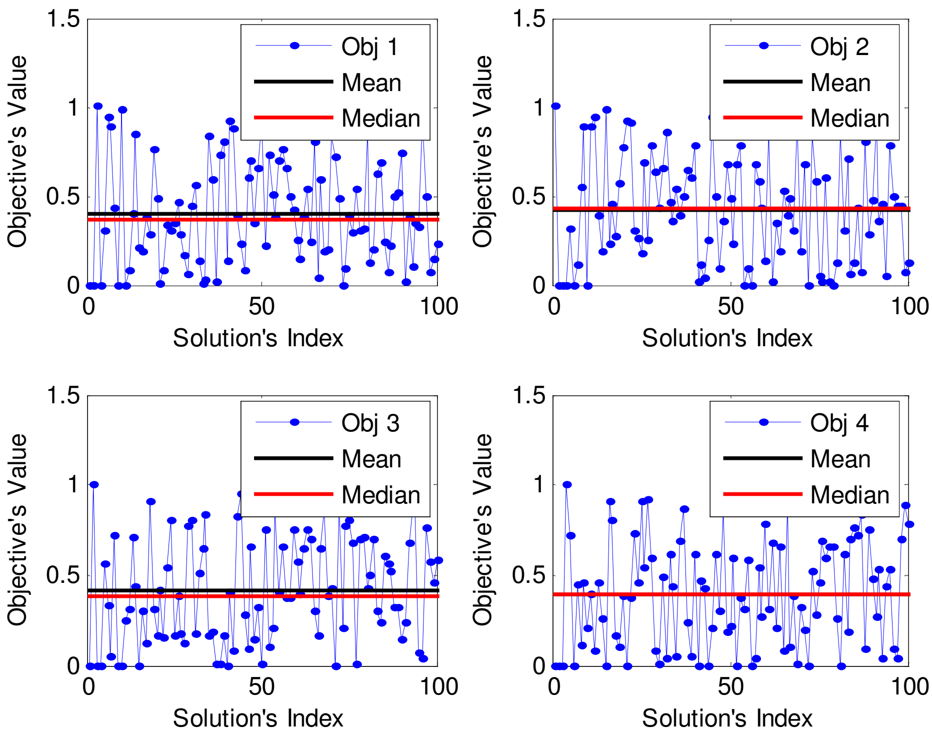


Figure 5. 12 The achieved results in terms of each objective (300th generation)

The maximum acceptable amount of degradation in terms of objectives 2 and 3 was 7 times larger than its counterpart in terms of objectives 1 and 4. Overall, objectives 1 and 4 were deemed equally important and were prioritized over objectives 2 and 3. The results achieved at the 200^{th} generation of G-MOEA were conforming to the articulated tradeoff values. Noting that the 4 objectives were commensurate, the median values achieved for objectives 1 and 4 presented a near optimal value (zero being the optimal value) alongside a mean value less than 0.1. On the other hand, the median and the mean values achieved for objectives 2 and 3 surpassed the value 0.5. At the 200th generation of the optimisation process, the DM's preferences were interactively altered. The new preferences consisted of a balanced optimisation scenario allocating the same priority to the four objectives. All the pair wise tradeoff values presented in Table 5.2 were set to the value 0.1. The results achieved at the 300th generation of the process and affected by the newly articulated preferences are illustrated in Figure 5.12. The mean and the median values achieved for the four objectives resided in the same vicinity and illustrated comparable and intermediate values close to 0.5.

5.3.2. Demonstration of the Biased Crowding as a PPA technique

The biased crowding measure was integrated in NSGA-II in order to assess its performance as a progressive preference articulation technique. NSGA-II used the same configuration deployed in Branke and Deb (2004): (i.e. population size = 100, SBX crossover (probability = 1, distribution parameter = 10) and a variable-wise polynomial mutation (probability = 1/n, n = number of decision variables, and distribution parameter = 20)).

Moreover, NSGA-II was equipped with a progressive capability allowing the decision maker to interactively articulate and modify his/her central direction of interest in the objective space (direction vector η). The decision maker was also provided with the facility of progressively articulating the parameter α in order to control the accuracy of the produced results. Once articulated by the decision maker, the direction vector η, which constitutes a linearly weighted utility function, and the control parameter α were fed to the biased crowding operator to adapt to the new preferences and bias the selection for recombination and survival processes conveniently. In Figure 5.13, a graphical user interface designed to accept any number of objectives is illustrated. The GUI is suggested to assist the decision maker with the progressive preference articulation and facilitate the use of the biased crowding operator. Additionally, the GUI was equipped with the utility of plotting the parallel coordinates graph which composes one of the most efficient techniques for visualising the interactions between the objectives in a high dimensional space. In optimisation problems with more than three objectives, the parallel coordinate graph is believed to assist the decision maker in articulating the direction vector η and formulating vague preferences.

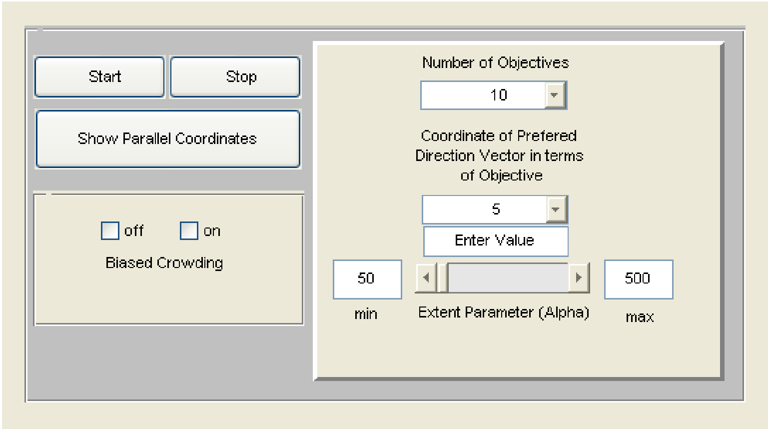


Figure 5. 13 GUI of the biased crowding based PPA technique

In Figures 5.14-5.17 correspondingly, a progressive articulation of a ROI was expressed for the convex test function ZDT1. This was performed by interactively modifying the direction of the projection line and the control parameter α, which are the basis of the biased crowding technique. In Figure 5.14, the results achieved at the 200th generation by NSGA-II deploying the biased crowding operator (NSGA-II/Biased Crowding) are illustrated by the red circles. The decision maker's preferences consisted of a vague interest in exploring solutions on the middle part of the convex Pareto front, using the direction vector η (1, 1) and an intensity bias parameter α = 200. At the 210th generation, the decision maker progressively articulated another vague preference, this time pointing in the direction of the upper part of the convex Pareto front. The coordinates of the direction vector η were therefore modified from (1, 1) to (1, 0.2).

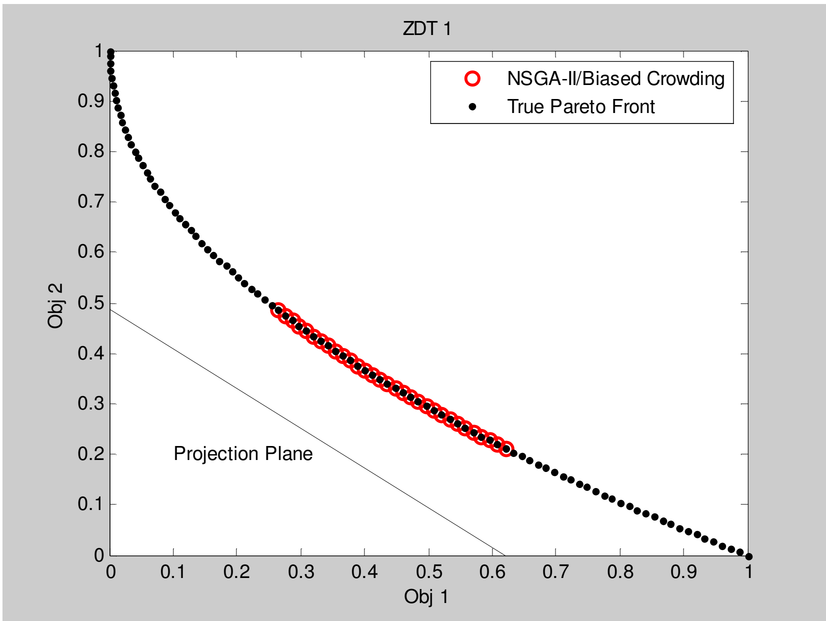


Figure 5. 14 NSGA-II/Biased Crowding running on ZDT1 (η = (1, 1), α=200)

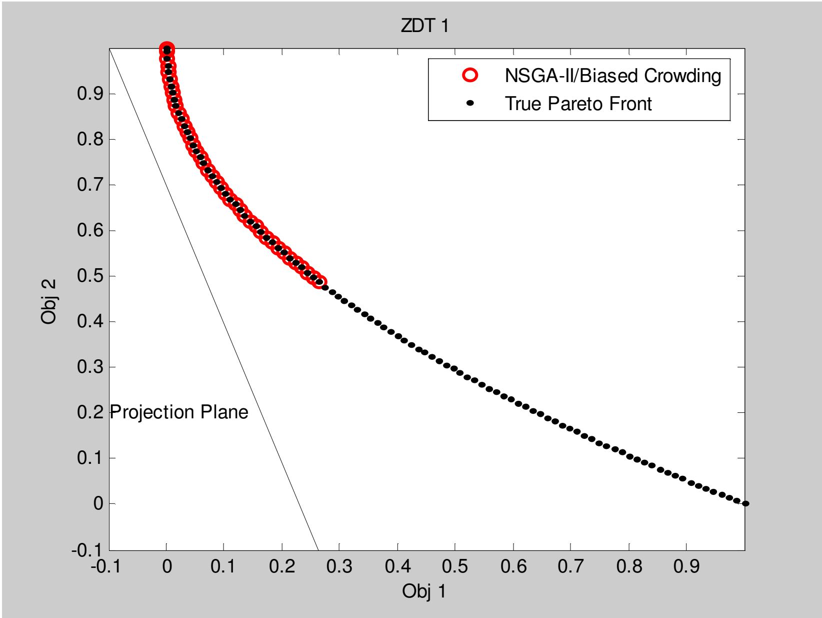


Figure 5. 15 NSGA-II/Biased Crowding running on ZDT1 (η = (1, 0.2), α=100)

The modified direction vector η therefore now constitutes the directed line joining the point with the coordinates (1, 0.2) and the axes origin (0, 0) and which is perpendicular to the projection (hyper)-plane illustrated for each scenario. The corresponding α parameter for the new scenario in Figure 5.15 was also reduced (α = 100) conveying a smaller bias intensity. In Figure 5.16, the DM articulated a new vague preference at the 300th generation of the same optimisation process. This time the requested ROI consisted of the lower part of the Pareto front while maintaining the same bias intensity (η = (0.2, 1), α = 100). The final part of the PPA scenario is illustrated in Figure 5.17, where the direction vector and the bias intensity had the following values respectively: (η = (1, 0.8), α = 500).

The increased value of the parameter α denoted a request for a higher bias intensity in the direction of the vector η. This was reflected by the produced results in Figure 5.17. The PPA utility of the biased crowding technique is then assessed on an optimisation problem with 4 competing objectives using DTLZ2 (4). In Figure 5.18, the mean and the median values achieved for each of the 4 competing objectives are illustrated at the 100th generation. The decision maker’s preferences were articulated at the first generation of the optimisation process. Noting the limitation of the biased crowding approach presented by its unsuitability for articulating exact preferences, the DM articulates an equal importance to the 4 objectives by devising a direction vector η with the coordinates (1, 1, 1, 1). The control parameter α was set to the value 100. Despite some, un-requested, minor bias in terms of objectives 2, 3 and 4 presented by the smaller median and mean values (≤ 0.4), the results presented in Figure 5.18 illustrated mean and median values in the range [0.4, 0.51] for all of the four objectives. At the 100th generation, the direction vector η was interactively modified to express a new direction of interest. The coordinates of η were set to (0, 1, 1, 0) and denoted a higher priority for objectives two and three.

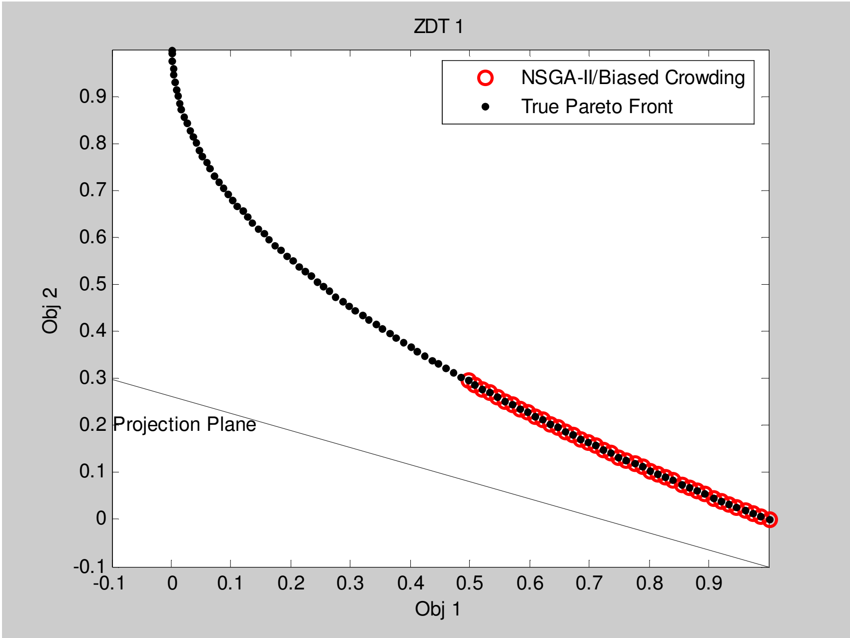


Figure 5. 16 NSGA-II/Biased Crowding running on ZDT1 (η = (0.2, 1), α=100)

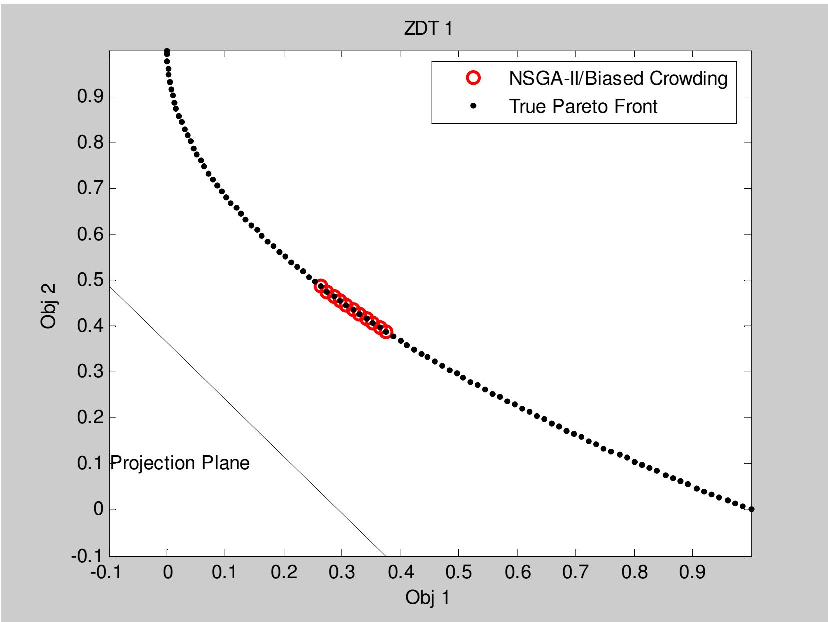


Figure 5. 17 NSGA-II/Biased Crowding running on ZDT1 (η = (1, 0.8), α=500)

Figure 5.19 illustrate the results achieved at the 200th generation of the optimisation process. Objectives 2 and 3 presented mean and median values lower than 0.4. One more time, an unexpected bias in terms of Objective 3 was observed with a median value less than 0.2. At the same time, the mean and median values achieved for the remaining two objectives (1 and 4) were larger than 0.4 as a direct response to their expressed lowered priorities.

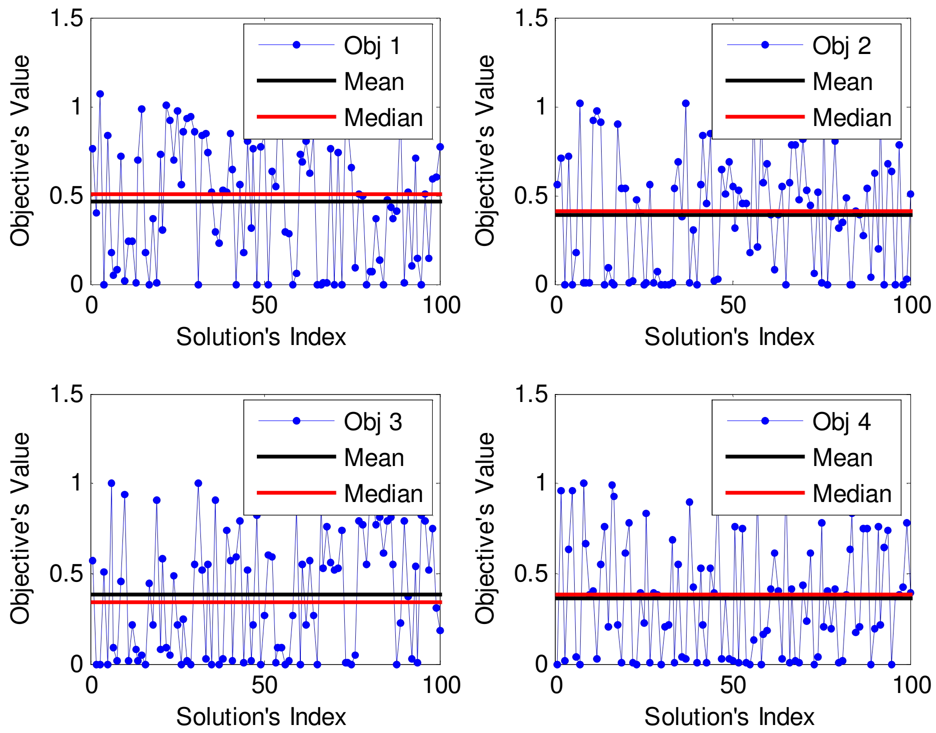


Figure 5. 18 DTLZ2 (4) Results achieved at the 100th generation (η = (1, 1, 1, 1), α = 100)

Finally, at the 200th generation, the preferences expressed at the 100th generation were reversed. Objectives 1 and 4 were now prioritized over objectives 2 and 3. The new interactively articulated direction of interest was articulated by modifying the coordinates of the direction vector η from (0, 1, 1, 0) to (1, 0, 0, 1). As a result, the mean and the median values of objectives 2 and 3 were deteriorated and increased towards the values 0.5 and 0.4 respectively (Figure 5.20). The mean and median values of objectives 1 and 4 were on the other hand improved and decreased. In particular, the median value achieved for objective 4 was significantly improved. Overall, the results achieved for the 4 objectives problem conformed to the vague DM's preferences. Because the biased crowding technique was designed for articulating vague preferences rather than exact preferences, the unexpected responses to exact preferences and the unpredictable biases in terms of certain objectives is explicable. Nevertheless, the utility of the biased crowding was shown to be particularly beneficial for articulating vague preferences and fulfill its intended purpose. The results presented in Figures 5.18, 5.19 and 5.20 are also illustrated using parallel coordinates graphs in Appendix D.

For these continuous well-shaped Pareto fronts (DTLZ2), the biased crowding operator seemed to perform well. However, when dealing with highly multimodal, discontinuous and mathematically ill-behaved Pareto fronts, setting the direction vector η can be a tricky process to the decision maker, especially in high dimensional search spaces which are difficult to visualise. As a result, when optimising a problem with an unknown or multimodal Pareto front, the decision maker might articulate a certain direction vector η which points to infeasible areas of the objective space or even to regions of the search space which differs from the decision maker's interest. On the other hand, despite its efficiency in shifting the search focus to certain preferred regions in the objective space, the biased crowding operator is not a suitable technique for reducing the size of the search space (which is particularly useful in scenarios comprising many competing objectives) as it does not really enforce preferences or manipulate the underlying Pareto dominance concept.

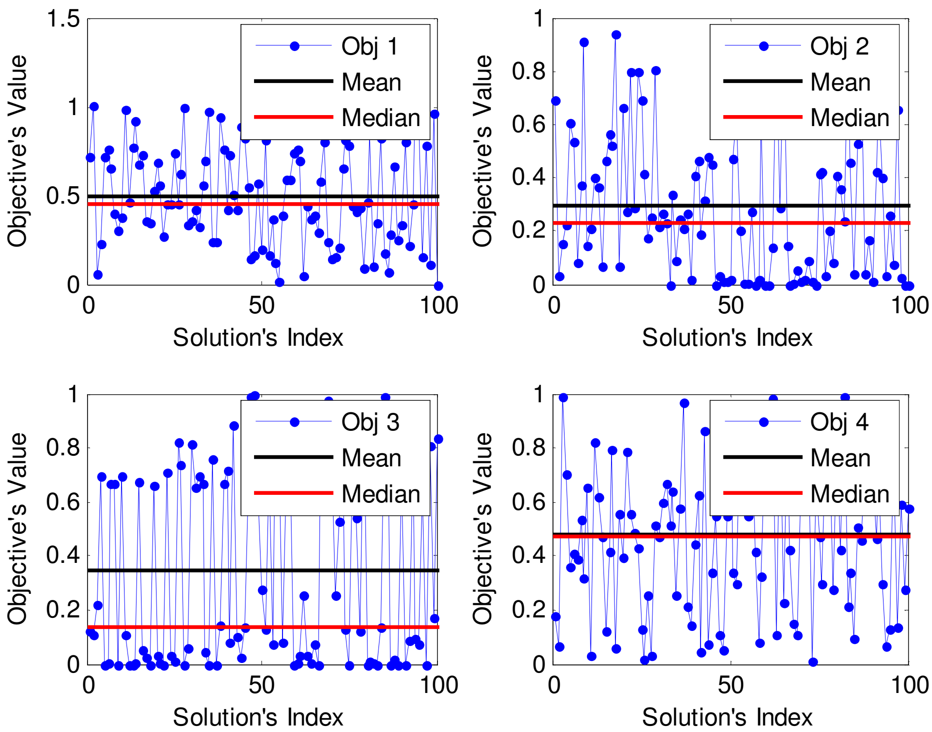


Figure 5. 19 DTLZ2 (4) Results achieved at the 200th generation
(η = (0, 1, 1, 0), α = 100)

The biased crowding operator artificially manipulates the density estimate around the current solutions in the objective space in a way that biases the solutions residing near the vague region of interest. Density estimates are usually incorporated in MOEAs for promoting diversity among the solutions of a multiobjective problem. However, the diversity measure itself is a secondary selection criterion in most MOEAs including NSGA-II. Hence, in many occurring scenarios, the diversity discrimination process can be obscured and phased out from the selection for recombination and/or the selection for survival procedures. For example, the deactivation of the diversity criteria, and therefore the effect of the crowding/biased crowding operators, occurs if the mating pool and/or the online archive are completely filled with solutions which are picked based on their non-domination level. In other words, the online archive or the mating pool is filled without necessitating the selection of a subset of solutions, which belong to the same non-domination level.

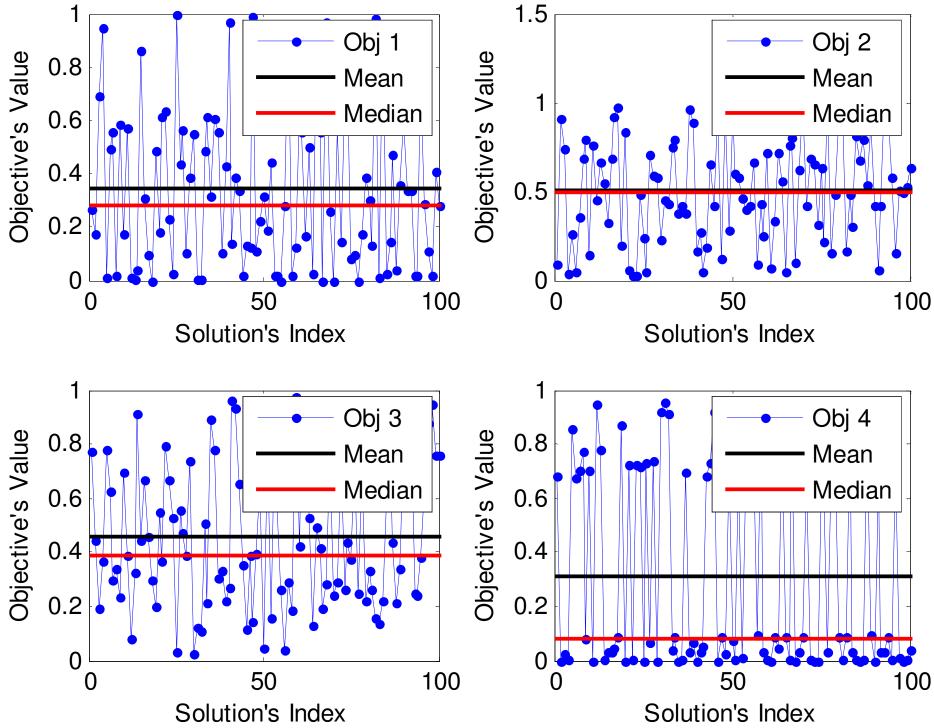


Figure 5. 20 DTLZ2 (4) Results achieved at the 300th generation
(η = (1, 0, 0, 1), α = 100)

Additionally, despite focusing on the areas of the Pareto optimal front which are parallel to the projection plane with the normal vector η, the biased crowding does not necessarily bias the solutions residing in the (hyper)-areas of the objective space which are logically bounded by the DM's vague preferences. This is illustrated in Figure 5.21, where the solutions D, E and F are biased against using the biased crowding.

Despite being in the preferred area of the objective space, the solutions D, E, and F are aligned in a near perpendicular way to the projection plane. Hence, the crowding measure of the solution E, for example, will be much bigger than the crowding measure based on the location of the projected images of E and its neighboring solutions D and F on the projection plane (solutions D', E' and F'). Solution E will be consequently biased against, as it would be considered artificially crowded using the biased crowding. On the other hand, solutions A, B and C will be favored for selection despite populating a non-desired region of the objective space. This is due to their alignment which is more closely parallel to the projection plane compared to the alignment of the solutions D, E and F. The effect of the biased crowding operator is therefore more efficient and practical around near optimal areas of the objective space, ideally when the achieved solutions start taking the form of the Pareto front.

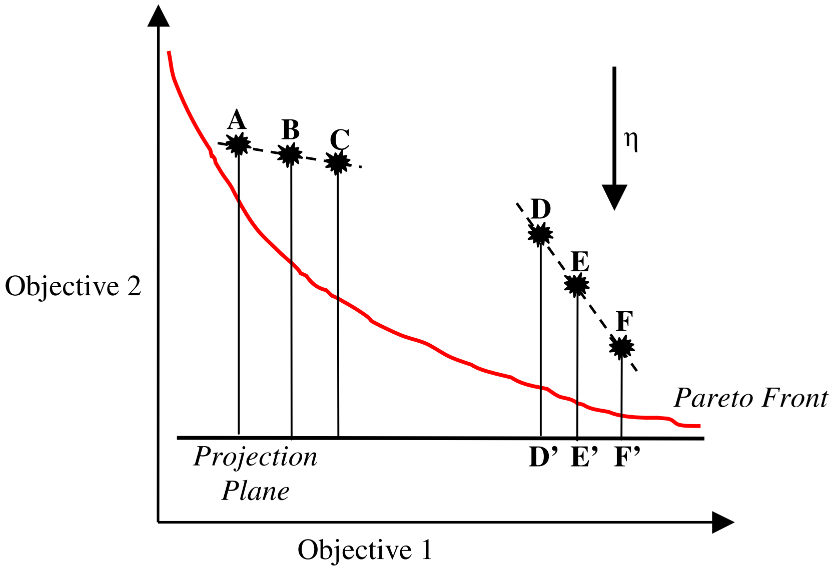


Figure 5. 21 Projection of the Solutions following the direction vector η

5.3.3. Demonstration of ε –Dominance as a PPA technique in the ε –MOEA Context

The ε-dominance concept is originally designed for specifying the precision of the required set of solutions to a multiobjective optimisation problem. The concept is also beneficial for reducing undesirable MOEA behaviours (oscillatory search process and divergence from the Pareto front in high dimensional objective spaces) usually caused by the strict Pareto dominance scheme. The precision specification process can be realized in terms of each objective separately by (linearly or logarithmically) rescaling the corresponding axis of performance. This is achieved by choosing a different epsilon value $\boldsymbol{\varepsilon_i}$ for each objective (*$1 \leq i \leq m$, where m is the total number of objectives*). The resulting vector of ε values denotes the tolerance in terms of each objective below which two values are deemed insignificant to the DM. The objective space is consequently divided into hyper-boxes whose sizes in the i^{th} objective are determined by the corresponding tolerance value $\boldsymbol{\varepsilon_i}$. This concept is illustrated in Figure 5.22 on a simple bi-objective problem with a convex Pareto front. Reducing the values of ε_1 and ε_2 in Figure 5.22 should ideally result more solutions being produced on the Pareto front, therefore controlling the solutions' distribution.

In this Section, the potentiality of the ε-dominance as a PPA technique is examined. The aim is to slightly alter the ε-dominance concept deployed in ε-MOEA such that the overall resulting technique allows the DM to reduce the dimensionality of the search space and stress out a certain ROI. Despite frequently stating its potential for articulating objective preferences in the literature, the ε-dominance concept is widely used in the EMO community to pre-fix a certain tolerance threshold in terms of a certain objective. Moreover the same ε value is usually specified for all of the objectives and used to determine the precision and the distribution of the final results without really exploiting the ε-dominance concept to articulate objective preferences or regions of interests.

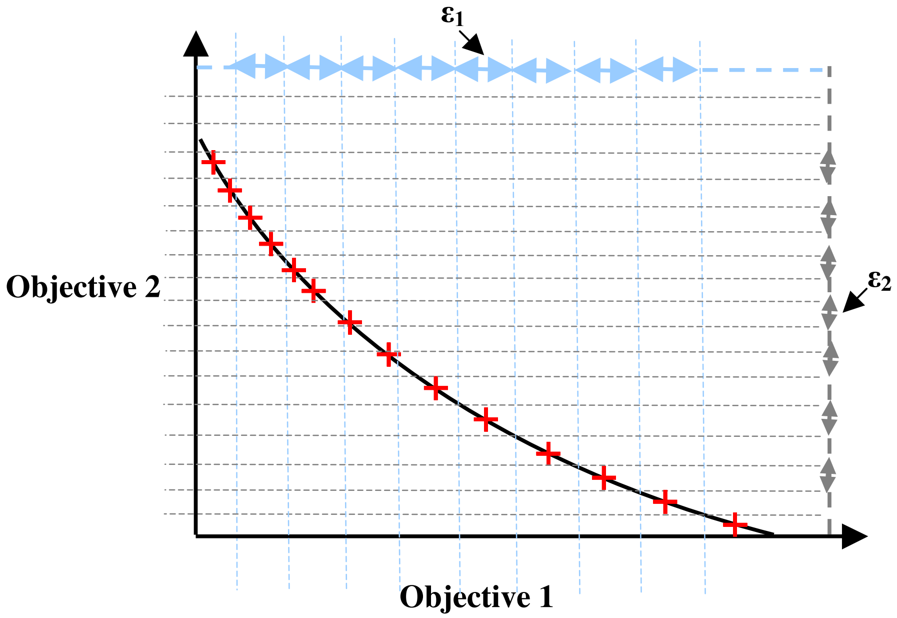


Figure 5. 22. ε-dominance defining the acceptable tolerance in each objective

A strategy similar to the ε-dominance approach used in ε-MOEA is suggested for articulating the decision maker preferences. Using the modified strategy, the axis of performance for each objective is divided into several contiguous ranges specified by the DM, each having its own ε value. In other words, instead of specifying a single fixed ε value for each objective, several ε values specified over a certain specific (vague or exact) range of performance are assigned for each objective. The number of performance's ranges in terms of each objective and their corresponding boundaries should be constructed in a way that reflects the decision maker's preferences. For example, in an optimisation problem consisting of **n** objectives (**n** is any positive integer) defined over the range **[min, max]** (**min** and **max** correspond to the minimum and the maximum possible value for each objective), the decision maker might decide that he/she is only interested in finding solutions with the following constraints:

*(In the following example, **n = 3, min = 0** and **max = 10** for all 3 objectives)*

- The values in terms of objective 1 lie in the ranges [2, 4] and [6, 8],
- The values in terms of objective 2 lie in the range [2, 8], and
- The values in terms of objective 3 are less than or equal to 5.

The previous decision maker's preferences are illustrated graphically in Figure 5.23 where $\mathbf{R_{i,j}}$ denotes the $\mathbf{j^{th}}$ range for the $\mathbf{i^{th}}$ objective and $\boldsymbol{\varepsilon_{i,j}}$ is the corresponding $\boldsymbol{\varepsilon}$ value for the $\mathbf{i^{th}}$ objective over the $\mathbf{j^{th}}$ range. The ranges highlighted in red constitute the regions of interest to the DM and therefore should be provided by the DM alongside their corresponding $\boldsymbol{\varepsilon_{i,j}}$ values which denote the required distribution and precision over each range. In *exact* preference scenarios where the DM provides exact ranges of interests, and is exclusively interested in his/her articulated ranges of interest in terms of each objective, the ε values for the remaining ranges such as $\boldsymbol{\varepsilon_{1,1}}$ or $\boldsymbol{\varepsilon_{2,1}}$ in Figure 5.23 are assigned the value zero (i.e. $\varepsilon_{1,1} = \varepsilon_{1,3} = \varepsilon_{1,5} = \varepsilon_{2,1} = \varepsilon_{2,3} = \varepsilon_{3,2} = 0$).

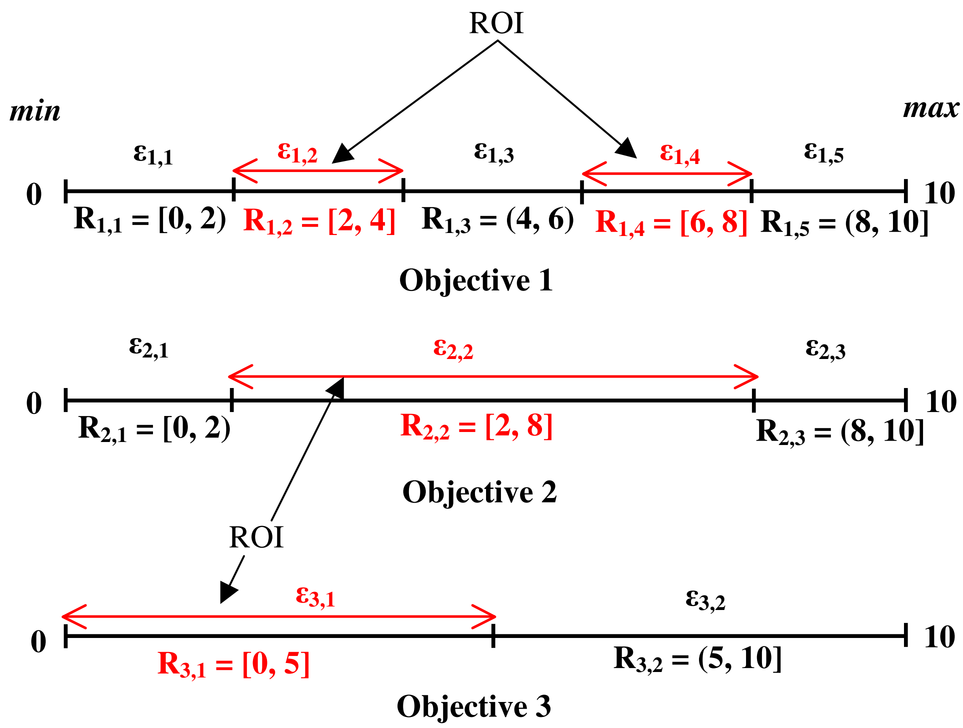


Figure 5. 23. The DM preferences using the ε-dominance strategy

On the other hand, when articulating *vague* preferences, the ε values corresponding to the non-preferred ranges such as $R_{1,1}$, $R_{2,1}$ or $R_{3,2}$ in Figure 5.23 should correspond to large tolerance values (*compared to the tolerance values of the ranges of interest*) decided by the DM and strictly defined within the boundaries of their corresponding ranges. Similar to the strategy used in Deb, Mohan and Mishra (2003), the archive acceptance process is applied on every candidate solution considered for inclusion in the archive after determining its corresponding identification array. The only difference is that the identification array for a certain solution **'y'** is now determined using the local attributes (f_i^{min} and $\boldsymbol{\varepsilon}$ value) of the corresponding ranges of performance to which each of its objective values belongs.

However, when solving an *exact* preference scenario, a slightly different process is used to determine the coordinates of the identification array of a certain solution **'y'** presenting some objective values in the non-preferred ranges of performance (i.e. with a corresponding ε value equal to zero). This is illustrated in the objective space of a bi-objective optimisation problem (Figure 5.24) without any loss of generality. The ROI in Figure 5.24 is the intersection of the ranges $\mathbf{R_{1,2}}$ and $\mathbf{R_{2,2}}$. The identification array of a certain solution **'y'** will have its coordinates in terms of the objectives whose values lie in the non-preferred ranges defined as follows:

1- Any solution **'y'** residing in $\mathbf{R_{2,1}}$ (with $\boldsymbol{\varepsilon_{2,1}} = 0$) will have an identification array whose coordinate in terms of objective 2 is the lower boundary of $\mathbf{R_{2,1}}$. Its coordinate in terms of objective 1 is locally specified using the corresponding attributes of the range $\mathbf{R_{1,j}}$ where the solutions **'y'** resides.

2- Any solution **'y'** residing in $\mathbf{R_{2,3}}$ (with $\boldsymbol{\varepsilon_{2,3}} = 0$) will have an identification array whose coordinate in terms of objective 2 is the upper boundary of $\mathbf{R_{2,3}}$. (The calculation of its coordinates in terms of objective 1 is similar to case 1 above).

3- Any solution **'y'** residing in $\mathbf{R_{2,2}}$ (with $\boldsymbol{\varepsilon_{2,3}} \neq 0$) will have an identification array whose coordinate in terms of objective 2 is calculated using Equation 5.4 (where f_2^{min} corresponds to the lower boundary of $\mathbf{R_{2,2}}$ and $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon_{2,3}}$). (The calculation of its coordinates in terms of objective 1 is similar to cases 1 and 2 above).

4- Any solution **'y'** residing in $\mathbf{R_{1,1}}$ (with $\boldsymbol{\varepsilon_{1,1}} = 0$) will have an identification array whose coordinate in terms of objective 1 is the upper boundary of $\mathbf{R_{1,1}}$. Its coordinate in terms of objective 2 will be locally specified using the corresponding attributes of the range $\mathbf{R_{2,j}}$ where the solutions **'y'** resides.

5- Any solution **'y'** residing in $\mathbf{R_{1,3}}$ (with $\boldsymbol{\varepsilon_{1,3}} = 0$) will have an identification array whose coordinate in terms of objective 1 is the lower boundary of $\mathbf{R_{1,3}}$. (The calculation of the coordinates in terms of objective 2 is similar to case 3 above)

6- Any solution **'y'** residing in $\mathbf{R_{1,2}}$ (with $\boldsymbol{\varepsilon_{1,2}} \neq 0$) will have an identification array whose coordinate in terms of objective 1 is calculated using Equation 5.4 (where f_1^{min} corresponds to the lower boundary of $\mathbf{R_{1,2}}$ and $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon_{1,2}}$). (The calculation of the coordinates in terms of objective 2 is similar to cases 4 and 5 above).

7- Finally, any solution **'y'** residing in the ROI (*between the two points A and B on the Pareto front*) defined by the intersection of $\mathbf{R_{1,2}}$ and $\mathbf{R_{2,2}}$ will have an identification array whose coordinates in terms of objectives 1 and 2 are locally specified by the procedure used in ε-MOEA for defining identification arrays (Equation 5.4) using the values $\boldsymbol{\varepsilon_{1,2}}$ and $\boldsymbol{\varepsilon_{2,2}}$ and the lower bounds of $\mathbf{R_{1,2}}$ and $\mathbf{R_{2,2}}$ respectively.

More generally, a solution **'y'** whose value in terms of a certain objective **'i'** resides in a range $\mathbf{R_{i,j}}$ (continuously) preceding a certain range of interest $\mathbf{R_{i,j+1}}$ for that objective will have an identification array whose coordinate in terms of the $\mathbf{i^{th}}$ objective is the upper boundary of $\mathbf{R_{i,j}}$. On the other hand, a solution **'y'** whose value in terms of a certain objective **'i'** resides in a range $\mathbf{R_{i,j}}$ (continuously) succeeding a certain range of interest $\mathbf{R_{i,j-1}}$ for that objective will have an identification array whose coordinate in terms of the $\mathbf{i^{th}}$ objective is the lower boundary of $\mathbf{R_{i,j}}$. The last case would be the case where a solution **'y'** presents a value in terms of a certain objective **'i'** which lies in a range $\mathbf{R_{i,j}}$ between two ranges of interest $\mathbf{R_{i,j-1}}$ and $\mathbf{R_{i,j+1}}$ for that objective. In such scenario, the identification array in terms of the $\mathbf{i^{th}}$ objective will be the boundary of $\mathbf{R_{i,j}}$ which is closest to the value of the i^{th} objective in **'y'**.

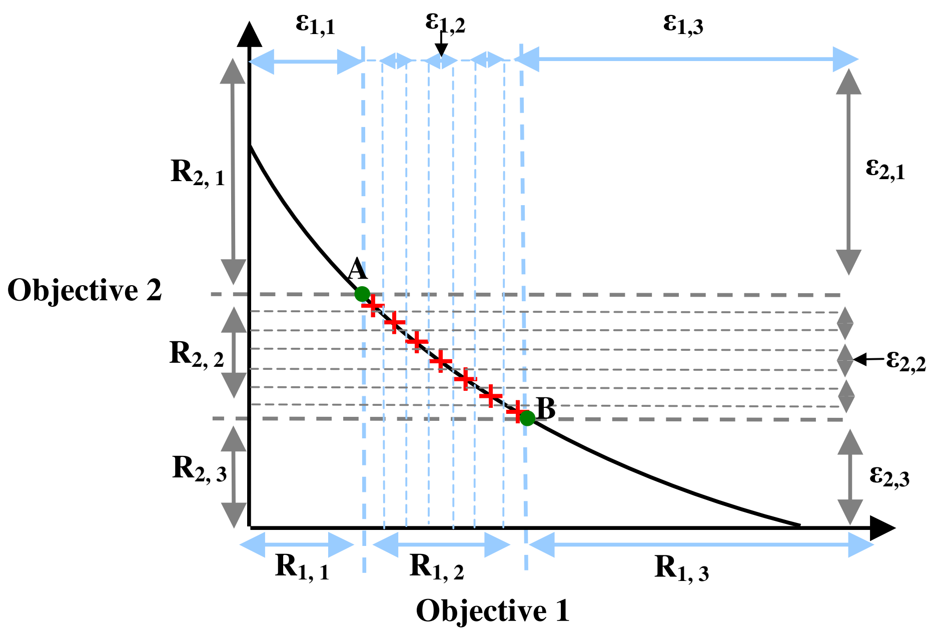


Figure 5. 24. Different ε-values for each objective over different ranges

The modified ε-dominance scheme previously described scales to any number of objectives and can be used to allow the DM to articulate regions of interests and objective priorities. Using the ε-dominance based PPA technique, the DM should define the (vague or exact) boundaries of the desired ROI in terms of each objective separately, alongside the required precisions in each objective.

Using the ε-dominance scheme, the progressive reduction of the ε resolution is achieved by merging neighbouring hyperboxes. However, it should be noted that progressive preference articulation techniques which are based on the ε-dominance scheme and coupled with a MOEA that maintains a fixed archive size[23] are known to present a major limitation. This limitation is presented by their inability of dealing with interactively

[23] A practical approach in real-life applications

reduced ε values (Laumanns, Thiele, Deb and Zitzler 2002a) which denotes the request for higher resolution compared to the initial ε specification. For example, at certain stages of the optimisation when the archive upper bound is reached, increasing the requested resolution might have no effect as this will require increasing the archive upper bound. A PPA framework, starting with the highest resolution required and interactively relaxing the resolution can address this limitation, but can however frequently cause the optimisation process to produce a single solution while merging neighbouring boxes in order to maintain the fixed archive size. As a remedial measure, Laumanns, Thiele, Deb and Zitzler (2002a) suggested the use of a multiple restart strategy to overcome this limitation.

In Figure 5.25, a suggested graphical user interface (GUI) is designed and implemented for the case of a 6 objective optimisation problem. Note that the GUI can be designed to automatically self-configure based on the number of objectives and the number of ranges required. The underlying optimisation process is the ε-MOEA procedure with the modified ε-dominance concept described above. Parallel coordinates (Inselberg 1985) are used to visualize the progress of the optimisation process and to assist the DM with the decision-making and the PPA processes.

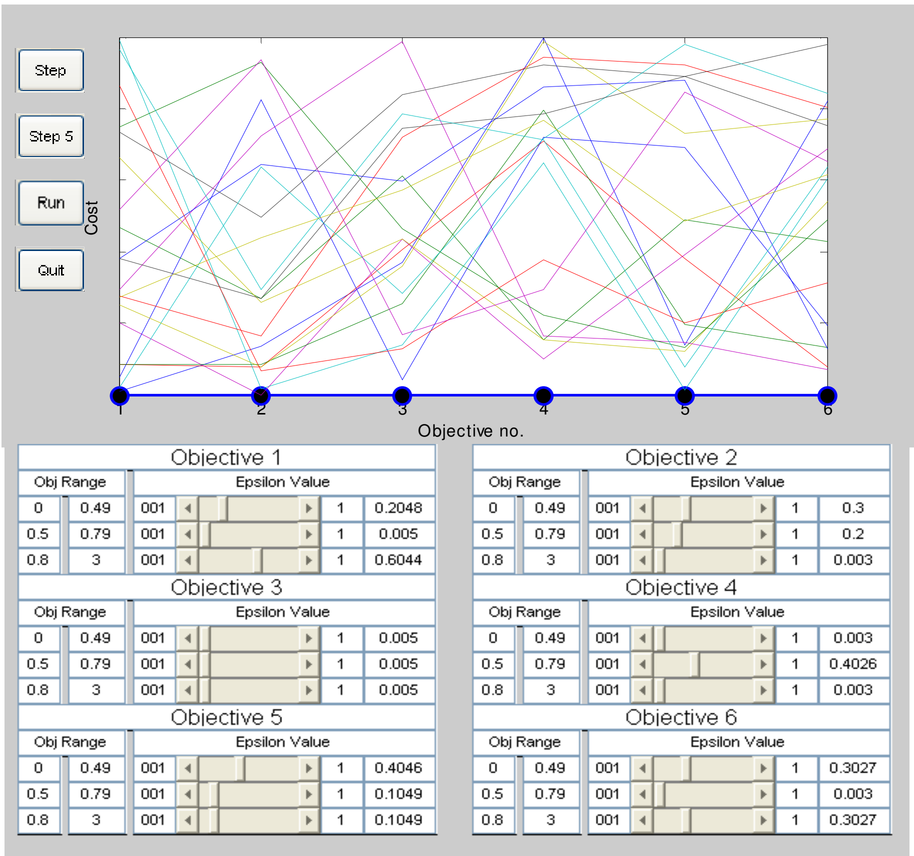


Figure 5. 25 GUI of the PPA technique based on the ε-dominance

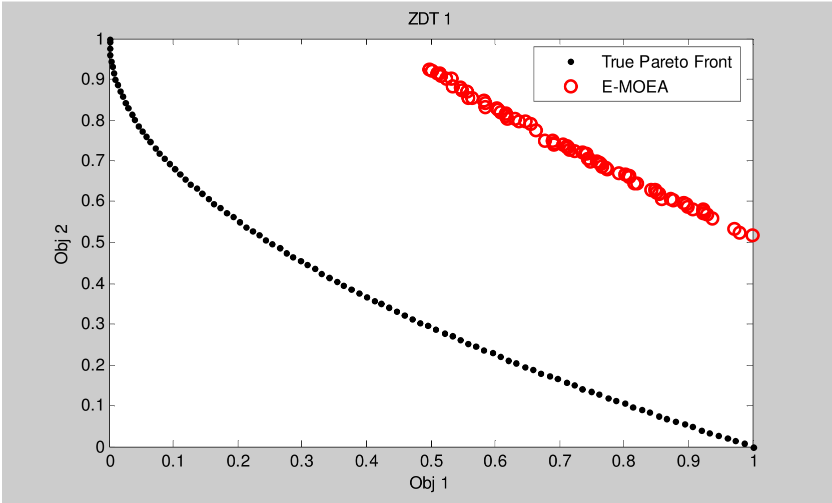


Figure 5. 26 ε –MOEA running on ZDT1 with $\varepsilon_{1,1}$ = 0, $\varepsilon_{1,2} = \varepsilon_2$ = 0.01 at generation 100

In Figures 5.26-5.29, the convex test function ZDT1 was deployed to investigate the utility of manipulating the ε–dominance concept in terms of each dimension separately as an attempt to simulate a progressive preference articulation scheme. The objective space was only divided into two continuous ranges in terms of objective 1. The two ranges $\mathbf{R_{1,1}}$ and $\mathbf{R_{1,2}}$ were correspondingly **[0, 0.5]** and **(0.5, ∞)**. The first preference articulation consisted of an exact scenario requesting solutions whose values in terms of objective 1 belong exclusively to $\mathbf{R_{1,2}}$. The articulated tolerance in terms of objective 1 in the range $\mathbf{R_{1,1}}$ was therefore nil (**$\varepsilon_{1,1}$ = 0**). Hence, any solution presenting an objective 1 value in the range $\mathbf{R_{1,1}}$ had an identification array whose coordinate in terms of the 1st objective had the value 0.5. On the other hand, the articulated tolerance in terms of objective 1 in the range $\mathbf{R_{1,2}}$ was 0.01 (**$\varepsilon_{1,2}$ = 0.01**) and it was equal to the tolerance value in terms of objective 2 (**ε_2**) which was defined over its whole range of definition **$\mathbf{R_2}$ = [0, ∞)**. Except for the ε values, in Figure 5.26-5.29 ε –MOEA was executed on ZDT1 using the same configuration used in Deb, Mohan and Mishra (2003). Starting with **$\varepsilon_{1,1}$ = 0** and **$\varepsilon_{1,2} = \varepsilon_2$ = 0.01**, it was clear that ε-MOEA was operating on a reduced search space bounded by the value 0.5 in terms of objective 1 (Figure 5.26). Within an additional 100 generations, the results achieved by ε-MOEA have converged to the desired ROI (Figure 5.27). However, in Figure 5.27, a new value for **$\varepsilon_{1,2}$** and **ε_2** was progressively articulated (**$\varepsilon_{1,2}=\varepsilon_2$=0.05**) indicating an increased tolerance over the ROI which resulted in the reduction of the number of solutions. In Figure 5.28, the value of **$\varepsilon_{1,1}$** was progressively modified from nil to 0.2 indicating a new *vague* preference of focusing on the previous ROI while obtaining some solutions in $\mathbf{R_{1,1}}$. The values of **$\varepsilon_{1,2}$** and **ε_2** were also both reduced to 0.02. The new articulated preferences at the generation 210 were manifested by the production of two solutions on the Pareto front in the range $\mathbf{R_{1,1}}$ reflecting the requested tolerance **$\varepsilon_{1,2}$** in the 1st objective (more than 0.2

units apart from each other). On the other hand, the number of solutions in $\mathbf{R}_{1,2}$ was increased following the tolerance reduction in this particular area of interest in the objective space. In the scenarios illustrated in Figures 5.26-5.28, the articulated ROIs reflected an overall priority of objective 2 over objective 1.

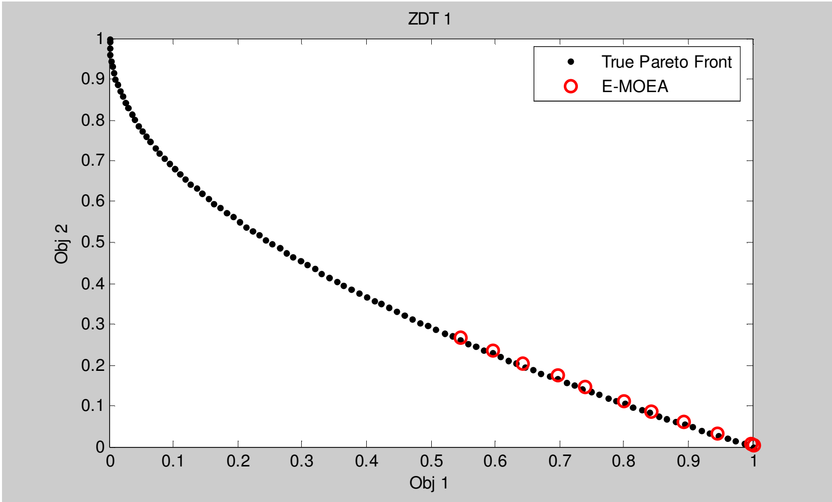


Figure 5. 27 ε –MOEA running on ZDT1 with $\varepsilon_{1,1}$ = 0, $\varepsilon_{1,2} = \varepsilon_2$ = 0.05 at generation 200

In Figure 5.29, the articulated ε values were all decreased to a value of 0.01 (**$\varepsilon_{1,1} = \varepsilon_{1,2} = \varepsilon_2$ = 0.01**), therefore eliminating the priority of objective 2 and articulating the preference of finding solutions all along the convex Pareto front. The results achieved at the 250th generation (Figure 5.29) were conforming to the desired preference which stated no priorities between the 2 objectives. Following the progressive preference articulation scenario presented in Figures 5.26-5.29, it was shown that the ε-dominance concept can be used as a PPA technique for articulating priorities among objectives, regions of interests and reducing the dimensionality of the search space.

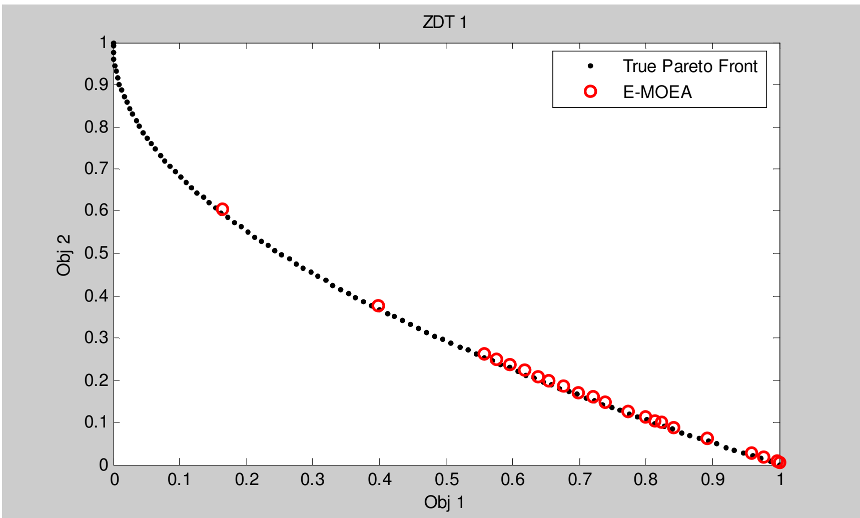


Figure 5. 28 ε –MOEA running on ZDT1 with $\varepsilon_{1,1}$ = 0.2, $\varepsilon_{1,2} = \varepsilon_2$ = 0.02 at generation 210

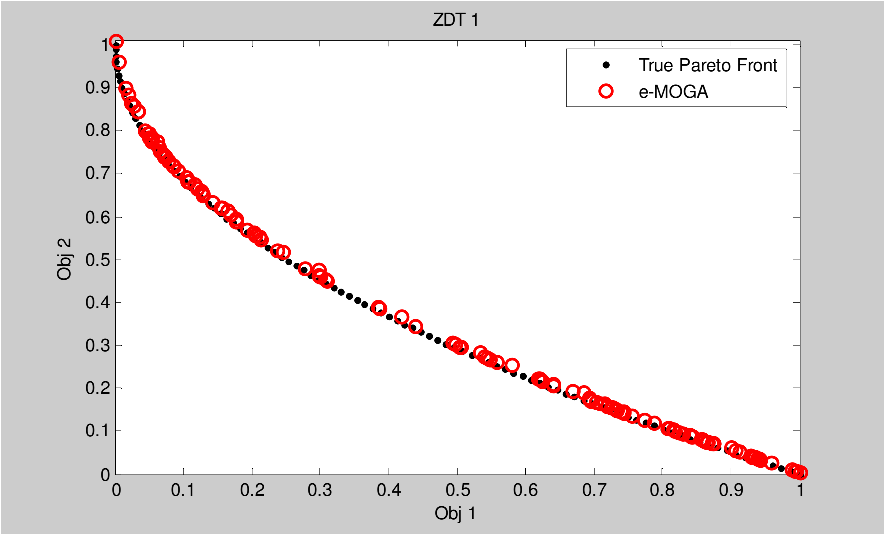


Figure 5. 29 ε –MOEA running on ZDT1 with $\varepsilon_{1,1} = \varepsilon_{1,2} = \varepsilon_2 = 0.05$ at generation 250

The described PPA technique allows the DM to control the selection for variation and the selection for survival strategies in a MOEA by manipulating the underlying Pareto dominance concept and hence the solutions' ranking process in a way translating the required preferences. Moreover, the ε-dominance based PPA technique does not require any assumption about the shape of the Pareto front or the search spaces. Therefore, from an algorithmic point of view, the technique is straightforward and well constructed and should have no difficulty in scaling to any number of objectives. However, for high number of objectives, this technique can be quite demanding as it requires the DM to specify the hyper-area of interest which is constituted by the intersection of many axis of performances. Therefore a certain vague knowledge about the topology of a high dimensional search space is required.

Visualising search spaces with high dimensionalities (more than 3 objectives) is a problematic issue by itself. In addition to its requirement for decomposing each objective's axis of performance into different ranges of performance, the ε-dominance PPA technique requires the DM to devise multiple 'weighting' values (ε values) defined over different ranges of performance for each single objective. Assigning quantitative weights to different objectives constituting an optimisation problem is a well-known difficulty that the DM usually faces when using the classical weighed sum approach. As a result, from a DM point of view, the PPA technique described in this section can be more demanding than the traditional weighted sum approach in certain scenarios, especially as the number of objectives increases.

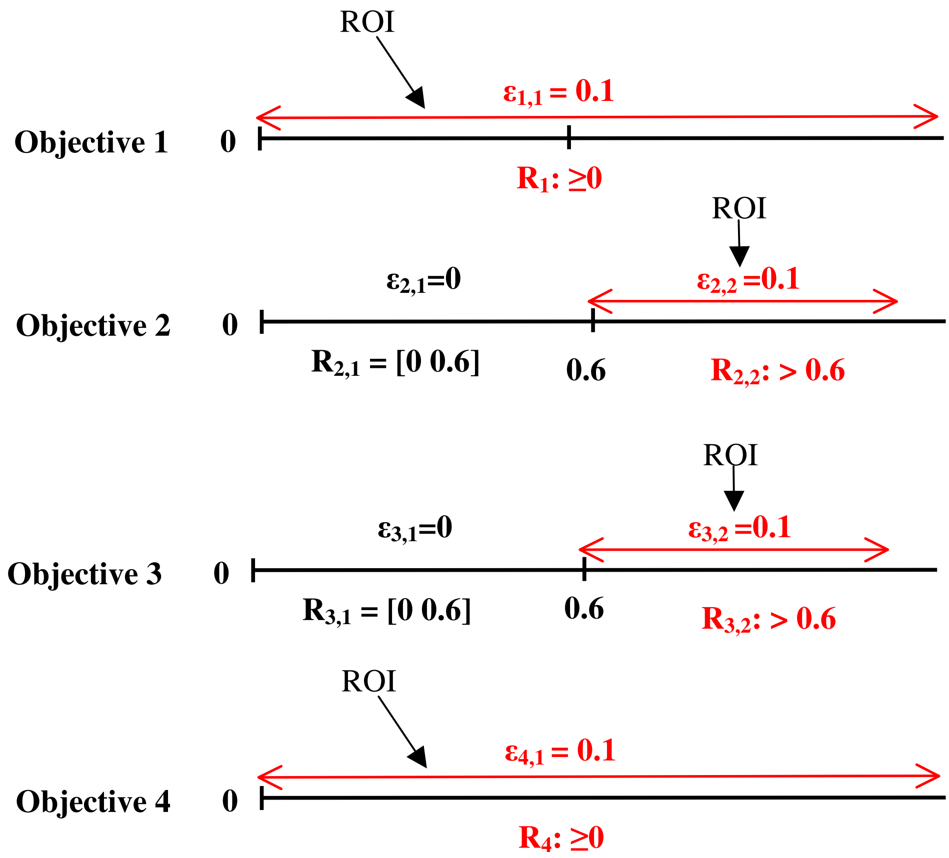


Figure 5. 30 DTLZ2 (4) Preferences of the First Scenario (after the 200th generations)

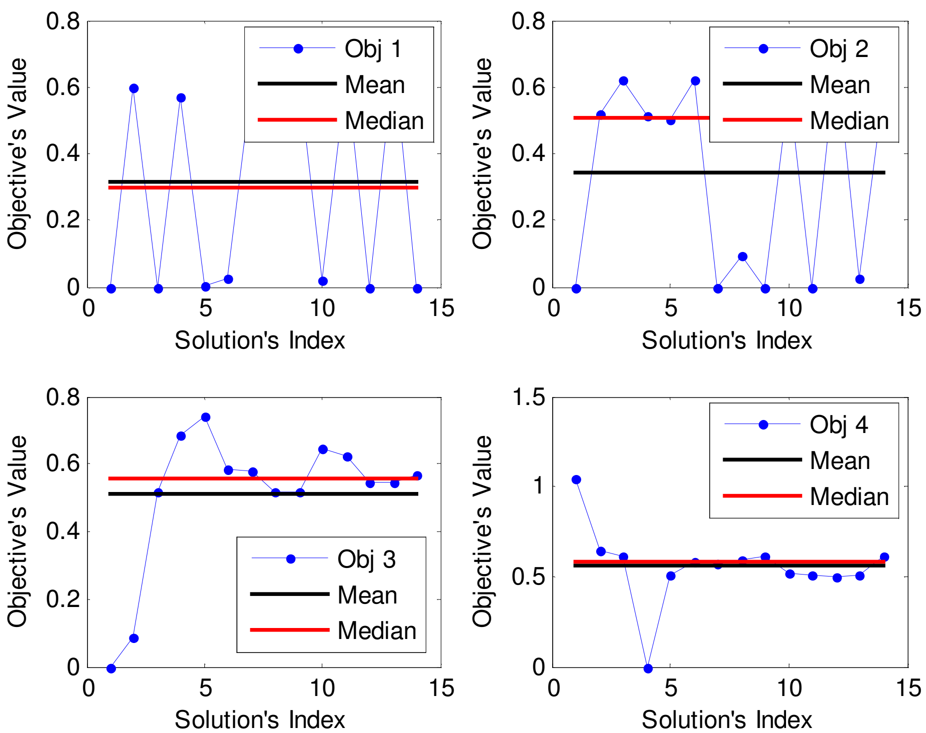


Figure 5. 31 DTLZ2 (4) Results achieved at the 200th generation

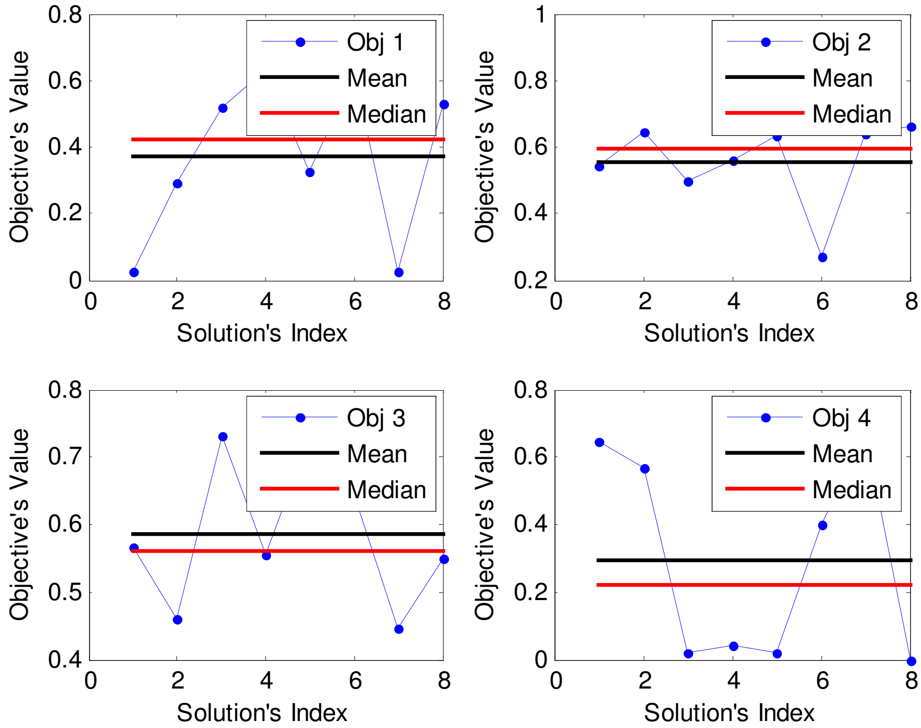


Figure 5. 32 DTLZ2 (4) Results achieved at the 300th generation

In order to assess the utility of the suggested ε-dominance strategy as a PPA technique in optimisation frameworks with more than two objectives, an optimisation scenario consisting of four objectives (DTLZ2 (4)) was deployed. In Figures 5.31 and 5.32 a statistical presentation of the results achieved by ε-MOEA is illustrated in terms of each objective. The first set of preferences was articulated at the first generation and allocated the same priority level to the 4 objectives. The aim was to produce a set of Pareto optimal solutions with a tolerance value $\varepsilon = 0.1$ in terms of the four objectives. In Figure 5.31 the corresponding results achieved at the 200th generation are illustrated. The mean and the median values in terms of the 4 objectives for the set of solutions achieved were closely located in the Pareto optimal range [0.3, 0.5]. The results presented in Figure 5.32 were achieved at the 300th generation of the optimisation process and were affected by the preferences articulated at the 200th generation and illustrated in Figure 5.30.

The new preferences decreased the priority level for objectives 2 and 3 by articulating an exact preference for solutions whose values in terms of these two objectives are bigger than 0.6. The results achieved at the 300th generation (Figure 5.32) conformed to the progressively articulated preferences by illustrating a reduced set of solutions which manifested the desired preferences. The results presented in Figures 5.31 and 5.32 are also illustrated using parallel coordinates graphs in Appendix D.

5.3.4. Demonstration of FF-PPA technique

Figures 5.33-5.36 illustrate an interactive optimisation scenario solving ZDT1. The PPA of the DM was expressed using Fonseca and Fleming's preferability operator. The underlying search algorithm consisted of NSGA-II with the same configuration used in sections 5.3.1 and 5.3.2. In the following scenario, the two objectives had the same priority level but different desired goals in order to emphasize the facility of reducing the search space and focusing on regions of interest. A certain ROI on the convex Pareto front was articulated in terms of the goal values for each objective. The goal value for a certain objective 'i' denoted the maximum acceptable value in terms of that objective. Consequently, any candidate solution whose value in terms of each objective was at least achieving the goal value for that objective would have therefore been considered as a preferred solution.

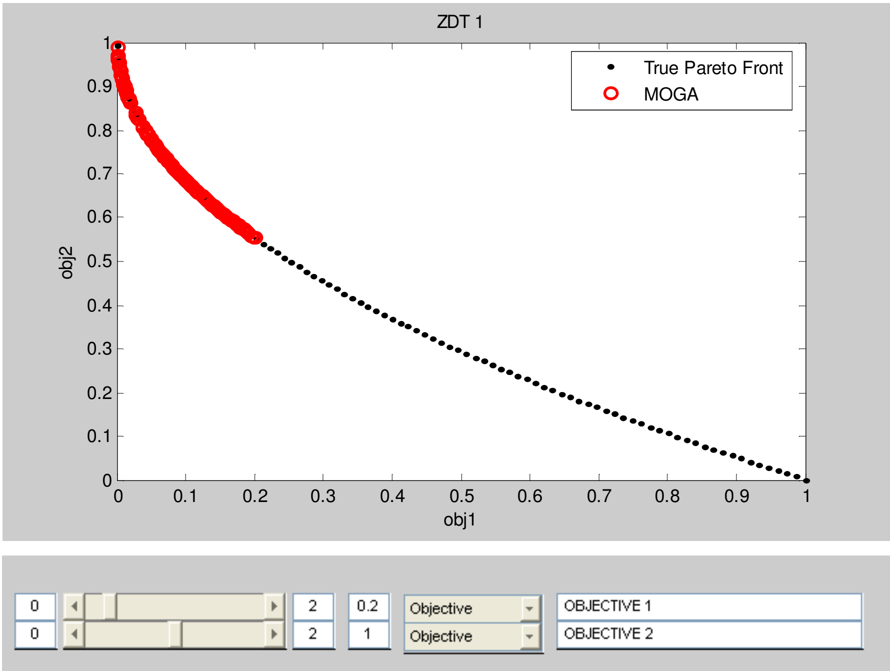


Figure 5. 33 FF-PPA technique running on ZDT1 (190th generation)

In Figures 5.33-5.36 the desired goal values for objective 1 and objective 2 were as follows respectively: Figure 5.33 →(0.2, 1), Figure 5.34 →(1, 0.2), Figure 5.35 →(0.5, 0.5), and Figure 5.36 →(0.3, 0.5). From the scenarios presented in Figures 5.33-5.36, it was very obvious that FF-PPA technique was a precise and DM-oriented facility. The numerical goal value in each dimension was the only information required from a DM to reduce the search space and focus on certain parts of a Pareto front. The goal values (or worst case scenario for each objective) progressively articulated by the DM are input to the optimiser to modify the concept of dominance and steer the search and selection process in the desired search region.

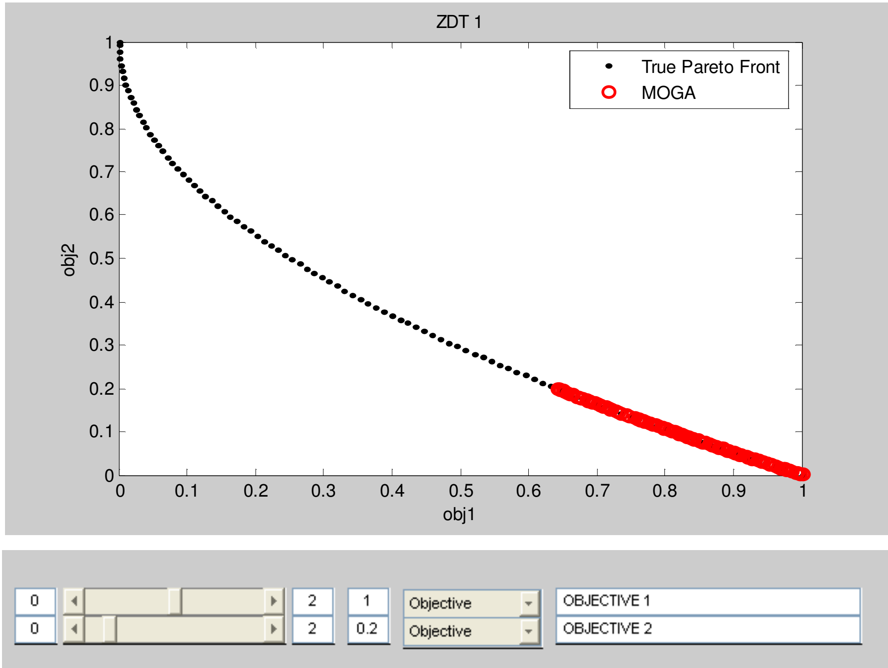


Figure 5. 34 FF-PPA technique running on ZDT1 (210th generation)

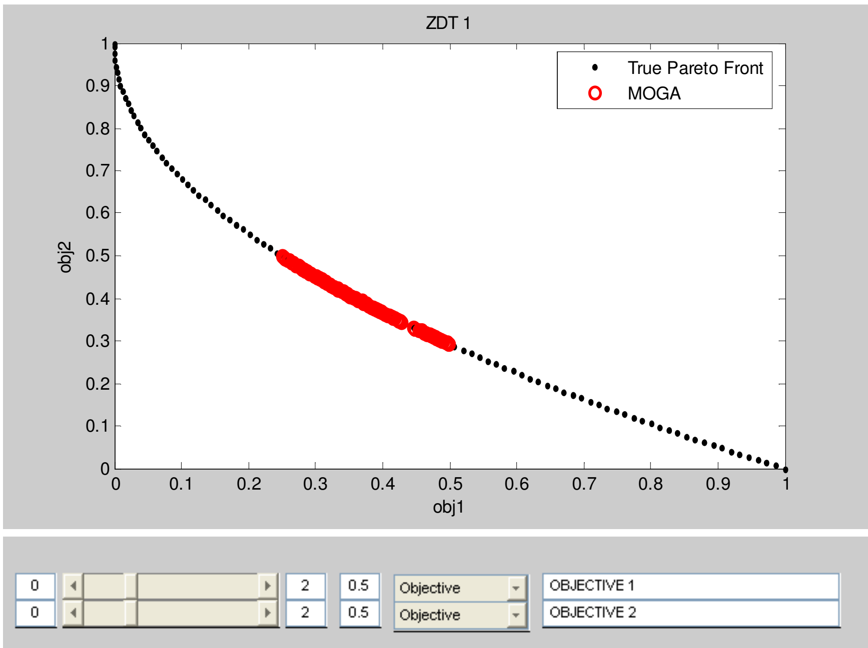


Figure 5. 35 FF-PPA technique running on ZDT1 (230th generation)

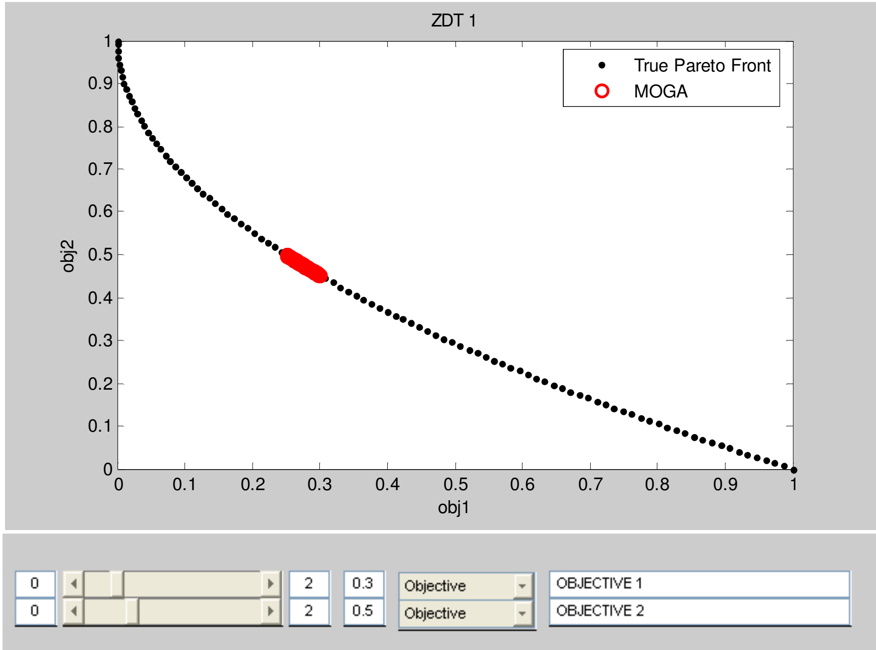


Figure 5. 36 FF-PPA technique running on ZDT1 (240th generation)

FF-PPA is a straightforward technique which allows the articulation of exact preferences, as well as the articulation of vague preferences by interactively relaxing the goal values for the objectives, and therefore enlarging the boundaries of the requested ROI. In Figure 5.37, the articulated goal value for the two objectives was 'one' (Figure 5.37 →(1, 1)).The aim was to produce results all along the Pareto optimal front. Figure 5.37 (a) illustrate the results achieved for the articulated preference (Figure 5.37 (c)) at the 300th generation. In Figure 5.37 (b) the parallel coordinate graph for the achieved solutions at the 300th generation is illustrated. All the solutions were achieving the goal values set for the two objectives.

Finally, in Figure 5.38, two snapshots of the optimisation progress are visualised at the 100th (a) and the 190th (b) generation of the optimisation process. The aim was to highlight the efficiency of Fonseca and Fleming's PPA technique in terms of reducing the search space. From Figure 5.38 (a), it was obvious that FF-PPA technique was restricting the search process of NSGA-II to a reduced region of the objective space. Space reduction is a desirable feature especially valued in high dimensional objective spaces. The FF-PPA technique was then assessed on an optimisation problem presenting an increased number of objectives. DTLZ2 (4) was again used to achieve the desired task. The results achieved at the 200th generation of the optimisation process are illustrated in Figure 5.39. These results were affected by the preferences showed in Figure 5.39c and articulated at the 1st generation. The parallel coordinate graph presented in Figure 5.39b included all the solutions achieved at the 200th generation and which presented the desired criteria articulated by the DM. At the 200th generation, the desired goal value for objective 2 was reduced to the value 0.2 (similar to the goal values of objectives 1 and 4) (Figure 5.40c). The new articulated preferences were efficiently fed to the underlying optimisation process which steered the search in the direction of the new ROI. The results achieved at the 250th generation (Figure 5.40a and 5.40b) illustrated the satisfaction of the new preferences by reducing the mean and the median values achieved for the 2nd objective.

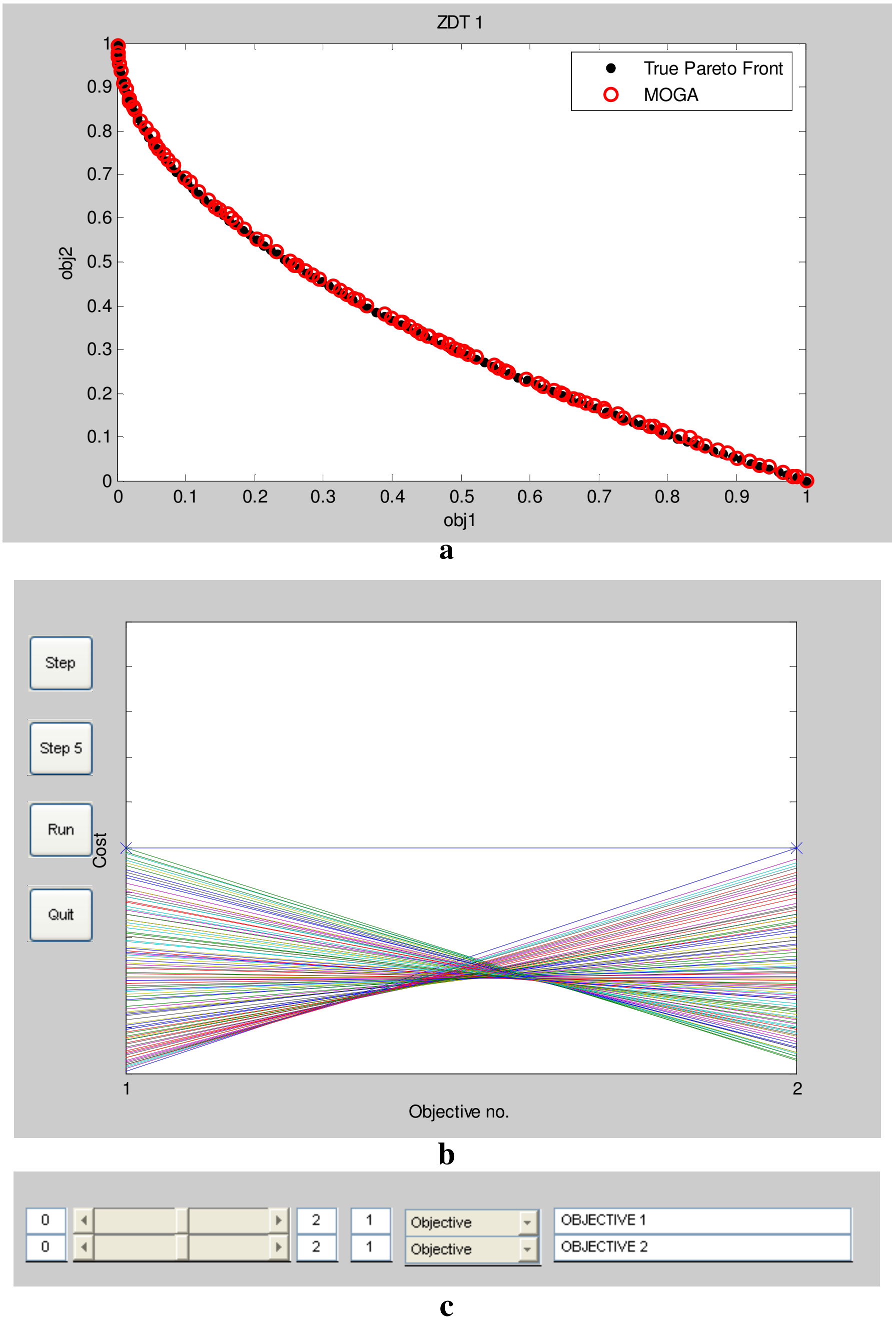
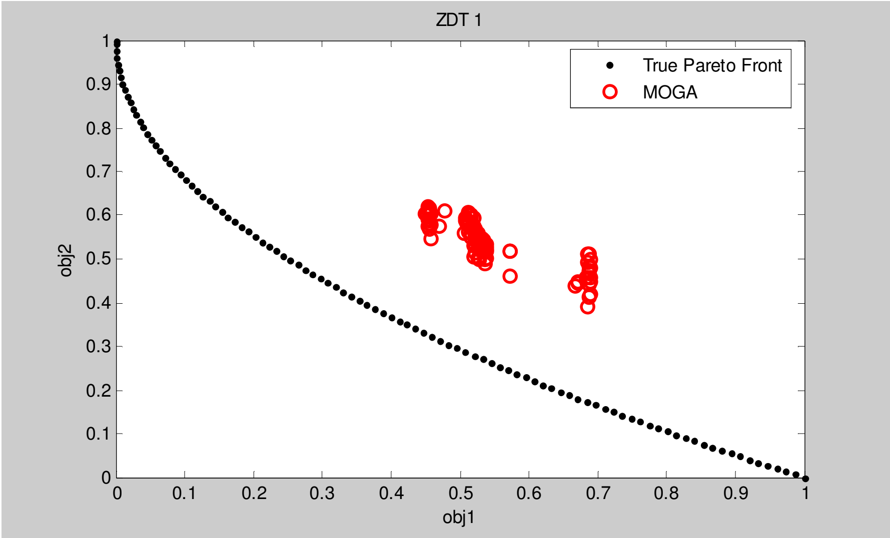
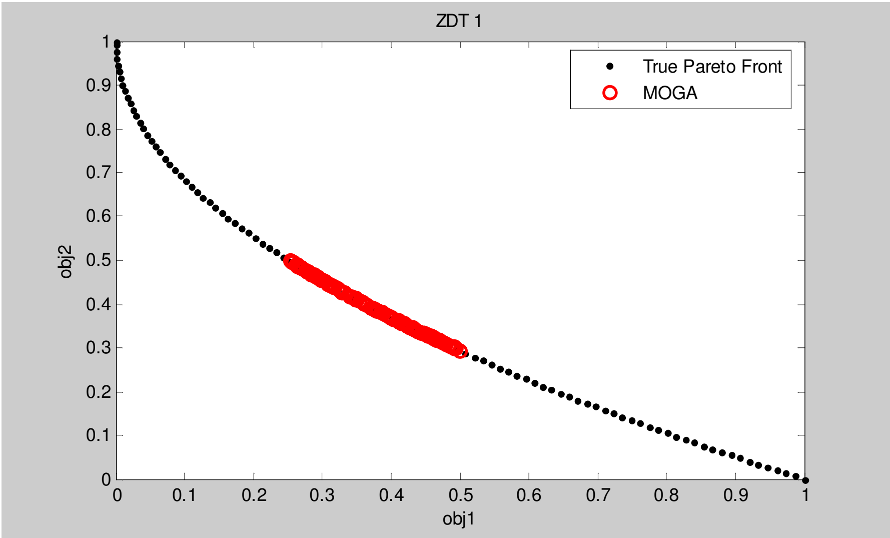


Figure 5. 37 FF-PPA technique running on ZDT1 (300th generation)



a



b

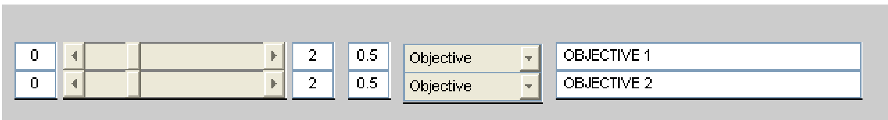
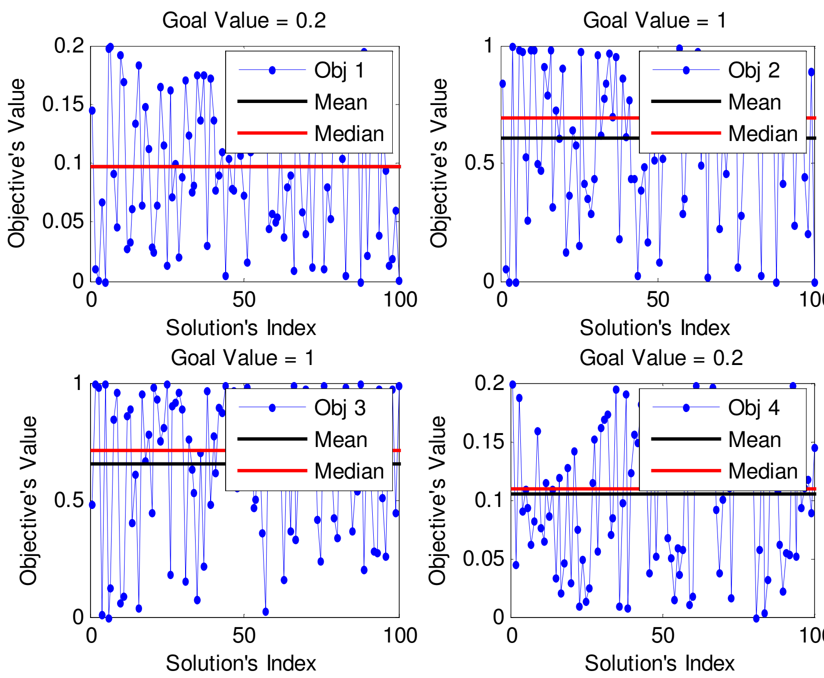
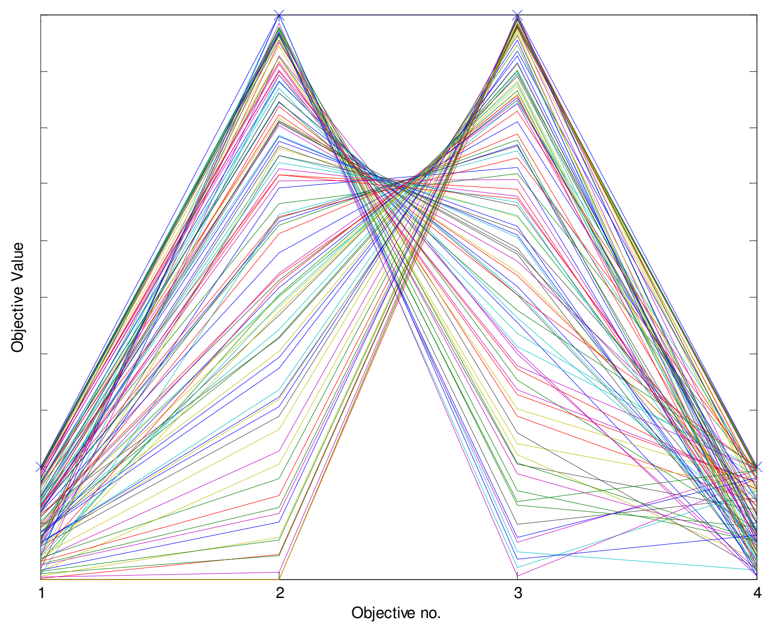


Figure 5. 38 FF-PPA technique reducing the search space (ZDT1)

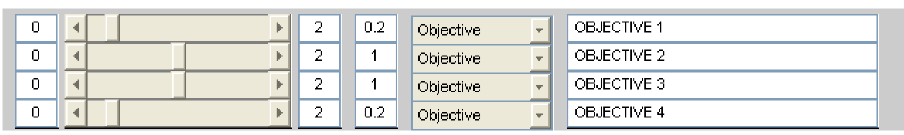
The last preference articulation was then articulated at the 250th generation. The goal values for all the objectives were set to the value 0.5 in an attempt to produce optimal solutions in terms of the four objectives and which reside on the intermediate region of the well-defined Pareto optimal front of the DTLZ2 test function. At the 300th generation (Figure 5.41), the median and the mean values for all the four objectives presented values in the range [0.4, 0.5] which conformed with the required goal values



a

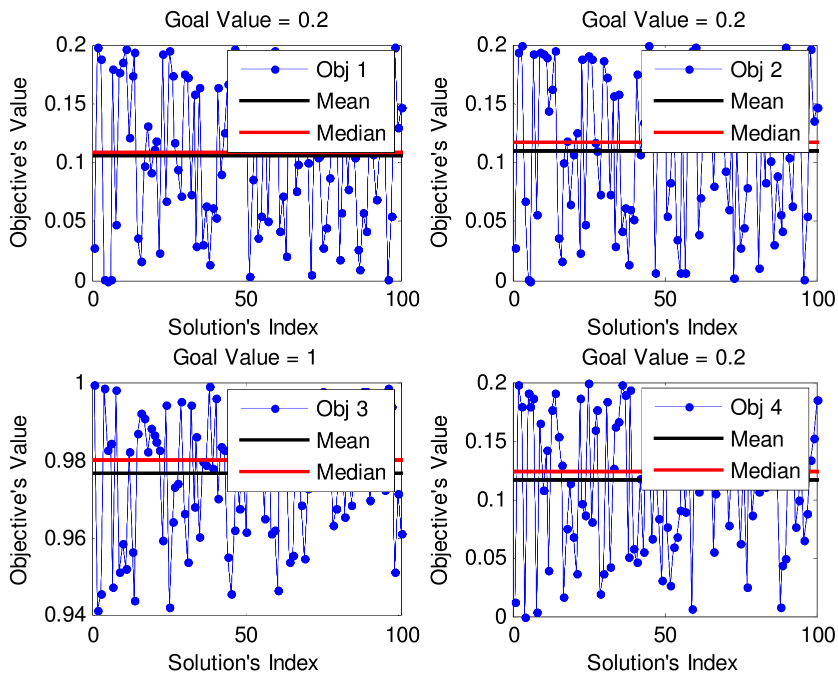


b

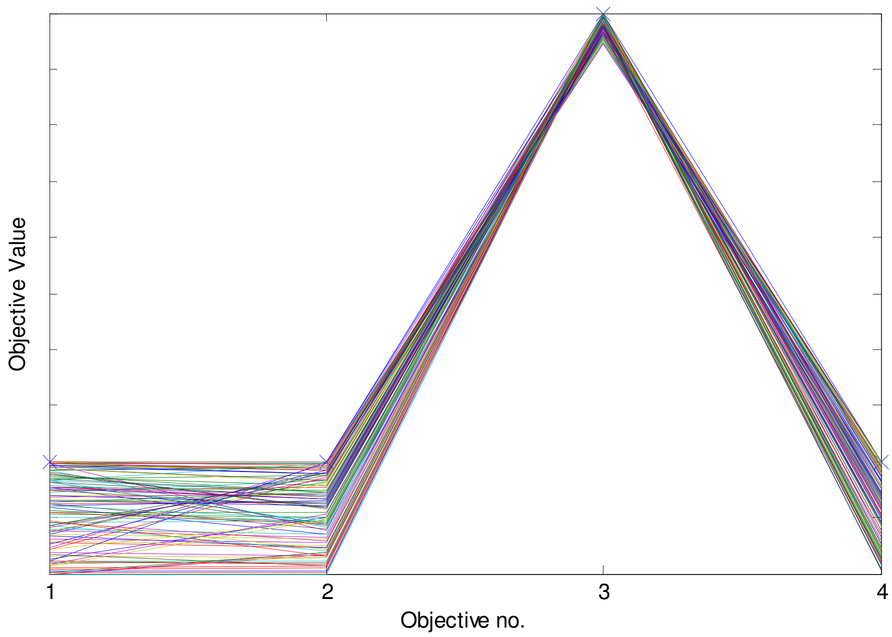


c

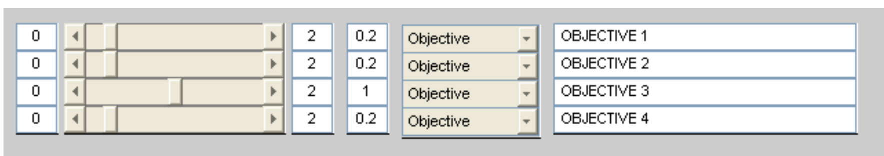
Figure 5. 39: (a) DTLZ2 (4) Results achieved at the 200th generation. (b) Parallel Coordinates of the results achieved affected by the preferences illustrated in (c) and articulated at the 1st generation of the optimisation process



a

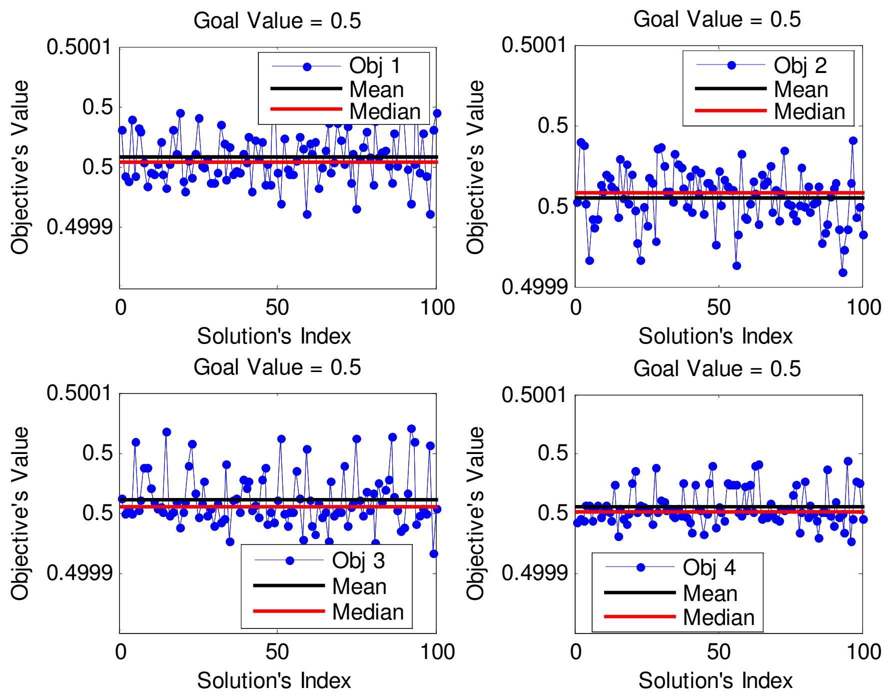


b

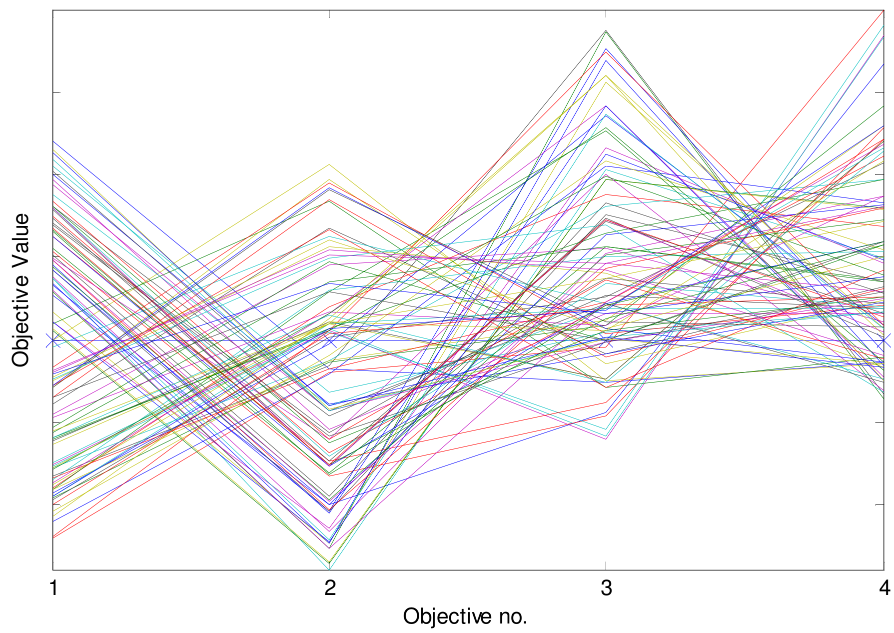


c

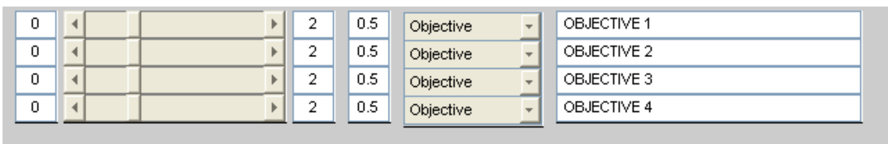
Figure 5. 40 (a) DTLZ2 (4) Results achieved at the 250th generation. (b) Parallel Coordinates of the results achieved affected by the preferences illustrated in (c) and articulated at the 200th generation of the optimisation process



a



b



c

Figure 5. 41 (a) DTLZ2 (4) Results achieved at the 300th generation. (b) Parallel Coordinates of the results achieved affected by the preferences illustrated in (c) and articulated at the 250th generation of the optimisation process

.

5.4. Discussion and Summary

Progressive preference articulation is a useful approach for reducing high-dimensional spaces and tackling evolutionary multiobjective optimisation problems. It has benefits when compared with its *a priori* preference articulation technique counterpart, which requires the DM to know his/her preferences in advance, and which makes no use of the information that becomes available during the search process.

In the previous sections, experiments were carried out using some of the most recent and most established PPA techniques. Although the deployed scenarios consisted of 2 and 4 dimensional scenarios only, the strengths, weaknesses, and therefore the efficiency and suitability of these PPA techniques for the many-objective optimisation were apparent.

The FF-PPA technique clearly still stands as an efficient, DM focused and truly "progressive" articulation technique. It is a user-friendly and direct technique which allows the articulation of exact as well as vague preferences. The accuracy and pertinence of the results achieved by the FF-PPA technique are realised with modest computational effort, and easily scales to any number of objectives. The utility of FF-PPA technique in reducing the search space and focusing on a certain ROI was also demonstrated in Fleming, Purshouse and Lygoe (2005) on an optimisation scenario consisting of 8 objectives. In evolutionary many-objective optimisation scenarios, supervised by an application-expert DM, the FF-PPA technique is a suitable optimisation technique. The technique's traditionally criticised weakness[24] of setting easily achievable, or contrarily very optimistic goal values that can hinder the search (e.g. Branke and Deb 2004), can be overcome by deploying an automated DM such as expert systems (Todd and Sen 1999) which can play the role of a progress sensor detecting such optimisation anomalies and modifying the goal values as appropriate with or without DM intervention.

The biased crowding concept is a well-established preference articulation technique that can be used in a progressive manner to focus on a certain ROI. It is mostly useful and practical from the DM's point of view when used with convex or concave optimisation problems with no more than 3 objectives. When dealing with multimodal, ill-behaved or high-dimensional problems, using the biased crowding is not efficient, especially in high-dimensional problems, because it can be very confusing for a DM to devise a plane or hyperplane of interest for solutions projection. This difficulty can be broadly compared to the difficulty of devising weight values for the objectives in the weighted sum approach.

[24] A weakness which is common to goal programming approaches

Even in the best scenarios, it was actually noted that the resolution of the ROI achieved by the biased crowding based PPA technique is not as precisely aligned with the DM preferences when compared with the FF-PPA technique, but which can be suitable for addressing vague user preferences.

On the other hand, setting epsilon values for each objective and ensuring the facility of progressively modifying the epsilon values can establish the ε –dominance concept within the ε-MOEA as another PPA technique. From the DM's point of view, it will remain, however, a demanding approach, involving the manipulation of ε-values rather than numerical goal values for the objectives. Nonetheless, the diversity promotion mechanism employed in ε-MOEA is a state-of-the-art multiobjective optimisation technique for limiting archive size and promoting diversity, and is highly commendable. Although it seems so far that the use of ε –dominance is better reserved for defining results precision and the magnitude of computational requirements - which can be used to reduce the effect of dominance resistance in high dimensional problems- future research into using ε –dominance as a method to articulate preferences is required.

Lastly, despite its simplicity and practicality for certain optimisation problems, the use of the guided dominance scheme as a PPA technique suffers from several weaknesses. Because the modification of the dominance scheme implicitly assumes linear utility functions, it can be quite complicated to handle multimodal and non-convex optimisation problems. In addition, when tackling high-dimensional problems this technique can be computationally expensive and demanding (Branke and Deb 2004), especially from a DM point of view, as the number of required pair-wise tradeoff values for this technique becomes very high ($n!/(n-2)!$ tradeoff value required, where n is the number of objectives). The guided dominance principle is best reserved for optimisation scenarios where specific tradeoffs among pairs of objectives are envisioned and devisable.

Chapter 6

Conclusions

6.1. Evolutionary Multiobjective Optimisation

6.1.1. Requirements of Multiobjective Optimisation

It is not possible to find a single optimal solution across all the objectives when tackling a multiobjective optimisation problem with conflicting objectives. Instead a set of optimal solutions, known as the Pareto front, is anticipated. This set of optimal solutions should ideally consist of a set of tradeoff solutions, each of which cannot be improved further in terms of a certain objective without introducing some deterioration in terms of one or more of the other competing objectives. The set of solutions achieved by a multiobjective optimiser is therefore required to be close to the true Pareto front. Because of the non-existence of an ideal single solution, the set of optimised solutions is also required to be well spread and covering wide areas of the Pareto front, presenting the decision maker with a well distributed set of solutions to choose from. Deciding on a certain solution is a process usually based on subjective preferences such as objectives' priorities or regions of interest. Moreover, and especially when dealing with real world applications, it is in the designer's best interest that the approximation set is achieved fast and within an acceptable amount of time and a limited budget of objective function evaluations. Therefore, convergence, diversity, pertinence to the DM (ROI) efficiency and speed of convergence are all desired and essential requirements of multiobjective optimisers and constitute their assessment basis.

6.1.2. Research Motivations

Over the last two decades, the field of evolutionary multiobjective optimisation has considerably evolved. Many major milestones and multiobjective evolutionary optimisers were devised over the years with sophisticated features addressing the requirements sought when solving a multiobjective problem. The requirement for solutions' *convergence* towards the Pareto front, the *diversity* of an approximation set and its *pertinence* to the user, and the efficiency, speed and practicality of MOEAs are all matters that have been investigated and researched over the years. Nevertheless, multiobjective optimisation has traditionally focused on problems consisting of two or three objectives. Real-world problems often require the optimisation of a considerably larger number of objectives. Research has shown that conclusions drawn from experimentations carried on two or three objectives cannot be generalized for a higher number of objectives. The curse of dimensionality is a problem that faces decision makers when confronted with many objectives. As a result in scenarios involving large numbers of competing objectives, many aspects of EMO remain under- exploited and -explored. In this thesis, research was undertaken and new ideas were proposed with the aim of exploring such critical areas of EMO. Moreover, the goal was to promote and preserve EAs' popularity and increase their suitability for solving multiobjective optimisation. This was achieved by introducing innovative remedial measures for some of their widely criticised drawbacks such as their heavy computational load and number of objective function evaluations for convergence towards Pareto fronts and the conflict between convergence and diversity in problems with many competing objectives. This last drawback is widely linked to dominance resistance and the increasing probability of producing, in a high dimensional objective space, non-dominated solutions which are redundant, most often excelling in a single objective and remote from the Pareto front. The previous problems are known to be caused by the impact of the Pareto-based selection mechanisms, generally used in EAs, when deployed in conjunction with active diversity promotion strategies.

6.1.3. Faster and Better Convergence

In order to address the requirement for an enhanced proximity and a fast convergence towards Pareto fronts, the latter feature being especially valuable for real world applications, a portable Convergence Accelerator Operator (CAO) was proposed for incorporation in MOEAs. Traditionally, EAs operate in decision space and perform decision space to objective space mapping but fail to exploit direct use of the objective space. The novel multiobjective optimisation accelerator (CAO) uses direct manipulation in objective space together with neural network mappings from objective space to decision space. More precisely, the CAO uses a specific type of neural networks, the radial basis function as opposed to the multilayer perceptron (MLP) used in a series of papers by Gaspar-Cunha *et al*. The main reason behind using an RBF neural network within the CAO is its much faster training process compared to MLPs [25]. Using an MLP neural network within a convergence accelerator strategy, such as the CAO, works against the purpose of the operator and imposes a considerable computational effort on the optimisation process. The use of RBF neural networks generalises the use of the CAO by making it practical to deploy the CAO on a wide variety of problems and not restricted to computationally

[25] Trained with the gradient descent back-propagation algorithm

expensive objective functions. Moreover, a correction step integrated within the CAO is applied as an attempt to rectify any inaccuracies that might be introduced by the neural network predictions. This is performed in order to maintain the exact fidelity of the solutions and the optimisation problem being solved.

The CAO is thus a transferable component that can be hybridized with any multiobjective optimisation algorithm. The purpose of this convergence acceleration operator is to enhance the search capability and the speed of convergence of the host algorithm. The operator acts directly in objective space to suggest improvements to solutions obtained by a multiobjective evolutionary algorithm (MOEA). These suggested improved objective vectors are then mapped into decision variable space and tested. The CAO was incorporated with two leading MOEAs, the Non-Dominated Sorting Genetic Algorithm (NSGA-II) and the Strength Pareto Evolutionary Algorithm (SPEA2) and tested on a variety of recognised test problems with increasing number of competing objectives. In all cases, the introduction of the CAO led to improved convergence and solution's diversity for comparable numbers of function evaluations.

6.1.4. The Diversity Requirement

The conflict of requiring solutions' convergence towards the Pareto front while promoting diversity in the approximation set of an optimisation problem with *many* conflicting objectives was approached from a new perspective. A strategy for monitoring, promoting and controlling diversity in high-dimensional objective spaces was introduced. The diversity requirement widely regarded as a secondary requirement had to be redefined as a local and adaptive requirement. As a result the approach consisted of setting an approximation for the notion of a good and targeted diversity requirement, so that the latter requirement can be monitored and maintained simultaneously. In other words, when a wide dispersal of solutions is detected, the suggested strategy freezes the diversity requirement as a discriminator when selecting solutions for variation and survivals. Such wide dispersal is easily attainable in high-dimensional objective spaces and generally comes on the expense of a local divergence from the Pareto front. As a result, the diversity requirement is redefined to be perceived as a requirement that needs be controlled in a way that does not deteriorate the primary requirement for convergence towards the tradeoff surface. The suggested strategy for controlling and promoting diversity, termed the diversity management operator (DMO), was used in the context of NSGA-II and deployed to solve a set of test functions with increasing number of competing objectives (6→20 objectives). For all the problem dimensions used, the utility and the beneficial contribution of the suggested technique was demonstrated. Compared to the results achieved for each of the test functions by the standalone NSGA-II, significantly enhanced results were achieved when the DMO was operating and governing the selection processes of NSGA-II.

Moreover, in scenarios with many competing objectives, it was shown that widely used metrics, such as the cover set metric or the dominated distance metric, which are usually considered suitable for comparing the performance of two MOEAs, cease to be informative. This is due to the fact that the majority of such metrics are built upon the Pareto dominance concept[26] which intuitively adopts an equal importance for the convergence and the diversity requirements. While this intuition might be beneficial in scenarios with two or three competing objectives, equally weighting the two requirements can ultimately be seen as reducing an optimisation problem with *many* competing objectives to multiple single objective problems. This analogy is due to the increasing probability of producing solutions which are particularly optimal in terms of a single objective, especially as the number of competing objectives increase. This problem was originally detected in Schaffer's (1985) VEGA, which is considered one of the first attempts for using EAs to solve multiobjective problems. In addition, it was shown that the non-dominated evaluation metric (Deb 2001) was more suitable and a more informative 'binary' metric. The latter metric can be manipulated, by means such as the guided dominance or the ε-dominance principle to set the precision in terms of each of the two 'objectives' (i.e. convergence and diversity), in order to assign the desired weighting value to the two requirements.

6.1.5. The Pertinence Requirement

Finally, the study presented in Chapter 5 was aimed at promoting the research and the use of progressive preference articulation techniques in the EMO community. Preference articulation techniques, and especially progressive preference articulation (PPA) techniques are effective methods for supporting the decision maker. Progressive preference articulation is a useful approach for reducing high-dimensional spaces and tackling evolutionary many-objective optimisation problems. It has benefits when compared with its a priori preference articulation technique counterpart, which requires the DM to know his/her preferences in advance, and which makes no use of the information that becomes available during the search process. PPA techniques also have benefits when compared with a posteriori approaches. The latter do not impose reductions on the search space and thus can be susceptible to problems such as speciation, dominance resistance and the conflict between convergence and diversity, especially as the number of competing

[26] They assign performance scores by finding pairs of non-dominated vectors, which becomes less likely when the dimensionality of the objective space increases

objectives increase. Moreover, optimisation approaches deploying a posteriori preference articulation are more likely to produce approximation sets with overwhelming sizes which complicates the decision making process.

Due to the fact that until recently most EMO research focused on bi-objective optimisation problems where the need for preference articulation is less apparent, preference articulation in EMO is somehow overlooked. In this study, some of the most recent, potential and most established PPA techniques are examined, contrasted and their utility for tackling multiobjective optimisation problems is discussed and compared from the viewpoint of the decision maker. The strengths and limitations of four well established and novel PPA techniques were illustrated and demonstrated on a set of multiobjective optimisation problems with varying scenarios of preferences.

6.2. Future Perspectives

The addition of deterministic improvement steps to stochastic MOEAs by exploiting the strategy of manipulating and performing direct search in the objective space to assist the evolutionary optimisation process is one of the major contributions of this thesis. Such deterministic processes have been shown to enhance the suitability of evolutionary algorithms for solving multiobjective optimisation problems by enhancing their practicality and their search convergence towards Pareto optimal regions of the search space. Further research concerning the improvement of the performance of neural networks investigating sophisticated learning algorithms and architectures is certainly profitable. More specifically, devising optimised strategies for training neural networks, especially for dealing with multimodal objective functions, is very beneficial. Increasing the confidence in the prediction quality of neural networks from the objective space to the decision variable space is an area of research that should be seriously considered, as this should confer a lot of potential improvements to multiobjective optimisation. Neural networks constitute a well-established branch of computational intelligence and are highly beneficial for multi- (as well as single) objective optimisation when merged with EAs. Investing data mining techniques from machine learning and artificial intelligence disciplines, as well as the investigation of alternatives to neural networks, such as Kriging models, self organizing maps and response surface models for capturing the mapping relationship from the objective space to the decision variable space, and vice versa, is another very promising and beneficial area of research.

Future work should be invested in interactively executing the convergence acceleration operator (CAO) on request by the DM and in deploying the operator in a progressive preference articulation technique to assist in guiding the search towards specific regions of interest (ROI). Being able to integrate progressive preference articulation techniques within the CAO is another major benefit; as an example, the objective space local improvement process might then be permitted to improve one of two competing objectives while ignoring or deteriorating the performance of the competing one by an acceptable amount reflecting the decision maker's preferences. Furthermore, the interpolation step factor used for objective space improvement is an application-dependent parameter and will be influenced by the landscape of the objective space. In the experiments undertaken in this study, step factors ranging from 0.01 up to 0.2 were tried before settling for h=0.1 as the step factor to be used for the tests. There is scope to explore the use of adaptive step factors as MOEAs explore the objective space. Future work will also include optimising test functions and real world problems with increasing number of competing objectives using MOEAs which simultaneously deploy the CAO and the diversity management operator (DMO).

In addition, the identification of pair-wise objective relationships can also be learned and detected during the NN training component of the CAO. This is can be endorsed by deploying visual techniques such as Parallel Coordinates (Fonseca and Fleming 1998 and Inselberg 1985) or Scatterplot matrices or quantitative approaches such as Kendall sample correlation statistic (Kendal 1938). Building up knowledge about objective-relationships as the NN is trained can be used to identify objective redundancies and suggest dimensionality reductions that can accelerate the optimisation process. It can also be used in the objective space improvement component of the CAO by improving the objectives in a certain order reflecting their relationships (e.g. (1) harmonious objectives→(2) independent objectives →(3) competing objectives).

Finally, a suggested list of various future work directions is presented below:

- Recent research and evidence (e.g. Knowles and Corne, 2007) has shown that as the number of competing objectives increase, random search algorithms become competitive with MOEAs. An interesting future investigation would therefore consist of assessing the performance of the CAO and the DMO (independently and combined) against the performance of random search algorithms on multiobjective problems with increasing number of objectives.
- Future research investigating the use of heat maps (Pryke, Mostaghim and Nazemi 2007), probabilistic models and novel approaches for determining and exploiting relationships between the decision variables and the different objectives constituting a multiobjective optimisation problem is highly beneficial and required.

- Thorough experimentation with other evolutionary computation techniques such as particle swarm or ant colony for solving multiobjective optimisation with many competing objectives is an interesting future work direction. A comparative study analysing and contrasting the performance of different evolutionary computation techniques including genetic algorithms on many-objective optimisation problems is essential to establish a detailed understanding of their strength and limitations. Important conclusions can be drawn from such comparative studies and might benefit the field of multiobjective optimisation.

- Checking the effect of decision space dimensionality on the optimisation process of many competing objectives. This can be achieved by using scalable objective functions where the number of objectives large is fixed and the number of decision variables is varied. The objective is to quantify and understand the effect of the objective space and the decision space dimensionalities, separately, on the optimisation process.

- Research has shown that despite being suitable for certain objective space dimensions the parameters for the widespread recombination and mutation operators are not suitable for higher dimensionalities (Purshouse and Fleming 2003b). Some research was dedicated for designing methodical approaches (Minsker and Goldberg (2000) and Lobo and Goldberg (2001)), known as *competent EAs*, for supporting the user in choosing the right parameters and designing an appropriate EA that accommodate the application at hand. Further research investigating the implementation of totally self-configurable and adaptive MOEA that adapts to the dimension of the optimisation problem and the nature of its search spaces is another interesting research direction.

- Examining and experimenting with the hypervolume metric or other performance metrics as an alternative to the usual ‘Pareto Dominance’ concept which widely governs the selection processes of MOEAs is very beneficial. Some research has already been produced in this area, however this interesting research topic is still far from being fully explored.

Appendix A

Simplified Dynamical Model of Aircraft Motion

A simplified dynamical model of an aircraft motion can be represented by a fourth order linear equation (Tabak, Schy, Giesy and Johnson 1979). The corresponding state equation is:

$$\overset{0}{x} = Ax + Bu \qquad \text{(A.1)}$$

$$\text{Where } x \text{ is the state vector: } x = \begin{bmatrix} \beta \\ r \\ p \\ \phi \end{bmatrix} \quad \begin{matrix} \textit{Sideslip} \\ \textit{Yaw rate} \\ \textit{Roll rate} \\ \textit{Bank angle} \end{matrix}$$

$$u \text{ is the control vector}: u = \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad \begin{matrix} \textit{Aileron control motions} \\ \textit{Rudder control motions} \end{matrix}$$

The control vector, u, is represented by Equation A.2 where u_p is the pilot's control input vector:

$$u = Cu_p + Kx \qquad (A.2)$$

C and K are the gain matrices of the form:

$$C = \begin{bmatrix} 1 & 0 \\ K_5 & 1 \end{bmatrix} \qquad K = \begin{bmatrix} k_6 & k_1 & k_2 & 0 \\ k_7 & k_3 & k_4 & 0 \end{bmatrix}$$

By substituting equation (A.2) into (A.1) we get:

$$\overset{0}{x} = (A + BK)x + BCu_p \qquad \text{(A.3)}$$

The Eigen values of the matrix *(A+BK)* define the stability properties of the system modelled in Equation A.3.

Appendix B

Below are presented the C-metric results achieved when comparing NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT1, ZDT3 and ZDT6 at each of the 10 executions:

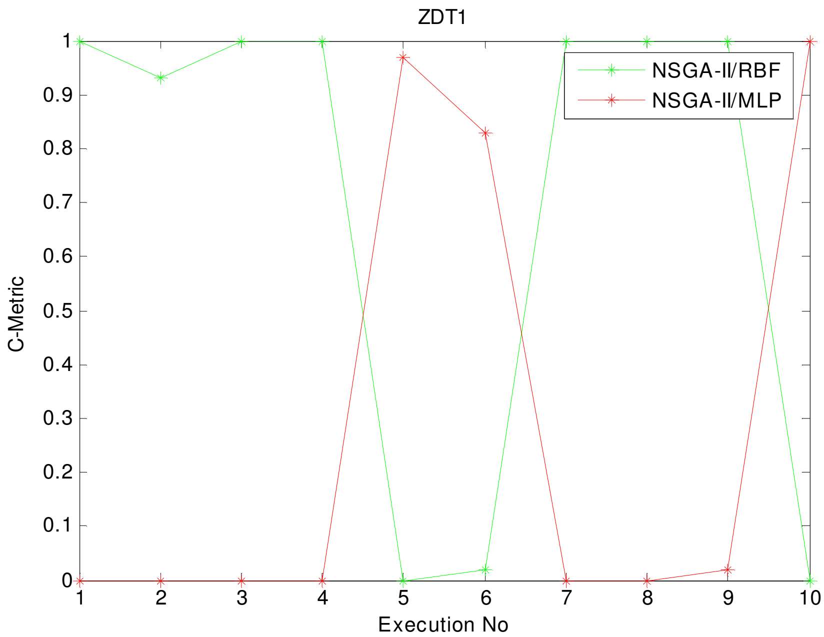


Figure B.1 C-Metric Values achieved by NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT1 at each of the 10 executions

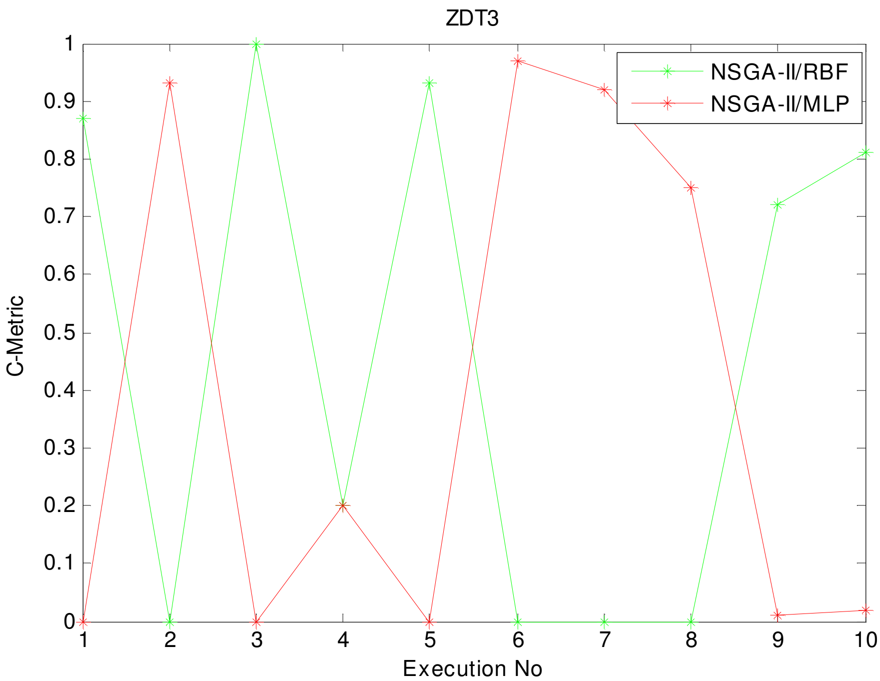


Figure B.2 C-Metric Values achieved by NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT3 at each of the 10 executions

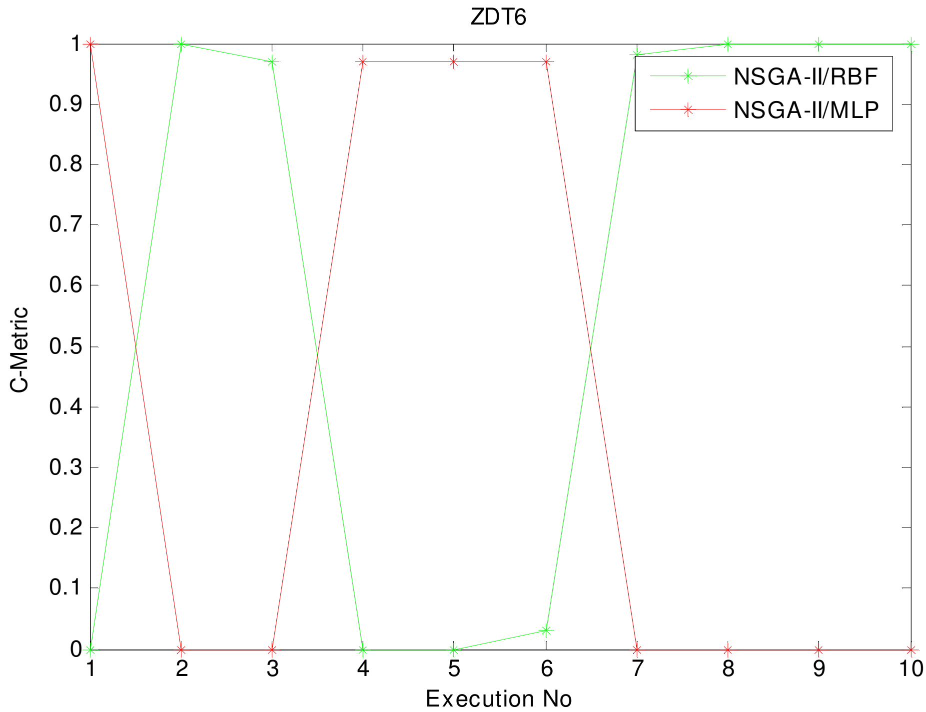
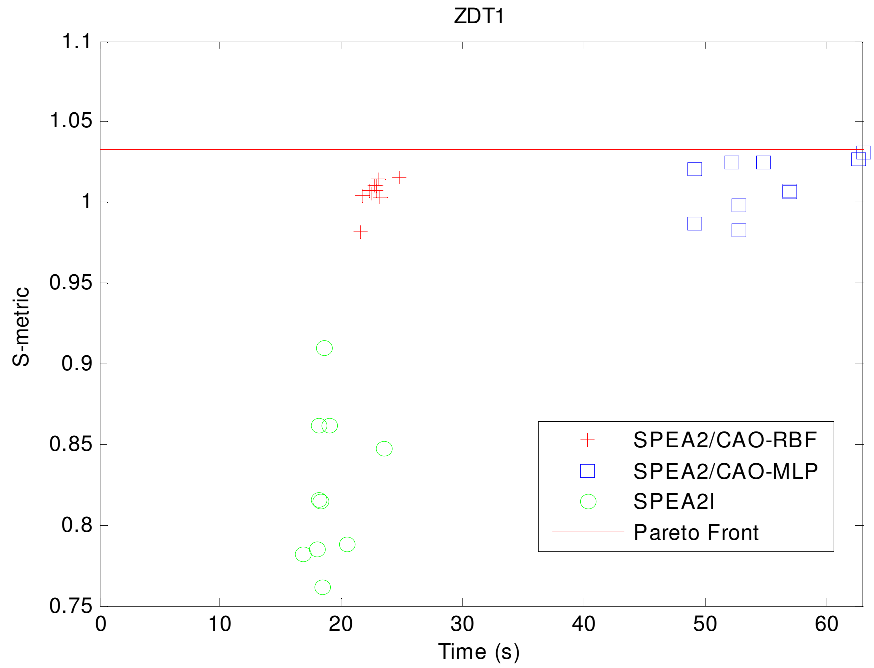


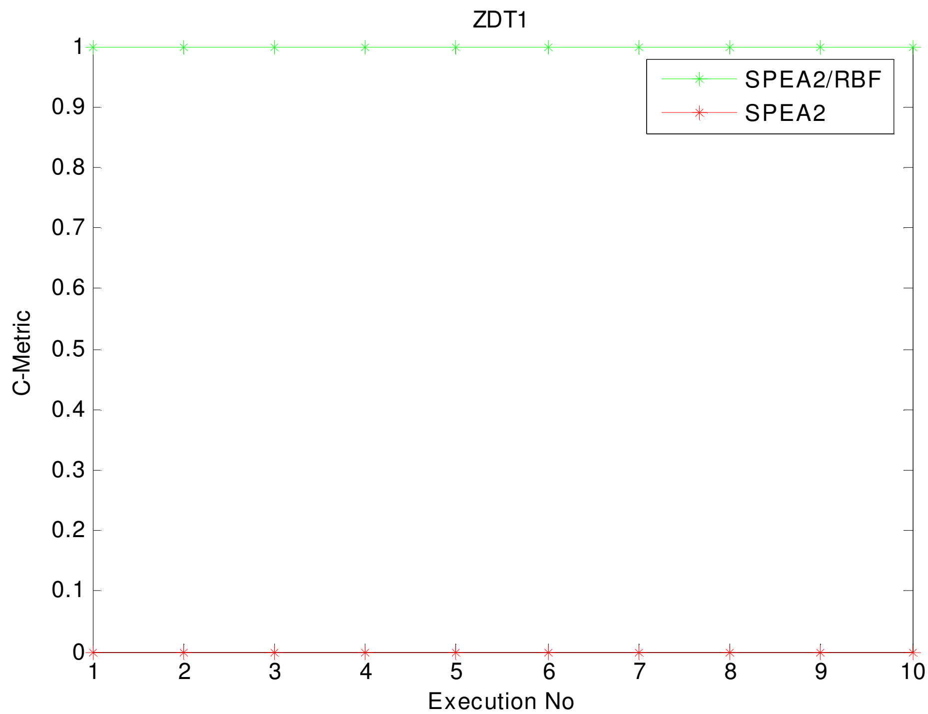
Figure B.3 C-Metric Values achieved by NSGA-II/CAO-RBF and NSGA-II/CAO-MLP on ZDT6 at each of the 10 executions

Appendix C

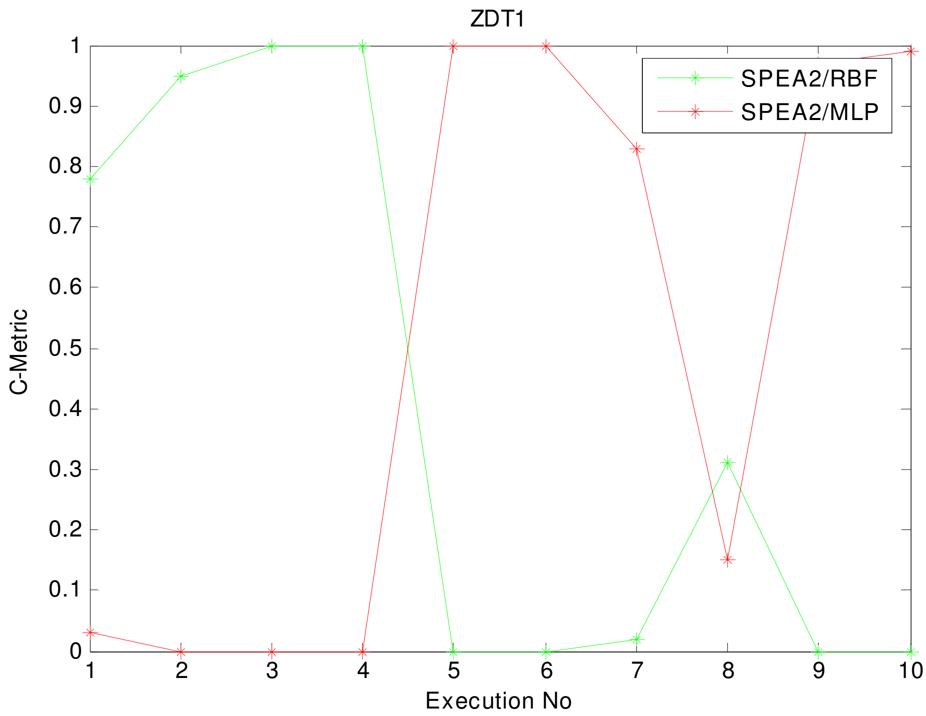
Below are presented the results achieved when the CAO is hybridized with SPEA2:



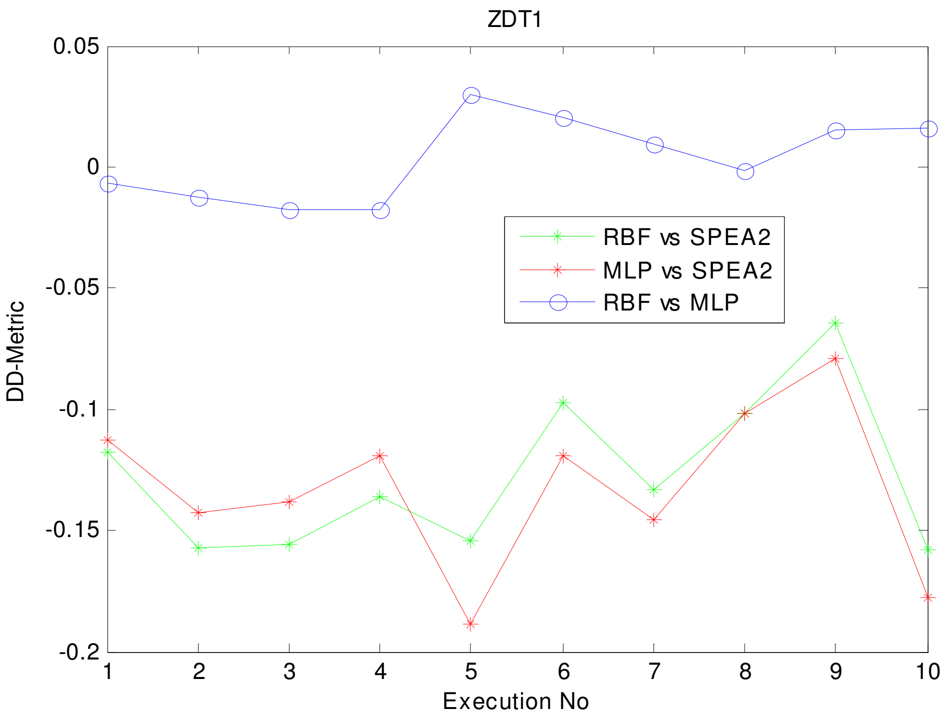
C.1 S-metric Values achieved by SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT1 at each of the 10 executions



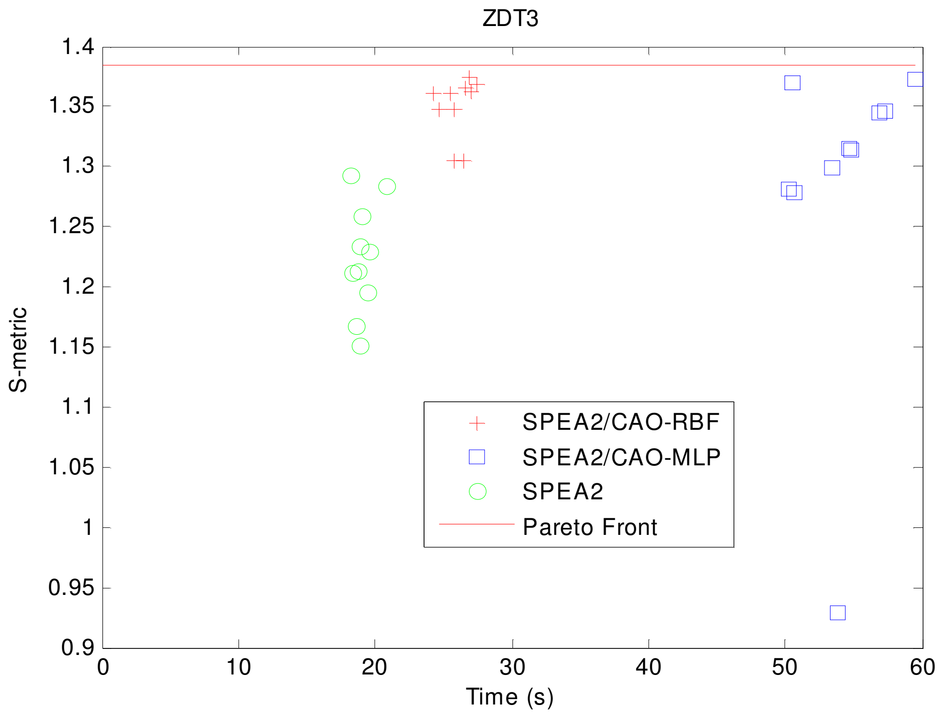
C.2 C-Metric Values achieved by SPEA2 and SPEA2/CAO-RBF on ZDT1 at each of the 10 executions



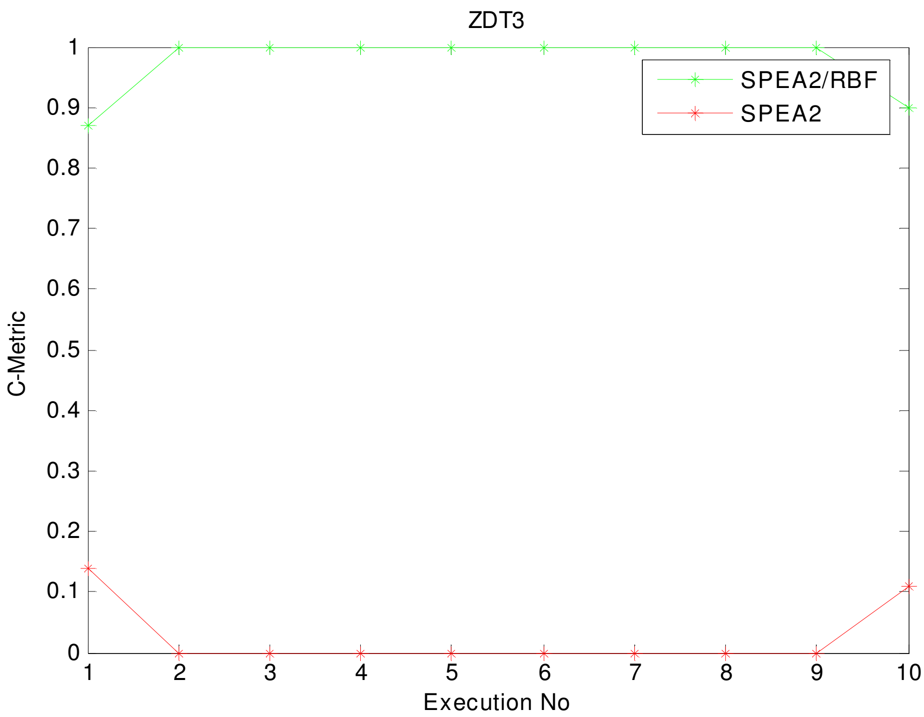
C.3 C-Metric Values achieved by SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT1 at each of the 10 executions



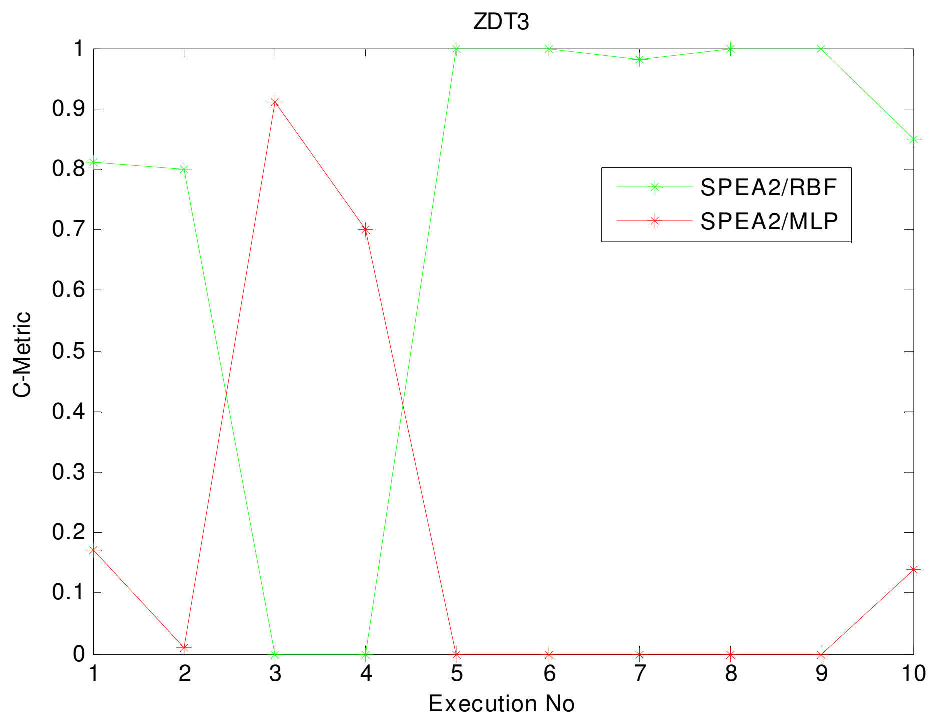
C.4 DD-Metric Values achieved by SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT1 at each of the 10 executions



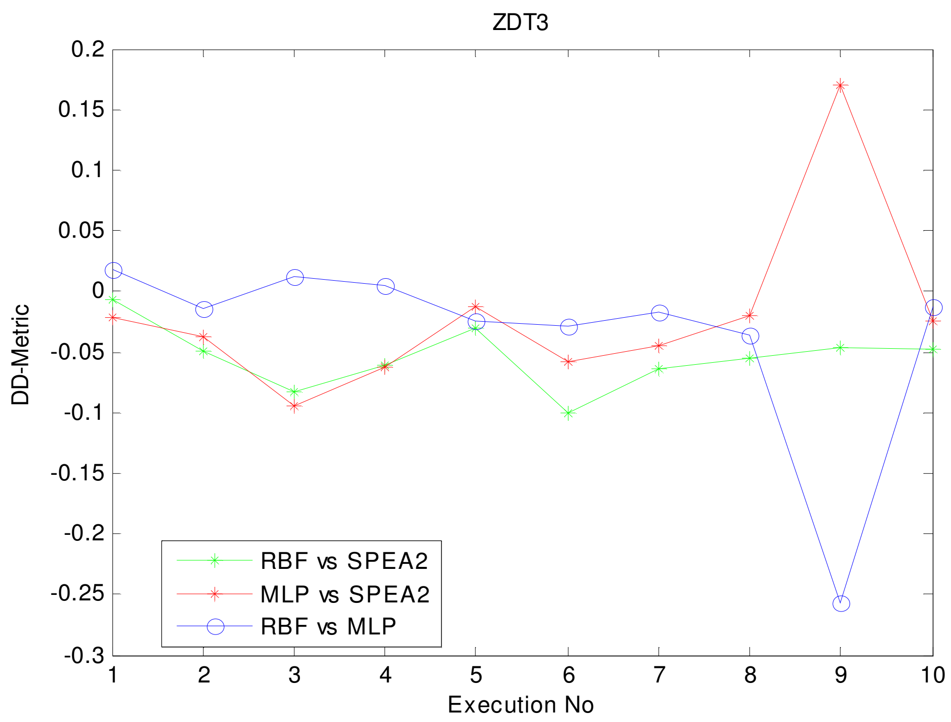
C.5 S-metric Values achieved by SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT3 at each of the 10 executions



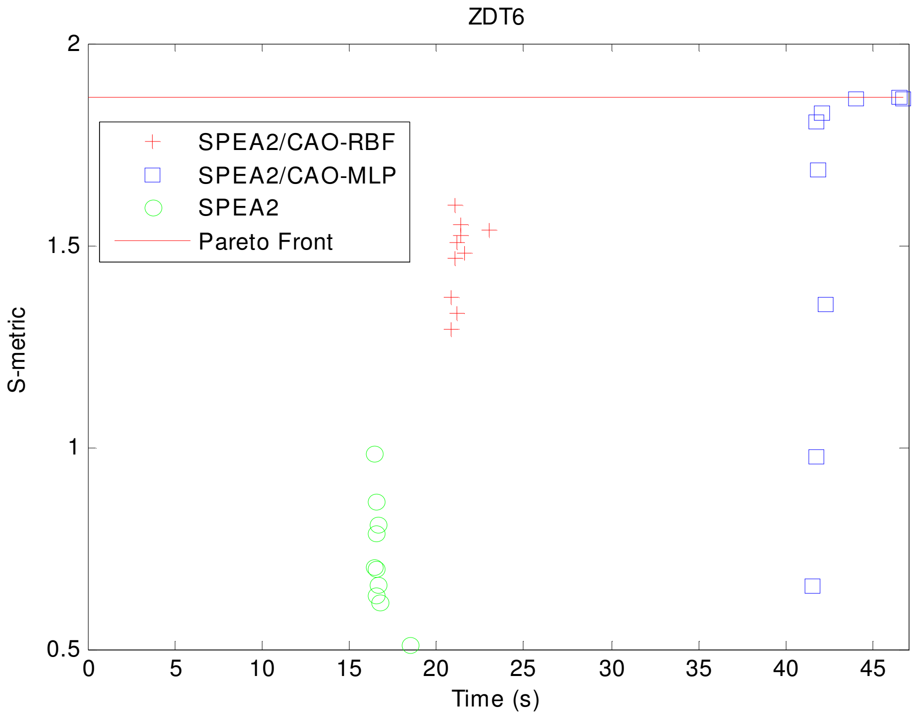
C.6 C-Metric Values achieved by SPEA2 and SPEA2/CAO-RBF on ZDT3 at each of the 10 executions



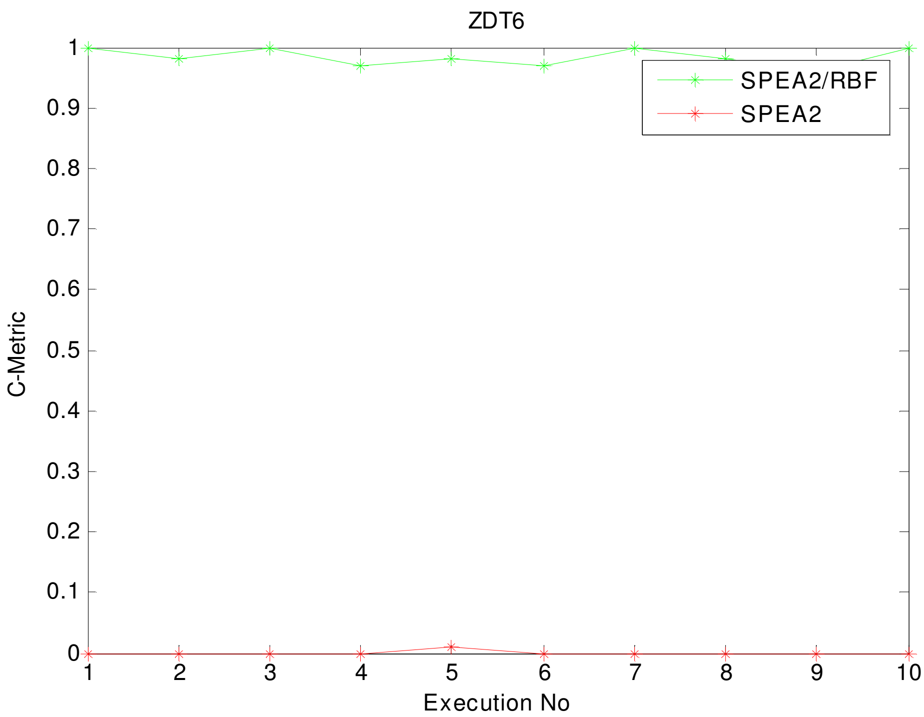
C.7 C-Metric Values achieved by SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT3 at each of the 10 executions



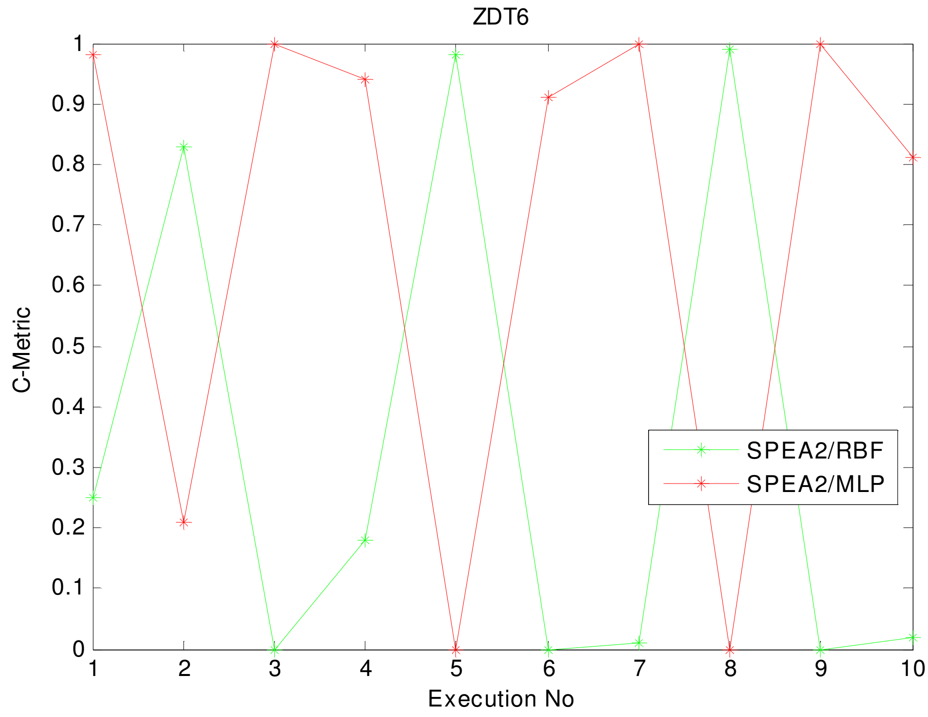
C.8 DD-Metric Values achieved by SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT3 at each of the 10 executions



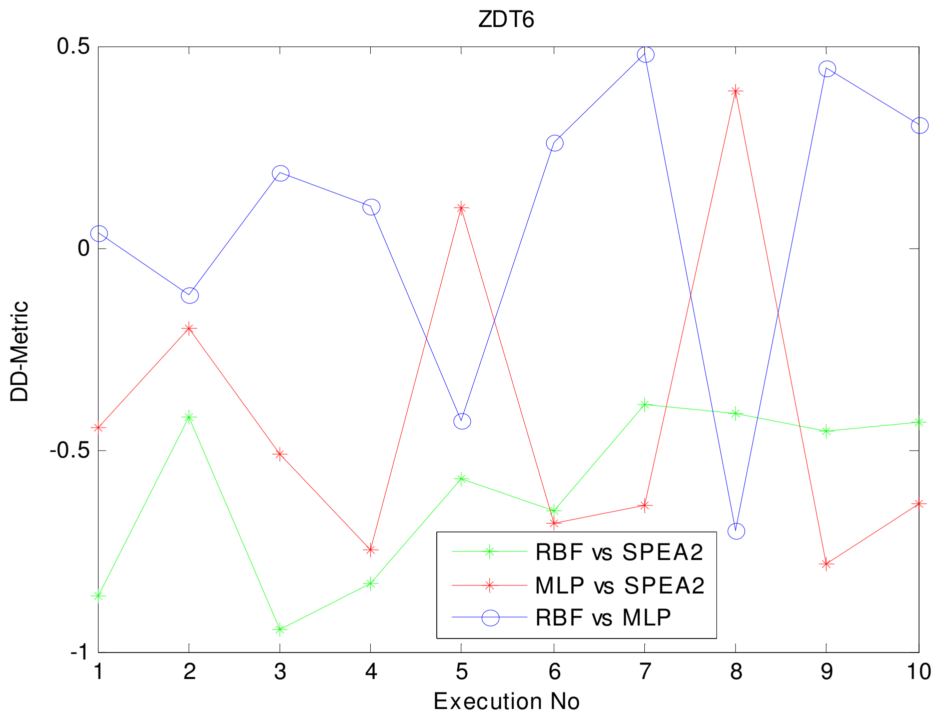
C.9 S-metric Values achieved by SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT6 at each of the 10 executions



C.10 C-Metric Values achieved by SPEA2 and SPEA2/CAO-RBF on ZDT6 at each of the 10 executions



C.11 C-Metric Values achieved by SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT6 at each of the 10 executions



C.12 DD-Metric Values achieved by SPEA2, SPEA2/CAO-RBF and SPEA2/CAO-MLP on ZDT6 at each of the 10 executions

Table C.1: C-metric results for DTLZ2 (3)

Run No:	DTLZ2 - 3 Objectives (A = SPEA2/CAO-RBF, B = SPEA2/CAO-MLP and C = SPEA2)			
	C-Metric (A, C)	C-Metric (C, A)	C-Metric (B, C)	C-Metric (C, B)
1	5%	0%	4%	0%
2	10%	1%	5%	1%
3	3%	0%	0%	1%
4	4%	2%	7%	0%
5	8%	1%	0%	0%
6	6%	0%	4%	0%
7	6%	0%	5%	0%
8	12%	0%	5%	2%
9	6%	2%	2%	0%
10	7%	0%	2%	2%
Mean Value:	6.7%	0.06%	3.4%	0.06%

Table C.2: DD-metric results for DTLZ2 (3)

Run No:	DTLZ2 - 3 Objectives (A = SPEA2/CAO-RBF, B = SPEA2/CAO-MLP and C = SPEA2)	
	DD-Metric (A, C) $.10^{-3}$	DD-Metric (B, C) $.10^{-3}$
1	-1.303	-0.881
2	-3.1246	-1.668
3	-0.423	0
4	0.980	-1.182
5	-1.368	0
6	-3.219	-0.457
7	-1.550	-1.863
8	-3.228	-1.803
9	2.472	-0.205
10	-4.131	-1.026
Mean Value:	-1.500	-0.9

Table C.3: C-metric results for DTLZ2 (8)

Run No:	DTLZ2 - 3 Objectives (A = SPEA2/CAO-RBF, B = SPEA2/CAO-MLP and C = SPEA2)			
	C-Metric (A, C)	C-Metric (C, A)	C-Metric (B, C)	C-Metric (C, B)
1	1%	0%	5%	0%
2	6%	0%	4%	0%
3	21%	0%	10%	0%
4	8%	0%	20%	0%
5	18%	0%	35%	0%
6	3%	0%	7%	0%
7	33%	0%	18%	0%
8	11%	0%	6%	0%
9	2%	0%	5%	0%
10	16%	0%	5%	0%
Mean Value:	12.9%	0%	11.5%	0%

Table C.4: DD-metric results for DTLZ2 (8)

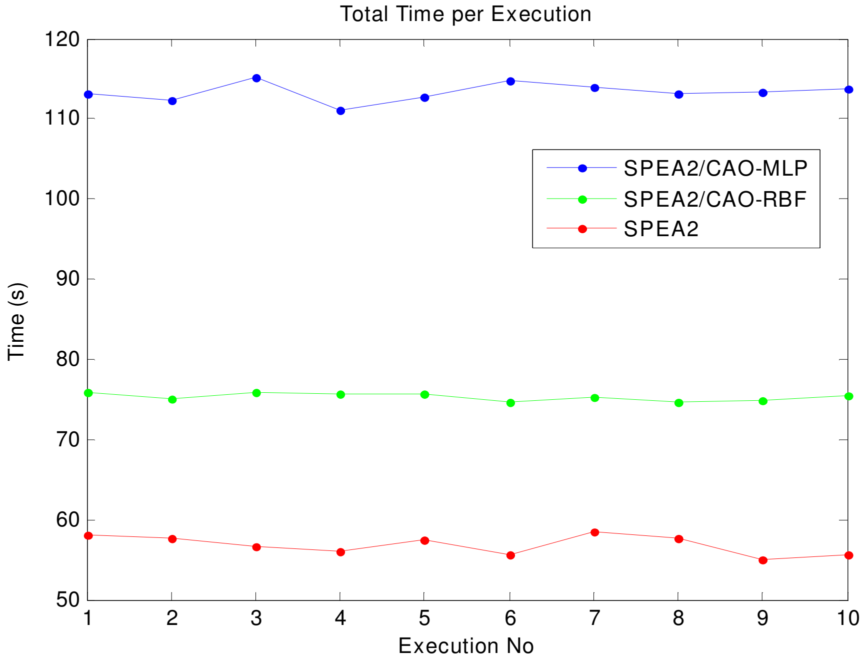
Run No:	DTLZ2 - 3 Objectives (A = SPEA2/CAO-RBF, B = SPEA2/CAO-MLP and C = SPEA2)	
	DD-Metric (A, C) $.10^{-3}$	DD-Metric (B, C) $.10^{-3}$
1	-50.30	-45.94
2	-49.45	-70.20
3	-209.41	-160.99
4	-270.92	-376.14
5	-190.24	-715.37
6	-37.59	-110.62
7	-807.08	-284.25
8	-205.77	-91.72
9	-54.01	-76.58
10	-95.05	-70.35
Mean Value:	-196.9820	-200.2160

Table C.5: C-metric results for DTLZ2 (12)

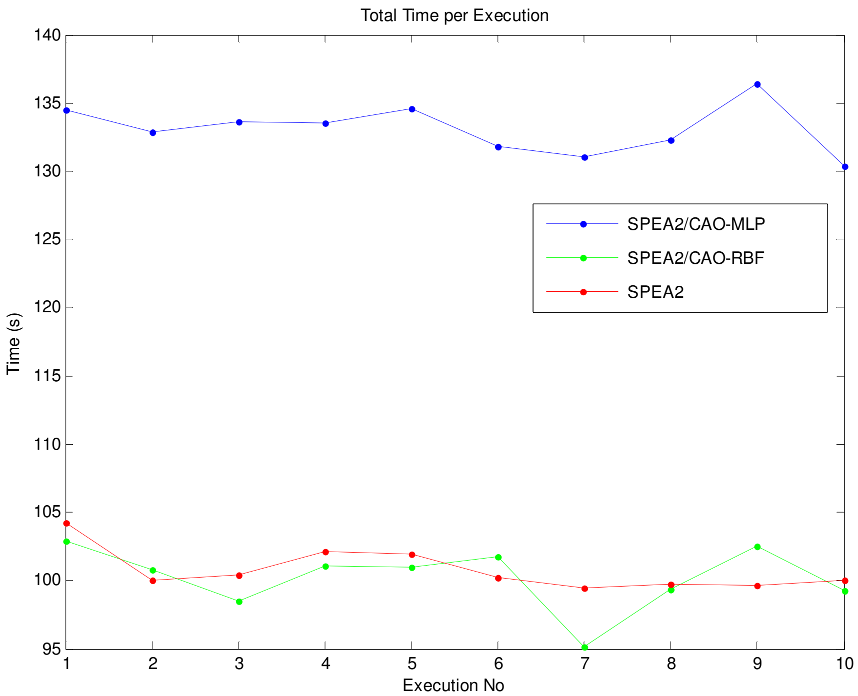
Run No:	DTLZ2 - 3 Objectives (A = SPEA2/CAO-RBF, B = SPEA2/CAO-MLP and C = SPEA2)			
	C-Metric (A, C)	C-Metric (C, A)	C-Metric (B, C)	C-Metric (C, B)
1	2%	0%	2%	0%
2	5%	0%	5%	0%
3	1%	0%	1%	0%
4	1%	0%	3%	0%
5	6%	0%	5%	0%
6	3%	0%	2%	0%
7	5%	0%	3%	0%
8	2%	0%	4%	0%
9	1%	0%	2%	0%
10	6%	0%	2%	0%
Mean Value:	2.7%	0%	2.9%	0%

Table C.6: DD-metric results for DTLZ2 (12)

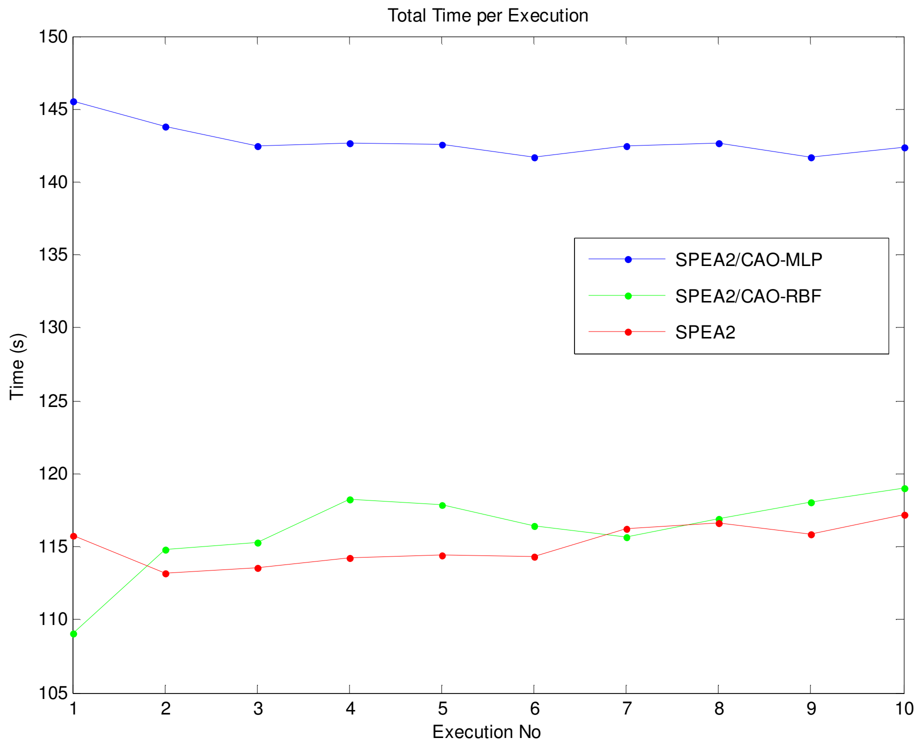
Run No:	DTLZ2 - 3 Objectives (A = SPEA2/CAO-RBF, B = SPEA2/CAO-MLP and C = SPEA2)	
	DD-Metric (A, C) $.10^{-3}$	DD-Metric (B, C) $.10^{-3}$
1	-12.620	-29.900
2	-5.921	-17.228
3	-32.017	-18.920
4	-18.742	-12.232
5	-26.200	-98.023
6	-51.027	-99.780
7	-67.332	-101.907
8	-179.892	-10.561
9	-28.920	-9.703
10	-100.320	-17.131
Mean Value:	-52.2991	-41.5385



C.13 Computational Time per execution for DTLZ2 (3)



C.14 Computational Time per execution for DTLZ2 (8)



C.15 Computational Time per execution for DTLZ2 (12)

Appendix D

Parallel Coordinates Graphs

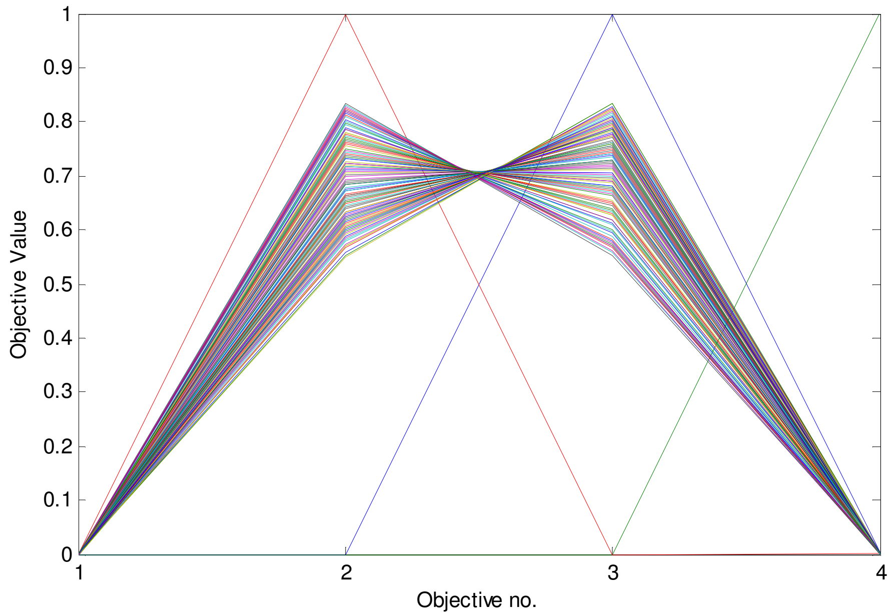


Figure D.1: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.11) for the scenario expressed using the guided dominance PPA technique (illustrated in Section 5.3.1)

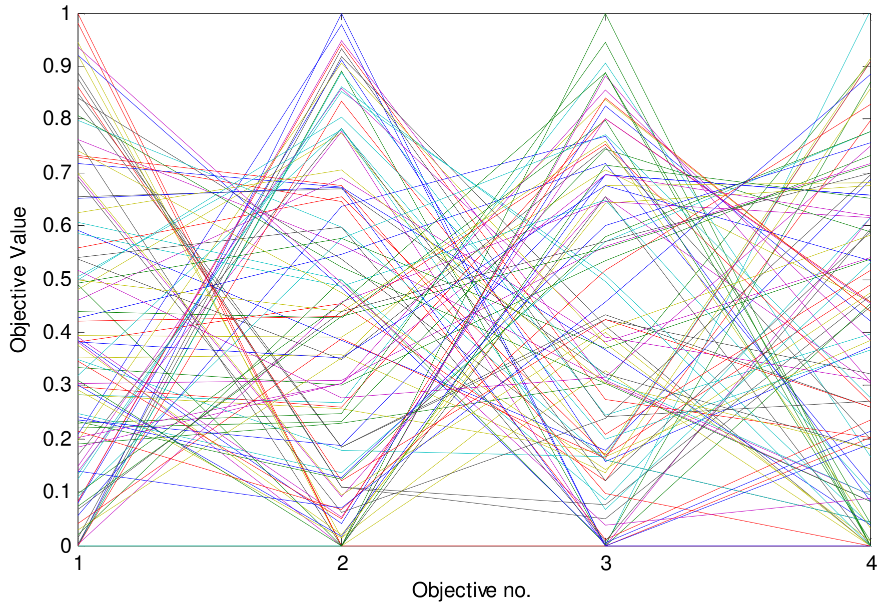


Figure D.2: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.12) for the scenario expressed using the guided dominance PPA technique (illustrated in Section 5.3.1)

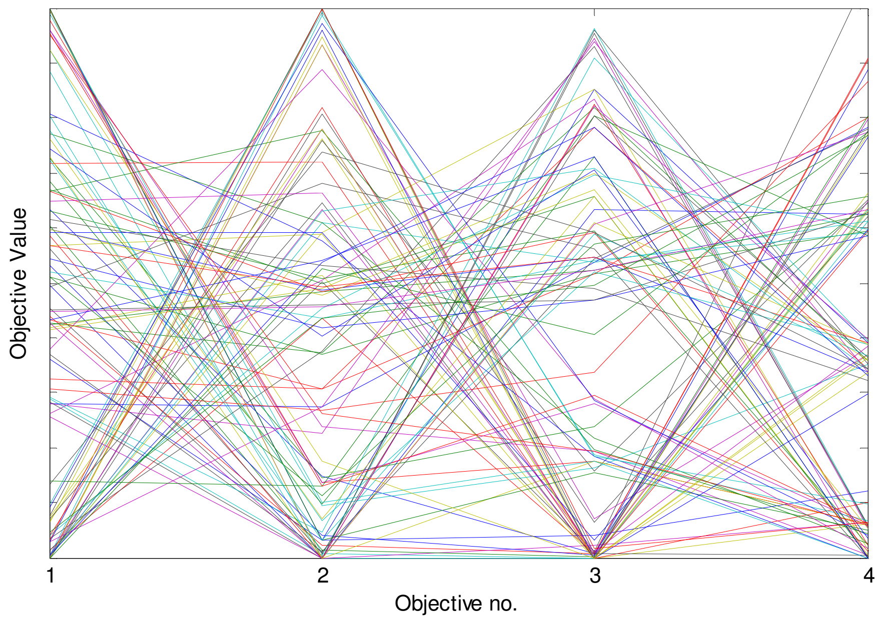


Figure D.3: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.18) for the scenario expressed using the biased crowding PPA technique (illustrated in Section 5.3.2)

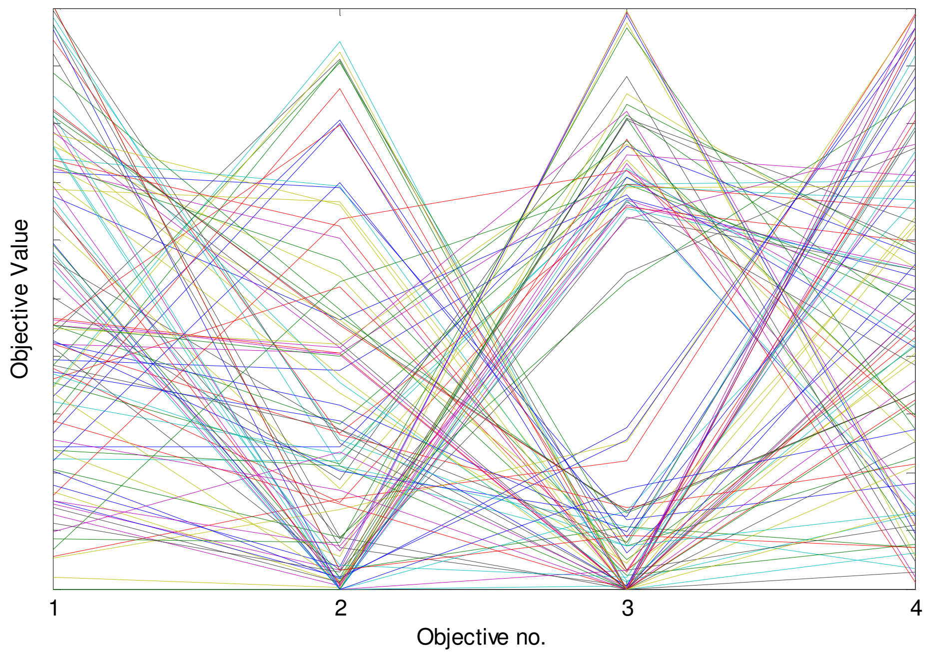


Figure D.4: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.19) for the scenario expressed using the biased crowding PPA technique (illustrated in Section 5.3.2)

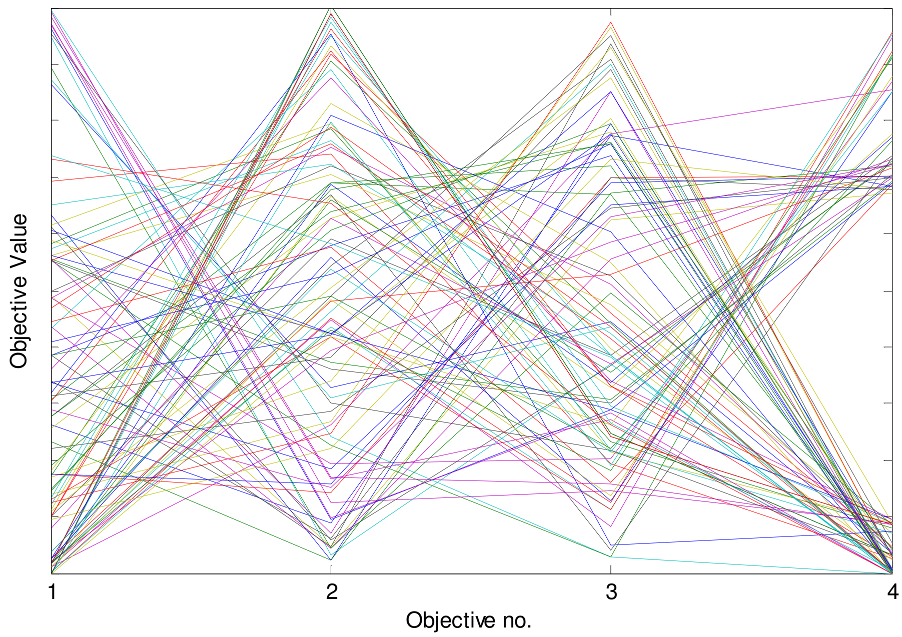


Figure D.5: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.20) for the scenario expressed using the biased crowding PPA technique (illustrated in Section 5.3.2)

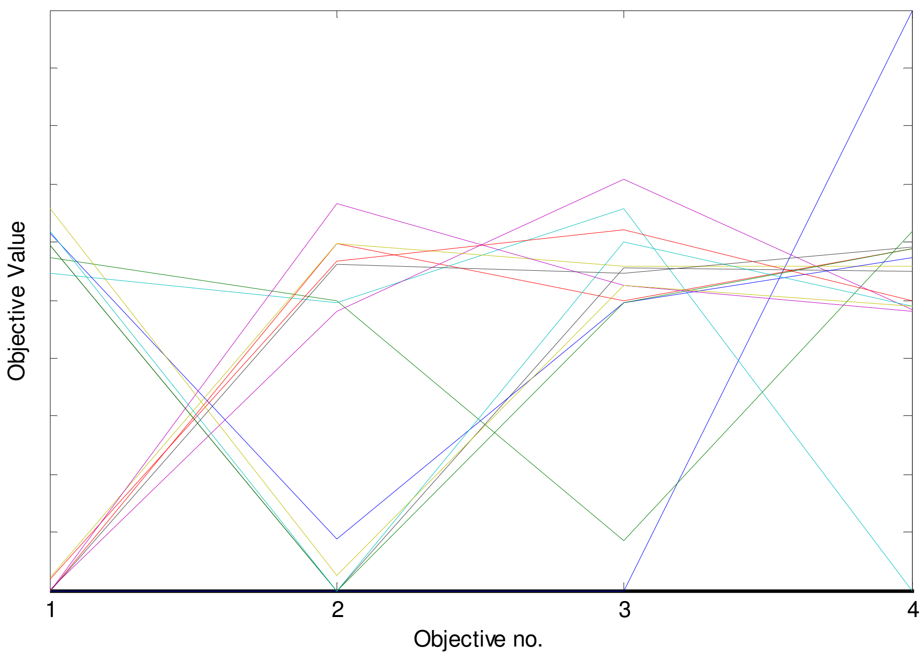


Figure D.6: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.31) for the scenario expressed using the ε-dominance based PPA technique (illustrated in Section 5.3.3)

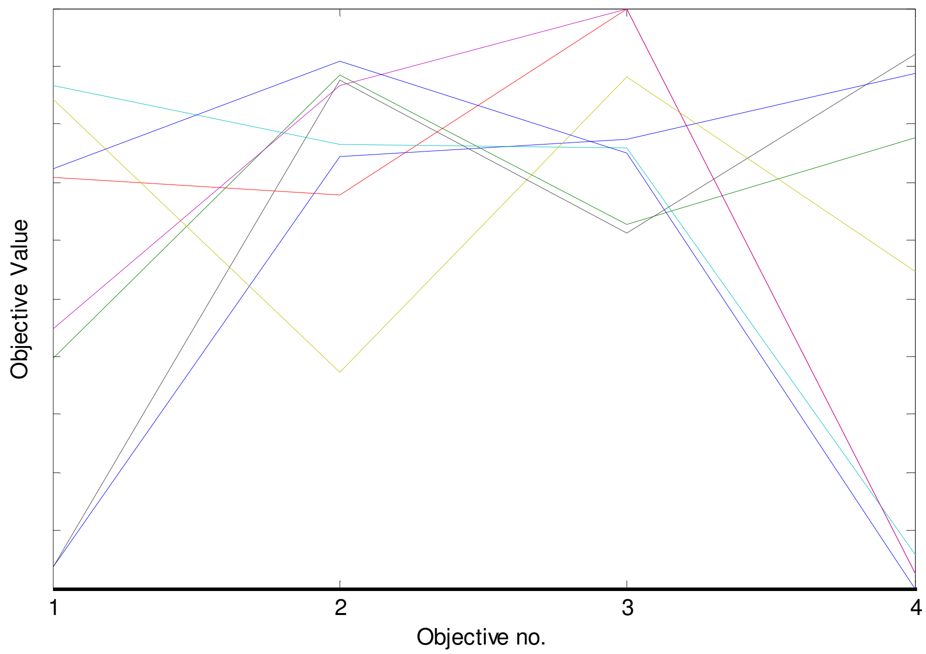


Figure D.7: Parallel Coordinates of the results achieved for DTLZ2 (4) (also presented in Figure 5.32) for the scenario expressed using the ε-dominance based PPA technique (illustrated in Section 5.3.3)

References

Abbass, H. A., Sarker, R. and Newton, C.: 2001, PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems, in IEEE Neural Networks Council (ed.), *Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001)*, Vol. 2, IEEE Service Center, Piscataway, New Jersey, pp. 971–978.

Abbass, H. A.: An Inexpensive Cognitive Approach for Biobjective Optimization using Bliss Points and Interaction, in Xin Yao et al. (editors), *Parallel Problem Solving from Nature - PPSN VIII*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 3242, pp. 712--721, Birmingham, UK, September 2004

Abboud, K. and Schoenauer, M.: 2002, Surrogate deterministic mutation: Preliminary results, in P. Collet, C. Fonlupt, J.-K. Hao, E. Lutton and M. Schoenauer (eds), *Proceedings of the Fifth European Conference on Artificial Evolution (EA 2001)*, Vol. 2310 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 104-116.

Adra, S. F., Griffin, I. and Fleming, P. J.: 2005a, Hybrid Multiobjective Genetic Algorithm with a New Adaptive Local Search Process, *Proceedings of the 2005 International Genetic and Evolutionary Computation Conference (GECCO2005)*, Washington DC.

Adra, S., F., Hamody, A., Griffin, I., and Fleming, P., J., 2005b, A Hybrid Multi-Objective Evolutionary Algorithm Using an Inverse Neural Network for Aircraft Control System Design. *In 2005 IEEE Congress on Evolutionary Computation (CEC 2005).* Edinburgh, Scotland: IEEE Service Center.

Adra, S. F., Griffin, I. and Fleming, P. J.: 2006, An Adaptive Memetic Algorithm for Enhanced Diversity, *Proceedings of the seventh conference in the Adaptive Computing in Design and Manufacture (ACDM 2006)*, Bristol, UK.

Adra, S. F., Griffin, I. and Fleming, P. J.: 2007a, A Comparative Study of Progressive Preference Articulation Techniques for Multiobjective Optimisation, *Proceedings of the fourth international Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, Japan, March, 2007

Adra, S. F., Griffin, I., and Fleming, P. J.: 2007b, A Convergence Accelerator for Multiobjective Optimisers, *Research Report No. 950*, The University of Sheffield.

Adra, S. F., Griffin, I. and Fleming, P. J.: 2007c, An Informed Convergence Accelerator for Evolutionary Multiobjective Optimisers, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*, London, July 2007.

Andersson, J.: *A Survey of Multiobjective Optimization in Engineering Design*. 2000, Department of Mechanical Engineering, Linkping University: Linkping, Sweden.

Aouni, B. and Kettani, O.: 2001, Goal programming model: A glorious past and a promising future, *European Journal of Operations Research* 133, 225 – 231.

Baker, J. E.: 1985, Adaptive selection methods for genetic algorithms, in J. J. Grefenstette (ed.), *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, New Jersey, pp. 101 – 111.

Baker, J. E.: 1987, Reducing bias and inefficiency in the selection algorithm, in J. J. Grefenstette (ed.), *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, New Jersey, pp. 14 – 21.

Bäck, T., Hoffmeister, F. and Schwefel, H.-P.: 1991, A survey of evolution strategies, in R. K. Belew and L. B. Booker (eds), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 2 – 9.

Bäck, T.: 1992, The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm, in R. M¨anner and B. Manderick (eds), *Parallel Problem Solving from Nature 2*, North-Holland, Amsterdam, pp. 85 – 94.

Bäck, T.: 1996, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, Oxford.

Benayoun, B.R.R. and Sussman, N.: *Manual de reference du programme electre, Note de Synthese et Formation*. Direction Scientifique SEMA, Paris, France, 1966. 25.

Beyer, H.-G. and Schwefel, H.-P.: 2002, Evolution strategies: A comprehensive introduction, *Natural Computing* 1, 35 – 52.

Bhanu, B., Lee, S. and Das, S.: Adapting Image Segmentation using Genetic and Hybrid Search Methods. *IEEE Transactions on Aerospace and Electronic Systems, 1995*. 31(4): p. 1268-1291.

Bishop, C.M.: 1995, *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press.

Blakelock, J. H.: 1995, *Automatic Control of Aircraft and Missile*, Wiley, New York (1995)

Blickle, T. and Thiele, L.: 1995, *A comparison of selection schemes used in genetic algorithms*, TIK-Report 11, TIK Institut fur Technische Informatik und Kommunikationsnetze, Computer Engineering and Networks Laboratory, ETH, Swiss Federal Institute of Technology.

Booker, L. B.: 1982, *Intelligent Behaviour as an Adaptation to the Task Environment*, PhD thesis, University of Michigan, Ann Arbor. Cited by Deb and Goldberg (1989).

Booker, L.: Improving search in genetic algorithms, *In Genetic Algorithms and Simulated Annealing*, L. Davis (Ed.), pp. 61-73, Morgan Kaufmann Publishers, 1987.

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

Bramlette, M. F.: Initialization, Mutation and Selection Methods in Genetic Algorithms for Function Optimization, *Proc ICGA 4*, pp. 100-107, 1991.

Branke, J., Kaußler, T. and Schmeck, H.: 2001, Guidance in evolutionary multi-objective optimization, *Advances in Engineering Software* 32, 499 – 507.

Branke, J. and Deb, K.: 2004, Integrating user preferences into evolutionary multiobjective optimization, in Y. Jin (ed.), *Knowledge Incorporation in Evolutionary Computation*, Springer, Berlin, pp. 461 – 477.

Brans, J. P., Vincke, P. and Mareschal, B.: 1986, How to select and how to rank projects: The PROMETHEE method, *European Journal of Operations Research* 24, 228 – 238.

Brockhoff, D. and Zitzler, E.: 2006a, Are all Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization, *Proc. of the PPSN IX*, Springer, Berlin, pp. 533-542.

Brockhoff, D. and Zitzler, E.: 2006b, On Objectives Conflicts in Multiple Criteria Optimization, *TIK Report 243*.

Bull, L.: 1999, On model-based evolutionary computation, *Soft Computing* 3, 76 – 82.

Brindle, A.: 1981, *Genetic Algorithms for Function Optimization*, PhD thesis, University of Alberta, Edmonton. Cited by Goldberg and Deb (1991).

Burke, E.K., Cowling, P., Silva, J. D. L. and Petrovic, S.: Combining Hybrid Metaheuristics and Populations for the Multiobjective Optimisation of Space Allocation Problems. In

Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001). 2001. San Francisco, California: Morgan Kaufmann Publishers.

Caruana, R. A., Eshelman, L. A. and Schaffer, J. D.: Representation and hidden bias II: Eliminating defining length bias in genetic search via shuffle crossover, In *Eleventh International Joint Conference on Artificial Intelligence*, N. S. Sridharan (Ed.), Vol. 1, pp. 750-755, Morgan Kaufmann Publishers, 1989.

Chafekar, D., Shi, Rasheed, K. and Xuan, J.: 2005, Multiobjective GA optimization using reduced models, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 35(2), 261-265

Charnes, A. and Cooper, W. W.: 1961, *Management Models and Industrial Applications of Linear Programming*. John Wiley and Sons, Inc., New York, NY.

Chellapilla, K. and Fogel, D. B.: 2001, Evolving an expert checkers playing program without using human expertise, *IEEE Transactions on Evolutionary Computation* 5(4), 422 – 428.

Chen, J-H., Goldberg, D.E., Ho, S-Y. and Sastry, K.: 2002, Fitness inheritance in multi-objective optimization. In: *Proceedings of genetic and Evolutionary Computation Conference*. Morgan Kaufmann

Censor, Y.: 1977, Pareto optimality in multiobjective problems, *Applied Mathematics and Optimization* 4, 41–59.

Chen, W., Allen, J., Mavris, D. and Mistree, F.: 1996, A concept exploration method for determining robust top-level specifications, *Engineering Optimization* 26, 137-158.

Chipperfield, A. J. and Fleming, P. J.: 1995, Parallel genetic algorithms, in A. Zomaya (ed.), *Parallel and Distributed Computing Handbook*, McGraw-Hill, New York, pp. 1034–1059.

Coello, C. A. C.: 1999, A comprehensive survey of evolutionary-based multiobjective optimization techniques, *Knowledge and Information Systems. An International Journal* 1(3), 269–308.

Coello, C. A. C.: 2000a, Handling preferences in evolutionary multiobjective optimization: a survey, in IEEE Neural Networks Council (ed.), *Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000)*, Vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 30–37.

Coello, C. A.: 2000b, Treating constraints as objectives for single-objective evolutionary optimization, *Engineering Optimization* 32(3), 275 – 308.

Coello, C. A. C. and Toscano Pulido, G.: 2001, A micro-genetic algorithm for multiobjective optimization, in E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello and D. Corne (eds), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Springer-Verlag, Berlin, pp. 126–140.

Coello, C.A.C., Veldhuizen, D.A.V., and Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2002, New York: Kluwer Academic Publishers.

Coello, C. A.: 2002, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering* 191(11), 1245 – 1287.

Coello, C. A.: 2006, Evolutionary multi-objective optimization: A historical view of the field, *IEEE Computational Intelligence Magazine* pp. 28 – 36.

Cohon, J. L. and Marks, D. H.: 1975, Evaluation of multiobjective programming techniques, *Water Resources Research* 11(2), 208 – 229.

Corne, D. W., Knowles, J. D. and Oates, M. J.: 2000, The Pareto envelope-based selection algorithm for multiobjective optimization, in M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo and H.-P. Schwefel (eds), *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Springer-Verlag, Berlin, pp. 839–848.

Courant, R.: 1943, Variational methods for the solution of problems of equilibrium and vibration, *Bulletin of the American Mathematical Society* 49, 1 – 23.

Craenen, B.G.W. and Eiben, A.E.: Computational Intelligence, online document consulted in 2006 May 28, *Encyclopedia of Life Support Sciences*, EOLSS, EOLSS Co. Ltd.

C.Varcol, M. and Emmerich, M.: 2005, Metamodel-assisted evolution strategies applied in electromagnetics, in R. Schilling, W. Haase, J. P'eriaux and H. Baier (eds), *Proceedings of the European Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems* (EUROGEN) 2005.

Cvetkovic, D. and Parmee, I. C.: 1999, Genetic algorithm-based multi-objective optimisation and conceptual engineering design, *Proceedings of the Congress on Evolutionary Computation (CEC) 1999*, IEEE, IEEE Press, pp. 29 – 36.

Cvetkovic, D. and Parmee, I. C.: 2002, Preferences and their application in evolutionary multiobjective optimization, *IEEE Transactions on Evolutionary Computation* 6(1), 42 – 57.

Dantzig, G. B.: 1963, *Linear Programming and Extensions*, Princeton University Press, New Jersey.

Darwin, C.: 1859, *The Origin of Species*, John Murray, London.

Davis, F. D.: 1989, Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quart, 13, 319-339.

Davis, L. (ed.): 1991, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

Dawkins, R.: 1976, *The Selfish Gene*, Oxford University Press, Oxford.

Deb, K. and Goldberg, D. E.: 1989, An investigation of niche and species formation in genetic function optimization, in J. D. Schaffer (ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 42 – 50.

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

Deb, K. and Goyal, M.: 1996, A combined genetic adaptive search (geneAS) for engineering design, *Computer Science and Informatics* 26(4), 30 – 45.

Deb, K.: 1999a, Evolutionary algorithms for multi-criterion optimization in engineering design, in K. Miettinen, M. M. Mäkelä, P. Neittaanmäki and J. Periaux (eds), *Evolutionary Algorithms in Engineering and Computer Science*, John Wiley & Sons, Chichester, UK, chapter 8, pp. 135–161.

Deb, K.: 1999b, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, *Evolutionary Computation* 7(3), 205 – 230.

Deb, K., Agrawal, S., Pratab, A. and Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, In Marc Schoenauer et al., editor, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T.: (2000a), A fast and elitist multi-objective genetic algorithm: NSGA-II. *Technical Report 2000001*, Indian Institute of Technology, Kanpur Genetic Algorithms Laboratory (KanGAL).

Deb, K.: 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York.

Deb, K. and Goel, T.: 2001, Controlled elitist non-dominated sorting genetic algorithms for better convergence, in E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello and D. Corne (eds), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Springer-Verlag, pp. 67–81.

Deb, K.: 2001b, Nonlinear goal programming using multi-objective genetic algorithms, *Journal of the Operational Research Society* 52(3), 291–302.

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

Deb, K., Thiele, L., Laumanns, M. and Zitzler, E.: 2002, Scalable multi-objective optimization test problems, *Proceedings of the Congress on Evolutionary Computation (CEC) 2002*, IEEE, IEEE Press, pp. 825 – 830.

Deb, K., Mohan, M. and Mishra, S.: 2003, Towards a quick computation of well-spread Pareto-optimal solutions, in C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele (eds), *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, Springer-Verlag, Berlin, pp. 222–236.

Deb, K. and Saxena, D.K.: 2005, *On Finding Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems*. 2005, Kanpur Genetic Algorithms Laboratory (KanGAL) Indian Institute of Technology Kanpur: Kanpur.

Deb, K., Sundar, J., N, U. B. R. and Chaudhuri, S.: 2006, Reference point based multi-objective optimization using evolutionary algorithms, *International Journal of Computational Intelligence Research* 2(3), 273 – 286.

DeJong, K. A.: 1975, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD thesis, University of Michigan.

De Jong, K. A. and Sarma, J.: Generation Gaps Revisited, In *Foundations of Genetic Algorithms 2*, L. D. Whitley (Ed.), Morgan Kaufmann Publishers, 1993.

Drechsler, N., Drechsler, R. and Becker, B.: 2001, Multi-objective optimisation based on relation favour, in E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello and D. Corne (eds), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Springer-Verlag, pp. 154–166.

Ducheyne, E. I., Baets, B. D. and De Wulf, R. R.: 2003, Is Fitness Inheritance Useful for Real-World Applications?, in Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele (editors), *Evolutionary Multi-Criterion Optimization.*

Second International Conference, EMO 2003, pp. 31--42, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, April 2003 .

Edgeworth, F.Y., *Mathematical psychics*. An essay on the application of mathematics to the moral sciences. 1932, London: The London School of Economics and Political Science.

Eiben, A. E., Raué, P. E. and Ruttkay, Z.: Genetic Algorithms with Multi-parent Recombination, *Parallel Problem Solving from Nature* – PPSN III, pp.78-87, Springer, 1994.

Eiben, A. E. and Schippers, C. A.: 1998, On evolutionary exploration and exploitation, *Fundamenta Informaticae* 35, 1– 16.

El-Beltagy, M.A., Nair, P.B. and Keane, A.J.: 1999, Metamodeling Techniques for Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations. in *Proc. of the Genetic and Evolutionary Computation Conf. GECCO-99*. 1999. San Francisco, CA: Morgan Kaufmann.

Emmerich, M. and Naujoks, B.: Metamodel-assisted multiobjective optimisation strategies and their application in airfoil design. In I. C. Parmee, editor, *Adaptive Computing in Design and Manufacture VI*, pages 249-260, Springer, London, 2004

Eshelman, L. J. and Schaffer, J. D.: 1993, Crossover's niche, in S. Forrest (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, California, pp. 9–14.

Etkin, B.: 1972, *Dynamics of Atmosheric Flight*, Wiely, New York (1972).

Farina, M. A.: 2002, A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm. in *Congress on Evolutionary Computation (CEC 2002)*. Piscataway, New Jersey: IEEE Service Center.

Farina, M. and Amato, P.: 2002, On the optimal solution definition for many-criteria optimization problems, in J. Keller and O. Nasraoui (eds), *Proceedings of the NAFIPS-FLINT International Conference 2002*, IEEE Press, pp. 233 –238.

Farina, M. and Amato, P.: 2004, A fuzzy definition of "optimality" for many-criteria optimization problems, *IEEE Transactions on System, Man and Cybernetics - Part A: Systems and Humans* 34(3), 315 – 326.

Fieldsend, J. E. and Everson, R. M.: 2005, Multi-objective Optimisation in the Presence of Uncertainty, in *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, pp. 243--250, Vol. 1, IEEE Service Center, Edinburgh, Scotland, September 2005.

Fisher, R.: 1930, *The Genetical Theory of Natural Selection*, Clarendon Press, Oxford.

Fleischer, M.: 2003, The measure of Pareto optima. applications to multi-objective metaheuristics, in C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele (eds), *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, Springer, Berlin, pp. 519–533.

Fleming, P. J. and Pashkevich, A. P.: 1985, Computer aided control system design using a multiobjective optimization approach, *Proceedings of the IEE Control '85 Conference*, pp. 174–179.

Fleming, P.J, Purshouse, R. C., and Lygoe, R. J.: Many-Objective Optimization: An Engineering Design Perspective. in Evolutionary Multi-Criterion Optimization. *Third International Conference, EMO 2005*. 2005. Guanajuato, Mexico: Springer. Lecture Notes in Computer Science Vol. 3410.

Fletcher, R.: 1981, *Practical Methods of Optimization*, Volume 2: Constrained Optimization, John Wiley & Sons, New York.

Friedberg, R. M.: 1958, A learning machine: Part I, *IBM Journal* 2(1), 2 – 13.

Friedberg, R. M., Dunham, B. and North, J. H.: 1959, A learning machine: Part 2, *IBM Journal* 3(7), 282 – 287.

Fogarty, T. C.: An Incremental Genetic Algorithm for Real-Time Learning, *Proc. 6th Int. Workshop on Machine Learning*, pp. 416-419, 1989.

Fogel, L. J., Owens, A. J. and Walsh, M. J.: 1966, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, New York.

Fonseca, C. M. and Fleming, P. J.: 1993, Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization, in S. Forrest (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 416 – 423.

Fonseca, C. M. and Fleming, P. J.: 1995, Multiobjective genetic algorithms made easy: Selection, sharing, and mating restriction, in A. M. S. Zalzala (ed.), *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications* (GALESIA 95), Institution of Electrical Engineers, Stevenage, UK, pp. 42–52.

Fonseca, C. M. and Fleming, P. J.: 1998, Multiobjective optimization and multiple constraint handling with evolutionary algorithms — Part I: A unified formulation, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans* 28(1), 26–37.

Fourman, M. P.: 1985, Compaction of symbolic layout using genetic algorithms, in J. Grefenstette (ed.), *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Mahwah, New Jersey, pp. 141–153.

Garzon, V. Honavar, M. Jakiela and R. E. Smith (eds), *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO 99)*, Vol. 2, Morgan Kaufmann Publishers, San Francisco, California, pp. 1738–1743.

Gaspar-Cunha, A., and Vieira, A., S.: 2004, A Hybrid Multi-Objective Evolutionary Algorithm Using an Inverse Neural Network. in Hybrid Metaheuristics, *First International Workshop, HM 2004*. Valencia, Spain: IOS Press.

Gaspar-Cunha, A., Vieira, A.S. and Fonseca, C.M.: 2004, Multi-Objective Optimization: Hybridization of an Evolutionary Algorithm with Artificial Neural Networks for fast Convergence, *Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics*, November, Nottingham, UK, 2004.

Gaspar-Cunha, A., and Vieira, A., S.: 2005, A Multi-Objective Evolutionary Algorithm Using Neural Network to Approximate Fitness Evaluations, in International Journal of Computers, Systems and Signals, *Vol. 6, No*.

Ghosh, A. and Dehuri, S.N.: 2004, Evolutionary algorithms for multicriterion optimization: A survey. *Intl. Journal. Computing and Information Sciences*, 2:1. 38-56.

Glover, F. and Laguna, M.: 1997, *Tabu Search*. Kluwer, Norwell, MA.

Goldberg, D. E. and Richardson, J.: 1987, Genetic algorithms with sharing for multimodal function optimization, in J. J. Grefenstette (ed.), *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, New Jersey, pp. 41 – 49.

Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning*. 1989, Reading, MA: Addison-Wesley.

Goldberg, D. E.: 1991, Real-coded genetic algorithms, virtual alphabets and blocking, *Complex Systems 5*, 139 – 167.

Goldberg, D. E. and Deb, K.: 1991, A comparative analysis of selection schemes used in genetic algorithms, in G. J. E. Rawlins (ed.), *Foundations of Genetic Algorithms*, Morgan Kaufmann, pp. 69 – 93.

Grefenstette, J. J.: 1984, GENESIS: a system for using genetic search procedures. *Proceedings of the Conference on Intelligent Systems and Machines Rochester*, MI, Pages 161–165

Griffin, I. A., Schroder, P., Chipperfield, A. J. and Fleming, P. J.: 2000, Multi-objective optimisation approach to the ALSTOM gasifier problem, Proc Inst Mech Engrs, Part I: J of Sys and Contr. Eng, vol 214, 2000, pp453-468

Haas, O.C.L., Burnham, K. J. and Mills, J.A.: Hybrid Optimisation Technique for Radiotherapy Treatment Planning. in *Proceedings of the 1998 IEEE International Conference on Control Applications*. 1998. Trieste, Italy.

Hadj-Alouane, A. B. and Bean, J. C.: 1997, A genetic algorithm for the multiple-choice integer program, *Operations Research* 45(1), 92 – 101.

Haimes, Y.Y., Lasdon, L.S. and Wismer, D.A.: On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man, and Cybernetics* 1(3), 1971, 296–297.

Hajela, P. and Lin, C. Y.: Genetic search strategies in multicriterion optimal design, *Structural Optimization*, 4:99-107, 1992.

Hancock, P. J. B.: 1994, An empirical comparison of selection methods in evolutionary algorithms, in T. C. Fogarty (ed.), *Evolutionary Computing - AISB Workshop*, Vol. 865 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 80 – 94.

Helbig, S. and Pateva, D.: 1994, On several concepts for ε-efficiency, *OR Spektrum* 16(3), 179–186.

Herrara, F., Lozano, M. and Verdegay, J. L.: 1998, Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis, *Artificial Intelligence Review* 12, 265 – 319.

Hillier, F. S. and Lieberman, G. J.: *Introductions to Operations Research*, 7th edition, McGraw-Hill 2001.

Hodgson, R.J.W: Memetic Algorithms and the Molecular Geometry Optimization Problem. *in Proc. of the 2000 Congress on Evolutionary Computation. 2000*. Piscataway, NJ: IEEE Service Center.

Holland, J. H.: 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Horn, J. and Nafpliotis, N.: 1993, *Multiobjective optimization using the niched Pareto genetic algorithm*, IlliGAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois.

Horn, J., Nafpliotis, N. and Goldberg, D. E.: 1994, A niched Pareto genetic algorithm for multiobjective optimization, *Proceedings of the Congress on Evolutionary Computation (CEC)* 1994, IEEE, IEEE Press, pp. 82 – 87.

Horn, J.: 1997, Multicriteria decision making, in T. B¨ack, D. B. Fogel and Z. Michalewicz (eds), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, pp. F1.9:1 – F1.9:15.

Hornik, K., Stinchcombe, M. and White, H.: 1989, Multilayer feedforward networks are universal approximators, Neural Networks 2(5), 359 – 366. IEE: 2005, *Proceedings of the Congress on Evolutionary Computation (CEC)* 2005, IEEE Press.

Huband, S., Barone, L., While, L., and Hingston, P.: A Scalable Multi-objective Test Problem Toolkit, in Evolutionary Multi-Criterion Optimization. *Third International Conference, EMO 2005*. 2005. Guanajuato, Mexico: Springer. Lecture Notes in Computer Science Vol. 3410, pp. 280-295.

Hughes, E. J.: 2001, Evolutionary multi-objective ranking with uncertainty and noise, in E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello and D. Corne (eds), *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, Springer-Verlag, pp. 329–343.

Husbands, P.: Distributed Coevolutionary Genetic Algorithms for multi-Criteria and Multi-Constraint Optimisation, In Terence C. Fogarty, editor, Evolutionary Computing. *AIS Workshop*. Selected Papers, Lecture Notes in Computer Science, vol. 865, pp. 150–165. Springer Verlag, Apr. 1994.

Hwang, A.S.M.M. C. L.: *Multiple Objective Decision Making-Methods and Applications*. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1979. 164.

Ikeda, K., Kita, H. and Kobayashi, S.: 2001, Failure of Pareto-based MOEAs: Does nondominated really mean near to optimal?, in IEEE Neural Networks Council (ed.), *Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001)*, Vol. 2, IEEE Service Center, Piscataway, New Jersey, pp. 957–962.

Inselberg, A.: 1985, The plane with parallel coordinates, *The Visual Computer* 1, 69–91.

Ito, K., Akagi, S. and Nishikawa, M.: A multiobjective optimization approach to design problem of heat insulation for thermal distribution piping network systems, *Journal of*

Mechanisms, Transmissions and Automation in Design (Transactions of the ASME) 105: 206-213, June 1983.

Jakob, W., Gorges-Schleuter, M. and Blume, C.: 1992, Application of genetic algorithms to task planning snd learning, in R. M¨anner and B. Manderick (eds), *Parallel Problem Solving from Nature 2*, North-Holland, Amsterdam, pp. 291 – 300.

Jin, Y., Olhofer, M. and Sendhoff, B.: 2000, On evolutionary optimization with approximate fitness function, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)* 2000, Morgan Kaufmann, pp. 786 – 792.

Jin, R., Chen, W. and Simpson, T. W.: 2001, Comparative studies of metamodelling techniques under multiple modelling criteria, *Structural and Multidisciplinary Optimization* 23(1), 1 – 13.

Jin, Y.: 2005, A comprehensive survey of fitness approximation in evolutionary computation, *Soft Computing* 9, 3 – 12.

Kendall, M. G.: 1938, A new measure of rank correlation, *Biometrika* 30, 81–93.

Khare, V., Yao, X. and Deb, K.: 2003, Performance scaling of multi-objective evolutionary algorithms, in C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele (eds), *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, Springer, Berlin, pp. 376–390.

Kirkpatrick, S. and Gelatt, C. D. and Vecchi, M. P.: 1983, *Optimization by Simulated Annealing, Science*, Vol 220, Number 4598, pages 671-680, 1983.

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

Knowles, J.D.: 2002, *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*, PhD Thesis, The University of Reading: Reading, UK.

Knowles, J. D. and Corne, D. W.: 2002, On metrics for comparing nondominated sets, in IEEE Neural Networks Council (ed.), *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 711–716.

Knowles, J. D. and Corne, D. W.: 2003b, Properties of an adaptive archiving algorithm for storing nondominated vectors, *IEEE Transactions on Evolutionary Computation* 7(2), 100–116.

Knowles, J. D. and Corne, D. W.: 2007, Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, Springer, Berlin, pp. 757–771.

Krasnogor, N.: *Studies on the Theory and Design Space of Memetic Algorithms*. Ph.D. Thesis, University of the West of England, Bristol, 2000.

Krasnogor, N and Smith, J.: A Memetic Algorithm With Self-Adaptive Local Search: TSP as a case study. *Proceedings of the 2000 International Genetic and Evolutionary Computation Conference (GECCO 2000)*, 2000.

Krasnogor, N. and Gustafson, S.: Towards Truly “memetic“ Memetic Algorihms: discussion and proof of concepts. In: D. Corne, G. Fogel, W. Hart, J. Knowles, N. Krasnogor, R. Roy, J. E. Smith, and A. Tiwari (eds.): *Advances in Nature-Inspired Computation. The PPSN 7 Workshops*. PEDAL (Parallel, Emergent and Distributed Architectures Lab). University of Reading, 2002.

Krasnogor, N. and Gustafson, S.: A Study on the use of “Self-Generation” in Memetic Algorithms. *Natural Computing*, 3 (1): 53-76, 2004.

Kursawe, F.: 1991, A variant of evolution strategies for vector optimization, in H.-P. Schwefel and R. Männer (eds), *Parallel Problem Solving from Nature* 1, Vol. 496 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 193 – 197.

Laumanns, M., Thiele, L., Deb, K. and Zitzler, E.: 2002a, Combining convergence and diversity in evolutionary multi-objective optimization, *Evolutionary Computation* 10(3), 263–282.

Laumanns, M. and Ocenasek, J.: 2002b, Bayesian optimization algorithms for multi objective optimization, in J. J. Merelo Guervós, P. Adamidis, H.-G. Beyer, J.-L. Fernández-Villacaňas and H.-P. Schwefel (eds*), Proceedings of the Parallel Problem Solving from Nature VII Conference*, Springer-Verlag, Berlin, pp. 298–307.

Laumanns, M., Zitzler, E. and Thiele, L.: 2001, On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization, in E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello and D. Corne (eds), Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001), Springer-Verlag, Berlin, pp. 181–196.

Laumanns, M., Zitzler, E. and Thiele, L.: 2000, A unified model for multi-objective evolutionary algorithms with elitism, in IEEE Neural Networks Council (ed.), Proceedings of the 2000 Congress on Evolutionary Computation (CEC 2000), Vol. 1, IEEE Service Center, Piscataway, New Jersey, pp. 46–53.

Leonard, J. A., Kramer, M. A., and Ungar, L. H.: 1992, Neural network architecture that computes its own reliability. *Computers and Chemical Engineering*. Vol. 16, no. 9, pp. 819-835.

Liepins, G. E. and Potter, W. D.: 1991, A genetic algorithm approach to multiplefault diagnosis, in L. Davis (ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, pp. 237 – 250.

Lindgaard, G.: Usability Testing and System Evaluation: A Guide for Designing Useful Computer Systems. London and New York: Chapman & Hall, 1994.

Lobo, F. G. and Goldberg, D. E.: 2001, *The parameter-less genetic algorithm in practice*, IlliGAL Report 2001022, University of Illinois at Urbana-Champaign, Urbana, Illinois.

Lohn, J. D., Kraus, W. F. and Haith, G. L.: 2002, Comparing a coevolutionary genetic algorithm for multiobjective optimization, in IEEE Neural Networks Council (ed.), *Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002)*, Vol. 2, IEEE Service Center, Piscataway, New Jersey, pp. 1157–1162.

Luenberger, D. G.: *Linear and nonlinear programming*, 2nd edition, Addison-Wesley 1984.

Manly, B.F.J.: 1991, *Randomization and Monte Carlo Methods in Biology*. 1991, London: Chapman and Hall.

Massebeuf, S., Fonteix, C., Kiss, L. N., Marc, I., Pla, F. and Zaras, K.: 1999, Multicriteria optimization and decision engineering of an extrusion process aided by a diploid genetic algorithm, in P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao and A. Zalzala (eds), *Proceedings of the Congress on Evolutionary Computation (CEC) 1999*, IEEE, IEEE Press, pp. 14 – 21.

McCulloch, W. S. and Pitts, W. H.: 1943, A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115-133.

Mezura-Montes, E. E., Coello, C. A. C., and Velázquez-Reyes, J.: 2006, Increasing Successful Offspring and Diversity in Differential Evolution for Engineering Design, in *Proc. Adaptive Computing in Design and Manufcature (ACDM 2006)*, Bristol, UK, 2006.

Michalewicz, Z.: 1996, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin.

Michalewicz, Z. and Schoenauer, M.: 1996, Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation* 4(1), 1 – 32.

Michalewicz, Z. and Fogel, D. B.: 2000, *How to Solve It: Modern Heuristics*, Springer, Berlin.

Miettinen, K.: 1999, *Nonlinear Multiobjective Optimization*, Kluwer, Boston.

Miller, G. A.: 1956, The magical number seven, plus or minus two: Some limits on our capacity for processing information, *The Psychological Review* 63(2), 81 – 97.

Miller, B. L. and Goldberg, D. E.: 1995, *Genetic algorithms, tournament selection, and the effects of noise*, IlliGAL Report 95006, University of Illinois.

Moscato, P.: 1989, *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*, C3P Report 826, Caltech.

Mühlenbein, H. and Schlierkamp-Voosen, D.: Predictive Models for the Breeder Genetic Algorithm, *Evolutionary Computation*, Vol. 1, No. 1, pp. 25- 49, 1993.

Nabney, I. T.: 2001, *NETLAB: Algorithms for Pattern Recognition*, Series: Advances in Pattern Recognition, 1st ed. 2001. Corr. 3rd printing, 2004, ISBN: 978-1-85233-440-6

Nariman-Zadeh, N., Atashkari, K., Jamali, A., Pilechi, A. and Yao, X.: 2005, Inverse modelling of multi-objective thermodynamically optimized turbojet engines using GMDH-type neural networks and evolutionary algorithms, *Engineering Optimization*, Vol. 37, No. 5, pp. 437--462, July 2005.

Osyczka, A. and Kundu, S.: A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm, *Structural Optimization*, 10, pp 94-99, 1995

Papadimitriou, C. H. and Yannakakis, M.: 2000, On the approximability of trade-offs and optimal access of Web sources, *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*, IEEE Computer Society, Los Alamitos, California, pp. 86–92.

Pareto, V.: *Manual of Political Economy*. 1906, 1971 translation of 1927 edition, New York: Augustus M. Kelley.

Parmee, I. C., Cvetkovic, D., Watson, A. H. and Bonham, C. R.: 2000, Multiobjective satisfaction within an interactive evolutionary design environment, *Evolutionary Computation* 8(2), 197 – 222.

Patterson, D.: 1996, Artificial Neural Networks, Theory and Applications. Prentice-Hall, Inc.

Pelikan, M., Goldberg, D. E. and Lobo, F.: 1999, *A survey of optimization by building and using probabilistic models*, IlliGAL Report 99018, University of Illinois at Urbana-Champaign, Urbana, Illinois.

Price, K. and Storn, R.: 1995, *Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report Technical Report TR-95-012, International Computer Science Institute, Berkeley.

Pryke, A., Mostaghim, S. and Nazemi, A.: 2007, Heatmap Visualization of Population Based Multi Objective Algorithms. Proceedings of the Evolutionary Multi-criterion Optimization conference (EMO 2007): 361-375

Purshouse, R.C. and Fleming, P. J.: 2003a, An Adaptive Divide-and-Conquer Methodology for Evolutionary Multi-criterion Optimisation. In *Proceedings of the Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*. 2003a. Faro, Portugal: Springer. Lecture Notes in Computer Science. Volume 2632.

Purshouse, R. C. and Fleming, P. J.: 2003b, Evolutionary many-objective optimisation: An exploratory analysis, in IEEE Neural Networks Council (ed.), *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, IEEE Service Center, Piscataway, New Jersey.

Purshouse, R. C.: 2004, *On the Evolutionary Optimisation of Many Objectives*, PhD thesis, University of Sheffield.

Rachmawati, L. and Srinivasan, D.: Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey, in 2006 *IEEE Congress on Evolutionary Computation (CEC'2006)*, pp. 3385--3391, IEEE, Vancouver, BC, Canada, July 2006.

Rasheed, K., Ni, X. and Vattam, S.: 2003, Comparison of methods for developing dynamic reduced models for design optimization, *Proceedings of the Congress on Evolutionary Computation (CEC) 2002*, IEEE Press, pp. 390 – 395.

Ray, T., Kang, T. and Chye, S. K.: 2001, Multiobjective design optimization by an evolutionary algorithm, *Engineering Optimization 33*(3), 399–424.

Rechenberg, I.: 1973, *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*, Frommann-Holzboog, Stuttgart.

Reed, P., Minsker, B. S. and Goldberg, D. E.: 2000, Designing a competent simple genetic algorithm for search and optimization, *Water Resources Research* 36(12), 3757–3761.

Regis, R. G. and Shoemaker, C. A.: 2004, Local function approximation in evolutionary algorithms for the optimization of costly functions, *IEEE Transactions on Evolutionary Computation* 8(5), 490 – 505.

Rekiek, B., Lit, P. D., Fabrice, P., L'Eglise, T., Emanuel, F. and Delchambre, A.: 2000, Dealing with user's preferences in hybrid assembly lines design, *Proceedings of MCPL*'2000.

Rosenberg, R. S.: 1967, *Simulation of genetic populations with biochemical properties*. Ph.D. Thesis, University of Michigan Press, Ann Arbor, MI.

Robert, C.P. and Casella, G.: *Monte Carlo Statistical Methods* (second edition). New York: Springer-Verlag, 2004.

Rojas, R.: Neural Networks – A Systematic Introduction. New York: Springer-Verlag, 1996.

Rudolph, G. and Agapie, A.: 2000, Convergence properties of some multi-objective evolutionary algorithms, *Proceedings of the Congress on Evolutionary Computation (CEC) 2000*, IEEE, IEEE Press, pp. 1010 – 1016.

Sarker, R., Liang, K. and Newton, C.: 2002, A new evolutionary algorithm for multiobjective optimization, *European Journal of Operational Research* 140(1), 12–23.

Sastry K, Goldberg D. E. and Pelikan, M.: 2001, Don't evaluate, inherit. In: *Proceedings of genetic and Evolutionary Computation Conference*, Morgan Kaufmann, pp. 551–558.

Schaffer, J. D.: 1985, Multiple objective optimization with vector evaluated genetic algorithms, in J. J. Grefenstette (ed.), *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, New Jersey, pp. 93-100.

Silverman, B. W.: 1986, Density Estimation for Statistics and Data Analysis, Vol. 26 of *Monographs on Applied Statistics and Probability*, Chapman and Hall, London.

Simpson, T. W., Mauery, T. M., Korte, J. J. and Mistree, F.: 1998, *Comparison of response surface and kriging models for multidisciplinary design optimization*, Technical Report AIAA-98-4755, American Institute of Aeronautics and Astronautics.

Spears, W. M.: 1998, *The Role of Mutation and Recombination in Evolutionary Algorithms*, PhD thesis, George Mason University.

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

Syswerda, G.: Uniform crossover in genetic algorithms, *Proc. ICGA 3*, pp.2-9, 1989.

Tabak, D., Schy, A. A., Giesy, D.P. and Johnson, K.G.: 1979, Application Of Multiobjective Optimization in Aircraft Control Systems Design, Pergamon Press Ltd. 1979, *Automatica* Vol. 15, pp. 595~600.

Takagi, H.: 2001, Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation, *Proceedings of the IEEE* 89(9), 1275 – 1296.

Tamaki, H., Mori, M., Araki, M., Mishima, Y. and Ogai, H.: 1994, Multi-criteria optimization by genetic algorithms: A case of scheduling in hot rolling process, in M. Fushimi and K. Tone (eds), *Proceedings of the Third Conference of the Association of Asian-Pacific Operational Research Societies within IFORS (APORS 94)*, World Scientific Publishing, pp. 374–381.

Tan, K. C., Lee, T. H. and Khor, E. F.: 1999, Evolutionary algorithms with goal and priority information for multi-objective optimization, in IEEE Neural Networks Council (ed.), *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99)*, IEEE Service Center, Piscataway, New Jersey, pp. 106–113.

Tan, K. C., Khor, E. F., Lee, T. H. and Sathikannan, R.: 2003, An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization, *Journal of Artificial Intelligence Research* 18, 183–215.

Tate, D. M. and Smith, A. E.: Expected Allele Convergence and the Role of Mutation in Genetic Algorithms, *Proc. ICGA 5*, pp.31-37, 1993.

Thierens, D. and Bosman, P. A. N.: 2001, Multi-objective optimization with iterated density estimation evolutionary algorithms using mixture models, *Proceedings of the Third International Symposium on Adaptive Systems — Evolutionary Computation and Probabilistic Graphical Models*, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba, pp. 129–136.

Todd, D. S. and Sen, P.: 1999, Directed multiple objective search of design spaces using genetic algorithms and neural networks, in W. Banzhaf, J. Daida, A. E. Eiben, M. H. Yu, P. L.: 1973, A class of solutions for group decision problems, *Management Science* 19(8), 936 – 946.

Ulmer, H., Streichert, F. and Zell, A.: 2003, Evolution strategies assisted by Gaussian processes with improved pre-selection criterion, *Proceedings of the Congress on Evolutionary Computation (CEC) 2003*, IEEE Press, pp. 692 – 699.

Van Veldhuizen, D. A.: 1999, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, PhD thesis, Air Force Institute of Technology.

Van Veldhuizen, D. A. and Lamont, G. B.: 2000a, Multiobjective Optimization with Messy Genetic Algorithms, *Proceedings of the 2000 ACM Symposium on Applied Computing SAC (1)* 2000: 470-476

Van Veldhuizen, D. A. and Lamont, G. B.: 2000b, On measuring multiobjective evolutionary algorithm performance, *Proceedings of the Congress on Evolutionary Computation (CEC)* 2000, IEEE, IEEE Press, pp. 204 – 211.

While, L.: 2005, A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume, *Third International Conference, EMO 2005*. 2005. Guanajuato, Mexico: Springer. Lecture Notes in Computer Science Vol. 3410.

Wienke, D., Lucasius, C. and Kateman, G.: 1992, Multicriteria target vector optimization of analytical procedures using a genetic algorithm - part I. theory, numerical simulations and application to atomic emission spectroscopy, *Analytica Chimica Acta* 265, 211 – 225.

Wierzbicki, A. P.: 1980, The use of reference objectives in multiobjective optimization, in G. Fandel and T. Gal (eds), *Multiple Criteria Decision Making Theory and Applications*, Springer-Verlag, Berlin, pp. 468 – 486.

Wright, A. H.: "Genetic Algorithms for Real Parameter Optimization", In *Foundations of Genetic Algorithms*, J. E. Rawlins (Ed.), Morgan Kaufmann, pp. 205-218, 1991.

Xunxue Cui and Chuang Lin. MOCGA: A Preferences-based Multiobjective Concordance Genetic Algorithm. *Journal of Software*, May 2005, 15(5): 761-770.

Zitzler, E.: 1999, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

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

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

Zitzler, E., Laumanns, M. and Thiele, L.: 2001, *SPEA2: Improving the strength Pareto evolutionary algorithm*, TIK Report 103, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Zitzler, E.: 2002, A tutorial on evolutionary multiobjective optimization, Invited Talk, 2002 *Workshop on Multi-Objective Meta-Heuristics*. 4–5 November 2002, Paris.

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

Zitzler, E., Laumanns, M. and Bleuler, S.: 2004, A tutorial on evolutionary multiobjective optimization, in X. Gandibleux, M. Sevaux, K. S¨orensen and V. T'Kindt (eds), *Methaheuristics for Multiobjective Optimisation*, Vol. 535 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin.