PREFERENCE-BASED FLEXIBLE MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İBRAHİM KARAHAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTERS OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JUNE 2008

Approval of the thesis

"**PREFERENCE-BASED FLEXIBLE MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS**"

submitted by **İbrahim Karahan** in partial fulfillment of the requirements for the degree of **Masters of Science in Industrial Engineering, Middle East Technical University** by,

Prof. Canan Özgen _____
Director, **Graduate School of Natural and Applied Sciences**

Prof. Nur Evin Özdemirel _____
Head of Department, **Industrial Engineering**

Prof. Murat Köksalan _____
Supervisor, **Industrial Engineering, METU**

Examining Committee Members:

Assoc. Prof. Canan Sepil _____
Industrial Engineering, METU

Prof. Murat Köksalan _____
Industrial Engineering, METU

Assoc. Prof. Esra Karasakal _____
Industrial Engineering, METU

Asst. Prof. Pelin Bayındır _____
Industrial Engineering, METU

Prof. Sencer Yeralan _____
Agricultural and Biological Engineering, University of Florida

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : İbrahim Karahan

Signature :

ABSTRACT

PREFERENCE-BASED FLEXIBLE MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Karahan, İbrahim

M.S., Department of Industrial Engineering

Supervisor: Prof. Murat Köksalan

June 2008, 96 pages

In this study, we develop an elitist multiobjective evolutionary algorithm for approximating the Pareto-optimal frontiers of multiobjective optimization problems. The algorithm converges the true Pareto-optimal frontier while keeping the solutions in the population well-spread over the frontier. Diversity of the solutions is maintained by the territory defining property of the algorithm rather than using an explicit diversity preservation mechanism. This leads to substantial computational efficiency. We test the algorithm on commonly used test problems and compare its performance against well-known benchmark algorithms.

In addition to approximating the entire Pareto-optimal frontier, we develop a preference incorporation mechanism to guide the search towards the decision maker's regions of interest. Based on this mechanism, we implement two variants of the algorithm. The first gathers all preference information before the optimization stage to find approximations of the desired regions. The second one is an interactive algorithm that focuses on the desired region by interacting with the decision maker during the solution process. Based on tests on 2- and 3-objective problems, we observe that both algorithms converge to the preferred regions.

Keywords: Multiobjective Evolutionary Algorithms, Multiobjective Optimization, Preference Incorporation, Interactive

ÖZ

TERCİHE DAYALI ESNEK ÇOK AMAÇLI EVRİMSEL ALGORİTMALAR

Karahan, İbrahim

Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Murat Köksalan

Haziran 2008, 96 sayfa

Bu çalışmada çok amaçlı eniyileme problemlerinin etkin yüzeylerine yaklaşmayı amaçlayan bir çok amaçlı evrimsel algoritma geliştirilmiştir. Algoritma, problemlerin gerçek etkin yüzeylerine yaklaşırken popülasyonundaki çözümleri yüzey üzerinde iyi dağıtmaktadır. Bunu yaparken dıştan dağılım sağlama mekanizması yerine algoritmanın alan kontrol özelliği kullanılmaktadır. Böylece önemli miktarda hesaplama verimliliği sağlanmaktadır. Algoritma yaygın kullanılan test problemleri üzerinde denenmiş ve başarımı literatürdeki diğer çok amaçlı evrimsel algoritmalar ile karşılaştırılmıştır.

Etkin yüzeyin tamamına yaklaşmanın yanı sıra, aramayı karar vericinin ilgilendiği alanlara yöneltmeyi amaçlayan bir tercih entegrasyonu mekanizması geliştirilmiştir. Bu mekanizmaya dayalı olarak yukarıda belirtilen algoritmanın iki çeşidi önerilmektedir. Bunlardan ilki bütün tercih bilgilerini çözüm sürecinin başında almaktadır. Diğeri ise karar verici ile eniyileme sürecinde etkileşimde bulunarak istenilen alanlara yönelmektedir. İki ve üç amaçlı problemler üzerinde yapılan testlerin sonucunda, algoritmaların istenilen alanlara başarıyla yaklaştığı görülmüştür.

Anahtar Kelimeler: Çok Kriterli Evrimsel Algoritmalar, Çok Amaçlı Optimizasyon, Tercihe Dayalı, Etkileşimli

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my thesis supervisor, Prof. Murat Köksalan, who guided me with his deep knowledge and experience in this study. He was always very helpful and involved, and I count myself very fortunate to work with such an excellent researcher. I hope that we will be able to work together in future studies.

I owe thanks to Prof. Meral Azizoğlu for her sincere support and friendship. She is a great guide who did her best for helping me in deciding my future academic career. She always believed and encouraged me that I can do even better. In addition, I would like to thank Prof. Sencer Yeralan for his cheerful presence and advices on my prospective life in *his* country. I would also like to thank my examining committee members, Assoc. Prof. Canan Sepil, Assoc. Prof. Esra Karasakal and Asst. Prof. Pelin Bayındır, for reviewing and contributing to this study.

It was a great honor for me to be a part of METU-IE as a research assistant. It would not be so enjoyable and full of remembrance without my friends in METU-IE. I would like to extend my thanks especially for my dear friends Melih Çelik and Güvenç Değirmenci for their endless patience for my random questions. I am also thankful to all my professors in the department for their contributions to my life.

I am grateful to my parents Yaşar Karahan and Cennet Karahan for their endless love and support. I also present my special thanks to my sister Hamide Karahan Turan and my brother Agâh Reha Turan, who contributed to my anxiety by providing me the opportunity to observe their lives as PhD students.

I would like to thank TÜBİTAK (The Scientific and Technological Research Council of Turkey) for the scholarship they have provided me for my graduate study.

My love and my best friend Nihan deserves my deepest gratitudes. She was present in every stage of this study with her ideas, support and patience. Without her, neither this study nor my life would be so complete.

To my family and my dearest

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

Many real-life problems are actually multiobjective, since they involve more than one objective that have to be optimized simultaneously. To solve them, many mathematical modeling-based approaches have been proposed. Although they are effective in finding nondominated solutions, they usually do not generate multiple solutions in a single run. Multiobjective evolutionary algorithms (MOEAs) overcome this shortcome. These algorithms imitate the genetic evolution in nature and apply it to optimization. They work with a population of solutions and progress with their genetic evolution. Consequently, they find multiple nondominated solutions in a single optimization run.

One of the goals of MOEAs is to minimize the proximity of solutions to the true Pareto-optimal frontier. In addition, they have to maintain the diversity among the solutions in the population so that a good approximation of the Pareto-optimal frontier is obtained with a limited number of solutions. The performance of an algorithm not only depends on how it converges, but also on how it keeps the diversity. Of course, the computational burden of achieving these should always be considered while designing the algorithm.

Recently, a third goal of converging to the desired portions of the Pareto-optimal frontier has emerged. It is claimed that the nondominated solutions found by an MOEA in fact do not provide any insight into the decision making (Coello, 2000). Furthermore, by incorporating the decision maker's preferences into the search procedure, an algorithm can obtain better details about the regions of interest. Such a method also avoids much of the computational issues related to the approximation of the entire Pareto-optimal frontier.

Considering these goals, we propose a fast MOEA, the Territory Defining Evolutionary Algorithm (TDEA), that converges to the true Pareto-optimal frontier while maintaining a uniform diversity among solutions. In addition, we propose two preference-based versions of the algorithm that are capable of focusing on the preference regions either interactively

or with prespecified preference information.

We begin by presenting some preliminary definitions and reviewing related literature on MOEAs in Chapter 2. This chapter also surveys the literature on preference incorporation in MOEAs.

In Chapter 3, we introduce TDEA and give the details of the algorithm. We also compare the performance of TDEA to well-known MOEAs in the literature on 2-, 3- and 5-objective problems.

In Chapter 4, we describe the preference incorporation mechanism incorporated into TDEA. We also present the preference-based versions of TDEA and test their performances on 2- and 3-objective test problems.

Lastly, we conclude by pointing out the contributions of this study and give future research directions in Chapter 5.

CHAPTER 2

DEFINITIONS AND LITERATURE REVIEW

In this chapter, we introduce the definitions that we use throughout this study, together with the related literature on multiobjective evolutionary algorithms. In Section 2.1, we define basic concepts related to multiobjective optimization. It will be followed by a literature review of the studies on multiobjective evolutionary algorithms in Section 2.2.

2.1 Definitions

A multiobjective optimization problem (MOP) has multiple objective functions, each is either maximized or minimized. As in the single objective optimization problems, there may be some constraints that must be satisfied. In its general form, a multiobjective optimization problem can be formulated as follows:

$$\text{“Maximize”} \quad \mathbf{z} = \mathbf{f}(\mathbf{x}) \tag{2.1}$$

$$\text{subject to} \quad \mathbf{x} \in \mathbf{X} \tag{2.2}$$

where,

$\mathbf{f}(\mathbf{x}) = (f_1(x), \ldots, f_i(x), \ldots, f_p(x))^T$	p-vector of objective functions
$\mathbf{x} = (x_1, \ldots, x_n)^T$	decision vector
$\mathbf{X} \subseteq \Re^n$	feasible decision space
$\mathbf{z} = \mathbf{f}(\mathbf{x})$	objective vector
$\mathbf{Z} = \mathbf{f}(\mathbf{X})$	feasible objective space (solution space)

$\Re^n$ may be restricted with constraints of the following types to form $\mathbf{X}$:

$\mathbf{g}(\mathbf{x}) \geq 0$	inequality type constraints
$\mathbf{h}(\mathbf{x}) = 0$	equality type constraints

Unlike single objective optimization problems, most multiobjective optimization problems do not have a single solution that optimizes all p objectives. Instead, the search is for finding those solutions for which improvement in one objective can only occur with the worsening of at least one other objective. These are called *nondominated solutions*.

2.1.1 Dominance and Efficiency

Without loss of generality, we will assume that the objectives are of maximization type in the following definitions (For further details, see Steuer (1986)):

Definition 2.1. Let $\mathbf{z}^1, \mathbf{z}^2 \in \mathbf{Z}$ be two objective vectors. Then, $\mathbf{z}^1$ *dominates* $\mathbf{z}^2$ if and only if $z_i^1 \geq z_i^2$ for all i and $z_i^1 > z_i^2$ for at least one i.

Definition 2.2. Let $\mathbf{z}^1, \mathbf{z}^2 \in \mathbf{Z}$ be two objective vectors. Then, $\mathbf{z}^1$ *strongly dominates* $\mathbf{z}^2$ if and only if $z_i^1 > z_i^2$ for all i.

Nondominated solutions are those that are not dominated by any other solution. Their decision vectors are said to be efficient. We present the formal definitions below:

Definition 2.3. Let $\mathbf{z} \in \mathbf{Z}$ be an objective vector. Then, $\mathbf{z}$ is *nondominated* if and only if no $\mathbf{z}' \in \mathbf{Z}$ dominates it. Otherwise, $\mathbf{z}$ is said to be *dominated*.

Definition 2.4. Let $\mathbf{x} \in \mathbf{X}$ be a feasible decision vector. Then, $\mathbf{x}$ is *efficient* if and only if there does not exist an $\mathbf{x}' \in \mathbf{X}$ with $\mathbf{f}(\mathbf{x}')$ dominating $\mathbf{f}(\mathbf{x})$. Otherwise, $\mathbf{x}$ is said to be *inefficient*.

Definition 2.5. Let $\mathbf{z} \in \mathbf{Z}$ be an objective vector. Then, $\mathbf{z}$ is *weakly nondominated* if and only if there does not exist a $\mathbf{z}' \in \mathbf{Z}$ that strongly dominates $\mathbf{z}$.

Definition 2.6. Let $\mathbf{x} \in \mathbf{X}$ be a decision vector. Then, an $\mathbf{x}$ is *weakly efficient* if and only if there does not exist $\mathbf{x}' \in \mathbf{X}$ with $\mathbf{f}(\mathbf{x}')$ strongly dominating $\mathbf{f}(\mathbf{x})$.

The set that covers all nondominated solutions is called the *nondominated set*. It is a subset of the *weakly nondominated set*, which may also contain dominated solutions.

Definition 2.7. Let $\hat{\mathbf{z}} \in \mathbf{Z}$. If there exist solutions $\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_k} \in \mathbf{Z}$ and weights $w_1, w_2, \ldots, w_k \geq 0$ satisfying $\sum_{i=1}^{k} w_i = 1$ and $\sum_{i=1}^{k} w_i \mathbf{z_i} \geq \hat{\mathbf{z}}$, then $\hat{\mathbf{z}}$ is said to be *convex dominated*.

It should be noted that a convex dominated solution can be nondominated. These solutions are called *unsupported nondominated solutions* and they cannot be found by using weighted-sum approaches (i.e. by maximizing a positive linear combination of objectives). An illustration of nondominated, dominated, unsupported nondominated and weakly nondominated solutions are given in Figure 2.1.

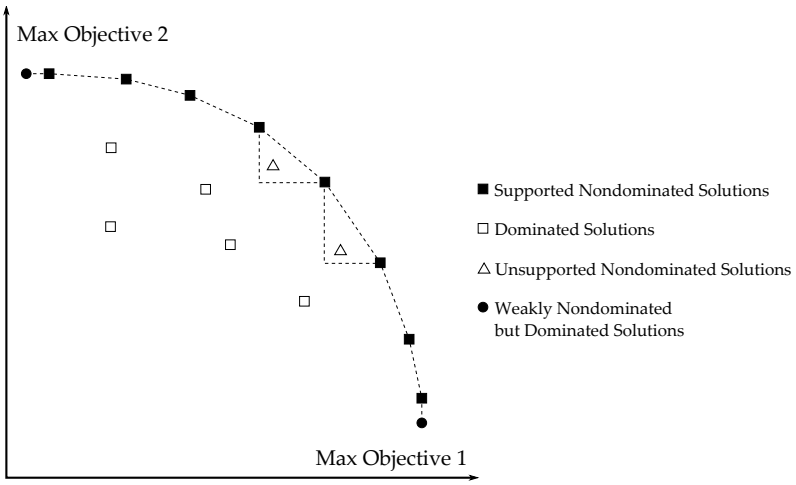


Figure 2.1: Classification of Solutions

2.1.2 Ideal and Nadir Objective Vectors

Definition 2.8. Let $\mathbf{z}^* = \left[z_1^*, z_2^*, \ldots, z_p^*\right]^T$ be a vector of objective values whose element i is the optimal value of the i^{th} objective function. Then, $\mathbf{z}^*$ is called the *ideal objective vector*.

Definition 2.9. Let $\mathbf{z^n} = \left[z_1^n, z_2^n, \ldots, z_p^n\right]^T$ be a vector of objective values whose element i is the worst value of i^{th} objective function among all nondominated solutions. Then, $\mathbf{z^n}$ is called the *nadir objective vector*.

Ideal and nadir vectors are illustrated in Figure 2.2. Note that neither the ideal nor the nadir objective vector needs to be feasible. The ideal objective vector is determined by individually optimizing each of the p objectives over the feasible region. However, finding the nadir objective vector is not straightforward, since the efficient set is not known explicitly. It may be estimated using a *payoff table*.

Consider a multiobjective optimization problem with p maximization-type objectives. A payoff table for this problem (see Table 2.1) is constructed by placing the objective vectors resulting from individually optimizing each of the p objectives over the feasible region. Then, the smallest value in the j^{th} column gives an estimate for the j^{th} element of the nadir objective vector. However, there is no guarantee for the quality of the estimate. If the row corresponding to the smallest value is weakly nondominated but dominated, then it may be an underestimate. Otherwise, it is the correct value or an overestimate. In either case, the deviation from the actual value can be large. Underestimation can be avoided by changing the objective function f_i with an augmented one as follows:

$$\text{Max } f_i + \epsilon \sum_{j \neq i}^{p} f_j \tag{2.3}$$

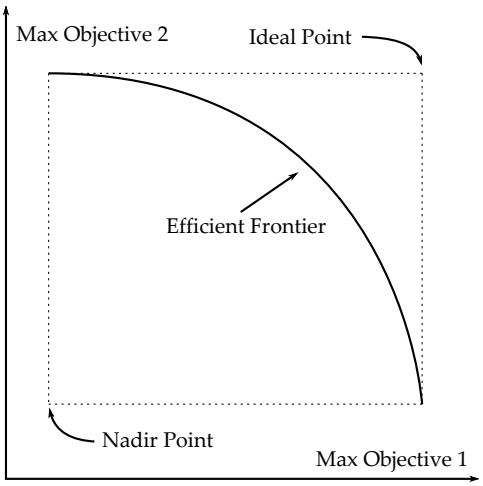


Figure 2.2: Ideal and Nadir Vectors

where ϵ is a very small number.

Table 2.1: Estimation of the Nadir Vector using a Payoff Table

Objectives	1	2	...	p
1	z_1^*	z_{12}	...	z_{1p}
2	z_{21}	z_2^*	...	z_{2p}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
p	z_{p1}	z_{p2}	...	z_p^*
Estimation	$\min_j\{z_{j1}\}$	$\min_j\{z_{j2}\}$	...	$\min_j\{z_{jp}\}$

2.1.3 Distance Metrics

A metric is a function that gives the distance between two vectors $\mathbf{x}, \mathbf{y} \in \Re^n$. The distance between $\mathbf{x}$ and $\mathbf{y}$ in terms of an L_q-metric can be computed as follows:

$$\left\|\mathbf{x}-\mathbf{y}\right\|_q = \left[\sum_{i=1}^{n} \left|x_i - y_i\right|^q\right]^{\frac{1}{q}} \tag{2.4}$$

The weighted L_q-distance is calculated using the following formula:

$$\left\|\mathbf{x}-\mathbf{y}\right\|_{w,q} = \left[\sum_{i=1}^{n} w_i \left|x_i - y_i\right|^q\right]^{\frac{1}{q}} \tag{2.5}$$

Commonly used weighted distance metrics are given in Table 2.2.

Table 2.2: Commonly used Weighted Distance Metrics

Name	Metric	Formula
Rectilinear	L_1	$\sum_{i=1}^{n} w_i \left\lvert x_i - y_i \right\rvert$
Euclidean	L_2	$\left[\sum_{i=1}^{n} w_i \left\lvert x_i - y_i \right\rvert^2\right]^{\frac{1}{2}}$
Tchebycheff	L_∞	$\max_i \left(w_i \left\lvert x_i - y_i \right\rvert\right)$

2.2 Literature Review

Solving some multiobjective optimization problems is a difficult and resource-demanding task. For this purpose, many different mathematical formulation-based methods ranging from the weighted-sums approach to goal programming and achievement scalarizing function-based methods are proposed. Although they are effective in finding good solutions, they often have significant computational requirements on some problems. On the other hand, multiobjective evolutionary algorithms (MOEAs) are suitable for solving these problems in a reasonable time, since they can generate multiple solutions that can approximate the entire efficient frontier in a single run. In addition, they are less sensitive to the shape of the efficient frontier than the traditional methods (Coello, 1999). These qualifications have made MOEAs popular among researchers in the past two decades.

The review papers by Fonseca and Fleming (1995), Tamaki et al. (1996), Coello (1999), Deb (1999) and Veldhuizen and Lamont (2000) examine different MOEA approaches and point out their increasing popularity among researchers. Also, the books by Deb (2001) and by Coello et al. (2006) cover many aspects of evolutionary algorithms in multiobjective optimization. In addition, Coello maintains a list of references to publications in the multiobjective evolutionary optimization area at `http://www.lania.mx/~ccoello/EMOO/EMOObib.html`.

Traditionally, MOEAs are considered to have two goals:

1. Converging to the efficient frontier
2. Maintaining a well-dispersed set of solutions to obtain a good approximation of the efficient frontier

Most of the MOEAs try to generate solutions that approximate the entire efficient frontier. However, as Coello (2000) indicates, this does not help decision making at all. Consequently, a third goal of converging to the regions that are appealing to the decision maker is also considered in the recent years. The review papers of Coello (2000), Cvetković and

Parmee (2002) and Rachmawati and Srinivasan (2006) stress the importance and examine the methods of converging to the interesting portions of the efficient frontier.

Since the algorithms in this study are capable of finding solutions both in the entire efficient frontier and in a particular region, we will divide our literature review into two. We will first review the evolutionary algorithms that approximate the entire efficient frontier. Following that, we will examine preference incorporation techniques.

2.2.1 MOEAs that Approximate the Entire Efficient Frontier

Knowles and Corne (2000) suggested an elitist multiobjective evolutionary algorithm, Pareto-Archived Evolution Strategy (PAES) that uses an evolution strategy. Unlike most of the MOEAs, the (1 + 1) evolution strategy proposed in their paper employs a local search using a population of size 1. However, while doing the local search, it makes use of an archive that consists of previously found solutions to identify the approximate dominance ranking of the current and candidate solution vectors. No recombination is applied to the solutions but they are mutated. For preserving diversity and calculating the density of solutions, the algorithm divides the entire search space into hypercubes. These hypercubes are used for choosing the next parent to be evolved.

The Pareto Envelope based Selection Algorithm 2 (PESA-II) proposed by Corne et al. (2001) defines the unit of selection as a hyperbox in the objective space. Instead of assigning a fitness to a solution, it assigns a fitness to the hyperbox itself. Diversity is maintained by the help of the hyperboxes. Although the algorithm allows multiple solutions to reside in one hyperbox, the number of solutions in a hyperbox directly affects the selection probability of a solution. This is done by assigning the selective fitness of a solution to the number of solutions that occupy the same hyperbox with that solution, and favoring smaller fitness values.

Deb et al. (2002) proposed the Nondominated Sorting Genetic Algorithm 2 (NSGA-II). This elitist algorithm uses the idea of nondominated sorting to classify the solutions in the population and assigning their ranks. At the end of each generation, the offspring population is combined with the parent population. The population of the next generation is then selected among the members of the combined population, accepting first those that have better ranks. When the next generation is unable to accept all solutions from a particular rank, a selection is made with respect to the crowding measure of the solutions having that rank.

Zitzler et al. (2002) suggested the Strength Pareto Evolutionary Algorithm 2 (SPEA2) that overcomes the deficiencies of its predecessor, Strength Pareto Evolutionary Algorithm (SPEA). The algorithm maintains two populations, namely the regular population of size N, and the archive with a maximum capacity of $\bar{N}$. At the end of each generation, the regular population is copied to the archive and dominated solutions are removed from the archive. If the size of the archive exceeds the maximum capacity, $\bar{N}$, then some members are removed using a clustering technique that preserves the characteristics of the archive. After that, each solution is assigned a *strength* that represents the number of solutions it dominates. Based on the strength and density values, the fitness of each solution is calculated.

The Indicator Based Evolutionary Algorithm (IBEA) proposed by Zitzler and Künzli (2004) aims to integrate the preference information into the selection procedure. The optimization goal of the algorithm is defined by a binary indicator that can be adapted to reflect the decision maker's preferences. It is also stated that the algorithm does not need any explicit diversity preservation operator. The fitness of a solution with respect to an indicator is a measure for the *loss in quality* if that solution is removed from the population. The authors suggest two different indicators in the paper, namely the hypervolume indicator and the additive ϵ-indicator.

Deb et al. (2005) suggested the steady-state Epsilon Multiobjective Evolutionary Algorithm (ϵ-MOEA). In ϵ-dominance, a solution is considered to be efficient if there is no solution that is at least an ϵ amount better than that solution in every objective and more than an ϵ amount better at least in one objective. The authors implement this by dividing the objective space into hyperboxes having side lengths of ϵ in all objectives. Diversity is maintained by disallowing more than one solution in each hyperbox. The algorithm maintains two populations, the archive and the regular population. In each generation, two offspring are created using one parent from each population. While the regular population is updated using regular domination, the archive population is updated using hyperboxes. The algorithm runs fast since it does not need any diversity preserving operator.

The S-Metric (Hypervolume metric) Selection Evolutionary Multiobjective Algorithm (SMS-EMOA) proposed by Beume et al. (2007) explicitly sets the optimization goal as the maximization of the dominated hypervolume. The authors state that the algorithm combines the nondominated sorting idea of NSGA-II with the archiving strategy presented by Knowles et al. In every generation, nondominated sorting is used for assigning ranks to the solutions, whereas the hypervolume metric is used for discarding the solution that contributes the least to the total dominated hypervolume. Since computation of hypervolume

metric is expensive, a single offspring is created in every generation.

Soylu and Köksalan (2006) suggested the Favorable Weights Evolutionary Algorithm (FWEA). The algorithm utilizes the favorable weights mechanism that allows each solution to choose its own favorable Tchebycheff weights for a Tchebycheff distance function. The direction defined by these weights is the best direction of that solution to contribute to convergence. On the other hand, the diversity among the solutions are maintained by incorporating a nearest neighbor information to the fitness of each solution. In addition, in each generation, nondominated sorting is applied to solutions in order to adjust the fitness values among different frontiers.

2.2.2 Incorporation of Preference Information to the MOEAs

Branke et al. (2001) proposed a method for utilizing the decision maker's preferences for guiding the search towards the regions of interest. The method asks the decision maker to specify his/her trade-offs between each pair of objectives. After that, it constructs the *minimal* and *maximal* utility functions. These utility functions are used for modifying the dominance scheme according to the decision maker's preferences. The authors also suggested the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) that uses this method. They showed that the algorithm is able to converge to the desired regions.

Phelps and Köksalan (2003) proposed an interactive multiobjective evolutionary algorithm for multiobjective combinatorial optimization problems. The algorithm interacts with the decision maker during the run to guide the search towards the preferred regions. At the initialization stage, the algorithm constructs a *club* for each objective, which keeps the high-achiever solutions in that objective. These clubs are then used for parent selection. The preference information is gathered by asking the decision maker to compare some solutions. Using these pairwise comparisons, weights of the estimated utility function can be calculated by the middlemost weights technique (see Köksalan et al. (1984)), which solves a linear program (LP). The fitness of solutions is computed using these weights. The algorithm also keeps an incumbent solution, which is selected to be the best solution by the decision maker so far. When an offspring challenges the incumbent, an interaction with the decision maker is initiated to update the incumbent and the estimated utility function.

Deb and Sundar (2006) suggested a method that seeks to find a set of solutions close to one or more reference points that are specified by the decision maker. They incorporate the method into NSGA-II by modifying its crowding distance operator. First, they calculate

each solution's normalized Euclidean distance to each reference point and assign the solutions having the minimum distances the rank of one. After that, every solution is assigned the minimum of all its ranks as its final rank. Solutions having smaller ranks are preferred. The method is shown to work with both feasible and infeasible reference points which need not to be nondominated.

Deb and Kumar (2007) adapt the reference direction based method described in Korhonen and Laakso (1986), which solves a sequence of achievement scalarizing functions. The method asks the decision maker to specify one or more reference directions. After that, a set of points $(\mathbf{r}(t), t = 0, 1, \ldots)$ are marked on the reference directions. The achievement scalarizing function value $\mathbf{s}(\mathbf{z}, \mathbf{r}, \mathbf{w})$ of each point $\mathbf{r}(t)$ corresponding to the chosen weight vector $\mathbf{w}$ and each population member $\mathbf{z}$ is calculated. The population members that have the smallest value of $\mathbf{s}$ for each $\mathbf{r}(t)$ is assigned to the first frontier. The members with the next smallest values are assigned to the second frontier and the procedure is repeated for the other members. The rest of the algorithm is the same as NSGA-II.

Köksalan and Phelps (2007) proposed the Evolutionary Metaheuristic for Approximating Preference-Nondominated Solutions (EMAPS) for the multiobjective combinatorial problems to find solutions that are appealing to the decision maker. The algorithm uses partial preference information provided by the decision maker, which is gathered through qualitative statements. That is, no precise utility weights or trade-off parameters are required. These statements are then used for restricting the weight space. The fitness of a solution is based on the relative strength of the solution over other solutions in the population. Calculation of fitness values requires the favorable weights of the solutions that are found by solving an LP that maximizes the fitness of the solution. This LP simultaneously provides both favorable weights and fitness values. The authors tested the algorithm on multiobjective knapsack and multiobjective spanning tree problems and reported good results.

CHAPTER 3

TERRITORY DEFINING EVOLUTIONARY ALGORITHM (TDEA)

In this chapter, we present the Territory Defining Evolutionary Algorithm (TDEA) for solving multiobjective optimization problems. TDEA is a steady-state algorithm that can be tailored to converge either to the entire efficient frontier or to a desired portion of the efficient frontier. This elitist algorithm promotes convergence by maintaining two populations: The archive population that consists only of nondominated solutions and the regular population that may contain both dominated and nondominated solutions. When updating the archive population, it defines a *territory* around the solution closest to the newcomer and rejects the newcomer if it violates the territory. The territory defining property of TDEA eliminates the need for an explicit diversity operator, resulting in a fast operation while always keeping a diverse set of solutions in the archive population. The idea is similar to that of ϵ-MOEA, however TDEA's territory defining property improves shortcomings of the hyperbox mechanism of ϵ-MOEA that causes losing solutions toward the ends of the Pareto-optimal frontier.

The sections of this chapter are organized as follows: We start by presenting the details of TDEA and its important properties in Section 3.1. We then present the results of the simulation runs and provide comparisons in Section 3.2.

3.1 The Details of TDEA

As mentioned before, TDEA maintains two populations. The first population is called *the regular population*, which we will denote as P from now on. It is formed by randomly created solutions at the beginning of the algorithm until its maximum size $\bar{N}$ is reached. It may contain both nondominated and dominated solutions, maintaining some lateral diversity.

An offspring is accepted to the regular population if it is nondominated relative to the solutions in the regular population.

The second population is *the archive* (A) that consists of only nondominated solutions. It is created from the copies of the nondominated solutions of the regular population. Although its size is not bounded by a fixed number, it depends on a parameter specified by the user. This parameter, τ, defines the extent of the territory of a solution. Only the solutions that are accepted to the regular population are eligible to be evaluated to enter the archive. The evaluation is done using the territory defining property of the algorithm, which is described in Section 3.1.3.

We will first give an outline of the algorithm before going into the details:

1. Ask the user to specify $\bar{N}$, τ and maximum number of iterations T. Set iteration count $t = 0$. Create $\bar{N}$ random solutions to initiate $P(0)$. Copy the nondominated solutions of $P(0)$ into $A(0)$.

2. Set $t \leftarrow t + 1$. Choose a parent from each population $P(t)$ and $A(t)$. Recombine parents to create a new offspring and apply mutation to it.

3. Test the offspring for acceptance into $P(t)$. If accepted, insert into $P(t)$ and go to the next step. Otherwise, go to Step 5.

4. Test the offspring for acceptance into $A(t)$. If accepted, insert into $A(t)$.

5. If the iteration limit is reached, that is, $t = T$, then stop and report the archive population. Otherwise go to Step 2.

3.1.1 Selection

TDEA uses one solution from each population for recombination. However, the selection scheme is different for the two populations. In the regular population, the binary tournament selection is used. Two solutions s_1 and s_2 are randomly picked from the regular population and the parent p_1 is determined according to the following procedure:

1. Test whether s_1 or s_2 dominates the other solution.

2. If s_1 dominates s_2, set $p_1 = s_1$ and vice versa.

3. If neither dominates the other, then select one of s_1 and s_2 as p_1 with equal probabilities.

Parent p_2 is selected from the archive population. Since all solutions in the archive are nondominated relative to each other, the binary tournament selection is not meaningful here. We randomly choose one solution as p_2, although some other techniques can be utilized.

3.1.2 Scaling

In multiobjective optimization problems, the range and the scale of objectives may vary greatly. These differences may cause MOEAs to have some bias for some of the objectives. To address this concern, many objective scaling techniques are proposed in the literature (see Steuer (1986) for details). In TDEA, we use the idea of Soylu and Köksalan (2006), who adapted the approach of Chang et al. (2003) for scaling the objectives. Their method treats the objective values between the ideal (f^*) and nadir point (f^n) differently from those beyond the nadir point. While they use linear scaling for the former, the latter are scaled using the following sigmoid function:

$$\Psi(y) = \left(\frac{1}{1 + e^{-\frac{y}{\lambda}}} - C \right) G \tag{3.1}$$

where C and G are parameters to control the shape, λ is a parameter for controlling the slope and y is the value to be scaled. We set $C = 0.5$ and $G = 2$ so that the sigmoid function takes values between 0 and 1. As done by the authors, we scale the values in the efficient range into a large portion of $[0, 1]$. The rest is scaled into the remaining narrow interval, since they are not as important as the nondominated range. For this purpose, we set λ in such a way that the sigmoid function has a very small slope at the nadir point (f^n). This is done by solving the following equation using a very small slope of the sigmoid function at the nadir point:

$$\left. \frac{d}{dy} \Psi(y) \right|_{y=f^n} = \frac{e^{-\frac{f^n}{\lambda}}}{\lambda \left(1 + e^{-\frac{f^n}{\lambda}}\right)^2} G = m \tag{3.2}$$

We present the steps of the scaling procedure below for a minimization type objective function. Note that maximization type objective functions have to be converted into minimization before scaling.

1. Shift the objective value and nadir point until the ideal point becomes 0, that is, set $\bar{f_i} = f_i - f_i^*$ and $\bar{f_i^n} = f_i^n - f_i^*$.

2. Using $\bar{f_i^n}$ and a very small m, solve (3.2) to determine λ.

3. The scaled objective $\hat{f}_i$ is then found as follows:

$$\hat{f}_i = \begin{cases} \dfrac{\Psi(\bar{f}_i^n)}{\bar{f}_i^n}\bar{f}_i & \text{if } f_i \leq f_i^n \\ \left(\dfrac{1}{1+e^{-\frac{\bar{f}_i}{\lambda}}} - C\right)G & \text{otherwise} \end{cases} \tag{3.3}$$

An illustration of scaling is given in Figure 3.1. In this study, we estimate the nadir points using payoff tables and determine λ values using a trial-and-error method.

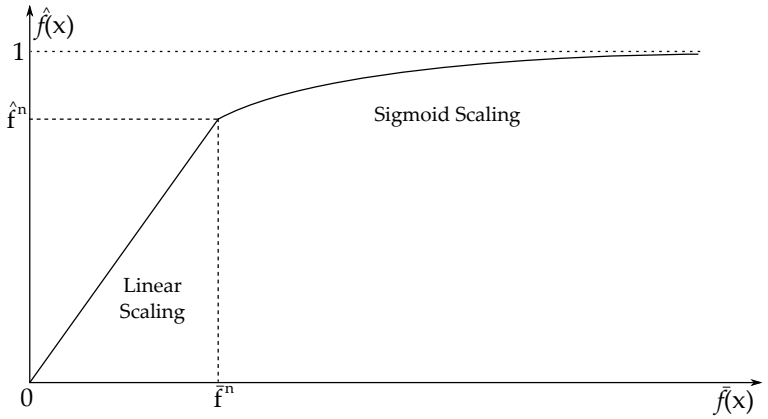


Figure 3.1: Scaling

Note that this scaling method scales all values into $[0, 1]$ interval. If only the efficient range is scaled $[0, 1]$ interval, the territory defining property may not work correctly for values beyond the nadir point, because the territory sizes are set according to the efficient range. But, the efficient range is more important, hence the larger portion of the $[0, 1]$ interval is allocated to those. If the nadir point is exactly known, then this portion can be accurately determined for each objective. Otherwise, they may be differences depending on the quality of the estimate. However, as long as the error of the estimate is not severe, the effect of these differences on the performance of the algorithm is small.

3.1.3 Population Updates

An offspring (denoted as c which stands for *child*) is first evaluated for acceptance into the regular population. The evaluation procedure is as follows:

1. Test c against each solution $s_i \in P(t)$ for dominance. Mark solutions dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to the next step.

2. Remove one of the marked solutions randomly from $P(t)$. If no solutions are marked, remove one randomly chosen solution from $P(t)$.

3. Insert c into $P(t)$ and test for acceptance in $A(t)$.

If c is rejected in the above procedure, then it is not evaluated for archive acceptance at all, since it is guaranteed that at least one solution in the archive dominates it. Otherwise, it is checked if it satisfies the requirements of the archive population. The archive evaluation process is more complicated. It consists of two stages. First, the offspring c is checked for dominance against the members of the archive. If it is not dominated by any solution in the archive, then we proceed to the second stage. The second stage begins with the removal of solutions dominated by c from the archive. After that, we determine the closest solution s_{i^*} to c in rectilinear distance. This distance is calculated using the scaled objective function values. Then, we check whether c is in the territory of s_{i^*}, which is defined as the region within τ distance in all objective values of s_{i^*}. If the maximum scaled objective distance between s_{i^*} and c is smaller than τ, then we say that the territory of s_{i^*} is violated by c in all directions. In that case, we reject c. Otherwise, the territory is said to be preserved in at least one direction and c is accepted.

The details of the procedure are given below:

1. Test c against each solution $s_i \in A(t)$ for dominance. Mark solutions dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to the next step.

2. Remove all marked solutions from $A(t)$.

3. If $A(t)$ is empty, accept c and insert it into $A(t)$. Otherwise, go to next step.

4. Calculate the rectilinear distance $d_{ci} = \sum_{j=1}^{p} \left| \hat{f}_{cj} - \hat{f}_{ij} \right|$ of c to each solution $s_i \in A(t)$ using the scaled objective function values.

5. Find $i^* = \text{argmin}_i(d_{ci})$, that is, the solution s_{i^*} closest to c.

6. Find the maximum scaled absolute objective difference between c and s_{i^*}. That is, find

$$\delta = \max_{j=1,2,\ldots,p} \left| \hat{f}_{cj} - \hat{f}_{i^*j} \right| \tag{3.4}$$

   where $\hat{f}_{cj}$ and $\hat{f}_{i^*j}$ are the scaled j^{th} objective values of offspring c and solution s_{i^*}, respectively.

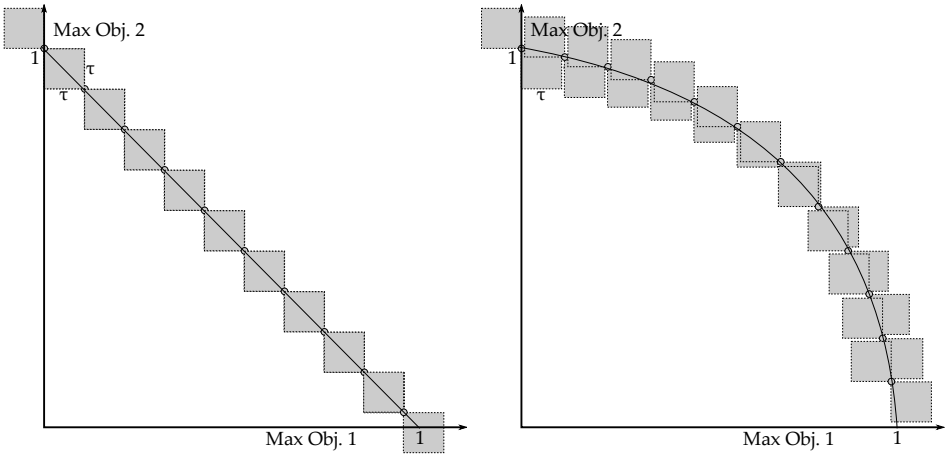
7. Accept c if $\delta \geq \tau$ and insert it into $A(t)$. Otherwise, reject c.

3.1.4 Fitness Function

TDEA does not have a fitness function associated with the solutions. Instead, each solution is tested for dominance with respect to the members of the population that they are evaluated for entering. However, it should be noted that TDEA does not disallow the use of a fitness function. A fitness function can be used for various purposes such as parent selection and guided search. For example, instead of binary tournament selection, one can compare the fitnesses of two solutions for choosing the parent.

3.1.5 Determination of τ

In TDEA, τ controls the territory size of solutions and the hypervolume that a solution occupies in the objective space. Since the total nondominated hypervolume is limited, the territory size affects the maximum number of solutions in the archive population. In a given problem, a smaller τ leads to a larger population compared to a larger τ. However, the size of the nondominated hypervolume depends on the shape of the Pareto-optimal frontier. Hence, a τ value used in a problem may work differently in another problem. We illustrate this in Figure 3.2.



(a) Linear (b) Curve

Figure 3.2: Ideal Placement of Solutions on the Pareto-optimal Frontiers

As can be seen in Figure 3.2, the same τ leads to a smaller population in the linear frontier. For that reason, there is no straightforward way to set τ. In this study, we determine τ by performing some preliminary runs.

3.1.6 Computational Complexity

TDEA is a steady-state multiobjective evolutionary algorithm. In each generation, only one offspring is created and evaluated to enter the populations. Assuming that an offspring c is accepted into the regular population P, it takes $p\bar{N}$ calculations to check whether c is nondominated with respect to all solutions in P. Denoting the size of archive population A as $\hat{N}$, the number of calculations needed to check whether c is nondominated with respect to all solutions in A is $p\hat{N}$. Note that since the size of A is not fixed, the number of calculations for dominance check in A is not known exactly. In addition to that, it takes $p\hat{N}$ calculations to determine the closest solution to c in A. At the end, $p\bar{N} + 2p\hat{N}$ calculations are made in each generation, excluding the ones made for crossover and mutation operators. If $\hat{N} \gg \bar{N}$, then the computational complexity of TDEA is dominated by the size of the archive population. In this case, the worst case computational complexity is $O\left(p\hat{N}\right)$.

3.2 Simulation Runs and Comparisons

In this section, we evaluate TDEA on some test problems and compare its performance against well-known MOEAs, namely SPEA2, NSGA-II, PAES, PESA-II, IBEA and ϵ-MOEA. We use test problems ZDT (Deb, 2001) and DTLZ (Deb et al., 2001), each of which has different characteristics as shown in Table 3.1.

We use the real-valued representation of the decision variables. That is, each gene in a chromosome is the value of the decision variable itself. As the crossover operator, we choose the Simulated Binary Crossover (SBX) (Deb and Agrawal, 1995) with *Prob*(crossover) = 1.0 and crossover parameter $\eta_c = 20$ for all metaheuristics. The mutation operator is the Polynomial Mutation Operator (Deb and Goyal, 1996) with *Prob*(mutation) = $\frac{1}{\text{number of variables}}$ and mutation parameter $\eta_m = 20$.

We run each algorithm 50 times with different seeds for the random number generator and limit each run by a prespecified maximum number of function evaluations. Initial populations are formed from randomly created solutions and population sizes are set based on the number of objectives. For metaheuristics without constant population sizes (ϵ-MOEA and TDEA), we make preliminary runs to ensure that the final nondominated population is approximately the same size as in the other metaheuristics. The detailed parameter settings are given in Table 3.2.

We implement our algorithm in C++ programming language and use ϵ-MOEA code

Table 3.1: Test Problems and Their Characteristics

Test Problem	Number of Objectives	Properties	Decision Variables
ZDT1	2	Convex, Continuous Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,30$
ZDT2	2	Non-Convex, Continuous Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,30$
ZDT3	2	Convex, Disconnected Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,30$
ZDT4	2	Convex, Multimodal Efficient Frontier	$x_1 \in [0,1]$ and $x_i \in [-5,5] \quad i = 2,\ldots,10$
ZDT6	2	Non-Convex, Non-Uniform Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,10$
DTLZ1	3	Linear Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,7$
	5		$x_i \in [0,1] \quad i = 1,2,\ldots,9$
DTLZ2	3	Spherical Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,12$
	5		$x_i \in [0,1] \quad i = 1,2,\ldots,14$
DTLZ3	3	Spherical, Multimodal Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,12$
DTLZ4	3	Spherical, Non-Uniform Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,12$
DTLZ5	3	Curve-Shaped Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,12$
DTLZ6	3	Multimodal, Curve-Shaped Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,12$
DTLZ7	3	Disconnected Efficient Frontier	$x_i \in [0,1] \quad i = 1,2,\ldots,22$

Table 3.2: Test Parameters

Number of Objectives	Population Size	Function Evaluations	Replications
2	100	40000	50
3	200	160000	50
5	400	320000	50

downloaded from the website of Kanpur Genetic Algorithms Laboratory[1]. Both programs are built with GNU C/C++ Compiler 4.2.3. For other metaheuristics, we use jMetal framework[2] with Sun Java JDK/JRE 1.6.0.3. All computational tests are made on a Pentium IV 2.8 GHz, 1 GB RAM computer running Kubuntu Linux 8.04.

3.2.1 Performance Metrics

In order to compare the performances of the algorithms, we use two performance metrics. The first one is the *Hypervolume metric* (Zitzler and Thiele, 1998), which measures the total objective space dominated by the final population P with respect to a given reference point W. This quality indicator improves when the resulting population has better convergence to the Pareto-optimal frontier and its distribution among the frontier is good. Formally, it can be described as follows:

$$Hypervolume = \bigcup_{i=1}^{|P|} V_i^W \tag{3.5}$$

where V_i is the objective space dominated by solution $i \in P$ with respect to the reference point W. The Hypervolume measure depends on the selected reference point. A larger value is desirable.

The other performance metric is the *Inverted Generational Distance* (Bosman and Thierens, 2003) that measures the algorithm's performance in both diversity and convergence. This metric calculates the Euclidean distance of each solution in the true nondominated frontier (PF_{true}) to the closest solution in the final population P and reports the average of the values. Formally, it can be stated as follows:

$$IGD = \frac{1}{|PF_{true}|} \sum_{i \in PF_{true}} \left(\min_{j \in P} \left\| z_i - z_j \right\|_2 \right) \tag{3.6}$$

For practical reasons, this metric is calculated using a set of true nondominated solutions that represent the nondominated front. In this study, we generate approximately $100,000$ well-dispersed true nondominated solutions for each problem.

[1] `http://www.iitk.ac.in/kangal/codes.shtml`

[2] `http://mallba10.lcc.uma.es/wiki/index.php/JMetal`

For every problem, we calculate the performance of each run of each algorithm in terms of Hypervolume (H) and Inverted Generational Distance (D). We use the nadir point as the reference point when calculating the Hypervolume values. Then, we compute their sample means $\bar{x}_H, \bar{x}_D$ and sample standard deviations s_H, s_D of these metrics. By the Central Limit Theorem, we assume that the sample means are normally distributed. Then, we test the following hypothesis at a 99% significance level to check whether there exists a statistically significant difference between the performances of TDEA (denoted as T) and other algorithms (denoted as C) in both metrics:

$$H_0 : \mu_{pm}^T = \mu_{pm}^C \tag{3.7}$$

$$H_1 : \mu_{pm}^T \neq \mu_{pm}^C \tag{3.8}$$

where pm stands for performance metric. For every problem, we report the estimated difference $\Delta_{pm} = \bar{x}_{pm}^T - \bar{x}_{pm}^C$ between means of TDEA and its contender for both metrics. We also present the p-value of the statistical test. For comparison, the metric results of the true nondominated sets are given.

3.2.2 2-Objective Problems

All 2-objective problems in this section the following form:

$$\text{Minimize } f_1(\mathbf{x}) \tag{3.9}$$

$$\text{Minimize } f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})) \tag{3.10}$$

ZDT1

The easiest of all 2-objective problems we test, ZDT1, has a convex Pareto-optimal frontier (Figure 3.4) and is defined as follows:

$$f_1(\mathbf{x}) = x_1 \tag{3.11}$$

$$g(\mathbf{x}) = 1 + \tfrac{9}{n-1} \sum_{i=2}^{n} x_i \tag{3.12}$$

$$h(f_1, g) = 1 - \sqrt{\tfrac{f_1}{g}} \tag{3.13}$$

where $n = 30$, $x_i \in [0, 1]$ for $i = 1, 2, \ldots, 30$. The Pareto-optimal frontier corresponds to $x_1^* \in [0, 1]$ and $x_i^* = 0$ for $i = 2, 3, \ldots, 30$. Table 3.3 shows the indicator values of different algorithms. All methods but IBEA successfully converge the Pareto-optimal frontier. As it can be seen from Table 3.4, TDEA is statistically better than all algorithms at a 99%

significance level in both metrics except SPEA2 in Inverted Generational Distance, where there is no significant difference. In addition, TDEA requires the second smallest time to complete $40,000$ function evaluations.

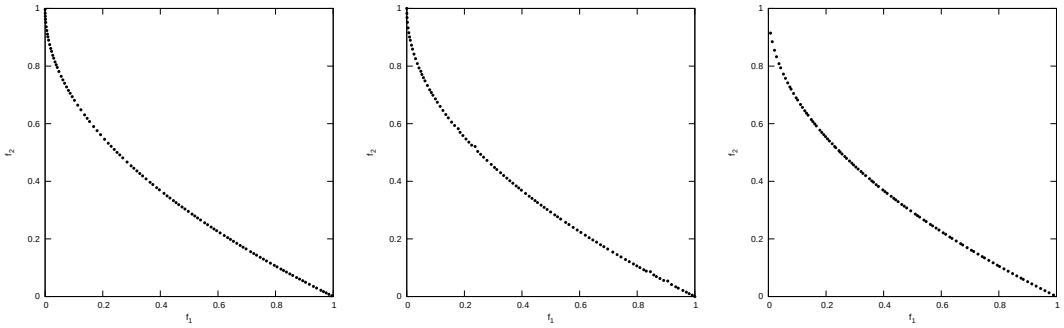
Table 3.3: Indicator Results for ZDT1

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.66090	0.000116	0.0000144	0.00000035	1.09
ϵ-MOEA	0.66067	0.000059	0.0000172	0.00000031	0.75
IBEA	0.45197	0.044280	0.0004920	0.00011600	2.48
NSGA2	0.65966	0.000232	0.0000183	0.00000104	5.68
PAES	0.65567	0.005750	0.0000370	0.00002780	1.68
PESA2	0.65685	0.001130	0.0000391	0.00002040	14.50
SPEA2	0.66084	0.000101	0.0000145	0.00000019	26.24
True Pareto	0.66599	-	0	-	-

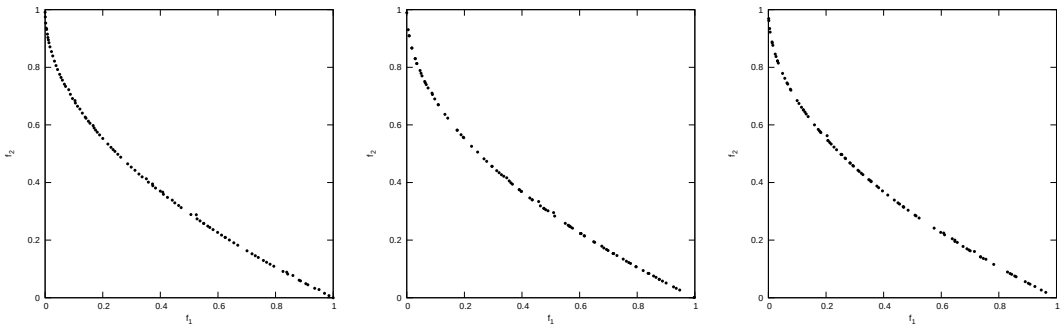
Table 3.4: Test Results for ZDT1

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.000233	0	TDEA	-0.000003	0	TDEA
IBEA	0.208929	0	TDEA	-0.000477	0	TDEA
NSGA2	0.001245	0	TDEA	-0.000004	0	TDEA
PAES	0.005236	0	TDEA	-0.000023	0	TDEA
PESA2	0.004047	0	TDEA	-0.000025	0	TDEA
SPEA2	0.000060	0.007	TDEA	0.000000	0.068	none

We present sample outputs from six algorithms in Figure 3.3. It can be seen that the final populations of TDEA and SPEA2 have better diversity than those of others. ϵ-MOEA loses solutions for the smaller values of the first objective. Although the others have good diversity in the overall, some gaps stand out in various places.



(a) TDEA (b) SPEA2 (c) ϵ-MOEA



(d) NSGA2 (e) PAES (f) PESA2

Figure 3.3: ZDT1 Plots

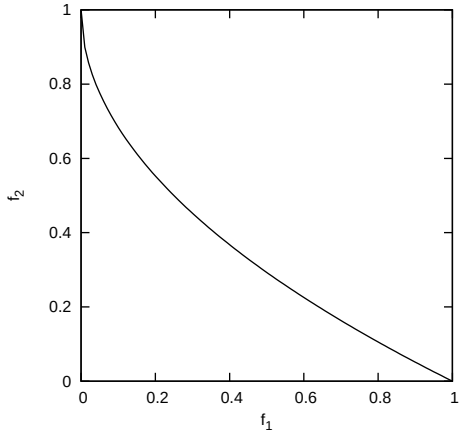


Figure 3.4: Pareto-Optimal Frontiers of ZDT1 and ZDT4

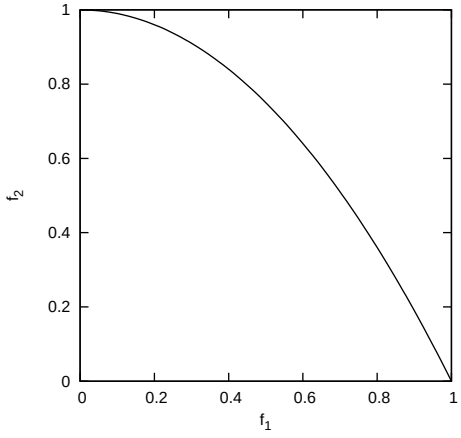


Figure 3.5: Pareto-Optimal Frontier of ZDT2

ZDT2

ZDT2 is an $n = 30$-variable problem with a nonconvex and uniform Pareto-optimal frontier (Figure 3.5). Its definition is as follows:

$$f_1(\mathbf{x}) = x_1 \tag{3.14}$$

$$g(\mathbf{x}) = 1 + \tfrac{9}{n-1} \sum_{i=2}^{n} x_i \tag{3.15}$$

$$h(f_1, g) = 1 - \left(\tfrac{f_1}{g}\right)^2 \tag{3.16}$$

where $x_i \in [0, 1]$ for $i = 1, 2, \ldots, 30$. The Pareto-optimal frontier is obtained when $x_1^* \in [0, 1]$ and $x_i^* = 0$ for $i = 2, 3, \ldots, 30$.

We observe similar results in indicator values (Table 3.5) as in ZDT1. IBEA again fails to converge while all others successfully find good approximations of the Pareto-optimal frontier. Table 3.6 shows that TDEA and SPEA2 outperform other algorithms, except they are at the same level as ϵ-MOEA with respect to the Hypervolume metric.

Table 3.5: Indicator Results for ZDT2

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.32833	0.000129	0.0000143	0.00000030	1.06
ϵ-MOEA	0.32828	0.000074	0.0000209	0.00000013	0.71
IBEA	0.07211	0.067230	0.0014630	0.00094300	2.50
NSGA2	0.32731	0.000207	0.0000187	0.00000087	5.70
PAES	0.32477	0.000836	0.0000301	0.00001290	1.37
PESA2	0.32476	0.000563	0.0000318	0.00000587	13.85
SPEA2	0.32829	0.000107	0.0000145	0.00000019	24.09
True Pareto	0.33332	-	0	-	-

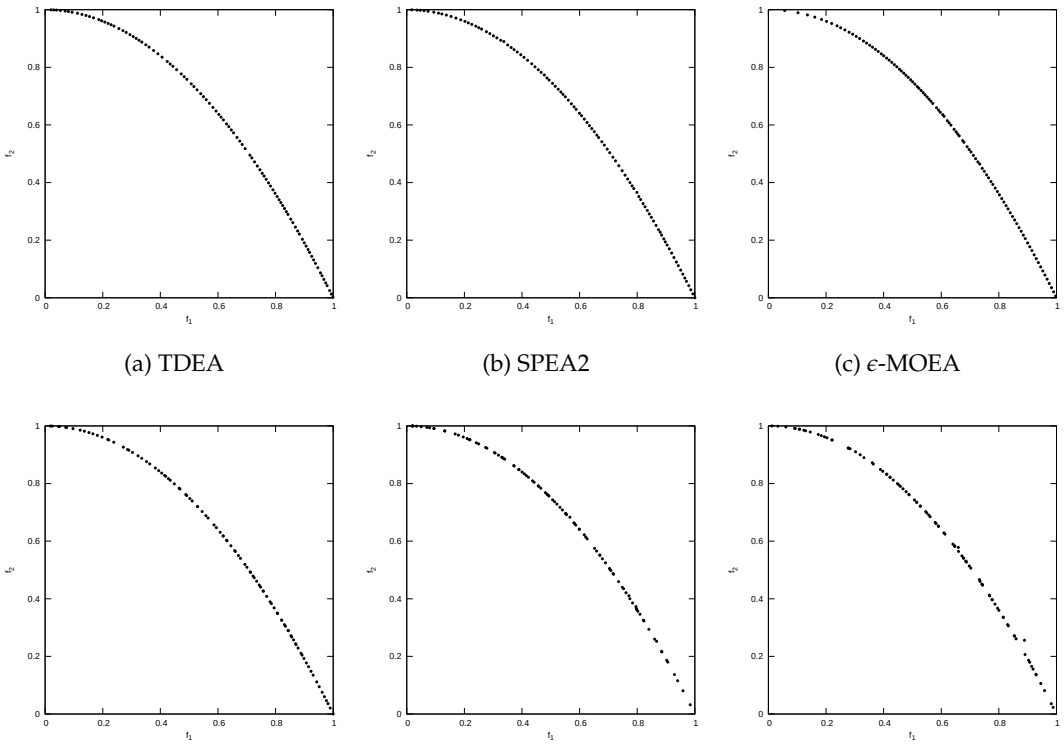
Figure 3.5 shows a plot for each of the six algorithms. The final populations of all algorithms display similar patterns to those of ZDT1.

ZDT3

ZDT3 is an $n = 30$-variable problem whose Pareto-optimal frontier consists of 5 discontinuous parts (Figure 3.9). It tests the ability of an MOEA to distribute the population to all

Table 3.6: Test Results for ZDT2

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.000052	0.016	none	-0.000007	0	TDEA
IBEA	0.256224	0	TDEA	-0.001449	0	TDEA
NSGA2	0.001021	0	TDEA	-0.000004	0	TDEA
PAES	0.003564	0	TDEA	-0.000016	0	TDEA
PESA2	0.003574	0	TDEA	-0.000018	0	TDEA
SPEA2	0.000045	0.063	none	0.000000	0	none



(a) TDEA (b) SPEA2 (c) ϵ-MOEA

(d) NSGA2 (e) PAES (f) PESA2

Figure 3.6: ZDT2 Plots

discontinuous regions. The problem is defined below:

$$f_1(\mathbf{x}) = x_1 \tag{3.17}$$

$$g(\mathbf{x}) = 1 + \tfrac{9}{n-1} \sum_{i=2}^{n} x_i \tag{3.18}$$

$$h(f_1, g) = 1 - \sqrt{\tfrac{f_1}{g}} - \tfrac{f_1}{g} \sin(10\pi f_1) \tag{3.19}$$

where $x_i \in [0, 1]$ for $i = 1, 2, \ldots, 30$. The Pareto-optimal frontier is obtained when $x_i^* = 0$ for $i = 2, 3, \ldots, 30$. Some solutions in the range $x_1 \in [0, 1]$ are dominated.

Indicator values (Table 3.7) show that performances of the algorithms are similar to those of ZDT1 and ZDT2. It can be observed from Table 3.8 that although TDEA is outperformed by SPEA2, ϵ-MOEA and NSGA2 in the Hypervolume metric, their differences in Inverted Generational Distance are not statistically significant. TDEA performs well with respect to the Hypervolume metric in the absolute sense, compared to results of the true Pareto-optimal set.

Table 3.7: Indicator Results for ZDT3

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.51391	0.000903	0.0000247	0.00005860	1.08
ϵ-MOEA	0.51487	0.000711	0.0000208	0.00004780	0.76
IBEA	0.01610	0.009410	0.0017500	0.00009360	2.48
NSGA2	0.51482	0.000083	0.0000128	0.00000061	5.67
PAES	0.49411	0.029830	0.0002460	0.00027200	1.33
PESA2	0.50906	0.012680	0.0000760	0.00015500	13.16
SPEA2	0.51497	0.000073	0.0000114	0.00000040	25.07
True Pareto	0.51695	-	0	-	-

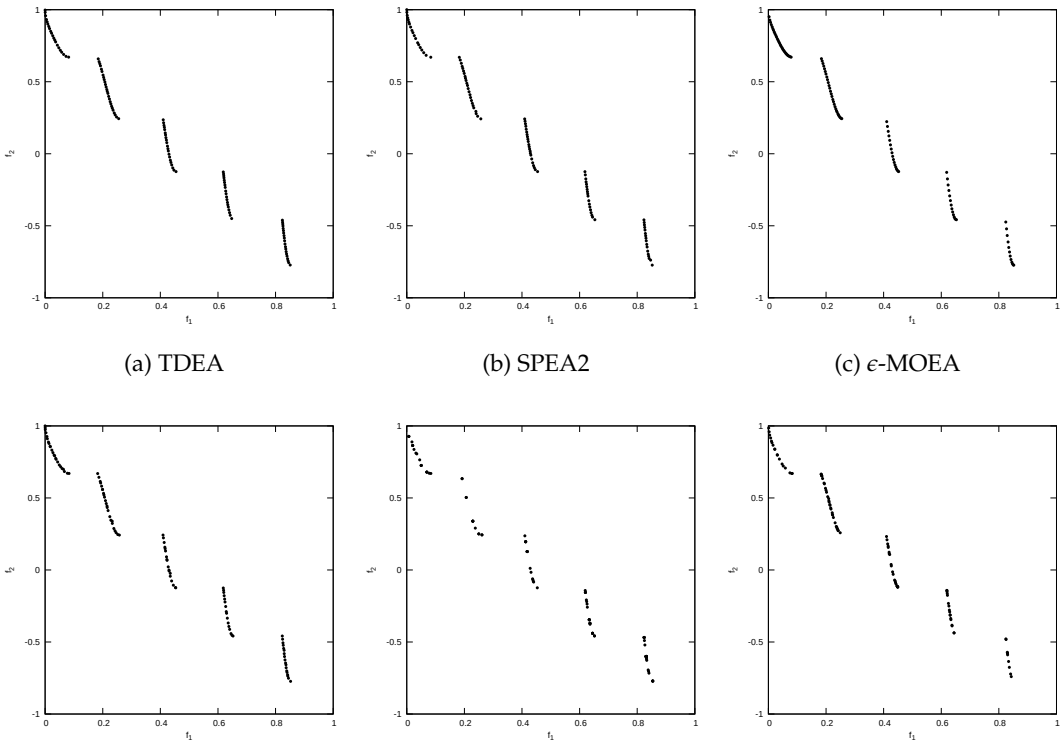
We present plots for TDEA, SPEA2, ϵ-MOEA, NSGA2, PAES and PESA2 in Figure 3.7. Plots clearly show TDEA's superior diversity preservation. SPEA2 also performs well. The final populations of other algorithms have varying densities at the different parts of the frontier.

ZDT4

This $n = 10$-variable problem has the same Pareto-optimal frontier as ZDT1 (Figure 3.4). However, there are 100 local optimal frontiers unlike ZDT1. Each of these local optimal

Table 3.8: Test Results for ZDT3

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	-0.000957	0	ϵ-MOEA	0.000004	0.716	none
IBEA	0.497812	0	TDEA	-0.001725	0	TDEA
NSGA2	-0.000911	0	NSGA2	0.000012	0.157	none
PAES	0.019795	0	TDEA	-0.000221	0	TDEA
PESA2	0.004851	0.01	TDEA	-0.000051	0.033	none
SPEA2	-0.001064	0	SPEA2	0.000013	0.115	none



(a) TDEA (b) SPEA2 (c) ϵ-MOEA

(d) NSGA2 (e) PAES (f) PESA2

Figure 3.7: ZDT3 Plots

frontiers can trap an algorithm and cause it to get stuck. This is one of the most difficult problems in the ZDT series. Below are the function definitions:

$$f_1(\mathbf{x}) = x_1 \tag{3.20}$$

$$g(\mathbf{x}) = 1 + 10(n-1) + \sum_{i=2}^{n} \left(x_i^2 - 10\cos(4\pi x_i)\right) \tag{3.21}$$

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \tag{3.22}$$

where $x_1 \in [0, 1]$ and $x_i \in [-5, 5]$ for $i = 2, \ldots, 10$. The Pareto-optimal frontier is obtained when $x_1^* \in [0, 1]$ and $x_i^* = 0$ for $i = 2, 3, \ldots, 10$.

Table 3.9 shows that IBEA is unable to find any solutions in the nondominated range while others find good approximation sets. We present the test results in Table 3.10. TDEA and SPEA2 outperform other metaheuristics and ϵ-MOEA follows them in performance. Figure 3.8 displays the qualities of the distributions in the final populations of these six algorithms.

Table 3.9: Indicator Results for ZDT4

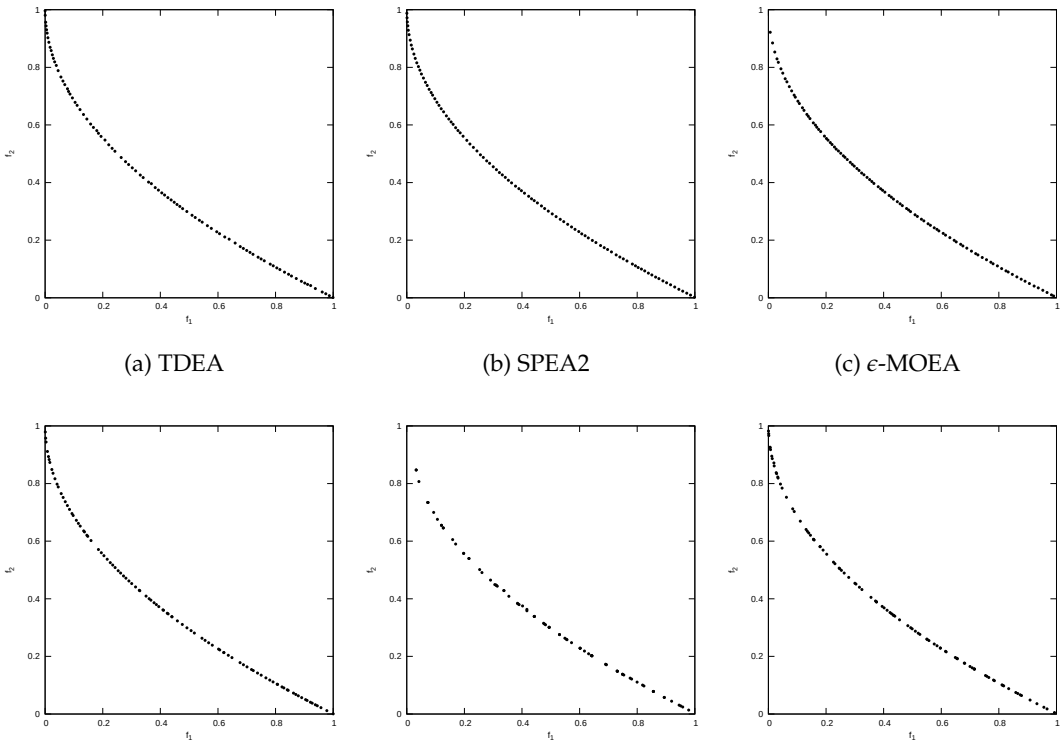
	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.65907	0.001090	0.0000160	0.00000127	0.66
ϵ-MOEA	0.65895	0.001320	0.0000182	0.00000353	0.34
IBEA	0.00000	0.000000	0.0077200	0.01107000	2.41
NSGA2	0.65822	0.001320	0.0000184	0.00000141	5.26
PAES	0.64861	0.007400	0.0000537	0.00004090	0.96
PESA2	0.65573	0.001400	0.0000320	0.00000681	10.89
SPEA2	0.65919	0.001550	0.0000153	0.00000176	17.17
True Pareto	0.66599	-	0	-	-

ZDT6

ZDT6 is an $n = 10$-variable problem with a nonconvex Pareto-optimal frontier (Figure 3.10). It features a non-uniform density among the frontier. In addition, the solutions are less dense as they get closer to the Pareto-optimal frontier. Both the nonconvexity and the varying density of the solutions cause difficulty in convergance and maintaining a diverse

Table 3.10: Test Results for ZDT4

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.000118	0.625	none	-0.000002	0	TDEA
IBEA	N/A	N/A	TDEA	N/A	N/A	TDEA
NSGA2	0.000845	0.001	TDEA	-0.000002	0	TDEA
PAES	0.010461	0	TDEA	-0.000038	0	TDEA
PESA2	0.003338	0	TDEA	-0.000016	0	TDEA
SPEA2	-0.000118	0.661	none	0.000001	0.035	none



(a) TDEA (b) SPEA2 (c) ϵ-MOEA

(d) NSGA2 (e) PAES (f) PESA2

Figure 3.8: ZDT4 Plots

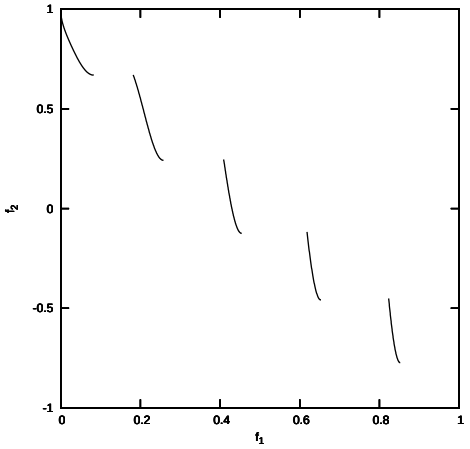


Figure 3.9: Pareto-Optimal Frontier of ZDT3

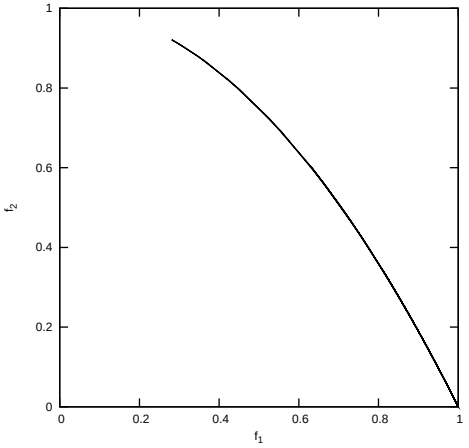


Figure 3.10: Pareto-Optimal Frontier of ZDT6

population. The problem is defined as follows:

$$f_1(\mathbf{x}) = 1 - e^{-4x_1} \sin^6(6\pi x_1) \tag{3.23}$$

$$g(\mathbf{x}) = 1 + 9\left[\frac{\sum_{i=2}^{10} x_i}{9}\right]^{\frac{1}{4}} \tag{3.24}$$

$$h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2 \tag{3.25}$$

where $x_i \in [0, 1]$ for $i = 1, \ldots, 10$. The Pareto-optimal frontier is obtained when $x_1^* \in [0, 1]$ and $x_i^* = 0$ for $i = 2, 3, \ldots, 10$.

Table 3.11: Indicator Results for ZDT6

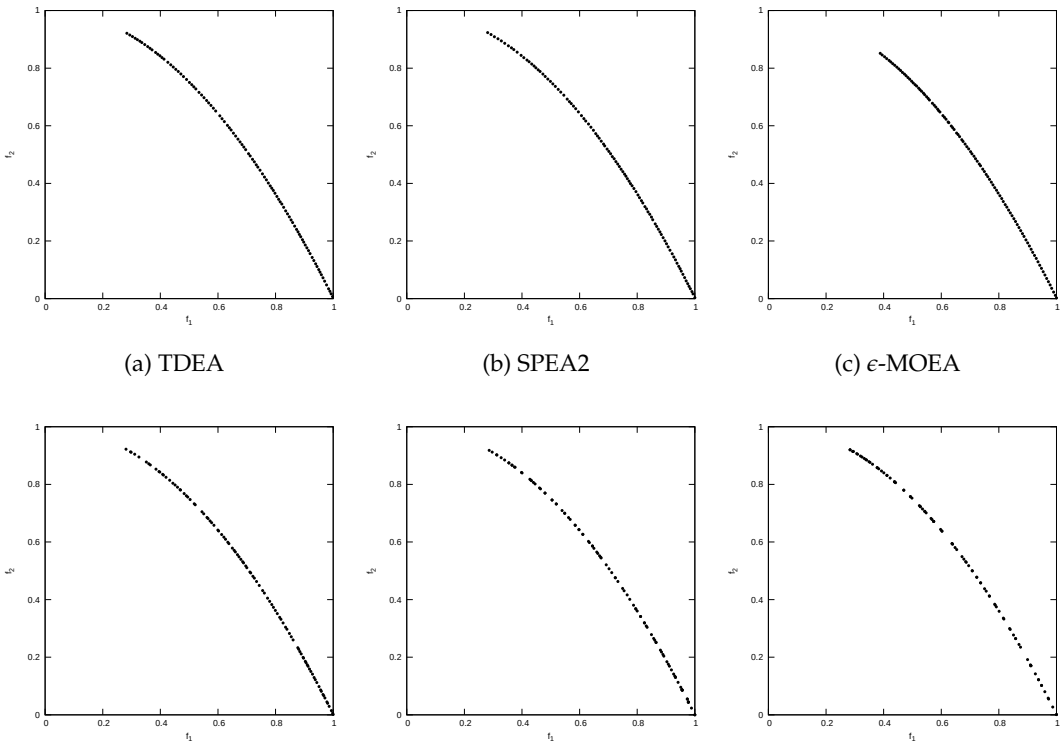
	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.39945	0.000226	0.0000154	0.00000248	0.78
ϵ-MOEA	0.39430	0.000142	0.0000619	0.00000027	0.38
IBEA	0.32898	0.022020	0.0002250	0.00007400	2.29
NSGA2	0.39907	0.000306	0.0000146	0.00000167	4.97
PAES	0.39245	0.015260	0.0000380	0.00004290	1.25
PESA2	0.39583	0.000765	0.0000335	0.00001390	12.29
SPEA2	0.39822	0.000573	0.0000138	0.00000109	18.94
True Pareto	0.40635	-	0	-	-

Table 3.11 indicates that all metaheuristics are able to converge to the Pareto-optimal frontier with good population diversity. IBEA performs better compared to the previous

Table 3.12: Test Results for ZDT6

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.005149	0	TDEA	-0.000046	0	TDEA
IBEA	0.070473	0	TDEA	-0.000210	0	TDEA
NSGA2	0.000379	0	TDEA	0.000001	0.065	none
PAES	0.007001	0.002	TDEA	-0.000023	0.001	TDEA
PESA2	0.003617	0	TDEA	-0.000018	0	TDEA
SPEA2	0.001234	0	TDEA	0.000002	0	SPEA2

problems. In statistical tests (Table 3.12), we see that TDEA outperforms all algorithms in the Hypervolume metric, whereas SPEA2 has the best Inverted Generational Distance score. We again observe in Figure 3.11 that ϵ-MOEA loses solutions towards the smaller values of objective 1. SPEA2 and TDEA have the best distribution among all.



(a) TDEA (b) SPEA2 (c) ϵ-MOEA

(d) NSGA2 (e) PAES (f) PESA2

Figure 3.11: ZDT6 Plots

3.2.3 3-Objective Problems

The DTLZ series can be modified to have an arbitrary number of objectives and decision variables. In all 3-objective DTLZ problems, we follow the recommendations of Deb et al. (2001) for setting the number of decision variables.

DTLZ1

DTLZ1 is the simplest of all 3-objective problems we test, with a linear Pareto-optimal frontier. It is defined as follows:

Minimize

$$f_1 = \tfrac{1}{2} x_1 x_2 \left(1 + g\left(x_M\right)\right) \tag{3.26}$$

$$f_2 = \tfrac{1}{2} x_1 (1 - x_2) \left(1 + g\left(x_M\right)\right) \tag{3.27}$$

$$f_3 = \tfrac{1}{2} (1 - x_1) \left(1 + g\left(x_M\right)\right) \tag{3.28}$$

where

$$g(x_M) = 100 \left[|x_M| + \sum_{x_i \in x_M} \left(x_i - 0.5\right)^2 - \cos\left(20\pi \left(x_i - 0.5\right)\right) \right] \tag{3.29}$$

$$x_i \in [0,1] \quad i = 1, 2, \ldots, 7 \text{ and } x_M = \{x_3, x_4, \ldots, x_7\}$$

The Pareto-optimal frontier is obtained when $x_M^* = 0$ and objective functions satisfy $\sum_{i=1}^{3} f_i = 0.5$. Performance metric scores and test results are given in Table 3.13 and Table 3.14, respectively. Test results show that TDEA is better than other metaheuristics in both metrics, except that it is at par with SPEA2 in the Hypervolume metric. NSGA2 converges better than ϵ-MOEA, but ϵ-MOEA has better diversity.

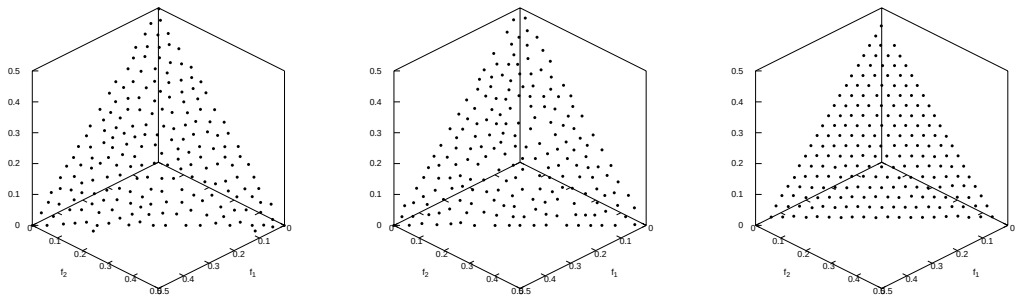
Table 3.13: Indicator Results for DTLZ1

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.80227	0.000690	0.0000913	0.00000123	4.46
ϵ-MOEA	0.75686	0.004030	0.0001010	0.00000610	2.37
IBEA	0.44669	0.063320	0.0008020	0.00033100	22.09
NSGA2	0.78431	0.003010	0.0001300	0.00000647	47.49
PAES	0.61510	0.170300	0.0006150	0.00058000	11.72
PESA2	0.29380	0.090900	0.0014020	0.00028500	201.70
SPEA2	0.80240	0.000455	0.0000947	0.00000134	353.18
True Pareto	0.83050	-	0	-	-

Table 3.14: Test Results for DTLZ1

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.045408	0	TDEA	-0.000010	0	TDEA
IBEA	0.355586	0	TDEA	-0.000711	0	TDEA
NSGA2	0.017960	0	TDEA	-0.000039	0	TDEA
PAES	0.187167	0	TDEA	-0.000524	0	TDEA
PESA2	0.508445	0	TDEA	-0.001310	0	TDEA
SPEA2	-0.000128	0.277	none	-0.000003	0	TDEA

Figure 3.12 shows the plots of the solutions found by TDEA, SPEA2 and ϵ-MOEA. Although ϵ-MOEA produces finely spaced solutions around the center of the objective space, it loses solutions towards the extremes, which causes it to have low scores. On the other hand, TDEA and SPEA2 maintain diversity throughout the entire frontier.



(a) TDEA (b) SPEA2 (c) ϵ-MOEA

Figure 3.12: DTLZ1 Plots

DTLZ2

DTLZ2 has a spherical Pareto-optimal frontier (Figure 3.14). Objective functions satisfy $\sum_{i=1}^{3} f_i^2 = 1$ at Pareto-optimality.

Minimize

$$f_1 = (1 + g(x_M)) \cos(0.5x_1\pi) \cos(0.5x_2\pi) \tag{3.30}$$

$$f_2 = (1 + g(x_M)) \cos(0.5x_1\pi) \sin(0.5x_2\pi) \tag{3.31}$$

$$f_3 = (1 + g(x_M)) \sin(0.5x_1\pi) \tag{3.32}$$

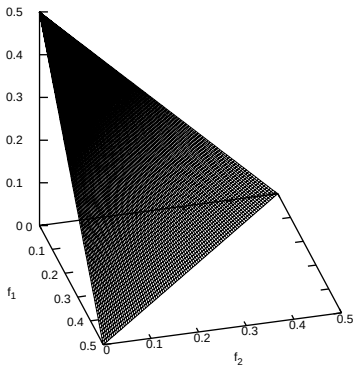


Figure 3.13: Pareto-Optimal Frontier of DTLZ1

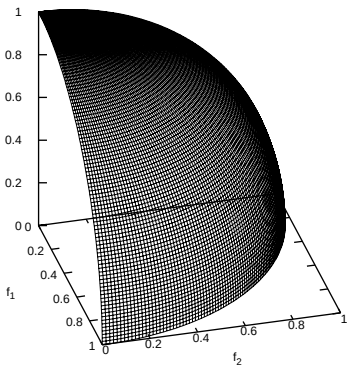


Figure 3.14: Pareto-Optimal Frontier of DTLZ2

where

$$g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \tag{3.33}$$

$$x_i \in [0,1] \quad i = 1, 2, \ldots, 12 \text{ and } x_M = \{x_3, x_4, \ldots, x_{12}\}$$

The problem's Pareto-optimal frontier corresponds to $x_M^* = 0.5$. We present indicator results in Table 3.15. In this problem, TDEA performs better than all other metaheuristics in both metrics (see Table 3.16). SPEA2, ϵ-MOEA and IBEA together come second. Interestingly, IBEA's performance is noticeably better than that in DTLZ1.

Table 3.15: Indicator Results for DTLZ2

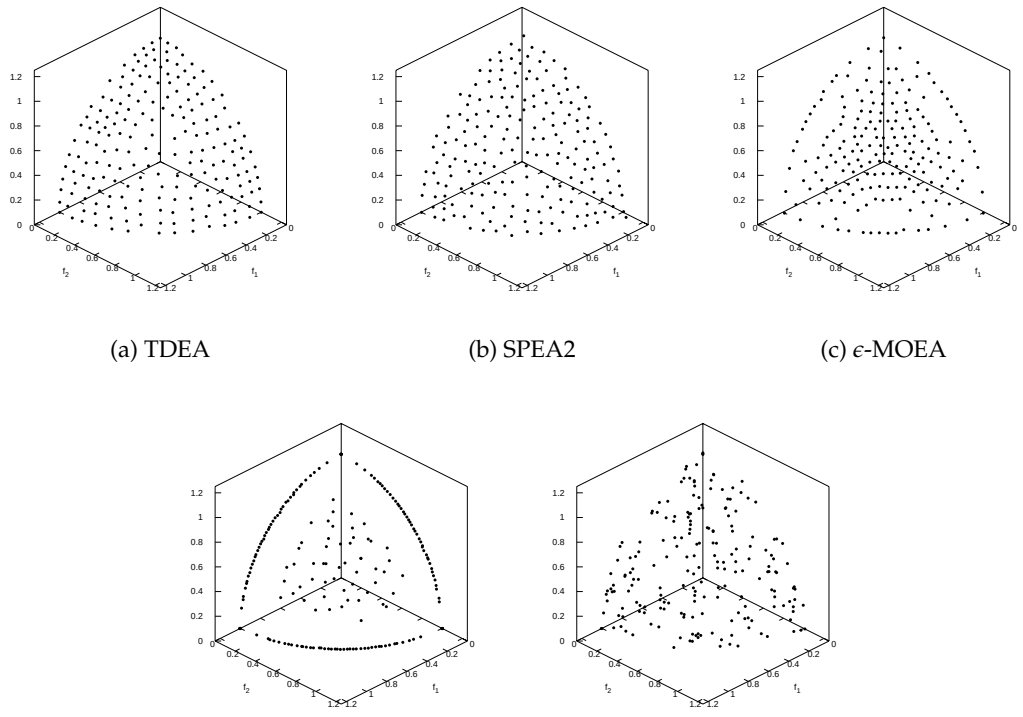
	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.43340	0.001060	0.0001220	0.00000234	7.51
ϵ-MOEA	0.43057	0.000498	0.0001620	0.00000212	4.37
IBEA	0.43250	0.000465	0.0002720	0.00000579	20.39
NSGA2	0.40364	0.003630	0.0001730	0.00000570	37.25
PAES	0.28964	0.062830	0.0006860	0.00030600	42.25
PESA2	0.09511	0.058380	0.0016690	0.00034400	245.07
SPEA2	0.42594	0.000960	0.0001300	0.00000169	683.86
True Pareto	0.47402	-	0	-	-

Figure 3.15 shows one example each for the best performing MOEAs. It can be seen that TDEA and SPEA2 maintain superior diversity than the others. Although IBEA converges

Table 3.16: Test Results for DTLZ2

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.002824	0	TDEA	-0.000040	0	TDEA
IBEA	0.000897	0	TDEA	-0.000150	0	TDEA
NSGA2	0.029753	0	TDEA	-0.000051	0	TDEA
PAES	0.143759	0	TDEA	-0.000564	0	TDEA
PESA2	0.338285	0	TDEA	-0.001548	0	TDEA
SPEA2	0.007458	0	TDEA	-0.000008	0	TDEA

well, most of its solutions are towards the edges. The solutions of ϵ-MOEA are not as finely-spaced as in DTLZ1. We see that there are gaps towards the ends and between edges and the center.



(a) TDEA (b) SPEA2 (c) ϵ-MOEA

(d) IBEA (e) NSGA2

Figure 3.15: DTLZ2 Plots

DTLZ3

DTLZ3 is the same problem with DTLZ2, except that its g function is changed. The change does not affect the Pareto-optimality conditions but adds multimodality. That is, it has many local Pareto-optimal frontiers. It tests the ability of an MOEA to converge the true Pareto-optimal frontier.

Minimize

$$f_1 = (1 + g(x_M)) \cos(0.5x_1\pi) \cos(0.5x_2\pi) \tag{3.34}$$

$$f_2 = (1 + g(x_M)) \cos(0.5x_1\pi) \sin(0.5x_2\pi) \tag{3.35}$$

$$f_3 = (1 + g(x_M)) \sin(0.5x_1\pi) \tag{3.36}$$

where

$$g(x_M) = 100\left[|x_M| + \sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right] \tag{3.37}$$

$$x_i \in [0,1] \quad i = 1, 2, \ldots, 12 \text{ and } x_M = \{x_3, x_4, \ldots, x_{12}\}$$

As can be seen from Table 3.17, SPEA2 and TDEA have better values in both metrics. SPEA2's values are better than those of TDEA, although Inverted Generational Distance difference is not statistically significant (Table 3.18). These two algorithms are followed by ϵ-MOEA and NSGA2. One remarkable note is that IBEA is unable to converge the Pareto-optimal front.

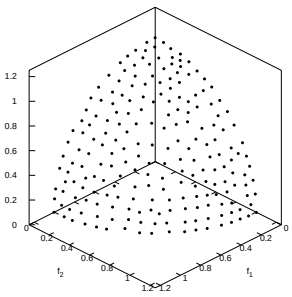
Table 3.17: Indicator Results for DTLZ3

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.42843	0.002540	0.0001240	0.00000332	4.34
ϵ-MOEA	0.42041	0.004660	0.0001820	0.00000669	2.48
IBEA	0	0	0.0460900	0.02726000	20.81
NSGA2	0.40775	0.004110	0.0001690	0.00000526	49.41
PAES	0.19530	0.110900	0.0012870	0.00079800	8.54
PESA2	0.07773	0.044680	0.0017860	0.00029800	186.62
SPEA2	0.43181	0.002030	0.0001260	0.00000161	251.61
True Pareto	0.47402	-	0	-	-

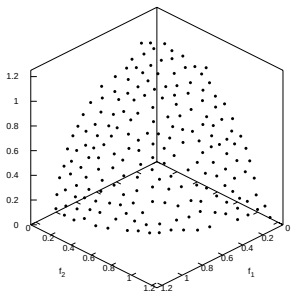
We present sample plots for TDEA, SPEA2, ϵ-MOEA and NSGA2 in Figure 3.16. We observe similar patterns to those in DTLZ2.

Table 3.18: Test Results for DTLZ3

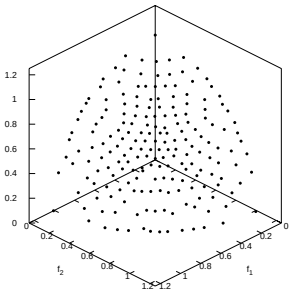
	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.008017	0	TDEA	-0.000058	0	TDEA
IBEA	N/A	N/A	TDEA	N/A	N/A	TDEA
NSGA2	0.020680	0	TDEA	-0.000045	0	TDEA
PAES	0.233127	0	TDEA	-0.001163	0	TDEA
PESA2	0.350698	0	TDEA	-0.001662	0	TDEA
SPEA2	-0.003386	0	SPEA2	-0.000002	0.004	none



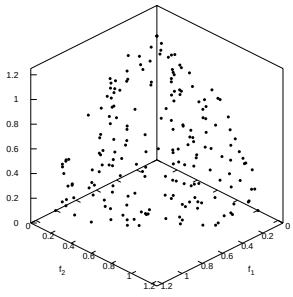
(a) TDEA



(b) SPEA2



(c) ϵ-MOEA



(d) NSGA2

Figure 3.16: DTLZ3 Plots

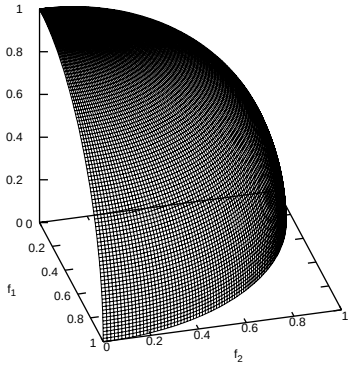


Figure 3.17: Pareto-Optimal Frontier of DTLZ3

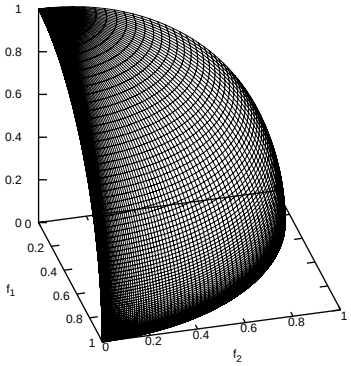


Figure 3.18: Pareto-Optimal Frontier of DTLZ4

DTLZ4

DTLZ4 has a concave and unimodal Pareto-optimal frontier (Figure 3.18). However, the density of solutions among the frontier is different. Hence, it tests an algorithm's ability to maintain a good diversity of solutions.

Minimize

$$f_1 = (1 + g(x_M)) \cos(0.5x_1^\alpha \pi) \cos(0.5x_2^\alpha \pi) \tag{3.38}$$

$$f_2 = (1 + g(x_M)) \cos(0.5x_1^\alpha \pi) \sin(0.5x_2^\alpha \pi) \tag{3.39}$$

$$f_3 = (1 + g(x_M)) \sin(0.5x_1^\alpha \pi) \tag{3.40}$$

where

$$g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \tag{3.41}$$

$$x_i \in [0, 1] \quad i = 1, 2, \ldots, 12 \text{ , } x_M = \{x_3, x_4, \ldots, x_{12}\} \text{ and } \alpha = 100$$

Table 3.19 shows the indicator values of algorithms. It can be seen that compared to the previous problems the standard deviations are very high for both metrics. This is because all of the metaheuristics are stuck in a particular region of the Pareto-optimal frontier during some of the runs. Among all MOEAs, NSGA2 has the lowest standard deviation. This indicates that NSGA2 is able to recover better than the other metaheuristics, most probably because its nondominated sorting mechanism preserves more lateral diversity than the other methods. However, this is not enough for it to outperform IBEA. TDEA is behind SPEA2 and ϵ-MOEA although the difference is not statistically significant.

Table 3.19: Indicator Results for DTLZ4

	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.36470	0.112000	0.0005330	0.00068800	6.61
ϵ-MOEA	0.40710	0.073100	0.0003120	0.00046700	4.28
IBEA	0.42764	0.025720	0.0002310	0.00008870	20.32
NSGA2	0.40712	0.003280	0.0001760	0.00001150	37.85
PAES	0.00506	0.028020	0.0023150	0.00016400	3.66
PESA2	0.11010	0.144000	0.0018050	0.00074500	101.75
SPEA2	0.40535	0.064980	0.0002640	0.00043300	645.33
True Pareto	0.47275	-	0	-	-

Table 3.20: Test Results for DTLZ4

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	-0.042404	0.028	none	0.000221	0.063	none
IBEA	-0.062915	0	IBEA	0.000303	0.003	IBEA
NSGA2	-0.042398	0.01	none	0.000357	0.001	NSGA2
PAES	0.359662	0	TDEA	-0.001782	0	TDEA
PESA2	0.254599	0	TDEA	-0.001272	0	TDEA
SPEA2	-0.040627	0.029	none	0.000269	0.022	none

In Figure 3.19, we present the plots of well-converged final populations of TDEA, SPEA2, ϵ-MOEA, IBEA and NSGA2. We observe that when algorithms are able to converge, TDEA and SPEA2 produce populations with best diversity.

DTLZ5

DTLZ5 is a modification of DTLZ2, which turns its Pareto-optimal frontier into a curve as seen in Figure 3.21.

Minimize

$$f_1 = (1 + g(x_M)) \cos(\theta_1) \cos(\theta_2) \tag{3.42}$$

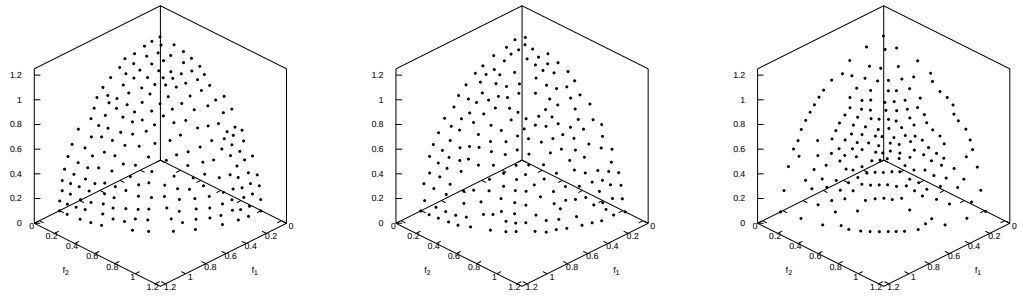
$$f_2 = (1 + g(x_M)) \cos(\theta_1) \sin(\theta_2) \tag{3.43}$$

$$f_3 = (1 + g(x_M)) \sin(\theta_1) \tag{3.44}$$

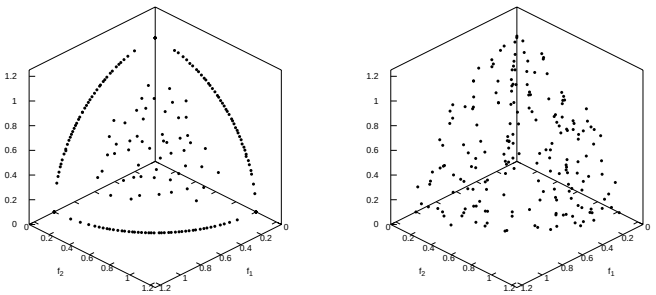
where

$$g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \tag{3.45}$$

$$\theta_1 = \tfrac{\pi}{2} x_1 \tag{3.46}$$



(a) TDEA (b) SPEA2 (c) ϵ-MOEA



(d) IBEA (e) NSGA2

Figure 3.19: DTLZ4 Plots

$$\theta_2 = \frac{\pi}{4(1+g(x_M))}(1 + 2g(x_M)x_2) \tag{3.47}$$
$$x_i \in [0,1] \quad i = 1,2,\ldots,12 \text{ and } x_M = \{x_3, x_4, \ldots, x_{12}\}$$

The Pareto-optimal frontier is found when $x_M^* = 0.5$ and objective functions satisfy $\sum_{i=1}^{3} f_i^2 = 1$. As it can be seen from Table 3.21, none of the algorithms has a problem in converging to the Pareto-optimal frontier. The test results in Table 3.22 show that all algorithms perform very well in converging the Pareto-optimal frontier.

We present plots of TDEA, SPEA2, ϵ-MOEA, IBEA and NSGA2 in Figure 3.20. While TDEA, SPEA2 and NSGA2 maintain good diversity throughout the entire frontier, IBEA and ϵ-MOEA lose solutions towards the ends.

DTLZ6

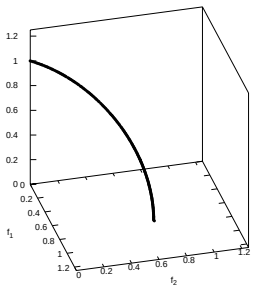
DTLZ6 is a modified version of DTLZ5 with a different g function, which makes it a more difficult problem. The change causes MOEAs to find dominated surfaces instead of a curve.

Table 3.21: Indicator Results for DTLZ5

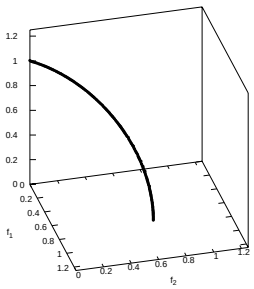
	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.09429	0.000037	0.0000102	0.00000021	5.94
ϵ-MOEA	0.09444	0.000012	0.0000272	0.00000041	3.69
IBEA	0.09376	0.000096	0.0000529	0.00000503	19.91
NSGA2	0.09419	0.000057	0.0000129	0.00000056	35.51
PAES	0.09248	0.000424	0.0000264	0.00000256	16.12
PESA2	0.09301	0.000189	0.0000210	0.00000153	140.05
SPEA2	0.09433	0.000051	0.0000099	0.00000017	404.69
True Pareto	0.09587	-	0	-	-

Table 3.22: Test Results for DTLZ5

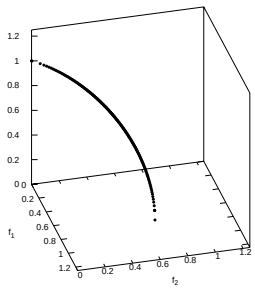
	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	-0.000151	0	ϵ-MOEA	-0.000017	0	TDEA
IBEA	0.000523	0	TDEA	-0.000043	0	TDEA
NSGA2	0.000102	0	TDEA	-0.000003	0	TDEA
PAES	0.001803	0	TDEA	-0.000016	0	TDEA
PESA2	0.001274	0	TDEA	-0.000011	0	TDEA
SPEA2	-0.000038	0	SPEA2	0.000000	0	none



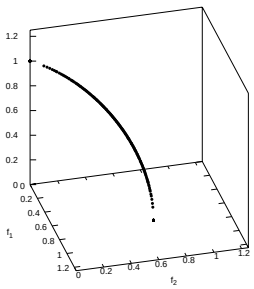
(a) TDEA



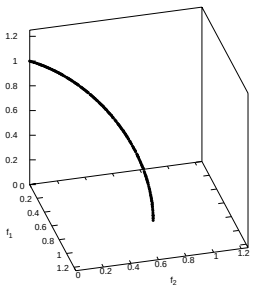
(b) SPEA2



(c) ϵ-MOEA



(d) IBEA



(e) NSGA2

Figure 3.20: DTLZ5 Plots

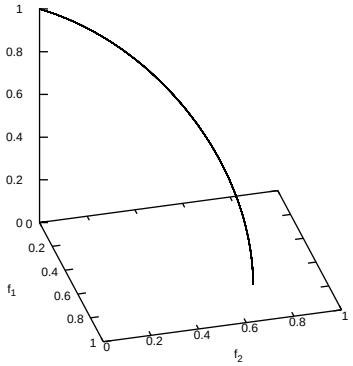


Figure 3.21: Pareto-Optimal Frontiers of DTLZ5 and DTLZ6

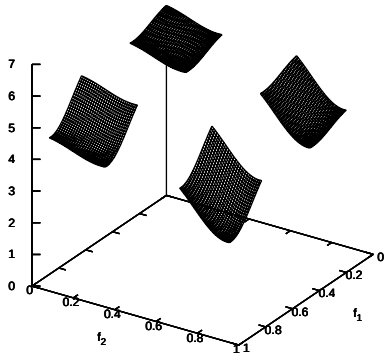


Figure 3.22: Pareto-Optimal Frontier of DTLZ7

It has the same Pareto-optimal frontier as DTLZ5 (Figure 3.21).

Minimize

$$f_1 = (1 + g(x_M)) \cos(\theta_1) \cos(\theta_2) \tag{3.48}$$

$$f_2 = (1 + g(x_M)) \cos(\theta_1) \sin(\theta_2) \tag{3.49}$$

$$f_3 = (1 + g(x_M)) \sin(\theta_1) \tag{3.50}$$

where

$$g(x_M) = \sum_{x_i \in x_M} (x_i)^{0.1} \tag{3.51}$$

$$\theta_1 = \frac{\pi}{2} x_1 \tag{3.52}$$

$$\theta_2 = \frac{\pi}{4(1+g(x_M))}(1 + 2g(x_M)x_2) \tag{3.53}$$

$$x_i \in [0, 1] \quad i = 1, 2, \ldots, 12 \text{ and } x_M = \{x_3, x_4, \ldots, x_{12}\}$$

Table 3.23 shows that SPEA2, NSGA2, PAES and PESA2 are not affected at all by the change. On the other hand, TDEA and ϵ-MOEA are unable to converge as well as they do in DTLZ5, while IBEA is unable to find any solutions in the nondominated range. Test results in Table 3.24 also confirm that.

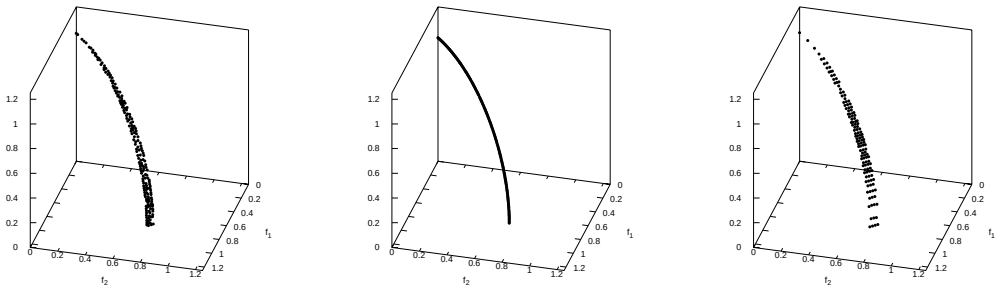
The plots given in Figure 3.23 display that TDEA and ϵ-MOEA find dominated surfaces as indicated above. It shows that these algorithms have some difficulty in converging to the Pareto-optimal frontier. To test whether they can converge better, we make further 50 runs with $320,000$ function evaluations for these two algorithms. As Table 3.25 indicates, TDEA remarkable improves with more iterations while ϵ-MOEA produces almost the same results. The test results in Table 3.26 as well as the plots given in Figure 3.24 also show the

Table 3.23: Indicator Results for DTLZ6

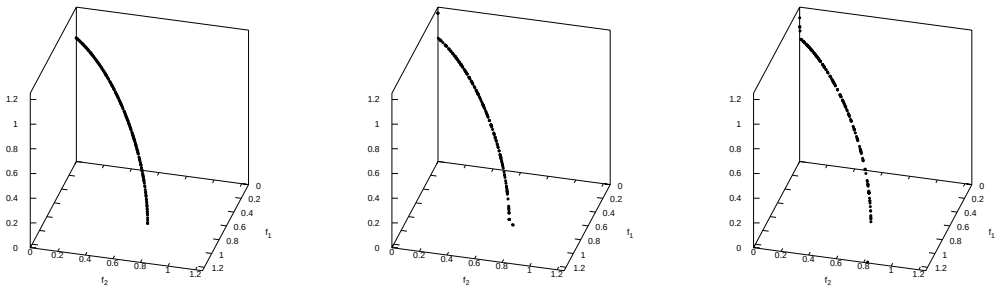
	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.06927	0.001243	0.0001300	0.00000830	6.48
ϵ-MOEA	0.05571	0.010250	0.0002480	0.00008330	3.27
IBEA	0.00000	0.000000	0.0041340	0.00202900	20.39
NSGA2	0.09245	0.005588	0.0000223	0.00002780	37.69
PAES	0.09247	0.001803	0.0000504	0.00012700	14.62
PESA2	0.08904	0.007370	0.0000418	0.00003010	172.39
SPEA2	0.09363	0.003571	0.0000131	0.00001540	547.52
True Pareto	0.09587	-	0	-	-

Table 3.24: Test Results for DTLZ6

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.013553	0	TDEA	-0.000118	0	TDEA
IBEA	N/A	N/A	TDEA	N/A	N/A	TDEA
NSGA2	-0.023186	0	NSGA2	0.000108	0	NSGA2
PAES	-0.023206	0	PAES	0.000080	0	PAES
PESA2	-0.019779	0	PESA2	0.000089	0	PESA2
SPEA2	-0.024366	0	SPEA2	0.000117	0	SPEA2



(a) TDEA (b) SPEA2 (c) ϵ-MOEA



(d) NSGA2 (e) PAES (f) PESA2

Figure 3.23: DTLZ6 Plots

significant difference. Note that both algorithms are still faster than the other algorithms despite the increased number of function evaluations.

Table 3.25: Indicator Results for DTLZ6 with 320000 Function Evaluations

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.08860	0.000995	0.0000349	0.00000195	10.48
ϵ-MOEA	0.05598	0.010370	0.0002458	0.00008427	6.41
True Pareto	0.09587	-	0	-	-

Table 3.26: Test Results for DTLZ6 with 320000 Function Evaluations

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.032610	0	TDEA	-0.000210	0	TDEA

DTLZ7

DTLZ7's Pareto-optimal frontier consists of 2^{p-1} disconnected parts. We present its frontier for the 3-objective case in (Figure 3.22). The density of solutions is uniform throughout the frontier. It tests an algorithm's ability to maintain populations in all disconnected regions.

Minimize

$$f_1 = x_1 \tag{3.54}$$

$$f_2 = x_2 \tag{3.55}$$

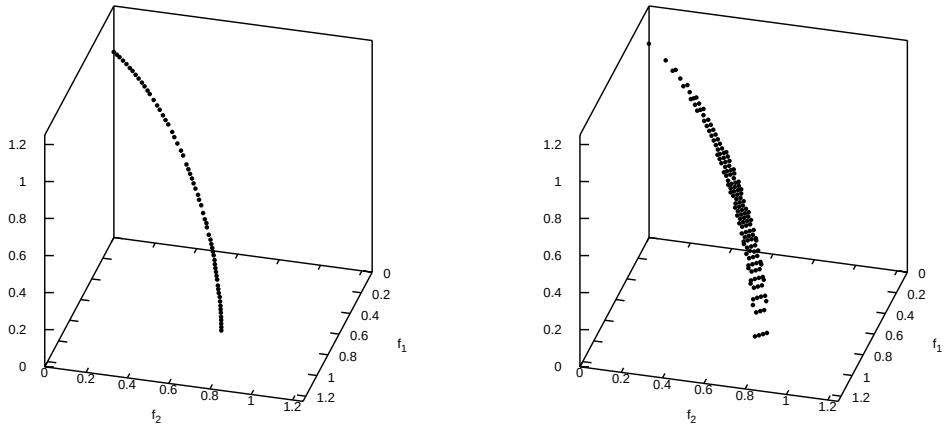
$$f_3 = (1 + g(x_M))\, h\,(f_1, f_2, g) \tag{3.56}$$

where

$$g(x_M) = 1 + \tfrac{9}{|x_M|} \sum_{x_i \in x_M} x_i \tag{3.57}$$

$$h(f_1, f_2, g) = 3 - \sum_{i=1}^{2} \left[\tfrac{f_i}{1+g} \left(1 + \sin(3\pi f_i)\right) \right] \tag{3.58}$$

$$x_i \in [0, 1] \quad i = 1, 2, \ldots, 22 \text{ and } x_M = \{x_3, x_4, \ldots, x_{22}\}$$



(a) TDEA (b) ϵ-MOEA

Figure 3.24: DTLZ6 Plots with 320000 Function Evaluations

As can be seen from Table 3.27, all algorithms except PAES and PESA2 converge successfully the Pareto-optimal frontier. IBEA and TDEA have large standard deviation values, which indicate that they perform significantly worse in some of the runs. Because of that, TDEA cannot outperform ϵ-MOEA and SPEA2 in the Hypervolume measure (Table 3.28) although it has a higher mean. A similar result is seen in the Inverted Generational Distance comparisons.

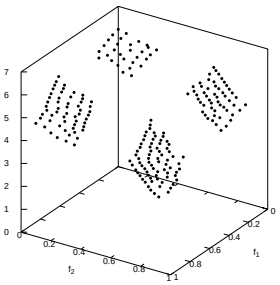
Table 3.27: Indicator Results for DTLZ7

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.32055	0.013960	0.0001620	0.00030400	7.58
ϵ-MOEA	0.31630	0.001370	0.0000803	0.00000054	4.84
IBEA	0.30160	0.037370	0.0006860	0.00057600	20.76
NSGA2	0.31084	0.002290	0.0001270	0.00000725	39.11
PAES	0.17241	0.025460	0.0021630	0.00025200	33.94
PESA2	0.16350	0.075000	0.0019150	0.00085800	248.82
SPEA2	0.31920	0.000951	0.0001030	0.00000268	551.75
True Pareto	0.34437	-	0	-	-

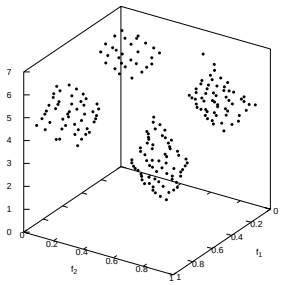
We present plots of TDEA, SPEA2, ϵ-MOEA, IBEA and NSGA2 in Figure 3.25. ϵ-MOEA displays the best diversity with TDEA and SPEA2 following it. NSGA2 produces an irregular pattern, while IBEA has difficulty in finding solutions towards the center.

Table 3.28: Test Results for DTLZ7

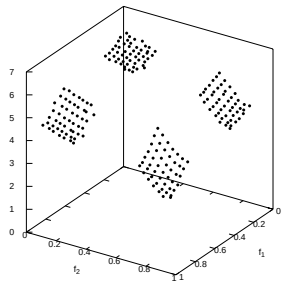
	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.004250	0.037	none	0.000082	0.062	none
IBEA	0.018949	0.001	TDEA	-0.000524	0	TDEA
NSGA2	0.009705	0	TDEA	0.000035	0.414	none
PAES	0.148138	0	TDEA	-0.002001	0	TDEA
PESA2	0.157003	0	TDEA	-0.001753	0	TDEA
SPEA2	0.001343	0.501	none	0.000059	0.174	none



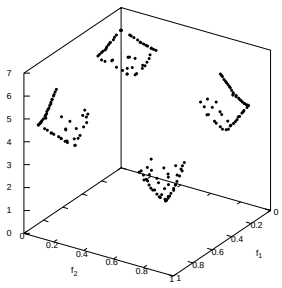
(a) TDEA



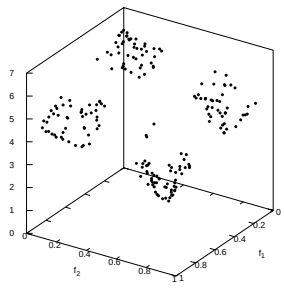
(b) SPEA2



(c) ϵ-MOEA



(d) IBEA



(e) NSGA2

Figure 3.25: DTLZ7 Plots

3.2.4 5-Objective Problems

In 5-objective tests, we use problems DTLZ1 and DTLZ2. However, we do not perform tests with PESA-II and PAES since their performances are significantly lower than the other algorithms in 3-objective tests.

DTLZ1-5D

As in 3-objective DTLZ1, DTLZ1-5D has a linear Pareto-optimal frontier. It is a multimodal problem with many local Pareto-optimal frontiers. These local frontiers attract MOEAs and cause difficulties for them to converge the true frontier.

Minimize

$$f_1 = \tfrac{1}{2}x_1x_2x_3x_4\,(1 + g\,(x_M)) \tag{3.59}$$

$$f_2 = \tfrac{1}{2}x_1x_2x_3(1 - x_4)\,(1 + g\,(x_M)) \tag{3.60}$$

$$f_3 = \tfrac{1}{2}x_1x_2(1 - x_3)\,(1 + g\,(x_M)) \tag{3.61}$$

$$f_4 = \tfrac{1}{2}x_1(1 - x_2)\,(1 + g\,(x_M)) \tag{3.62}$$

$$f_5 = \tfrac{1}{2}(1 - x_1)\,(1 + g\,(x_M)) \tag{3.63}$$

where

$$g(x_M) = 100\left[|x_M| + \sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos\left(20\pi\,(x_i - 0.5)\right)\right] \tag{3.64}$$

$$x_i \in [0,1] \quad i = 1, 2, \ldots, 9 \text{ and } x_M = \{x_5, x_6, \ldots, x_9\}$$

Table 3.29 shows that TDEA and ϵ-MOEA are the most successful algorithms in converging the Pareto-optimal frontier. While IBEA and NSGA2 partially converge, SPEA2 is unable to find any solutions within the nondominated range. In test results (Table 3.30), TDEA outperforms all algorithms in both metrics. Note that the standard deviation values of TDEA are also the smallest, which means that it has the least difficulty in finding quality final populations compared to other algorithms. One another remark is the duration. TDEA's average time to complete its runs is the second among all algorithms.

DTLZ2-5D

DTLZ2-5D is the 5-objective version of DTLZ2, sharing the same properties.

Minimize

$$f_1 = (1 + g(x_M))\cos(0.5x_1\pi)\cos(0.5x_2\pi)\cos(0.5x_3\pi)\cos(0.5x_4\pi) \tag{3.65}$$

Table 3.29: Indicator Results for DTLZ1-5D

	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.95202	0.001570	0.0003520	0.00001990	25.745
ϵ-MOEA	0.91499	0.008360	0.0003920	0.00003100	13.3264
IBEA	0.56667	0.051770	0.0009150	0.00010600	114.295
NSGA2	0.15390	0.294300	0.0120600	0.01318000	163.369
SPEA2	0	0	0.1183300	0.03057000	2140

Table 3.30: Test Results for DTLZ1-5D

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			**Inverted Gen. Distance**		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.037025	0	TDEA	-0.000040	0	TDEA
IBEA	0.385347	0	TDEA	-0.000562	0	TDEA
NSGA2	0.798130	0	TDEA	-0.011709	0	TDEA
SPEA2	N/A	N/A	TDEA	N/A	N/A	TDEA

$$f_2 = (1 + g(x_M)) \cos(0.5x_1\pi) \cos(0.5x_2\pi) \cos(0.5x_3\pi) \sin(0.5x_4\pi) \tag{3.66}$$

$$f_3 = (1 + g(x_M)) \cos(0.5x_1\pi) \cos(0.5x_2\pi) \sin(0.5x_3\pi) \tag{3.67}$$

$$f_4 = (1 + g(x_M)) \cos(0.5x_1\pi) \sin(0.5x_2\pi) \tag{3.68}$$

$$f_5 = (1 + g(x_M)) \sin(0.5x_1\pi) \tag{3.69}$$

where

$$g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \tag{3.70}$$

$$x_i \in [0, 1] \quad i = 1, 2, \ldots, 14 \text{ and } x_M = \{x_5, x_6, \ldots, x_{14}\}$$

Indicator results in Table 3.31 show that all algorithms converge the Pareto-optimal frontier, although there are significant differences in Hypervolume and Inverted Generational Distance metrics. This is also confirmed in test results (Table 3.31). TDEA and IBEA display better Hypervolume performances than other algorithms, indicating better convergence. However, TDEA scores much better in the Inverted Generational Distance compared to IBEA. In addition, it needs less time to finish its runs than all algorithms except ϵ-MOEA.

3.2.5 Effects of Changing τ

In TDEA, τ controls the extent of the territory of a solution. Its value changes the hypervolume that a solution occupies in the objective space. Since the total nondominated

Table 3.31: Indicator Results for DTLZ2-5D

	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
TDEA	0.71209	0.001900	0.0004340	0.00000892	41.1262
ϵ-MOEA	0.69352	0.001850	0.0004860	0.00000593	22.4666
IBEA	0.72338	0.000542	0.0006330	0.00001590	106.964
NSGA2	0.41157	0.030360	0.0007940	0.00006910	165.432
SPEA2	0.59258	0.010080	0.0005950	0.00003600	4692.72

Table 3.32: Test Results for DTLZ2-5D

	$H_0 : \mu_{TDEA} = \mu_{Contender}$ vs $H_1 : \mu_{TDEA} \neq \mu_{Contender}$					
	Hypervolume			Inverted Gen. Distance		
Contender	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ϵ-MOEA	0.018571	0	TDEA	-0.000052	0	TDEA
IBEA	-0.011289	0	IBEA	-0.000199	0	TDEA
NSGA2	0.300515	0	TDEA	-0.000360	0	TDEA
SPEA2	0.119504	0	TDEA	-0.000161	0	TDEA

hypervolume is limited, the size of the population depends on τ. The population size is expected to be smaller with a larger τ compared to that with a smaller τ. TDEA scales all objectives into the $[0, 1]$ range, hence τ is always between 0 and 1. However, there is no straightforward way of determining the right value for τ, because the space between solutions changes with the shape of the Pareto-optimal frontier. Also the number of objectives affects the value of τ. For large number of objectives, τ needs to be large in order to keep the population in size reasonable.

A smaller τ yields a better approximation of the Pareto-optimal frontier than a larger τ. However, computational time requirement increases as τ decreases. To observe the effect of different τ values, we make test runs on problems ZDT4, DTLZ1 and DTLZ2 with various τ values and compare their results.

Table 3.33: Indicator Results for ZDT4

	Hypervolume		Inverted Gen. Distance			**Final**
τ	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**	**Pop. Size**
0.0005	0.66459	0.000465	0.0000031	0.00000285	3.7	~ 1400
0.001	0.66439	0.000531	0.0000031	0.00000098	2.7	~ 750
0.005	0.66258	0.000379	0.0000087	0.00000032	1.7	~ 170
True Pareto	0.66599	-	0	-	-	

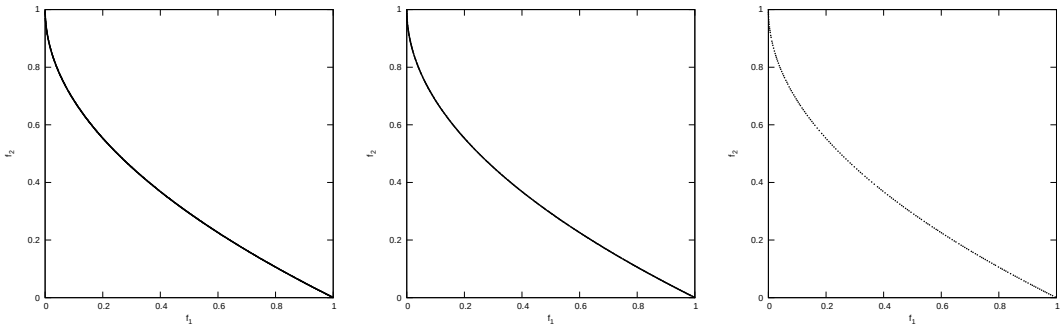
Table 3.34: Indicator Results for DTLZ1

	Hypervolume		**Inverted Gen. Distance**			**Final**
τ	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**	**Pop. Size**
0.01	0.82311	0.000777	0.0000254	0.00000101	86.1	~ 3000
0.02	0.81760	0.000414	0.0000456	0.00000048	22.4	~ 800
0.03	0.81100	0.000347	0.0000664	0.00000092	15.4	~ 400
True Pareto	0.83050	-	0	-	-	

Table 3.35: Indicator Results for DTLZ2

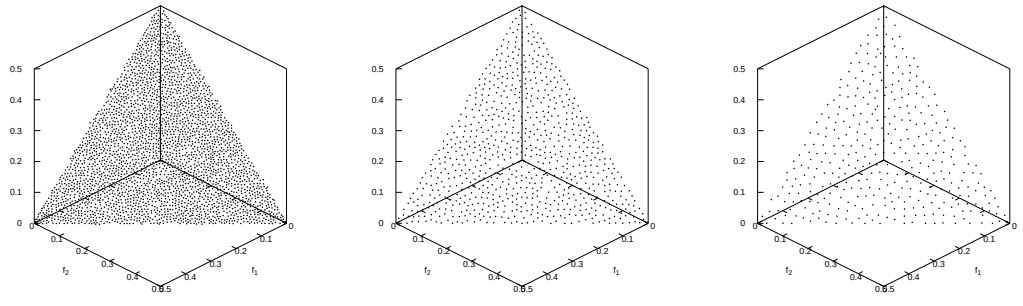
	Hypervolume		**Inverted Gen. Distance**			**Final**
τ	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**	**Pop. Size**
0.02	0.46140	0.000164	0.0000394	0.00000038	87.1	~ 2000
0.03	0.45666	0.000175	0.0000550	0.00000045	45.1	~ 1000
0.05	0.44541	0.000554	0.0000876	0.00000100	25.7	~ 400
True Pareto	0.47402	-	0	-	-	

In Table 3.33, we observe that Hypervolume values slightly increase as τ decreases. Average duration of runs also increase when τ is decreased. Figure 3.26 illustrates the difference in the details of final populations. Tables 3.34 and 3.35 give results for 3-objective problems DTLZ1 and DTLZ2. It can be seen that decreasing τ gives more detail in return for increased computational cost. In all problems, decreasing τ increases the size of final populations. However, using the same τ for different problems does not lead to the same number of solutions in final populations. The plots in Figures 3.27 and 3.28 clearly display the difference in the densities of the final populations when τ is changed. Note that in all of the problems and for all τ values, TDEA successfully maintains the diversity among the entire population.



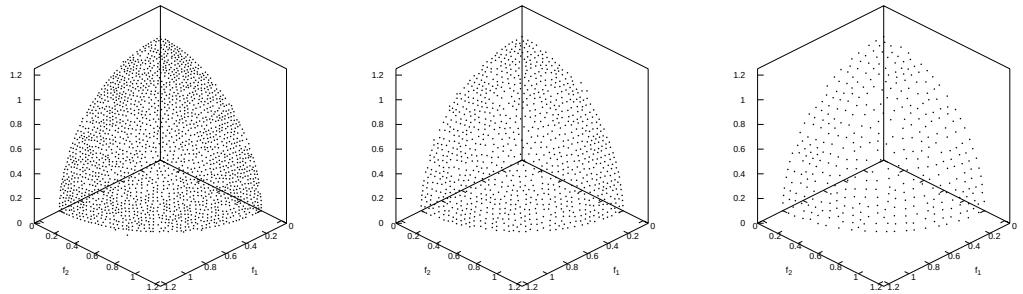
(a) τ = 0.0005 (b) τ = 0.001 (c) τ = 0.005

Figure 3.26: ZDT4 Plots with Different τ Values



(a) τ = 0.01 (b) τ = 0.03 (c) τ = 0.05

Figure 3.27: DTLZ1 Plots with Different τ Values



(a) τ = 0.02 (b) τ = 0.03 (c) τ = 0.05

Figure 3.28: DTLZ2 Plots with Different τ Values

3.2.6 Discussions

The tests show that TDEA is able to converge the true Pareto-optimal frontier in all of the problems, regardless of the number of objectives and the shape of the Pareto-optimal frontier. It maintains a uniform diversity of solutions over the entire Pareto-optimal frontier and obtains a good representation of the true frontier. We observe that it performs better than other metaheuristics in most of the problems with respect to both the Hypervolume and the Inverted Generational Distance metrics. Thanks to its territory defining property, it eliminates the need for an explicit diversity preserving operator, providing the fastest execution time after ϵ-MOEA. As shown in the plots, this mechanism also helps the algorithm to obtain the best spread of solutions together with SPEA2.

CHAPTER 4

PREFERENCE INCORPORATION

Although many multiobjective evolutionary algorithms try to approximate the entire Pareto-optimal frontier, the final goal in solving multiobjective optimization problems is to find a single or few solutions that satisfy the preferences of the decision maker (DM). These MOEAs are *a posteriori* methods that do not involve any preference-based decision making until the optimization stage ends. They first generate many solutions all over the Pareto-optimal frontier. After that, they present these solutions to the DM for him/her to make a decision. Although they avoid some computational issues by relegating decision making to post optimization, some new problems are introduced (Rachmawati and Srinivasan, 2006). First of all, approximating the entire frontier is a difficult and computationally expensive task, since most of the procedures used in these algorithms are costly. In addition, this is usually an unnecessary task. Most of the time, DMs know beforehand that they are not interested at all in some parts of the Pareto-optimal frontier. Since a posteriori methods do not incorporate such information, they spend needless effort in searching these regions. Furthermore, a search without a focus may not provide the necessary resolution about the regions that the DM has interest. On the other hand, guiding the search towards these regions allows algorithms to discover more details about them.

Considering these disadvantages of a posteriori methods, we implement a preference incorporation mechanism into the Territory Defining Evolutionary Algorithm. This mechanism alters the territories of solutions based on their locations. For solutions in the preferred regions, the territory size is set to a lower value so as to accept more solutions in these regions. On the other hand, solutions outside these regions control larger territories, causing closely located newcomers to be rejected. Such a territory modification scheme guides the search so that solutions in regions of interest are densely packed, while the rest of the regions are sparse.

Without changing the underlying preference incorporation mechanism, we propose *a priori* and *interactive* versions of the Territory Defining Evolutionary Algorithm. We call the former as the Preferred-Region Territory Defining Evolutionary Algorithm (prTDEA) and the latter as the Interactive Territory Defining Evolutionary Algorithm (iTDEA). In this chapter, we will first describe the common features of preference incorporation in two algorithms in Section 4.1. It is followed by the details and computational results of prTDEA and iTDEA in Sections 4.2 and 4.3, respectively.

4.1 Common Features

4.1.1 Variable Territory Sizes

In Section 3.2.5, we investigate the effects of different τ values on the population size and the quality of Pareto-optimal frontier approximation. The results show that a smaller τ brings better details, though with extra computational cost. However, as mentioned before, approximating the entire Pareto-optimal frontier is not very useful. On the contrary, focusing a particular region helps to avoid much of the additional effort. This is only possible if preferences of the DM can be integrated into the solution procedure. TDEA's territory defining property can be effectively utilized for this purpose. If the preferred regions are defined, then the algorithm can concentrate on these regions by shrinking the territories of the solutions falling within. That is, a smaller τ is used for such an offspring in the archive evaluation stage. This way, more solutions are allowed in those areas in the archive population, which means higher resolution and better approximation. On the other hand, solutions located elsewhere are evaluated using a larger τ. This prohibits dense populations in the regions in which the DM is not interested. Consequently, the search in these regions are discouraged, because unfavorable solutions have less chance to participate in genetic operations. In addition, while the size of the entire population is kept in a smaller size, the size of the focused region will remain large. Thus, each generation takes a shorter time, since the population size is trimmed down and fewer comparisons are to be done.

Note that the final population of the algorithm is *still* the approximation of the entire Pareto-optimal frontier. That is, solutions outside the preferred regions are still accepted, but their densities are lower than those in the areas of interest.

4.1.2 Favorable Weights

To make variable territory sizes operational, we have to define the regions to be focused on and determine whether solutions fall within these regions. In TDEA, we use the idea of favorable weights (Soylu and Köksalan, 2006) for this purpose. The favorable weights $\mathbf{w^s} = \{w_1^s, w_2^s, \ldots, w_p^s\}$ of solution s are a set of weights that minimizes its weighted Tchebycheff distance from the ideal point. They are computed as shown in the following equation. Note that we dropped s from the superscript to simplify the notation.

$$w_i = \begin{cases} \frac{1}{f_i^* - f_i} \left[\sum_{j=1}^{p} \frac{1}{f_j^* - f_j} \right]^{-1} & \text{if } f_j \neq f_j^* \text{ for all } j = 1, 2, \ldots, p \\ 1 & \text{if } f_i = f_i^* \\ 0 & \text{if } f_i \neq f_i^* \text{ but } \exists j \text{ such that } f_j = f_j^* \end{cases} \tag{4.1}$$

where f_i is the i^{th} objective value and f_i^* is the i^{th} element of the ideal objective vector.

These weights determine the direction in which a solution contributes the highest to the convergence. More specifically, the corresponding solution is in the direction $\frac{1}{w_i}, i = 1, 2, \ldots, p$ from the ideal point. We define a *preferred region* R_P by a set of Tchebycheff weights $\mathbf{W}^P$. Then, we say that a solution s is in R_P if its weights, $\mathbf{w}^s$, satisfy $\mathbf{w}^s \in \mathbf{W}^P$. Otherwise, s is not in R_P. In other words, a solution is a favorable one if its favorable weights are covered by the weight ranges of a preferred region.

4.2 Preferred-Region Territory Defining Evolutionary Algorithm

In this section, we present Preferred-Region the Territory Defining Evolutionary Algorithm. We start by explaining the details of the algorithm in Section 4.2.1. It will be followed by the simulation runs and comparisons in Section 4.2.2.

4.2.1 Details of the Algorithm

Preferred-Region Territory Defining Evolutionary Algorithm (prTDEA) is an application of the preference incorporation mechanism in TDEA. In prTDEA, DM states all his/her preference information before optimization. This is done by explicitly specifying $m = 1, 2, \ldots$ preference regions by weight sets. Then, each region and the remaining space is assigned its own τ value. An offspring is evaluated using the τ of the region to which it corresponds. An outline of the algorithm is given below:

1. Set $\bar{N}$ and T. Ask the user to specify his/her non-overlapping preferred regions

$R_1, R_2, \ldots, R_m$. Set $\tau_1, \tau_2, \ldots, \tau_m$. Specify an τ_U value corresponding to regions in which the DM is not interested. Set iteration count $t = 0$.

2. Create $\bar{N}$ random solutions to fill $P(0)$. Copy the nondominated solutions in $P(0)$ into $A(0)$.

3. Set $t \leftarrow t+1$. Choose a parent from each population $P(t)$ and $A(t)$. Recombine parents to create new offspring and apply mutation to it.

4. Test the offspring for acceptance into $P(t)$. If accepted, insert into $P(t)$ and go to the next step. Otherwise, go to Step 6.

5. Test the offspring for acceptance into $A(t)$. If accepted, insert into $A(t)$.

6. If the iteration limit is hit, that is, $t = T$, then stop and report the archive population. Otherwise go to Step 3.

It can be observed that prTDEA and TDEA are very similar. prTDEA allows multiple preference regions to be defined, whereas in TDEA there exists a single preference region, R_p, whose weight set is equal to the entire weight space. This leads to a difference between the archive acceptance procedure. In prTDEA, this stage involves the determination of favorable weights of an offspring. Also it is modified so that the offspring are evaluated with different τ values. The process starts with a dominance check. If any solution in the archive dominates offspring c, then c is rejected and the process is terminated. Otherwise, the second stage of the archive acceptance starts by removing all solutions dominated by c. Before checking for territory violation, we determine the preference region that contains c. For this purpose, we compute the favorable weights, $\mathbf{w}^c$, of c and determine the region i whose weight set contain these weights to set $\tau = \tau_i$. If c is not a favorable solution, then $\tau = \tau_U$. The rest is the same as TDEA. First, the closest solution s_{i^*} to c with respect to rectilinear distance is determined, using the scaled objective function values. Then, we check whether s_{i^*} is in the territory of c, which is defined as the region within τ distance in all objective values of c. c is rejected if the maximum scaled objective distance between s_{i^*} and c is smaller than the τ value. Otherwise, it is accepted. We give the details of the procedure below:

1. Test c against each solution $s_i \in A(t)$ for dominance. Mark solutions dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to next step.

2. Remove all marked solutions from $A(t)$.

3. If $A(t)$ is empty, accept c and insert into $A(t)$. Otherwise, go to the next step.

4. Calculate the favorable weights $\mathbf{w}^c$ of c with Equation 4.1 using scaled objective values.

5. Find the region i whose weight set contains the favorable weights of c and set $\tau = \tau_i$. If there exists no such region, then set $\tau = \tau_U$. That is, set τ as follows:

$$\tau = \begin{cases} \tau_i & \text{if } \exists R_i \text{ such that } \mathbf{w^c} \in \mathbf{W}^i \\ \tau_U & \text{otherwise} \end{cases} \tag{4.2}$$

6. Calculate the rectilinear distance $d_{ci} = \sum_{j=1}^{p} \left| \hat{f}_{cj} - \hat{f}_{ij} \right|$ of c to each solution $s_i \in A(t)$ using the scaled objective function values.

7. Find $i^* = \text{argmin}_i(d_{ci})$, that is, the solution s_{i^*} closest to c.

8. Find maximum scaled absolute objective difference between c and s_{i^*}. That is, find

$$\delta = \max_{j=1,2,\ldots,p} \left| \hat{f}_{cj} - \hat{f}_{i^*j} \right| \tag{4.3}$$

   where $\hat{f}_{cj}$ and $\hat{f}_{i^*j}$ are the scaled j^{th} objective values of offspring c and solution s_{i^*}, respectively.

9. If $\delta \geq \tau$, accept c and insert it into $A(t)$. Otherwise, reject c.

4.2.2 Simulation Runs and Comparisons

Since prTDEA focuses on regions specified by the DM, we expect that it finds better approximations of the preferred regions than does TDEA under the same test conditions. To test this claim, we make simulation runs on problems ZDT4, DTLZ1 and DTLZ2. For each problem, we create 4 tests, each having a different preference region. We define preference regions by a set of Tchebycheff weight ranges $\left[\mathbf{l^P}, \mathbf{u^P}\right] = \left\{\left[l_1^P, u_1^P\right], \left[l_2^P, u_2^P\right], \ldots, \left[l_p^P, u_p^P\right]\right\}$. In prTDEA, a small τ_s value is used for these preference regions, whereas a larger τ_l value is used for the remaining regions. On the other hand, TDEA uses the same small τ_s for the entire Pareto-optimal frontier. After runs are complete, we filter out solutions of two algorithms corresponding to preference regions and compute their Hypervolume and Inverted Generational Distance values as defined in Section 3.2.1. We also filter the true Pareto-optimal frontier representations that are used in Section 3.2 according to the preference regions. The details of each test are given in Table 4.1.

Table 4.1: Preference Test Parameters

	ZDT4	**DTLZ1**	**DTLZ2**
Test 1 Weight Ranges	[0.4, 0.6], [0.4, 0.6]	[0.25, 0.5], [0.25, 0.5], [0.25, 0.5]	[0.25, 0.5], [0.25, 0.5], [0.25, 0.5]
Test 2 Weight Ranges	[0.3, 0.7], [0.3, 0.7]	[0.2, 0.6], [0.2, 0.6], [0.2, 0.6]	[0.2, 0.6], [0.2, 0.6], [0.2, 0.6]
Test 3 Weight Ranges	[0.1, 0.3], [0.7, 0.9]	[0.1, 0.4], [0.3, 0.6], [0.1, 0.6]	[0.1, 0.4], [0.3, 0.6], [0.1, 0.6]
Test 4 Weight Ranges	[0.75, 1], [0, 0.25]	[0.4, 0.7], [0.1, 0.3], [0.2, 0.5]	[0.4, 0.7], [0.1, 0.3], [0.2, 0.5]
Function Evaluations	80000	160000	160000
Regular Population Size	200	400	400
τ_s	0.00001	0.005	0.01
τ_u	0.01	0.05	0.1
Replications	30	30	30

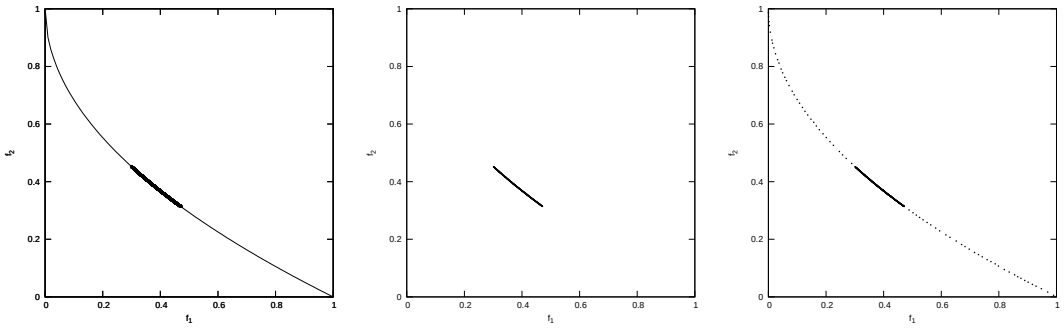
2-Objective Tests

We use problem ZDT4 to conduct 2-objective preference tests.

Test 1 The first test covers a region in the middle of the Pareto-optimal frontier (Figure 4.1a). As it can be observed from Table 4.2, prTDEA outperforms TDEA in both Hypervolume and Inverted Generational Distance metrics. Table 4.6 shows that the differences are statistically significant. In addition, it takes a shorter time to finish its runs on average. Plots of the results of the two algorithms are given in Figure 4.1.

Table 4.2: Indicator Results for ZDT4 Preference Test 1

	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
prTDEA	0.51388	0.00217	0.0000226	0.0000111	3.76
TDEA	0.50764	0.00470	0.0000513	0.0000232	6.72
True Pareto	0.51850	-	0	-	-



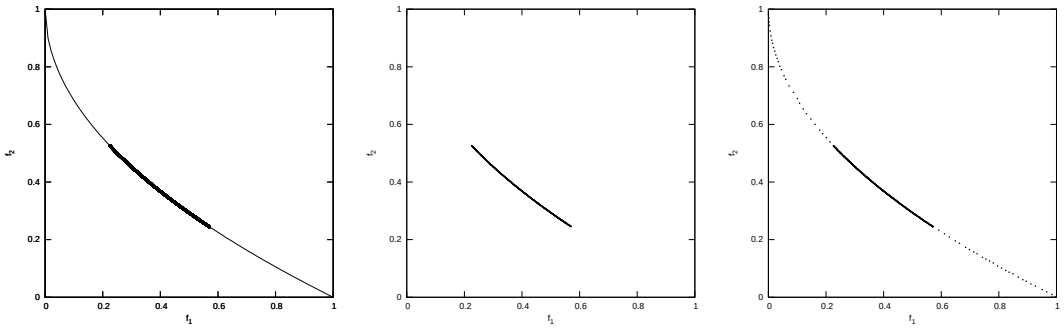
(a) Preference Region 1 (b) Filtered TDEA (c) prTDEA

Figure 4.1: ZDT4 Test 1 Plots

Test 2 The region in this test is an enlarged version of the region in Test 1 as seen in Figure 4.2a. Although prTDEA still gives better results than TDEA (Tables 4.3 and 4.6), the gap between the two algorithms decreases. This is because the focused region is larger. We present the plots of the results of the two algorithms in Figure 4.2.

Table 4.3: Indicator Results for ZDT4 Preference Test 2

	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
prTDEA	0.53478	0.00135	0.0000108	0.00000462	4.28
TDEA	0.53273	0.00227	0.0000174	0.00000779	6.72
True Pareto	0.53802	-	0	-	-



(a) Preference Region 2 (b) Filtered TDEA (c) prTDEA

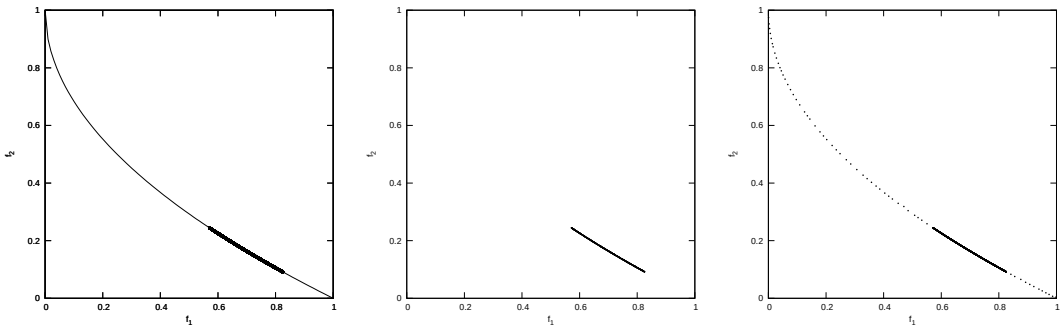
Figure 4.2: ZDT4 Test 2 Plots

Test 3 Figure 4.3a displays the preference region of this test, which is towards the better values of the second objective. In Table 4.4, we observe that prTDEA finds better results in both metrics. However, the difference between Inverted Generational Distance values is not statistically significant (Table 4.6), although the estimated difference is the largest among all tests. This is due to TDEA's high standard deviation. Figure 4.2 exhibits the plots of the results of the two algorithms.

Table 4.4: Indicator Results for ZDT4 Preference Test 3

	Hypervolume		Inverted Gen. Distance		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
prTDEA	0.51157	0.00195	0.0000145	0.00000139	4.26
TDEA	0.50299	0.02097	0.0000875	0.00005440	6.72
True Pareto	0.51527	-	0	-	-

Test 4 The preference region in Test 4 covers the best values of the first objective (Figure 4.4a). Similar to the previous results, Table 4.4 indicates that prTDEA performs better than TDEA in both metrics as well as the average time required to complete the runs. The



(a) Preference Region 3 (b) Filtered TDEA (c) prTDEA

Figure 4.3: ZDT4 Test 3 Plots

differences are statistically significant as seen in Table 4.6. The plots of the final populations are presented in Figure 4.4.

Table 4.5: Indicator Results for ZDT4 Preference Test 4

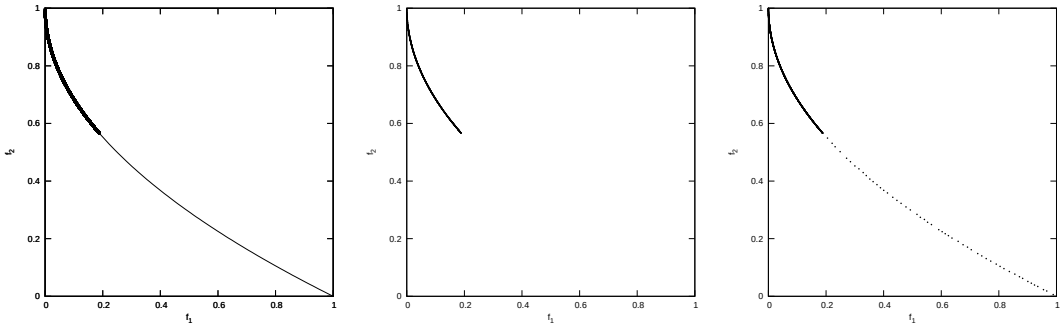
	Hypervolume		**Inverted Gen. Distance**		
Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
prTDEA	0.66225	0.00163	0.0000141	0.00000857	4.67
TDEA	0.66086	0.00183	0.0000204	0.00000914	6.72
True Pareto	0.66508	-	0	-	-

Table 4.6: Statistical Tests for ZDT4 Preference Tests

	$H_0 : \mu_{prTDEA} = \mu_{TDEA}$ vs $H_1 : \mu_{prTDEA} \neq \mu_{TDEA}$					
	Hypervolume			**Inverted Gen. Distance**		
Test	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
ZDT4 - 1	0.006241	0	prTDEA	-0.000029	0	prTDEA
ZDT4 - 2	0.002050	0	prTDEA	-0.000007	0	prTDEA
ZDT4 - 3	0.008584	0.033	prTDEA	-0.000073	0.191	none
ZDT4 - 4	0.001391	0.003	prTDEA	-0.000006	0.008	prTDEA

3-Objective Tests

We conduct 3-objective tests on DTLZ1 and DTLZ2 problems using the same desirable regions for both problems. The reason behind using two problems is to test whether the



(a) Preference Region 4 (b) Filtered TDEA (c) prTDEA

Figure 4.4: ZDT4 Test 4 Plots

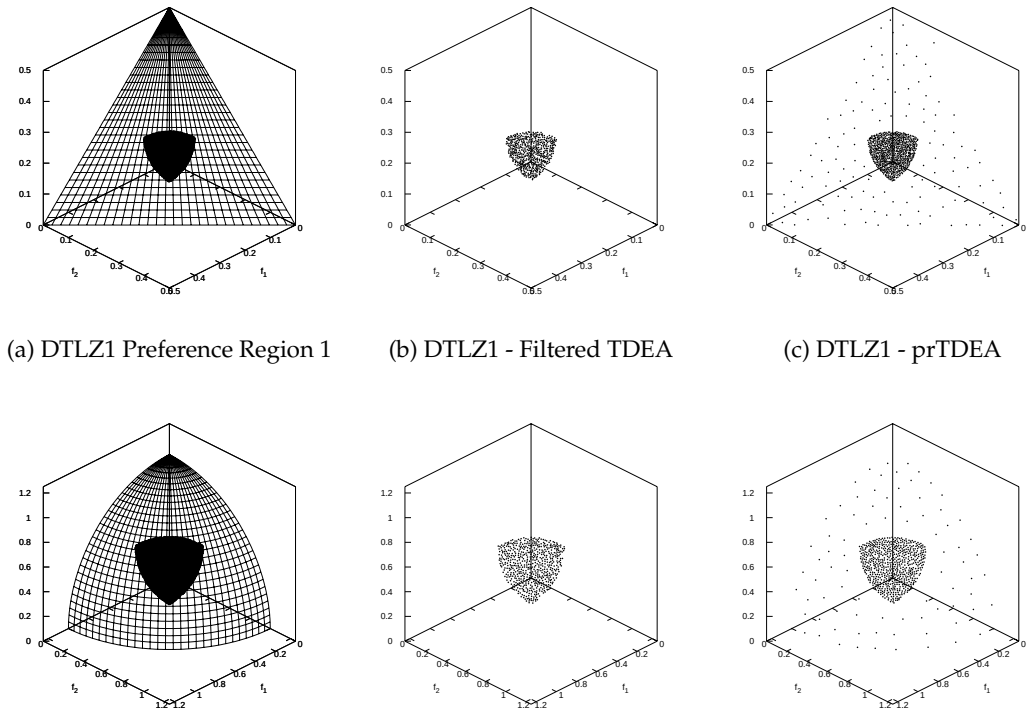
shape of the Pareto-optimal frontier affects the performance.

Test 1 The first preference region (Figures 4.5a and 4.5d) is set as the middle of the Pareto-optimal frontier. We see from Table 4.7 that prTDEA outperforms TDEA as in ZDT4 tests and the difference between computation times are substantially higher in favor of prTDEA. The difference between plots (Figure 4.5) are also clearly observable.

Table 4.7: Indicator Results for DTLZ1 and DTLZ2 Preference Test 1

		Hypervolume		**Inverted Gen. Distance**		
	Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
DTLZ1	prTDEA	0.29512	0.00249	0.000258	0.0000111	10.43
	TDEA	0.27598	0.00638	0.000419	0.0000558	205.23
	True Pareto	0.31194	-	0	-	-
DTLZ2	prTDEA	0.26707	0.000523	0.000204	0.00000189	17.40
	TDEA	0.25580	0.000768	0.000252	0.00000362	183.85
	True Pareto	0.28396	-	0	-	-

Test 2 We expand the preference region in Test 1 to form this test (Figures 4.6a and 4.6d). We observe that the differences between their metric values are smaller compared to those in Test 1, since the preference region is larger. However, prTDEA's superiority over TDEA persists as seen in Tables 4.8 and 4.11. We present the plots of the results in Figure 4.6 for both algorithms.



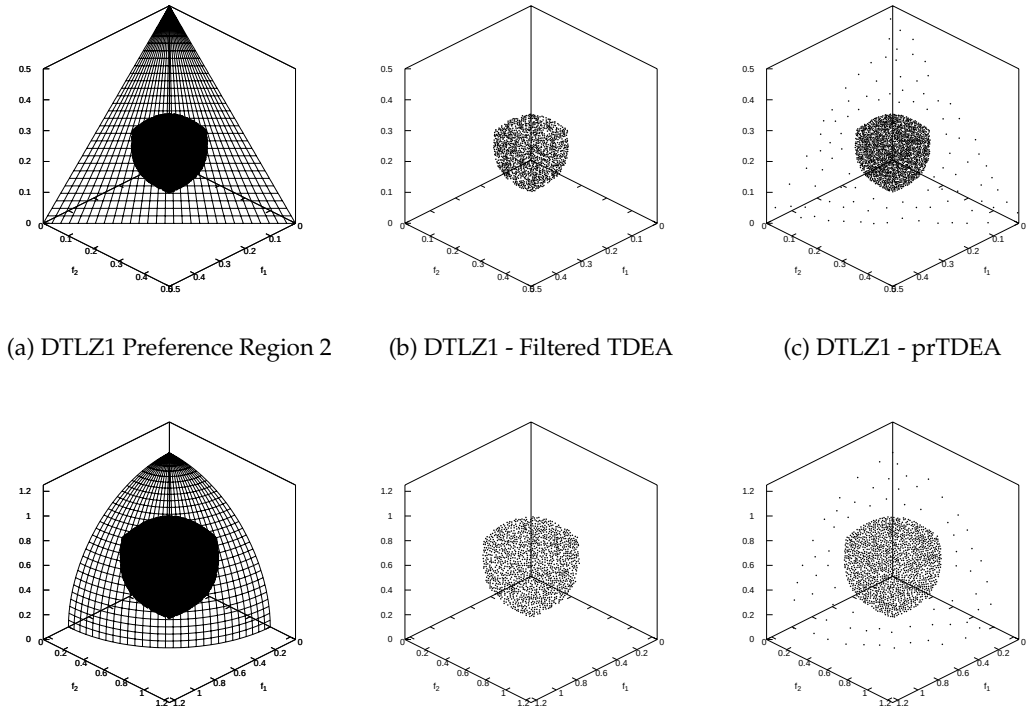
(a) DTLZ1 Preference Region 1 (b) DTLZ1 - Filtered TDEA (c) DTLZ1 - prTDEA

(d) DTLZ2 Preference Region 1 (e) DTLZ2 - Filtered TDEA (f) DTLZ2 - prTDEA

Figure 4.5: DTLZ1 and DTLZ2 Test 1 Plots

Table 4.8: Indicator Results for DTLZ1 and DTLZ2 Preference Test 2

		Hypervolume		**Inverted Gen. Distance**		
	Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
DTLZ1	prTDEA	0.39768	0.00219	0.000117	0.00000845	19.85
	TDEA	0.38809	0.00459	0.000166	0.00002160	205.23
	True Pareto	0.41386	-	0	-	-
DTLZ2	prTDEA	0.34523	0.000607	0.0000911	0.000000741	35.53
	TDEA	0.33943	0.000688	0.0001070	0.000001430	183.85
	True Pareto	0.36065	-	0	-	-



(a) DTLZ1 Preference Region 2 (b) DTLZ1 - Filtered TDEA (c) DTLZ1 - prTDEA

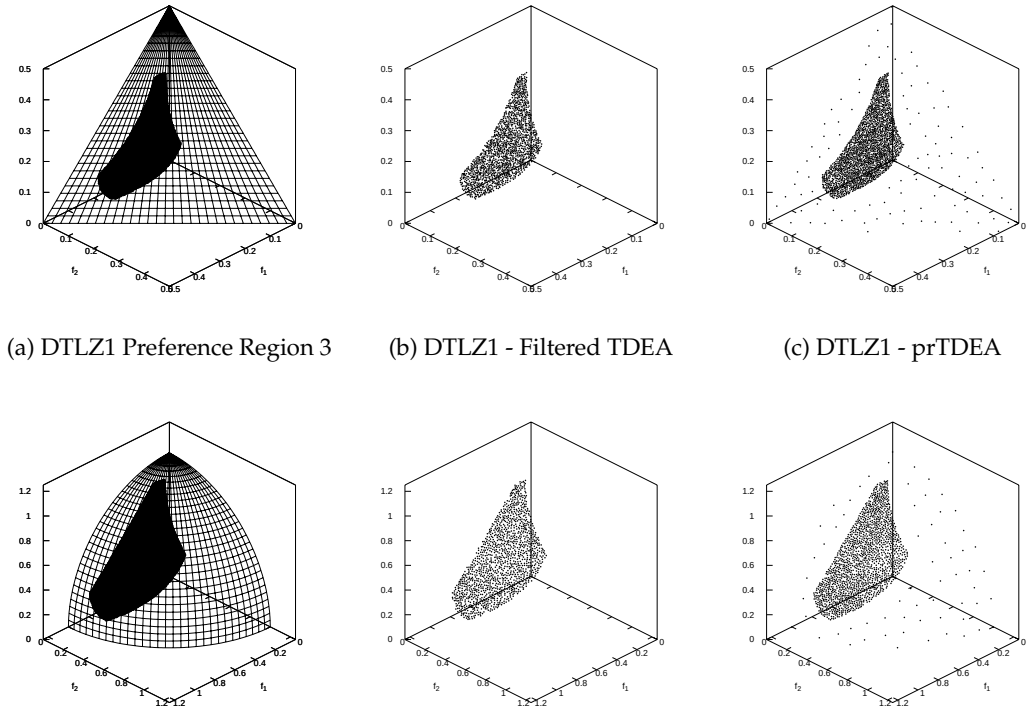
(d) DTLZ2 Preference Region 2 (e) DTLZ2 - Filtered TDEA (f) DTLZ2 - prTDEA

Figure 4.6: DTLZ1 and DTLZ2 Test 2 Plots

Test 3 Here we assign each objective different weight ranges and test whether this creates difficulties in focusing on a preferred region. prTDEA has no problems in finding better approximations than TDEA, with a noticeably shorter average computational time (Table 4.11). The statistical tests in Table 4.11 indicate significance. The difference can be also seen from the plots given in Figure 4.7.

Table 4.9: Indicator Results for DTLZ1 and DTLZ2 Preference Test 3

		Hypervolume		**Inverted Gen. Distance**		
	Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
DTLZ1	prTDEA	0.55521	0.00227	0.000095	0.00000568	19.71
	TDEA	0.54526	0.00410	0.000132	0.00001480	205.23
	True Pareto	0.56979	-	0	-	-
DTLZ2	prTDEA	0.35584	0.000452	0.0000851	0.000000865	31.59
	TDEA	0.34914	0.000540	0.0000997	0.000001230	183.85
	True Pareto	0.36842	-	0	-	-



(a) DTLZ1 Preference Region 3 (b) DTLZ1 - Filtered TDEA (c) DTLZ1 - prTDEA

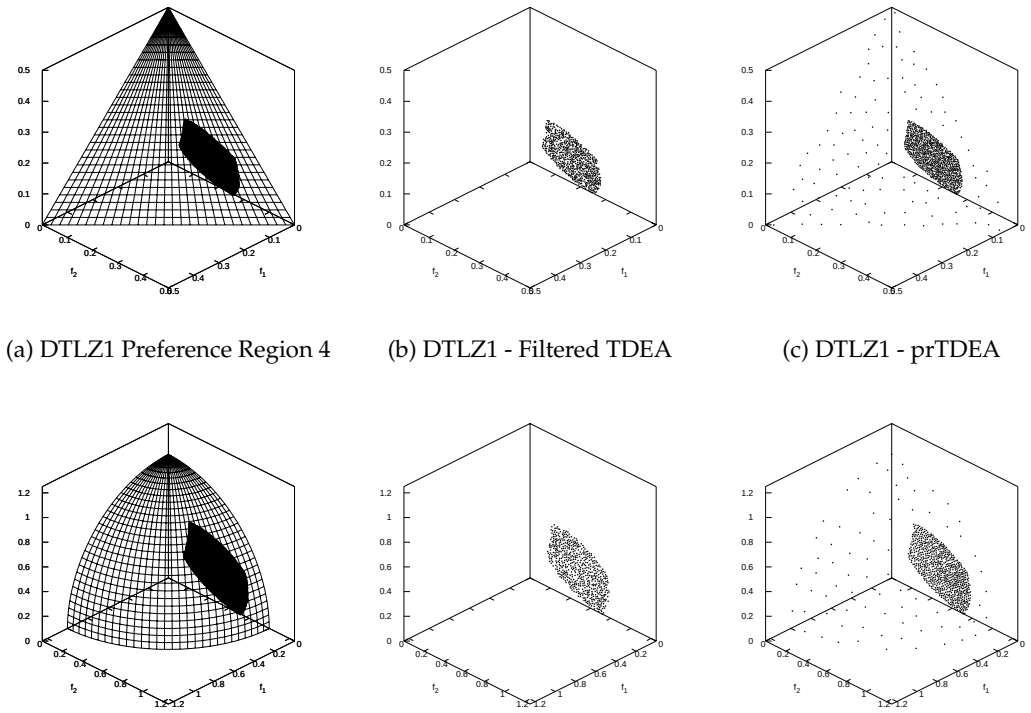
(d) DTLZ2 Preference Region 3 (e) DTLZ2 - Filtered TDEA (f) DTLZ2 - prTDEA

Figure 4.7: DTLZ1 and DTLZ2 Test 3 Plots

Test 4 The last test is similar to Test 3, but it features a smaller preference region as shown in Figures 4.8a and 4.8d. We expect that the difference between prTDEA and TDEA will increase in both metrics and average computational time, since the region is smaller than that of Test 3. The indicator results given in Table 4.10 confirm that expectation. We also present the plots in Figure 4.8 to illustrate the difference.

Table 4.10: Indicator Results for DTLZ1 and DTLZ2 Preference Test 4

		Hypervolume		**Inverted Gen. Distance**		
	Algorithm	$\bar{x}_H$	s_H	$\bar{x}_D$	s_D	**Duration (sec)**
DTLZ1	prTDEA	0.44510	0.00226	0.000198	0.00000524	11.55
	TDEA	0.42890	0.00588	0.000295	0.00003450	205.23
	True Pareto	0.46002	-	0	-	-
DTLZ2	prTDEA	0.33666	0.000747	0.000180	0.00000158	18.14
	TDEA	0.32416	0.001040	0.000231	0.00000354	183.85
	True Pareto	0.34909	-	0	-	-



(a) DTLZ1 Preference Region 4 (b) DTLZ1 - Filtered TDEA (c) DTLZ1 - prTDEA

(d) DTLZ2 Preference Region 4 (e) DTLZ2 - Filtered TDEA (f) DTLZ2 - prTDEA

Figure 4.8: DTLZ1 and DTLZ2 Test 4 Plots

Table 4.11: Statistical Tests for DTLZ1 and DTLZ2 Preference Tests

	$H_0 : \mu_{prTDEA} = \mu_{TDEA}$ vs $H_1 : \mu_{prTDEA} \neq \mu_{TDEA}$					
	Hypervolume			**Inverted Gen. Distance**		
Test	Δ_H	**P-Value**	**Winner**	Δ_D	**P-Value**	**Winner**
DTLZ1 - 1	0.019142	0	prTDEA	-0.000161	0	prTDEA
DTLZ1 - 2	0.009583	0	prTDEA	-0.000049	0	prTDEA
DTLZ1 - 3	0.009956	0	prTDEA	-0.000037	0	prTDEA
DTLZ1 - 4	0.016198	0	prTDEA	-0.000097	0	prTDEA
DTLZ2 - 1	0.011270	0	prTDEA	-0.000048	0	prTDEA
DTLZ2 - 2	0.005795	0	prTDEA	-0.000015	0	prTDEA
DTLZ2 - 3	0.006701	0	prTDEA	-0.000015	0	prTDEA
DTLZ2 - 4	0.012492	0	prTDEA	-0.000052	0	prTDEA

4.2.3 Discussions

The test results show that focusing on a preference region provides both a better approximation and a substantial computational time advantage in all settings. The positive effects of focusing are clearer when the region is smaller. In addition, computation time differences get larger when the number of objectives increase. This is because the number of solutions needed to properly approximate the Pareto-optimal frontier increases with the number of objectives. We can conclude that DM preferences should be exploited whenever possible. prTDEA's preference incorporation mechanism effectively serves this purpose. Note that it does not lose the rest of the Pareto-optimal frontier while concentrating on these regions. It displays an overview of the entire Pareto-optimal frontier, while presenting finely detailed solutions in the regions of interest.

4.3 Interactive Territory Defining Evolutionary Algorithm

In this section, we present the Interactive Territory Defining Evolutionary Algorithm. In Section 4.3.1, we describe the details of the algorithm. It is followed by simulation runs and comparisons in Section 4.3.2.

4.3.1 Details of the Algorithm

In the previous section, we showed that prTDEA works well when preference information is gathered from the DM. However, in some cases the DM may not know his/her regions of interest before the optimization stage and consequently may not be able to specify his/her preferences. In such a situation, it will not be easy to guide prTDEA. To overcome this difficulty, we propose another version of TDEA, the Interactive Territory Defining Evolutionary Algorithm (iTDEA). This interactive algorithm starts by finding solutions over all of the Pareto-optimal frontier that we denote as region R_0 with a low resolution (i.e small number of solutions per unit hypervolume). At generation G_1, it pauses for initiating an interaction with the DM. Here, the algorithm presents a sample of the solutions found so far to the DM and receives his/her most preferred solution among these. This solution is then used for estimating a preferred region R_1 to focus on. Subsequent interaction stages are scheduled at generations $G_2, G_3, \ldots, G_H$. At each interaction stage h, a new preferred region R_h is estimated. Note that the resolution of a preferred region is always higher than previously estimated preferred regions whereas its size is smaller. In this way, the

algorithm converges to the final preference region of the DM. In the end, a set of solutions from this region are presented to DM for the final selection. We give an outline of the algorithm below:

1. Ask the user to specify the regular population size $\bar{N}$, the maximum number of generations T, the number of interactions H, a starting τ_0 and a final τ_H for the final preference region.

2. Set iteration count $t = 0$ and interaction count $h = 1$. Set the first focus region R_0 to the entire Pareto-frontier. That is, set $[l_i^{(0)}, u_i^{(0)}] = [0, 1]$ for all $i = 1, 2, \ldots, p$. Determine generations $G_1, G_2, \ldots, G_H$ at which an interaction with the DM is scheduled.

3. Create $\bar{N}$ random solutions to fill $P(0)$. Copy the nondominated solutions in $P(0)$ into $A(0)$.

4. Set $t \leftarrow t + 1$. Choose a parent from each population $P(t)$ and $A(t)$. Recombine parents to create new offspring and apply mutation to it.

5. Test the offspring for acceptance into $P(t)$. If accepted, insert into $P(t)$ and go to the next step. Otherwise, go to Step 7.

6. Test the offspring for acceptance into $A(t)$. If accepted, insert into $A(t)$.

7. If $t = G_h$, then pause and interact with the DM. Find the new preferred region R_h and determine τ_h. Set $h \leftarrow h + 1$. Otherwise, go to next step.

8. If the iteration limit is hit, that is, $t = T$, then stop and report the archive population. Otherwise go to Step 4.

We previously defined a *preferred region* R_P by a set of Tchebycheff weights $\mathbf{W}^P$ in Section 4.1.2. In iTDEA, we change the definition. From now on, a *preferred region* R_P is defined by a set of Tchebycheff weight ranges $\left[\mathbf{l}^\mathbf{P}, \mathbf{u}^\mathbf{P}\right] = \left\{\left[l_1^P, u_1^P\right], \left[l_2^P, u_2^P\right], \ldots, \left[l_p^P, u_p^P\right]\right\}$.

iTDEA modifies the archive selection scheme as prTDEA does. Unlike prTDEA, there is a single preferred region at the beginning of the algorithm, that is, the entire Pareto-optimal frontier. Other preferred regions, $R_1, R_2, \ldots, R_H$, are estimated during the process, at interaction stages $1, 2, \ldots, H$. Each preferred region is smaller than the previous one. Once a new preferred region R_h is estimated, it is also assigned a new τ_h value. A newer preferred region always has a smaller τ value than the previous ones to ensure that new focus regions are emphasized more than the former regions. Here, the important issue

is the determination of these preferred regions and assigning τ values to them. For this purpose, we use the method in Steuer (1986, pp.446–450) at each interaction stage h.

Let H be the number of interactions, λ be the desired $[l_i, u_i]$ width of i^{th} objective in the final preference region, r be the reduction factor in each interaction stage and P be the number of solutions to be presented to the DM in each interaction stage. Then, the following can be used as a guide to set these parameters as in Steuer (1986, pp.446–450):

$$P \gtrapprox p \tag{4.4}$$

$$H \approx p \tag{4.5}$$

$$\tfrac{1}{2p} \lessapprox \lambda \lessapprox \tfrac{3}{2p} \tag{4.6}$$

$$\sqrt[k]{\tfrac{1}{P}} \lessapprox r \lessapprox \sqrt[H-1]{\lambda} \tag{4.7}$$

where p is the number of objectives. The algorithm starts with $h = 1$ and R_0 having $[l_i^0, u_i^0] = [0, 1]$ for $i = 1, 2, \ldots, p$. Suppose that DM chooses s^* as his/her best solution in interaction stage h. We compute the favorable weights $\mathbf{w}^*$ of the chosen solution s^* using Equation 4.1. Then, the next preferred region $R_h = [\mathbf{l^{(h)}}, \mathbf{u^{(h)}}]$ is determined using the following formula:

$$\left[l_i^{(h)}, u_i^{(h)}\right] = \begin{cases} \left[0, r^h\right] & \text{if } w_i^* - \frac{r^h}{2} \leq 0 \\ \left[1 - r^h, 0\right] & \text{if } w_i^* + \frac{r^h}{2} \geq 1 \\ \left[w_i^* - \frac{r^h}{2}, w_i^* + \frac{r^h}{2}\right] & \text{otherwise} \end{cases} \tag{4.8}$$

for all $i = 1, 2, \ldots, p$.

As seen in Equation 4.8, regions are shrunk faster in the first interaction stages. Towards the end, the amount of shrinkage gets smaller. A similar method is applied for assigning τ_h to R_h to control the number of solutions in the population. In the first interaction stages, τ is decreased faster and the decrease gets slower with each interaction stage. At interaction stage h, τ_h is found using the following formula:

$$\tau_h = \tau_H e^{(H-h)\rho} \tag{4.9}$$

where,

$$\rho = \frac{\ln \frac{\tau_0}{\tau_H}}{H} \tag{4.10}$$

Since it is hard for a DM to choose his/her best among all solutions, we choose a small subset of solutions to present. Here, the chosen solutions should represent the last focus region well. Otherwise, the solution selected by the DM may mislead the algorithm and

cause an inaccurate estimate for the next preferred region. For choosing solutions to present the DM, we use a filtering procedure that utilizes ϵ-dominance. In ϵ-dominance, a solution is considered to be nondominated if there is no solution that is at least an ϵ amount better than that solution in every objective and more than an ϵ amount better at least in one objective. The steps of the filtering procedure at interaction stage h are as follows:

1. Form the first filtered list F_h^1 by taking solutions only from region R_{h-1}. That is, $F_h^1 = \left\{ i \in A(G_h) | \mathbf{w}^i \in \left[\mathbf{l}^{(h-1)}, \mathbf{u}^{(h-1)} \right] \right\}$.

2. Test each pair of solutions $i, j \in F_h^1$ for ϵ-dominance using $\epsilon = \tau_{h-1}$. If solution i ϵ-dominates solution j and is not ϵ-dominated by j, remove j from F_h^1, and vice versa.

3. Calculate rectilinear distances d_{ij} between each pair of solutions $i, j \in F_h^1$.

4. Initialize the second filtered list F_h^2 by moving two solutions $k, l = \text{argmax}_{i,j \in F_h^1} \left(d_{ij} \right)$ from F_h^1 to F_h^2. That is, choose the pair of solutions that are most distant to each other and move them from F_h^1 to F_h^2.

5. Fill F_h^2 until its size is equal to P, each time moving solution $k = \text{argmax}_{i \in F_h^1} \left(\text{min}_{j \in F_h^2} \left(d_{ij} \right) \right)$ to F_h^2. That is, move the solution k in F_h^1 which maximizes the minimum distance to all solutions in F_h^2.

6. Present solutions in F_h^2 to the DM.

Note that we used $P = 2p$ for all interaction stages except the first one. In the first interaction stage and the final presentation, we set $P = 4p$.

The archive acceptance procedure of iTDEA is almost the same as prTDEA. It starts with a dominance check. If any solution in the archive dominates offspring c, then c is rejected and the process terminates. Otherwise, we start the second stage of the archive acceptance by removing all solutions dominated by c. Then, we determine the preferred region that contains c. For this purpose, we compute the favorable weights, $\mathbf{w}^c$, of c and determine the regions whose weight ranges contain these weights. Among these regions, we find the one having the smallest τ_i to set $\tau = \tau_i$. As in prTDEA, the rest of the process is the same as in TDEA. First, the closest solution s_{i^*} to c with respect to rectilinear distance is determined, using the scaled objective function values. Then, we check whether s_{i^*} is in the territory of c, which is defined as the region within τ distance in all objective values of c. c is rejected if the maximum scaled objective distance between s_{i^*} and c is smaller than the τ value. Otherwise, it is accepted. We give the details of the procedure below:

1. Test offspring c with each solution $s_i \in A(t)$ for dominance. Mark solutions dominated by c. If c is dominated by at least one s_i, reject c. Otherwise, go to next step.

2. Remove all marked solutions from $A(t)$.

3. If $A(t)$ is empty, accept c and insert it into $A(t)$. Otherwise, go to next step.

4. Calculate the favorable weights $\mathbf{w}^c$ of c with Equation 4.1 using scaled objective values.

5. Find the preferred regions whose weight ranges contain the favorable weights of c. Then find the region j with the smallest τ_i to set $\tau = \tau_j$. That is, set $\tau = \min_{j=0}^{h}\{\tau_i | \mathbf{w}^c \in \left[\mathbf{l^j}, \mathbf{u^j}\right] \quad i = 1, 2, \ldots, h\}$.

6. Calculate the rectilinear distance $d_{ci} = \sum_{j=1}^{p} \left|\hat{f}_{cj} - \hat{f}_{ij}\right|$ of c to each solution $s_i \in A(t)$ using the scaled objective function values.

7. Find $i^* = \text{argmin}_i(d_i)$, that is, the solution s_{i^*} closest to c.

8. Find maximum scaled absolute objective difference between c and s_{i^*}. That is, find

$$\delta = \max_{j=1,2,\ldots,p} \left|\hat{f_{cj}} - \hat{f_{i^*j}}\right| \tag{4.11}$$

   where $\hat{f_{cj}}$ and $\hat{f_{i^*j}}$ are the scaled j^{th} objective values of offspring c and solution s_{i^*}, respectively.

9. Accept c if $\delta \geq \tau$ and insert into $A(t)$. Otherwise, reject c.

In iTDEA, scheduling the generations $G_1, G_2, \ldots, G_H$ at which an interaction will occur is important for the correct functioning of the algorithm. The weight mechanism works well if the population is converged to the Pareto-optimal frontier. Otherwise, the selected solution's weights may mislead the algorithm. Hence, first interaction should be set at a generation where the population is considered to converge. Selection of the generation for the last interaction is also important. Since the final preference region is found, the algorithm should be allowed to spend enough search effort to concentrate and converge well to that region.

4.3.2 Simulation Runs and Comparisons

We conduct our simulation runs on test problems ZDT4, DTLZ1 and DTLZ2. We simulate the DM's preferences using a Tchebycheff utility function of the following form:

$$U = \underset{z \in \mathbf{Z}}{\text{Min}} \max_{i=1,2,\ldots,p} \left[w_i \left|f_i^* - f_i\right|\right] \tag{4.12}$$

where f_i is the i^{th} objective value, f_i^* is the i^{th} element of the ideal vector and w_i is the weight corresponding to i^{th} objective. For each problem, we choose three utility functions. To observe the effects of the number of interactions with the DM, we try 4 and 6 interactions. Each run is replicated 50 times. The details of the simulation runs are given in Table 4.12.

To see how filtering affects the results of the algorithm, we run each instance with filtering and no filtering. In the unfiltered mode, we assume that the DM chooses his/her best solution among all solutions. In the end, we choose the solution (denoted as "No Filter") with the best utility to report. In the filtering case, we filter P solutions ($2P$ in the first interaction) to present to the DM at each interaction. At the end of the run, we report two solutions. The first one (denoted as "Filter 1") is the best utility solution among all generated solutions, whereas the second (denoted as "Filter 2") is again chosen among $2P$ filtered solutions from the final preference region.

We calculate the absolute deviation (Δ) and percentage deviation ($\bar{\Delta}$) values as follows:

$$\Delta = U^* - U \tag{4.13}$$

$$\bar{\Delta} = \frac{U^* - U}{U^* - U^n} \tag{4.14}$$

where U is the utility value of the solution reported by the algorithm, U^* is the true optimal solution's utility value and U^n is the utility value of the worst solution among all nondominated solutions.

The first interaction with the decision maker is scheduled at the generation when the $\frac{1}{3}$ of the generations are completed. The algorithm is also allowed to run $\frac{1}{6}$ of the generations after the final interaction. The rest of the generations are uniformly distributed to each interaction.

2-Objective Tests

Test 1: In the first test, the best solution is at the middle of the Pareto-optimal frontier. Table 4.13 shows the results found for all cases. We observe that more interactions with the DM allows better convergence to the desired locations. In addition, it can be seen that iTDEA successfully converges in both filtered and unfiltered cases. However, deviation from the optimal solution increases when filtering is applied to the final population. Figure 4.9 displays the plots of the runs. Note that the final regions of 6-interaction cases are smaller.

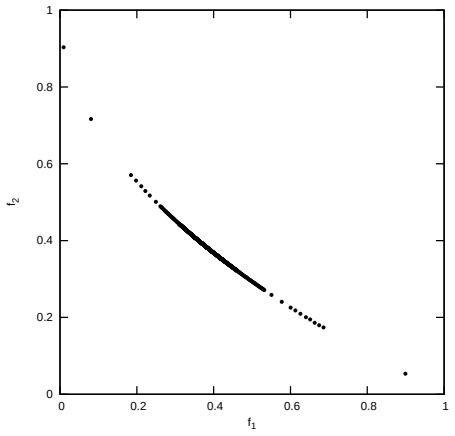
Test 2: In the second test, the utility function favors the second objective more than the first objective. It tests whether such a bias affects the convergence. As seen in Table 4.14,

Table 4.12: Interactive Test Parameters

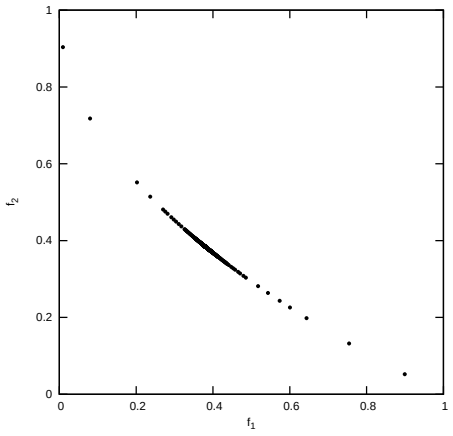
	ZDT4	**DTLZ1**	**DTLZ2**
Test 1 Weights, w	$(0.5, 0.5)$	$(0.33, 0.33, 0.33)$	$(0.33, 0.33, 0.33)$
Test 2 Weights, w	$(0.2, 0.8)$	$(0.2, 0.3, 0.5)$	$(0.2, 0.3, 0.5)$
Test 3 Weights, w	$(0.65, 0.35)$	$(0.7, 0.2, 0.1)$	$(0.7, 0.2, 0.1)$
Ideal Vector, f*	$(0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
Interactions	4, 6	4, 6	4, 6
Population Size	200	400	400
τ_0	0.1	0.1	0.1
τ_H	0.00001	0.005	0.005
Function Evaluations	80000	320000	320000
Replications	50	50	50

Table 4.13: ZDT4 Interactive Test 1 Results

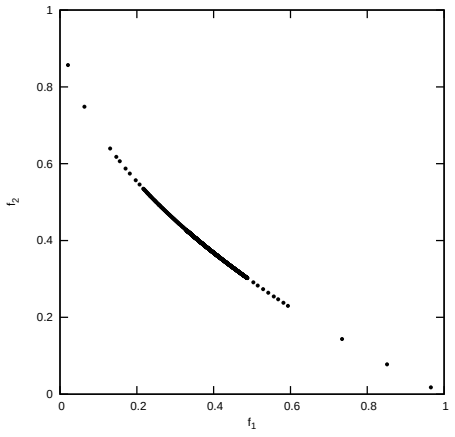
Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.19115	0.000132	0.19098	0.500	0.00017	0.0540%
Filter 1	4	0.19114	0.000100	0.19098	0.500	0.00016	0.0508%
Filter 2	4	0.19281	0.001290	0.19098	0.500	0.00183	0.5912%
No Filter	6	0.19111	0.000099	0.19098	0.500	0.00013	0.0411%
Filter 1	6	0.19110	0.000080	0.19098	0.500	0.00012	0.0379%
Filter 2	6	0.19143	0.000239	0.19098	0.500	0.00045	0.1446%



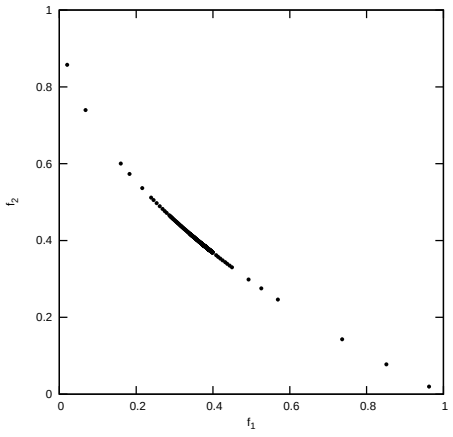
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions



(c) Filtered with 4 Interactions



(d) Filtered with 6 Interactions

Figure 4.9: ZDT4 Interactive Test 1 Plots

the results are very similar to those of Test 1. iTDEA with 6 interactions outperform the one with 4 interactions. No filter and Filter 1 cases perform very close to each other. We present the plots of the runs in Figure 4.10.

Table 4.14: ZDT4 Interactive Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.13739	0.000092	0.13726	0.800	0.00013	0.0199%
Filter 1	4	0.13740	0.000100	0.13726	0.800	0.00014	0.0214%
Filter 2	4	0.13894	0.001100	0.13726	0.800	0.00168	0.2537%
No Filter	6	0.13734	0.000057	0.13726	0.800	0.00008	0.0123%
Filter 1	6	0.13736	0.000070	0.13726	0.800	0.00010	0.0153%
Filter 2	6	0.13756	0.000197	0.13726	0.800	0.00030	0.0455%

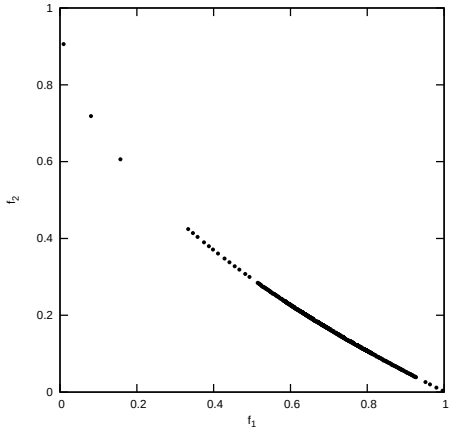
Test 3: In test 3, we reverse the bias to the first objective's side. Interestingly, the algorithm performs worse than the previous test in the filtered case with final population filtering (Table 4.15), although solutions are still good in the absolute sense. Note that standard deviation is highest in the Filter 2, which shows that the final preference region has the largest variation in this case. This is significantly reduced by two additional interactions. Figure 4.11 displays the plots of of the final population of well-guided runs.

Table 4.15: ZDT4 Interactive Test 3 Results

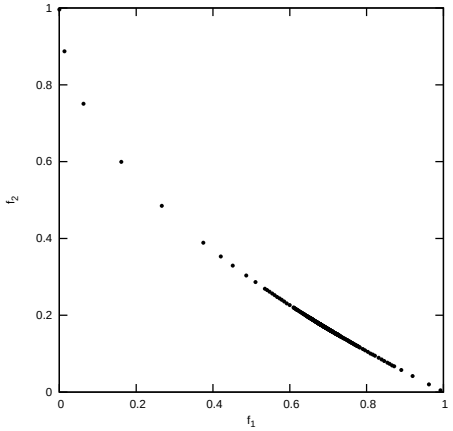
Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.17081	0.000095	0.17066	0.650	0.00015	0.0313%
Filter 1	4	0.17083	0.000121	0.17066	0.650	0.00017	0.0355%
Filter 2	4	0.17215	0.001050	0.17066	0.650	0.00149	0.3108%
No Filter	6	0.17076	0.000063	0.17066	0.650	0.00010	0.0209%
Filter 1	6	0.17075	0.000050	0.17066	0.650	0.00009	0.0188%
Filter 2	6	0.17105	0.000199	0.17066	0.650	0.00039	0.0814%

3-Objective Tests

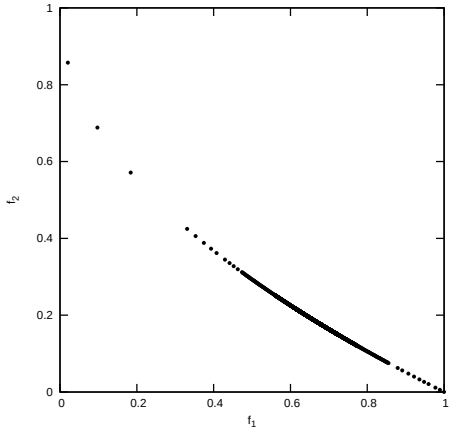
We use problems DTLZ1 and DTLZ2 for 3-objective tests. The Pareto-optimal frontiers of these two problems have different shapes and we test whether this creates difficulty for the algorithm.



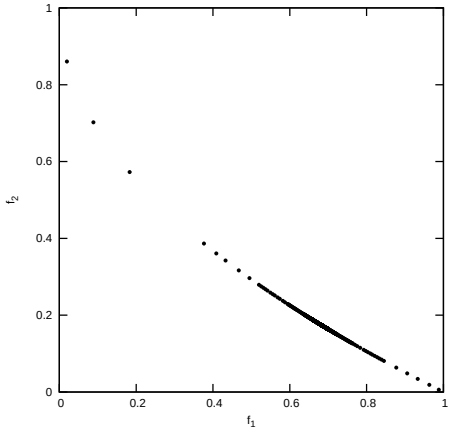
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions

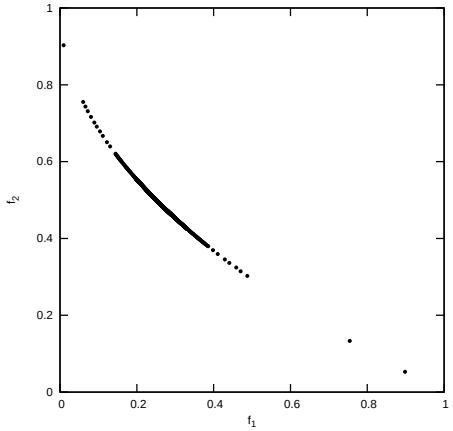


(c) Filtered with 4 Interactions

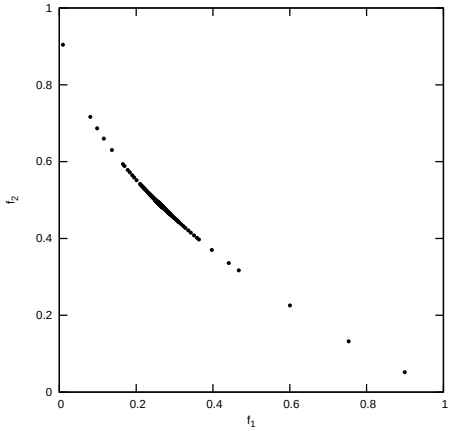


(d) Filtered with 6 Interactions

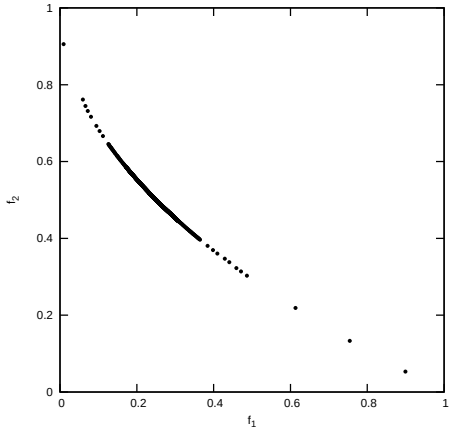
Figure 4.10: ZDT4 Interactive Test 2 Plots



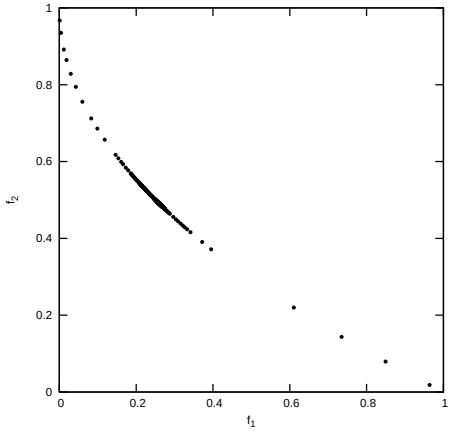
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions



(c) Filtered with 4 Interactions



(d) Filtered with 6 Interactions

Figure 4.11: ZDT4 Interactive Test 3 Plots

Test 1: In both DTLZ1 and DTLZ2 tests (Tables 4.16 and 4.17), we observe that the differences between filtered and unfiltered cases are larger than those of ZDT4 tests. The standard deviation of filtered cases are also higher, which means that in some runs the results are worse, whereas in other runs better results are obtained. We also observe that although the algorithm finds good solutions in the filtered case, the final filtering may cause worse solutions to be reported. Note that although the performances are better in 2-objective cases, the algorithm still obtains good solutions here in the absolute sense. In the plots (Figures 4.12 and 4.12) we observe the progress of the preferred regions. In the unfiltered cases, the algorithm finely advances to the desired final preference region. However, in the filtered cases, the solution chosen by the DM misleads the algorithm a little bit in the first interaction stages. This is somewhat prevented when the number of interaction stages is increased from 4 to 6.

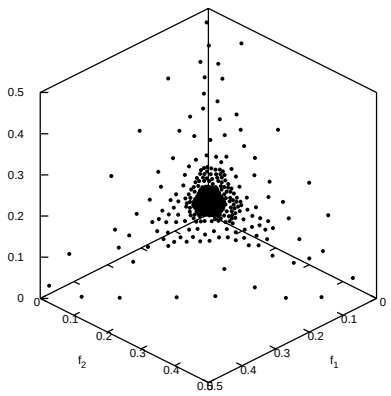
Table 4.16: DTLZ1 Interactive Test 1 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.05539	0.000161	0.05500	0.165	0.00039	0.3573%
Filter 1	4	0.05591	0.001221	0.05500	0.165	0.00091	0.8291%
Filter 2	4	0.05793	0.002646	0.05500	0.165	0.00293	2.6618%
No Filter	6	0.05536	0.000149	0.05500	0.165	0.00036	0.3273%
Filter 1	6	0.05593	0.001380	0.05500	0.165	0.00093	0.8455%
Filter 2	6	0.05619	0.001669	0.05500	0.165	0.00119	1.0791%

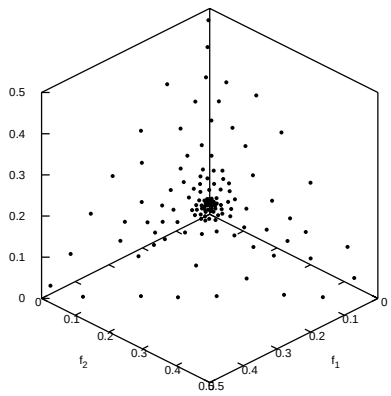
Table 4.17: DTLZ2 Interactive Test 1 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.19136	0.000376	0.19053	0.330	0.00083	0.5983%
Filter 1	4	0.19154	0.000961	0.19053	0.330	0.00101	0.7273%
Filter 2	4	0.19757	0.006420	0.19053	0.330	0.00704	5.0507%
No Filter	6	0.19137	0.000346	0.19053	0.330	0.00084	0.6054%
Filter 1	6	0.19226	0.001880	0.19053	0.330	0.00173	1.2435%
Filter 2	6	0.19391	0.003770	0.19053	0.330	0.00338	2.4265%

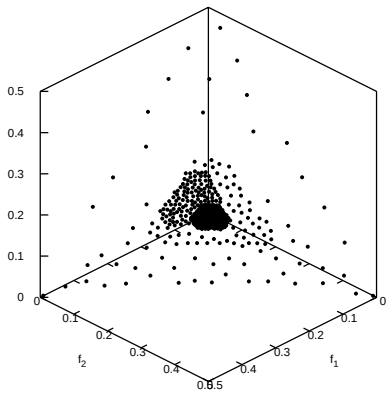
Test 2: In this test, the second objective is favored more than the other two objectives. Incorporating such a bias to the utility function does not affect the results, as they are very



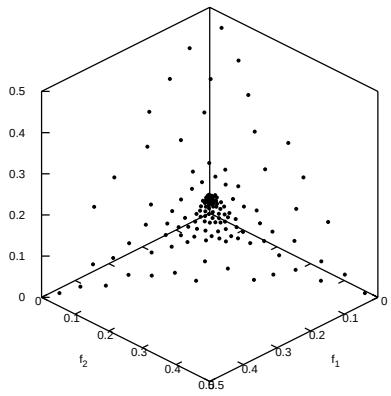
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions

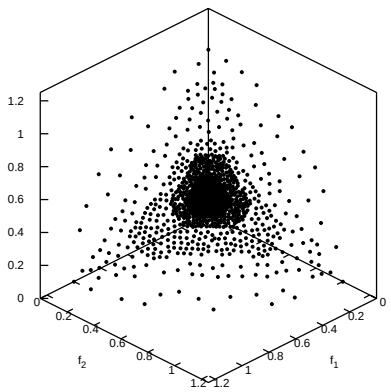


(c) Filtered with 4 Interactions

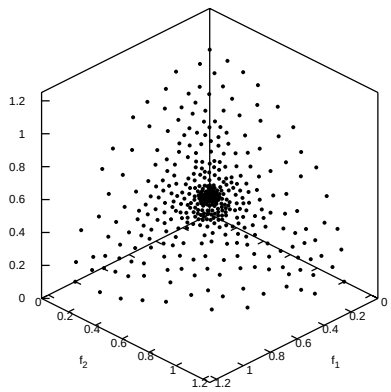


(d) Filtered with 6 Interactions

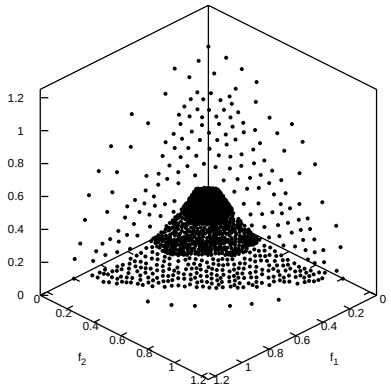
Figure 4.12: DTLZ1 Interactive Test 1 Plots



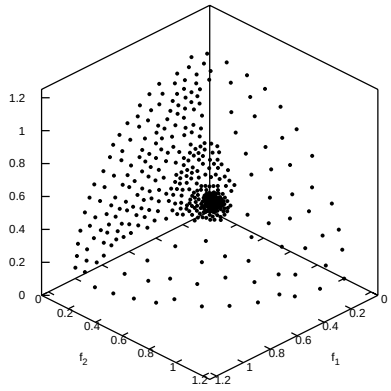
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions



(c) Filtered with 4 Interactions



(d) Filtered with 6 Interactions

Figure 4.13: DTLZ2 Interactive Test 1 Plots

similar to those found in Test 2 (Tables 4.18 and 4.19). One unusual observation is that Filter 2 results with 4 interactions are better than those with 6 interactions in problem DTLZ2 (Table 4.19). This indicates that the algorithm converges to an incorrect region after the last interaction in some runs. Since the final region is larger with 4 interactions, this does not affect it as severely as it does the 6-interaction case. We observe similar patterns to Test 1 in the plots given in Figures 4.14 and 4.15.

Table 4.18: DTLZ1 Interactive Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.04870	0.000154	0.04839	0.250	0.00031	0.1537%
Filter 1	4	0.04960	0.001203	0.04839	0.250	0.00122	0.6031%
Filter 2	4	0.05257	0.003496	0.04839	0.250	0.00418	2.0737%
No Filter	6	0.04872	0.000125	0.04839	0.250	0.00033	0.1641%
Filter 1	6	0.04961	0.001220	0.04839	0.250	0.00122	0.6046%
Filter 2	6	0.05091	0.003194	0.04839	0.250	0.00252	1.2499%

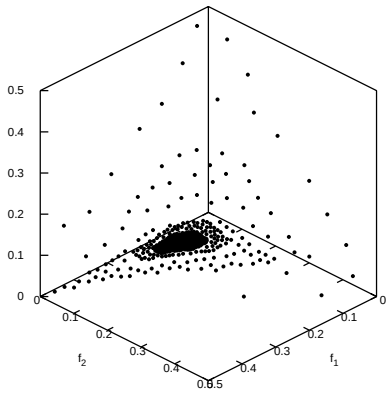
Table 4.19: DTLZ2 Interactive Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.15839	0.000190	0.15789	0.500	0.00050	0.1448%
Filter 1	4	0.15998	0.001790	0.15789	0.500	0.00209	0.6095%
Filter 2	4	0.16568	0.004380	0.15789	0.500	0.00779	2.2757%
No Filter	6	0.15835	0.000200	0.15789	0.500	0.00046	0.1331%
Filter 1	6	0.16078	0.002400	0.15789	0.500	0.00289	0.8434%
Filter 2	6	0.16456	0.004830	0.15789	0.500	0.00667	1.9483%

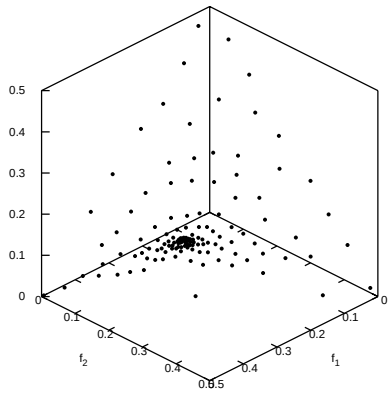
Test 3: In Test 3, the third objective has the greatest bias. As can be seen from Tables 4.20 and 4.21, iTDEA shows its best 3-objective performances in this test. We still observe the same issues in the filtered case without final population filtering. Also, the patterns in the plots (Figures 4.16 and 4.17) are similar to the previous 3-objective tests.

Other Utility Functions

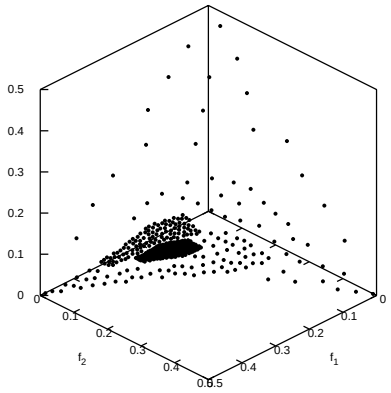
In order to inspect the effects of different utility functions, we repeat the same tests using linear and quadratic utility functions. Their formal definitions are given in Equations 4.15



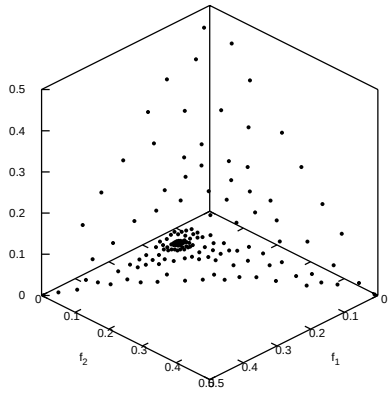
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions



(c) Filtered with 4 Interactions



(d) Filtered with 6 Interactions

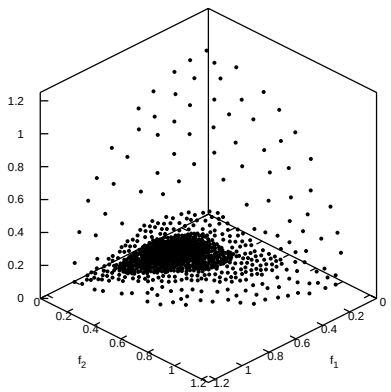
Figure 4.14: DTLZ1 Interactive Test 2 Plots

Table 4.20: DTLZ1 Interactive Test 3 Results

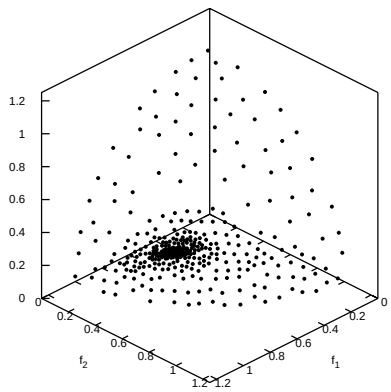
Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.03062	0.000080	0.03043	0.350	0.00019	0.0592%
Filter 1	4	0.03112	0.000780	0.03043	0.350	0.00068	0.2135%
Filter 2	4	0.03337	0.002389	0.03043	0.350	0.00293	0.9169%
No Filter	6	0.03080	0.000324	0.03043	0.350	0.00037	0.1155%
Filter 1	6	0.03146	0.000953	0.03043	0.350	0.00102	0.3199%
Filter 2	6	0.03307	0.002540	0.03043	0.350	0.00264	0.8256%

Table 4.21: DTLZ2 Interactive Test 3 Results

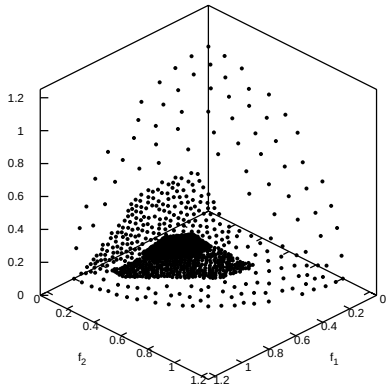
Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev.	Rel. Dev.
No Filter	4	0.08895	0.000148	0.08872	0.700	0.00022	0.0366%
Filter 1	4	0.09005	0.001271	0.08872	0.700	0.00132	0.2165%
Filter 2	4	0.09504	0.003928	0.08872	0.700	0.00632	1.0335%
No Filter	6	0.08901	0.000197	0.08872	0.700	0.00029	0.0479%
Filter 1	6	0.08969	0.000911	0.08872	0.700	0.00097	0.1588%
Filter 2	6	0.09267	0.004392	0.08872	0.700	0.00395	0.6461%



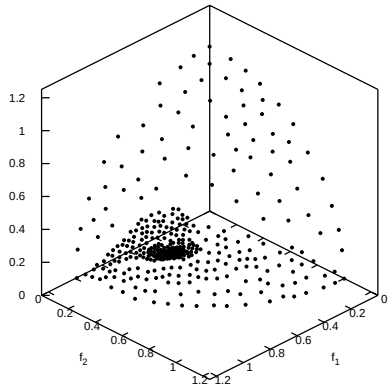
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions

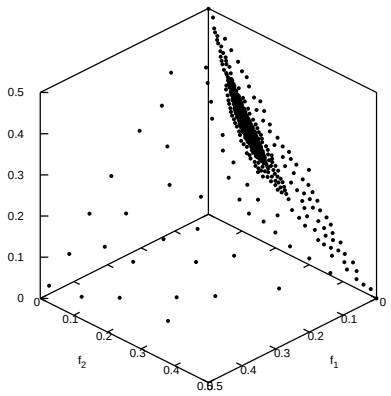


(c) Filtered with 4 Interactions

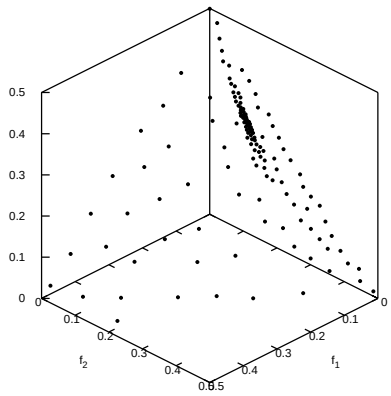


(d) Filtered with 6 Interactions

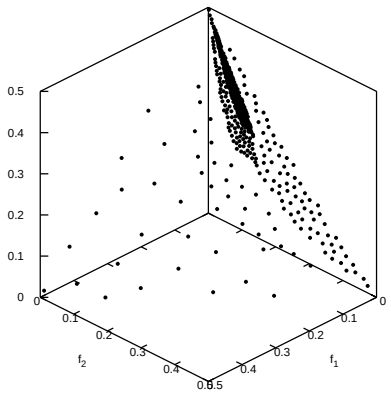
Figure 4.15: DTLZ2 Interactive Test 2 Plots



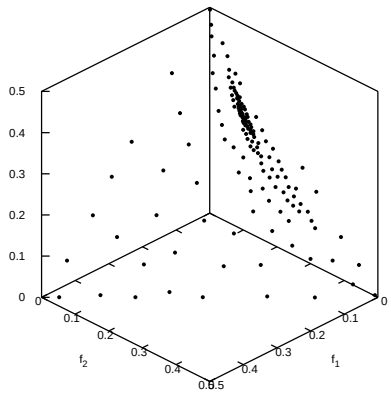
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions

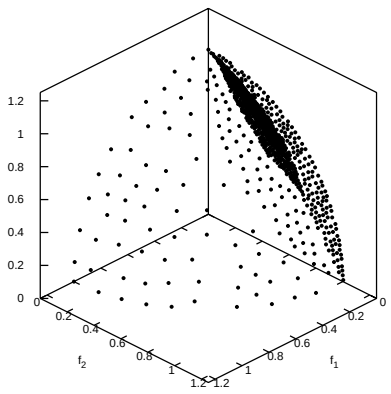


(c) Filtered with 4 Interactions

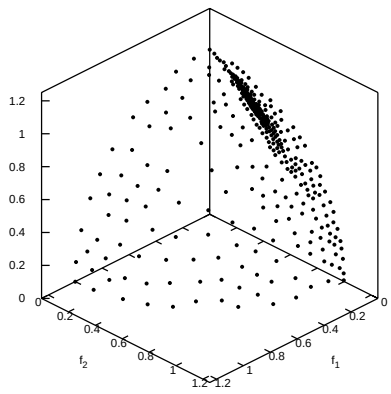


(d) Filtered with 6 Interactions

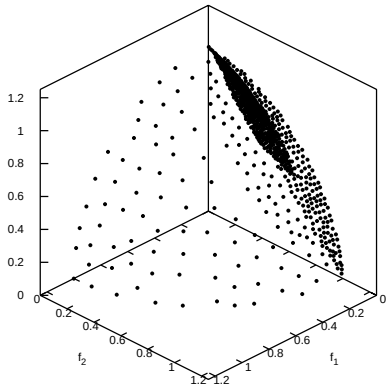
Figure 4.16: DTLZ1 Interactive Test 3 Plots



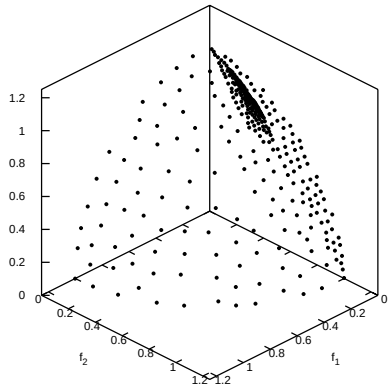
(a) Unfiltered with 4 Interactions



(b) Unfiltered with 6 Interactions



(c) Filtered with 4 Interactions



(d) Filtered with 6 Interactions

Figure 4.17: DTLZ2 Interactive Test 3 Plots

and 4.16, respectively:

$$U = \underset{z \in \mathbf{Z}}{\text{Min}} \sum_{i=1,2,\ldots,p} w_i \left| f_i^* - f_i \right| \tag{4.15}$$

$$U = \underset{z \in \mathbf{Z}}{\text{Min}} \sqrt{\sum_{i=1,2,\ldots,p} \left[w_i \left(f_i^* - f_i \right) \right]^2} \tag{4.16}$$

where f_i is the i^{th} objective value, f_i^* is the i^{th} element of the ideal vector and w_i is the weight corresponding to i^{th} objective.

Since DTLZ1 has a linear Pareto-optimal frontier, all nondominated solutions are optimal for linear utility functions having equal weights in all objectives. Similarly, due to DTLZ2's spherical Pareto-optimal frontier, all nondominated solutions in DTLZ2 are optimal for quadratic utility functions that have equal weights in all objectives. For these reasons, we omit those tests.

In Tables 4.22, 4.23 and 4.24, we present the results of 2-objective tests with linear utility function. It can be observed that iTDEA successfully converges to the final preference region of the DM. The results show that relative deviations from optimal solutions are less than 0.26%.

Table 4.22: ZDT4 Linear Utility Function Test 1 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.37520	0.000126	0.37500	0.500	0.00020	0.16000%
Filter 1	4	0.37523	0.000189	0.37500	0.500	0.00023	0.18400%
Filter 2	4	0.37533	0.000227	0.37500	0.500	0.00033	0.26400%
No Filter	6	0.37512	0.000096	0.37500	0.500	0.00012	0.09600%
Filter 1	6	0.37514	0.000075	0.37500	0.500	0.00014	0.11200%
Filter 2	6	0.37517	0.000089	0.37500	0.500	0.00017	0.13600%

Table 4.23: ZDT4 Linear Utility Function Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.20042	0.000364	0.20000	0.800	0.00042	0.07000%
Filter 1	4	0.20042	0.000365	0.20000	0.800	0.00042	0.07000%
Filter 2	4	0.20043	0.000379	0.20000	0.800	0.00043	0.07167%
No Filter	6	0.20025	0.000196	0.20000	0.800	0.00025	0.04167%
Filter 1	6	0.20026	0.000202	0.20000	0.800	0.00026	0.04333%
Filter 2	6	0.20026	0.000203	0.20000	0.800	0.00026	0.04333%

Table 4.24: ZDT4 Linear Utility Function Test 3 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.30309	0.000152	0.30300	0.650	0.00009	0.02594%
Filter 1	4	0.30306	0.000111	0.30300	0.650	0.00006	0.01729%
Filter 2	4	0.30320	0.000162	0.30300	0.650	0.00020	0.05764%
No Filter	6	0.30303	0.000111	0.30300	0.650	0.00003	0.00865%
Filter 1	6	0.30302	0.000103	0.30300	0.650	0.00002	0.00576%
Filter 2	6	0.30307	0.000147	0.30300	0.650	0.00007	0.02017%

In the following five tables (Tables 4.25, 4.26, 4.27, 4.28 and 4.29, the results of 3-objective tests with linear utility function are shown. Although the relative deviations from the optimal solutions are higher than 2-objective case, the algorithm still finds good solutions in the absolute sense. Note that the performance of iTDEA is better when linear utility function is used instead of Tchebycheff utility function.

Table 4.25: DTLZ1 Linear Utility Function Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.10031	0.000350	0.10000	0.25000	0.00031	0.20667%
Filter 1	4	0.10043	0.000397	0.10000	0.25000	0.00043	0.28667%
Filter 2	4	0.10154	0.003110	0.10000	0.25000	0.00154	1.02667%
No Filter	6	0.10037	0.000334	0.10000	0.25000	0.00037	0.24667%
Filter 1	6	0.10042	0.000355	0.10000	0.25000	0.00042	0.28000%
Filter 2	6	0.10113	0.001520	0.10000	0.25000	0.00113	0.75333%

Table 4.26: DTLZ1 Linear Utility Function Test 3 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.05016	0.000231	0.05000	0.35000	0.00016	0.05367%
Filter 1	4	0.05014	0.000209	0.05000	0.35000	0.00014	0.04600%
Filter 2	4	0.05052	0.002293	0.05000	0.35000	0.00052	0.17400%
No Filter	6	0.05017	0.000233	0.05000	0.35000	0.00017	0.05567%
Filter 1	6	0.05014	0.000191	0.05000	0.35000	0.00014	0.04567%
Filter 2	6	0.05111	0.003978	0.05000	0.35000	0.00111	0.36933%

The following three tables (Tables 4.30, 4.31 and 4.32) show the performance of iTDEA on 2-objective tests with quadratic utility function. They are followed by the results of

Table 4.27: DTLZ2 Linear Utility Function Test 1 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.33002	0.000027	0.33000	0.572	0.00002	0.00826%
Filter 1	4	0.33003	0.000109	0.33000	0.572	0.00003	0.01240%
Filter 2	4	0.33112	0.003100	0.33000	0.572	0.00112	0.46281%
No Filter	6	0.33005	0.000133	0.33000	0.572	0.00005	0.02066%
Filter 1	6	0.33004	0.000127	0.33000	0.572	0.00004	0.01653%
Filter 2	6	0.33255	0.005770	0.33000	0.572	0.00255	1.05372%

Table 4.28: DTLZ2 Linear Utility Function Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.20100	0.001540	0.20000	0.617	0.00100	0.23981%
Filter 1	4	0.20283	0.004480	0.20000	0.617	0.00283	0.67866%
Filter 2	4	0.21504	0.030630	0.20000	0.617	0.01504	3.60671%
No Filter	6	0.20175	0.002950	0.20000	0.617	0.00175	0.41966%
Filter 1	6	0.20327	0.005300	0.20000	0.617	0.00327	0.78417%
Filter 2	6	0.21611	0.030270	0.20000	0.617	0.01611	3.86331%

Table 4.29: DTLZ2 Linear Utility Function Test 3 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.10038	0.000968	0.10000	0.735	0.00038	0.05984%
Filter 1	4	0.10070	0.002060	0.10000	0.735	0.00070	0.11024%
Filter 2	4	0.10269	0.014120	0.10000	0.735	0.00269	0.42362%
No Filter	6	0.10046	0.000975	0.10000	0.735	0.00046	0.07244%
Filter 1	6	0.10040	0.000837	0.10000	0.735	0.00040	0.06299%
Filter 2	6	0.10293	0.015870	0.10000	0.735	0.00293	0.46142%

3-objective tests in Tables 4.33, 4.34, 4.35, 4.36 and 4.37. We obtain similar results to linear utility function case in these tests. However, the algorithm performs better in quadratic utility function than those in the other two utility functions.

Table 4.30: ZDT4 Quadratic Utility Function Test 1 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.26907	0.000101	0.26892	0.500	0.00015	0.06491%
Filter 1	4	0.26908	0.000126	0.26892	0.500	0.00016	0.06924%
Filter 2	4	0.26914	0.000152	0.26892	0.500	0.00022	0.09521%
No Filter	6	0.26901	0.000061	0.26892	0.500	0.00009	0.03895%
Filter 1	6	0.26903	0.000078	0.26892	0.500	0.00011	0.04760%
Filter 2	6	0.26906	0.000095	0.26892	0.500	0.00014	0.06059%

Table 4.31: ZDT4 Quadratic Utility Function Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.18071	0.000151	0.18057	0.800	0.00014	0.02260%
Filter 1	4	0.18070	0.000126	0.18057	0.800	0.00013	0.02099%
Filter 2	4	0.18078	0.000173	0.18057	0.800	0.00021	0.03390%
No Filter	6	0.18066	0.000081	0.18057	0.800	0.00009	0.01453%
Filter 1	6	0.18065	0.000062	0.18057	0.800	0.00008	0.01292%
Filter 2	6	0.18067	0.000068	0.18057	0.800	0.00010	0.01614%

Table 4.32: ZDT4 Quadratic Utility Function Test 3 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.23301	0.000098	0.23288	0.650	0.00013	0.03117%
Filter 1	4	0.23300	0.000100	0.23288	0.650	0.00012	0.02877%
Filter 2	4	0.23306	0.000147	0.23288	0.650	0.00018	0.04315%
No Filter	6	0.23297	0.000055	0.23288	0.650	0.00009	0.02158%
Filter 1	6	0.23298	0.000071	0.23288	0.650	0.00010	0.02397%
Filter 2	6	0.23300	0.000087	0.23288	0.650	0.00012	0.02877%

Table 4.33: DTLZ1 Quadratic Utility Function Test 1 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.09527	0.000011	0.09526	0.165	0.00001	0.00946%
Filter 1	4	0.09528	0.000068	0.09526	0.165	0.00002	0.02667%
Filter 2	4	0.09532	0.000074	0.09526	0.165	0.00006	0.08116%
No Filter	6	0.09527	0.000004	0.09526	0.165	0.00000	0.00373%
Filter 1	6	0.09527	0.000019	0.09526	0.165	0.00001	0.00803%
Filter 2	6	0.09527	0.000019	0.09526	0.165	0.00001	0.01233%

Table 4.34: DTLZ1 Quadratic Utility Function Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.07895	0.000006	0.07895	0.250	0.00001	0.00327%
Filter 1	4	0.07899	0.000070	0.07895	0.250	0.00004	0.02374%
Filter 2	4	0.07945	0.000906	0.07895	0.250	0.00051	0.29558%
No Filter	6	0.07895	0.000005	0.07895	0.250	0.00000	0.00269%
Filter 1	6	0.07902	0.000107	0.07895	0.250	0.00008	0.04478%
Filter 2	6	0.07926	0.000589	0.07895	0.250	0.00031	0.18158%

Table 4.35: DTLZ1 Quadratic Utility Function Test 3 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.04437	0.000008	0.04436	0.350	0.00001	0.00236%
Filter 1	4	0.04460	0.000538	0.04436	0.350	0.00024	0.07859%
Filter 2	4	0.04570	0.001675	0.04436	0.350	0.00134	0.43915%
No Filter	6	0.04437	0.000017	0.04436	0.350	0.00001	0.00334%
Filter 1	6	0.04462	0.000468	0.04436	0.350	0.00026	0.08546%
Filter 2	6	0.04560	0.001549	0.04436	0.350	0.00124	0.40479%

Table 4.36: DTLZ2 Quadratic Utility Function Test 2 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.20001	0.000046	0.20000	0.500	0.00001	0.00333%
Filter 1	4	0.20004	0.000083	0.20000	0.500	0.00004	0.01333%
Filter 2	4	0.20022	0.000913	0.20000	0.500	0.00022	0.07333%
No Filter	6	0.20001	0.000027	0.20000	0.500	0.00001	0.00333%
Filter 1	6	0.20006	0.000153	0.20000	0.500	0.00006	0.02000%
Filter 2	6	0.20030	0.000913	0.20000	0.500	0.00030	0.10000%

Table 4.37: DTLZ2 Quadratic Utility Function Test 3 Results

Solution	Interactions	Mean	Std. Dev.	Optimal	Worst	Abs. Dev	Rel. Dev
No Filter	4	0.10001	0.000019	0.10000	0.700	0.00001	0.00167%
Filter 1	4	0.10001	0.000015	0.10000	0.700	0.00001	0.00167%
Filter 2	4	0.10012	0.000450	0.10000	0.700	0.00012	0.02000%
No Filter	6	0.10000	0.000013	0.10000	0.700	0.00000	0.00000%
Filter 1	6	0.10000	0.000009	0.10000	0.700	0.00000	0.00000%
Filter 2	6	0.10009	0.000540	0.10000	0.700	0.00009	0.01500%

4.3.3 Discussions

In this section, we demonstrated that iTDEA converges to the final preference region of the DM interactively on the selected test problems with different utility functions. The algorithm finely converges to the correct region when the specified best solution accurately indicates the true preferences of the DM. In addition, when more interactions are made with the DM, the final focus region is usually more accurate. However, when solutions are filtered before they are presented to the DM, the selected solution may mislead the algorithm. To prevent this, a more appropriate filtering method is needed.

CHAPTER 5

CONCLUSIONS

In this study, we proposed a new multiobjective evolutionary algorithm (MOEA), the Territory Defining Evolutionary Algorithm (TDEA) and tested its performance against well-known MOEAs in the literature. We discuss how the territory defining property of the algorithm helps to preserve diversity and allows fast execution.

We tested the algorithm on 2-, 3- and 5-objective problems, each having different characteristics. We observe that TDEA performs well on the Hypervolume and the Inverted Generational Distance performance metrics in all problems. In most problems, it outperforms other algorithms in both metrics. In addition, it is the second fastest algorithm among all contenders.

In addition to approximating the the entire Pareto-optimal frontier, we proposed a preference incorporation mechanism that focuses on the desired regions of the decision maker. Based on this mechanism, we suggested two preference-based variants of TDEA, namely the Preference-Region Territory Defining Evolutionary Algorithm (prTDEA) and Interactive Territory Defining Evolutionary Algorithm (iTDEA). The former makes use of the preference information specified before the optimization stage to guide the search. On the other hand, iTDEA interactively guides the search towards the regions of interest. In computational tests, we observed that both algorithms converge accurately to the regions that are appealing to the decision maker. We also showed that incorporating preference information allows better details and decreases computational requirements.

This study contributes the MOEA literature by introducing a multiobjective evolutionary algorithm that obtains well-spread final populations without the need of a computationally demanding diversity operator. Instead, the algorithm makes use of its territory defining property and reduces computation time requirements substantially. This property is also used for integrating a preference incorporation mechanism into the algorithm.

This mechanism is utilized for decreasing computational efforts further by integrating the preferences of the decision maker into the search process.

In this study, the preference information in prTDEA is gathered from the decision maker by explicit specification of the preferred regions by weight sets. However, this is not practical. In a future research, a mechanism that converts qualitative statements of the decision maker into weight sets can be implemented.

We observe that filtering solutions before presenting them to the decision maker affects the performance of iTDEA significantly. It should be noted that presenting the entire population to a decision maker for him/her choose the best solution may not be practical. Therefore, there is a need for better filtering mechanisms and implementing it is a future research direction.

In guiding the search towards the preferred regions, we used Tchebycheff weights to specify the preferred regions of the decision maker. In future research, the variable territory sizes property of the algorithm can be combined with other types of preference information as well.

We have tested the algorithms on various hypothetical test problems having 2, 3 and 5 objectives. In order to assess the performances of the algorithms in real-life problems, they should be evaluated in real-life applications as a future research direction. In addition, the performance of algorithm can be tested on various multiobjective combinatorial optimization problems.

REFERENCES

Beume, N., B. Naujoks, and M. Emmerich (2007, 16 September). SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *European Journal of Operational Research 181*(3), 1653–1669.

Bosman, P. A. and D. Thierens (2003, April). The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation 7*(2), 174–188.

Branke, J. and K. Deb (2005). Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Y. Jin (Ed.), *Knowledge Incorporation in Evolutionary Computation*, pp. 461–477. Berlin Heidelberg: Springer. ISBN 3-540-22902-7.

Branke, J., T. Kaußler, and H. Schmeck (2001). Guidance in Evolutionary Multi-Objective Optimization. *Advances in Engineering Software 32*, 499–507.

Chang, W.-C., A. Sutcliffe, and R. Neville (2003, July). A Distance Function-Based Multi-Objective Evolutionary Algorithm. In J. Foster (Ed.), *2003 Genetic and Evolutionary Computation Conference. Late-Breaking Papers*, Chicago, Illinois, USA, pp. 47–53. AAAI.

Coello, C. A. C. (1999, 6-9). An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala (Eds.), *Proceedings of the Congress on Evolutionary Computation*, Volume 1, Mayflower Hotel, Washington D.C., USA, pp. 3–13. IEEE Press.

Coello, C. A. C. (2000). Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *Proc. of the 2000 Congress on Evolutionary Computation*, Piscataway, NJ, pp. 30–37. IEEE Service Center.

Coello, C. A. C., G. B. Lamont, and D. A. V. Veldhuizen (2006). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.

Corne, D. W., N. R. Jerram, J. D. Knowles, and M. J. Oates (2001, 7-11). PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, San Francisco, California, USA, pp. 283–290. Morgan Kaufmann.

Cvetković, D. and I. C. Parmee (2002, February). Preferences and their Application in Evolutionary Multiobjective Optimisation. *IEEE Transactions on Evolutionary Computation 6*(1), 42–57.

Deb, K. (1999). 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*, pp. 135–161. Chichester, UK: John Wiley & Sons, Ltd.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley.

Deb, K. and R. B. Agrawal (1995). Simulated Binary Crossover for Continuous Search Space. *Complex Systems 9*(2), 115–148.

Deb, K. and M. Goyal (1996). A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics 26*(4), 30–45.

Deb, K. and A. Kumar (2007, July). Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. In D. Thierens (Ed.), *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, Volume 1, London, UK, pp. 781–788. ACM Press.

Deb, K., M. Mohan, and S. Mishra (2005, Winter). Evaluating the ϵ-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation 13*(4), 501–525.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002, April). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation 6*(2), 182–197.

Deb, K. and J. Sundar (2006, July). Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In M. K. et al. (Ed.), *2006 Genetic and Evolutionary*

Computation Conference (GECCO'2006), Volume 1, Seattle, Washington, USA, pp. 635–642. ACM Press. ISBN 1-59593-186-4.

Deb, K., L. Thiele, M. Laumanns, and E. Zitzler (2001). Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Fonseca, C. M. and P. J. Fleming (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation 3*(1), 1–16.

Köksalan, M., M. H. Karwan, and S. Zionts (1984). An Improved Method for Solving Multiple Criteria Problems Involving Discrete Alternatives. *IEEE Transactions on Systems, Man, and Cybernetics 14*(1), 24–34.

Köksalan, M. and S. P. Phelps (2007, Spring). An Evolutionary Metaheuristic for Approximating Preference-Nondominated Solutions. *Informs Journal on Computing 19*(2), 291–301.

Knowles, J. D. and D. W. Corne (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation 8*(2), 149–172.

Korhonen, P. J. and J. Laakso (1986). A Visual Interactive Method for Solving the Multiple Criteria Problem. *European Journal of Operational Research 24*(2), 277–287.

Marler, R. T. and J. S. Arora (2004). Survey of Multi-objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization 26*(6), 369–395.

Phelps, S. P. and M. Köksalan (2003, December). An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization. *Management Science 49*(12), 1726–1738.

Rachmawati, L. and D. Srinivasan (2006, July). Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, Vancouver, BC, Canada, pp. 3385–3391. IEEE.

Soylu, B. and M. Köksalan (2006). A Favorable Weight Based Evolutionary Algorithm for Multiple Criteria Problems. Technical report, Middle East Technical University.

Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 546 pp.

Tamaki, H., H. Kita, and S. Kobayashi (1996). Multi-Objective Optimization by Genetic Algorithms : A Review. In T. Fukuda and T. Furuhashi (Eds.), *Proceedings of the 1996 International Conference on Evolutionary Computation (ICEC'96)*, Nagoya, Japan, pp. 517–522. IEEE.

Veldhuizen, D. A. V. and G. B. Lamont (2000). Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation 8*(2), 125–147.

Zitzler, E. and S. Künzli (2004, September). Indicator-based Selection in Multiobjective Search. In X. Y. et al. (Ed.), *Parallel Problem Solving from Nature - PPSN VIII*, Volume 3242 of *Lecture Notes in Computer Science*, Birmingham, UK, pp. 832–842. Springer-Verlag.

Zitzler, E., M. Laumanns, and L. Thiele (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty (Eds.), *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, Athens, Greece, pp. 95–100.

Zitzler, E. and L. Thiele (1998). Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature - PPSN V*, Volume 1498 of *Lecture Notes in Computer Science*, pp. 292–301. Springer.