Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico
Nacional

Unidad Zacatenco
Departamento de Computación

Esquemas de Selección Alternativos para Algoritmos Evolutivos Multi-Objetivo

Tesis que presenta

Adriana Menchaca Méndez

para obtener el Grado de

Doctora en Ciencias en Computación

Director de la Tesis

Dr. Carlos Artemio Coello Coello

México, Distrito Federal. Septiembre, 2015

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

Zacatenco Campus
Computer Science Department

Alternative Selection Schemes for Multi-Objective Evolutionary Algorithms

Submitted by

Adriana Menchaca Méndez

as the fulfillment of the requirement for the degree of

Ph.D. in Computer Science

Advisor

Dr. Carlos Artemio Coello Coello

Mexico City. September, 2015

Resumen

Muchas de las aplicaciones existentes en la ingeniería, la ciencia ó la industria requieren de la resolución de problemas de optimización que involucran varias funciones objetivo (normalmente en conflicto entre sí), las cuales deben ser optimizadas de manera simultánea (v.g., una función objetivo no puede ser mejorada sin deteriorar otra). Estos son los llamados problemas de optimización multi-objetivo (MOPs por sus siglas en inglés). Por lo anterior, cuando resolvemos MOPs no aspiramos a obtener una única solución óptima sino mas bien un conjunto de soluciones óptimas. Este conjunto es conocido como "conjunto de óptimos de Pareto" y a su imagen se le conoce como "frente de Pareto óptimo".

Aunque existen diversas técnicas de programación matemática que nos permiten resolver MOPs, éstas adolecen de varias desventajas, p.ej., la mayor parte de ellas generan una única solución en cada ejecución, tienen dificultades para lidiar con MOPs que cuentan con frentes de Pareto desconectados o que tienen varios falsos frentes de Pareto y requieren que las funciones objetivo y las restricciones se proporcionen en forma algebraica. Estas limitantes han motivado el uso de algoritmos evolutivos para resolver MOPs, dando pie a los denominados algoritmos evolutivos multi-objetivo (MOEAs por sus siglas en inglés).

A pesar de las ventajas que presenta el uso de MOEAs con respecto a las técnicas de programación matemática, este tipo de algoritmos también tienen dificultades en cierto tipo de problemas. De entre dichas limitantes, quizás la más relevante es el hecho de que los MOEAs basados en jerarquización de Pareto (la técnica de selección más utilizada en los MOEAs que se han propuesto en los últimos 25 años) tienen dificultades para lidiar con MOPs que tienen muchas funciones objetivo (4 o más). Esto se debe a que el número de soluciones no dominadas crece muy rápidamente conforme aumentamos el número de funciones objetivo, lo que, en consecuencia, produce que la presión

de selección se diluya a la misma velocidad. Esta limitante ha motivado un número importante de estudios, dando pie a un área que hoy se conoce como "optimización con muchas funciones objetivo", cuyo objetivo fundamental es el diseño de mecanismos de selección que escalen adecuadamente en la presencia de 4 o más funciones objetivo.

En esta tesis estudiamos diferentes mecanismos de selección que no están basados en la relación de dominancia Pareto con el objetivo de identificar sus principales desventajas y limitantes. Como resultado de este estudio, proponemos varios mecanismos de selección enfocados a la resolución de problemas de optimización con muchas funciones objetivos. Dichos mecanismos pueden ser clasificados en tres tipos: (i) los basados en el indicador de "hipervolumen", (ii) los basados en una función de aptitud llamada "maximin" y (ii) los basados tanto en el indicador de "hipervolumen" como en la función de aptitud "maximin". Los diferentes mecanismos de selección propuestos se incorporan a un MOEA, surgiendo así, seis nuevos algoritmos, los cuales son comparados con respecto a MOEAs representativos del estado del arte (SMS-EMOA y MOEA/D) usando problemas de prueba y medidas de desempeño estándar de la literatura especializada. Con base en los experimentos y en el análisis estadístico realizado podemos decir que todos los MOEAs propuestos son competitivos con respecto a SMS-EMOA, si consideramos solamente la calidad de las soluciones encontradas, y todos ellos son superiores a él, si consideramos el tiempo de ejecución requerido para generar dichas soluciones. Con respecto a MOEA/D, los algoritmos aquí propuestos son superiores si consideramos únicamente la calidad de las soluciones generadas, aunque resultan inferiores si consideramos el tiempo de ejecución requerido. Es importante mencionar que aunque el tiempo de ejecución de los algoritmos propuestos es mayor al requerido por el MOEA/D, éste sigue siendo aceptable para problemas con muchas funciones objetivo.

Finalmente, cabe destacar que los MOEAs propuestos en esta tesis no requieren de información adicional a diferencia, por ejemplo, de MOEA/D o de los MOEAs basados en el indicador $R2$, que requieren de un conjunto de pesos convexos bien distribuidos, o de los MOEAs basados en el indicador Δ_p que requieren de un conjunto de referencia el cual idealmente debiera ser el "frente de Pareto óptimo".

Abstract

Many of the existing applications in engineering, science or industry, require of the solution of problems involving several (normally conflicting) objective functions, which must be simultaneously optimized (i.e., an objective function cannot be improved without worsening another one). These are the so-called multi-objective optimization problems (MOPs). Thus, when solving MOPs, we don't aim to obtain a single optimum solution, but a set of them. Such set is known as the "Pareto optimal set" and its image is known as "Pareto optimal front."

Although a variety of mathematical programming techniques are available for solving MOPs, they have several disadvantages, e.g., most of them generate a single solution per run, have difficulties to deal with MOPs having disconnected Pareto fronts or with multi-frontal MOPs, and they require that both the objective functions and the constraints are provided in algebraic form. These limitations have motivated the use of evolutionary algorithms for solving MOPs, giving rise to the so-called multi-objective evolutionary algorithms (MOEAs).

In spite of the advantages that MOEAs have with respect to mathematical programming techniques, they also present difficulties in certain types of problems. From such limitations, perhaps the most relevant is the fact that MOEAs based on Pareto ranking (the most common selection technique adopted by the MOEAs that have been proposed in the last 25 years) have difficulties to deal with MOPs having many objective functions (4 or more). This problem arises because the number of nondominated solutions grows very quickly as we increase the number of objective functions which, in consequence, dilutes the selection pressure at the same rate. This limitation has motivated an important number of studies, giving rise to an area called "many-objective optimization", whose main goal is to design selection mechanisms that can properly scale in the presence of 4 or more objective functions.

In this thesis, we study different selection mechanisms that are not based on the Pareto dominance relation, with the aim of identifying their main disadvantages and limitations. As a result of this study, we propose several selection mechanisms focused on the solution of many-objective optimization problems. Such mechanisms can be classified in three types: (i) those based on the “hypervolume” indicator, (ii) those based on a fitness function called “maximin” and (iii) those based both on the “hypervolume” indicator and the “maximin” fitness function. The proposed selection mechanisms are incorporated into a MOEA, giving rise to six new algorithms, which are compared with respect to MOEAs that are representative of the state-of-the-art (SMS-EMOA and MOEA/D) using standard test problems and performance measures taken from the specialized literature. Based on our experiments and on our statistical analysis of results, we conclude that all the proposed MOEAs are competitive with respect to SMS-EMOA, if we only consider the quality of the solutions found, and all of them are superior to SMS-EMOA if we consider the execution time required to generate such solutions. With respect to MOEA/D, all the algorithms proposed here are superior if we only consider the quality of the solutions obtained, although they are all inferior to MOEA/D if we consider the required execution time. It is worth noticing, however, that, although the execution time required by the algorithms proposed here is higher than the time required by MOEA/D, such an execution time remains affordable for solving many-objective optimization problems.

Finally, it is worth indicating that the MOEAs proposed in this thesis do not require any additional information, unlike, for example, MOEA/D or MOEAs based on the $R2$ indicator, which require a set of well-distributed convex weights, or MOEAs based on the Δ_p indicator, which require of a reference set which, ideally, should be the “Pareto optimal front.”

Agradecimientos

Las primeras palabras de este texto se las quiero dedicar a mi asesor el Dr. Carlos A. Coello Coello, porque gracias a sus enseñanzas, apoyo y guía, este trabajo doctoral es posible. Recuerdo mucho una frase que me dijo antes de ingresar al doctorado, sus palabras aunque no exactas, fueron las siguientes: "Considera que el doctorado no es una meta final, más bien es una etapa de entrenamiento como investigadora en la cual te voy a acompañar y aconsejar. Una vez que pases esta etapa deberás trabajar fuerte para consolidarte como investigadora". Estas palabras las tendré presentes siempre durante mi vida profesional.

En segundo lugar, quiero agradecer a todas las personas que han participado en las revisiones de este trabajo, pues gracias a sus valiosos comentarios pude mejorarlo y extenderlo. Entre estas personas están la Dra. Katya Rodríguez, el Dr. Luis Gerardo De la Fraga, el Dr. Carlos Eduardo Mariano, el Dr. Amilcar Meneses y el Dr. Gregorio Toscano. Así como los revisores de los artículos sometidos y mis profesores y compañeros de seminarios.

Agradezco profundamente a las Doctoras Cristina Riff, Elizabeth Montero y Sanghamitra Bandyopadhyay pues gracias a ellas tuve unas estancias amenas en los países de Chile e India y a pesar de estar lejos de mi familia y amigos siempre me sentí respaldada y tranquila.

Quiero agradecer también a todo el personal del Departamento de Computación del CINVESTAV, en especial a Sofía Reza y Felipa Rosas por su apoyo constante. Su gran sencillez y calidez, generan un ambiente de cordialidad dentro del Departamento.

Agradezco a todos mis compañeros de sala. Su apoyo, sus pláticas, sus risas y hasta sus bromas me ayudaron a inspirarme y motivarme. Agradezco especialmente a Adriana Lara, Alfredo Arias, Antonio López, Eduardo Vazquez y Saúl Zapotecas pues fueron un ejemplo a seguir. Recuerdo mucho mis primeros días en la sala perteneciente a los alumnos de doctorado, los escuchaba discutir con mucho entusiasmo temas relacionados con nuestra área de investigación, podía notar como disfrutaban intercambiar sus ideas y escuchar sus opiniones. De igual forma agradezco a Raquel Hernández y Luis Miguel ya que su gran dedicación por lo que hacen fue una motiviación para trabajar más fuerte.

Quiero agradecer profundamente a Saúl Zapotecas porque en los años que coincidimos me dio muchas enseñanzas tanto académicas como personales, las cuales, sin duda marcaron mi vida y me ayudaron a crecer tanto personalmente como profesionalmente.

Agradezco a mis amigos y familiares cercanos por su apoyo constante durante todo este tiempo: Elia Barajas, Nidia Cortez, Miriam Mecate, Ma. Luisa Méndez, Luz María Méndez, Sara Morales, Ana Lilia Palma, Daniel Acoltzi, Jesús Castro, Edmundo Menchaca, Fernando Reyes y Adrian Sosa. En especial agradezco a Eduardo Lomelí ya que siempre tuvo las palabras exactas para hacerme sentir bien en los momentos difíciles.

Agradezco enormemente a Ivan Negrete quien me ha acompañado y apoyado durante la última etapa de este proyecto. Gracias por esas noches que te desvelaste junto a mí mientras trabajaba, gracias por tu gran esfuerzo para poder acompañarme a algún congreso y compartir conmigo mi vida académica, gracias por expresarme tu confianza y motivarme a seguir adelante con mis sueños, gracias por regalarme tantos momentos de felicidad y sobre todo gracias por completar mi vida.

Agradezco infinitamente a mi familia, a mis padres Reina C. Méndez y F. Rolando Menchaca y a mis hermanos Rolando Menchaca y Ricardo Menchaca pues siempre han sido mi primer fuente de motivación y ejemplo para ser cada día una mejor persona en todos los ámbitos. Agradezco su gran apoyo en todos los aspectos, siempre me han hecho sentir respaldada y me han impulsado a trabajar en lo que me gusta y alcanzar mis objetivos. Sin duda, tuve mucha suerte de nacer en una familia tan extraordinaria como la mía, llena de amor, cordialidad y respeto.

Finalmente, quiero agradecer tanto a CONACyT como a CINVESTAV por las becas económicas que me otorgaron durante mis estudios doctorales y aprovecho para mencionar que el trabajo reportado en esta tesis fue derivado del proyecto CONACyT titulado "Nuevos Paradigmas Algorítmicos en Optimización Evolutiva Multi-Objetivo" (Ref. 221551) el cual es dirigido por el Dr. Carlos A. Coello Coello.

Contributions

In the following, we present the products obtained during the development of this thesis.

Journal Papers

1. A. Menchaca-Mendez and C. A. Coello Coello. An alternative hypervolume-based selection mechanism for multi-objective evolutionary algorithms. *Soft Computing*, accepted for publication, 2015.

2. A. Menchaca-Mendez and C. A. Coello Coello. Selection Mechanisms based on the Maximin Fitness Function to solve Multi-Objective Optimization Problems. *Submitted to Information Sciences*, 2015.

International Conference Papers

1. A. Menchaca-Mendez and C. A. Coello Coello. Solving Multi-Objective Optimization Problems using Differential Evolution and a Maximin Selection Criterion. In *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*, pages 3143–3150, Brisbane, Australia, June 10-15 2012. IEEE Press.

2. A. Menchaca-Mendez and C. A. Coello Coello. Selection Operators Based on Maximin Fitness Function for Multi-Objective Evolutionary Algorithms. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, editors, *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, pages 215–229. Springer. Lecture Notes in Computer Science Vol. 7811, Sheffield, UK, March 19-22 2013.

3. A. Menchaca-Mendez and C. A. Coello Coello. A New Selection Mechanism Based on Hypervolume and its Locality Property. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 924–931, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

4. A. Menchaca-Mendez and C. A. Coello Coello. MD-MOEA : A New MOEA based on the Maximin Fitness Function and Euclidean Distances between Solutions. In *2014 IEEE Congress on Evolutionary Computation (CEC'2014)*, pages 2148–2155, Beijing, China, 6-11 July 2014. IEEE Press. ISBN 978-1-4799-1488-3.

5. A. Menchaca-Mendez and C. A. Coello Coello. MH-MOEA: A New Multi-Objective Evolutionary Algorithm Based on the Maximin Fitness Function and the Hypervolume Indicator. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature - PPSN XIII, 13th International Conference*, pages 652–661. Springer. Lecture Notes in Computer Science Vol. 8672, Ljubljana, Slovenia, September 13-17 2014.

6. A. Menchaca-Mendez and C. A. Coello Coello. GD-MOEA: A New Multi-Objective Evolutionary Algorithm Based on the Generational Distance Indicator. In A. Gaspar-Cunha, C. H. Antunes, and C. Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 156–170. Springer. Lecture Notes in Computer Science Vol. 9018, Guimarães, Portugal, March 29 - April 1 2015.

7. A. Menchaca-Mendez and C. A. Coello Coello. GDE-MOEA : A New MOEA based on the Generational Distance indicator and ϵ-dominance. In *2015 IEEE Congress on Evolutionary Computation (CEC'2015)*, pages 947–955, Sendai, Japan, 25-28 May 2015. IEEE Press. ISBN 978-1-4799-7492-4.

8. A. Menchaca-Mendez, E. Montero, M.-C. Riff, and C. A. Coello Coello. A More Efficient Selection Scheme in iSMS-EMOA. In A. L. Bazzan and K. Pichara, editors, *Advances in Artificial Intelligence – IBERAMIA 2014, 14th Ibero-American Conference on AI*, pages 371–380. Springer. Lecture Notes in Artificial Intelligence Vol. 8864, Santiago de Chile, Chile, November 24-27 2014.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many applications of Engineering, Science and Industry there are problems which involve multiple objective functions [20], which must be simultaneously optimized. They are known as *Multi-objective Optimization Problems (MOPs)*. Since their objectives are in conflict with each other, the notion of optimality refers to the best possible trade-offs among the objectives. Consequently, there is no single optimal solution but a set of solutions, the so-called Pareto optimal set, whose image is known as the Pareto front.

There exist several mathematical programming techniques available for solving MOPs [62]. However, these techniques have several disadvantages, e.g., they have difficulties in MOPs with disconnected Pareto fronts or when the problem is multi-frontal (i.e, it has many false Pareto fronts). Also, they usually generate only one Pareto optimal solution per run. Since many applications in the real world involve solving MOPs with the above features and it is desirable to generate a set of Pareto optimal solutions in a single algorithm's execution, the use of the *Evolutionary Algorithms (EAs)* to solve MOPs has become increasingly popular. Such algorithms are known as *Multi-Objective Evolutionary Algorithms (MOEAs)* and they have several advantages with respect to traditional mathematical programming techniques, e.g., they can generate several Pareto optimal solutions in one run and they are less sensitive to the shape and continuity of the Pareto front. Indeed, MOEAs can work with objective functions which are the result of a simulation.

MOEAs are stochastic search techniques inspired on Darwin's evolutionary theory. They generate a random initial population (i.e., a set of candidate solutions to solve the MOP). After that, they apply variation operators (crossover and mutation) to the current solutions to generate new solutions and, finally, they choose the solutions which will be part of the next generation. This process is repeated during a certain (pre-defined) number of generations. For several years, the most commonly adopted method to select solutions within a MOEA was the Pareto dominance relation. Pareto-based MOEAs work well in MOPs with two or three objective functions. However, in MOPs with more than three objective functions, they have an important disadvantage: a poor scalability. The quick increase in the number of non-dominated solutions as we increase the number of objective functions, rapidly dilutes the effect of the selection mechanism of a MOEA [38]. This has motivated a significant amount of research regarding the scalability of MOEAs with respect to the number of objective functions. This area is known as *many-objective optimization* and usually refers to the solution of MOPs with four or more objective functions.

In this thesis, we are interested in the study of selection mechanisms for MOEAs which are not based on Pareto dominance to solve many-objective optimization problems. As part of our work, we studied the following approaches: (i) those which consider one objective function at a time, (ii) those which decompose the MOP in several single-objective problems, (iii) those which adopt a selection scheme that does not rely on Pareto optimality and, (iv) those which are based on a performance measure. Finally, we propose some alternative selection mechanisms based on approaches of types (iii) and (iv) and we incorporate them into a MOEA. Our results show that our selection mechanisms are a good alternative to solve MOPs having both low dimensionality and high dimensionality in objective function space.

1.1 Problem Statement

It is well-known that MOEAs based on Pareto dominance cannot scale with respect to the number of objective functions. So, there has been a lot of research regarding the design of alternative selection mechanisms. However, the new proposals have some disadvantages, e.g., techniques which consider one objective function at a time need to set an ordering on the objective functions and this is not always possible; techniques which decompose the MOP in several single-objective problems need to generate a set of well-distributed convex weights and this task is difficult in several cases; techniques which adopt a selection scheme that does not rely on Pareto optimality or techniques which are based on a performance measure might not consider all the required aspects of an approximation of the Pareto front. Other common disadvantages between selection schemes are that they require additional information (e.g., knowing the ideal vector or the nadir vector) or their computational cost is high. Therefore, the following question arises: **Is it possible to propose a new selection scheme for Multi-Objective Evolutionary Algorithms (MOEAs) which has the following features?**

- It requires little additional information.
- It is computationally efficient.
- It has a good performance in terms of the two following aspects:
  - It allows to find solutions that are, as close as possible, to the true Pareto front and,
  - it allows to produce solutions that are spread along the Pareto front as uniformly as possible.
- It is scalable with respect to the number of objective functions.

1.2 General and Specific Goals of the Thesis

1.2.1 Main goal

The main goal of this thesis is to contribute to advance of the state of the art in evolutionary multi-objective optimization, particularly regarding the design of selection techniques for MOEAs which are not based on the concept of Pareto optimality. Also, it is of our interest to study density estimators in the context of many-objective problems.

1.2.2 Specific goals

1. Develop a framework of study to compare different selection techniques. Within this framework, we aim to analyze properties such as: required information, computational cost, performance and scalability regarding the number of objective functions.

2. Perform a comparative study of different selection techniques which are not based on the concept of Pareto optimality.

3. Propose at least one new selection scheme which is not based on the concept of Pareto optimality. This scheme must comply with desirable features (little additional information, efficiency, good performance and scalability).

4. Study some density estimators in the same way as selection mechanisms.

5. Propose al least one new density estimator. This estimator must comply with desirable features (little additional information, efficiency, good performance and scalability).

6. Both the new selection scheme and the new density estimator must be incorporated into a MOEA. The new MOEA should show competitive advantages in at least some of the desirable features. The new MOEA should be compared with respect to both MOEAs based on Pareto optimality and indicator-based MOEAs. Also, the study must consider problems with low (2 and 3 objective functions) and high dimensionality (4 or more objective functions).

1.2.3 Expected contributions

- A detailed analysis of selection mechanisms, which are not based on Pareto optimality. This analysis will be focused on the following features: scalability with respect to the number of objective functions, computational cost, required information and performance. As part of this analysis, an appropriate study framework will be defined in order to identify weaknesses and strengths of each technique studied.

- A new MOEA which combines a new selection scheme (not based on Pareto optimality) and a new density estimator. This MOEA must be competitive with respect to the MOEAs of the state of the art considering the features analyzed in the framework indicated before.

- A detailed analysis of the advantages and disadvantages of the schemes proposed (the new selection scheme and the new density estimator) when they are incorporated in an elitist MOEA. This analysis must be based on an exhaustive statistical analysis which will consider well-known MOEAs, standard test functions and performance measures commonly adopted in the specialized literature.

1.3 Structure of the Document

This document is organized as follows. Chapter 2 presents a brief introduction to multi-objective optimization, including basic concepts and some methods to solve multi-objective optimization problems. In Chapter 3, we describe what is a multi-objective evolutionary algorithm and how can we assess its performance. Chapter 4 presents four well-known selection mechanisms for MOEAs: one based on Pareto dominance, other based on a preference relation, other based on decomposition and one more based on a performance measure. In Chapters 5 and 6, we present the contributions of this thesis. Six new MOEAs are presented, evaluated and compared with respect to well-known MOEAs. Finally, in Chapter 7, we present our conclusions and some possible paths for future work.

Chapter 2

Background

Optimization is the process of finding the best possible solution to a given problem. Problems which involve only one objective function are called *single-objective optimization problems (SOPs)* and problems which involve more than one objective function are called *multi-objective optimization problems (MOPs)*.

In the case of SOPs, we can determine if one solution is better than another solution by comparing their values obtained when the solution is evaluated with the objective function. Therefore, we usually obtain a single optimal solution (the global optimum).[1]

On the other hand, in MOPs, the objective functions are normally in conflict. For this reason, solving MOPs implies finding trade-offs among all the objective functions. Consequently, we obtain a set of optimal solutions instead of a single one as in the case of SOPs. This is because we do not have a single optimal solution which optimizes all objective functions at the same time. The decision maker is responsible for choosing only one solution from all those that are available. The notion of "optimality" normally adopted in multi-objective optimization was originally proposed by Edgeworth in 1881 [34] and later generalized by Pareto in 1896 [65].

This chapter presents some basic concepts related to multi-objective optimization. And also, it presents some optimization methods available to solve MOPs. The most important aim of this chapter is that the reader familiarizes with the basic concepts, definitions and notations used in the remainder of this document.

[1] The global minimum may not be unique (i.e., it may happen that different solutions can attain the same objective function value).

2.1 Multi-Objective Optimization

In the following, we present some general concepts and notations used in Multi-objective optimization [21, 62].

Definition 1 (Multi-objective optimization problem) *The general multi-objective optimization problem (MOP) is defined as follows:*

Minimize:

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \cdots, f_k(\vec{x})]^T \tag{2.1}$$

Subject to:

$$\begin{aligned} g_i(\vec{x}) \leq 0 \qquad & i = 1, 2, \cdots, m \\ h_j(\vec{x}) = 0 \qquad & j = 1, 2, \cdots, p \end{aligned}$$

where $\vec{x} = [x_1, x_2, \cdots, x_n]^T$ is the vector of decision variables. The decision variables can be continuous or discrete; however, in this work, we are only interested in continuous domains. $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \cdots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$ are the inequality and equality constraint functions, respectively. According to the type of the functions f, g and h, we can classify MOPs as follows:

- **Multi-objective linear programming**. When all the objective functions and the constraint functions are linear, then the multi-objective optimization problem is called *linear*.

- **Nonlinear multi-objective optimization**. If at least one of the objective functions or the constraint functions are nonlinear, the problem is called *nonlinear*.

- **Convex multi-objective optimization**. The multi-objective optimization problem is *convex* if all the objective functions and the feasible region are convex.

In this work, we are interested in solving nonlinear unconstrained MOPs.

Definition 2 (Decision variables) *The decision variables are the numerical quantities for which values are to be chosen in an optimization problem. The vector $\vec{x}$ of n decision variables is represented by: $\vec{x} = [x_1, x_2, \cdots, x_n]^T$.*

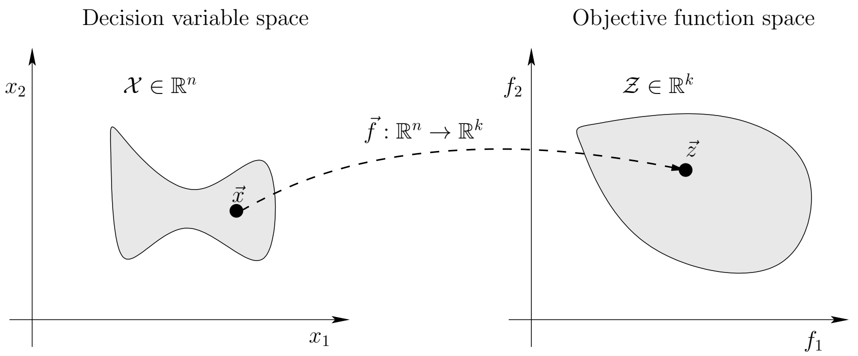

Decision variable space
Objective function space
x_2
$\mathcal{X} \in \mathbb{R}^n$
$\vec{x}$
x_1
$\vec{f} : \mathbb{R}^n \rightarrow \mathbb{R}^k$
f_2
$\mathcal{Z} \in \mathbb{R}^k$
$\vec{z}$
f_1

Figure 2.1: Search spaces for a multi-objective optimization problem with two decision variables and two objective functions. The light gray area denotes the feasible region Ω in both spaces, the decision variable space and the objective function space.

Definition 3 (Decision variable space) *The decision variable space is the n-dimensional space of the decision variables, in which each coordinate axis corresponds with one component of vector $\vec{x}$.*

Definition 4 (Objective functions) *The objective functions evaluate how good a given solution is. They are usually denoted as $f_i(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$. In MOPs, we want to solve more than one objective function and they are denoted by an objective function vector: $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \cdots, f_k(\vec{x})]^T$, where $\vec{f}(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^k$.*

Definition 5 (Objective function space) *The objective function space is the k-dimensional space of the objective functions, in which each coordinate axis corresponds with one component of vector $\vec{f}(\vec{x})$.*

Definition 6 (Feasible region) *The feasible region, $\Omega \in \mathbb{R}^n$, is determined by m inequality constraint functions: $g_i(\vec{x}) \leq 0$ such that $i = 1, 2, \cdots, m$ and p equality constraint functions: $h_j(\vec{x}) = 0$ such that $j = 1, 2, \cdots, p$. We say that a vector $\vec{x}$ is feasible, if it satisfies all constraint functions of the problem ($\vec{x} \in \Omega$).*

In Figure 2.1, we illustrate the above definitions for a MOP with two decision variables and two objective functions. A vector $\vec{x}$ in the decision variable space ($\vec{x} \in \mathcal{X}$) is mapped to a vector $\vec{z}$ in the objective function space ($\vec{z} \in \mathcal{Z}$) using $\vec{f}$.

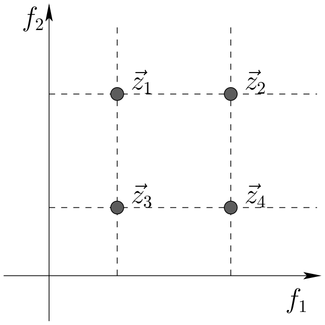


Figure 2.2: Pareto dominance relation. $\vec{z}_3 \prec\prec \vec{z}_2$, $\vec{z}_3 \prec \vec{z}_1$, $\vec{z}_3 \prec \vec{z}_4$, $\vec{z}_4 \prec \vec{z}_2$, $\vec{z}_1 \prec \vec{z}_2$, $\vec{z}_1 \preceq \vec{z}_3$, $\vec{z}_4 \preceq \vec{z}_3$ and $\vec{z}_2 \preceq \vec{z}_4$.

2.2 Notions of Optimality in a Multi-objective Optimization Problem

As we mentioned above, it is not possible to find a single solution that would be optimal for all the objective functions simultaneously when such objectives are in conflict with each other. In MOPs, we only produce partially ordered sets of solutions (e.g., we can say that $[1,1]^T$ is less than $[3,3]^T$, but $[1,3]^T$ and $[3,1]^T$ are incomparable). In the following, we present some important concepts related to Pareto optimality.

Definition 7 (Pareto dominance) *We say that a vector $\vec{x} = [x_1, \ldots, x_n]^T$ dominates vector $\vec{y} = [y_1, \ldots, y_n]^T$, denoted by $\vec{x} \prec \vec{y}$, if and only if $f_i(\vec{x}) \leq f_i(\vec{y})$ for all $i \in \{1, ..., k\}$ and there exists an $i \in \{1, \ldots, k\}$ such that $f_i(\vec{x}) < f_i(\vec{y})$.*

Definition 8 (Weak dominance) *We say that a vector $\vec{x} = [x_1, \ldots, x_n]^T$ weakly dominates vector $\vec{y} = [y_1, \ldots, y_n]^T$, denoted by $\vec{x} \preceq \vec{y}$, if $\vec{x}$ is not worse than $\vec{y}$ in all objectives.*

Definition 9 (Strict dominance) *We say that a vector $\vec{x} = [x_1, \ldots, x_n]^T$ scrictly dominates vector $\vec{y} = [y_1, \ldots, y_n]^T$, denoted by $\vec{x} \prec\prec \vec{y}$, if and only if $f_i(\vec{x}) < f_i(\vec{y})$ for all $i \in \{1, ..., k\}$.*

Figure 2.2 illustrates the Pareto dominance relation for a MOP with two objective functions. For example, vector $\vec{z}_3$ strictly dominates $\vec{z}_2$, vector $\vec{z}_3$ dominates $\vec{z}_1$, vector $\vec{z}_1$ weakly dominates $\vec{z}_3$ and vectors $\vec{z}_1$ and $\vec{z}_4$ are incomparable (i.e., $\vec{z}_1$ and $\vec{z}_4$ are both nondominated).

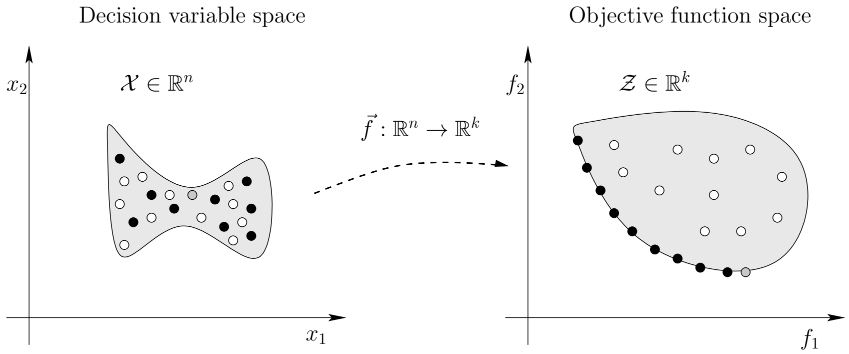

Decision variable space
Objective function space
x_2
$\mathcal{X} \in \mathbb{R}^n$
x_1
$\vec{f} : \mathbb{R}^n \rightarrow \mathbb{R}^k$
f_2
$\mathcal{Z} \in \mathbb{R}^k$
f_1

Figure 2.3: Black points are nondominated vectors and they define the Pareto optimal set in decision variable space and the Pareto front in objective function space. White points are dominated vectors (they are dominated by some black point) and the gray point is weakly Pareto optimal.

Definition 10 (Pareto optimal) *A point $\vec{x}^* \in \Omega$ is Pareto optimal, if there is no other solution $\vec{x} \in \Omega$ such that $\vec{x} \prec \vec{x}^*$.*

Definition 11 (Weak Pareto optimality) *A point $\vec{x}^* \in \Omega$ is weakly Pareto optimal if there is no $\vec{x} \in \Omega$ such that $\vec{x} \prec\prec \vec{x}^*$.*

Definition 12 (Kuhn-Tucker conditions for noninferiority) *If a solution $\vec{x}$ to the general MOP is noninferior, then there exist $w_l \geq 0$, $l = 1, \cdots, k$ (w_l must be strictly positive for some $l = 1, \cdots, k$), and $\lambda_i \geq 0$, $i = 1, \cdots, m$, such that:*

$$\sum_{l=1}^{k} w_l \nabla f_l(\vec{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\vec{x}) = 0 \quad \textit{and} \tag{2.2}$$

$$\vec{x} \in \Omega$$

$$\lambda_i g_i(\vec{x}) = 0, \quad i = 1, \cdots, m$$

*These conditions are **necessary** for a noninferior solution, and when all of the $f_l(\vec{x})$ are concave and Ω is a convex set, they are **sufficient** as well.*

Definition 13 (Pareto optimal set) *For a given MOP, $\vec{f}(\vec{x})$, the Pareto optimal set is defined as: $\mathcal{P}^* = \{\vec{x} \in \Omega | \neg \exists \vec{y} \in \Omega : \vec{y} \prec \vec{x}\}$.*

Definition 14 (Pareto front) *Let $\vec{f}(\vec{x})$ be a given MOP and $\mathcal{P}^*$ the Pareto optimal set. Then, the Pareto Front is defined as: $\mathcal{PF}^* = \{\vec{f}(\vec{x}) \mid \vec{x} \in \mathcal{P}^*\}$.*

In general, it is not easy to find an analytical expression of the line or surface that defines the Pareto front and in most cases, it turns out to be impossible. The common procedure to generate the Pareto front is to compute a sufficient number of points in Ω, and then choose the nondominated vectors from them. Figure 2.3 shows the Pareto optimal set and the Pareto front of a MOP with two decision variables and two objective functions.

Definition 15 (Ideal objective vector) *The ideal objective vector is denoted by $\vec{z}^* = [z_1^*, z_2^*, \cdots, z_k^*]^T$ and it is obtained by minimizing each of the objective functions individually subject to the constraints (i.e., $z_i^* = \min f_i(\vec{x})$ subject to $\vec{x} \in \Omega$).*

Definition 16 (Utopian objective vector) *The utopian objective vector is denoted by $\vec{z}^{**} = [z_1^{**}, z_2^{**}, \cdots, z_k^{**}]^T$. It is an infeasible objective vector whose components are formed by $\vec{z}_i^{**} = \vec{z}_i^* - \epsilon_i$ where $\vec{z}_i^*$ is a component of the ideal objective vector and $\epsilon_i > 0$ is a relatively small but computationally significant scalar.*

Definition 17 (Nadir objective vector) *The nadir objective vector is denoted by $\vec{z}^{nad} = [z_1^{nad}, z_1^{nad}, \cdots, z_k^{nad}]^T$ and its components are the upper bounds of the Pareto optimal set.*

$\vec{z}^{nad}$ can be estimated from a payoff table which is formed by using the decision vector obtained when calculating the ideal objective vector. Row i of the payoff table displays the values of all objective functions calculated at the point where f_i obtained its minimal value. The value of the component z_j^{nad} can be estimated by using the maximum value of the column j. Figure 2.4 illustrates the ideal vector, the utopian vector and the nadir vector for a MOP with two objective functions.

2.3 Optimization Methods to Solve MOPs

Since the size of the Pareto optimal set might be infinite, the goal of most optimization methods is finding a reasonably good approximation of the Pareto optimal front.

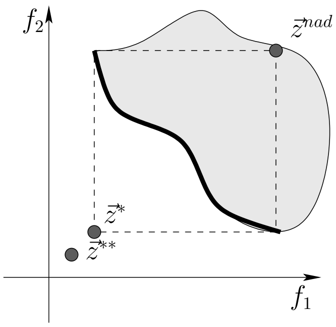


Figure 2.4: Ideal vector ($\vec{z}^{*}$), utopian vector ($\vec{z}^{**}$) and nadir vector ($\vec{z}^{nad}$) for a MOP with two objective functions.

Definition 18 (Approximate Pareto front) *An approximate Pareto front is a subset of the objective space $\mathcal{Z}$ composed of mutually nondominated vectors (e.g., $\mathcal{A} \subset \mathcal{Z}$ such that for any two vectors $\vec{z}_1, \vec{z}_2 \in \mathcal{A}$ is true that $\vec{z}_1 \not\prec \vec{z}_2$ and $\vec{z}_2 \not\prec \vec{z}_1$).*

Currently, it is well-accepted that the quality of an approximate Pareto front is determined by:

- Minimizing the distance of the approximate Pareto front with respect to the true Pareto front,
- Maximizing the spread of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible and,
- Maximizing the number of elements of the approximate Pareto front.

The optimization methods to solve MOPs can be classified in many ways according to different criteria (e.g., enumerative, deterministic and stochastic methods [21]). *Enumerative methods* are the simplest search strategy, since they evaluate each possible solution within some finite search space. It is easy to note that these methods can be infeasible when the search space is too large. *Deterministic algorithms* incorporate problem domain knowledge. Some examples of this type of algorithms are: *greedy methods* which make locally optimal choices and *hill-climbing methods* which use the direction of steepest ascent from the current position. Deterministic algorithms have been successfully used in solving a wide variety of problems. However, when the MOP is high-dimensional, discontinuous, multimodal, and/or NP-complete, they are often ineffective. Since many real world multi-objective optimization problems are irregular, *stochastic search* and optimization methods such as *simulated anneling*, *Monte Carlo methods*, *tabu search* and *evolutionary computation* have been developed as alternative approaches for solving them. Stochastic methods require a function to assign fitness values to the possible solutions of the problem. Also, they need mechanisms for encoding/decoding for doing the mapping between the problem and the domain of the algorithm. Stochastic methods can eventually find the optimum but they cannot guarantee finding the optimal solution. In general, they provide good solutions to a wide range of optimization problems which traditional deterministic search methods find difficult to solve. In this work, we are interested in solving MOPs using **evolutionary computation** (we are going to describe this type of methods in the following chapter).

Although generating the Pareto optimal set plays an important role in multi-objective optimization, and mathematically the problem is considered to be solved when the Pareto optimal set is found, this is not always enough because we need to select only one solution. For this reason, we need a *decision maker* and her or his preferences. The *operations research community* has proposed several optimization techniques (deterministic and stochastic) to solve MOPs which are known as *mathematical programming techniques*. These techniques can be *linear* or *nonlinear*. *Linear programming* is designed to solve problems in which the objective functions and all constraint relations are linear. *Nonlinear programming techniques* solve MOPs which do not meet those restrictions but usually require convex constraint functions. Finally, *stochastic programming* is used when random-valued parameters and objective functions subject to statistical perturbations are part of the problem formulation. Cohon and Marks [24] proposed one of the most popular classifications of techniques within the operations research community:

- **A priori methods**. The decision maker must define the preferences of the objective functions before starting the search. When the decision maker has not properly defined her/his expectations, this type of methods are not recommended.

- **A posteriori methods**. First, the approximate Pareto front is generated, and then, it is presented to the decision maker, who selects the most preferred one(s) according to her/his preferences. Some disadvantages of this type of methods are: they are computationally expensive, it is difficult to generate well-distributed solutions along the Pareto front. Also, it is hard for the decision maker to select from a large set of alternatives.

- **Interactive methods**. Both optimizer and decision maker work progressively. The optimizer produces solutions and the decision maker provides preference information. These methods do not require to have any previous knowledge about the preference structure. However, the information which is presented to the decision maker should be meaningful and easy to understand.

2.3.1 A priori methods

- **Goal Programming**. The ideas of this method were originally introduced by Charnes et al. [18], but the term "goal programming" was introduced by Charnes and Cooper [17]. It is one of the first methods explicitly created for multi-objective optimization. In this method, the decision maker specifies aspiration levels $\bar{z}_i (i = 1, \cdots, k)$ for each objective function and any deviations from these aspiration levels are minimized. An objective function jointly with an aspiration level forms a goal. This method has several variants (e.g., weighted and lexicographic approaches). In the weighted approach, we must solve the following problem:

$$\min \sum_{i=1}^{k} w_i |f_i(\vec{x}) - \bar{z}_i|, \quad \text{subject to:} \quad \vec{x} \in \Omega \tag{2.3}$$

  where w_i are weights previously pre-defined. In the lexicographic approach, the decision maker must specify a lexicographic order on the goals in addition to the aspiration levels. A combination of the weighted and the lexicographic approaches is quite popular. In this case, several objective functions may belong to the same class of importance in the lexicographic order. In each priority class, a weighted sum of the deviational variables is minimized.

  Since goal-setting is an understandable and easy way of making decisions, goal programming is a widely used method to solve practical MOPs. However, the specification of the weighted coefficients or the lexicographic ordering may be difficult. Goal programming is not appropiate if we wish to obtain trade-offs. More details of this method can be found in [62].

- **Lexicographic Method**. In this method, the objective functions are ranked in order to importance by the decision maker (from best to worst). After ranking, the most important objective function is minimized subject to the original constraints. If this problem has a unique solution, it is the solution of the whole MOP. Otherwise, the second most important objective function is minimized but a new constraint is added. The new constraint guarantees that the most important objective function preserves its optimal value. If this problem has a unique solution, it is the solution of the original MOP. Otherwise, the process goes on as above. Suppose that f_1 is the most important objective function. Then the first problem is formulated as follows:

$$\min f_1(\vec{x}), \quad \text{subject to} \quad g_j(\vec{x}) \leq 0; \quad j = 1, 2, \cdots, m \tag{2.4}$$

Let $\vec{x}_1^*$, $f_1^* = f_1(\vec{x}_1^*)$ be the solution to the first problem (eq. 2.4) and f_2 is the second most important objective function. Then, the second problem is formulated as follows:

$$\min f_2(\vec{x}), \quad \text{subject to} \tag{2.5}$$

$$g_j(\vec{x}) \leq 0; \quad j = 1, 2, \cdots, m$$
$$f_1(\vec{x}) = f_1^*$$

And so on, until all k objectives have been considered. The solution to the lexicographic problem is Pareto optimal. However, this method has several disadvantages. The decision maker may have difficulties in establishing an absolute order of importance among the objective functions. Also, it is very likely that the least important objective functions are not taken into consideration at all.

2.3.2 A posteriori methods

- **Weighting Method**. It was presented by Gass and Saaty [40]. This method associates to each objective function one weighting coefficient and its goal is to minimize the weighted sum of the objectives. The weighting coefficients w_i are real numbers and $\sum_{i=1}^{k} w_i = 1$. In this way, the multi-objective optimization problem is transformed into a single objective function as follows:

$$\min \sum_{i=1}^{k} w_i f_i(\vec{x}), \quad \text{subject to} \quad \vec{x} \in \Omega \tag{2.6}$$

where $w_i \geq 0$ for all $i = 1, \cdots, k$ and $\sum_{i=1}^{k} w_i = 1$. Zadeh [86] was the first to show that the third of the Kuhn-Tucker conditions for noninferior solutions implies that these noninferior solutions might be found by solving a scalar optimization problem in which the objective function is a weighted sum of the components of the original vector-valued function. Indeed, the weighting method is a simple way to generate different Pareto optimal solutions. Pareto optimality is guaranteed if the weighting coefficients are positive or the solution is unique. The weakness of the weighting method is that not all of the Pareto optimal points can be found if the problem is nonconvex. The same weakness may also occur in problems with discontinuous objective functions. More details of this method can be found in [62].

- **ϵ-Constraint Method**. This method was introduced by Haimes et al. [85]. It also follows directly from the Kuhn-Tucker conditions for noninferior solutions. The idea of this method is to minimize one (the most preferred or primary) objective function at a time, considering the other objectives as constraints bounded by some allowable levels ϵ_j. By varing these levels ϵ_j, the nondominated solutions of the problem can be obtained. The problem to be solved is now of the form:

$$\min f_i(\vec{x}), \quad \text{subject to} \tag{2.7}$$

$$f_j(\vec{x}) \leq \epsilon_j \text{ for all } j = 1, \cdots, k, j \neq i, \qquad \vec{x} \in \Omega$$

where $i = 1, \cdots, k$. Thus, every Pareto optimal solution of any MOP can be found by the ϵ-constraint method by altering the upper bounds and the function to be minimized. In fact, even the existence of duality gaps in nonconvex problems do not disturb the functioning of the

ϵ-constraint method. However, computationally speaking, this method is expensive, i.e., the ϵ-constraint method needs to perform k optimizations for all f_i as objective functions in order to generate one Pareto optimal solution.

- **Method of weighted metrics**. This method is sometimes called compromise programming [88] and it obtains different solutions by altering the weighted coefficients w_i in the weighted L_p- and Tchebycheff metrics. The weighted L_p-problem for minimizing distances is:

$$\min \left(\sum_{i=1}^{k} w_i |f_i(\vec{x}) - z_i^*|^p \right)^{1/p}, \quad \text{subject to} \quad \vec{x} \in \Omega \tag{2.8}$$

for $1 \leq p < \infty$. The weighted Tchebycheff problem was originally introduced by Bowman [9] and it is of the form:

$$\min \max_{i=1,\cdots,k} [w_i |f_i(\vec{x}) - z_i^*|], \quad \text{subject to} \quad \vec{x} \in \Omega \tag{2.9}$$

If $p = 1$, the problem to be solved is equal to the weighted problem except for a constant (if $\vec{z}^*$ is known globally). If $p = 2$, we have a method of least squares. As p gets larger, the minimization of the largest deviation becomes more and more important. Finally, when $p = \infty$, the only thing that matters is the weighted relative deviation of one objective function. Although the weighted Tchebycheff problem can find every Pareto optimal solution, weakly Pareto optimal solutions may also be included and they need to be filtered out. More details of this method can be found in [62].

2.3.3 Interactive methods

- **Tchebycheff Method**. This method was refined by Steuer [75]. It is an interactive weighting vector space reduction method. This method is easy to use and it does not require complicated information. To start with, an utopian vector below the ideal vector is established. Then, the distance from the utopian vector to the feasible region (measured by a weighted Tchebycheff metric) is minimized. Different solutions are obtained with different weighting vectors in the metric. The solution space is reduced by working with sequences of smaller and smaller subsets of the weighting vector space. Thus, the idea is to develop a sequence of progressively smaller subsets of the Pareto optimal set until a final solution is located. At each iteration, different alternative objective vectors are presented to the decision maker and (s)he must select the most preferred of them. The feasible region is then reduced and alternatives from the reduced space are presented to the decision maker for selection. We know that some of the generated solutions may be weakly Pareto optimal. However, producing weakly Pareto optimal solutions is overcome in the Tchebycheff method by formulating the distance problem as a lexicographic weighted Tchebycheff problem.

- **GUESS Method**. This method is presented by Buchanan [16]. It requires that the ideal vector $\vec{z}^*$ and the nadir vector $\vec{z}^{nad}$ are available. The general idea is to maximize the minimum weighted deviation from the nadir objective vector. In this case, the decision maker specifies a reference point (or a guess) $\vec{z}^h$ and a solution with equal proportional achievements is generated. Then, the decision maker specifies a new reference point and the iteration continues until the decision maker is satisfied with the solution produced. The scales of the objective functions are normalized:

$$\frac{z_i^{nad} - f_i(\vec{x})}{z_i^{nad} - z_i^*} \quad \text{for all} \quad i = 1, \cdots, k; \tag{2.10}$$

The weighted max-min problem to be solved is:

$$\max \quad \min_{i=1,\cdots,k} \left[\frac{1}{w_i} \frac{z_i^{nad} - f_i(\vec{x})}{z_i^{nad} - z_i^*} \right] \quad \text{subject to} \quad \vec{x} \in \Omega \tag{2.11}$$

where the weighting coefficients w_i are positive and they must not be equal to zero. If $w_i = \frac{z_i^{nad} - \vec{z}_i^h}{z_i^{nad} - z_i^*}$ for all $i = 1, \cdots, k$. We can write the problem to be solved in the form:

$$\max \quad \min_{i=1,\cdots,k} \left[\frac{z_i^{nad} - f_i(\vec{x})}{z_i^{nad} - z_i^h} \right] \quad \text{subject to} \quad \vec{x} \in \Omega \tag{2.12}$$

Therefore, the GUESS method is based on a trial and error procedure. The decision maker can examine what kind of an effect her or his input has on the solution obtained and then modify the input. The system does not provide any additional or supporting information about the problem to be solved. The weakness of the GUESS method is its heavy reliance on the availability of the nadir objective vector and we know that it is not easy to determine it (it is usually only an approximation). More details of this method can be found in [62].

Chapter 3

Multi-Objective Evolutionary Algorithms

Mathematical programming techniques have some disadvantages such as the following: (i) they cannot be applicable or have a poor performance in some MOPs (e.g., problems in which the objective functions are non-differentiable or are obtained from a simulation model); (ii) it is necessary to run those algorithms several times to obtain several elements of the Pareto optimal set; (iii) many of them require domain knowledge about the problem to be solved (e.g., the derivatives of the objective functions and the constraints) and; (iv) most of them are very sensitive to the shape or continuity of the Pareto front. This has motivated that a number of stochastic optimization techniques had been used to solve MOPs (e.g., simulated annealing, tabu search, ant colony optimization and evolutionary algorithms).

Evolutionary algorithms (EAs) operate on a population of solutions and adopt a reproduction process (to generate new solutions) and a selection mechanism (to decide which solutions are better). In this way, EAs can find several members of the Pareto optimal set in a single run instead of performing several of them, which is the case for some of the conventional stochastic processes. Rosenberg [69] first proposed the use of genetic algorithms (a particular type of evolutionary algorithm) to solve MOPs at the end of the 1960s. However, it was until 1984, when David Schaffer [71] proposed the first actual implementation of what it is now called a Multi-Objective Evolutionary Algorithm (MOEA). After that, several different algorithms have been proposed and have been successfully applied to a wide variety of problems [22, 74, 96, 46, 47, 39, 92, 29, 89, 35]. Other advantages of using

MOEAs is that they require very little knowledge about the problem that we want to solve and they are also less susceptible to the shape or continuity of the Pareto front than mathematical programming techniques. Additionally, MOEAs are easy to implement, robust, and can be easily implemented in a parallel environment.

The goal of this chapter is to describe in detail the way in which a MOEA works and how can we assess their performance.

3.1 Multi-Objective Evolutionary Algorithms

Single objective EAs and MOEAs share a similar structure. The main difference is the fitness assignment mechanism because a MOEA deals with more than one objective function at a time. Finding an approximate Pareto front is, by itself, a bi-objective problem whose objectives are:

- Minimize the distance of the generated solutions to the Pareto optimal front and,
- maximize the diversity among the solutions as much as possible.

Therefore, the fitness assignment must consider these two objectives. The basic structure of a MOEA is described in Algorithm 1. Usually, the initial population is generated in a random way. The fitness assignment allows us to rank the individuals by defining a preference relation on the objective functions. Pareto optimality is the most common preference relation adopted in MOEAs, but it's not the only one. The selection of individuals for reproduction is normally based on the preference relation adopted for ranking the population, but a density estimator is also required in order to generate different solutions in a single run of a MOEA.

The key elements of a MOEA are:

1. **Fitness assignment**. In a MOEA, we need an additional process to transform a fitness vector into a scalar value. Mainly, there are three schemes to carry out this process:
    - **Criterion-based**. This approach alternately chooses each of the objective functions during the selection stage, i.e., in order to select an individual or group of individuals, only one objective is considered. The Vector Evaluated Genetic Algorithm (VEGA) [71]

Algorithm 1: A generic Multi-objective Evolutionary Algorithm

```
$t \leftarrow 0$;
Generate an initial population ($\mathcal{P}(t)$);
while the stopping criterion is not fulfilled do
    Evaluate the objective vector $\vec{f}$ for each individual in $\mathcal{P}(t)$;
    Assign the fitness of each individual in $\mathcal{P}(t)$. For this task, we can
    use one of the following approaches (or a combination of them):
    criterion-based, aggregation-based, preference-based or
    indicator-based. Usually, Pareto optimality is the most common
    preference relation used in MOEAs and a density estimator is also
    required for breaking ties between competing nondominated
    solutions;
    Select from $\mathcal{P}(t)$, a group of parents ($\mathcal{P}'(t)$), preferring the fitter
    ones;
    Recombine individuals of $\mathcal{P}'(t)$ to obtain an offspring population
    ($\mathcal{P}''(t)$);
    Mutate individuals in $\mathcal{P}''(t)$;
    Combine $\mathcal{P}$ and $\mathcal{P}''(t)$ and select the best individuals to get the
    new population ($P(t+1)$). It is necessary to apply elitism (we
    must prefer solutions in the nondominated front);
    $t \leftarrow t + 1$;
end
```

is an example of this technique. VEGA divides the population into k subpopulations (where k is the number of objectives of the MOP) and a different objective is used to assign fitness within each subpopulation.

- **Aggregation-based**. In this case, the objective functions are aggregated or combined into a single scalar value. During the optimization process, the parameters are systematically varied to generate different elements of the Pareto optimal set. Note that, although an aggregation-based approach can be formulated as a preference relation, the solutions are not compared in objective function space (vectors are mapped from $\mathbb{R}^k$ to $\mathbb{R}$ before the comparison).

- **Preference-based**. In this method, a preference relation is used to induce a partial order of the population in objective function space. Then, a rank (scalar score) is assigned to each solution based on how the solution compares with respect to the other solutions. Pareto dominance is the preference relation most commonly adopted in MOEAs.

- **Indicator-based**. This approach uses performance indicators (also called quality indicators) to assign the fitness of each individual based on how much a solution contributes to a given indicator. The hypervolume indicator has been the most popular quality indicator used for this purpose [52, 35, 8, 50, 63]. However, in recent years other quality indicators such as $R2$ [78, 32, 66, 45] or Δ_p [41, 77, 68, 87] have also been adopted.

2. **Elitism**. It refers to retaining the best solutions during the optimization process since they could be lost when applying the evolutionary operators. This concept plays an important role in modern MOEAs since, along with mutation, guarantees global convergence [94, 96, 93, 54, 55, 29]. In multi-objective optimization, the implementation of elitism is more complicated than in single-objective optimization because we have several solutions that are equally good (i.e., the elements of the Pareto optimal set) and their number can significantly grow over time. There are two approaches to implement elitism:

   - To combine the offspring and parent populations, and then use a deterministic selection mechanism to preserve the best solutions from this combined population into the next generation.

- To mantain an external set of individuals called "archive" which stores the nondominated solutions found during the search process. It is worth emphasizing that the size of this external archive is normally bounded, because if it grows too much, it may dilute the selection pressure.

3. **Density Estimators**. As we already know, one challenge of MOEAs is to obtain a set of nondominated solutions which are well-distributed along the Pareto front. For this reason, several techniques to maintain diversity in the population have been proposed. Some of them are:

   - **Fitness sharing / Niching**. The idea of this technique is to consider fitness as a resource that needs to be shared among individuals in the same niche. The size (or radius) of a neighborhood (or niche) is controlled through the σ_{share} value (niche radius) which is normally a user defined parameter. Then, one must count how many solutions are located within the same niche, and the fitness of each individual located in the same niche is decreased proportionally to the number of individuals sharing the same neighborhood [43, 28]. Formally, the shared fitness F_{S_i} of individual i is defined by:

     $$f_{S_i} = \frac{f_i}{\sum_{j=1}^{N} \phi(d_{ij})}, \tag{3.1}$$

     where f_i is the fitness of individual i, and $\phi(d_{ij})$ is the sharing function, defined by:

     $$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right), & d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

     where d_{ij} is the distance between individuals i and j. Note the following:

     - The definition of the σ_{share} parameter is critical.
     - Distances between solutions can be measured in genotypic or phenotypic space.
     - It is possible to adopt different topologies for defining neighborhoods.

- **Hypergrids**. A hypergrid divides objective function space in regions called hypercubes [54]. Each nondominated solution occupies a hypercube. The idea is to accept nondominated solutions belonging to underpopulated hypercubes. Although the number of divisions in the hypergrid in each dimension is constant, the position and extension of the grid can be adapted during the search process.
- **Clustering**. A clustering technique partitions a set of points in groups (clusters) [94]. The idea is that each cluster contains points very similar to each other and, the points of one cluster are very different from the points of other clusters. In a MOEA, we use clustering as follows: First, we partition the population using a clustering technique. Second, we select the most representative individual of each cluster (i.e., the centroid). Finally, we remove all the other individuals in the cluster.
- **Relaxed forms of Pareto dominance**. Laumanns et al. [56] proposed a relaxed form of Pareto dominance called ϵ-dominance. ϵ-dominance defines a set of boxes of size ϵ and only one nondominated solution is retained for each box (e.g., the one closest to the lower left-hand corner). The use of ϵ-dominance guarantees that the retained solutions are nondominated with respect to all solutions generated during the run.

3.2 Performance Assessment of Multi-Objective Evolutionary Algorithms

The comparison of the performance of different MOEAs is an important issue in multi-objective optimization. The notion of performance includes both the **quality of the outcome and the computational cost for generating that outcome**. Regarding the latter aspect, we can monitor either the number of fitness evaluations or the overall runtime on a particular computer. In this sense, there is no difference between single and multi-objective optimization. However, in the quality aspect, there is an important difference:

- In single-objective optimization, we can define quality of a solution by means of the objective function values: the smaller (to minimize) or the larger (to maximize) value corresponds to a better solution.

- In multi-objective optimization, we can use the concept of Pareto dominance but it is also possible that two solutions are incomparable.

However, it gets even more complicated when we compare two sets of solutions because some solutions in either set may be dominated by solutions in the other set, while others may be incomparable. According to the goals of a MOEA, we need to consider the following aspects to design a good quality measure:

- Minimize the distance of the approximate Pareto front produced by our algorithm with respect to the true Pareto optimal front (assuming we know its location),
- Maximize the spread of solutions found, aiming also to have a distribution of solutions as smooth and uniform as possible along the Pareto front.
- Maximize the number of elements of the approximate Pareto optimal set found.

Several quality measures have been proposed. The most popular are unary quality measures. In this case, the measure assigns each approximate Pareto front a number that reflects a certain quality aspect, and usually a combination of them is used, e.g., [82]. Other methods are based on binary quality measures, which assign numbers to pairs of approximate Pareto fronts, e.g., [95]. Finally, we have the attainment function approach [26], which consists of estimating the probability of attaining arbitrary goals in objective space from multiple approximate Pareto fronts.

For the purposes of this section it is sufficient to deal with objective function space. For each objective vector $\vec{z} = [z_1, z_2, \cdots, z_k]$, there is a decision vector $\vec{x}$ such that $\vec{z} = \vec{f}(\vec{x}) = [f_1(\vec{x}), \cdots, f_k(\vec{x})]$. Also, we consider the outcome of a MOEA (or other heuristic) as a set of incomparable solutions, which will be denoted as **approximate Pareto front**.

Definition 19 *Let $\mathcal{A} \subseteq \Omega$ be a set of objective vectors. $\mathcal{A}$ is called an **approximate Pareto front** if any element of $\mathcal{A}$ does not weakly dominate any other objective vector in $\mathcal{A}$. The set of all approximate Pareto fronts is denoted by Ω.*

It is hard to evaluate an approximate Pareto front considering all aspects such as closeness to the Pareto optimal set, diversity, etc. However, we can make statements about the quality of approximate Pareto fronts in comparison to other approximate Pareto fronts. Table 3.1 shows a summary of the relations between solutions (objective vectors) and approximate Pareto fronts. If we consider the approximate Pareto fronts shown in Figure 3.1, we can say that both $\mathcal{A}_1$ and $\mathcal{A}_2$ *dominate* $\mathcal{A}_3$ ($\mathcal{A}_1 \prec \mathcal{A}_3$ and $\mathcal{A}_2 \prec \mathcal{A}_3$) because any objective vector in $\mathcal{A}_3$ is dominated by at least one vector in $\mathcal{A}_1$ (the same occurs with $\mathcal{A}_2$). Furthermore, $\mathcal{A}_1$ can be considered better than $\mathcal{A}_2$ ($\mathcal{A}_1 \lhd \mathcal{A}_2$) because it contains all objective vectors in $\mathcal{A}_2$ and another vector not included in $\mathcal{A}_2$ (this statement is weaker than the previous one).

Weak dominance ($\mathcal{A} \preceq \mathcal{B}$) means that any objective vector in $\mathcal{B}$ is weakly dominated by a vector in $\mathcal{A}$. In this case, we cannot say that $\mathcal{A}$ is better than $\mathcal{B}$. If $\mathcal{A}$ weakly dominates $\mathcal{B}$, then either $\mathcal{A}$ is better than $\mathcal{B}$ or they are equal. In the same Figure, we can see that $\mathcal{A}_1$ is better than $\mathcal{A}_2$ and $\mathcal{A}_3$ and, $\mathcal{A}_2$ is better than $\mathcal{A}_3$. *The dominance relation* ($\mathcal{A} \prec \mathcal{B}$) is a straightforward extension of Pareto dominance to approximate Pareto fronts. It does not allow that two objective vectors in $\mathcal{A}$ and $\mathcal{B}$ are incomparable, and therefore is stricter than what we usually require. $\mathcal{A}_1$ and $\mathcal{A}_2$ dominate $\mathcal{A}_3$, but $\mathcal{A}_1$ does not dominate $\mathcal{A}_2$. *Strict dominance* stands for the highest level of superiority and means an approximate Pareto front is superior to another approximate Pareto front in the sense that for any objective vector in the latter there exists a vector in the former that is better in all objectives. For example, $\mathcal{A}_1$ strictly dominates $\mathcal{A}_3$, but $\mathcal{A}_2$ does not since the objective vector (11,4) is not strictly dominated by any objective in $\mathcal{A}_2$. Note that there is a natural ordering among the relations: $\mathcal{A} \prec\prec \mathcal{B} \Rightarrow \mathcal{A} \prec \mathcal{B} \Rightarrow \mathcal{A} \lhd \mathcal{B} \Rightarrow \mathcal{A} \preceq \mathcal{B}$.

3.2.1 Quality indicators

Quality measures have been introduced to compare the outcomes of multiobjective optimizers in a quantitative manner. They map approximate Pareto fronts to the set of real numbers.

Definition 20 (Quality Indicator) *An m-ary quality indicator I is a function $I : \Omega^m \rightarrow \mathbb{R}$, which assigns each vector $(\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m)$ of m approximate Pareto fronts a real value $I(\mathcal{A}_1, \cdots, \mathcal{A}_m)$.*

Not a single indicator but rather a combination of different quality indicators is often used in order to assess approximate Pareto fronts. Van

relation	**objective vectors**		**approximate Pareto fronts**	
strictly dominates	$\vec{z}_1 \prec\prec \vec{z}_2$	$\vec{z}_1$ is better than $\vec{z}_2$ in all objectives	$\mathcal{A} \prec\prec \mathcal{B}$	every $\vec{z}_2 \in \mathcal{B}$ is strictly dominated by at least one $\vec{z}_1 \in \mathcal{A}$
dominates	$\vec{z}_1 \prec \vec{z}_2$	$\vec{z}_1$ is not worse than $\vec{z}_2$ in all objectives and better in at least one objective	$\mathcal{A} \prec \mathcal{B}$	every $\vec{z}_2 \in \mathcal{B}$ is dominated by at least one $\vec{z}_1 \in \mathcal{A}$
better	does not apply	does not apply	$\mathcal{A} \lhd \mathcal{B}$	every $\vec{z}_2 \in \mathcal{B}$ is weakly dominated by at least one $\vec{z}_1 \in \mathcal{A}$ and $\mathcal{A} \neq \mathcal{B}$
weakly dominates	$\vec{z}_1 \preceq \vec{z}_2$	$\vec{z}_1$ is not worse than $\vec{z}_2$ in all objectives	$\mathcal{A} \preceq \mathcal{B}$	every $\vec{z}_2 \in \mathcal{B}$ is weakly dominated by at least one $\vec{z}_1 \in \mathcal{A}$
incomparable	$\vec{z}_1 \parallel \vec{z}_2$	neither $\vec{z}_1$ weakly dominates $\vec{z}_2$ nor $\vec{z}_2$ weakly dominates $\vec{z}_1$	$\mathcal{A} \parallel \mathcal{B}$	neither $\mathcal{A}$ weakly dominates $\mathcal{B}$ nor $\mathcal{B}$ weakly dominates $\mathcal{A}$

Table 3.1: Relations on objective vectors and approximate Pareto fronts.

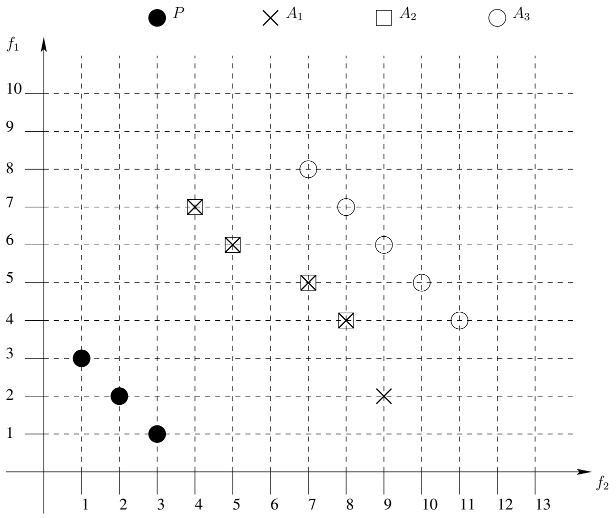


Figure 3.1: Let $\mathcal{P}$ be the Pareto optimal set and let $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ be approximate Pareto fronts. The following dominance relations hold. $\mathcal{A}_1 \prec \mathcal{A}_3$, $\mathcal{A}_2 \prec \mathcal{A}_3$, $\mathcal{A}_1 \prec\prec \mathcal{A}_3$, $\mathcal{A}_1 \preceq \mathcal{A}_1$, $\mathcal{A}_1 \preceq \mathcal{A}_2$, $\mathcal{A}_1 \preceq \mathcal{A}_3$, $\mathcal{A}_2 \preceq \mathcal{A}_2$, $\mathcal{A}_2 \preceq \mathcal{A}_3$, $\mathcal{A}_3 \preceq \mathcal{A}_3$, $\mathcal{A}_1 \lhd \mathcal{A}_2$, $\mathcal{A}_1 \lhd \mathcal{A}_3$ and $\mathcal{A}_2 \lhd \mathcal{A}_3$.

Veldhuizen and Lamont [82] applied a combination of three indicators $\mathbf{I} = (I_{GD}, I_S, I_{ONVG})$, where $I_{GD}(\mathcal{A})$ denotes the average distance of objective vectors in $\mathcal{A}$ to the Pareto optimal front, $I_S(\mathcal{A})$ measures the variance of distances between neighboring objective vectors in $\mathcal{A}$, and $I_{ONVG}(\mathcal{A})$ gives the number of elements in $\mathcal{A}$.

An **interpretation function** maps vectors of real numbers to booleans, e.g., we would define $E(I_{GD}(\mathcal{A}), I_{GD}(\mathcal{B})) := (I_{GD}(\mathcal{A}) = 0 \quad \wedge \quad I_{GD}(\mathcal{B}) > 0)$. E is true if and only if $I_{GD}(\mathcal{A}) = 0$ and at the same time $I_{GD}(\mathcal{B}) > 0$ (all objective vectors in $\mathcal{A}$ have zero distance to the Pareto optimal set P, and therefore $\mathcal{A} \subseteq \mathcal{P}$ and $\mathcal{B} \not\preceq \mathcal{A}$ for any approximate Pareto front $\mathcal{B} \not\subseteq \mathcal{P}$). A combination of one or more quality indicators, $\mathbf{I}$, and an interpretation function E is also called a **comparison method** $C_{\mathbf{I},E}$. In the above example, the comparison method is $C_{I_{GD},E}(\mathcal{A}, \mathcal{B}) = E(I_{GD}(\mathcal{A}), I_{GD}(\mathcal{B}))$ and the conclusion is that $C_{I_{GD},E}(\mathcal{A}, \mathcal{B}) \Leftrightarrow \mathcal{A} \subseteq \mathcal{P} \wedge \mathcal{B} \not\subseteq \mathcal{P} \wedge \mathcal{B} \not\preceq \mathcal{A}$. The comparison method is formally defined as follows:

Definition 21 (Comparison Method) *Let $\mathcal{A}, B \in \Omega$ be two approximate Pareto fronts, $\mathbf{I} = (I_1, I_2, \cdots, I_k)$ a combination of quality indicators, and $E : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \{false, true\}$ an interpretation function which maps two real* *vectors of length k to a Boolean value. If all indicators in $\mathbf{I}$ are unary, the comparison method $C_{\mathbf{I},E}$ defined by $\mathbf{I}$ and E is a function of the form*

$$C_{\mathbf{I},E}(\mathcal{A}, B) = E(\mathbf{I}(\mathcal{A}), \mathbf{I}(B)) \tag{3.3}$$

where $\mathbf{I}(\mathcal{A}') = (I_1(\mathcal{A}'), I_2(\mathcal{A}'), \cdots, I_k(\mathcal{A}'))$ for all $\mathcal{A}' \in \Omega$. If $\mathbf{I}$ contains only binary indicators, the comparison method $C_{\mathbf{I},E}$ is defined as a function of the form

$$C_{\mathbf{I},E}(\mathcal{A}, B) = E(\mathbf{I}(\mathcal{A}, B), \mathbf{I}(B, \mathcal{A})) \tag{3.4}$$

where $\mathbf{I}(\mathcal{A}', B') = (I_1(\mathcal{A}', B'), I_2(\mathcal{A}', B'), \cdots, I_k(\mathcal{A}', B'))$ for all $\mathcal{A}', B' \in \Omega$.

3.2.2 Linking comparison methods and dominance relations

It is important to know if a comparison method $C_{I,E}(\mathcal{A}, \mathcal{B})$ is a sufficient condition to say that $\mathcal{A}$ is better than $\mathcal{B}$, i.e., $C_{I,E}(\mathcal{A}, \mathcal{B}) \Rightarrow \mathcal{A} \lhd \mathcal{B}$, and, if $C_{I,E}(\mathcal{A}, \mathcal{B})$ is, in addition, a necessary condition for $\mathcal{A} \lhd \mathcal{B}$, i.e., $C_{I,E}(\mathcal{A}, \mathcal{B}) \Leftrightarrow \mathcal{A} \lhd \mathcal{B}$. The compatibility and completeness terms are used in order to characterize a comparison method.

Definition 22 (Compatibility and Completeness) *Let $\blacktriangleleft$ be an arbitrary binary relation on approximate Pareto fronts (e.g., the relations shown in Table 3.1). The comparison method $C_{\mathbf{I},E}$ is denoted as $\blacktriangleleft$-compatible if either for any $\mathcal{A}, \mathcal{B} \in \Omega$*

$$C_{\mathbf{I},E}(\mathcal{A}, \mathcal{B}) \Rightarrow \mathcal{A} \blacktriangleleft \mathcal{B}$$

or for any $\mathcal{A}, \mathcal{B} \in \Omega$

$$C_{\mathbf{I},E}(\mathcal{A}, \mathcal{B}) \Rightarrow \mathcal{B} \blacktriangleleft \mathcal{A}$$

The comparison method $C_{\mathbf{I},E}(\mathcal{A}, \mathcal{B})$ is denoted as $\blacktriangleleft$-complete if either for any $\mathcal{A}, \mathcal{B} \in \Omega$

$$\mathcal{A} \blacktriangleleft \mathcal{B} \Rightarrow C_{\mathbf{I},E}(\mathcal{A}, \mathcal{B})$$

or for any $\mathcal{A}, \mathcal{B} \in \Omega$

$$\mathcal{B} \blacktriangleleft \mathcal{A} \Rightarrow C_{\mathbf{I},E}(\mathcal{A}, \mathcal{B})$$

Let's assume that we have a comparison method that is $\blacktriangleleft$-complete but not *compatible* with respect to the $\lhd$ relation. If we use this comparison method to compare two sets $\mathcal{A}$ and $\mathcal{B}$ with $\mathcal{A} \lhd \mathcal{B}$ ($\mathcal{A}$ is better than $\mathcal{B}$), then our comparison method returns true. However, there are also sets $\mathcal{A}$ and

$\mathcal{B}$ with $\mathcal{A} \not\triangleleft \mathcal{B}$ ($\mathcal{A}$ is not better than $\mathcal{B}$) for which the comparison method returns true. If we use a comparison method that is $\blacktriangleleft$-compatible, then the above situation is safe: if our comparison method yields true, we can be sure that $\mathcal{A}$ is better than $\mathcal{B}$. However, if the comparison method is not $\triangleleft$-complete, there may be sets $\mathcal{A}$ and $\mathcal{B}$ where $\mathcal{A}$ is better than $\mathcal{B}$, but our comparison method returns false. Zitzler et al. [97] presented the following key results.

- Unary quality indicators are, in general, not capable of indicating whether an approximate Pareto front is better than another even if we use several of them. This also holds, if we consider approximate Pareto fronts containing a single objective vector only.

- There are unary indicators which allow to infer if an approximate Pareto front is not worse than another, e.g., the distance indicator by Czyzak and Jaszkiewicz [25], the hypervolume indicator by Zitzler and Thiele [95], or the unary ϵ-indicator presented in [97].

- Binary indicators in principle do not possess the theoretical limitations of unary indicators. The binary ϵ-indicator proposed in [97] is capable of detecting whether an approximate Pareto front is better than another. However, not all existing binary indicators have this property.

Knowles et al. [53] presented a similar idea on the performance assessment of stochastic multi-objective optimizers in which they recommend having quality indicators that are only Pareto dominance compliant or Pareto compliant.

Definition 23 (Pareto compliant) *An indicator $I : \Omega \to \mathbb{R}$ is Pareto compliant if for all $\mathcal{A}, \mathcal{B} \in \Omega : \mathcal{A} \preceq \mathcal{B} \Rightarrow I(\mathcal{A}) \geq I(\mathcal{B})$, assuming that greater indicator values correspond to higher quality (otherwise $\mathcal{A} \preceq \mathcal{B} \Rightarrow I(\mathcal{A}) \leq I(\mathcal{B})$). In the context of order theory, a Pareto compliant indicator I is an order-preserving function from $(\Omega, \preceq)$ to $(\mathbb{R}, \geq)$ (respectively, $(\mathbb{R}, \leq)$).*

Many of the indicators that are employed in the MOEA literature are not Pareto compliant. Several popular indicators are designed to assess just one isolated aspect of an approximate Pareto front's quality, e.g., its proximity to the Pareto optimal front, or its spread in objective space. These quality indicators are known as Pareto non-compliant.

Definition 24 (Pareto Noncompliant) *Any indicator that can yield for any approximate Pareto fronts $\mathcal{A}, \mathcal{B} \in \Omega$ a preference for $\mathcal{A}$ over $\mathcal{B}$, when $\mathcal{B}$ is preferable to $\mathcal{A}$ with respect to weak Pareto dominance ($\mathcal{B} \preceq \mathcal{A} \wedge \mathcal{A} \npreceq \mathcal{B}$, or $\mathcal{B} \lhd \mathcal{A}$ for short), is Pareto non-compliant.*

3.2.3 Unary quality indicators

Let $\mathcal{PF}$ be the optimal Pareto front and $\mathcal{A}$ an approximate Pareto front. Next, we define some unary quality indicators:

- **Error Ratio** (I_{ER}): I_{ER} reports the percentage of the number of vectors in $\mathcal{A}$ that are not members of $\mathcal{PF}$ [81, 79]. I_{ER} is Pareto compliant and it is defined as follows:

  $$I_{ER} = \frac{\sum_{i=1}^{|\mathcal{A}|} e_i}{|\mathcal{A}|} \tag{3.5}$$

  where $e_i = 0$ when the i^{th} vector of $\mathcal{A}$ is an element of $\mathcal{PF}$ and $e_i = 1$ when the i^{th} vector of $\mathcal{A}$ is not an element of $\mathcal{PF}$. If $I_{ER} = 0$ then $\mathcal{A} \subseteq \mathcal{PF}$; but when $I_{ER} = 1$ none of the points in $\mathcal{A}$ are in $\mathcal{PF}$. A lower I_{ER} value is better.

- **Generational Distance** (I_{GD}): I_{GD} reports how far, on average, $\mathcal{A}$ is from $\mathcal{PF}$ [23, 80, 81]. I_{GD} is Pareto non-compliant and it is defined as:

  $$I_{GD} = \frac{1}{|\mathcal{A}|} \left(\sum_{i=1}^{|\mathcal{A}|} d_i^p \right)^{\frac{1}{p}} \tag{3.6}$$

  where $|\mathcal{A}|$ is the number of vectors in $\mathcal{A}$, $p = 2$ and d_i is the euclidean phenotypic distance between each member, i, of $\mathcal{A}$ and the closest member in $\mathcal{PF}$ to that member, i. If $I_{GD} = 0$, $\mathcal{A} \subseteq \mathcal{PF}$.

- **Inverted Generational Distance** (I_{IGD}). I_{IGD} does not only indicate the proximity of the set $\mathcal{A}$ to $\mathcal{PF}$, but also provides an estimate of its maximum spread [19]. I_{IGD} is Pareto non-compliant and it is analogous to GD, but measured from $\mathcal{PF}$ to $\mathcal{A}$. I_{IGD} is defined by:

  $$I_{IGD} = \frac{1}{|\mathcal{PF}|} \left(\sum_{i=1}^{|\mathcal{PF}|} d_i^p \right)^{\frac{1}{p}} \tag{3.7}$$

where $|\mathcal{PF}|$ is the number of vectors in $\mathcal{PF}$, $p = 2$ and d_i is the euclidean phenotypic distance between each member, i, of $\mathcal{PF}$ and the closest member in $\mathcal{A}$ to that member, i. If $I_{IGD} = 0$, $\mathcal{A} = \mathcal{PF}$.

- **Spacing (I_S)**: I_S describes the spread of the vectors in $\mathcal{A}$ [23, 72]. This Pareto non-compliant indicator measures the distance variance of neighboring vectors in $\mathcal{A}$. Formally, I_S is defined as follows:

$$I_S = \sqrt{\frac{1}{|\mathcal{A}| - 1} \sum_{i=1}^{|\mathcal{A}|} (\bar{d} - d_i)^2} \tag{3.8}$$

where $d_i = \min_{j=1\cdots,|\mathcal{A}|,i\neq j} \sum_{m=1}^{k} |f_m^i - f_m^j|$, $i, j = 1, \cdots, |\mathcal{A}|$, k is the number of objective functions and $\bar{d}$ is the mean of all d_i. When $I_S = 0$, all members are spaced evenly apart. Note that this indicator assumes that a MOEA has already converged to the true Pareto front.

- **Hypervolume (I_H)**. I_H was originally proposed by Zitzler and Thiele in [95], and it's defined as the size of the space covered by the Pareto optimal solutions. If Λ denotes the Lebesgue measure, I_H is defined as:

$$I_H(A, \vec{z}_{ref}) = \Lambda \left(\bigcup_{\vec{z} \in \mathcal{A}} \{\vec{y} \mid \vec{z} \prec \vec{y} \prec \vec{z}_{ref}\} \right) \tag{3.9}$$

where $\vec{z}_{ref} \in \mathbb{R}^k$ denotes a reference point that should be dominated by all the Pareto optimal points. A high I_H value, indicates that $\mathcal{A}$ is close to $\mathcal{PF}$ and has a good spread towards the extreme portions of the $\mathcal{PF}$. This is the only unary indicator that is known to be strictly Pareto compliant.

- ***R*2-Indicator**. It belongs to the family of R indicators proposed by Hansen and Jaszkiewicz [44]. R-indicators use utility functions for evaluating approximate Pareto fronts. The $R2$-indicator is weakly monotonic and simultaneously evaluates several desired aspects of an approximate Pareto front. Let U be a set of general utility functions, the $R2$-indicator is thus defined as [14]:

$$I_{R2}(\mathcal{A}, U) = -\frac{1}{|U|} \sum_{u \in U} \max_{\vec{a} \in \mathcal{A}} \{u(\vec{a})\} \tag{3.10}$$

Regarding the choice of the utility functions u, there are several possibilities, such as: weighted sum, least squares, weighted Tchebycheff metric, etc. These utility functions have an associated set of uniformly distributed weight vectors W and a reference point $\vec{z}^{*}$, in order to maintain diversity. The most common utility function is the weighted Tchebycheff metric:

$$R2(\mathcal{A} : W, \vec{z}^{*}) = \frac{1}{|W|} \sum_{\vec{w} \in W} \min_{\vec{a} \in \mathcal{A}} \left[\max_{i=1,\cdots,k} w_i |a_i - z_i^*| \right] \tag{3.11}$$

- **Δ_p Indicator (I_{Δ_p}).** I_{Δ_p} was proposed by Schütze et al. [73]. It can be seen as an "averaged Hausdorff distance" between the approximate Pareto front and the Pareto optimal front. It is composed of slight modifications of two well-known performance indicators: generational distance (I_{GD}) and inverted generational distance (I_{IGD}). I_{Δ_p} is defined as:

$$I_{\Delta_p} = \max(I_{GD_p}, I_{IGD_p}) \tag{3.12}$$

with:

$$I_{GD_p} = \left(\frac{1}{|\mathcal{A}|} \sum_{i=1}^{|\mathcal{A}|} d_i^p \right)^{\frac{1}{p}} \tag{3.13}$$

d_i is the euclidean phenotypic distance between each member, i, of $\mathcal{A}$ and the closest member in $\mathcal{PF}$ to that member, i. And,

$$I_{IGD_p} = \left(\frac{1}{|\mathcal{PF}|} \sum_{i=1}^{|\mathcal{PF}|} d_i^p \right)^{\frac{1}{p}} \tag{3.14}$$

d_i is the euclidean phenotypic distance between each member, i, of $\mathcal{PF}$ and the closest member in $\mathcal{A}$ to that member, i. I_{Δ_p} simultaneously evaluates proximity to the Pareto optimal front and spread of solutions along it. I_{Δ_p} is Pareto non-compliant.

3.2.4 Binary quality indicators

- **ϵ-Indicator (I_ϵ).** It is a Pareto compliant measure. Given two approximate Pareto fronts, $\mathcal{A}$ and $\mathcal{B}$, the ϵ-indicator measures the smallest amount, ϵ, that must be used to translate the set $\mathcal{A}$, so that every point in $\mathcal{B}$ is covered [97].

Definition 25 *Let's assume, without loss of generality, a minimization problem with k objective functions, then, an objective vector $\vec{z}^1 = [z_1^1, \cdots, z_k^1]$ is said to ϵ-**dominate** another objective vector $\vec{z}^2 = [z_1^2, \cdots, z_k^2]$, denoted by $\vec{z}^1 \preceq_\epsilon \vec{v}^2$, if and only if $\forall 1 \leq i \leq k : z_i^1 \leq \epsilon \cdot z_i^2$ for a given $\epsilon > 0$.*

Loosely speaking, a vector $\vec{z}^1$ is said to ϵ-dominate another vector $\vec{z}^2$, if we can multiply each objective value in $\vec{z}^2$ by a factor ϵ and the resulting objective vector is still weakly dominated by $\vec{z}^1$. Therefore, $\vec{z}^1 \prec\prec \vec{z}^2$ implies that there exists an $\epsilon < 1$ such that $\vec{z}^1$ ϵ-dominates $\vec{z}^2$. We define the binary ϵ-indicator I_ϵ as:

$$I_\epsilon(\mathcal{A}, \mathcal{B}) = \min\{\epsilon \in \mathbb{R} \quad | \quad \forall \vec{b} \in \mathcal{B} \quad \exists \vec{a} \in \mathcal{A} : \vec{a} \prec_\epsilon \vec{b}\} \tag{3.15}$$

So, when $I_\epsilon(\mathcal{A}, \mathcal{B}) < 1$, all solutions in $\mathcal{B}$ are dominated by a solution in $\mathcal{A}$. If $I_\epsilon(\mathcal{A}, \mathcal{B}) = 1$ and $I_\epsilon(\mathcal{B}, \mathcal{A}) = 1$, then $\mathcal{A}$ and $\mathcal{B}$ represent the same approximate Pareto front. If $I_\epsilon(\mathcal{A}, \mathcal{B}) > 1$ and $I_\epsilon(\mathcal{B}, \mathcal{A}) > 1$, then $\mathcal{A}$ and $\mathcal{B}$ are incomparable because they both contain solutions not dominated by the other set.

- **Two Set Coverage (I_{SC}).** It was proposed by Zitzler et al. [90] and it is a Pareto compliant indicator. Let $\mathcal{A}, \mathcal{B}$ be two approximate Pareto fronts, I_{SC} is defined as follows:

$$I_{SC}(\mathcal{A}, \mathcal{B}) = \frac{|\vec{b} \in \mathcal{B} \text{ such that } \exists \vec{a} \in \mathcal{A} \text{ with } \vec{a} \prec \vec{b}|}{|\mathcal{B}|} \tag{3.16}$$

If all points in $\mathcal{A}$ dominate or are equal to all points in $\mathcal{B}$, then, by definition $I_{SC} = 1$. $I_{SC} = 0$ implies that no element in $\mathcal{B}$ is dominated by any element of $\mathcal{A}$. In general, $I_{SC}(\mathcal{A}, \mathcal{B})$ and $I_{SC}(\mathcal{B}, \mathcal{A})$ both have to be considered due to set intersections not being empty.

Chapter 4

Selection mechanisms for MOEAs

An important element of MOEAs is their selection mechanism because its aim is to guide the search. Therefore, it is a key element for ensuring that a MOEA converges to the optimal solutions that we aim to obtain. Based on their selection mechanism, MOEAs can be classified in two groups: (i) those that incorporate the concept of Pareto optimality, and (ii) those that do not use Pareto dominance to select individuals. Although the use of Pareto-based selection (mainly through the use of some Pareto ranking scheme [21]) has been the most popular choice within the specialized literature for the last 15 years, such type of approach has several limitations. From them, its poor scalability when increasing the number of objective functions is, perhaps, the most remarkable. The quick increase in the number of nondominated solutions as we increase the number of objective functions, rapidly dilutes the effect of the selection mechanism of a MOEA [38]. This has triggered an important amount of research on an area called "many-objective optimization", which refers to the study of problems having four or more objective functions.

In the current literature, we can identify several approaches to cope with many-objectives problems, such as:

- To adopt or propose a preference relation that induces a finer grain order on the solutions than that induced by the Pareto dominance relation [31, 37, 76, 70].
- To reduce the number of objectives of the problem during the search

process [15] or, *a posteriori*, during the decision making process [13, 60].

- To adopt a selection scheme that does not rely on Pareto optimality (e.g., using compromise functions [7], alternative ranking schemes [61] or a selection mechanism based on a performance measure (from which hypervolume has been a popular choice, in spite of its considerably high computational cost [91, 8]).

In this chapter, we will study four well-known selection schemes: (i) one based on Pareto dominace (used by *NSGA-II* [29]), (ii) one based on alternative ranking schemes (the Maximin fitness function [4, 6, 5]), (iii) one based on decomposition (used by *MOEA/D* [89]), and finally (iv) one based on a quality indicator (used by *SMS-EMOA* [8]).

4.1 Pareto-dominance approach

Perhaps the most popular MOEA based on Pareto dominance is the nondominated Sorting Genetic Algorithm II (*NSGA-II*) [29]. This is an improved version of the nondominated Sorting Genetic Algorithm (*NSGA*) [74] which is based on the ranking procedure originally proposed by Goldberg [42]. The *NSGA-II* builds a population of individuals (candidate solutions), and then ranks and sorts each individual according to its non domination level. After that, it applies the evolutionary operators (crossover and mutation) to create a new pool of offspring and it combines the parents and offspring in a single population. Finally, it partitions the new population into fronts. Then, *NSGA-II* breaks ties between nondominated solutions, by computing a crowding distance. This mechanism keeps the population diverse and helps the algorithm to produce Pareto fronts with well-distributed solutions. Although, the *NSGA-II* was originally proposed 15 years ago, it is still frequently adopted to compare results of current MOEAs

The ranking procedure works as follows: for each individual p (in the objective space), we obtain the set of solutions that p dominates (we called it $\mathcal{S}_p$) and the number of solutions which dominate $\vec{p}$ (we called it n_p). All the individuals with domination count equal to zero belong to the nondominated front. We put the nondominated individuals in a separate set ($\mathcal{F}_i$), where i indicates the rank of the individuals which are in that set (i starts with 1). For each solution p in the nondominated front, we visit each member q of its set $\mathcal{S}_p$ and its domination count is reduced by one. We increment i by one

Algorithm 2: Fast nondominated sort

```
Input  : Population $\mathcal{P}$
Output: Partition in fronts of the population
foreach $p \in \mathcal{P}$ do
    $\mathcal{S}_p = \emptyset$;
    $n_p = 0$;
    foreach $q \in \mathcal{P}$ do
        if $p.\vec{f} \prec q.\vec{f}$ then
            $\mathcal{S}_p = \mathcal{S}_p \cup \{q\}$; //Add $q$ to the set of solutions dominated
            by $p$
        else
            if  $q.\vec{f} \prec p.\vec{f}$ then
                $n_p = n_p + 1$; //Increment the domination counter of $p$
            end
        end
    end
    if $n_p = 0$ then
        $p_{rank} = 1$;
        $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$
    end
end
$i = 1$;                              //Initialize the front counter
while $\mathcal{F}_i$ do
    $\mathcal{Q} = \emptyset$;           //Used to store the members of the next front
    foreach $p \in \mathcal{F}_i$ do
        foreach $q \in \mathcal{S}_p$ do
            $n_q = n_q - 1$;
            //$q$ belongs tho the next front
            if $n_q = 0$ then
                $q_{rank} = i + 1$;
                $\mathcal{Q} = \mathcal{Q} \cup \{q\}$
            end
        end
    end
    $i = i + 1$;
    $\mathcal{F}_i = \mathcal{Q}$;
end
```

Algorithm 3: Crowding distance

```
Input : Front $\mathcal{F}_i$, vector of maximum and minimum values of each
        objective functions ($\vec{f}^{max}$ and $\vec{f}^{min}$ respectively)
Output: Each individual with its crowding distance
$l = |\mathcal{F}_i|$;
foreach $p \in \mathcal{F}_i$ do
    $p.dist = 0$;
end
foreach Objective function $k$ do
    Sort individuals $\mathcal{F}_i$ regarding to the $k$th objective function in
    ascending order;
    $p_1.dist = p_l.dist = \infty$;
    for $j = 2$ to $l - 1$ do
        $p_j.dist = p_j.dist + (p_{j+1}.f_k - p_{j-1}.f_k)/(f_f^{max} - f_k^{min})$;
    end
end
```

and repeat this process until all individuals have a rank. Algorithm 2 shows this process. Since the initial domination counter of each solution can be at most $N - 1$, where N is the size of the population, the total complexity is $O(mN^2)$, where m is the number of objective functions.

To mantain diversity, *NSGA-II* uses the crowding distance which estimates the density of solutions surrounding a particular solution in the population. The crowding distance value of a point is calculated as the average distance of the two points on either side of the point in question along each of the objective functions. This distance serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices. The extreme solutions have an infinite distance value. In Algorithm 3, we show the process to assign crowding distance to each individual in a nondominated front. The complexity of this procedure is $O(mN \log N)$ and it is governed by the sorting of the population regarding each objective function. Figure 4.1 illustrates these two processes (Pareto ranking and crowding distance). It is important to note that the complexity to calculate the rank and the crowding distance of all individuals in the population is linear with respect to the number of objective functions.

As we mentioned above, an important limitation of all selection tech-

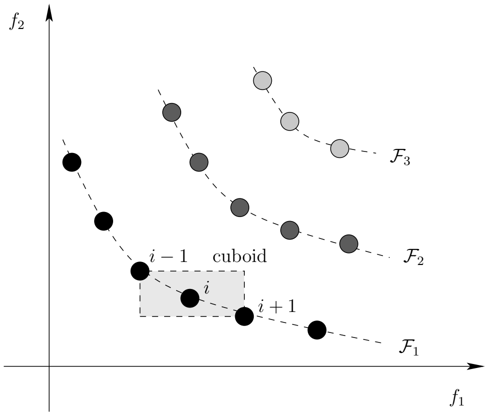


Figure 4.1: Pareto ranking and crowding distance in *NSGA-II*. Points marked in filled circles with the same color belong to the same front. The crowding distance of the ith solution in the first front is the average side length of the cuboid shown with a dashed box.

niques based on Pareto dominace is the quick increase in the number of nondominated solutions as we increase the number of objectives.

4.2 Maximin Fitness Function

The maximin fitness function (MFF) was proposed by Richard Balling and Scott Wilson in [4],[6]. It is derived from Definition 7 (page 10) and it works as follows. Let's consider a MOP with K objective functions and an evolutionary algorithm whose population is $\mathcal{P}$. Let f_k^i be the normalized value of the kth objective for the ith individual in a particular generation. Assuming minimization problems, we have that the jth individual weakly dominates the ith individual if:

$$\min_k(f_k^i - f_k^j) \geq 0 \tag{4.1}$$

The ith individual, in a particular generation, will be weakly dominated by another individual, in the generation, if:

$$\max_{j \neq i}(\min_k(f_k^i - f_k^j)) \geq 0 \tag{4.2}$$

Then, the maximin fitness function of individual i is defined as:

$$\text{fitness}^i = \max_{j \neq i}(\min_k(f_k^i - f_k^j)) \tag{4.3}$$

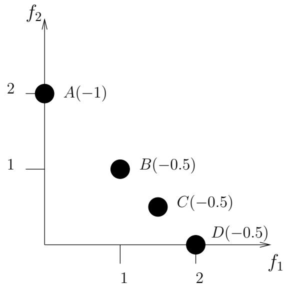


Figure 4.2: We can see that the maximin fitness function penalizes individuals B, C and D because they are close from each other. It also rewards individual A, because it is far away from the other individuals.

where the min is taken over all the objectives from 1 to K, and the max is taken over all the individuals in the population from 1 to $|\mathcal{P}|$, except for the same individual i. From eq. (4.3), we can say the following:

1. Any individual whose maximin fitness is greater than zero is a dominated individual,

2. Any individual whose maximin fitness is less than zero is a nondominated individual.

3. Finally, any individual whose maximin fitness is equal to zero is a weakly-dominated individual.

The procedure to calculate the fitness of each individual of the population is shown in Algorithm 4. The complexity of this algorithm is $O(mN^2)$, where m is the number of objective functions and N is the population size. In Algorithm 4, we multiply the fitness of each individual by -1 in order to obtain a higher fitness for the individuals which are nondominated and a lower fitness for the individuals which are dominated.

The following properties of the maximin fitness function were presented in [5]:

1. The maximin fitness function penalizes clustering of nondominated individuals. In the limit, the maximin fitness of duplicate nondominated individuals is zero. See Figure 4.2.

2. The maximin fitness function rewards individuals at the middle of convex nondominated fronts, see Figure 4.3. Also, it rewards individuals

Algorithm 4: Maximin Fitness Function

```
Input   : Population $\mathcal{P}$
Output: Population $\mathcal{P}$ with the fitness of each individual
foreach $p \in \mathcal{P}$ do
    $maximum \leftarrow -\infty$;
    foreach $q \in \mathcal{P}$ do
        if $p \neq q$ then
            $minimum \leftarrow \infty$;
            foreach Objective function do
                if $p.f_k - q.f_k < minimum$ then
                    $minimum \leftarrow p.f_k - q.f_k$;
                end
            end
            if $minimum > maximum$ then
                $maximum \leftarrow minimum$;
            end
        end
    end
    $p.fitness \leftarrow -1 * maximum$;
end
```

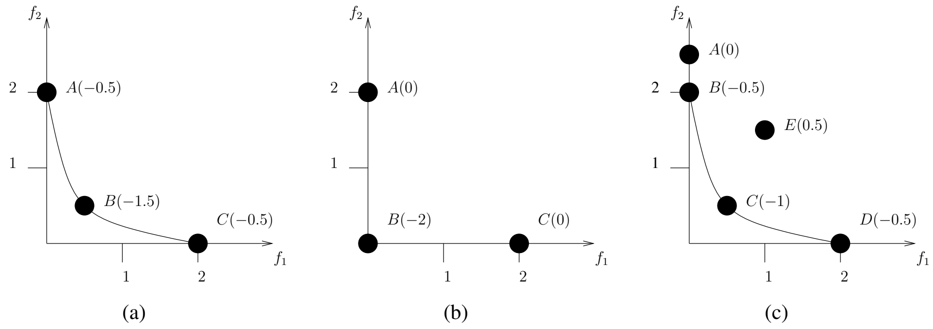


Figure 4.3: In all cases, we can see that the maximin fitness function rewards individuals at the middle of convex nondominated fronts. In (c), individual A has a maximin fitness equal to zero because it is a weakly dominated solution, and individual E has a positive maximin fitness equal to 0.5 because it is a dominated solution.

at the extremes of concave nondominated fronts, see Figure 4.4. The maximin fitness function is a continuous function of objective values.

3. The maximin fitness of dominated individuals is a metric of the distance to the nondominated front. See Figure 4.5.

4. The *max* function in the maximin fitness of a dominated individual is always controlled by a nondominated individual and is indifferent to clustering. The *max* function in the maximin fitness of a nondominated individual may be controlled by a dominated or a nondominated individual. See Figure 4.5.

In Figure 4.5, we can see that the fitness of the nondominated individual B is affected by the dominated individual D. Then, the maximin fitness function penalizes nondominated individuals, if they are close to another individual (no matter whether or not it is a dominated solution). The author of the maximin fitness function proposed in [6] the following modified maximin fitness function:

$$fitness^i = max_{j \neq i, j \in \mathcal{ND}}(min_k(f_k^i - f_k^j)) \tag{4.4}$$

where $\mathcal{ND}$ is the set of nondominated individuals. Using eq. (4.4) to assign the fitness of each individual, we guarantee that the fitness of a nondominated

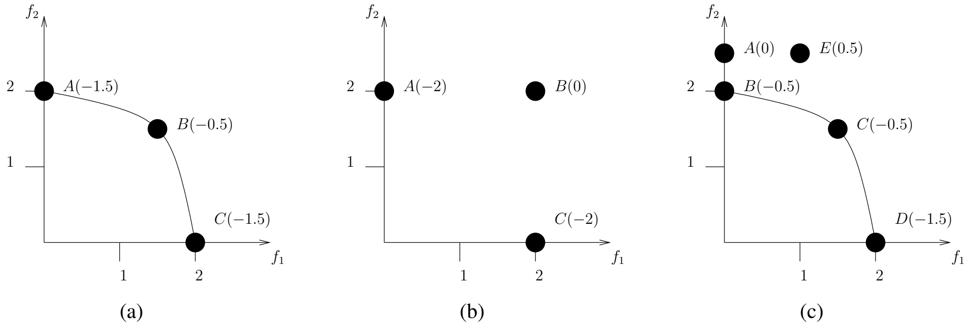


Figure 4.4: In all cases, we can see that the maximin fitness function rewards individuals at the extremes of concave nondominated fronts. In (c), individual A has a maximin fitness equal to zero because it is a weakly dominated solution, and individual E has a positive maximin fitness equal to 0.5 because it is a dominated solution.

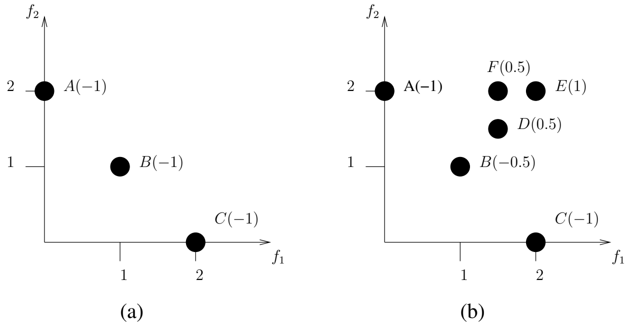


Figure 4.5: In (b), we can see that the fitness of individuals D, E and F is controlled by the nondominated individual B, and the value of their fitness is a metric of the distance to the individual B. Also, we can see that the fitness of B is affected by the dominated individual D.

individual is controlled only by nondominated individuals, and then, we only penalize clustering between nondominated individuals.

MFF and its modified version have been incorporated into evolutionary algorithms such as genetic algorithms [6, 5], particle swarm optimizers [58, 59] and ant colony optimizers [57]. However, in those papers, only low dimensionality MOPs were considered and no extra diversity mechanism was adopted based on the idea that MFF penalizes clustering.

4.3 Decomposition approach

Decomposing a MOP into a number of scalar objective optimization subproblems is the basic idea behind many traditional mathematical programming methods. It is well-known that a Pareto optimal solution of a MOP, under certain conditions, could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all objective functions. Therefore, an approximate Pareto front can be decomposed into a number of scalar objective optimization subproblems. Several methods for constructing aggregation functions can be found in the literature (e.g., see Subsection 2.3.2 in page 16). The most popular ones include the weighted sum approach and Tchebycheff approach. Recently, the boundary intersection methods have also attracted a lot of attention [27].

MOEA/D [89] is the first MOEA that associates each individual solution with some particular scalar optimization problem. It decomposes the MOP into N scalar optimization subproblems and then it simultaneously solves these subproblems using an evolutionary algorithm. At each generation, the best solution found so far for each subproblem survives. Although this MOEA has a low computational cost and is able to find an approximate Pareto front with a good distribution (this depends of the approach used to decompose the MOP), *MOEA/D* always needs to generate a set of well-distributed convex weights and perhaps this is its most important disadvantage because this task is difficult in several cases. There are some proposals to generate these weights [27, 84]. However, none of these techniques ensures obtaining a uniformly distributed set of weights for high dimensionality and they also have some disadvantages. For example, in [27], when we increase the number of objective functions, the number of weights grows according to $\binom{n+p-1}{p}$, where n is the number of objective functions and $p = \frac{1}{\delta}$ (δ is the stepsize). Then, if we use $\delta = 0.1$, this technique generates

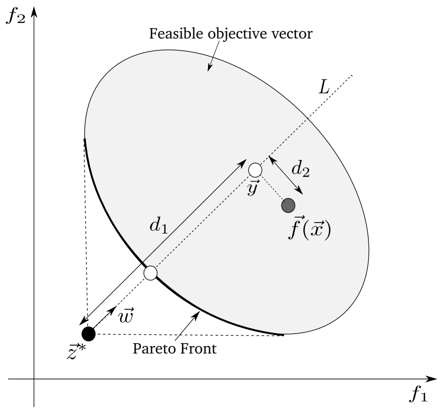


Figure 4.6: Example of the Penalty Boundary Intersection (PBI) approach, using the weighted vector $\vec{w}$ and the solution $\vec{x}$.

66, 286, 1001, 3003, 8008 and 19448 weights for 3, 4, 5, 6, 7 and 8 objective functions, respectively. Since using population sizes as large as these values suggest is not practical, it is necessary to use another technique (for example, clustering) to select a subset of these weights for running the algorithm. Algorithm 5 shows the pseudocode of *MOEA/D*. In [89], the authors of *MOEA/D* describe three approaches to decompose a MOP into a number of scalar optimization problems: weighted sum approach, Tchebycheff approach and boundary intersection (BI) approach. The authors mention that boundary intersection approach obtains better approximations of the Pareto front regarding distribution for two reasons: (i) the resultant optimal solutions in the BI approach should be much more uniformly distributed than those obtained by the Tchebycheff approach and (ii) if $\vec{x}$ dominates $\vec{y}$, it is still possible that $g^{te}(\vec{x}|\vec{w}, \vec{z}^*) = g^{te}(\vec{y}|\vec{w}, \vec{z}^*)$ while it is rare for g^{bi}, where g^{te} is the subproblem generated by the Tchebycheff approach and g^{bi} is the subproblem generated by the BI approach.

4.3.1 Boundary intersection approach

Boundary intersection (BI) approaches were designed for continuous MOPs. Under some conditions and assuming minimization problems, the Pareto optimal front of a continuous MOP is part of the leftmost bottom boundary of its feasible objective region. Geometrically, these BI approaches aim to find intersection points of the leftmost bottom boundary and a set of lines. If

Algorithm 5: MOEA/D

```
Input  : The number of subproblems considered ($N$), a uniform spread of
         $N$ convex weight vectors ($\mathcal{W} = \{\vec{w}^1, \cdots, \vec{w}^N\}$) and the number of
         weight vectors in the neighborhood of each weight vector ($T$).
Output: Approximate Pareto optimal set ($\mathcal{ND}$)
$\mathcal{ND} = \emptyset$;                       //Set of nondominated individuals
Generate an initial population ($\mathcal{P} = \{p^1, \cdots, p^N\}$);
foreach $\vec{w} \in \mathcal{W}$ do
    $B(\vec{w}) = \{\vec{w}^1, \cdots, \vec{w}^T\}$ where $\vec{w}^1, \cdots, \vec{w}^T$ are the $T$ closest weighting
    vectors to $\vec{w}$;                            //Set the neighborhood of $\vec{w}$
end
$\vec{z} = [+\infty, \cdots, +\infty]$;                       //Set the reference vector
while the stopping criterion is not fulfilled do
    foreach $p \in \mathcal{P}$ do
        $i \leftarrow$ Index of subproblem corresponding to $p$;
        Randomly select two subproblems $k$ and $l$ from $B(\vec{w}^i)$ and their
        corresponding solutions $p^k$ and $p^l$, respectively;
        Generate a new individual $p^{new}$ from $p^k$ and $p^l$;   //Using genetic
        operators
        Apply a problem-specific repair/improvement heuristic on $p^{new}$ to
        produce $p^{new'}$;
        foreach objective function do
            if $z_k < p^{new'}.f_k$ then
                $z_k = p^{new'}.f_k$;
            end
        end
        foreach $\vec{w}^j \in B(\vec{w}^i)$ do
            //$g$ is the subproblem generated with the weight
              vector $\vec{w}^j$
            if $g(p^{new'}.\vec{f}, \vec{w}^j, \vec{z}) \leq g(p.\vec{f}, \vec{w}^j, \vec{z})$ then
                $p \leftarrow p^{new'}$;
            end
        end
        Remove from $\mathcal{ND}$ all the individuals dominated by $p^{new'}$;
        Add $p^{new'}$ to $\mathcal{ND}$ if no individuals in $\mathcal{ND}$ dominate $p^{new'}$;
    end
end
```

these lines are evenly distributed in a sense, one can expect that the resultant intersection points provide a good approximation to the whole Pareto optimal front. The penalty boundary intersection (PBI) method can be classified as a BI approach and it is defined as follows:

$$g^{bip}(\vec{x}|\vec{w}, \vec{z}^{*}) = d_1 + \theta d_2 \quad \text{subject to} \quad \vec{x} \in \Omega \tag{4.5}$$

where $d_1 = \frac{||(\vec{z}^{*} - \vec{f}(\vec{x}))^T \vec{w}||}{||\vec{w}||}$ and $d_2 = ||\vec{f}(\vec{x}) - (\vec{z}^{*} - d_1 \vec{w})||$. $\theta > 0$ is a preset penalty parameter. Let $\vec{y}$ be the projection of $\vec{f}(\vec{x})$ on the line L generated by the weighted vector $\vec{w}$. As shown in Figure 4.6, d_1 is the distance between $\vec{z}^{*}$ and $\vec{y}$ and d_2 is the distance between $\vec{f}(\vec{x})$ and $\vec{y}$.

4.4 Indicator-based approach

Since classical MOEAs based on Pareto dominance have difficulties when we increase the number of objective functions, MOEAs based on quality indicators have been popular in recent years. Perhaps, the most popular has been the hypervolume indicator (I_H) because it is the only unary indicator that is known to be strictly Pareto compliant (see Subsection 3.2.1 in page 28). I_H is defined as the size of the space covered by the Pareto optimal solutions and the contribution to I_H of a solution $\vec{z}$ reflects the influence of that point on the quality of the approximate Pareto front. The contribution of a solution $\vec{x}$ to I_H is defined as follows:

$$C_H(\vec{z}, \mathcal{A}) = I_H(\mathcal{A}, \vec{y}_{ref}) - I_H(\mathcal{A} \setminus \vec{z}, \vec{y}_{ref}) \tag{4.6}$$

where $\vec{z} \in \mathcal{A}$. Then, the contribution of $\vec{z}$ is the space that is only covered by $\vec{z}$. Figure 4.7 illustrates these two concepts. The main disadvantage of I_H is its high computational cost, since the problem of computing C_H is *NP-hard* (i.e., C_H cannot be computed exactly in polynomial time in the number of objective functions unless $P = NP$ [11]).

There are several MOEAs based on I_H. Knowles and Corne [52] used a bounded archive to save the nondominated solutions found at each generation. When the archive was full and Pareto dominance could no longer discard solutions then, they proposed to use I_H as follows: calculate the contribution of each solution to I_H; if the contribution of the new solution was better than the contribution of the solution with the worst contribution,

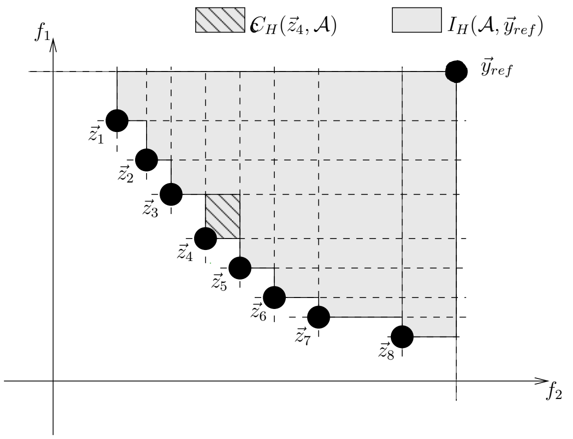


Figure 4.7: Let $\mathcal{A} = \{\vec{z}_1, \vec{z}_2, \cdots, \vec{z}_8\}$ be the approximation of the Pareto optimal set and $\vec{y}_{ref}$ be the reference point. Then, the gray area is the hypervolume of set $\mathcal{A}$ and the hatched area is the contribution to the hypervolume of the solution $\vec{z}_4$.

then the new solution would replace it; otherwise, the archive would remain the same.

Zitzler and Künzli [91] proposed a general selection mechanism based on a performance indicator. The general algorithm was called *IBEA*, and it assigns the fitness of each individual using: $F(\vec{z}) = \sum_{\vec{y} \in P \setminus \{\vec{z}\}} -e^{-I(\{\vec{y}\},\{\vec{z}\})/k}$, where P is the population, $\vec{z}, \vec{y} \in P$ and k is a scalar fitness factor. When the worst individual is eliminated, IBEA updates the fitness of all other individuals with: $F(\vec{z}) = F(\vec{z}) + e^{-I(\{\vec{z}'\},\{\vec{z}\})/k}$, where $\vec{z}'$ is the eliminated individual.

Emmerich et al. [35] proposed an algorithm based on *NSGA-II* and the archived strategies proposed by Knowles, Corne and Fleisher. They called it *SMS-EMOA*. *SMS-EMOA* creates an initial population and in the following, it generates only one solution by iteration using the operators (crossover and mutation) of the *NSGA-II*. After that, it applies Pareto ranking. When the last front has more than one solution, *SMS-EMOA* uses the contribution to I_H to decide which solution will be removed. Beume et al. [8] proposed not to use the contribution to I_H when in the Pareto ranking we obtain more than one front. In that case, they proposed to use the number of solutions which dominate to one solution (the solution that is dominated by more solutions is removed). The authors argue that the motivation to use I_H is to improve the distribution in the true Pareto front and then it is not necessary to perform such improvement in fronts different of the true Pareto front. Perhaps, this is the best known MOEA based on I_H. Algorithm 6 shows the version of the *SMS-EMOA* algorithm proposed in [8].

Algorithm 6: SMS-EMOA

```
$t \leftarrow 0$;
Generate an initial population ($\mathcal{P}(t)$);
while the stopping criterion is not fulfilled do
    $q \leftarrow$ Generate a new individual from $\mathcal{P}(t)$;
    $\mathcal{Q} \leftarrow \mathcal{P}(t) \cup \{q\}$;
    $\{\mathcal{F}_1, \cdots, \mathcal{F}_v\} \leftarrow$ Fast-nondominated-sort($\mathcal{Q}$);
    if $v > 1$ then
        $q \leftarrow$ Choose the individual which is dominated by more solutions;
    else
        $q \leftarrow$ Choose the individual with the worst contribution to $I_H$;
    end
    $\mathcal{P}(t+1) \leftarrow \mathcal{Q} \setminus q$;
    $t \leftarrow t + 1$;
end
```

Igel et al. [50] used an evolution strategy with Pareto ranking as a primary criterion selection and crowding or hypervolume as a second selection criterion. This approach operates in a similar way to *SMS-EMOA*. Mostaghim [63] designed a MOEA based on particle swarm optimization in which I_H was used in the leader selection mechanism.

All the above approaches have an important disadvantage. It is well known that the computation of C_H has a high computational cost. When all the individuals are nondominated, all the above approaches need to calculate $N + 1$ (where N is the size of the population) times the contribution to I_H. This turns these approaches impractical when we want to solve MOPs with many objective functions. Because of this, some researchers have proposed techniques to approximate I_H or the contribution to I_H. In the next chapter, we will study them and also we will make a new proposal to address this problem.

As we mentioned in previous chapters, there are other indicators like the $R2$-indicator (I_{R2}) or the Δ_p-indicator (I_{Δ_p}). After the study on the properties of I_{R2} presented by Brockhoff et al. [14], a number of proposals of MOEAs based on I_{R2} have been introduced [78, 45, 66, 83]. Although I_{R2}-based MOEAs can solve MOPs with many objective functions at an affordable computational cost, this type of algorithms also has an important disadvantage: They need to generate a set of well-distributed convex weights and this task becomes more difficult as we increase the number of objective functions.

Recently, I_{Δ_p} was introduced [73] and some MOEAs based on it have already been proposed [41, 68, 33]. As we saw in Chapter 3, I_{Δ_p} is composed of slight modifications of two well-known indicators: generational distance (I_{GD}) [79] and inverted generational distance (I_{IGD}) [19]. It is well-known that for computing I_{GD} and I_{IGD}, it is necesary to know the true Pareto front. Therefore, the most important disadvantage of MOEAs based on I_{Δ_p} is perhaps that they need a reference set which must contain well-distributed solutions. Not being able to produce a good reference set could produce a diversity loss in the population which might cause that the algorithm cannot generate the complete Pareto front, or that it generates poorly distributed solutions. In extreme cases, the lack of an appropriate reference set could event prevent convergence.

Chapter 5

New Selection Mechanism Based on the Hypervolume Indicator

During the last 10 years, MOEAs based on the hypervolume indicator have become relatively popular. However, the hypervolume indicator has one important disadvantage: its high computational cost. Because of this, MOEAs based on the hypervolume normally become impractical when we want to solve MOPs with many objectives. In order to address this problem, some researchers have proposed to approximate the hypervolume indicator or the contribution to it. In this chapter, we study some of these approximation techniques and also an interesting property of the hypervolume called "locality". Finally, we propose a new selection mechanism based on the hypervolume indicator and on its locality property and we also show that we can obtain better results if we use it in combination with a scheme to approximate the contribution to the hypervolume than if we use the traditional selection mechanism adopted by the well-known *SMS-EMOA* (also using a scheme to approximate the contribution to the hypervolume).

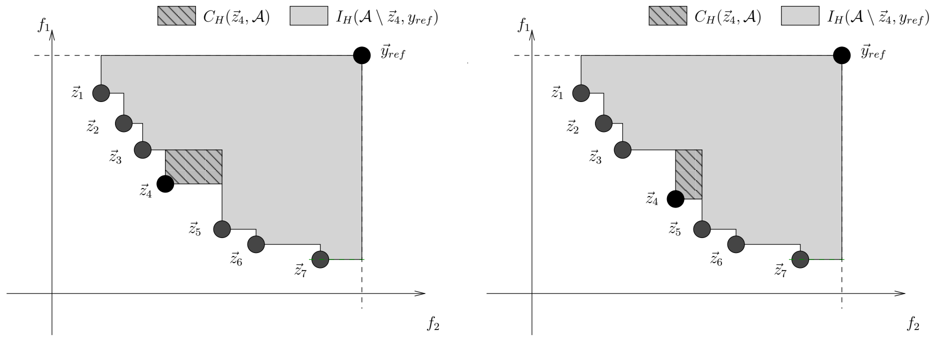


Figure 5.1: Let $\mathcal{A} = \{\vec{z}_1, \vec{z}_2, \cdots, \vec{z}_7\}$ be the approximate Pareto front. If we move $\vec{z}_4$ between $\vec{z}_3$ and $\vec{z}_5$, the covered space by $\{\mathcal{A} \setminus \vec{z}_4\}$ is not affected and only the contribution to the hypervolume of $\vec{z}_4$ is affected.

5.1 Hypervolume indicator

As we mentioned in Chapter 3, the hypervolume indicator, I_H, is defined as follows: If Λ denotes the Lebesgue measure, I_H is defined as:

$$I_H(A, \vec{z}_{ref}) = \Lambda\left(\bigcup_{\vec{z}\in\mathcal{A}} \{\vec{y} \mid \vec{z} \prec \vec{y} \prec \vec{z}_{ref}\}\right) \tag{5.1}$$

where $\vec{z}_{ref} \in \mathbb{R}^k$ denotes a reference point that should be dominated by all the Pareto optimal points. A high I_H value, indicates that $\mathcal{A}$ is close to $\mathcal{PF}$ and has a good spread towards the extreme portions of the $\mathcal{PF}$. This is the only unary indicator that is known to be **strictly Pareto compliant**.

Auger et al. [2] conducted a study about the optimal μ-distributions and the choice of the reference point in the hypervolume indicator. They mentioned one interesting property of this indicator when $k = 2$ (two objective functions), called **locality** which says: *given three consecutive points on the Pareto front, moving the middle point will only affect the hypervolume contribution that is solely dedicated to this point, but the joint hypervolume contribution remains fixed.* See Figure 5.1. Also, Auger et al. conducted a similar study for $d = 3$ in [1] and they mentioned that the optimal placement of a single solution is not determined by only two neighbors, anymore, as it is the case for $d = 2$.

In recent years, there have been several proposals to incorporate the hypervolume into a MOEA. However, in most cases, the MOEAs use the same competition scheme: if we have a population $\mathcal{P}$ and a new individual p_{new}, we calculate the contribution to the hypervolume of each individual in $\mathcal{P}$ and the contribution of the new individual. If $p_{new}.\vec{f}$ is better than $p_{worst}.\vec{f}$ (according to the contribution), p_{new} replaces p_{worst}. Otherwise, the population remains the same. See Section 4.4 in page 49.

As we mentioned in the previous chapter, if we use the above competition scheme into a MOEA, we need to calculate $|\mathcal{P}| + 1$ contributions to the hypervolume indicator, and then, this MOEA won't be able to deal with MOPs with more than five objective functions (a hypervolume-based MOEA requires about eight hours per run to solve a MOP with five objective functions). In order to address this problem, Bradstreet et al. [10] proposed a method to calculate the contribution to the hypervolume indicator of each solution in a fast way without calculating the hypervolume for each solution. The main idea is the following: when we eliminate one solution of the population, not all the contributions of the other solutions are affected. Emmerich and Fonseca [36] proposed a dimension sweep algorithm for computing all contributions to the hypervolume in three dimensions with a time complexity equal to $O(n \log n)$. Also, they showed that for $k > 3$ (more than three objective functions), the time complexity is bounded below by $\Omega(n \log n)$. It is important to note that this a lower bound but the calculation of the minimal contribution is an **NP-hard** [12] problem.

Other authors have chosen to approximate the contribution to the hypervolume. For example, Ishibuchi et al. [51] proposed using a number of achievement scalarizing functions with uniformly distributed weight vectors to approximate the hypervolume. They measure the distance from the reference point to the solution set, using scalarizing functions, see Figure 5.2.

Bader and Zitzler [3] proposed to assign a fitness to each individual using an approximation of the hypervolume based on the idea that is not necessary to know the exact contribution to the hypervolume of each solution, since we only aim to obtain a good ranking of the solutions in the population. The technique that they use to assign fitness to each individual is not easy because they do not consider only the contribution to the hypervolume as we defined in eq. (4.6), but also all the space dominated by one solution, see Figure 5.3. They used Monte Carlo simulation to approximate the dominated regions and then assign fitness. Also, they proposed a method to remove m individuals from a population $\mathcal{P}$ considering the expected loss in hypervolume that can be attributed to a particular solution when exactly m solutions are removed, see Figure 5.4.

Bringmann and Friedrich [12] indicated that it is not necessary to calculate the hypervolume of all the individuals in the population, in order to know the hypervolume contribution of a single solution. Let $\mathcal{A}$ be the approximate Pareto front; they proposed to approximate the contribution of a solution $\vec{z} \in \mathcal{A}$ as follows: Let $BB_{\vec{z}}$ be the bounding box of $\vec{z}$; then, we do a random sampling in $BB_{\vec{z}}$. For each random point, we have to check if it is uniquely dominated by $\vec{z}$, and then we can approximate the contribution of $\vec{z}$ using:

$$\tilde{C}_H(\vec{z}, \mathcal{A}) = \frac{SuccessSamples}{Samples} VOL(BB_{\vec{z}}) \tag{5.2}$$

where $SuccessSamples$ is the number of samples that were only dominated by $\vec{z}$ (there does not exist another $\vec{y} \in \mathcal{A}$ such that $\vec{y}$ dominates the sample) and $Samples$ is the total number of samples. See Figure 5.5. To determine $BB_{\vec{z}}$, we construct a bounding box, $B_{\vec{y}}$, for each solution $\vec{y} \in \mathcal{A}$, using the reference point $\vec{y}_{ref}$ as in Figure 5.5. Then, we can cut $BB_{\vec{z}}$ as follows: start with the box $B_{\vec{z}}$ itself, iterating over all other boxes $B_{\vec{y}}$, such that $\vec{z} \neq \vec{y}$. If $B_{\vec{y}}$ dominates $B_{\vec{z}}$ in all but one dimension, then we can cut the bounding box, $B_{\vec{z}}$, in the nondominated dimension to obtain $BB_{\vec{z}}$. See Figure 5.6.

5.2 A New Selection Mechanism based on the Hypervolume and its Locality Property

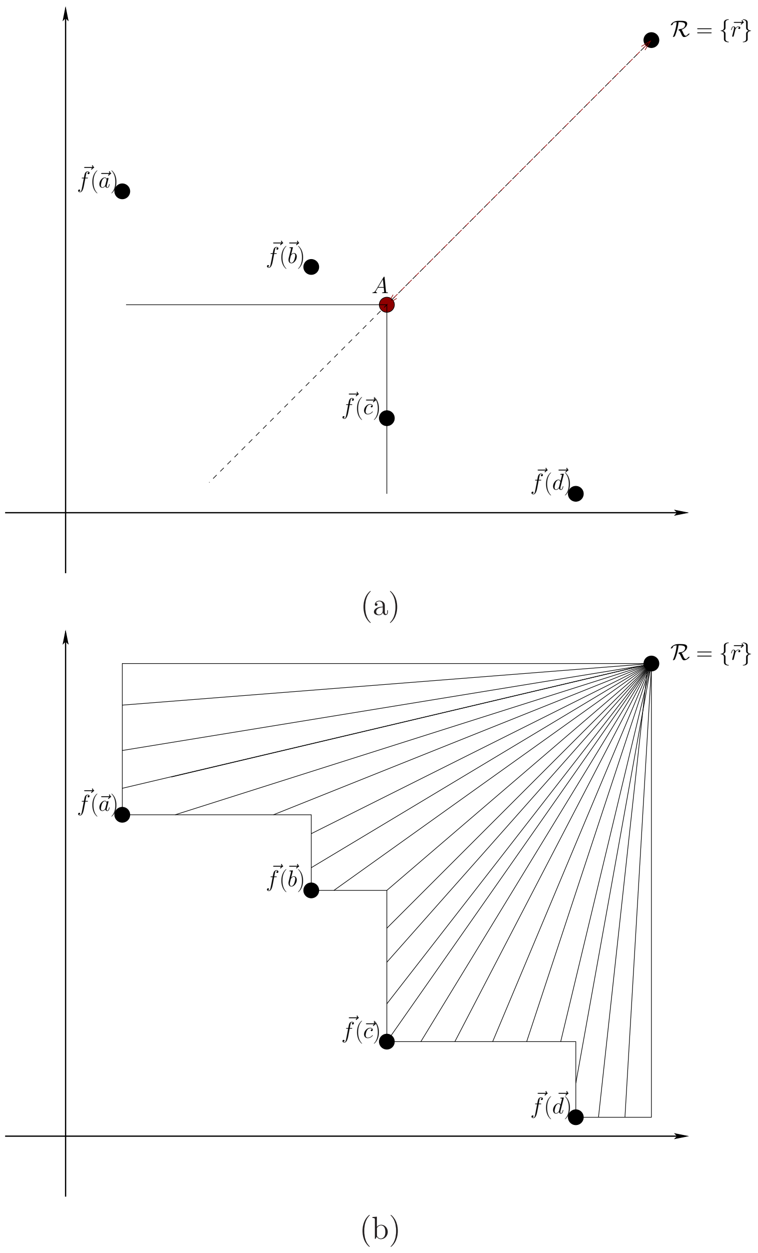
Here, we propose a new selection mechanism that exploits the locality property of the hypervolume indicator as follows. Let's assume that at each iteration of a MOEA, only one solution p_{new} is created and the current population is $\mathcal{P}$. After that, we calculate the Euclidean distance of the new solution to each solution in the current population:

$$dist_i = \|p_i.\vec{f} - p_{new}.\vec{f}\| \text{ such that } p_i \in \mathcal{P} \tag{5.3}$$

and, we choose the nearest solution:

$$p_{near} \text{ such that } dist_{near} = \min dist_i \tag{5.4}$$

where $i = \{1, \cdots, |\mathcal{P}|\}$. These two solutions (the new solution, p_{new}, and its nearest neighbor, p_{near}) compete to survive. The core idea is to move a solution within its neighborhood with the aim of improving its contribution



(a)

(b)

Figure 5.2: In (a) the dotted line represents a scalarizing function. Let $\mathcal{A} = \{\vec{a}, \vec{b}, \vec{c}, \vec{d}\}$ be the approximation of the Pareto optimal set and $\vec{r}$ the reference point. Ishibuchi et al. [51] propose to measure the distance from $\vec{r}$ to A. In (b) many uniformly distributed weight vectors are used and the average length is taken as an approximation of the hypervolume.

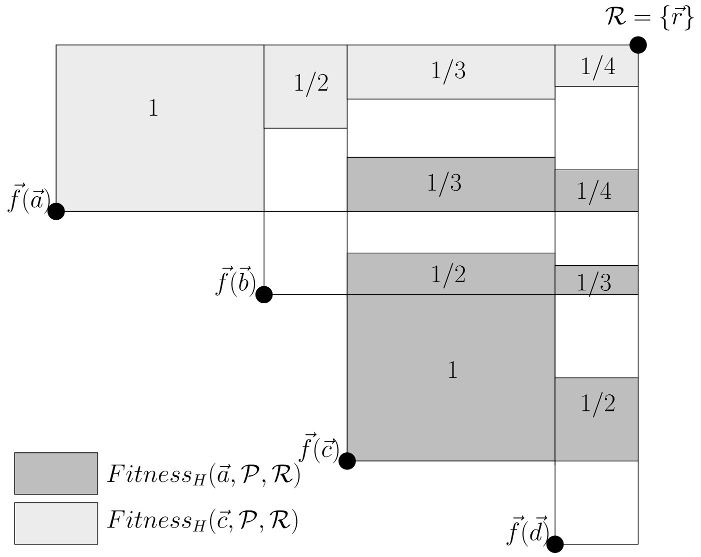

$\mathcal{R} = \{\vec{r}\}$
1
1/2
1/3
1/4
1/3
1/4
1/2
1/3
1
1/2
$\vec{f}(\vec{a})$
$\vec{f}(\vec{b})$
$\vec{f}(\vec{c})$
$\vec{f}(\vec{d})$
$Fitness_H(\vec{a}, \mathcal{P}, \mathcal{R})$
$Fitness_H(\vec{c}, \mathcal{P}, \mathcal{R})$

Figure 5.3: In this Figure, we illustrate the fitness assignment proposed by Bader and Zitzler [3]. The space dominated by the four solutions ($\vec{a}$, $\vec{b}$, $\vec{c}$ and $\vec{d}$) is divided in regions. Let's assume that we want to calculate the fitness of individuals $\vec{a}$ and $\vec{c}$. Regions labeled with number 1 indicate that this portion of the space is only dominated by the solution $\vec{a}$ or $\vec{c}$. Therefore, this region is attributed to $\vec{a}$ or $\vec{c}$. Regions labeled with 1/2 indicate that this portion of the space is dominated by two solutions, and then, to each of these two solutions, it corresponds half of this region. Regions with 1/3 and 1/4 indicate that the portion is dominated by three and four solutions, respectively. Thus, for each of them, it corresponds a third or a fourth of this region.

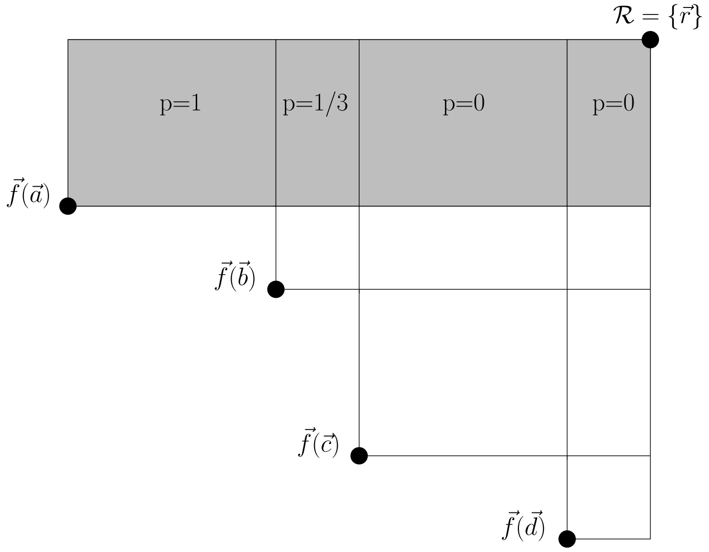


Figure 5.4: This Figure shows the probability p that a dominated region is lost if solution $\vec{a}$ is removed together with any other solution ($m = 2$). It is interesting to look at the region with probability $p = 1/3$. If we remove solution $\vec{a}$, then only solution $\vec{b}$ can dominate this region and the probability of choosing $\vec{b}$ is 1/3. Regions with probability $p = 0$ indicate that if we remove $\vec{a}$ together with any solution, this region is still dominated by one of the remaining solutions.

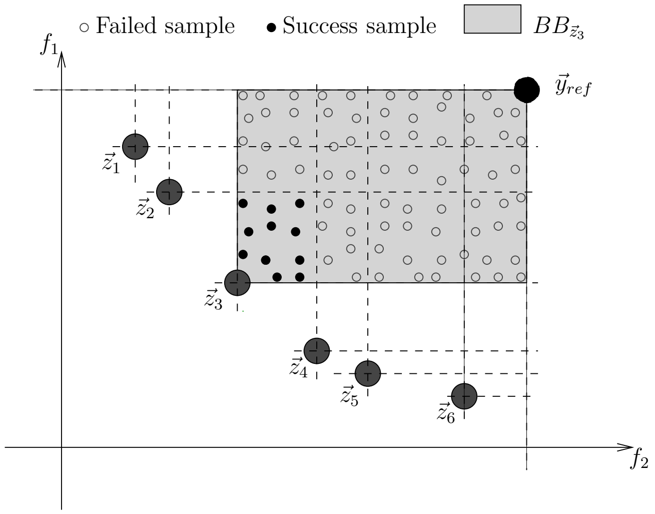
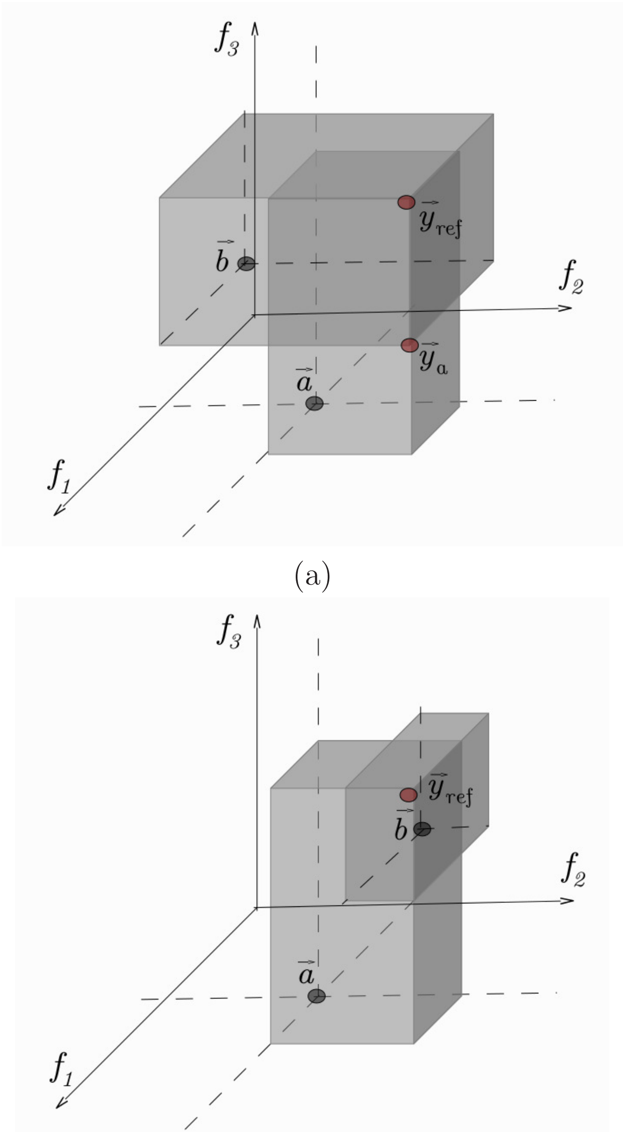


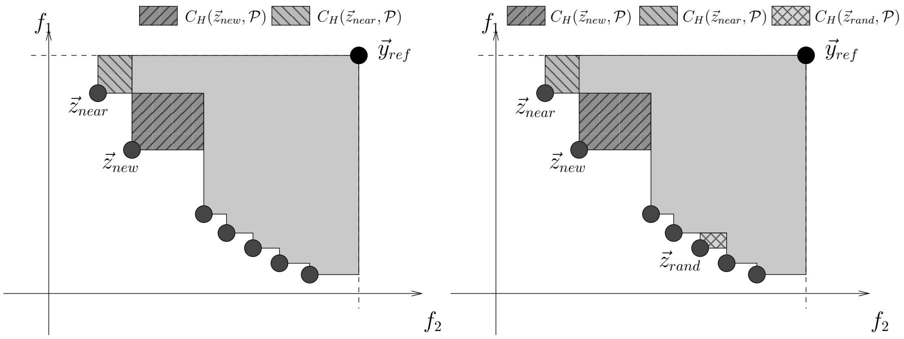
Figure 5.5: Let $\mathcal{A} = \{\vec{z}_1, \vec{z}_2, \vec{z}_3, \vec{z}_4, \vec{z}_5, \vec{z}_6\}$ be the approximate Pareto front and $\vec{y}_{ref}$ be the reference point. We approximate the contribution of solution $\vec{z}_3$ as follows. We construct the bounding box $BB_{\vec{z}_3}$ from the reference point $\vec{y}_{ref}$. After that, we generate random points in $BB_{\vec{z}_3}$. The black points into $BB_{\vec{z}_3}$ are success samples; these points are only dominated by $\vec{z}_3$. The remaining random points in $BB_{\vec{z}_3}$ are also dominated by other points and therefore they are failed samples. Finally, the contribution is approximately $\frac{SuccessSamples}{SuccessSamples+FailedSamples}VOL(BB_{\vec{z}_3})$. It is important to mention that this is an example simply for illustrating the procedure by which we can approximate the hypervolume contribution of a solution. However, for two dimensions, we can calculate the exact contribution, when we execute the procedure to cut the $BB_{\vec{z}_3}$.



(a)

(b)

Figure 5.6: In (a), we can see that $\vec{a}$ is dominated by $\vec{b}$ in all objective functions except for f_3. Then, we can cut the bounding box of $\vec{a}$ and use $\vec{y}_a$ instead of $\vec{y}_{ref}$ because the cut region is completely covered by $\vec{b}$. In (b), $\vec{b}$ is better than $\vec{a}$ only in f_1. Therefore, we cannot move the reference point and cut the bounding box because the cut region is not completely covered by $\vec{b}$.


▨ $C_H(\vec{z}_{new},\mathcal{P})$
▧ $C_H(\vec{z}_{near},\mathcal{P})$
f_1
$\vec{y}_{ref}$
$\vec{z}_{near}$
$\vec{z}_{new}$
f_2
▨ $C_H(\vec{z}_{new},\mathcal{P})$
▧ $C_H(\vec{z}_{near},\mathcal{P})$
▦ $C_H(\vec{z}_{rand},\mathcal{P})$
f_1
$\vec{y}_{ref}$
$\vec{z}_{near}$
$\vec{z}_{new}$
$\vec{z}_{rand}$
f_2

(a) (b)

Figure 5.7: In (a), $\vec{z}_{new}$ competes only with its nearest neighbor $\vec{z}_{near}$ and then $\vec{z}_{near}$ is eliminated. In (b), we choose $\vec{z}_{rand}$ randomly. Then, $\vec{z}_{new}$, $\vec{z}_{near}$ and $\vec{z}_{rand}$ compete to survive and $\vec{z}_{rand}$ is eliminated.

to the hypervolume. See Figure 5.7(a). However, we must consider the case in which the new solution is located in an unexplored region (a region with few solutions) as shown in Figure 5.7(a). In this case, it is not a good idea to remove the new solution or its nearest neighbor. To address this problem, we propose to choose randomly another solution, p_{rand}, such that $p_{rand} \neq p_{new}$ and $p_{rand} \neq p_{near}$. Then p_{rand} will also compete with the other two (p_{new} and p_{near}). This is considering that the probability of choosing a solution in an unexplored region is low. See Figure 5.7(b).

To validate our selection mechanism, we decided to incorporate it into *SMS-EMOA* [8]. In Figure 5.8, we can see the results obtained by our selection mechanism, without using the random solution p_{rand}, in the WFG1 and WFG4 test problems [48], with two objective functions and twenty-four decision variables. For this experiment, we used 100 individuals and 500 generations.[1] In this Figure, we can see that the modified *SMS-EMOA* is able to generate a good approximation of the Pareto optimal front. However, for the WFG1 test problem, this approach can only produce a portion of the Pareto optimal front and in the WFG4 test problem, some solutions are dominated and the nondominated solutions are not well-distributed. In Figure 5.9, we can see the results using the random individual. In these figures, we can note that if we use a random solution, the distribution of the solutions on the Pareto front significantly improves. Therefore, if we use

[1] Each time we generate 100 individuals; this is considered as one generation.

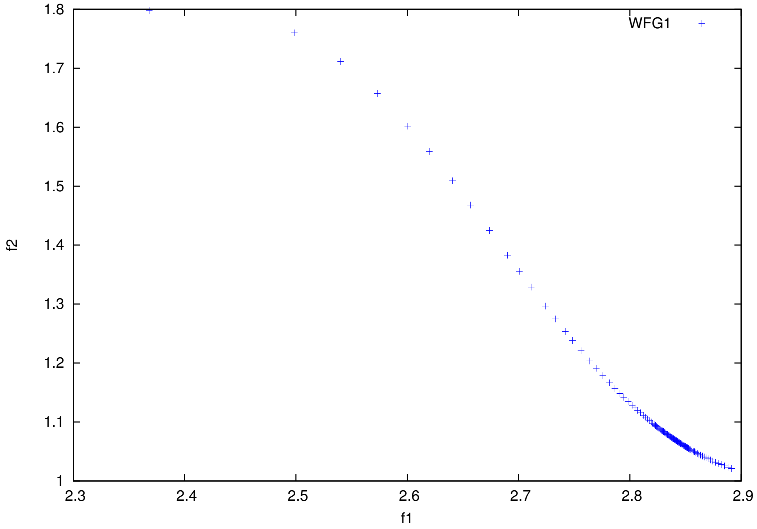
our proposed selection mechanism, we can significantly improve the computational time required, since we would need to compute the contribution to the hypervolume only three times for each iteration (regardless of the population size adopted) unlike *SMS-EMOA* which, in the worst case needs to compute $|\mathcal{P}| + 1$ contributions. Because of this, the proposed approach is called **improved SMS-EMOA** (*iSMS-EMOA*).

We studied the behavior of our selection mechanism with respect to the random solution. Specifically, we wanted to know: how many times is the random solution eliminated? For this, we performed 30 independent runs for the test problems: DTLZ1, DTLZ2 and DTLZ6 [30] with three-objective functions, seven decision variables for DTLZ1 and DTLZ6 and twelve decision variables for DTLZ2. Also, we used WFG1, WFG3 and WFG4 [48] with three-objective functions and twenty-four decision variables. Per run, we calculate at each generation the number of times in which the random solution is eliminated. Figures 5.10 and 5.11 show that the random solutions are eliminated at the begining of the search process. And, as the search process progresses, the new solution or its nearest neighbor is eliminated more often.

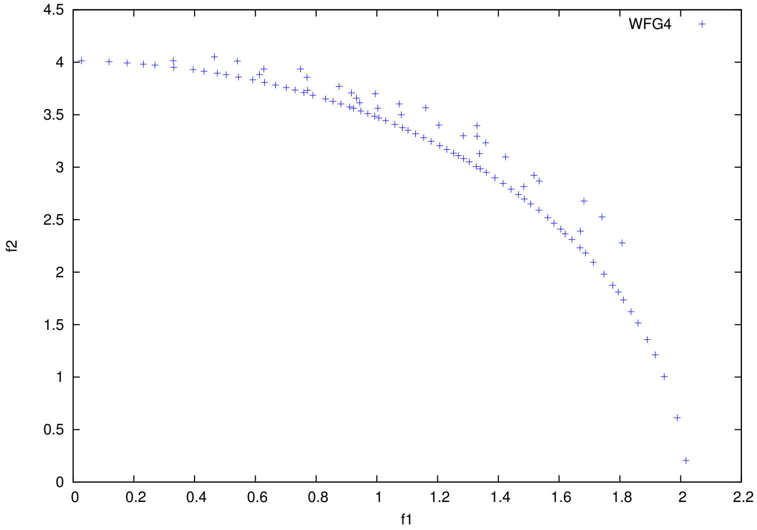
Although in [1] it is shown that, for three dimensions, the optimal placement of a single solution is not determined by only two neighbors, the competition scheme based on I_H and its locality property still works. This is because this selection scheme does not need to know the entire neighborhood, it only considers to move one solution in the direction corresponding to its nearest neighbor. Therefore, it is not important if the optimal placement of one solution is determined by many (even all) solutions of the population. Therefore, we can claim the following: while the Pareto front is being generated, we need of the randomness for exploring the entire Pareto front. Later on, we only need to exploit the "locality property" to maximize the hypervolume indicator and improve the distribution on the Pareto front.

5.2.1 Experimental results

We validate the proposed selection mechanism comparing our *iSMS-EMOA* with respect to *SMS-EMOA* [8] and a version of the original *SMS-EMOA* that uses the fitness assignment scheme proposed in [3] instead of calculating

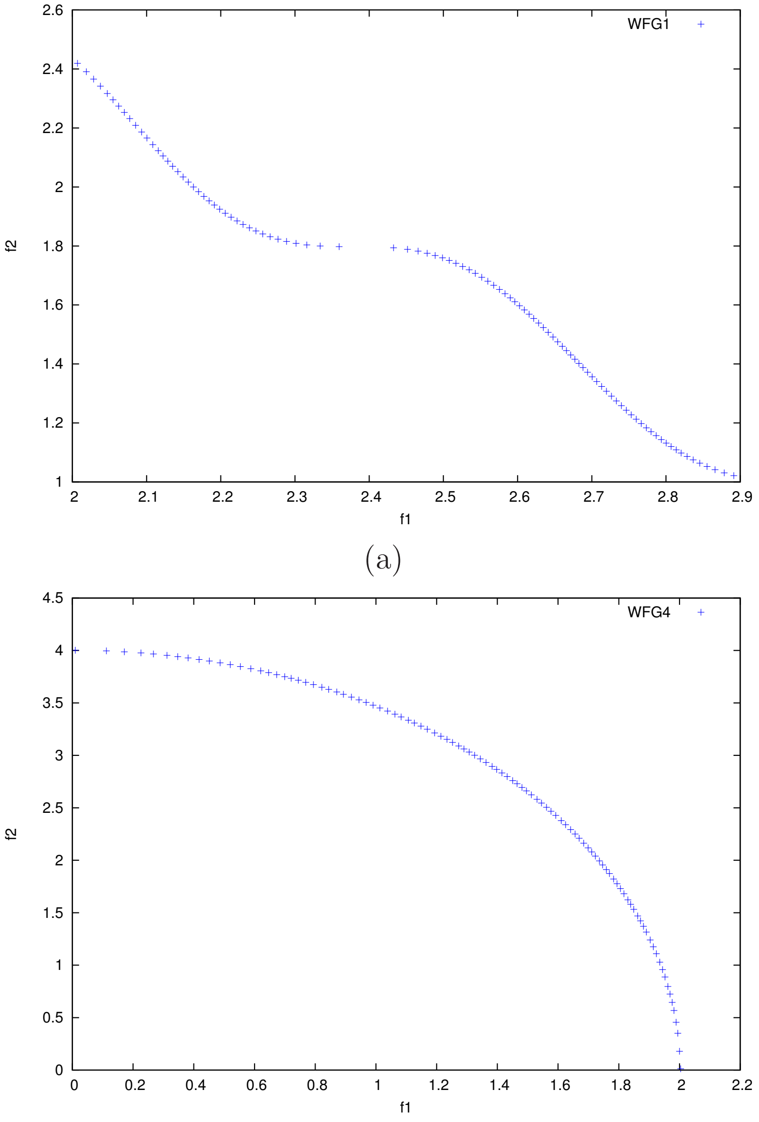


(a)



(b)

Figure 5.8: Pareto fronts obtained with the modified *SMS-EMOA* algorithm without using the random solution, in the WFG1 and WFG4 test problems with two objective functions and 24 variables.



(a)

(b)

Figure 5.9: Pareto fronts obtained with the modified *SMS-EMOA* algorithm using the random solutions, in the WFG1 and WFG4 test problems with two objective functions and 24 variables.

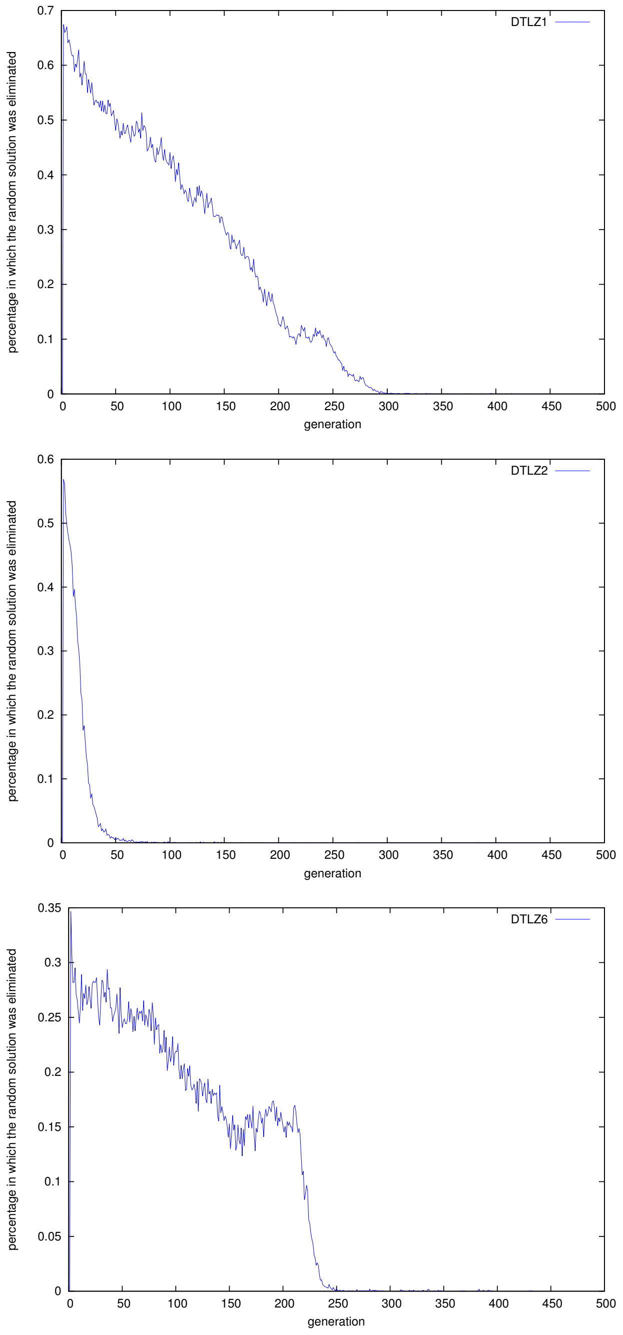


Figure 5.10: Mean of 30 independent runs corresponding to the percentage of times in which the random solution was eliminated. We considered problems DTLZ1, DTLZ2, DTLZ6, all of them with three objective functions.

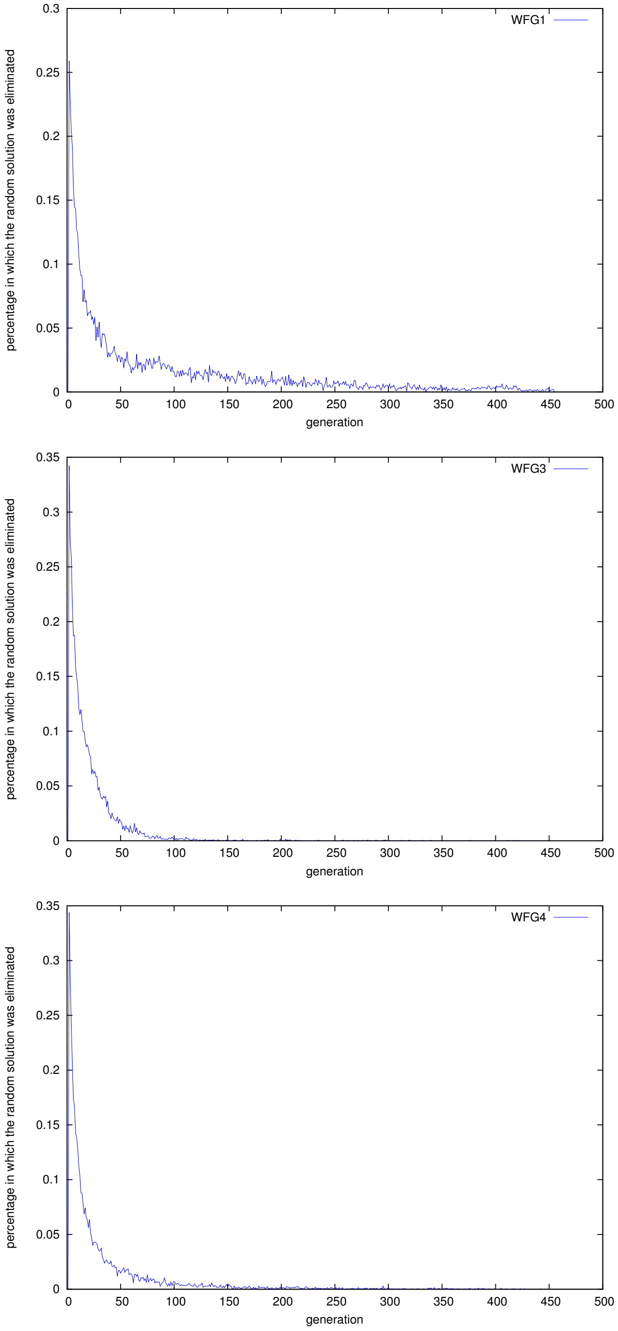


Figure 5.11: Mean of 30 independent runs corresponding to the percentage of times in which the random solution was eliminated. We considered problems WFG1, WFG3 and WFG4, all of them with three objective functions.

the exact hypervolume contributions,[2] and which we called "HyPE version of *SMS-EMOA* (*hypeSMS-EMOA*)". For this sake, we used the source code of HyPE available in the public domain [3] adopting 10^4 as our number of samples.

For our experiments, we used problems with up to five objective functions, seven of which were taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [30] and seven more were taken from the Walking Fish Group (WFG) toolkit [48]. We used $k = 5$ for DTLZ1, DTLZ3 and DTLZ6 and $k = 10$ for the remaining test problems. We used $k_{factor} = 2$ and $l_{factor} = 10$ for the WFG test problems. In Appendix A, we describe these two sets of test problems. For each test problem, we performed 30 independent runs. For all algorithms, we adopted the parameters suggested by the authors of NSGA-II: $p_c = 0.9$ (crossover probability), $p_m = 1/n$ (mutation probability), where n is the number of decision variables. For the crossover and mutation operators, we adopted $\eta_c = 15$ and $\eta_m = 20$, respectively. We performed a maximum of 50,000 fitness function evaluations (we used a population size of 100 individuals and we iterated for 500 generations). However, *we adopted four hours as our maximum running time* because we know that the computation of the exact hypervolume contribution has a high computational cost. All MOEAs considered in our experiments were compiled using the GNU C compiler and they were executed on a computer with a 2.66GHz processor and 4GB in RAM.

Performance indicators

We adopted only I_H to validate our results because it rewards both convergence towards the Pareto front as well as the maximum spread of the solutions obtained. Also, all of the algorithms used in this comparison have as their aim to maximize the hypervolume and, therefore, it makes sense to use this indicator to assess their performance. To calculate the hypervolume indicator, we normalized the approximations of the Pareto optimal set, generated by the MOEAs, and we used $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$ as our reference point. The normalization was performed considering all approximations generated by the different MOEAs (i.e., we place, in one set, all nondominated solutions found by *SMS-EMOA*, *hypeSMS-EMOA* and *iSMS-*

[2] It is important to mention that our aim was to validate the selection mechanism. Therefore, we decided to use the same MOEA in all cases and we only changed the selection mechanism. For this reason, we did not use the original HyPE.

EMOA and from this set we calculate the maximum and minimum for each objective function).

Discussion of results

First, we will review the results of the DTLZ test problems. In Table 5.1, we can observe that our *iSMS-EMOA* obtained the first place in twelve problems, it had the second place in nine problems and it was never in the third place. The original *SMS-EMOA* obtained the first place in nine problems, it was in second place in twelve problems and it was never in the third place. Finally, we can see that the version in which the contribution to the hypervolume is approximated obtained the third place in all the problems adopted. Regarding the WFG test problems, we can see in Table 5.3 that our *iSMS-EMOA* had the first place in nine problems and it had the second in twelve problems. The original *SMS-EMOA* had the first place in twelve problems and it had the second in nine problems. Finally, *hypeSMS-EMOA* had again the third place in all problems. We conducted an statistical analysis using Wilcoxon's rank sum to validate our experiments, see Tables 5.5 and 5.6. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level. From Tables 5.5 and 5.6, we can assess that our *iSMS-EMOA* outperformed *hypeSMS-EMOA* significantly in all problems because in all cases we can reject the null hypothesis "medians are equal". With respect to *SMS-EMOA*, we can see that in fourteen problems our *iSMS-EMOA* outperformed it significantly, *iSMS-EMOA* was outperformed significantly in nine cases and in nineteen problems both MOEAS had a similar behavior. From these results, we can say that *iSMS-EMOA* is better than *hypeSMS-EMOA* and it is competitive with respect to *SMS-EMOA* (in fact, we can say that it is better because it was better in most problems). Also, we can observe that the quality of the approximate Pareto front decreases if we approximate the hypervolume. Although, in some cases, *hypeSMS-EMOA* performed more objective function evaluations than *SMS-EMOA* and *iSMS-EMOA* (this algorithm was not allowed to exceed the maximum running time of 4 hours imposed for the other approaches), it was unable to obtain better results than any of the other approaches.

However, if we consider the running time that each MOEA required to obtain the approximate Pareto front, see Tables 5.2 and 5.4, *hypeSMS-EMOA* is the fastest in both sets of test problems: DTLZ and WFG. Our *iSMS-EMOA* came third only in one problem, DTLZ1 with three objective functions, and it ranked second in the remaining problems. Finally, *SMS-EMOA* ranked third in twenty DTLZ test problems and in twenty-one WFG test problems and it came second in one DTLZ test problem. We can also observe that our *iSMS-EMOA* significantly decreased the running time required by *SMS-EMOA*. For example, in the DTLZ test problems with five objective functions, *SMS-EMOA* consumes the maximum allowable time (4 hours) and is unable to converge, whereas our *iSMS-EMOA* requires a maximum of sixty minutes to converge in any of the DTLZ test problems. With respect to the WFG test problems, we can also see that our *iSMS-EMOA* requires less time than *SMS-EMOA*. However, it is important to note that in this case, *SMS-EMOA* doesn't consume the maximum allowable time in some problems with 5 objectives. This indicates that the algorithm uses Pareto ranking most of the time, since otherwise its computational cost would had been much higher. This is remarkable, because our *iSMS-EMOA* always uses hypervolume to discard one individual, but it still requires less running time than *SMS-EMOA* in the WFG test problems.

Finally, Figures 5.12, 5.13 and 5.14 show the Pareto fronts obtained by the three algorithms in their median with respect to the hypervolume indicator in some of the test problems adopted. Here, we can see again that *hypeSMS-EMOA* loses quality in most cases, unlike our *iSMS-EMOA* which achieves a good distribution. Therefore, from these experimental results we can say that our selection mechanism based on the hypervolume and on its locality property is an excellent option to solve MOPs. In fact, our proposal is better than the original selection mechanism used by *SMS-EMOA* because our selection mechanism is more practical to solve MOPs with many objective functions.

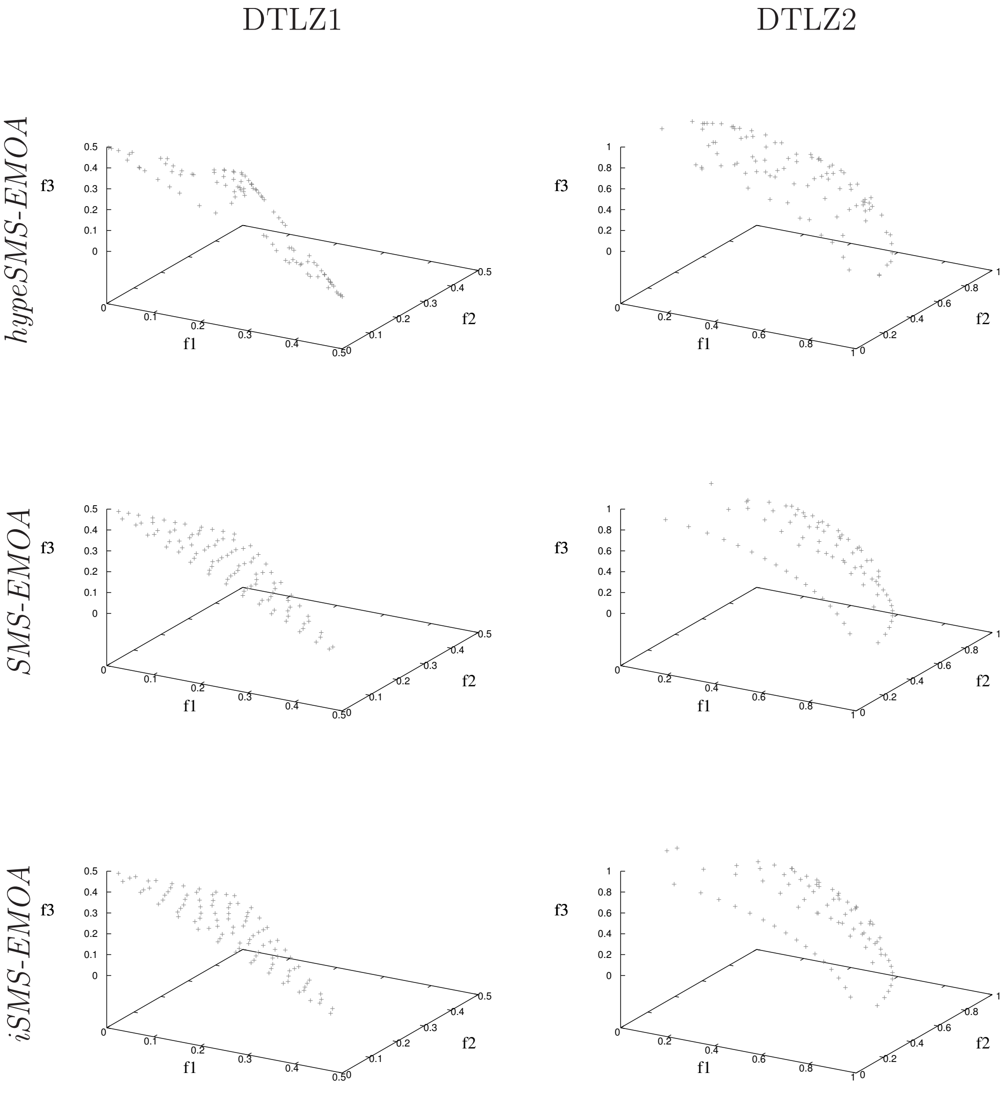


Figure 5.12: Pareto fronts obtained by the three MOEAs (*hypeSMS-EMOA*, *SMS-EMOA* and *iSMS-EMOA*) in the median (with respect to the hypervolume indicator) of their thirty independent runs for the test problems DTLZ1 and DTLZ2.

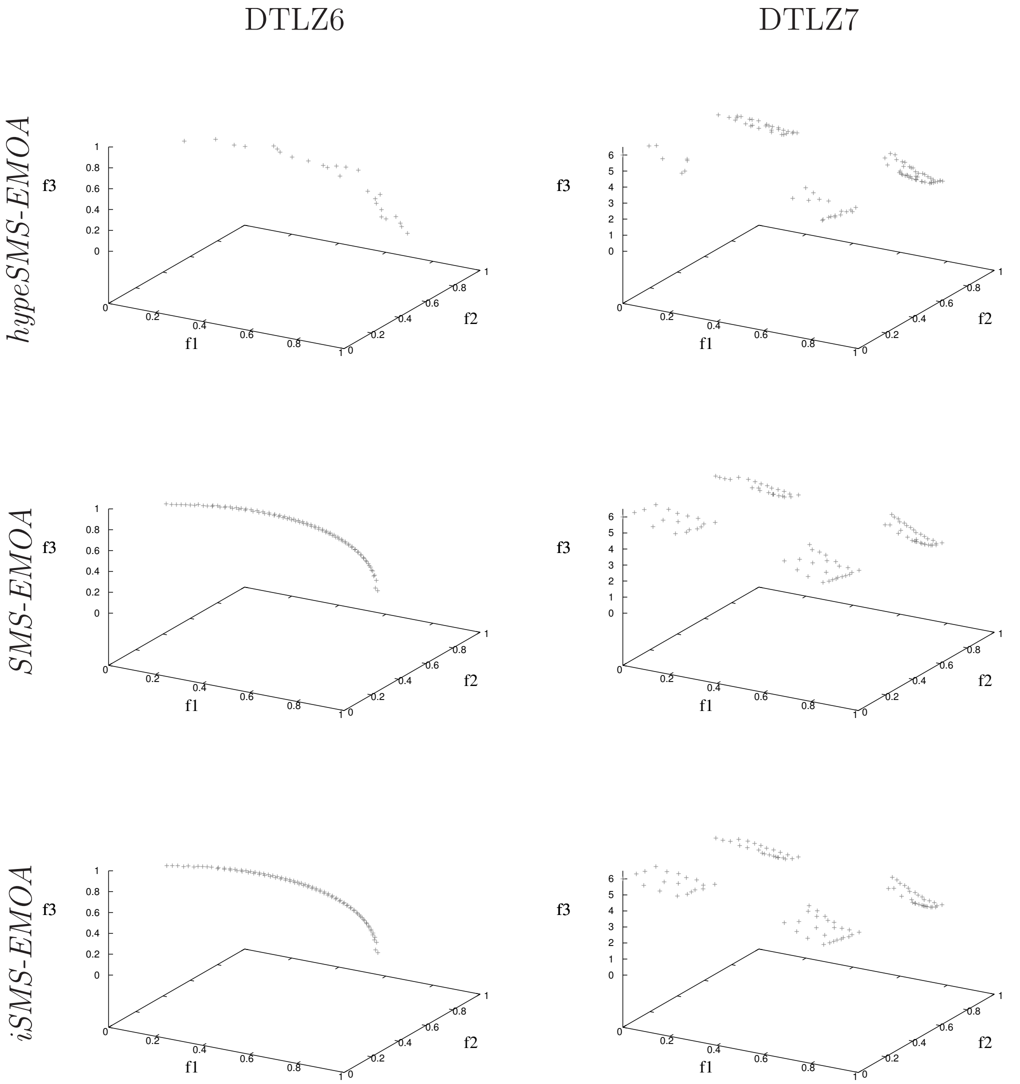


Figure 5.13: Pareto fronts obtained by the three MOEAs (*hypeSMS-EMOA*, *SMS-EMOA* and *iSMS-EMOA*) in the median (with respect to the hypervolume indicator) of their thirty independent runs for the test problems DTLZ6 and DTLZ7.

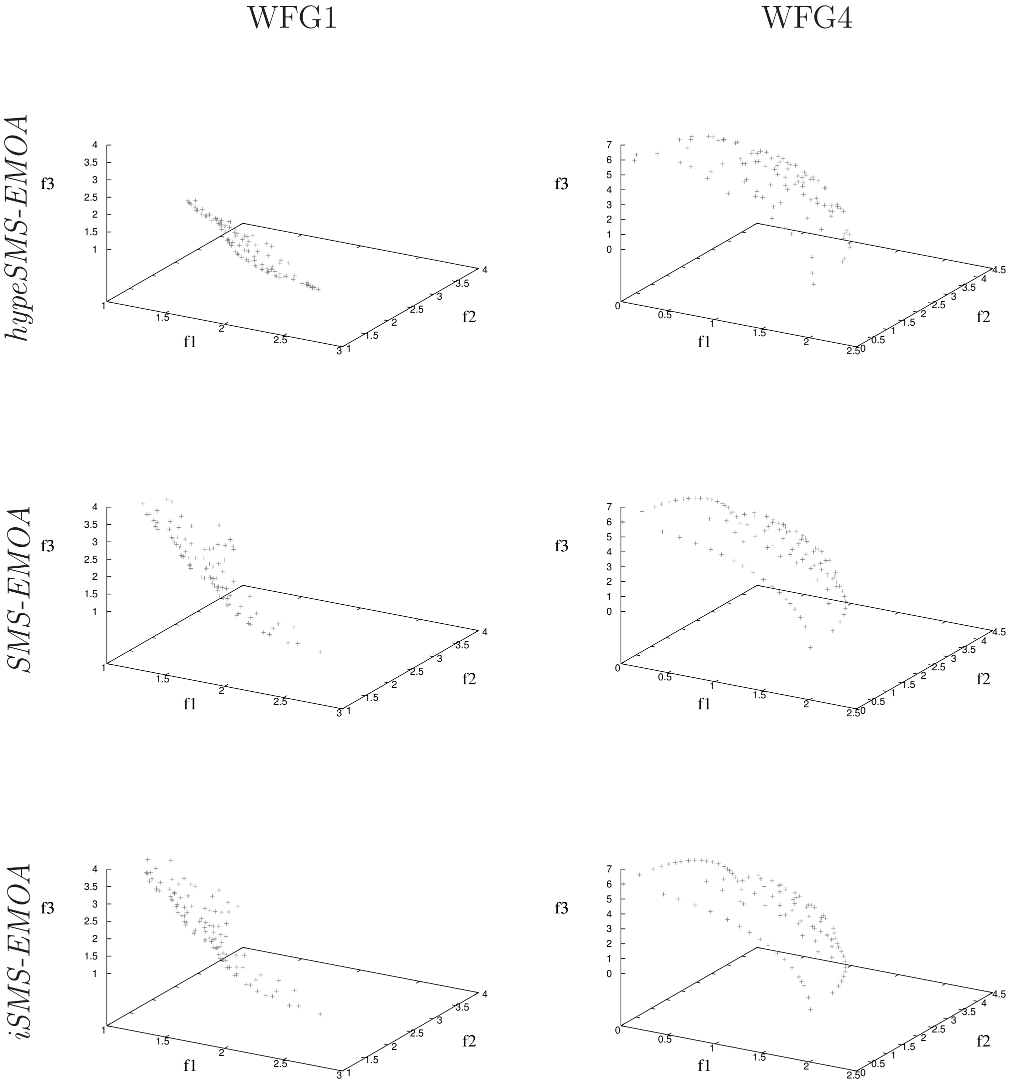
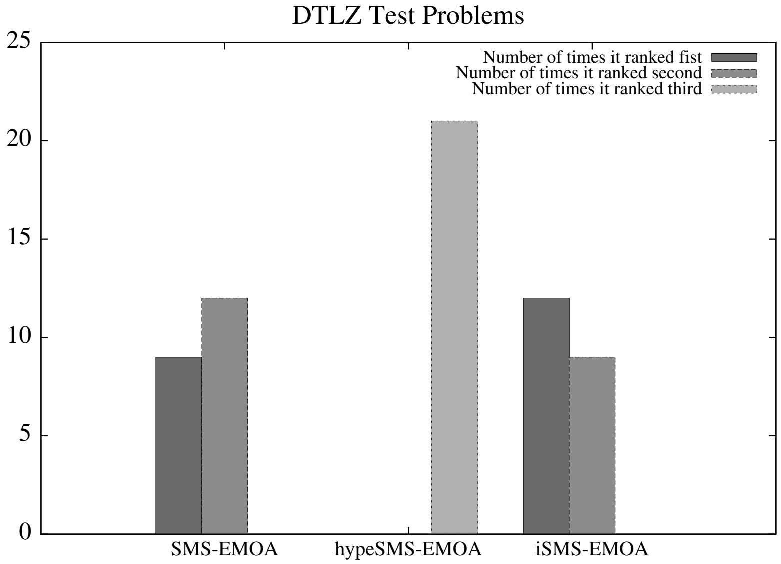
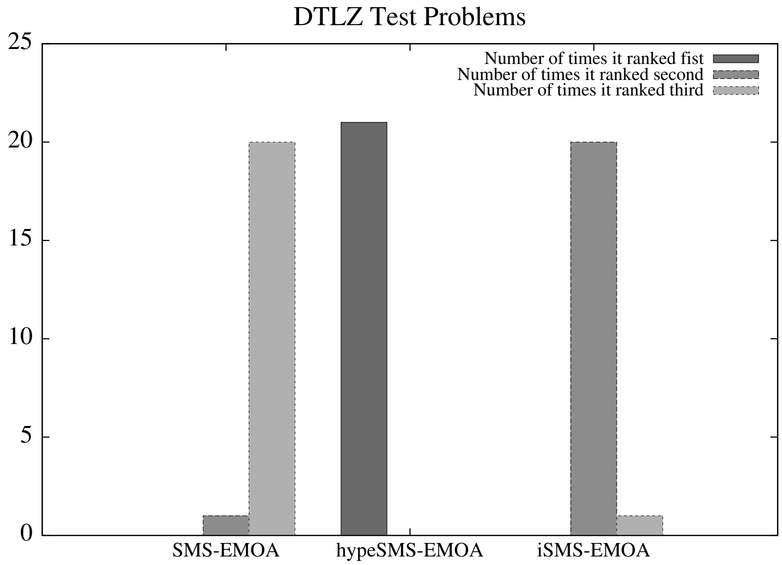


Figure 5.14: Pareto fronts obtained by the three MOEAs (*hypeSMS-EMOA*, *SMS-EMOA* and *iSMS-EMOA*) in the median (with respect to the hypervolume indicator) of their thirty independent runs for the test problems WFG1 and WFG4.



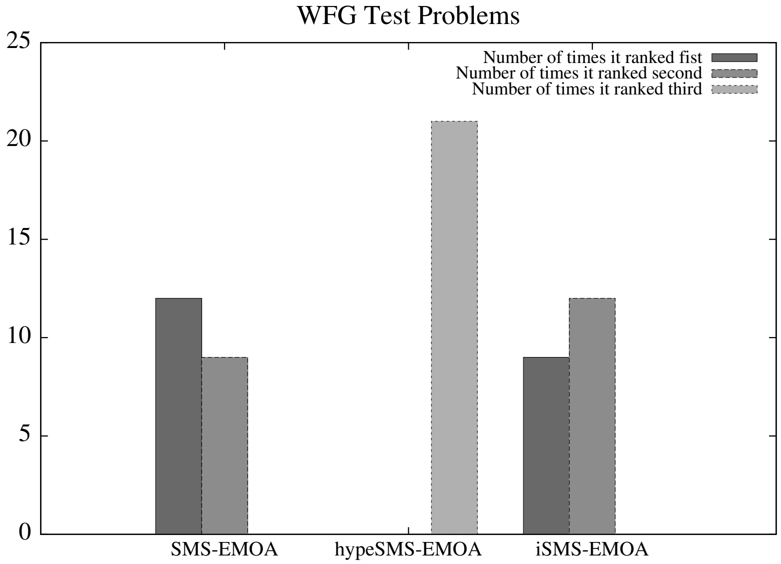
$\vec{f}$	SMS-EMOA	hypeSMS-EMOA	iSMS-EMOA
	I_H	I_H	I_H
DTLZ1 (3)	**1.1230 (0.000)**	1.0785 (0.016)	1.1230 (0.001)
DTLZ2 (3)	**0.7581 (0.000)**	0.7415 (0.002)	0.7580 (0.000)
DTLZ3 (3)	1.3308 (0.000)	1.3307 (0.000)	**1.3308 (0.000)**
DTLZ4 (3)	**0.7585 (0.000)**	0.7442 (0.002)	0.7584 (0.000)
DTLZ5 (3)	**0.2667 (0.000)**	0.2651 (0.000)	0.2667 (0.000)
DTLZ6 (3)	**1.0951 (0.004)**	1.0726 (0.026)	1.0951 (0.005)
DTLZ7 (3)	0.5282 (0.057)	0.5138 (0.050)	**0.5583 (0.044)**
DTLZ1 (4)	**1.3737 (0.000)**	1.1833 (0.071)	1.3737 (0.000)
DTLZ2 (4)	**1.0500 (0.000)**	1.0121 (0.004)	1.0496 (0.000)
DTLZ3 (4)	1.4639 (0.000)	1.4639 (0.000)	**1.4639 (0.000)**
DTLZ4 (4)	**1.0482 (0.000)**	1.0192 (0.003)	1.0479 (0.000)
DTLZ5 (4)	0.5460 (0.000)	0.5131 (0.006)	**0.5462 (0.000)**
DTLZ6 (4)	**1.2410 (0.004)**	1.0706 (0.023)	1.2408 (0.004)
DTLZ7 (4)	0.5646 (0.030)	0.4875 (0.028)	**0.5813 (0.022)**
DTLZ1 (5)	1.5665 (0.001)	1.1579 (0.307)	**1.5678 (0.000)**
DTLZ2 (5)	1.4322 (0.000)	1.3996 (0.007)	**1.4333 (0.000)**
DTLZ3 (5)	1.6102 (0.000)	1.6102 (0.000)	**1.6103 (0.000)**
DTLZ4 (5)	1.3040 (0.000)	1.2619 (0.006)	**1.3050 (0.000)**
DTLZ5 (5)	0.9318 (0.001)	0.8472 (0.015)	**0.9360 (0.001)**
DTLZ6 (5)	1.4649 (0.003)	1.3033 (0.027)	**1.4876 (0.002)**
DTLZ7 (5)	0.5597 (0.038)	0.4566 (0.050)	**0.5822 (0.006)**

Table 5.1: Results obtained in the DTLZ test problems with up to five objective functions. We compare *SMS-EMOA*, *hypeSMS-EMOA* and *iSMS-EMOA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



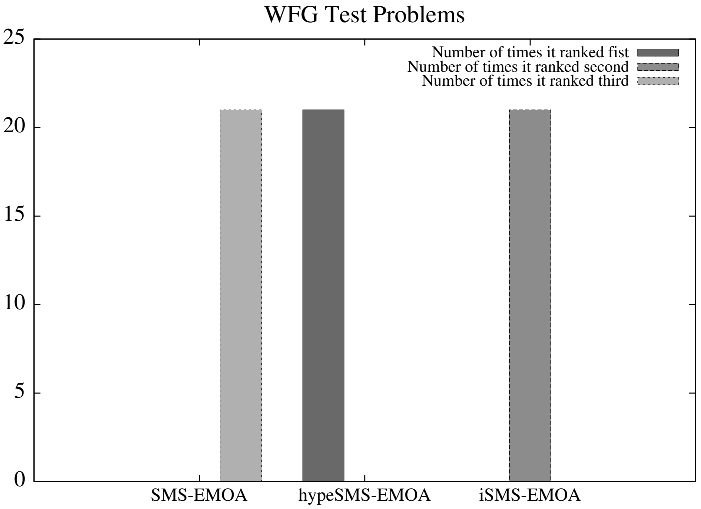
$\vec{f}$	SMS-EMOA time	hypeSMS-EMOA time	iSMS-EMOA time
DTLZ1 (3)	365.5593 (27.897)	**9.9760 (0.113)**	368.4773 (40.879)
DTLZ2 (3)	486.6670 (31.554)	**15.5313 (0.259)**	382.9243 (36.386)
DTLZ3 (3)	499.1160 (39.988)	**11.7833 (0.817)**	440.1397 (38.473)
DTLZ4 (3)	523.4633 (47.806)	**15.6460 (0.255)**	451.0887 (36.375)
DTLZ5 (3)	381.1447 (17.116)	**13.0363 (0.159)**	379.8520 (27.687)
DTLZ6 (3)	403.4967 (15.122)	**13.5823 (1.575)**	370.1153 (24.928)
DTLZ7 (3)	485.5560 (29.970)	**19.3843 (0.606)**	382.2553 (26.468)
DTLZ1 (4)	1578.3723 (54.450)	**12.8663 (0.193)**	429.2957 (13.493)
DTLZ2 (4)	2754.8697 (100.357)	**19.5447 (0.312)**	441.9803 (17.658)
DTLZ3 (4)	2390.6237 (1153.111)	**14.3620 (0.842)**	470.9413 (86.973)
DTLZ4 (4)	2884.5260 (88.896)	**19.5833 (0.375)**	468.5887 (68.708)
DTLZ5 (4)	1905.9827 (107.565)	**22.9647 (0.606)**	418.9777 (14.317)
DTLZ6 (4)	2356.9200 (50.540)	**21.8453 (0.616)**	436.5570 (14.477)
DTLZ7 (4)	1844.2347 (96.855)	**27.0250 (1.119)**	399.1967 (13.255)
DTLZ1 (5)	14441.8333 (16.546)	**15.4097 (0.242)**	1797.6717 (66.814)
DTLZ2 (5)	14448.5523 (27.680)	**22.6683 (0.585)**	1891.2693 (81.428)
DTLZ3 (5)	12229.6320 (3327.420)	**16.1650 (0.717)**	2392.4797 (1266.078)
DTLZ4 (5)	14449.7207 (31.780)	**22.5047 (0.425)**	2052.9640 (87.407)
DTLZ5 (5)	14439.4680 (21.651)	**30.3343 (1.089)**	1731.5287 (109.424)
DTLZ6 (5)	14442.1650 (24.369)	**30.7600 (0.978)**	1749.2400 (41.552)
DTLZ7 (5)	14182.2843 (696.870)	**34.9543 (1.854)**	768.9817 (56.556)

Table 5.2: Results obtained in the DTLZ test problems with up to five objective functions. We compare *SMS-EMOA*, *hypeSMS-EMOA* and *iSMS-EMOA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	SMS-EMOA I_H	hypeSMS-EMOA I_H	iSMS-EMOA I_H
WFG1 (3)	1.2097 (0.029)	0.8864 (0.033)	**1.2162 (0.011)**
WFG2 (3)	**0.8323 (0.051)**	0.6608 (0.046)	0.8293 (0.072)
WFG3 (3)	0.6367 (0.002)	0.6003 (0.006)	**0.6378 (0.001)**
WFG4 (3)	**0.7543 (0.002)**	0.6999 (0.006)	0.7540 (0.002)
WFG5 (3)	0.5573 (0.002)	0.5371 (0.003)	**0.5575 (0.002)**
WFG6 (3)	**0.5669 (0.002)**	0.5538 (0.004)	0.5668 (0.001)
WFG7 (3)	**0.7521 (0.004)**	0.5017 (0.029)	0.7489 (0.004)
WFG1 (4)	**1.4272 (0.005)**	0.9985 (0.036)	1.4219 (0.009)
WFG2 (4)	**0.8659 (0.130)**	0.4703 (0.206)	0.8642 (0.120)
WFG3 (4)	**0.5993 (0.006)**	0.5189 (0.014)	0.5957 (0.006)
WFG4 (4)	**1.0346 (0.003)**	0.9048 (0.013)	1.0330 (0.003)
WFG5 (4)	**0.5996 (0.002)**	0.5542 (0.006)	0.5995 (0.002)
WFG6 (4)	0.6128 (0.006)	0.5712 (0.008)	**0.6142 (0.007)**
WFG7 (4)	**0.9339 (0.007)**	0.3559 (0.022)	0.9288 (0.004)
WFG1 (5)	1.3657 (0.018)	1.0722 (0.041)	**1.5656 (0.007)**
WFG2 (5)	0.8914 (0.170)	0.5326 (0.252)	**0.9233 (0.145)**
WFG3 (5)	**0.5926 (0.025)**	0.4267 (0.026)	0.5851 (0.028)
WFG4 (5)	1.2219 (0.009)	1.0734 (0.021)	**1.2777 (0.005)**
WFG5 (5)	0.6562 (0.002)	0.5622 (0.011)	**0.6566 (0.002)**
WFG6 (5)	**0.6473 (0.024)**	0.5101 (0.037)	0.6469 (0.021)
WFG7 (5)	0.7680 (0.036)	0.2579 (0.018)	**1.0266 (0.015)**

Table 5.3: Results obtained in the WFG test problems with up to five objective functions. We compare *SMS-EMOA*, *hypeSMS-EMOA* and *iSMS-EMOA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	SMS-EMOA time	hypeSMS-EMOA time	iSMS-EMOA time
WFG1 (3)	568.2757 (54.687)	**25.5767 (0.207)**	419.1100 (17.560)
WFG2 (3)	431.6637 (25.635)	**18.6463 (0.864)**	420.8630 (17.313)
WFG3 (3)	495.4100 (30.949)	**25.2423 (0.296)**	420.0893 (16.889)
WFG4 (3)	539.6530 (46.373)	**20.7107 (0.309)**	415.2110 (15.874)
WFG5 (3)	531.9113 (44.412)	**26.8690 (0.138)**	414.9457 (18.544)
WFG6 (3)	496.9443 (32.744)	**28.4663 (0.393)**	419.7743 (17.266)
WFG7 (3)	581.8233 (58.042)	**26.2263 (0.633)**	417.2413 (20.361)
WFG1 (4)	3734.0803 (180.641)	**32.0923 (0.441)**	468.2133 (44.586)
WFG2 (4)	862.0040 (48.570)	**24.5730 (1.171)**	397.2097 (22.681)
WFG3 (4)	957.0357 (16.880)	**31.9290 (0.367)**	390.2180 (17.506)
WFG4 (4)	3053.7340 (76.497)	**26.0010 (0.586)**	451.8720 (41.150)
WFG5 (4)	1271.6740 (11.726)	**36.5713 (0.283)**	391.1927 (18.414)
WFG6 (4)	1105.0230 (22.982)	**37.6103 (0.541)**	394.1047 (23.013)
WFG7 (4)	3165.9447 (79.757)	**34.3960 (0.512)**	444.5883 (39.056)
WFG1 (5)	14457.7380 (34.416)	**37.8360 (0.283)**	2322.8240 (157.156)
WFG2 (5)	2567.6657 (229.868)	**31.0053 (1.328)**	338.8497 (31.956)
WFG3 (5)	1609.6663 (40.743)	**38.8517 (0.413)**	264.8837 (40.684)
WFG4 (5)	14454.3767 (34.993)	**31.7297 (0.774)**	1894.0353 (106.299)
WFG5 (5)	3021.6070 (159.832)	**44.9227 (0.158)**	305.5377 (32.373)
WFG6 (5)	2916.7467 (216.336)	**46.2050 (0.443)**	314.7670 (30.785)
WFG7 (5)	14459.3203 (26.034)	**41.7163 (0.400)**	1597.0273 (97.936)

Table 5.4: Results obtained in the WFG test problems with up to five objective functions. We compare *SMS-EMOA*, *hypeSMS-EMOA* and *iSMS-EMOA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.

$\vec{f}$	SMS-EMOA & iSMS-EMOA $P(H)$	hypeSMS-EMOA & iSMS-EMOA $P(H)$
DTLZ1 (3)	0.491771 (0)	0.000000 (1)
DTLZ2 (3)	0.000032 (1)	0.000000 (1)
DTLZ3 (3)	0.211295 (0)	0.000000 (1)
DTLZ4 (3)	0.000000 (1)	0.000000 (1)
DTLZ5 (3)	0.000016 (1)	0.000000 (1)
DTLZ6 (3)	0.894139 (0)	0.000011 (1)
DTLZ7 (3)	0.000066 (1)	0.000000 (1)
DTLZ1 (4)	0.630872 (0)	0.000000 (1)
DTLZ2 (4)	0.000000 (1)	0.000000 (1)
DTLZ3 (4)	0.001233 (1)	0.000014 (1)
DTLZ4 (4)	0.000000 (1)	0.000000 (1)
DTLZ5 (4)	0.000101 (1)	0.000000 (1)
DTLZ6 (4)	0.888303 (0)	0.000000 (1)
DTLZ7 (4)	0.023243 (1)	0.000000 (1)
DTLZ1 (5)	0.000000 (1)	0.000000 (1)
DTLZ2 (5)	0.000000 (1)	0.000000 (1)
DTLZ3 (5)	0.000925 (1)	0.000000 (1)
DTLZ4 (5)	0.000000 (1)	0.000000 (1)
DTLZ5 (5)	0.000000 (1)	0.000000 (1)
DTLZ6 (5)	0.000000 (1)	0.000000 (1)
DTLZ7 (5)	0.000149 (1)	0.000000 (1)

Table 5.5: Statistical analysis using Wilcoxon's rank sum for the DTLZ test problems. For this, we used I_H, see Table 5.1. A light gray color in the columns means that *iSMS-EMOA* was outperformed by another MOEA (*SMS-EMOA* or *hypeSMS-EMOA*). A gray color means that *iSMS-EMOA* was better than the other MOEA (*SMS-EMOA* or *hypeSMS-EMOA*). P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	SMS-EMOA & iSMS-EMOA $P(H)$	hypeSMS-EMOA & iSMS-EMOA $P(H)$
WFG1 (3)	0.982307 (0)	0.000000 (1)
WFG2 (3)	0.630876 (0)	0.000000 (1)
WFG3 (3)	0.105470 (0)	0.000000 (1)
WFG4 (3)	0.450840 (0)	0.000000 (1)
WFG5 (3)	0.529782 (0)	0.000000 (1)
WFG6 (3)	0.569220 (0)	0.000000 (1)
WFG7 (3)	0.014412 (1)	0.000000 (1)
WFG1 (4)	0.011228 (1)	0.000000 (1)
WFG2 (4)	0.620404 (0)	0.000000 (1)
WFG3 (4)	0.016285 (1)	0.000000 (1)
WFG4 (4)	0.055546 (0)	0.000000 (1)
WFG5 (4)	0.599689 (0)	0.000000 (1)
WFG6 (4)	0.340288 (0)	0.000000 (1)
WFG7 (4)	0.002624 (1)	0.000000 (1)
WFG1 (5)	0.000000 (1)	0.000000 (1)
WFG2 (5)	0.501144 (0)	0.000000 (1)
WFG3 (5)	0.115362 (0)	0.000000 (1)
WFG4 (5)	0.000000 (1)	0.000000 (1)
WFG5 (5)	0.264326 (0)	0.000000 (1)
WFG6 (5)	0.982307 (0)	0.000000 (1)
WFG7 (5)	0.000000 (1)	0.000000 (1)

Table 5.6: Statistical analysis using Wilcoxon's rank sum for the WFG test problems. For this, we used I_H, see Table 5.3. A light gray color in the columns means that *iSMS-EMOA* was outperformed by another MOEA (*SMS-EMOA* or *hypeSMS-EMOA*). A gray color means that *iSMS-EMOA* was better than the other MOEA (*SMS-EMOA* or *hypeSMS-EMOA*). P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

5.3 Approximating the Contribution to the Hypervolume

As we mentioned above, there are several proposals to approximate the hypervolume or its contribution. However, these techniques were incorporated into MOEAs that use the traditional selection scheme in which all individuals compete to survive. In this section, we propose to use the approximation technique proposed by Bringmann and Friedrich in [12] into the selection mechanism proposed in the previous section. There are two main motivations to adopt this approach: First, Bringmann and Friedrich proposed a technique to approximate the contribution to the hypervolume of an individual without having to calculate the hypervolume and our new selection mechanism is based on the hypervolume contributions. And second, we have the following hypothesis: since the new selection mechanism needs to calculate the contribution of only three individuals, we can reduce the error of the approximation, by increasing the number of samples and this will not increase the running time in an excessive manner. Also, the probability of deleting the individual with the lowest contribution is greater than if we use the traditional competition scheme because if we randomly choose one individual of a set of three individuals, the probability of choosing the worst individual is $1/3$. And, if we choose one individual of a set of P individuals, the probability of choosing the worst individual is $1/P$. In addition, we only deal with three errors and not with P errors where P is the size of the population. Furthermore, we expect that the contributions of the new solution and its nearest neighbor are different because our idea is to decide if we move the current solution in the population (nearest neighbor of the new solution) to the position of the new solution. Therefore, we expect that the joint contribution of the other solutions is fixed and the contribution of the new solution and its nearest neighbor are different (locality property).

We designed an experimental test to validate these last claims. We show experimentally our hypothesis in the following sections. In Algorithm 7, we can see the procedure to approximate the contribution to the hypervolume of one individual in the population. And we can see our alternative selection mechanism in Algorithm 8.

Algorithm 7: Approximating the contribution to the hypervolume of individual $\vec{x}$.

```
Input  : Current population ($\mathcal{P}_t$), individual ($\vec{x}$), reference point
         ($\vec{y}_{ref}$), and the number of samples ($n_{samples}$).
Output: Approximation of the contribution to the hypervolume of
         individual $\vec{x}$, $\tilde{C}_H(\vec{x}, \mathcal{P}_t)$.

/*Defining the bounding box                                   */
$\vec{y}_{box} \leftarrow \vec{y}_{ref}$;
foreach $\vec{x}_i \in \mathcal{P}_t$ such that $\vec{x}_i \neq \vec{x}$ do
    if $\vec{x}$ is dominated by $\vec{x}_i$ in all objective functions except in $f_k$ then
        $\vec{y}_{box}[k] \leftarrow \vec{x}_i[k]$;
    end
end

/*Calculating the volume of the box $BB_x$                     */
$volumeBB_x \leftarrow 1$;
foreach Objective function $k$ do
    $volumeBB_x \leftarrow (volumeBB_x)(\vec{y}_{box}[k] - \vec{x}[k])$;
end

/*Doing sampling                                               */
$SuccessSamples \leftarrow 0$;
for $j \leftarrow 1$ to $n_{samples}$ do
    Generate a random point $\vec{x}_r$, such that $\vec{x}_r \in BB_x$;
    if Not exists another point $\vec{x}_i \in \mathcal{P}_t$ such that $\vec{x}_i$ dominates $\vec{x}_r$ then
        $SuccessSamples \leftarrow SuccessSamples + 1$;
    end
end

return $\frac{SuccessSamples}{n_{samples}}(volumeBB_x)$;
```

Algorithm 8: Alternative selection mechanism based on the hypervolume.

```
Input  : Current population ($\mathcal{P}_t$) and the new solution ($\vec{x}_{new}$).
Output: The new population ($\mathcal{P}_{t+1}$).

/*Calculate the distance of each solution in $\mathcal{P}_t$ to $\vec{x}_{new}$ */
foreach $\vec{x}_i \in \mathcal{P}_t$ do
    $dist_i \leftarrow \|\vec{x}_i - \vec{x}_{new}\|$;
end

/*Choose the nearest solution to $\vec{x}_{new}$ */
$\vec{x}_{near} \mid dist_{near} = \min dist_i$;

/*Choose one random solution */
Choose randomly $\vec{x}_{rand}$ such that $\vec{x}_{rand} \in \mathcal{P}_t$ and $\vec{x}_{rand} \neq \vec{x}_{new}$ and
$\vec{x}_{rand} \neq \vec{x}_{near}$;

/*Approximate the contributions to the hypervolume, using
  Algorithm 7 */
$\tilde{C}_{new} \leftarrow \tilde{C}_H(\vec{x}_{new}, \mathcal{P}_t)$;
$\tilde{C}_{near} \leftarrow \tilde{C}_H(\vec{x}_{near}, \mathcal{P}_t)$;
$\tilde{C}_{rand} \leftarrow \tilde{C}_H(\vec{x}_{rand}, \mathcal{P}_t)$;

/*Remove the solution with the worst contribution */
$\vec{x}_{worst} \mid \tilde{C}_{worst} = \min\{\tilde{C}_{new}, \tilde{C}_{near}, \tilde{C}_{rand}\}$;
$\mathcal{P}_{t+1} \leftarrow \mathcal{P} \setminus \vec{x}_{worst}$;
```

5.3.1 Experimental results

To validate our alternative selection mechanism based on the hypervolume indicator, we incorporated it into the original *SMS-EMOA* [8] and we called it *"approximate version of improved SMS-EMOA (aviSMS-EMOA)"*. For our experiments, we used problems with up to six objective functions, seven of which were taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [30] and seven more were taken from the Walking Fish Group (WFG) toolkit [48]. We used $k = 5$ for DTLZ1, DTLZ3 and DTLZ6 and $k = 10$ for the remaining test problems. We used $k_{factor} = 2$ and $l_{factor} = 10$ for the WFG test problems. For each test problem, we performed 30 independent runs. For all algorithms, we adopted the parameters suggested by the authors of NSGA-II: $p_c = 0.9$ (crossover probability), $p_m = 1/n$ (mutation probability), where n is the number of decision variables. For the crossover and mutation operators, we adopted $\eta_c = 15$ and $\eta_m = 20$, respectively. We performed a maximum of 50,000 fitness function evaluations (we used a population size of 100 individuals and we iterated for 500 generations). However, *we adopted four hours as our maximum running time* because we know that the computation of the exact hypervolume contribution has a high computational cost. All MOEAs considered in our experiments were compiled using the GNU C compiler and they were executed on a computer with a 2.66GHz processor and 4GB in RAM.

5.3.2 Performance indicators

We adopted I_H again to validate our results because it rewards both convergence towards the Pareto front as well as the maximum spread of the solutions obtained. Also, most of the algorithms used in this section have as their aim to maximize the hypervolume and, therefore, it makes sense to use this indicator to assess their performance. To calculate the hypervolume indicator, we normalized the approximations of the Pareto optimal set, generated by the MOEAs, and we used $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$ as our reference point. The normalization was performed considering all approximations generated by the different compared MOEAs (i.e., we place, in one set, all nondominated solutions found and from this set we calculate the maximum and minimum for each objective function).

Only in some experiments, we consider other two quality indicators. The first one is called "two set coverage (I_{SC})" and has the aim of assessing only the convergence of the MOEAs. As we mentioned in Chapter 3, it is a Pareto compliant binary indicator. Let $\mathcal{A}, \mathcal{B}$ be two approximations of the Pareto optimal set, I_{SC} is defined as follows:

$$I_{SC}(\mathcal{A}, \mathcal{B}) = \frac{|\vec{b} \in \mathcal{B} \text{ such that } \exists \vec{a} \in \mathcal{A} \text{ with } \vec{a} \prec \vec{b}|}{|\mathcal{B}|}$$

If all points in $\mathcal{A}$ dominate or are equal to all points in $\mathcal{B}$, then by definition $I_{SC} = 1$. $I_{SC} = 0$ implies that no element in $\mathcal{B}$ is dominanted by any element of $\mathcal{A}$. In general, both $I_{SC}(\mathcal{A}, \mathcal{B})$ and $I_{SC}(\mathcal{B}, \mathcal{A})$ have to be considered.

The second one is called "inverted generational distance" indicator (I_{IGD}). As we mentioned in Chapter 3, I_{IGD} reports how far, on average, $\mathcal{PF}$ is from $\mathcal{A}$, where $\mathcal{PF}$ is the true Pareto front and $\mathcal{A}$ is an approximation of the true Pareto front. I_{IGD} is Pareto non-compliant and it is defined as:

$$I_{IGD}(\mathcal{A}) = \frac{1}{|\mathcal{PF}|} \left(\sum_{i=1}^{|\mathcal{PF}|} d_i^p \right)^{\frac{1}{p}}$$

where $|\mathcal{PF}|$ is the number of vectors in $\mathcal{PF}$, $p = 2$ and d_i is the Euclidean phenotypic distance between each member, i, of $\mathcal{PF}$ and the closest member in $\mathcal{A}$ to that member, i. I_{IGD} measures both convergence and distribution. Since we cannot use the true Pareto front because in most cases we cannot obtain it, we need to use a reference set that provides a reasonably good approximation of the true Pareto front. The result of this indicator strongly depends on this reference set.

5.3.3 Approximate version of original SMS-EMOA vs approximate version of improved SMS-EMOA

As we saw in this section, we propose to approximate the contributions to the hypervolume used by the selection mechanism based on the hypervolume indicator and on its locality property. However, this gives rise to the following question: *why don't we approximate the contributions in the original version of the SMS-EMOA?* Our hypothesis is that it is better to use the improved *SMS-EMOA* for two reasons: First, we need to do sampling for each solution for which we need to know its contribution. Therefore, if we use a large number of samples, the running time drastically increases when employing the traditional selection mechanism adopted by the original *SMS-EMOA*. This is because in this case, it is required to know the contributions of all the individuals at each iteration, unlike the selection mechanism used by the improved *SMS-EMOA* that only requires to know the contribution of three individuals per iteration. And second, if we use the improved *SMS-EMOA*, we can decrease the probability of not choosing the worst individual because we only deal with three errors instead of dealing with P errors (P is the population size). Also, it is important to consider that the selection mechanism used in the improved *SMS-EMOA* exploits the locality property of the hypervolume. Thus, we estimate that the contributions of the new individual and its nearest neighbor will be different and that the joint contribution of the remaining individuals is fixed.

In order to validate our hypothesis, we compare our *aviSMS-EMOA* with respect to a version of the original *SMS-EMOA* that approximates the contributions to the hypervolume using Algorithm 7; this version is called "approximate version of the original *SMS-EMOA* (*avoSMS-EMOA*)." We ran tests with up to six objective functions because the *avoSMS-EMOA* algorithm has a manageable running time up to this dimensionality. We used $k(10^3)$ as our number of samples for both algorithms, where k is the number of objective functions (e.g., if we have a MOP with three objective functions, we use $3(10^3) = 3000$ samples). We decided to use this number of samples with the aim that both MOEAs can finish the search or they can execute the largest possible number of generations in the allowable time (we must remember that we adopted four hours as our maximum running time). However, later, we study the behavior of our *aviSMS-EMOA* with respect to the number of samples.

Tables 5.7 and 5.8 show the results of the DTLZ and WFG test problems with respect to the hypervolume indicator. Also, they show the statistical analysis applied to the experiments using Wilcoxon's rank sum. In these tables, we can see that our *aviSMS-EMOA* obtains better results in most cases, *avoSMS-EMOA* obtained better results only in nine cases. However, if we check the statistical analysis, we can see that *avoSMS-EMOA* outperforms our *aviSMS-EMOA* only in four cases because only in these cases the hypothesis that "medians are equal" can be rejected. Moreover, our *aviSMS-EMOA* obtains better results than *avoSMS-EMOA* in forty-seven cases and the hypothesis that "medians are equal" can be rejected in forty-three cases. In summary, our *aviSMS-EMOA* outperforms *avoSMS-EMOA* in forty-three problems, it is outperformed in four problems, and both algorithms obtain similar results in nine problems. Now, let's check Table 5.9, which shows the running time required by the two algorithms to obtain the approximate Pareto front of each test problem. In this table, we can see that our proposed *aviSMS-EMOA* is significantly better in all cases (we can say that, on average, it is ten times faster than *avoSMS-EMOA*). It is important to clarify that *aviSMS-EMOA* is not thirty-three times faster than *avoSMS-EMOA* because *avoSMS-EMOA* does not always calculate 100 contributions to I_H (only when after applying Pareto ranking, a single front is obtained). Also, *aviSMS-EMOA* only uses the selection mechanism based on I_H and its locality property when, after applying Pareto ranking, we only obtain a single front. Otherwise, both algorithms use the number of solutions that dominate certain solution as suggested by Beume et al. in [8]. This validates, experimentally, the first part of our hypothesis with respect to the running time.

For validating the second part, we executed both algorithms, *avoSMS-EMOA* and *aviSMS-EMOA*, but we also calculate the exact contribution to the hypervolume and we verify if the algorithm chooses the correct individual to be deleted (worst individual) per iteration. With the aim of calculating the success percentage of each selection mechanism, we consider a success when the algorithm deletes the worst individual. Also, we calculate the percentage in which the new solution and its nearest neighbor had different values in their contribution to the hypervolume because we expect that most of the time these two contributions will be different. For this experiment, we only used problems with three and four objective functions, in order to keep the running times within manageable values. In Table 5.10, we can see that the alternative selection mechanism used in *aviSMS-EMOA* achieves a high success rate (above 98% in all cases). This does not happen with *avoSMS-EMOA*, which cannot even reach a success rate of 1%. Also, we can see that most of the time the new solution and its nearest neighbor had a different contribution (above 95% in most cases) as we expected due to the locality property of the hypervolume. Thus, we can conclude that our proposed *aviSMS-EMOA* is better than *avoSMS-EMOA*. This is an important result because we show experimentally that we can significantly improve the current MOEAs based on approximations of I_H if we use the selection mechanism based on the hypervolume indicator and on its locality property.

5.3.4 HyPE version of SMS-EMOA vs approximate version of improved SMS-EMOA

In this section, we adopt a version of the original *SMS-EMOA* that uses the fitness assignment scheme proposed in [3] instead of calculating the exact contributions and which we called "HyPE version of *SMS-EMOA* (*hypeSMS-EMOA*)". It is important to mention that our aim was to validate the selection mechanism. Therefore, we decided to use the same MOEA in all cases and we only changed the selection mechanism. For this reason, we did not use the original HyPE. For *hypeSMS-EMOA*, we used the source code of HyPE, which is available in the public domain. In both techniques, we used $k(10^3)$ as our number of samples, where k is the number of objective functions.

Tables 5.11 and 5.12 show the results for the DTLZ and WFG test problems, with up to six objective functions, with respect to the hypervolume indicator. Also, they present the statistical analysis of our experiments using Wilcoxon's rank sum. In these tables, we can see that our *aviSMS-EMOA* obtains better results than *hypeSMS-EMOA* in forty-eight problems and in forty-six of these problems, the hypothesis "medians are equal" can be rejected. Only in eight problems, *hypeSMS-EMOA* obtains better results than our *aviSMS-EMOA* and only in four cases, we can say that it outperforms our *aviSMS-EMOA* because the hypothesis can be rejected. Summarizing, our *aviSMS-EMOA* outperforms *hypeSMS-EMOA* in forty-six problems, it is outperformed in four problems and in six problems both algorithms obtain similar results. In Table 5.13, we can see the running time required by the two algorithms and we can note that *hypeSMS-EMOA* is better than our *aviSMS-EMOA* in all cases. However, as we saw in Tables 5.11 and 5.12 the *hypeSMS-EMOA* algorithm loses quality in its solutions and although our *aviSMS-EMOA* is slower than *hypeSMS-EMOA*, its time requirements are still manageable (in the worst case, it requires approximately twenty-four minutes to solve problems with up to six objective functions).

In Figures 5.15, 5.16 and 5.17, we can see the Pareto fronts obtained by the algorithms *hypeSMS-EMOA*, *aviSMS-EMOA* and the original *SMS-EMOA*, in the median of thirty independent runs (with respect to the hypervolume indicator) for some of the problems used. In these figures, we can see that *hypeSMS-EMOA* loses quality in the distribution of the solutions and, in some problems, it cannot even generate the entire Pareto front (for example, in DTLZ6, WFG1 and WFG7). On the other hand, our *aviSMS-EMOA* obtains a good distribution in all cases, similar to those obtained by the original *SMS-EMOA* but at a much lower computational cost.

5.3.5 Approximate version of the improved SMS-EMOA vs the original SMS-EMOA

In this section, we compare our *aviSMS-EMOA* with respect to the original *SMS-EMOA*. We tested it only with up to five objective functions because the original *SMS-EMOA* already exceeds the allowable time in problems with five objective functions. Tables 5.14, 5.15, 5.16 and 5.17 show that our *aviSMS-EMOA* is competitive with respect to the original *SMS-EMOA*. In Tables 5.15 and 5.16, we present the results with respect to the hypervolume indicator and we also present the results of the statistical analysis that we made to validate our experiments, using Wilcoxon's rank sum. Although, in most problems, the original *SMS-EMOA* obtains better results than our *aviSMS-EMOA*, the aim of this work was to design a new MOEA based on the approximations of I_H which can significantly reduce the computational cost of MOEAs based on the exact calculation of I_H in many-objective optimization problems but without losing much quality. In Table 5.17, we can see that our algorithm requires at most 24 minutes to solve problems with five objective functions (9.5% of the allowable time), while the original *SMS-EMOA* spends all the allowable time (four hours) and it is unable to finish the search. It is important to note that our *aviSMS-EMOA* outperformed the original *SMS-EMOA* in four problems (DTLZ6, WFG1, WFG4 and WFG7 with five objective functions) in spite of the fact that it requires much less running time.

Finally, in Table 5.14, we show the results corresponding to the "two set coverage" indicator I_{SC}. To calculate it, we merged all solutions found by our *aviSMS-EMOA* in a set called $\mathcal{A}$, considering all the 30 independent runs, and we merged all solutions found by the original *SMS-EMOA* in a set called $\mathcal{B}$. From this table, we can say that in only eleven problems *SMS-EMOA* covered some solutions generated by *aviSMS-EMOA* and that *aviSMS-EMOA* could not cover any solution generated by *SMS-EMOA*, i.e., in these eleven problems, *SMS-EMOA* was better than our *aviSMS-EMOA* in terms of convergence. However, in the remaining thirty-one problems, our *aviSMS-EMOA* covered some solutions generated by *SMS-EMOA* and, therefore, we cannot say which algorithm is better. There were no cases in which *SMS-EMOA* was able to cover all solutions generated by our *aviSMS-EMOA*

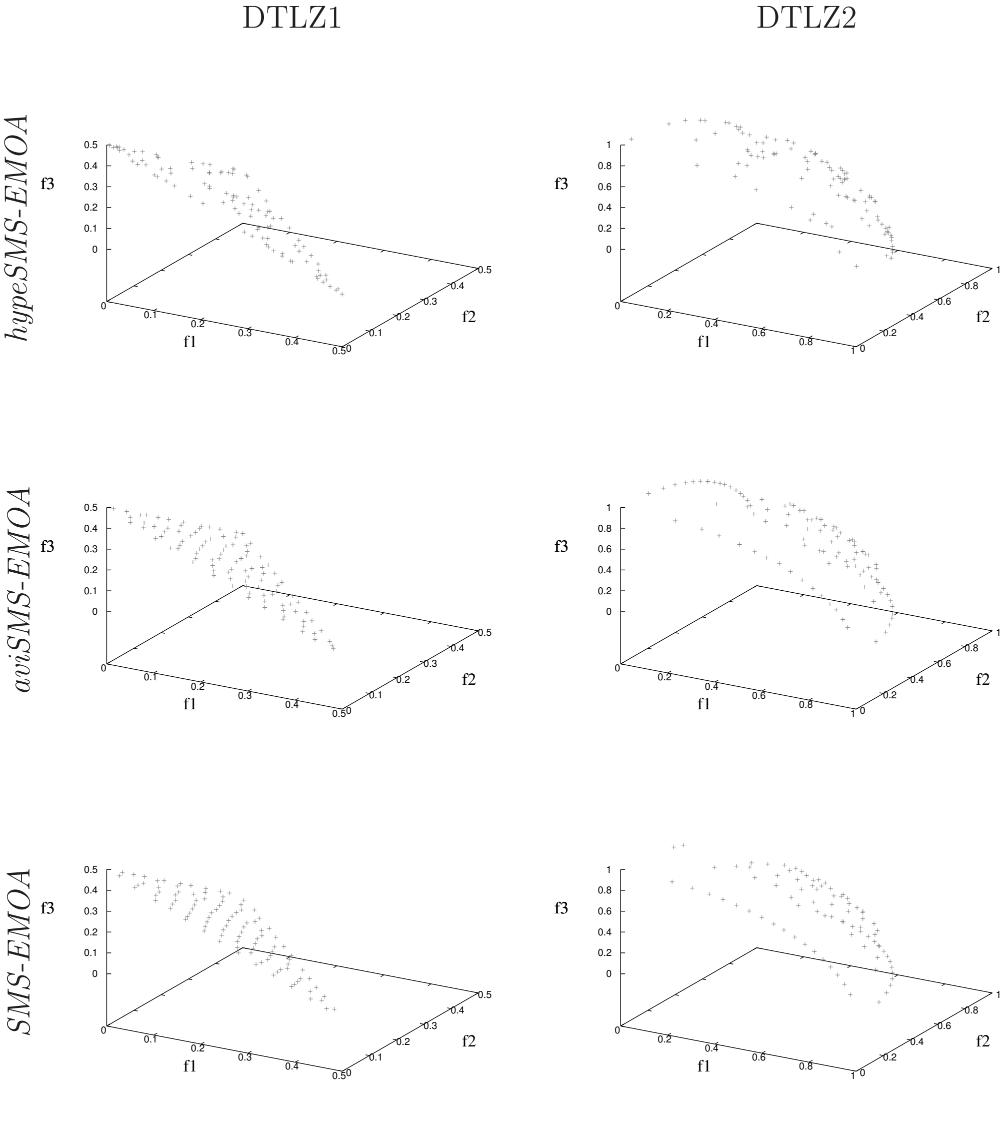


Figure 5.15: Pareto fronts obtained by the three algorithms in the median (with respect to the hypervolume indicator) of their independent runs for the test problems DTLZ1 and DTLZ2.

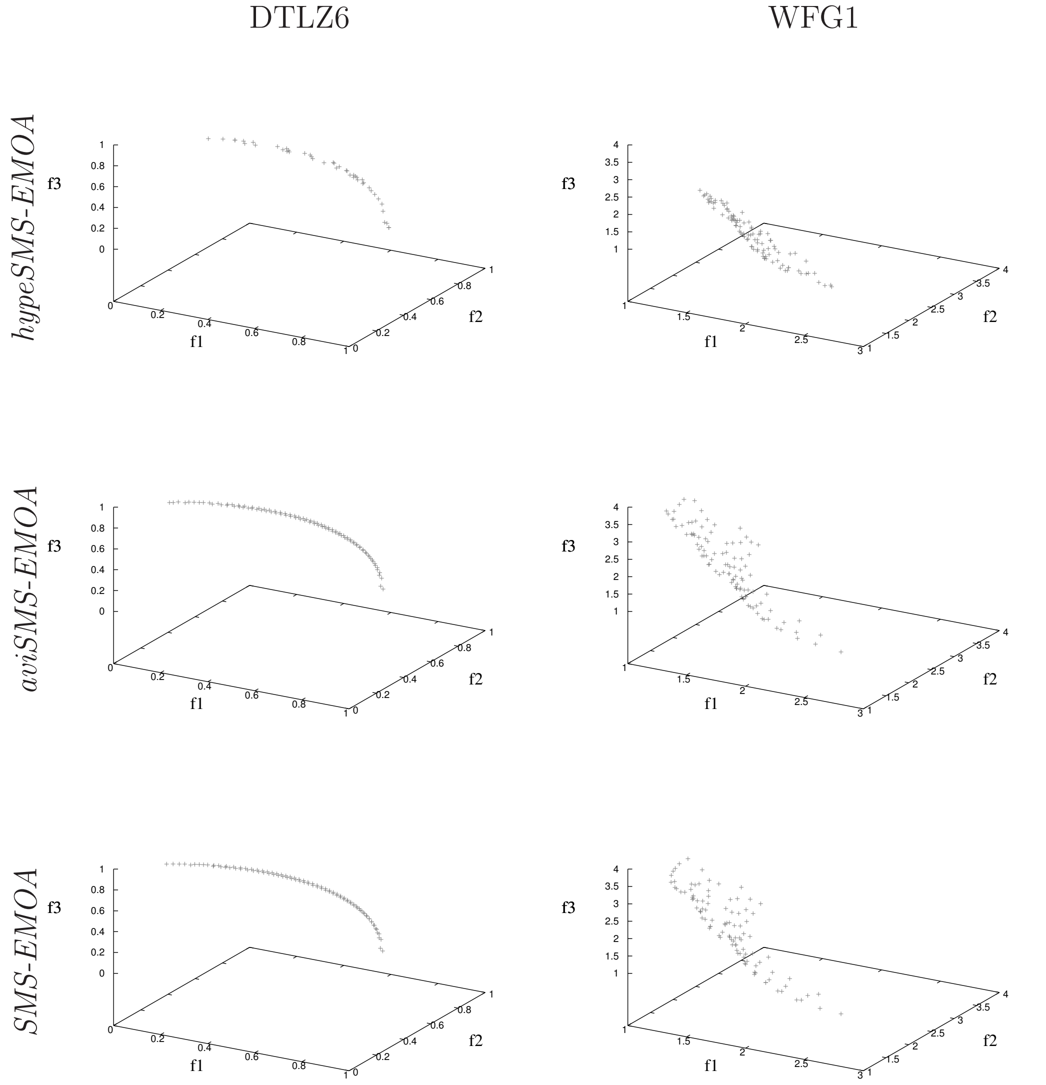


Figure 5.16: Pareto fronts obtained by the three algorithms in the median (with respect to the hypervolume indicator) of their independent runs for the test problems DTLZ6 and WFG1.

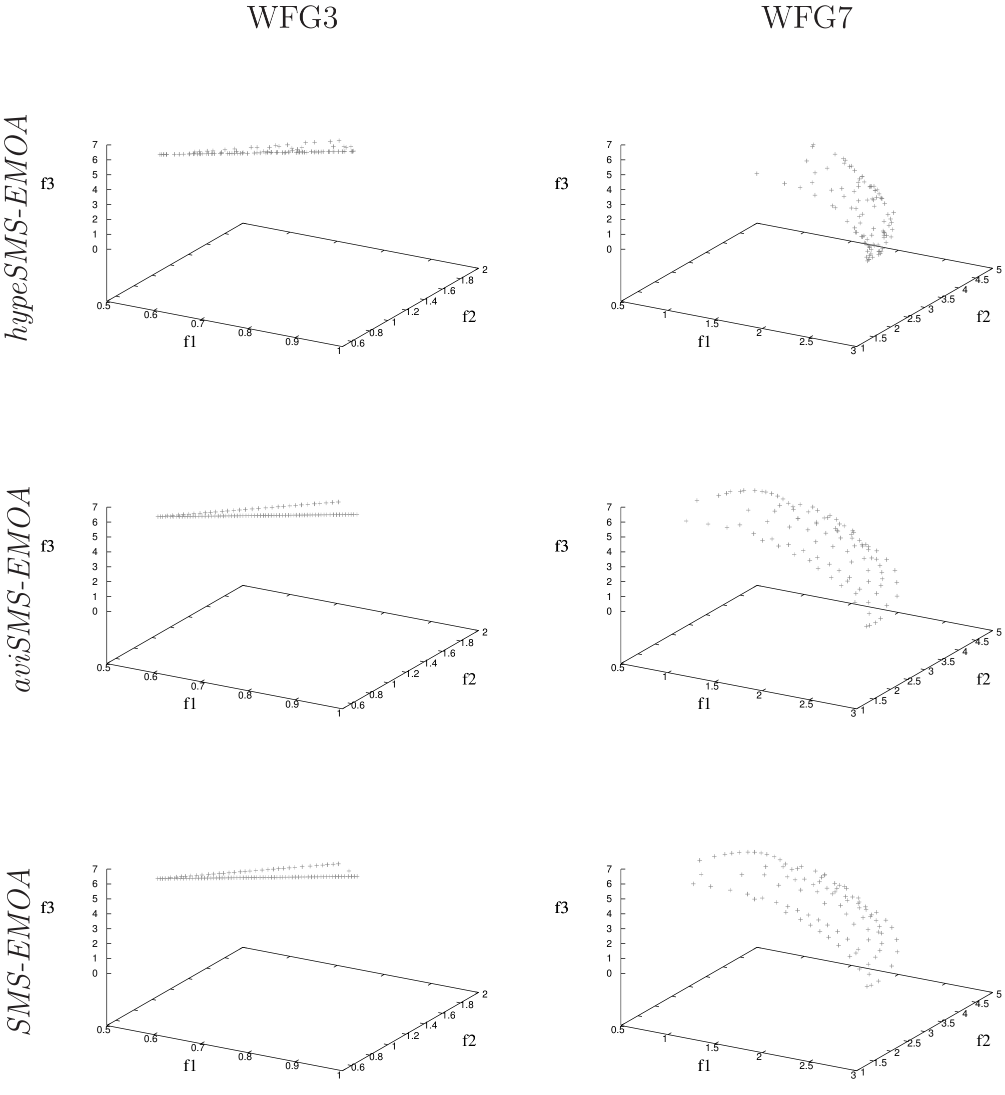


Figure 5.17: Pareto fronts obtained by the three algorithms in the median (with respect to the hypervolume indicator) of their independent runs for the test problems WFG3 and WFG7.

and in which *aviSMS-EMOA* was unable to cover any solution generated by *SMS-EMOA*. Therefore, we can say that only in 26% of the problems (eleven cases) *SMS-EMOA* outperforms our *aviSMS-EMOA* in terms of convergence. In the other 74% of the problems (thirty-one cases) both algorithms had a similar performance in terms of this indicator.

5.3.6 Approximate version of the improved SMS-EMOA vs MOEA/D

Finally, in this section, we compare our *aviSMS-EMOA* with respect to another well-known MOEA which is called *MOEA/D*. We chose this MOEA because it has been an alternative to deal with many-objective optimization problems in recent years and its computational cost is very low, see Chapter 4. For our experiments, we used the version in which *MOEA/D* adopts PBI (Penalty Boundary Intersection) to decompose the MOP. We decided to use PBI because the resulting optimal solutions with PBI are normally much better distributed than those obtained by the Tchebycheff approach [89]. To generate the convex weights we used the technique proposed in [27] and after that, we applied clustering (k-means) to obtain a specific number of weights.

Tables 5.19 and 5.20 show the results for the DTLZ and WFG test problems, with up to six objective functions, with respect to the hypervolume indicator. These tables also present the statistical analysis of our experiments using Wilcoxon's rank sum. From these results, we can say that our *aviSMS-EMOA* outperforms *MOEA/D* in fifty-one problems, it is outperformed by *MOEA/D* in only one problem and both algorithms have a similar behavior in the remaining four problems. Table 5.18 shows that our *aviSMS-EMOA* was able to cover a big percentage of the solutions generated by *MOEA/D* in some problems and *MOEA/D* did not cover any solutions generated by our *aviSMS-EMOA* in many cases. Then, we can say that the convergence of our *aviSMS-EMOA* is better than the convergence of *MOEA/D*. Table 5.21 shows the results with respect to I_{IGD} in problems with three objective functions,[3] the reference sets that we adopted were taken from [21]. In this table, we can see that our *aviSMS-EMOA* is better than *MOEA/D* in most cases (ten out of fourteen) because it obtained a better result according to I_{IGD} and also

[3] We decided to use I_{IGD} only in MOPs with three objective functions because the results of the indicator depend of the reference set that we use and we know that generating a good reference set in MOPs with many objective functions is a difficult task.

the statistical analysis says that we can reject the null hypothesis ("medians are equal"). In one case, both MOEAs have a similar behavior because the null hypothesis cannot be rejected and only in three cases *MOEA/D* outperformed our *aviSMS-EMOA*. Regarding I_{IGD} and the DTLZ test problems, we can say that *MOEA/D* is better in MOPs with concave Pareto fronts (DTLZ2, DTLZ3 and DTLZ4), both MOEAs had a similar behavior in a MOP with a linear Pareto front (DTLZ1) and our *aviSMS-EMOA* is better in MOPs with degenerate Pareto fronts (DTLZ5 and DTLZ6) and with disconnected Pareto fronts (DTLZ7). These results are logical because we know that in linear Pareto fronts both MOEAs converge to a uniform distribution; therefore, we expect that both MOEAS have a similar behavior. In concave Pareto fronts we know that both MOEAs converge to a different distribution, therefore, we expect to obtain different values in the indicator I_{IGD}. Finally, we know that *MOEA/D* has difficulties in MOPs with disconnected Pareto fronts or degenerate Pareto fronts, since it uses a set of well distributed weights (in the whole objective space) to guide the search. With respect to the running time required by each MOEA, *MOEA/D* outperforms our *aviSMS-EMOA* because it only needs a maximum of one second to solve MOPs with 6 objective functions while *aviSMS-EMOA* needs twenty-two minutes.

Finally, we present a brief study on the effect of the population size on the performance of these two MOEAs. It is normally assumed that if we increase the number of objective functions, we should increase the population size as well. However, MOEAs based on I_H are not practical when we use large populations because their computational costs increase rapidly (we need to compute more times the contribution to I_H). Our *aviSMS-EMOA* is more practical in this sense for two reasons: (i) it only needs to calculate three contributions to I_H per iteration regardless of the population size and (ii) it does not compute the exact contributions to I_H, it only approximates them. For our study, we only used the DTLZ2 test problem with 3, 4 and 5 objective functions and we used a population size equal to 300, 350 and 400 individuals, respectively. Table 5.22 shows the results. In (a), we can see that our *aviSMS-EMOA* is better than *MOEA/D* regarding I_H because it obtains better results and we can also reject the null hypothesis in all three cases. In (b), we can see that our *aviSMS-EMOA* is better than *MOEA/D* in terms of convergence in two cases because *aviSMS-EMOA* was able to cover some solutions found by *MOEA/D* and *MOEA/D* could not cover any solution found by *aviSMS-EMOA*. Only in DTLZ2 with three objective functions both MOEAs have a similar behavior because *MOEA/D* was able to cover some solutions found by *aviSMS-EMOA* and, therefore, we cannot say if one of these MOEAs is better. With respect to the running time, we can see in (c) that *MOEA/D* is much faster than our *aviSMS-EMOA* because it only needs two seconds to solve problems with 3, 4 or 5 objective functions while our *aviSMS-EMOA* consumes all the allowable running time (4 hours). Although *MOEA/D* is very fast, it is important to keep in mind that *MOEA/D* needs to generate a well-distributed set of convex weights and this task is not easy when we increase the number of objective functions.

5.3.7 Study: aviSMS-EMOA and the number of samples

As we mentioned before (see Section 5.1), Bringmann and Friedrich proposed a method to approximate the contribution to I_H of one solution. However, their main goal was to find the solution from a set of solutions with the least contribution to I_H. Thus, they present a way in which we can determine the number of samples to guarantee that for any given δ and $\epsilon \geq 0$ the obtained solution is with a probability of $(1-\delta)$, larger by at most a factor of $(1+\epsilon)$ than the least contributor. Also, Nowak et al. [64] made an empirical study about the number of samples in the approximation method proposed by Bringmann and Friedrich to find the least contributor. They used $\delta = 10^{-6}$ and $\epsilon = 10^{-2}$. In one of their experiments in which they used 200 points, the number of samples was: 10^4 for points with dimension equal to 3, between 10^4 and 10^5 for points with dimensions equal to 4 and 5, 10^5 for points with dimension equal to 6, between 10^5 and 10^6 for points with dimension equal to 7, between 10^6 and 10^7 for points with dimension equal to 8, 10^7 and 10^8 for points with dimensions equal to 9 and 10, 10^8 for points with dimension equal to 11, 10^9 for points with dimension equal to 12. And, an interesting thing is that for points with dimension greater than 12 the number of samples can significantly decrease, e.g., for points with dimension equal to 90 the number of samples was between 10^4 and 10^5. Some of their conclusions were the following:

1. there is a dependence between the number of samples and the dimension,

2. the number of samples increases when two or more points differ very

little in their contribution (this was called "hardness of approximation" in the original work by Bringmann and Friedrich [12]) and,

3. the effect "hardness of approximation" seems to be inversely proportional to the dimension when the number of points is fixed. The authors mentioned that this can be attributed to a relatively sparse distribution of points as the dimension increases, leading to fewer occurrences of hard cases.

We do not use the complete algorithm proposed by Bringmann and Friedrich in which they want to find the point with the least contribution to I_H from a set of points. Instead, we only use the way in which the contribution is approximated. We decided to conduct a study, in which we use 10^2, 10^3, 10^4 and 10^5 samples to approximate the contribution to I_H in our *aviSMS-EMOA* and we consider the DTLZ2 test problem with 3, 4, 5, 6, 7, 8, 9 and 10 objective functions. We made our experiments using a population size equal to 100. We decided to use up to 10^5 samples because according to the empirical results presented by Nowak et al. we need at most 10^5 samples for solving problems with dimension greater or equal than 2 and less or equal than 7 and problems with dimension greater or equal than 40 and less or equal than 90 to obtain a good approximation (when $\delta = 10^{-6}$ and $\epsilon = 10^{-2}$ as suggested by Bringmann and Friedrich). And also, for this number of samples our *aviSMS-EMOA* requires less than four hours (maximum allowable running time) to obtain the approximation of the Pareto front. Therefore, we can think that this is a good number of samples that our *aviSMS-EMOA* could adopt. As in the above sections, to calculate the hypervolume indicator, we normalized the approximations of the Pareto optimal set, generated by *aviSMS-EMOA* and we used $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$ as our reference point. The normalization was performed considering all approximations generated by *aviSMS-EMOA* using a different number of samples (i.e., we place, in one set, all nondominated solutions found and from this set we calculate the maximum and minimum for each objective function). In Table 5.23, we can see that *aviSMS-EMOA* significantly improved the quality in its solutions when we increased the number of samples. And, an interesting thing is that in Table 5.24 we can see that our *aviSMS-EMOA* is still faster than *avoSMS-EMOA* (*avoSMS-EMOA* required 8418 seconds to solve the DTLZ2 test problems with six objective funtions using 6000 samples while *aviSMS-EMOA* required 3546 seconds to solve the same problem using 10^5 samples). It is important to mention that the running time decreases as we increase the number of objective functions and this is because *aviSMS-EMOA* only uses the selection mechanism based on I_H and its locality property when, after applying Pareto ranking, we only obtain one front. Therefore, we claim that our *aviSMS-EMOA* is a good option to solve MOPs with low or high dimensionality in objective function space.

$\vec{f}$	avo SMS-EMOA I_H	avi SMS-EMOA I_H	$P(H)$
DTLZ1 (3)	1.119355 (0.000752)	**1.122217 (0.000494)**	**0.000 (1)**
DTLZ2 (3)	0.752558 (0.000827)	**0.758602 (0.000340)**	**0.000 (1)**
DTLZ3 (3)	**1.328649 (0.000456)**	1.328265 (0.000700)	**0.000 (1)**
DTLZ4 (3)	0.869511 (0.000606)	**0.873943 (0.000250)**	**0.000 (1)**
DTLZ5 (3)	0.266479 (0.000073)	**0.266585 (0.000048)**	**0.000 (1)**
DTLZ6 (3)	1.093792 (0.003768)	**1.094423 (0.004341)**	0.739 (0)
DTLZ7 (3)	0.550945 (0.040583)	**0.552256 (0.048770)**	**0.000 (1)**
DTLZ1 (4)	1.359120 (0.003350)	**1.366684 (0.001205)**	**0.000 (1)**
DTLZ2 (4)	1.020956 (0.002523)	**1.035152 (0.001397)**	**0.000 (1)**
DTLZ3 (4)	1.462183 (0.002832)	**1.463597 (0.000130)**	**0.000 (1)**
DTLZ4 (4)	1.021915 (0.001837)	**1.035100 (0.000920)**	**0.000 (1)**
DTLZ5 (4)	0.542823 (0.000578)	**0.545064 (0.000508)**	**0.000 (1)**
DTLZ6 (4)	1.204105 (0.005438)	**1.206271 (0.004888)**	0.057 (0)
DTLZ7 (4)	**0.558208 (0.045049)**	0.550050 (0.061062)	0.228 (0)
DTLZ1 (5)	0.000157 (0.000847)	**1.541811 (0.005706)**	**0.000 (1)**
DTLZ2 (5)	1.283019 (0.007232)	**1.300072 (0.004737)**	**0.000 (1)**
DTLZ3 (5)	1.602555 (0.003153)	**1.608407 (0.003140)**	**0.000 (1)**
DTLZ4 (5)	1.242106 (0.006623)	**1.264793 (0.004457)**	**0.000 (1)**
DTLZ5 (5)	0.928496 (0.001337)	**0.930685 (0.001020)**	**0.000 (1)**
DTLZ6 (5)	**1.519338 (0.002185)**	1.519103 (0.001631)	0.853 (0)
DTLZ7 (5)	0.585090 (0.009424)	**0.589229 (0.020254)**	**0.001 (1)**
DTLZ1 (6)	0.000000 (0.000000)	**1.549649 (0.202423)**	**0.000 (1)**
DTLZ2 (6)	1.643348 (0.006121)	**1.655438 (0.003776)**	**0.000 (1)**
DTLZ3 (6)	1.765905 (0.003950)	**1.770941 (0.000163)**	**0.000 (1)**
DTLZ4 (6)	1.551908 (0.011219)	**1.572087 (0.006064)**	**0.000 (1)**
DTLZ5 (6)	1.023163 (0.001465)	**1.031885 (0.001188)**	**0.000 (1)**
DTLZ6 (6)	**1.634232 (0.003208)**	1.631369 (0.002293)	**0.000 (1)**
DTLZ7 (6)	0.760641 (0.009346)	**0.773352 (0.010180)**	**0.000 (1)**

Table 5.7: Results obtained in the DTLZ test problems by *avoSMS-EMOA* and *aviSMS-EMOA*, using the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	avo SMS-EMOA I_H	avi SMS-EMOA I_H	$P(H)$
WFG1 (3)	**1.213067 (0.017999)**	1.205361 (0.024067)	0.067 (0)
WFG2 (3)	0.773147 (0.090704)	**0.799503 (0.074819)**	0.318 (0)
WFG3 (3)	0.634272 (0.000975)	**0.636772 (0.003130)**	**0.000 (1)**
WFG4 (3)	0.745912 (0.001248)	**0.752357 (0.001500)**	**0.000 (1)**
WFG5 (3)	0.554767 (0.001288)	**0.557581 (0.001661)**	**0.000 (1)**
WFG6 (3)	0.562236 (0.001805)	**0.565928 (0.001409)**	**0.000 (1)**
WFG7 (3)	0.736256 (0.004506)	**0.748956 (0.003767)**	**0.000 (1)**
WFG1 (4)	**1.416596 (0.009155)**	1.414339 (0.008604)	0.245 (0)
WFG2 (4)	0.001735 (0.008153)	**0.300548 (0.254003)**	**0.000 (1)**
WFG3 (4)	0.583248 (0.003331)	**0.592232 (0.006450)**	**0.000 (1)**
WFG4 (4)	1.014467 (0.003034)	**1.030276 (0.003034)**	**0.000 (1)**
WFG5 (4)	0.591930 (0.001643)	**0.598550 (0.001842)**	**0.000 (1)**
WFG6 (4)	0.589722 (0.007086)	**0.610031 (0.006558)**	**0.000 (1)**
WFG7 (4)	0.901884 (0.007163)	**0.915375 (0.006581)**	**0.000 (1)**
WFG1 (5)	**1.548501 (0.006609)**	1.547656 (0.006617)	0.795 (0)
WFG2 (5)	0.022475 (0.038981)	**0.407945 (0.205793)**	**0.000 (1)**
WFG3 (5)	0.568612 (0.017204)	**0.576101 (0.022876)**	0.093 (0)
WFG4 (5)	1.248211 (0.005244)	**1.270178 (0.004494)**	**0.000 (1)**
WFG5 (5)	0.644069 (0.003874)	**0.655659 (0.002092)**	**0.000 (1)**
WFG6 (5)	0.516213 (0.037908)	**0.598269 (0.027623)**	**0.000 (1)**
WFG7 (5)	1.023308 (0.008889)	**1.032123 (0.012809)**	**0.004 (1)**
WFG1 (6)	**1.703600 (0.010057)**	1.691179 (0.014593)	**0.000 (1)**
WFG2 (6)	0.017461 (0.036667)	**0.487496 (0.254908)**	**0.000 (1)**
WFG3 (6)	0.550745 (0.034929)	**0.590424 (0.049438)**	**0.000 (1)**
WFG4 (6)	1.468713 (0.005597)	**1.492717 (0.007779)**	**0.000 (1)**
WFG5 (6)	0.700614 (0.003190)	**0.720380 (0.002684)**	**0.000 (1)**
WFG6 (6)	0.526582 (0.042342)	**0.637238 (0.050036)**	**0.000 (1)**
WFG7 (6)	**1.014516 (0.061564)**	0.907848 (0.066396)	**0.000 (1)**

Table 5.8: Results obtained in the WFG test problems by *avoSMS-EMOA* and *aviSMS-EMOA*, using the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	avo	avi	$\vec{f}$	avo	avi
	SMS-EMOA	SMS-EMOA		SMS-EMOA	SMS-EMOA
	time	time		time	time
DTLZ1 (3)	≈ 4593 s	≈ **385 s**	WFG1 (3)	≈ 13915 s	≈ **1176 s**
DTLZ2 (3)	≈ 10349 s	≈ **889 s**	WFG2 (3)	≈ 4706 s	≈ **474 s**
DTLZ3 (3)	≈ 14300 s	≈ **1298 s**	WFG3 (3)	≈ 6627 s	≈ **554 s**
DTLZ4 (3)	≈ 10482 s	≈ **888 s**	WFG4 (3)	≈ 13945 s	≈ **1230 s**
DTLZ5 (3)	≈ 897 s	≈ **172 s**	WFG5 (3)	≈ 10697 s	≈ **896 s**
DTLZ6 (3)	≈ 5203 s	≈ **409 s**	WFG6 (3)	≈ 9574 s	≈ **736 s**
DTLZ7 (3)	≈ 8469 s	≈ **738 s**	WFG7 (3)	≈ 14415 s	≈ **1351 s**
DTLZ1 (4)	≈ 4215 s	≈ **394 s**	WFG1 (4)	≈ 11621 s	≈ **882 s**
DTLZ2 (4)	≈ 12071 s	≈ **961 s**	WFG2 (4)	≈ 6353 s	≈ **693 s**
DTLZ3 (4)	≈ 11419 s	≈ **1399 s**	WFG3 (4)	≈ 7828 s	≈ **628 s**
DTLZ4 (4)	≈ 12062 s	≈ **948 s**	WFG4 (4)	≈ 14419 s	≈ **1238 s**
DTLZ5 (4)	≈ 4679 s	≈ **495 s**	WFG5 (4)	≈ 10827 s	≈ **878 s**
DTLZ6 (4)	≈ 5794 s	≈ **567 s**	WFG6 (4)	≈ 10351 s	≈ **799 s**
DTLZ7 (4)	≈ 11439 s	≈ **974 s**	WFG7 (4)	≈ 14319 s	≈ **1129 s**
DTLZ1 (5)	≈ 5488 s	≈ **350 s**	WFG1 (5)	≈ 7432 s	≈ **614 s**
DTLZ2 (5)	≈ 9542 s	≈ **798 s**	WFG2 (5)	≈ 9955 s	≈ **1007 s**
DTLZ3 (5)	≈ 12807 s	≈ **1346 s**	WFG3 (5)	≈ 9945 s	≈ **815 s**
DTLZ4 (5)	≈ 9558 s	≈ **776 s**	WFG4 (5)	≈ 13473 s	≈ **1091 s**
DTLZ5 (5)	≈ 5950 s	≈ **557 s**	WFG5 (5)	≈ 13090 s	≈ **1123 s**
DTLZ6 (5)	≈ 6878 s	≈ **649 s**	WFG6 (5)	≈ 12392 s	≈ **1000 s**
DTLZ7 (5)	≈ 13077 s	≈ **1135 s**	WFG7 (5)	≈ 12685 s	≈ **992 s**
DTLZ1 (6)	≈ 8364 s	≈ **436 s**	WFG1 (6)	≈ 5974 s	≈ **558 s**
DTLZ2 (6)	≈ 8418 s	≈ **750 s**	WFG2 (6)	≈ 13436 s	≈ **1339 s**
DTLZ3 (6)	≈ 14410 s	≈ **1164 s**	WFG3 (6)	≈ 11710 s	≈ **1058 s**
DTLZ4 (6)	≈ 8426 s	≈ **726 s**	WFG4 (6)	≈ 12737 s	≈ **1042 s**
DTLZ5 (6)	≈ 6913 s	≈ **662 s**	WFG5 (6)	≈ 14413 s	≈ **1332 s**
DTLZ6 (6)	≈ 7954 s	≈ **754 s**	WFG6 (6)	≈ 14416 s	≈ **1359 s**
DTLZ7 (6)	≈ 13879 s	≈ **1324 s**	WFG7 (6)	≈ 12346 s	≈ **992 s**

Table 5.9: Time required by *avoSMS-EMOA* and *aviSMS-EMOA* for the test problems adopted. s = seconds. Both algorithms were compiled using the GNU C compiler and they were executed on a computer with a 2.66GHz processor and 4GB in RAM.

$\vec{f}$	avo	avi	avi
	SMS-EMOA	SMS-EMOA	SMS-EMOA
	success	success	diff
DTLZ1 (3)	0.010551 (0.003024)	**0.973352 (0.002994)**	**0.898300 (0.002532)**
DTLZ2 (3)	0.010578 (0.003577)	**0.976261 (0.001304)**	**0.946450 (0.001322)**
DTLZ3 (3)	0.008350 (0.003084)	**0.953463 (0.011867)**	**0.942000 (0.005657)**
DTLZ4 (3)	0.009864 (0.002434)	**0.968088 (0.001255)**	**0.946100 (0.001700)**
DTLZ5 (3)	0.009462 (0.001427)	**0.869048 (0.009573)**	**0.920750 (0.001757)**
DTLZ6 (3)	0.011180 (0.001651)	**0.967014 (0.015320)**	**0.714150 (0.085841)**
DTLZ7 (3)	0.009721 (0.003322)	**0.990437 (0.001576)**	**0.937000 (0.002449)**
DTLZ1 (4)	0.009780 (0.002375)	**0.929687 (0.004549)**	**0.899200 (0.002768)**
DTLZ2 (4)	0.009171 (0.002870)	**0.958048 (0.001183)**	**0.946500 (0.000975)**
DTLZ3 (4)	0.007843 (0.002046)	**0.871917 (0.022897)**	**0.923600 (0.009324)**
DTLZ4 (4)	0.009949 (0.002545)	**0.937088 (0.001944)**	**0.942850 (0.001621)**
DTLZ5 (4)	0.009950 (0.002343)	**0.945747 (0.002642)**	**0.943850 (0.002725)**
DTLZ6 (4)	0.010536 (0.002724)	**0.963123 (0.002079)**	**0.930850 (0.003198)**
DTLZ7 (4)	0.009045 (0.002350)	**0.983524 (0.004796)**	**0.950350 (0.001108)**
WFG1 (3)	0.009368 (0.002253)	**0.981253 (0.001282)**	**0.954100 (0.000768)**
WFG2 (3)	0.010965 (0.002512)	**0.976954 (0.004540)**	**0.930550 (0.005005)**
WFG3 (3)	0.009335 (0.001776)	**0.986898 (0.000819)**	**0.952900 (0.001221)**
WFG4 (3)	0.009588 (0.003385)	**0.997338 (0.000251)**	**0.955500 (0.000742)**
WFG5 (3)	0.009214 (0.003495)	**0.981864 (0.000761)**	**0.958000 (0.001000)**
WFG6 (3)	0.009902 (0.002373)	**0.986261 (0.000806)**	**0.951150 (0.000963)**
WFG7 (3)	0.009955 (0.003873)	**0.986393 (0.000630)**	**0.963250 (0.000887)**
WFG1 (4)	0.008880 (0.003976)	**0.970744 (0.001421)**	**0.953550 (0.001396)**
WFG2 (4)	0.011041 (0.002897)	**0.930375 (0.010810)**	**0.931550 (0.003471)**
WFG3 (4)	0.010296 (0.004187)	**0.979382 (0.001668)**	**0.952700 (0.001453)**
WFG4 (4)	0.009413 (0.002984)	**0.994394 (0.000436)**	**0.959100 (0.000943)**
WFG5 (4)	0.009564 (0.003095)	**0.981475 (0.000773)**	**0.958550 (0.000865)**
WFG6 (4)	0.010444 (0.002686)	**0.971220 (0.002110)**	**0.951650 (0.001152)**
WFG7 (4)	0.008602 (0.004414)	**0.984837 (0.000790)**	**0.961850 (0.000726)**

Table 5.10: Success rate (column called "success") achieved by both, *avoSMS-EMOA* and *aviSMS-EMOA*. Since the two selection mechanisms delete the individual with the worst contribution, we define success when the following occurs: When the algorithm deletes the true worst individual (in order to know which is the true worst individual, we compute the exact contribution). In the case of *aviSMS-EMOA*, we consider the worst individual among three individuals (new, near and rand). In the case of *avoSMS-EMOA*, we consider the worst individual among all individuals in the population. The column called "diff" shows the percentage in which the new solution and its nearest neighbor had different values in their contribution to I_H (this column can only be applied to *aviSMS-EMOA*).

$\vec{f}$	hype	avi	$P(H)$
	SMS-EMOA	SMS-EMOA	
	I_H	I_H	
DTLZ1 (3)	1.101012 (0.006067)	**1.122217 (0.000494)**	**0.000 (1)**
DTLZ2 (3)	0.743232 (0.002097)	**0.758602 (0.000340)**	**0.000 (1)**
DTLZ3 (3)	**1.328533 (0.000169)**	1.328265 (0.000700)	0.487 (0)
DTLZ4 (3)	0.863879 (0.001943)	**0.873943 (0.000250)**	**0.000 (1)**
DTLZ5 (3)	0.265428 (0.000238)	**0.266585 (0.000048)**	**0.000 (1)**
DTLZ6 (3)	1.082554 (0.015066)	**1.094423 (0.004341)**	**0.000 (1)**
DTLZ7 (3)	0.534802 (0.033129)	**0.552256 (0.048770)**	**0.000 (1)**
DTLZ1 (4)	1.258490 (0.056806)	**1.366684 (0.001205)**	**0.000 (1)**
DTLZ2 (4)	1.008560 (0.003414)	**1.035152 (0.001397)**	**0.000 (1)**
DTLZ3 (4)	**1.463624 (0.000067)**	1.463597 (0.000130)	0.695 (0)
DTLZ4 (4)	1.014310 (0.004246)	**1.035100 (0.000920)**	**0.000 (1)**
DTLZ5 (4)	0.518095 (0.004575)	**0.545064 (0.000508)**	**0.000 (1)**
DTLZ6 (4)	1.056882 (0.017816)	**1.206271 (0.004888)**	**0.000 (1)**
DTLZ7 (4)	0.507082 (0.030202)	**0.550050 (0.061062)**	**0.002 (1)**
DTLZ1 (5)	1.237122 (0.348331)	**1.541811 (0.005706)**	**0.000 (1)**
DTLZ2 (5)	1.281626 (0.004608)	**1.300072 (0.004737)**	**0.000 (1)**
DTLZ3 (5)	**1.608858 (0.000222)**	1.608407 (0.003140)	**0.004 (1)**
DTLZ4 (5)	1.254611 (0.005040)	**1.264793 (0.004457)**	**0.000 (1)**
DTLZ5 (5)	0.871195 (0.008001)	**0.930685 (0.001020)**	**0.000 (1)**
DTLZ6 (5)	1.433997 (0.008634)	**1.519103 (0.001631)**	**0.000 (1)**
DTLZ7 (5)	0.473364 (0.056567)	**0.589229 (0.020254)**	**0.000 (1)**
DTLZ1 (6)	1.500805 (0.234972)	**1.549649 (0.202423)**	0.162 (0)
DTLZ2 (6)	**1.658514 (0.002679)**	1.655438 (0.003776)	**0.000 (1)**
DTLZ3 (6)	**1.771045 (0.000047)**	1.770941 (0.000163)	**0.000 (1)**
DTLZ4 (6)	**1.583394 (0.003860)**	1.572087 (0.006064)	**0.000 (1)**
DTLZ5 (6)	0.941583 (0.011495)	**1.031885 (0.001188)**	**0.000 (1)**
DTLZ6 (6)	1.460519 (0.022557)	**1.631369 (0.002293)**	**0.000 (1)**
DTLZ7 (6)	0.450393 (0.124927)	**0.773352 (0.010180)**	**0.000 (1)**

Table 5.11: Comparison of the results obtained in the DTLZ test problems by *hypeSMS-EMOA* and *aviSMS-EMOA*, with respect to the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	hype	avi	$P(H)$
	SMS-EMOA	SMS-EMOA	
	I_H	I_H	
WFG1 (3)	1.017526 (0.067671)	**1.205361 (0.024067)**	**0.000 (1)**
WFG2 (3)	0.647049 (0.054860)	**0.799503 (0.074819)**	**0.000 (1)**
WFG3 (3)	0.606017 (0.006630)	**0.636772 (0.003130)**	**0.000 (1)**
WFG4 (3)	0.701693 (0.005039)	**0.752357 (0.001500)**	**0.000 (1)**
WFG5 (3)	0.537158 (0.002760)	**0.557581 (0.001661)**	**0.000 (1)**
WFG6 (3)	0.547413 (0.003932)	**0.565928 (0.001409)**	**0.000 (1)**
WFG7 (3)	0.558748 (0.028965)	**0.748956 (0.003767)**	**0.000 (1)**
WFG1 (4)	1.147478 (0.026432)	**1.414339 (0.008604)**	**0.000 (1)**
WFG2 (4)	**0.422296 (0.256235)**	0.300548 (0.254003)	0.084 (0)
WFG3 (4)	0.527767 (0.016347)	**0.592232 (0.006450)**	**0.000 (1)**
WFG4 (4)	0.932772 (0.008365)	**1.030276 (0.003034)**	**0.000 (1)**
WFG5 (4)	0.558728 (0.005313)	**0.598550 (0.001842)**	**0.000 (1)**
WFG6 (4)	0.562704 (0.011569)	**0.610031 (0.006558)**	**0.000 (1)**
WFG7 (4)	0.417145 (0.032298)	**0.915375 (0.006581)**	**0.000 (1)**
WFG1 (5)	1.245126 (0.027633)	**1.547656 (0.006617)**	**0.000 (1)**
WFG2 (5)	**0.496898 (0.219141)**	0.407945 (0.205793)	0.176 (0)
WFG3 (5)	0.424716 (0.034430)	**0.576101 (0.022876)**	**0.000 (1)**
WFG4 (5)	1.116985 (0.018688)	**1.270178 (0.004494)**	**0.000 (1)**
WFG5 (5)	0.564973 (0.012296)	**0.655659 (0.002092)**	**0.000 (1)**
WFG6 (5)	0.437411 (0.035684)	**0.598269 (0.027623)**	**0.000 (1)**
WFG7 (5)	0.309892 (0.022794)	**1.032123 (0.012809)**	**0.000 (1)**
WFG1 (6)	1.356902 (0.031802)	**1.691179 (0.014593)**	**0.000 (1)**
WFG2 (6)	0.369049 (0.247980)	**0.487496 (0.254908)**	0.072 (0)
WFG3 (6)	0.353758 (0.043983)	**0.590424 (0.049438)**	**0.000 (1)**
WFG4 (6)	1.282154 (0.024684)	**1.492717 (0.007779)**	**0.000 (1)**
WFG5 (6)	0.523809 (0.020982)	**0.720380 (0.002684)**	**0.000 (1)**
WFG6 (6)	0.369149 (0.052829)	**0.637238 (0.050036)**	**0.000 (1)**
WFG7 (6)	0.264190 (0.017844)	**0.907848 (0.066396)**	**0.000 (1)**

Table 5.12: Comparison of the results obtained in the WFG test problems by *hypeSMS-EMOA* and *aviSMS-EMOA*, with respect to the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	hype sms-emoa time	avi SMS-EMOA time	$\vec{f}$	hype sms-emoa time	avi SMS-EMOA time
DTLZ1 (3)	≈ **47 s**	≈ 385 s	WFG1 (3)	≈ **147 s**	≈ 1176 s
DTLZ2 (3)	≈ **106 s**	≈ 889 s	WFG2 (3)	≈ **98 s**	≈ 474 s
DTLZ3 (3)	≈ **135 s**	≈ 1298 s	WFG3 (3)	≈ **148 s**	≈ 554 s
DTLZ4 (3)	≈ **107 s**	≈ 888 s	WFG4 (3)	≈ **107 s**	≈ 1230 s
DTLZ5 (3)	≈ **64 s**	≈ 172 s	WFG5 (3)	≈ **153 s**	≈ 896 s
DTLZ6 (3)	≈ **59 s**	≈ 409 s	WFG6 (3)	≈ **168 s**	≈ 736 s
DTLZ7 (3)	≈ **98 s**	≈ 738 s	WFG7 (3)	≈ **151 s**	≈ 1351 s
DTLZ1 (4)	≈ **59 s**	≈ 394 s	WFG1 (4)	≈ **233 s**	≈ 882 s
DTLZ2 (4)	≈ **156 s**	≈ 961 s	WFG2 (4)	≈ **170 s**	≈ 693 s
DTLZ3 (4)	≈ **165 s**	≈ 1399 s	WFG3 (4)	≈ **247 s**	≈ 628 s
DTLZ4 (4)	≈ **157 s**	≈ 948 s	WFG4 (4)	≈ **157 s**	≈ 1238 s
DTLZ5 (4)	≈ **143 s**	≈ 495 s	WFG5 (4)	≈ **206 s**	≈ 878 s
DTLZ6 (4)	≈ **129 s**	≈ 567 s	WFG6 (4)	≈ **216 s**	≈ 799 s
DTLZ7 (4)	≈ **185 s**	≈ 974 s	WFG7 (4)	≈ **252 s**	≈ 1129 s
DTLZ1 (5)	≈ **79 s**	≈ 350 s	WFG1 (5)	≈ **335 s**	≈ 614 s
DTLZ2 (5)	≈ **188 s**	≈ 798 s	WFG2 (5)	≈ **269 s**	≈ 1007 s
DTLZ3 (5)	≈ **177 s**	≈ 1346 s	WFG3 (5)	≈ **378 s**	≈ 815 s
DTLZ4 (5)	≈ **190 s**	≈ 776 s	WFG4 (5)	≈ **220 s**	≈ 1091 s
DTLZ5 (5)	≈ **229 s**	≈ 557 s	WFG5 (5)	≈ **276 s**	≈ 1123 s
DTLZ6 (5)	≈ **225 s**	≈ 649 s	WFG6 (5)	≈ **274 s**	≈ 1000 s
DTLZ7 (5)	≈ **296 s**	≈ 1135 s	WFG7 (5)	≈ **358 s**	≈ 992 s
DTLZ1 (6)	≈ **98 s**	≈ 436 s	WFG1 (6)	≈ **383 s**	≈ 558 s
DTLZ2 (6)	≈ **233 s**	≈ 750 s	WFG2 (6)	≈ **377 s**	≈ 1339 s
DTLZ3 (6)	≈ **185 s**	≈ 1164 s	WFG3 (6)	≈ **445 s**	≈ 1058 s
DTLZ4 (6)	≈ **234 s**	≈ 726 s	WFG4 (6)	≈ **316 s**	≈ 1042 s
DTLZ5 (6)	≈ **336 s**	≈ 662 s	WFG5 (6)	≈ **246 s**	≈ 1332 s
DTLZ6 (6)	≈ **340 s**	≈ 754 s	WFG6 (6)	≈ **259 s**	≈ 1359 s
DTLZ7 (6)	≈ **377 s**	≈ 1324 s	WFG7 (6)	≈ **408 s**	≈ 992 s

Table 5.13: Time required by *hypeSMS-EMOA* and *aviSMS-EMOA* for the test problems adopted. $s =$ seconds. Both algorithms were compiled using the GNU C compiler and they were executed on a computer with a 2.66GHz processor and 4GB in RAM.

$\vec{f}$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{A})$	$\vec{f}$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{A})$
DTLZ1 (3)	0.002333	**0.031000**	WFG1 (3)	0.000000	**0.003667**
DTLZ2 (3)	0.000000	**0.063333**	WFG2 (3)	0.426333	**0.634667**
DTLZ3 (3)	0.039333	**0.276000**	WFG3 (3)	**0.379333**	0.186333
DTLZ4 (3)	0.000000	**0.067667**	WFG4 (3)	0.078667	**0.256667**
DTLZ5 (3)	0.002000	**0.087333**	WFG5 (3)	0.004333	**0.061000**
DTLZ6 (3)	0.693667	**0.703333**	WFG6 (3)	0.272333	**0.445333**
DTLZ7 (3)	**0.002333**	0.001667	WFG7 (3)	0.003667	**0.020333**
DTLZ1 (4)	0.000000	**0.013333**	WFG1 (4)	0.000000	**0.001000**
DTLZ2 (4)	0.000000	**0.208000**	WFG2 (4)	0.033000	**0.966000**
DTLZ3 (4)	0.002000	**0.169333**	WFG3 (4)	**0.159333**	0.137333
DTLZ4 (4)	0.000000	**0.192667**	WFG4 (4)	0.027000	**0.228333**
DTLZ5 (4)	0.000667	**0.127333**	WFG5 (4)	0.003333	**0.050667**
DTLZ6 (4)	0.226667	**0.308333**	WFG6 (4)	0.086000	**0.387000**
DTLZ7 (4)	0.000333	**0.002000**	WFG7 (4)	0.001000	**0.002667**
DTLZ1 (5)	0.000000	**0.112667**	WFG1 (5)	**0.000000**	0.000000
DTLZ2 (5)	0.000000	**0.327667**	WFG2 (5)	0.029667	**0.966667**
DTLZ3 (5)	0.091333	**0.191000**	WFG3 (5)	0.106667	**0.207667**
DTLZ4 (5)	0.000000	**0.330667**	WFG4 (5)	**0.419000**	0.000667
DTLZ5 (5)	**0.092333**	0.044000	WFG5 (5)	0.002000	**0.044000**
DTLZ6 (5)	**0.273333**	0.134000	WFG6 (5)	0.023667	**0.393667**
DTLZ7 (5)	0.000000	**0.004667**	WFG7 (5)	**0.000000**	0.000000

Table 5.14: Results obtained in the DTLZ and WFG test problems by *aviSMS-EMOA* and *SMS-EMOA*, using the two set coverage indicator (I_{SC}). In this case, $\mathcal{A}$ is the set composed by all solutions found by *aviSMS-EMOA* considering all 30 independent runs and $\mathcal{B}$ is the set composed by all solutions found by *SMS-EMOA* considering all 30 independent runs.

$\vec{f}$	avi SMS-EMOA I_H	SMS-EMOA I_H	$P(H)$
DTLZ1 (3)	1.122217 (0.000494)	**1.123180 (0.000283)**	**0.000 (1)**
DTLZ2 (3)	0.758602 (0.000340)	**0.759983 (0.000048)**	**0.000 (1)**
DTLZ3 (3)	**1.328265 (0.000700)**	1.328074 (0.000341)	0.051 (0)
DTLZ4 (3)	0.873943 (0.000250)	**0.875118 (0.000042)**	**0.000 (1)**
DTLZ5 (3)	0.266585 (0.000048)	**0.266762 (0.000021)**	**0.000 (1)**
DTLZ6 (3)	1.094423 (0.004341)	**1.095866 (0.003607)**	0.145 (0)
DTLZ7 (3)	**0.552256 (0.048770)**	0.548923 (0.056148)	**0.045 (1)**
DTLZ1 (4)	1.366684 (0.001205)	**1.373796 (0.000307)**	**0.000 (1)**
DTLZ2 (4)	1.035152 (0.001397)	**1.046741 (0.000063)**	**0.000 (1)**
DTLZ3 (4)	1.463597 (0.000130)	**1.463689 (0.000083)**	**0.000 (1)**
DTLZ4 (4)	1.035100 (0.000920)	**1.044891 (0.000093)**	**0.000 (1)**
DTLZ5 (4)	0.545064 (0.000508)	**0.546139 (0.000159)**	**0.000 (1)**
DTLZ6 (4)	1.206271 (0.004888)	**1.208642 (0.003669)**	**0.003 (1)**
DTLZ7 (4)	0.550050 (0.061062)	**0.579217 (0.046162)**	**0.000 (1)**
DTLZ1 (5)	1.541811 (0.005706)	**1.566729 (0.000759)**	**0.000 (1)**
DTLZ2 (5)	1.300072 (0.004737)	**1.334594 (0.000329)**	**0.000 (1)**
DTLZ3 (5)	1.608407 (0.003140)	**1.609056 (0.000319)**	**0.004 (1)**
DTLZ4 (5)	1.264793 (0.004457)	**1.299259 (0.000224)**	**0.000 (1)**
DTLZ5 (5)	0.930685 (0.001020)	**0.931459 (0.001521)**	**0.000 (1)**
DTLZ6 (5)	**1.519103 (0.001631)**	1.504474 (0.002545)	**0.000 (1)**
DTLZ7 (5)	0.589229 (0.020254)	**0.599077 (0.019488)**	**0.008 (1)**

Table 5.15: Results obtained in the DTLZ test problems by *aviSMS-EMOA* and *SMS-EMOA*, using the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	avi SMS-EMOA I_H	SMS-EMOA I_H	$P(H)$
WFG1 (3)	1.205361 (0.024067)	**1.210076 (0.025345)**	**0.029 (1)**
WFG2 (3)	0.799503 (0.074819)	**0.809164 (0.067653)**	0.245 (0)
WFG3 (3)	0.636772 (0.003130)	**0.636873 (0.002070)**	0.646 (0)
WFG4 (3)	0.752357 (0.001500)	**0.754175 (0.001647)**	**0.000 (1)**
WFG5 (3)	0.557581 (0.001661)	**0.557814 (0.001690)**	**0.015 (1)**
WFG6 (3)	0.565928 (0.001409)	**0.567213 (0.001614)**	**0.003 (1)**
WFG7 (3)	0.748956 (0.003767)	**0.750817 (0.003654)**	0.055 (0)
WFG1 (4)	1.414339 (0.008604)	**1.422808 (0.008483)**	**0.000 (1)**
WFG2 (4)	0.300548 (0.254003)	**0.861447 (0.126454)**	**0.000 (1)**
WFG3 (4)	0.592232 (0.006450)	**0.599850 (0.006850)**	**0.000 (1)**
WFG4 (4)	1.030276 (0.003034)	**1.038021 (0.002107)**	**0.000 (1)**
WFG5 (4)	0.598550 (0.001842)	**0.599677 (0.001846)**	**0.000 (1)**
WFG6 (4)	0.610031 (0.006558)	**0.616532 (0.006956)**	**0.000 (1)**
WFG7 (4)	0.915375 (0.006581)	**0.925977 (0.007987)**	**0.000 (1)**
WFG1 (5)	**1.547656 (0.006617)**	1.372422 (0.018408)	**0.000 (1)**
WFG2 (5)	0.407945 (0.205793)	**0.913807 (0.125715)**	**0.000 (1)**
WFG3 (5)	0.576101 (0.022876)	**0.590381 (0.027869)**	**0.000 (1)**
WFG4 (5)	**1.270178 (0.004494)**	1.224230 (0.008055)	**0.000 (1)**
WFG5 (5)	0.655659 (0.002092)	**0.658808 (0.002015)**	**0.000 (1)**
WFG6 (5)	0.598269 (0.027623)	**0.631408 (0.028629)**	**0.000 (1)**
WFG7 (5)	**1.032123 (0.012809)**	0.753463 (0.054363)	**0.000 (1)**

Table 5.16: Results obtained in the WFG test problems by *aviSMS-EMOA* and *SMS-EMOA*, using the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	avi SMS-EMOA time	SMS-EMOA time	$\vec{f}$	avi SMS-EMOA time	SMS-EMOA time
DTLZ1 (3)	≈ 385 s	≈ **197 s**	WFG1 (3)	≈ 1176 s	≈ **369 s**
DTLZ2 (3)	≈ 889 s	≈ **302 s**	WFG2 (3)	≈ 474 s	≈ **236 s**
DTLZ3 (3)	≈ 1298 s	≈ **502 s**	WFG3 (3)	≈ 554 s	≈ **288 s**
DTLZ4 (3)	≈ 888 s	≈ **303 s**	WFG4 (3)	≈ 1230 s	≈ **340 s**
DTLZ5 (3)	≈ **172 s**	≈ 201 s	WFG5 (3)	≈ 896 s	≈ **342 s**
DTLZ6 (3)	≈ 409 s	≈ **236 s**	WFG6 (3)	≈ 736 s	≈ **298 s**
DTLZ7 (3)	≈ 738 s	≈ **270 s**	WFG7 (3)	≈ 1351 s	≈ **388 s**
DTLZ1 (4)	≈ **394 s**	≈ 1422 s	WFG1 (4)	≈ **882 s**	≈ 3471 s
DTLZ2 (4)	≈ **961 s**	≈ 2527 s	WFG2 (4)	≈ **693 s**	≈ 751 s
DTLZ3 (4)	≈ **1399 s**	≈ 6093 s	WFG3 (4)	≈ **628 s**	≈ 783 s
DTLZ4 (4)	≈ **948 s**	≈ 2589 s	WFG4 (4)	≈ **1238 s**	≈ 2809 s
DTLZ5 (4)	≈ **495 s**	≈ 1695 s	WFG5 (4)	≈ **878 s**	≈ 1067 s
DTLZ6 (4)	≈ **567 s**	≈ 2157 s	WFG6 (4)	≈ **799 s**	≈ 939 s
DTLZ7 (4)	≈ **974 s**	≈ 1402 s	WFG7 (4)	≈ **1129 s**	≈ 2948 s
DTLZ1 (5)	≈ **350 s**	≈ 14431 s	WFG1 (5)	≈ **614 s**	≈ 14463 s
DTLZ2 (5)	≈ **798 s**	≈ 14449 s	WFG2 (5)	≈ **1007 s**	≈ 2486 s
DTLZ3 (5)	≈ **1346 s**	≈ 14474 s	WFG3 (5)	≈ **815 s**	≈ 1424 s
DTLZ4 (5)	≈ **776 s**	≈ 14440 s	WFG4 (5)	≈ **1091 s**	≈ 14456 s
DTLZ5 (5)	≈ **557 s**	≈ 14433 s	WFG5 (5)	≈ **1123 s**	≈ 2742 s
DTLZ6 (5)	≈ **649 s**	≈ 14444 s	WFG6 (5)	≈ **1000 s**	≈ 2738 s
DTLZ7 (5)	≈ **1135 s**	≈ 13256 s	WFG7 (5)	≈ **992 s**	≈ 14445 s

Table 5.17: Time required by *aviSMS-EMOA* and *SMS-EMOA* for the test problems adopted. s = seconds. Both algorithms were compiled using the GNU C compiler and they were executed on a computer with a processor running at 2.66GHz and with 4GB in RAM.

$\vec{f}$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{A})$	$\vec{f}$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{A})$
DTLZ1 (3)	**0.036333**	0.000000	WFG1 (3)	**0.001667**	0.000000
DTLZ2 (3)	**0.002667**	0.000333	WFG2 (3)	**1.000000**	0.004000
DTLZ3 (3)	**0.167667**	0.003333	WFG3 (3)	**0.922000**	0.014667
DTLZ4 (3)	**0.001333**	0.000000	WFG4 (3)	**0.989333**	0.000000
DTLZ5 (3)	**0.501667**	0.020333	WFG5 (3)	**0.085333**	0.050000
DTLZ6 (3)	**0.983333**	0.453667	WFG6 (3)	**0.621000**	0.162667
DTLZ7 (3)	**0.501333**	0.000000	WFG7 (3)	**0.165667**	0.002000
DTLZ1 (4)	**0.000000**	0.000000	WFG1 (4)	**0.000000**	0.000000
DTLZ2 (4)	**0.002667**	0.000000	WFG2 (4)	**1.000000**	0.009333
DTLZ3 (4)	**0.060333**	0.004667	WFG3 (4)	**0.865000**	0.005000
DTLZ4 (4)	**0.001667**	0.000000	WFG4 (4)	**0.811667**	0.000000
DTLZ5 (4)	**0.285667**	0.062333	WFG5 (4)	**0.040667**	0.010000
DTLZ6 (4)	**0.811667**	0.057333	WFG6 (4)	**0.571667**	0.014667
DTLZ7 (4)	**0.124667**	0.000000	WFG7 (4)	**0.002667**	0.000000
DTLZ1 (5)	0.000000	**0.000333**	WFG1 (5)	**0.000000**	0.000000
DTLZ2 (5)	**0.002333**	0.000000	WFG2 (5)	**0.993333**	0.020667
DTLZ3 (5)	**0.052000**	0.008667	WFG3 (5)	**0.936333**	0.000000
DTLZ4 (5)	**0.002333**	0.000000	WFG4 (5)	**0.627333**	0.000000
DTLZ5 (5)	**0.159000**	0.055000	WFG5 (5)	**0.007667**	0.000000
DTLZ6 (5)	**0.423667**	0.133333	WFG6 (5)	**0.381333**	0.010667
DTLZ7 (5)	**0.011333**	0.000000	WFG7 (5)	**0.000000**	0.000000
DTLZ1 (6)	0.000000	**0.006000**	WFG1 (6)	**0.000000**	0.000000
DTLZ2 (6)	**0.000333**	0.000000	WFG2 (6)	**1.000000**	0.002333
DTLZ3 (6)	**0.025000**	0.007667	WFG3 (6)	**0.911333**	0.000000
DTLZ4 (6)	**0.000333**	0.000000	WFG4 (6)	**0.500000**	0.010667
DTLZ5 (6)	**0.148667**	0.053000	WFG5 (6)	**0.000000**	0.000000
DTLZ6 (6)	**0.374000**	0.089333	WFG6 (6)	**0.333000**	0.010000
DTLZ7 (6)	**0.000333**	0.000000	WFG7 (6)	**0.000000**	0.000000

Table 5.18: Results obtained in the DTLZ and WFG test problems by *aviSMS-EMOA* and *MOEA/D*, using the two set coverage indicator (I_{SC}). In this case, $\mathcal{A}$ is the set composed by all solutions found by *aviSMS-EMOA* considering all 30 independent runs and $\mathcal{B}$ is the set composed by all solutions found by *MOEA/D* considering all 30 independent runs.

$\vec{f}$	pbi MOEA/D I_H	avi SMS-EMOA I_H	$P(H)$
DTLZ1 (3)	1.071328 (0.002556)	**1.122217 (0.000494)**	**0.000 (1)**
DTLZ2 (3)	0.718988 (0.000212)	**0.758602 (0.000340)**	**0.000 (1)**
DTLZ3 (3)	1.294000 (0.002000)	**1.328265 (0.000700)**	**0.000 (1)**
DTLZ4 (3)	0.709501 (0.000134)	**0.873943 (0.000250)**	**0.000 (1)**
DTLZ5 (3)	0.246682 (0.000807)	**0.266585 (0.000048)**	**0.000 (1)**
DTLZ6 (3)	0.197818 (0.029819)	**1.094423 (0.004341)**	**0.000 (1)**
DTLZ7 (3)	0.448768 (0.026011)	**0.552256 (0.048770)**	**0.000 (1)**
DTLZ1 (4)	1.311857 (0.003695)	**1.366684 (0.001205)**	**0.000 (1)**
DTLZ2 (4)	0.887228 (0.000914)	**1.035152 (0.001397)**	**0.000 (1)**
DTLZ3 (4)	1.439263 (0.004172)	**1.463597 (0.000130)**	**0.000 (1)**
DTLZ4 (4)	0.878865 (0.001268)	**1.035100 (0.000920)**	**0.000 (1)**
DTLZ5 (4)	0.471816 (0.003958)	**0.545064 (0.000508)**	**0.000 (1)**
DTLZ6 (4)	0.592195 (0.014757)	**1.206271 (0.004888)**	**0.000 (1)**
DTLZ7 (4)	0.337272 (0.008258)	**0.550050 (0.061062)**	0.994 (0)
DTLZ1 (5)	1.506309 (0.008970)	**1.541811 (0.005706)**	**0.000 (1)**
DTLZ2 (5)	0.987833 (0.003838)	**1.300072 (0.004737)**	**0.000 (1)**
DTLZ3 (5)	1.608395 (0.000366)	**1.608407 (0.003140)**	**0.000 (1)**
DTLZ4 (5)	0.982714 (0.003994)	**1.264793 (0.004457)**	**0.000 (1)**
DTLZ5 (5)	0.669155 (0.022872)	**0.930685 (0.001020)**	**0.000 (1)**
DTLZ6 (5)	0.802058 (0.020102)	**1.519103 (0.001631)**	**0.018 (1)**
DTLZ7 (5)	0.075921 (0.070973)	**0.589229 (0.020254)**	0.589 (0)
DTLZ1 (6)	**1.690367 (0.003587)**	1.549649 (0.202423)	**0.000 (1)**
DTLZ2 (6)	0.973263 (0.008725)	**1.655438 (0.003776)**	**0.000 (1)**
DTLZ3 (6)	1.766545 (0.001449)	**1.770941 (0.000163)**	0.661 (0)
DTLZ4 (6)	0.986331 (0.006574)	**1.572087 (0.006064)**	**0.000 (1)**
DTLZ5 (6)	0.585905 (0.017193)	**1.031885 (0.001188)**	**0.000 (1)**
DTLZ6 (6)	0.708188 (0.043127)	**1.631369 (0.002293)**	**0.000 (1)**
DTLZ7 (6)	0.013435 (0.003121)	**0.773352 (0.010180)**	1.000 (0)

Table 5.19: Results obtained in the DTLZ test problems by *MOEA/D* and *aviSMS-EMOA*, using the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	pbi	avi	$P(H)$
	MOEA/D	SMS-EMOA	
	I_H	I_H	
WFG1 (3)	0.910507 (0.016598)	**1.205361 (0.024067)**	**0.000 (1)**
WFG2 (3)	0.145574 (0.198499)	**0.799503 (0.074819)**	**0.000 (1)**
WFG3 (3)	0.499214 (0.025639)	**0.636772 (0.003130)**	**0.000 (1)**
WFG4 (3)	0.595609 (0.013100)	**0.752357 (0.001500)**	**0.000 (1)**
WFG5 (3)	0.471079 (0.010426)	**0.557581 (0.001661)**	**0.000 (1)**
WFG6 (3)	0.453757 (0.006661)	**0.565928 (0.001409)**	**0.000 (1)**
WFG7 (3)	0.494583 (0.056148)	**0.748956 (0.003767)**	**0.000 (1)**
WFG1 (4)	1.100204 (0.057651)	**1.414339 (0.008604)**	**0.000 (1)**
WFG2 (4)	0.007223 (0.031709)	**0.300548 (0.254003)**	**0.000 (1)**
WFG3 (4)	0.287483 (0.034365)	**0.592232 (0.006450)**	**0.000 (1)**
WFG4 (4)	0.652634 (0.025612)	**1.030276 (0.003034)**	**0.000 (1)**
WFG5 (4)	0.366984 (0.015366)	**0.598550 (0.001842)**	**0.000 (1)**
WFG6 (4)	0.268060 (0.015468)	**0.610031 (0.006558)**	**0.000 (1)**
WFG7 (4)	0.293433 (0.036496)	**0.915375 (0.006581)**	**0.000 (1)**
WFG1 (5)	1.206775 (0.062432)	**1.547656 (0.006617)**	**0.000 (1)**
WFG2 (5)	0.029223 (0.064926)	**0.407945 (0.205793)**	**0.000 (1)**
WFG3 (5)	0.191112 (0.031531)	**0.576101 (0.022876)**	**0.000 (1)**
WFG4 (5)	0.640835 (0.023757)	**1.270178 (0.004494)**	**0.000 (1)**
WFG5 (5)	0.238371 (0.013922)	**0.655659 (0.002092)**	**0.000 (1)**
WFG6 (5)	0.193513 (0.027379)	**0.598269 (0.027623)**	**0.000 (1)**
WFG7 (5)	0.218223 (0.014294)	**1.032123 (0.012809)**	**0.000 (1)**
WFG1 (6)	1.167832 (0.030511)	**1.691179 (0.014593)**	**0.000 (1)**
WFG2 (6)	0.003634 (0.018966)	**0.487496 (0.254908)**	**0.000 (1)**
WFG3 (6)	0.040186 (0.034593)	**0.590424 (0.049438)**	**0.000 (1)**
WFG4 (6)	0.591344 (0.028666)	**1.492717 (0.007779)**	**0.000 (1)**
WFG5 (6)	0.153001 (0.017135)	**0.720380 (0.002684)**	**0.000 (1)**
WFG6 (6)	0.152732 (0.038473)	**0.637238 (0.050036)**	**0.000 (1)**
WFG7 (6)	0.189978 (0.014303)	**0.907848 (0.066396)**	**0.000 (1)**

Table 5.20: Results obtained in the WFG test problems by *MOEA/D* and *aviSMS-EMOA*, using the hypervolume indicator. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	pbi MOEA/D I_{IGD}	avi SMS-EMOA I_{IGD}	$P(H)$
DTLZ1 (3)	0.0004 (0.000)	0.0004 (0.000)	0.945 (0)
DTLZ2 (3)	**0.0009 (0.000)**	0.0012 (0.000)	**0.000 (1)**
DTLZ3 (3)	**0.0009 (0.000)**	0.0038 (0.002)	**0.000 (1)**
DTLZ4 (3)	**0.0009 (0.000)**	0.0012 (0.000)	**0.000 (1)**
DTLZ5 (3)	0.0002 (0.000)	**0.0001 (0.000)**	**0.000 (1)**
DTLZ6 (3)	0.0007 (0.000)	**0.0003 (0.000)**	**0.000 (1)**
DTLZ7 (3)	0.0026 (0.003)	**0.0024 (0.002)**	**0.000 (1)**
WFG1 (3)	0.0066 (0.000)	**0.0051 (0.000)**	**0.000 (1)**
WFG2 (3)	0.0138 (0.000)	**0.0124 (0.001)**	**0.000 (1)**
WFG3 (3)	0.0056 (0.000)	**0.0055 (0.000)**	**0.000 (1)**
WFG4 (3)	0.0009 (0.000)	**0.0007 (0.000)**	**0.000 (1)**
WFG5 (3)	0.0022 (0.000)	**0.0021 (0.000)**	**0.000 (1)**
WFG6 (3)	0.0155 (0.000)	**0.0151 (0.000)**	**0.000 (1)**
WFG7 (3)	0.0038 (0.000)	**0.0035 (0.000)**	**0.000 (1)**

Table 5.21: Results obtained in the DTLZ and WFG test problems with three objective functions by *MOEA/D* and *aviSMS-EMOA*, using the inverted generational distance indicator (I_{IGD}). We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. The third column shows the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	$\|\mathcal{P}\|$	pbi MOEA/D I_H	avi SMS-EMOA I_H	$P(H)$
DTLZ2 (3)	300	0.7678(0.000)	**0.7935 (0.000)**	0.000 (1)
DTLZ2 (4)	350	0.9913(0.001)	**1.1046 (0.001)**	0.000 (1)
DTLZ2 (5)	400	1.2083(0.002)	**1.4338 (0.001)**	0.000 (1)

(a)

$\vec{f}$	$\|\mathcal{P}\|$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{A})$
DTLZ2 (3)	300	0.000222	**0.000667**
DTLZ2 (4)	350	0.000000	**0.001143**
DTLZ2 (5)	400	0.000000	**0.001583**

(b)

$\vec{f}$	$\|\mathcal{P}\|$	pbi MOEA/D time	avi SMS-EMOA time
DTLZ2 (3)	300	≈**2.38**	≈2935.09
DTLZ2 (4)	350	≈**2.36**	≈4384.97
DTLZ2 (5)	400	≈**2.74**	≈4551.01

(c)

Table 5.22: Results obtained in the DTLZ2 test problem by *MOEA/D* and *aviSMS-EMOA*. $|\mathcal{P}|$ is the population size. In the case of the hypervolume indicator (I_H), we show average values over 30 independent runs. The values in parentheses correspond to the standard deviations. Also, in the case of I_H, we present the results of the statistical analysis applied to our experiments using Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level. In the case of the two set coverage indicator, all solutions found by our *MOEA/D* were merged in a set called $\mathcal{A}$, considering all the 30 independent runs, and all solutions found by the original *aviSMS-EMOA* are merged in a set called $\mathcal{B}$. In the case of running time (*time*), we present the time required by both MOEAs in seconds. Both algorithms were compiled using the GNU C compiler and they were executed on a computer with a processor running at 2.66GHz and with 4GB in RAM.

$\vec{f}$	100-samples	1000-samples	10000-samples	100000-samples
	I_H	I_H	I_H	I_H
DTLZ2 (3)	0.7456 (0.002)	0.7549 (0.001)	0.7577 (0.000)	**0.7581 (0.000)**
DTLZ2 (4)	0.9670 (0.007)	1.0207 (0.003)	1.0405 (0.001)	**1.0451 (0.000)**
DTLZ2 (5)	**1.0448 (0.000)**	**1.0448 (0.000)**	**1.0448 (0.000)**	**1.0448 (0.000)**
DTLZ2 (6)	1.1124 (0.041)	1.4090 (0.019)	1.5153 (0.009)	**1.5588 (0.002)**
DTLZ2 (7)	1.1884 (0.074)	1.6036 (0.024)	1.7477 (0.007)	**1.8043 (0.003)**
DTLZ2 (8)	1.5072 (0.125)	1.8152 (0.043)	1.9893 (0.010)	**2.0498 (0.004)**
DTLZ2 (9)	.0565 (0.063)	2.0905 (0.046)	2.2676 (0.009)	**2.3135 (0.003)**
DTLZ2 (10)	2.3874 (0.071)	2.3114 (0.074)	2.5323 (0.008)	**2.5683 (0.003)**

Table 5.23: Results obtained in the DTLZ2 test problem with 3, 4, 5, 6, 7, 8, 9 and 10 objective functions by *aviSMS-EMOA* using 10^2, 10^3, 10^4 and 10^5 samples to approximate the contribution to I_H. We show average values over 30 independent runs using the hypervolume indicator I_H. The values in parentheses correspond to the standard deviations.

$\vec{f}$	100-samples	1000-samples	10000-samples	100000-samples
	time	time	time	time
DTLZ2 (3)	**82.050 (5.654)**	178.459 (3.499)	969.871 (15.051)	8904.640 (104.433)
DTLZ2 (4)	**84.605 (3.655)**	160.543 (3.103)	792.204 (10.598)	7464.558 (127.088)
DTLZ2 (5)	**86.760 (3.023)**	135.513 (2.812)	528.903 (8.302)	5135.625 (88.684)
DTLZ2 (6)	**88.786 (2.877)**	126.949 (2.811)	402.015 (4.699)	3546.932 (83.094)
DTLZ2 (7)	**93.726 (3.195)**	125.395 (2.989)	367.851 (3.811)	2914.540 (66.569)
DTLZ2 (8)	**95.947 (1.954)**	110.165 (3.706)	374.036 (4.304)	2812.759 (43.800)
DTLZ2 (9)	**98.796 (3.043)**	112.677 (2.650)	391.567 (3.303)	3146.004 (42.011)
DTLZ2 (10)	**100.950 (3.005)**	115.722 (3.212)	414.836 (4.425)	3311.591 (37.305)

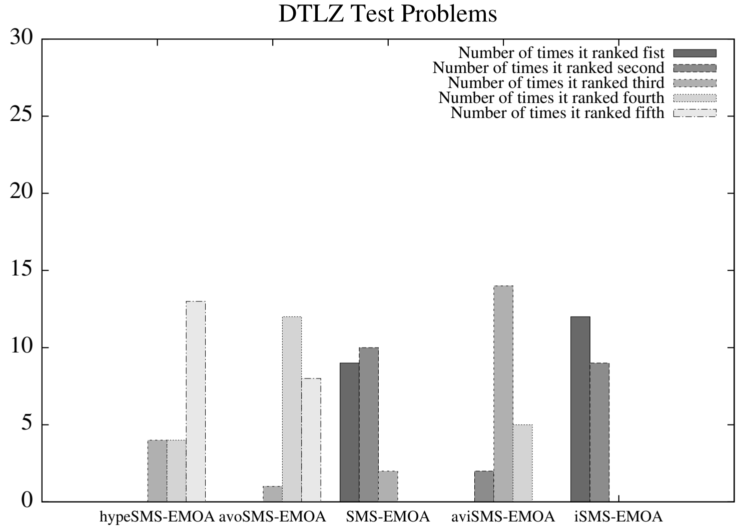
Table 5.24: Time required by *aviSMS-EMOA* for the DTLZ2 test problem with 3, 4, 5, 6, 7, 8, 9 and 10 objective functions, using 10^2, 10^3, 10^4 and 10^5 samples to approximate the contribution to I_H. $s =$ seconds. *aviSMS-EMOA* was compiled using the GNU C compiler and it was executed on a computer with a processor running at 2.66GHz and with 4GB in RAM.

5.3.8 A general comparison among MOEAs based on the hypervolume

In this section, we compare all versions of *SMS-EMOA* studied in this chapter. For our study we consider only the hypervolume indicator (I_H) because the aim of all these MOEAs is to maximize this indicator. For this experiment, we used 10^3 as our number of samples for *hypeSMS-EMOA*, *avoSMS-EMOA* and *aviSMS-EMOA*. Table 5.25 shows the results for the DTLZ test problems. In this case, *iSMS-EMOA* came first in twelve problems and it came second in nine problems, followed by *SMS-EMOA* which was first in nine cases, it came second in ten problems and it came third in two problems. After *SMS-EMOA*, we have that *aviSMS-EMOA* came second in two problems, third in fourteen cases and fourth in five problems, followed by *avoSMS-EMOA* and *hypeSMS-EMOA*. *avoSMS-EMOA* was third in one problem, fourth in twelve problems and fifth in eight cases. *hypeSMS-EMOA* came third in four problems, fourth in four cases and fifth in thirteen problems. In this table, we can also see that *aviSMS-EMOA* is quite competitive with respect to *SMS-EMOA* and *iSMS-EMOA* regarding I_H. For example, only in the DTLZ1 test problem with five objective functions, *aviSMS-EMOA* lost a significant percentage of quality but *avoSMS-EMOA* cannot properly find the Pareto optimal front.

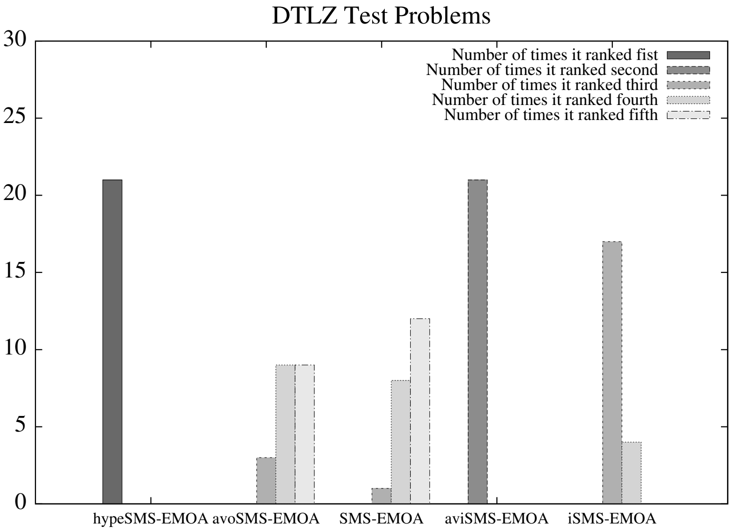
With respect to the WFG test problems, Table 5.27 shows the corresponding results. The best algorithms were *iSMS-EMOA* and *SMS-EMOA*, followed by *aviSMS-EMOA* and *avoSMS-EMOA*, while *hypeSMS-EMOA* came last again. *iSMS-EMOA* was first in nine problems, second in eleven problems and third in one problem. *SMS-EMOA* came first in twelve problems, second in five problems, third in one problem and fourth in three problems. *aviSMS-EMOA* was second in two problems, third in fourteen problems and fourth in five problems. *avoSMS-EMOA* was second in three problems, third in three problems, fourth in eleven problems and fifth in four problems. Finally, *hypeSMS-EMOA* came third in two problems, fourth in two problems and fifth in seventeen problems.

If we consider the running time required by each MOEA to obtain the approximate Pareto front, see Tables 5.26 and 5.28, we can see that the fastest is *hypeSMS-EMOA* followed by *aviSMS-EMOA* and an interesting thing is that *iSMS-EMOA* is faster than *avoSMS-EMOA* in most problems. From these results, we can conclude that *iSMS-EMOA* is an excellent alternative to solve MOPs because is competitive (and even better in some cases) with respect to *SMS-EMOA* but at a much lower computational cost. If we have time constraint, *aviSMS-EMOA* is an excellent alternative because it does not lose much quality in its solutions and it requires at most two minutes to solve MOPs with up to five objective functions.



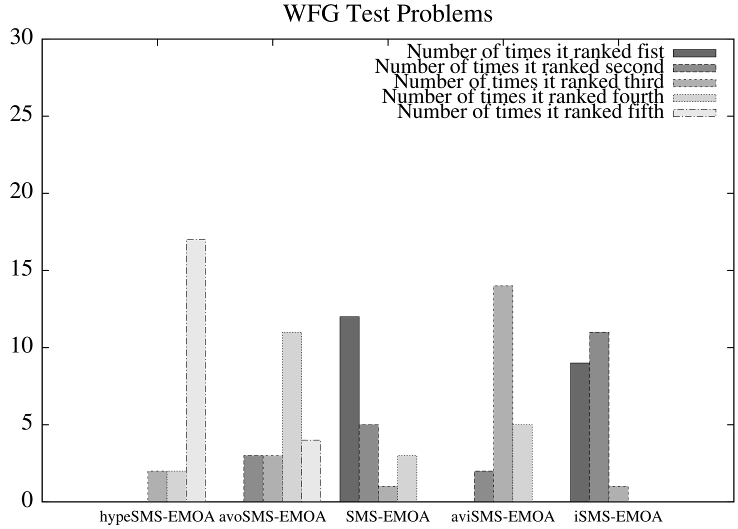
$\vec{f}$	hype SMS-EMOA I_H	avo SMS-EMOA I_H	SMS-EMOA I_H	avi SMS-EMOA I_H	iSMS-EMOA I_H
DTLZ1 (3)	1.0785 (0.016)	1.1194 (0.001)	**1.1230 (0.000)**	1.1211 (0.001)	1.1230 (0.001)
DTLZ2 (3)	0.7415 (0.002)	0.7520 (0.002)	**0.7581 (0.000)**	0.7548 (0.001)	0.7580 (0.000)
DTLZ3 (3)	1.3307 (0.000)	1.3306 (0.000)	1.3308 (0.000)	1.3308 (0.000)	**1.3308 (0.000)**
DTLZ4 (3)	0.7442 (0.002)	0.7524 (0.002)	**0.7585 (0.000)**	0.7551 (0.001)	0.7584 (0.000)
DTLZ5 (3)	0.2651 (0.000)	0.2661 (0.000)	**0.2667 (0.000)**	0.2663 (0.000)	0.2667 (0.000)
DTLZ6 (3)	1.0726 (0.026)	1.0935 (0.004)	**1.0951 (0.004)**	1.0919 (0.004)	1.0951 (0.005)
DTLZ7 (3)	0.5146 (0.050)	0.5286 (0.057)	0.5289 (0.057)	0.5286 (0.057)	**0.5592 (0.044)**
DTLZ1 (4)	1.1833 (0.071)	1.3045 (0.132)	**1.3737 (0.000)**	1.3606 (0.002)	1.3737 (0.000)
DTLZ2 (4)	1.0121 (0.004)	1.0032 (0.005)	**1.0500 (0.000)**	1.0248 (0.003)	1.0496 (0.000)
DTLZ3 (4)	1.4639 (0.000)	1.4536 (0.014)	1.4639 (0.000)	1.4637 (0.000)	**1.4639 (0.000)**
DTLZ4 (4)	1.0192 (0.003)	1.0049 (0.005)	**1.0482 (0.000)**	1.0263 (0.002)	1.0479 (0.000)
DTLZ5 (4)	0.5139 (0.006)	0.5385 (0.004)	0.5470 (0.000)	0.5431 (0.001)	**0.5472 (0.000)**
DTLZ6 (4)	1.0708 (0.023)	1.2090 (0.013)	**1.2411 (0.004)**	1.2352 (0.005)	1.2409 (0.004)
DTLZ7 (4)	0.4994 (0.026)	0.5655 (0.041)	0.5748 (0.032)	0.5729 (0.031)	**0.5932 (0.024)**
DTLZ1 (5)	1.1579 (0.307)	0.0873 (0.316)	1.5665 (0.001)	1.2699 (0.376)	**1.5678 (0.000)**
DTLZ2 (5)	1.3996 (0.007)	1.3262 (0.019)	1.4322 (0.000)	1.3764 (0.007)	**1.4333 (0.000)**
DTLZ3 (5)	1.6102 (0.000)	1.5890 (0.013)	1.6102 (0.000)	1.6078 (0.007)	**1.6103 (0.000)**
DTLZ4 (5)	1.2619 (0.006)	1.1615 (0.017)	1.3040 (0.000)	1.2263 (0.007)	**1.3050 (0.000)**
DTLZ5 (5)	0.8466 (0.015)	0.9156 (0.007)	0.9313 (0.001)	0.9228 (0.002)	**0.9355 (0.001)**
DTLZ6 (5)	1.3821 (0.020)	1.4657 (0.011)	1.5015 (0.002)	1.5048 (0.003)	**1.5190 (0.002)**
DTLZ7 (5)	0.4820 (0.039)	0.5799 (0.015)	0.5858 (0.049)	0.5902 (0.012)	**0.6170 (0.011)**

Table 5.25: Results obtained in the DTLZ test problems with up to five objective functions. We compare *hypeSMS-EMOA*, *avoSMS-EMOA*, *SMS-EMOA*, *aviSMS-EMOA* and *iSMS-EMOA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



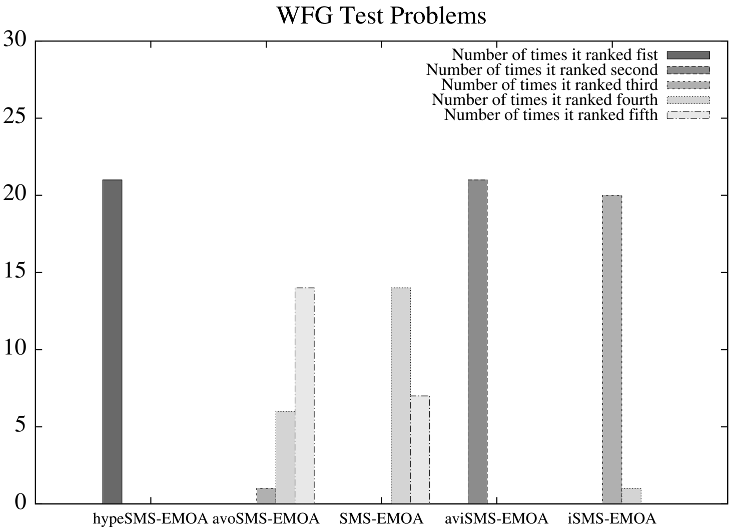
$\vec{f}$	hype SMS-EMOA time	avo SMS-EMOA time	SMS-EMOA time	avi SMS-EMOA time	iSMS-EMOA time
DTLZ1 (3)	**9.97 (0.11)**	1368.84 (56.34)	365.55 (27.89)	42.99 (7.01)	368.47 (40.87)
DTLZ2 (3)	**15.53 (0.25)**	3186.92 (33.95)	486.66 (31.55)	92.84 (12.31)	382.92 (36.38)
DTLZ3 (3)	**11.78 (0.81)**	2516.99 (131.55)	499.11 (39.98)	88.00 (16.68)	440.13 (38.47)
DTLZ4 (3)	**15.64 (0.25)**	3209.57 (28.94)	523.46 (47.80)	87.49 (13.50)	451.08 (36.37)
DTLZ5 (3)	**13.03 (0.15)**	485.14 (13.94)	381.14 (17.11)	25.27 (1.06)	379.85 (27.68)
DTLZ6 (3)	**13.58 (1.57)**	1872.86 (494.69)	403.49 (15.12)	56.57 (13.51)	370.11 (24.92)
DTLZ7 (3)	**19.38 (0.60)**	3047.79 (245.35)	485.55 (29.97)	78.59 (11.63)	382.25 (26.46)
DTLZ1 (4)	**12.86 (0.19)**	953.90 (72.11)	1578.37 (54.45)	40.20 (5.65)	429.29 (13.49)
DTLZ2 (4)	**19.54 (0.31)**	2696.84 (28.35)	2754.86 (100.35)	78.40 (10.97)	441.98 (17.65)
DTLZ3 (4)	**14.36 (0.84)**	1652.69 (191.96)	2390.62 (1153.11)	56.14 (11.07)	470.94 (86.97)
DTLZ4 (4)	**19.58 (0.37)**	3186.07 (23.88)	2884.52 (88.89)	76.02 (8.52)	468.58 (68.70)
DTLZ5 (4)	**22.96 (0.60)**	1559.16 (56.48)	1905.98 (107.56)	51.20 (6.44)	418.97 (14.31)
DTLZ6 (4)	**21.84 (0.61)**	1758.40 (30.01)	2356.92 (50.54)	57.55 (11.47)	436.55 (14.47)
DTLZ7 (4)	**27.02 (1.11)**	3240.00 (293.24)	1844.23 (96.85)	81.36 (17.77)	399.19 (13.25)
DTLZ1 (5)	**15.40 (0.24)**	1029.04 (92.77)	14441.83 (16.54)	41.47 (6.50)	1797.67 (66.81)
DTLZ2 (5)	**22.66 (0.58)**	1879.08 (19.56)	14448.55 (27.68)	67.51 (10.17)	1891.26 (81.42)
DTLZ3 (5)	**16.16 (0.71)**	1617.62 (75.71)	12229.63 (3327.42)	54.23 (8.04)	2392.47 (1266.07)
DTLZ4 (5)	**22.50 (0.42)**	2191.77 (21.18)	14449.72 (31.78)	70.63 (12.46)	2052.96 (87.40)
DTLZ5 (5)	**30.33 (1.08)**	1788.92 (49.23)	14439.46 (21.65)	54.18 (7.43)	1731.52 (109.42)
DTLZ6 (5)	**30.76 (0.97)**	1957.41 (25.55)	14442.16 (24.36)	63.51 (10.03)	1749.24 (41.55)
DTLZ7 (5)	**34.95 (1.85)**	2822.85 (171.83)	14182.28 (696.87)	76.70 (11.13)	768.98 (56.55)

Table 5.26: Results obtained in the DTLZ test problems with up to five objective functions. We compare *hypeSMS-EMOA*, *avoSMS-EMOA*, *SMS-EMOA*, *aviSMS-EMOA* and *iSMS-EMOA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	hype SMS-EMOA I_H	avo SMS-EMOA I_H	SMS-EMOA I_H	avi SMS-EMOA I_H	iSMS-EMOA I_H
WFG1 (3)	0.8864 (0.033)	1.2126 (0.024)	1.2097 (0.029)	1.2099 (0.021)	**1.2162 (0.011)**
WFG2 (3)	0.6629 (0.046)	0.7410 (0.138)	**0.8328 (0.052)**	0.8106 (0.058)	0.8301 (0.071)
WFG3 (3)	0.6003 (0.006)	0.6340 (0.002)	0.6367 (0.002)	0.6356 (0.003)	**0.6378 (0.001)**
WFG4 (3)	0.6999 (0.006)	0.7500 (0.002)	**0.7543 (0.002)**	0.7514 (0.002)	0.7540 (0.002)
WFG5 (3)	0.5372 (0.003)	0.5567 (0.002)	0.5574 (0.002)	0.5565 (0.002)	**0.5575 (0.002)**
WFG6 (3)	0.5538 (0.004)	0.5639 (0.002)	**0.5669 (0.002)**	0.5645 (0.001)	0.5668 (0.001)
WFG7 (3)	0.5004 (0.029)	0.7478 (0.004)	**0.7506 (0.004)**	0.7457 (0.005)	0.7475 (0.004)
WFG1 (4)	0.9985 (0.036)	1.4143 (0.008)	**1.4272 (0.005)**	1.4155 (0.009)	1.4219 (0.009)
WFG2 (4)	0.4703 (0.206)	0.0121 (0.035)	**0.8659 (0.130)**	0.1017 (0.142)	0.8642 (0.120)
WFG3 (4)	0.5189 (0.014)	0.5589 (0.013)	**0.5993 (0.006)**	0.5793 (0.006)	0.5957 (0.006)
WFG4 (4)	0.9048 (0.013)	1.0059 (0.004)	**1.0346 (0.003)**	1.0144 (0.004)	1.0330 (0.003)
WFG5 (4)	0.5542 (0.006)	0.5898 (0.002)	**0.5996 (0.002)**	0.5940 (0.002)	0.5995 (0.002)
WFG6 (4)	0.5712 (0.008)	0.5783 (0.009)	0.6128 (0.006)	0.5982 (0.008)	**0.6143 (0.007)**
WFG7 (4)	0.3559 (0.022)	0.9152 (0.007)	**0.9339 (0.007)**	0.9132 (0.006)	0.9288 (0.004)
WFG1 (5)	1.0722 (0.041)	1.5099 (0.009)	1.3657 (0.018)	1.5296 (0.012)	**1.5656 (0.007)**
WFG2 (5)	0.5457 (0.246)	0.0000 (0.000)	0.8966 (0.164)	0.0824 (0.100)	**0.9278 (0.140)**
WFG3 (5)	0.4618 (0.022)	0.1572 (0.089)	**0.6091 (0.021)**	0.5584 (0.026)	0.6027 (0.024)
WFG4 (5)	1.0734 (0.021)	1.2087 (0.008)	1.2219 (0.009)	1.2322 (0.006)	**1.2777 (0.005)**
WFG5 (5)	0.5626 (0.011)	0.6380 (0.004)	0.6566 (0.002)	0.6466 (0.002)	**0.6570 (0.002)**
WFG6 (5)	0.5370 (0.031)	0.3616 (0.072)	**0.6548 (0.020)**	0.5645 (0.026)	0.6545 (0.018)
WFG7 (5)	0.2558 (0.018)	0.9963 (0.012)	0.7622 (0.036)	0.9821 (0.026)	**1.0197 (0.015)**

Table 5.27: Results obtained in the WFG test problems with up to five objective functions. We compare *hypeSMS-EMOA*, *avoSMS-EMOA*, *SMS-EMOA*, *aviSMS-EMOA* and *iSMS-EMOA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	hype SMS-EMOA time	avo SMS-EMOA time	SMS-EMOA time	avi SMS-EMOA time	iSMS-EMOA time
WFG1 (3)	**25.57 (0.20)**	4308.42 (25.22)	568.27 (54.68)	120.63 (9.37)	419.11 (17.56)
WFG2 (3)	**18.64 (0.86)**	1513.22 (155.00)	431.66 (25.63)	63.94 (12.95)	420.86 (17.31)
WFG3 (3)	**25.24 (0.29)**	2084.27 (17.67)	495.41 (30.94)	70.11 (10.96)	420.08 (16.88)
WFG4 (3)	**20.71 (0.30)**	4153.42 (30.09)	539.65 (46.37)	118.96 (8.80)	415.21 (15.87)
WFG5 (3)	**26.86 (0.13)**	3551.45 (41.54)	531.91 (44.41)	96.25 (15.39)	414.94 (18.54)
WFG6 (3)	**28.46 (0.39)**	3064.41 (27.78)	496.94 (32.74)	80.93 (9.89)	419.77 (17.26)
WFG7 (3)	**26.22 (0.63)**	5333.10 (47.85)	581.82 (58.04)	92.10 (21.52)	417.24 (20.36)
WFG1 (4)	**32.09 (0.44)**	2830.80 (29.26)	3734.08 (180.64)	85.50 (9.91)	468.21 (44.58)
WFG2 (4)	**24.57 (1.17)**	1471.52 (222.98)	862.00 (48.57)	71.12 (8.10)	397.20 (22.68)
WFG3 (4)	**31.92 (0.36)**	1690.03 (18.24)	957.03 (16.88)	63.48 (12.05)	390.21 (17.50)
WFG4 (4)	**26.00 (0.58)**	3919.72 (52.83)	3053.73 (76.49)	96.94 (6.98)	451.87 (41.15)
WFG5 (4)	**36.57 (0.28)**	3103.72 (30.16)	1271.67 (11.72)	82.31 (7.64)	391.19 (18.41)
WFG6 (4)	**37.61 (0.54)**	2524.66 (34.02)	1105.02 (22.98)	71.06 (8.11)	394.10 (23.01)
WFG7 (4)	**34.39 (0.51)**	3888.73 (56.77)	3165.94 (79.75)	66.59 (11.79)	444.58 (39.05)
WFG1 (5)	**37.83 (0.28)**	1695.30 (19.92)	14457.73 (34.41)	64.95 (7.90)	2322.82 (157.15)
WFG2 (5)	**31.00 (1.32)**	1803.49 (258.39)	2567.66 (229.86)	84.83 (15.97)	338.84 (31.95)
WFG3 (5)	**38.85 (0.41)**	1598.17 (31.16)	1609.66 (40.74)	65.81 (8.28)	264.88 (40.68)
WFG4 (5)	**31.72 (0.77)**	2923.89 (48.86)	14454.37 (34.99)	87.74 (10.15)	1894.03 (106.29)
WFG5 (5)	**44.92 (0.15)**	3149.27 (70.53)	3021.60 (159.83)	87.82 (10.37)	305.53 (32.37)
WFG6 (5)	**46.20 (0.44)**	2355.60 (41.59)	2916.74 (216.33)	74.18 (7.17)	314.76 (30.78)
WFG7 (5)	**41.71 (0.40)**	2869.61 (55.36)	14459.32 (26.03)	63.24 (12.14)	1597.02 (97.93)

Table 5.28: Results obtained in the WFG test problems with up to five objective functions. We compare *hypeSMS-EMOA*, *avoSMS-EMOA*, *SMS-EMOA*, *aviSMS-EMOA* and *iSMS-EMOA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.

Chapter 6

New selection mechanisms based on the Maximin Fitness Function

In previous chapters, we studied some selection mechanisms used by well-known MOEAs. However, we could see that all of them have some disadvantages. MOEAs based on Pareto dominace have difficulties when we want to solve MOPs with many objective functions. MOEAs based on the hypervolume indicator (I_H) are computationally expensive, since selecting individuals with I_H is an NP-hard problem with respect to the number of objective functions. Therefore, the use of a selection mechanism based on the hypervolume indicator is impractical to solve MOPs with more than five objective functions. MOEAs based on decomposition need to generate a set of well-distributed weight vectors and this task becomes harder as we increase the number of objective functions. Finally, the Maximin Fitness Function (MFF) appears to be a good option because it has interesting properties and it also has a linear complexity with respect to the number of objective functions. However, until now, it had only been used in problems with low dimensionality (two or three objective functions). Therefore, in this chapter, we study the properties of MFF in depth and we identify some of its disadvantages. Finally, we propose some ways to overcome such disadvantages.

6.1 Reviewing the Properties of the Maximin Fitness Function

As we mentioned in Chapter 4, the Maximin Fitness Function (MFF) [4] is defined as:

$$fitness^i = max_{j \neq i}(min_k(f_k^i - f_k^j)) \tag{6.1}$$

where the *min* is taken over all the objective functions, and the *max* is taken over all individuals in the population, except for the same individual i. There is also a modified version [6] which is defined as follows:

$$fitness^i = max_{j \neq i, j \in \mathcal{ND}}(min_k(f_k^i - f_k^j)) \tag{6.2}$$

where $\mathcal{ND}$ is the set of nondominated individuals. If we use eq. (6.2), we guarantee that the fitness of a nondominated individual is controlled only by its non-dominance and then, we only penalize clustering between nondominated individuals. It is interesting to observe that MFF allows to design, in an easy way, an interactive method to solve MOPs when the decision maker can define preferences. For example, at each iteration of the algorithm, we can present to the decision maker the set of nondominated solutions obtained so far, and then he/she can choose which solutions will be considered to calculate the fitness of each solution in the population. In our case, we use the following equation to assign fitness:

$$fitness^i = max_{j \neq i, j \in \mathcal{A}}(min_k(f_k^i - f_k^j)) \tag{6.3}$$

where $\mathcal{A}$ is the set of nondominated individuals which were chosen by the decision maker.

6.1.1 Disadvantages of the maximin fitness function

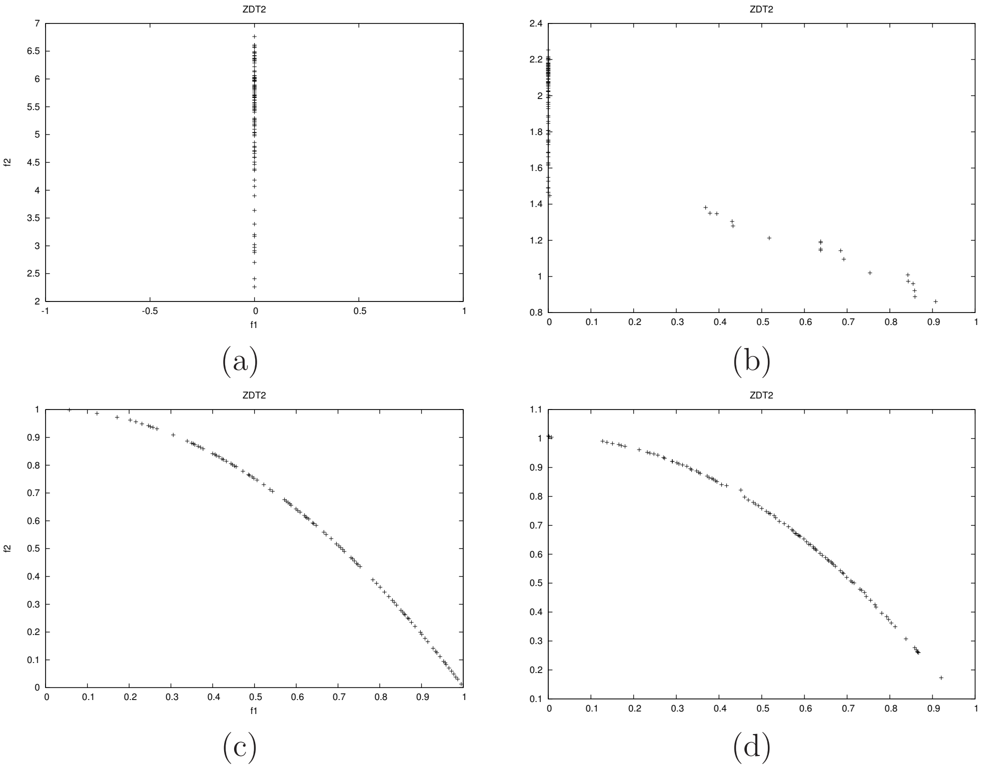
For analyzing the behavior of the MFF, we designed two MOEAs. The first algorithm adopts the crossover and mutation operators of *NSGA-II* to create the offspring population. After that, it combines the population of parents and offspring and uses MFF to assign fitness to each individual. Finally, the best individuals according to their fitness values are selected to be part of the next generation. The second algorithm works in the same way that the first algorithm but instead of adopting the crossover and mutation operators of *NSGA-II*, it adopts the operators of differential evolution (*DE*) [67] to create new individuals. Figures 6.1(a) and 6.1(b) show the results obtained by each of these two MOEAs in the ZDT2 test problem which is defined as follows:

$$\begin{aligned} f_1(\vec{x}) &= x_1 \\ f_2(\vec{x}) &= g(\vec{x})\left(1 - (x_1/g(\vec{x}))^2\right) \\ g(\vec{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \end{aligned} \tag{6.4}$$

In these Figures, we can see that the MOEA based on *DE* obtains only weakly Pareto optimal points and the MOEA based on a Genetic Algorithm cannot obtain all the Pareto front. Why is this? If we adopt MFF to solve ZDT2, within an evolutionary algorithm (i.e., we assign the fitness of each individual, using MFF, and then we sort the individuals according to their fitness values), we can obtain many (even only) weakly Pareto points because f_1 is easier to optimize than f_2 and then, we quickly obtain weakly dominated solutions in this extreme of the Pareto front. Figure 6.1(a) shows that if we use Differential Evolution and MFF, we only obtain weakly Pareto optimal points. Figure 6.1(b) shows that if we use a Genetic Algorithm and MFF, the convergence to the Pareto optimal front is slow because we obtain many weakly Pareto optimal points during the search. In [58], the authors proposed a MOEA based on Particle Swarm and MFF, and also reported problems in ZDT2. Due to the foregoing, the following question arises: **Is it better to prefer weakly dominated individuals than dominated individuals?** The answer is that it is not good to prefer weakly dominated individuals or individuals which are close to being weakly dominated (even if they are weakly dominated by any dominated individual). For example, in Figure 6.2(a), it is better to select E instead of A.

In order to address the above problem, we propose the following constraint: *Any individual that we want to select must not be similar (in objective function space) to another (already selected) individual.* The process to verify similarity between two individuals is shown in Algorithm 9.

By adding this constraint, we can find the true Pareto front of ZDT2 when we use a MOEA based on Differential Evolution, see Figure 6.1(c). Also, we speed up convergence when we use a MOEA based on a Genetic Algorithm, see Figure 6.1(d). The complete selection mechanism using MFF and our proposed constraint is shown in Algorithm 10. First, we select individuals doing a similarity checking (see Lines 4-8). If we have considered all individuals in the population and we have not selected the number of



(a) (b)

(c) (d)

Figure 6.1: In (a), we used a MOEA based on Differential Evolution and the maximin fitness function. In (b), we used a MOEA based on a Genetic Algorithm and the maximin fitness function. In cases (c) and (d), we adopted the same MOEAs used in (a) and (b), respectively, but introducing our proposed constraint to verify similarity. In all cases, we used a population size of 100 individuals. In cases (a) and (c), we iterated for 100 generations. Finally, in cases (b) and (d), we iterated for 150 generations.

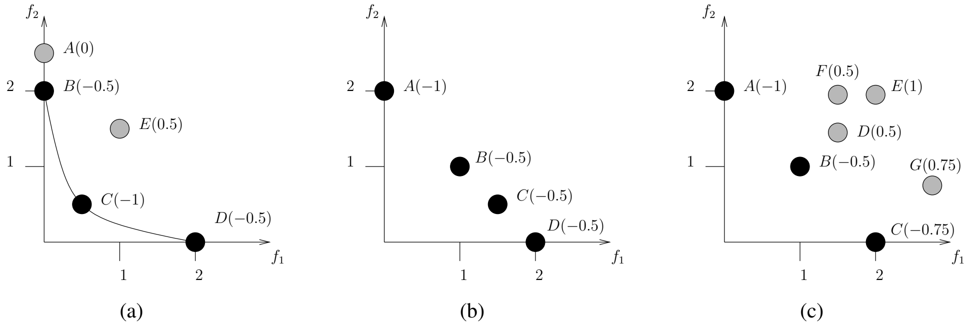


Figure 6.2: In (a), individuals B, C and D are nondominated, individual A is a weakly dominated point and E is a dominated point. In (b), individuals A, B, C and D are nondominated; individuals B, C and D are penalized because they are close from each other. In (c), we can see that the fitness of individuals D, E and F is controlled by the nondominated individual B, and their fitness is a metric of the distance to the individual B. The same occurs with individual G but its fitness is controlled by the nondominated individual C. Also, we can see that the fitness of individual B is affected by the dominated individual D because they are close and the fitness value of individual C is affected by the dominated individual G.

Algorithm 9: Verify similarity

```
Input  : $x$ (individual), $\mathcal{P}$ (population) and $\epsilon$ (minimum difference
         between components).
Output: Returns 1, if the individual $x$ is similar to any individual in
         the population $\mathcal{P}$; otherwise, returns 0.
foreach $y \in \mathcal{P}$ do
    foreach objective function "k" do
        if $|x.\vec{f}[k] - y.\vec{f}[k]| < \epsilon$ then
            return 1;
        end
    end
end
return 0;
```

individuals that we want, we select the remaining individuals considering only the maximin fitness values (see Lines 9-13). One could think that we can use MFF by itself without selecting solutions whose maximin fitness values are equal to zero (because they are weakly dominated). However, it is important to note that the above constraint avoids that we select both: (i) solutions which are weakly dominated by nondominated solutions and (ii) solutions which are weakly dominated by dominated solutions. For example, let's assume that we want to select five individuals in Figure 6.2(c). If we only use MFF, then we select individuals A, C, B, D, F. If we use MFF and the above constraint, then we sort them according to their fitness values: A(-1), C(-0.75), B(-0.5), D(0.5), F(0.5), G(0.75) and E(1). Finally, we select individuals A, C, B, D and we consider the individual F but we do not select it because is similar to individual D (in objective function f_1) which had been already selected. Consequently, we select individual G.

Algorithm 10: Maximin Selection

```
Input  : $\mathcal{P}$ (population), $N$ (number of individuals that we want to
         choose such that $N < |\mathcal{P}|$) and $\epsilon$ (minimum difference
         between objectives).
Output: $\mathcal{S}$ (selected individuals).
/*Sorting with respect to the maximin fitness values                  */
AssignFitness($\mathcal{P}$);
$Sort(\mathcal{P})$;
/*Fill up the new population with the best copies
  according to the maximin fitness values, verifying that
  no solution is similar to one that had been previously
  selected                                                             */
$\mathcal{S} \leftarrow \emptyset$;
foreach $y \in \mathcal{P}$ do
    if $|\mathcal{S}| < N$ and VerifySimilarity($y$, $\mathcal{S}$, $\epsilon$) = 0 then
        $\mathcal{S} \leftarrow \mathcal{S} \cup y$;
    end
end
/*Choose the remaining individuals considering only the
  maximin fitness values                                               */
if $|\mathcal{S}| < N$ then
    foreach $y \in \mathcal{P}$ such that $y$ has not been selected do
        $\mathcal{S} \leftarrow \mathcal{S} \cup y$;
    end
end
return $\mathcal{S}$;
```

The second disadvantage that we identified has to do with the clustering property. In Figure 6.2(b), we can observe that individuals B, C and D have the same maximin fitness. Then, if we use MFF, we can not know which of them is the best individual to form part of the next generation. In order to address this problem, we have proposed some techniques to improve the diversity in the population at each generation. In the next sections, we will describe them, one by one.

6.2 MFF and a Clustering Technique

In this section, we propose a selection mechanism based on MFF and a clustering technique to select solutions from a set of nondominated solutions. This mechanism works as follows. If we want to select N individuals from a population of nondominated individuals called $\mathcal{ND}$, then, we choose the best N individuals with respect to their maximin fitness values, and we use them as centers of the clusters. Then, we proceed to place each individual in its nearest cluster. Finally, for each of the resulting clusters, we recompute the center, and we choose the individual closest to it. This procedure is shown in Algorithm 11. With this technique, if we return to Figure 6.2(b) and we assume that we want to choose two individuals, we can see that, regardless of the individual (B, C or D) that we choose as an initial center of the cluster, we always obtain two clusters: one of them contains individual A, and the other one contains individuals B, C and D. After applying this procedure, we always choose individuals A and C. See Figure 6.3. It is important to note that clustering selection does not iterate many times to improve the distribution of the centers because we choose the initial centers regarding the maximin fitness values and we only want to do a small correction based on the idea that MFF penalizes clustering.

It is necessary to consider that if we want to select from a set which contains dominated solutions, this selection mechanism is not effective. For example, in Figure 6.2(c), if we want to select three individuals, the clustering technique selects individuals A, D and C, penalizing individual B. This is clearly not good because individual B dominates individual D. Therefore, the complete selection mechanism that we propose is a combination of Algorithms 10 and 11. Let's assume that we want to select N individuals from a population $\mathcal{P}$. First, we obtain the set of nondominated solutions which we call "$\mathcal{ND}$". Then, if the number of nondominated solutions is greater than N (i.e., $|\mathcal{ND}| > N$), we use Algorithm 11; otherwise, we use Algorithm 10.

Although with this selection mechanism we are able to address some difficulties of MFF, it still has some disadvantages. For example, if we see Figure 6.4, and we assume that we want to select six individuals, if we only use MFF to select, we would choose individuals: A, B, C, D, E and F. If we use the selection mechanism based on MFF and the above clustering technique, we would choose individuals: A, B, C, D, E and K. None of two results is correct. This is because MFF penalizes all solutions: G, H, ..., O and prefers to select solutions in other parts of the Pareto front, leaving

Algorithm 11: Clustering Selection (setting the centers using maximin)

```
Input  : $\mathcal{ND}$ (population of nondominated individuals) and $N$
         (number of individuals that we want to choose such that
         $N < |\mathcal{P}|$).
Output: $\mathcal{S}$ (selected individuals).
/*Choose the best $N$ individuals, according to maximin
  fitness, as centers of the clusters $\mathcal{C}$                */
AssignFitness($\mathcal{ND}$);
$Sort(\mathcal{ND})$;
for $j \leftarrow 1$ to $N$ do
    $\mu_j = y_j$ such that $y_j \in \mathcal{ND}$;
    $\mathcal{C}_j = \{\emptyset\}$;
end
/*Do one iteration of clustering                                    */
foreach $y \in \mathcal{ND}$ do
    if $\mu_j$ is closest to $y$ then
        $\mathcal{C}_j \leftarrow \mathcal{C}_j \cup y$;
    end
end
/*Obtain the new centers of the clusters                            */
for $j \leftarrow 1$ to $N$ do
    $\mu_j \leftarrow \frac{1}{|\mathcal{C}_j|} \sum_{y_i \in \mathcal{C}_j} y_i$;
end
/*Select individuals who are closest to the centers of the
  clusters                                                          */
$\mathcal{S} \leftarrow \emptyset$;
for $j \leftarrow 1$ to $N$ do
    if $y_i \mid y_i \in \mathcal{C}_j$ is the nearest to the center $\mu_j$ then
        $\mathcal{S} \leftarrow \mathcal{S} \cup y_i$;
    end
end
return $\mathcal{S}$;
```

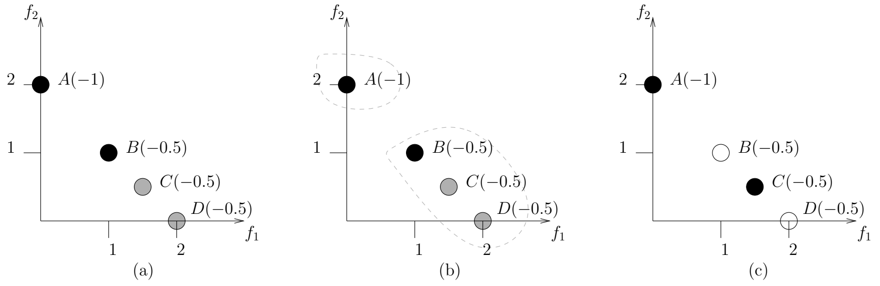


Figure 6.3: Let's assume that we want to select two individuals. If we use only the maximin fitness function and we assume that A, B, C and D is the ordering of the solutions after sorting them with respect to their fitness value, then we select individuals A and B and individuals C and D are not considered (see (a)). This is clearly not a good selection procedure. If we use the clustering technique, we take A and B as initial centers of the clusters and we obtain two clusters: the first one only has A and the second has B, C and D, see (b). When we recalculate the centers of the clusters and choose the closest solution to the centers, we select A and C, see (c).

big gaps in the front. Finally, another question arises with respect to the MFF: what version of the MFF should we use? To answer this question, we designed three selection operators:

- **Operator 1**. It always uses the original MFF.
- **Operator 2**. It adopts the original MFF when it selects individuals using only the MFF and our proposed constraint and, it adopts the modified version of MFF when it selects individuals using MFF and the clustering technique. This idea arises because the original MFF penalizes clustering between nondominated solutions. In fact, a nondominated individual is penalized even if it is near to a dominated solution. Thus, we believe that it is a good idea to use the original MFF at the beginning of the search with the aim of improving the diversity in the population. However, when we have many nondominated solutions and we use the clustering technique, we believe that it is a good idea to use the modified MFF because at this moment we are only interested in a good distribution along the nondominated front and the modified MFF only penalizes clustering between nondominated solutions.
- **Operator 3**. It always uses the modified version of MFF.

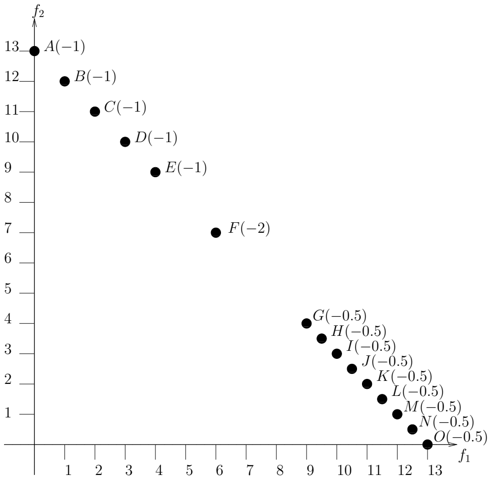


Figure 6.4: The maximin fitness function penalizes all solutions G, H, ... O. This is undesirable, because it leaves gaps when we select individuals.

Table 6.1 shows the results with respect to the hypervolume indicator (I_H) for these three versions using seven DTLZ test problems. We called *MC-V1* to the version that uses operator 1, *MC-V2* to the version that uses operator 2 and *MC-V3* to the version that uses operator 3. To calculate I_H, we normalized the approximations of the Pareto optimal set, generated by the MOEAs, and we used $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$, as our reference point. The normalization was performed considering all approximations generated by the different MOEAs (i.e., we place, in one set, all nondominated solutions found by *MC-V1*, *MC-V2* and *MC-V3* and from this set we calculate the maximum and minimum for each objective function). In this table, we can observe that *MC-V1* was ranked as first only in seven test problems and it was third in twenty-one problems. Also, we can observe that *MC-V2* and *MC-V3* had a very similar behavior and then when we normalized our results, both approaches obtained the same results with respect to I_H. They ranked first in twenty-one problems and ranked second in seven problems. From these results, we decided to use always the modified MFF in the following.

6.3 MFF and Euclidean Distances

Here, we propose to combine MFF with a technique based on Euclidean distances to improve the distribution of the solutions in objective function space. We decided to use Euclidean distances because our aim is that the solutions are uniformly distributed. This selection mechanism works as follows:

Let's assume that we want to select N individuals from a population called $\mathcal{P}$. First, we assign fitness to each individual using the modified version of MFF. Then, we proceed to select individuals according to their fitness value, verifying similarity between the selected individuals (see Algorithm 9). We put the selected individuals in the set called $\mathcal{S}$. If we have already selected N individuals but there are still nondominated individuals which have not participated in the selection process, then we proceed to do the following. For each nondominated individual y who has not participated in the selection process (because its fitness value is low), we obtain its nearest neighbor from $\mathcal{S}$ ($s_{nearest}$) and we choose a random individual from $\mathcal{S}$ (s_{random}, such that $s_{nearest} \neq s_{random}$). We assume that the probability of choosing an individual in a crowded region is higher than the probability of choosing an individual in an unexplored region. Then, y will compete with s_{random} and $s_{nearest}$ to survive. We use $s_{nearest}$ with the idea of improving the diversity locally: If we move $s_{nearest}$ to y, do we increase the distance with respect to its nearest neighbor in $\mathcal{S}$? And, we use s_{random} because we consider the scenario in which the solution $s_{nearest}$ is in an unexplored region and, therefore, it is not a good idea to delete $s_{nearest}$ or y. Therefore, first, y competes with the randomly chosen solution s_{random}: If the Euclidean distance from y to its nearest neighbor in $\mathcal{S}$ is greater than the Euclidean distance from s_{random} to its nearest neighbor in $\mathcal{S}$, we replace s_{random} with y. If y loses the competition, then y competes with its nearest neighbor to survive. If the Euclidean distance from y to its nearest neighbor in $\mathcal{S}$ (without considering $s_{nearest}$) is greater than the Euclidean distance from $s_{nearest}$ to its nearest neighbor in $\mathcal{S}$, then we replace $s_{nearest}$ with y. It is important to mention that if all the objectives are equally important, we need to calculate the Euclidean distance on the normalized values of the objective functions. The complete selection mechanism is shown in Algorithm 12.

Figure 6.5 shows the selection process using MFF and Euclidean distances. Since individuals C and D are not considered in (a), in (b), C competes with A and B, and C replaces B. In (c), D competes with A and C, and D replaces C. With this selection mechanism, if we return to Figure 6.4, we can avoid that the approximate Pareto front has big gaps. Because of that, all individuals G, H, $\cdots$, O have the same fitness value.

```
Algorithm 12: Maximin-Euclidean Selection
Input  : $\mathcal{P}$ (population), $N$ (number of individuals to choose $N < |\mathcal{P}|$).
Output: $\mathcal{S}$ (selected individuals).
/*Sorting with respect to the maximin fitness                        */
AssignFitness($\mathcal{P}$);
Sort($\mathcal{P}$);
$\mathcal{ND} \leftarrow$ The nondominated solutions in $\mathcal{P}$;
/*Fill up the new population with the best copies according to the
  maximin fitness, verifying that there is not a similar one        */
$\mathcal{S} \leftarrow \emptyset$;
foreach $y \in \mathcal{P}$ do
    if $|\mathcal{S}| < N$ and VerifySimilarity($y$, $\mathcal{S}$, $\epsilon$) = 0 then
        $\mathcal{S} \leftarrow \mathcal{S} \cup y$;
    end
end
if $|\mathcal{S}| \leq N$ then
    /*Choose the remaining individuals considering only the maximin
      fitness                                                        */
    foreach $y \in \mathcal{P}$ such that $y$ has been not selected do
        $\mathcal{S} \leftarrow \mathcal{S} \cup y$;
    end
else
    /*Improve diversity according to the Euclidean distances between
      solutions.                                                     */
    foreach $y \in \mathcal{ND}$ who has not participated in the selection process do
        if VerifySimilarity($y$, $\mathcal{S}$, $\epsilon$) = 0 then
            $s_{nearest} \leftarrow$ The nearest neighbor of $y$ in $\mathcal{S}$;
            $dy1 \leftarrow$ Distance from $y$ to $s_{nearest}$;
            $s_{random} \leftarrow$ Obtain a random individual from $\mathcal{S}$ such that
            $s_{nearest} \neq s_{random}$;
            $dsrandom \leftarrow$ Distance from $s_{random}$ to its nearest neighbor in $\mathcal{S}$;
            if $dy1 > dsrandom$ then
                Replace $s_{random}$ with $y$;
            else
                $dsnearest \leftarrow$ Distance from $s_{nearest}$ to its nearest neighbor in $\mathcal{S}$;
                $dy2 \leftarrow$ Distance from $y$ to its nearest neighbor in $\mathcal{S}$ without
                regarding $s_{nearest}$;
                if $dy2 > dsnearest$ then
                    Replace $s_{nearest}$ with $y$;
                end
            end
        end
    end
end
return $\mathcal{S}$;
```

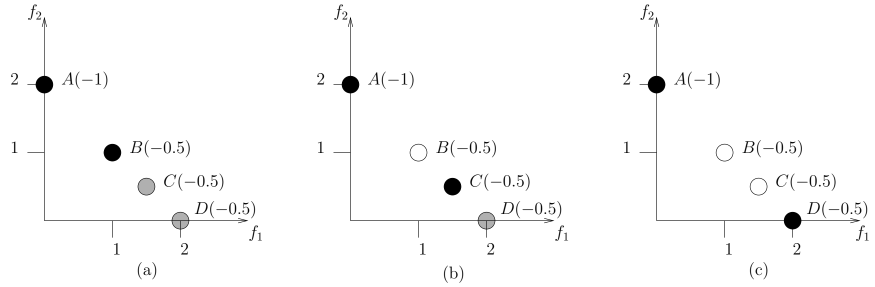


Figure 6.5: Let's assume that we want to select two individuals. If we use the technique based on MFF and Euclidean distances, first we select A and B ($\mathcal{S} = A, B$), see (a). After that, we consider individual C; its nearest neighbor is B and we choose A as a random solution. First, C competes with A and C loses because the distance from A to B is greater than the distance from C to B. Then, C competes with B and C wins because the distance from C to A is greater than the distance from B to A, see (b). Finally, we consider D, and D loses with A but it wins with C. Then, we select A and D, see (c).

6.4 MFF and the Hypervolume indicator

As we mentioned in Chapters 3 and 5, there are different indicators to assess the quality of the approximate Pareto optimal set generated by a MOEA. However, the *hypervolume indicator* (I_H) is the only unary indicator which is strictly "Pareto compliant" [97]. Besides, I_H rewards convergence towards the Pareto front as well as the maximum spread of the solutions obtained. For these reasons, many MOEAs based on it have been proposed [52, 49, 91, 35, 8, 50, 63]. However, this indicator has an important disadvantage: its high computational cost. In the previous chapter, we proposed a new competition scheme for MOEAs based on I_H and on its locality property. Under this scheme, we only calculate three contributions to I_H per iteration (instead of the $P+1$ contributions that the traditional competition scheme used by *SMS-EMOA* requires, where P is the size of the population). Also, we showed that if we approximate the contribution to I_H into this new competition scheme we can obtain better results than if we approximate it using the traditional competition scheme.

In this section, we propose to combine MFF and I_H. Our idea is to use MFF as the main selection mechanism and I_H is used only to correct the possible errors produced when selecting with MFF. This is due to two things: (i) we want to reduce the number of times that the contribution to I_H is calculated and (ii) we want to improve the diversity obtained by MOEAs based on MFF. One interesting thing of this selection mechanism is that, to the best of our knowledge, it is the only one based on I_H that is known to work with a population-based scheme. This is because MFF determines the order in which each individual competes to survive using I_H. This scheme also uses the competition scheme proposed in the previous chapter in which each individual only competes with two other individuals of the population. Therefore, the original combinatorial problem no longer exists.

Our selection mechanism based on MFF and I_H works as follows: If we want to select N individuals from a population $\mathcal{P}$, we assign first a fitness value to each individual using the modified MFF. Then, we proceed to select the individuals according to their fitness values, verifying similarity between selected individuals, see Algorithm 9. If we consider all individuals in the population and we do not select N individuals, we select the remaining individuals considering only the maximin fitness. If we have already selected N individuals but there are still nondominated individuals in $\mathcal{P}$ who have not participated in the selection process, then, we proceed to use the contribution to I_H as follows: Let $\mathcal{S}$ be the set of current selected individuals. Then, for each nondominated individual y who has not participated in the selection process, we obtain its nearest neighbor in $\mathcal{S}$ (we call it $y_{nearest}$) and we choose a random individual called y_{random} such that $y_{nearest} \neq y_{random}$. Finally, we calculate the contribution to I_H of y, $y_{nearest}$ and y_{random}. If y has a better contribution than $y_{nearest}$ or y_{random}, then y replaces the individual with the worst contribution ($y_{nearest}$ or y_{random}). The full selection mechanism is shown in Algorithm 13.

Figure 6.6 shows the selection process using MFF and I_H. Since individuals C and D are not considered, in (a), C competes with A and B, and C replaces B. In (b), D competes with A and C, and it loses. Also, with this selection mechanism, if we return to Figure 6.4, we can avoid having big gaps in the front.

As we mentioned before, calculating I_H or its contribution is a **#P-hard** problem. Therefore, although with the selection mechanism based on MFF and I_H, we can reduce the number of times that we need to calculate the contribution to I_H, if we want to solve MOPs with many objective functions, e.g., more than six, this MOEA is not practical. For this reason, we also propose to use a version of the selection mechanism based on MFF and I_H

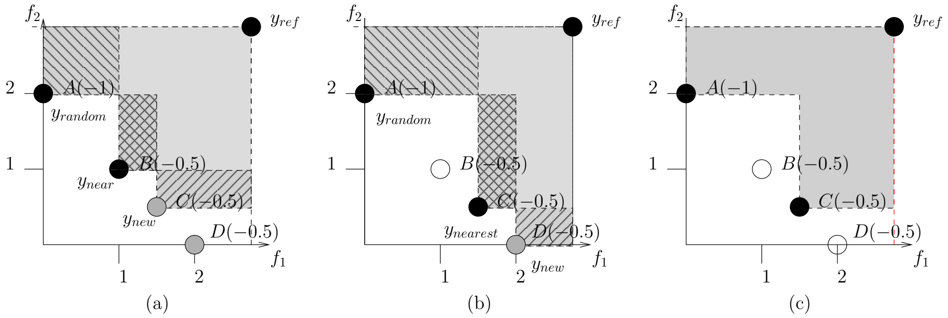


Figure 6.6: Let's assume that we want to select two individuals. If we use the technique based on MFF and I_H, first we select A and B ($\mathcal{S} = A, B$). After that, we consider individual C; its nearest neighbor is B and we choose A as a random solution. Individual B is eliminated because it has the worst contribution, see (b). Finally, we consider D; its nearest neighbor is C and we choose A as a random solution. Individual D is eliminated because it has the worst contribution, see (b). Finally, we choose individuals A and C, see (c).

which approximates the contributions to I_H, using the technique proposed by Bringmann and Friedrich, see Chapter 5.

6.5 Experimental results

Each of the four selection mechanism described in this chapter were incorporated into a MOEA that uses the crossover and mutation operators of *NSGA-II* to create new individuals, giving rise to the four following MOEAs: "*MC-MOEA*: Maximin-Clustering Multi-Objective Evolutionary Algorithm", "*MD-MOEA*: Maximin-Distances Multi-Objective Evolutionary Algorithm", "*MH-MOEA*: Maximin-Hypervolume Multi-Objective Evolutionary Algorithm" and "*MAH-MOEA*: Maximin-Approximated Hypervolume Multi-Objective Evolutionary Algorithm". These MOEAs work as follows: If the size of the population is P, then we create P new individuals. We use a binary tournament to select the parents. At each tournament, two individuals are randomly selected and the one with the higher maximin fitness value is chosen. After that, we combine the population of parents and offspring to obtain a population of size $2P$. Then, we use one of the four selection mechanisms to choose the P individuals that will take part of

Algorithm 13: Maximin-Hypervolume Selection

```
Input  : $\mathcal{P}$ (population), $n$ (number of individuals to choose $N < |\mathcal{P}|$).
Output: $\mathcal{S}$ (selected individuals).
AssignFitness($\mathcal{P}$);
Sort($\mathcal{P}$);
$\mathcal{ND} \leftarrow$ The nondominated solutions in $\mathcal{P}$;
$\mathcal{S} \leftarrow \emptyset$;
/*Fill up the new population with the best copies according to the
  maximin fitness, verifying that there is not a similar one       */
$\mathcal{S} \leftarrow \emptyset$;
foreach $y \in \mathcal{P}$ do
    if $|\mathcal{S}| < N$ and VerifySimilarity($y$, $\mathcal{S}$, $\epsilon$) = 0 then
        $\mathcal{S} \leftarrow \mathcal{S} \cup y$;
    end
end
if $|\mathcal{S}| \leq N$ then
    /*Choose the remaining individuals considering only the maximin
      fitness                                                       */
    foreach $y \in \mathcal{P}$ such that $y$ has been not selected do
        $\mathcal{S} \leftarrow \mathcal{S} \cup y$;
    end
else
    /*Improve the diversity according to the contribution to $I_H$   */
    foreach $y \in \mathcal{ND}$ who had not participated in the selection process do
        if VerifySimilarity($y$, $\mathcal{S}$, $\epsilon$) = 0 then
            $y_{nearest} \leftarrow$ The nearest neighbor of $y$ in $\mathcal{S}$;
            $y_{random} \leftarrow$ A randomly selected individual in $\mathcal{S}$ such that
            $y_{nearest} \neq y_{random}$;
            /*Calculate the contributions to the hypervolume        */
            $C_{nearest} \leftarrow C_H(y_{nearest}, \mathcal{S})$;
            $C_{random} \leftarrow C_H(y_{random}, \mathcal{S})$;
            $C_y \leftarrow C_H(y, \mathcal{S})$;
            /*Remove the individual with the worst contribution     */
            $worst \leftarrow$ Individual with the worst contribution ($y$, $y_{nearest}$ or
            $y_{random}$);
            if $worst = y_{nearest}$ or $worst = y_{random}$ then
                Replace $worst$ with $y$;
            end
        end
    end
end
return $\mathcal{S}$;
```

the following generation. This process is repeated for a certain (pre-defined) number of generations.

For our experiments, we used the following test problems: DTLZ [30] and WFG [48]. We used MOPs with up to ten objective functions. We used $k = 5$ for DTLZ1, DTLZ3 and DTLZ6 and $k = 10$ for the remaining DTLZs. And, we used $k\_factor = 2$ and $l\_factor = 10$ for the WFG test problems. In Appendix A, we describe these two sets of test problems. We adopted the parameters suggested by the authors of *NSGA-II*: $p_c = 0.9$ (crossover probability), $p_m = 1/n$ (mutation probability), where n is the number of decision variables. For the crossover and mutation operators, we adopted $\eta_c = 15$ and $\eta_m = 20$, respectively. Our maximum number of fitness function evaluations was set to 50,000 (we used a population size of 100 individuals and we iterated for 500 generations). In the case of *MAH-MOEA*, we used 10^4 as our number of samples.

6.5.1 Performance indicators

To assess performance, we adopted the following indicators:

- **Hypervolume indicator** (I_H). It is defined as the size of the space covered by the Pareto optimal solutions. I_H rewards both convergence towards the Pareto front as well as the maximum spread of the solutions obtained. A high $I_H(\mathcal{A})$ value, indicates that $\mathcal{A}$ is close to $\mathcal{PF}$ and has a good spread towards the extreme portions of the $\mathcal{PF}$. See Chapter 3, Section 3.2.1. To calculate I_H, we normalized the approximations of the Pareto optimal set, generated by the MOEAs, and we used $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$ as our reference point. The normalization was performed considering all approximations generated by the different MOEAs (i.e., we place, in one set, all nondominated solutions found by the MOEAs which are being compared and from this set we calculate the maximum and minimum for each objective function).

- **Two Set Coverage** (I_{SC}). We decided to use this indicator with the aim of assessing the convergence of the MOEAs. I_{SC} is a binary Pareto compliant indicator. If all points in $\mathcal{A}$ dominate or are equal to all points in $\mathcal{B}$, then by definition $I_{SC}(\mathcal{A}, \mathcal{B}) = 1$. $I_{SC}(\mathcal{A}, \mathcal{B}) = 0$ implies that no element in $\mathcal{B}$ is dominated by any element of $\mathcal{A}$. In general, both $I_{SC}(\mathcal{A},\mathcal{B})$ and $I_{SC}(\mathcal{B},\mathcal{A})$ have to be considered. See Chapter 3, Section 3.2.1.

- **Spacing** (I_S). It measures the spread of solutions in the approximate Pareto optimal front. When $I_S(\mathcal{A}) = 0$ all the solutions in A are uniformly spread. Note that this indicator assumes that a MOEA has already converged to the true Pareto front. See Chapter 3, Section 3.2.1.

6.5.2 Comparison of MOEAs based on MFF

In this section, we compare the four MOEAs based on MFF: *MC-MOEA*, *MD-MOEA*, *MH-MOEA* and *MAH-MOEA*. Table 6.2 shows the results with respect to I_H for the DTLZ test problems with up to six objective functions. In this table, we can see that *MC-MOEA* ranked fourth in all twenty-eight cases; *MD-MOEA* ranked third in twenty-four cases, second in two cases and first in two cases; *MH-MOEA* ranked first in twenty-five cases and only in three cases ranked second; finally, *MAH-MOEA* ranked second in twenty-three cases, third in four cases and first in one case. Tables 6.6 and 6.7 show the results of the statistical analysis that we made to validate our experiments, for which we used Wilcoxon's rank sum. In this case, we decided to compare the fourth place with the third place (*MC-MOEA* and *MD-MOEA*, respectively), the third place with the second place (*MD-MOEA* and *MAH-MOEA*, respectively) and the second place with the first place (*MAH-MOEA* and *MH-MOEA*, respectively). For *MC-MOEA* and *MD-MOEA*, we can see that in twenty-six cases we can reject the null hypothesis (medians are equal) and only for DTLZ6 with four objective functions and DTLZ1 with five objective functions we can say that these two algorithms have a similar behavior. For *MD-MOEA* and *MAH-MOEA*, we can see that in twenty-five cases we can reject the null hypothesis and only for three problems both algorithms have a similar behavior. Finally, for *MAH-MOEA* and *MH-MOEA*, we have that for sixteen cases we can reject the null hypothesis and for twelve cases we can say that both algorithms have a similar behavior. This result is interesting because we can say that *MAH-MOEA* is really competitive with respect to *MH-MOEA*.

Since *MD-MOEA* outperformed *MC-MOEA* in all cases, we can say that the technique based on Euclidean distances was able to correct some disadvantages of the technique based on clustering, e.g., it can avoid that the approximate Pareto front has big gaps. However, both MOEAs (*MC-MOEA* and *MD-MOEA*) have difficulties when the MOP has a degenerate Pareto front (see problems DTLZ5 and DTLZ6). We think that this problem arises because these two selection mechanisms have as their aim to distribute the solutions uniformly and then, it is hard for the MOEA to converge to a front with a lower dimensionality than the dimensionality of the problem. However, we can see that *MH-MOEA* and *MAH-MOEA* were able to correct this disadvantage. This is because the aim of these selection mechanisms is to maximize I_H and the maximum I_H corresponds to a distribution into the degenerate Pareto front.

An interesting thing is that *MAH-MOEA*, the version of *MH-MOEA* that approximates the contribution to I_H, obtained results very close to *MH-MOEA* but at a lower computational cost (see Tables 6.2, 6.3 and 6.6). This is an important result because, as we know, MOEAs based on the use of the exact I_H values are not practical when we want to solve MOPs with more than five or six objective functions. In order to address this disadvantage, some authors have proposed different techniques to approximate I_H or its contribution. However, the quality of the solutions obtained by these MOEAs considerably degrades in most cases, unlike *MAH-MOEA* which does not lose much quality due to two reasons: First, it approximates the contribution to I_H in the competition scheme based on I_H and on its locality property as we suggested in the previous chapter. And second, it produces a ranking using MFF to perform an initial selection and then, it uses the contribution to I_H only to correct the possible errors in this first selection procedure, i.e., I_H is not used as the primary selection mechanism.

From Tables 6.2 and 6.3, we can say that the best option to solve MOPs with low and high dimensionality (in objective function space) is *MAH-MOEA*. However, if we need to obtain the approximate Pareto optimal set in the shortest time possible, *MD-MOEA* is a good option but we should be careful when dealing with MOPs having degenerate Pareto fronts.

Table 6.4 shows the results with respect to I_H for the WFG test problems with up to six objective functions and we can see that *MC-MOEA* ranked fourth in twenty-three cases, third in three cases and first in two cases. *MD-MOEA* ranked third in eighteen cases, fourth in four cases, second in one case and first in five cases. *MH-MOEA* ranked first in sixteen cases, second in ten cases and third in two cases. Finally, *MAH-MOEA* ranked second in seventeen cases, first in five cases, third in five cases and fourth in one case. In the same way as with the DTLZ test problems, we conducted a statistical analysis using Wilcoxon's rank sum to validate our experiments, see Table 6.7. For *MC-MOEA* and *MD-MOEA*, we can see that in four problems both algorithms have a similar behavior and in twenty-four cases we can reject the null hypothesis. For *MD-MOEA* and *MAH-MOEA*, we can reject the null hypothesis in twenty-one cases and only in seven cases both algorithms have a similar behavior. Finally, for *MAH-MOEA* and *MH-MOEA*, we can say that for eighteen problems both algorithms have a similar behavior and only in ten cases we can reject the null hypothesis. With these results, we can corroborate that *MAH-MOEA* is really competitive with *MH-MOEA*. Then, these two algorithms are the best, followed by *MD-MOEA* in the third place and *MC-MOEA* in the fourth place. However, if we consider the running time, *MH-MOEA* is the worst algorithm followed by *MAH-MOEA* in the third place, *MD-MOEA* in the second place and *MC-MOEA* is the best algorithm (see Table 6.5).

Finally, we can observe that *MC-MOEA* and *MD-MOEA* have difficulties in problems WFG1, WFG4 and WFG7. However, we cannot identify particular features of these problems which indicate the possible disadvantages of these MOEAs in solving them.

It is important to note that if we use I_H to compare the different MOEAs, then it is evident that MOEAs based on this indicator have advantages over those which don't adopt it, because the aim of the former type of MOEAs is to maximize I_H. For this reason, we decided to use two other indicators to compare the approximate Pareto optimal sets obtained by the MOEAs. We adopted the two set coverage indicator (I_{SC}) to measure convergence to the Pareto front and the spacing indicator (I_S) to measure distribution of the solutions found. Since we can use these two indicators to evaluate approximations which involve any number of objective functions, we decided to use up to ten objective functions. However, in this comparison we only considered *MD-MOEA* and *MAH-MOEA* due to two reasons: First, Tables 6.2 and 6.4 show clearly that *MD-MOEA* obtained better results than *MC-MOEA*. And second, although *MH-MOEA* is better than *MAH-MOEA*, it cannot be used to solve MOPs with more than six objective functions (its running time is too high, and it would require weeks or even months to complete all the required experiments).

Table 6.8 shows the results for the DTLZ test problems with respect to I_{SC} and we can see that in fifty-four cases the solutions found by *MAH-MOEA* were able to cover a larger percentage of the solutions found by *MD-MOEA* than the percentage of solutions found by *MAH-MOEA* which are covered by at least one solution found by *MD-MOEA*. However, only in the DTLZ6 test problem we can assure that *MAH-MOEA* is better than *MD-MOEA* because only in this problem the percentage of solutions found by *MAH-MOEA* which are covered by at least one solution found by *MD-MOEA* is zero or close to zero and the percentage of solutions found by *MD-MOEA* which are covered by at least one solution found by *MAH-MOEA* is close to one. Table 6.9 shows the results regarding I_S and we can observe that *MD-MOEA* ranked second in thirty-nine cases and first in seventeen cases. With these two tables, we can corroborate the results found when we use I_H: *MAH-MOEA* is better than *MD-MOEA* in most cases. However, we can say that the technique based on Euclidean distances to improve the distribution of the solutions is effective because regarding I_S, *MD-MOEA* outperformed *MAH-MOEA* in seventeen cases.

Finally, Tables 6.10 and 6.11 show the results with respect to the WFG test problems. According to I_{SC}, *MD-MOEA* was ranked in second place in twenty-four cases and in first place in twenty cases. However, there are no cases in which we can assure that one MOEA is better than the others. Regarding I_S, we can see that *MD-MOEA* was ranked in first place in forty-nine cases and it was ranked in second place in seven cases. Therefore, for these test problems *MAH-MOEA* and *MD-MOEA* are competitive because from I_{SC} we cannot assure that *MAH-MOEA* outperformed *MD-MOEA* in terms of convergence and from I_S we observe that *MD-MOEA* is better in most problems. Also, in these problems we can see in a clearer way that the technique based on Euclidean distances is effective.

As final conclusions of this section, we can say that *MD-MOEA* and *MAH-MOEA* are the best options to solve MOPs with high and low dimensionality in objective function space. Although *MAH-MOEA* is better than *MD-MOEA* according to I_H, regarding I_{SC} and I_S they are competitive. Also, *MD-MOEA* is much faster than *MAH-MOEA*. However, it is important to be careful when we use *MD-MOEA* because it has difficulties to solve certain types of MOPs, e.g., MOPs with a degenerate Pareto front.

6.5.3 MOEAs based on MFF vs MOEAs not based on MFF

In this section, we compare *MD-MOEA* and *MAH-MOEA* with respect to two well-known MOEAs: The first one is *MOEA/D* (See Chapter 4, Section 4.3). We chose this MOEA because it has been a viable alternative to deal with many-objective optimization problems in recent years. Also, its computational cost is very low. *MOEA/D* [89] decomposes the MOP into N scalar optimization subproblems and then it solves these subproblems simultaneously using an evolutionary algorithm. For our experiments, we used the version in which *MOEA/D* adopts PBI (Penalty Boundary Intersection) to decompose the MOP. We decided to use PBI because the resulting optimal solutions with PBI are normally much better distributed than those obtained by the Tchebycheff approach [89]. To generate the convex weights we used the technique proposed in [27] and after that, we applied clustering (k-means) to obtain a specific number of weights.

The second one is *SMS-EMOA* [35]. We chose this MOEA because it is the most popular hypervolume-based MOEA. *SMS-EMOA* creates an initial population and then, it generates only one solution by iteration. After that, it applies Pareto ranking. When the last front has more than one solution, *SMS-EMOA* calculates the contribution to I_H of each individual in the last front and it eliminates the individual with the worst contribution. Beume et al. [8] proposed not to use the contribution to I_H when in the Pareto ranking process we obtain more than one front. In that case, they proposed to use the number of solutions which dominate to one solution (the solution that is dominated by more solutions is removed). In this work, we used the version proposed by Beume et al. but instead of calculating the exact contribution to I_H, we approximate it using the same technique that we adopted for *MAH-MOEA*.

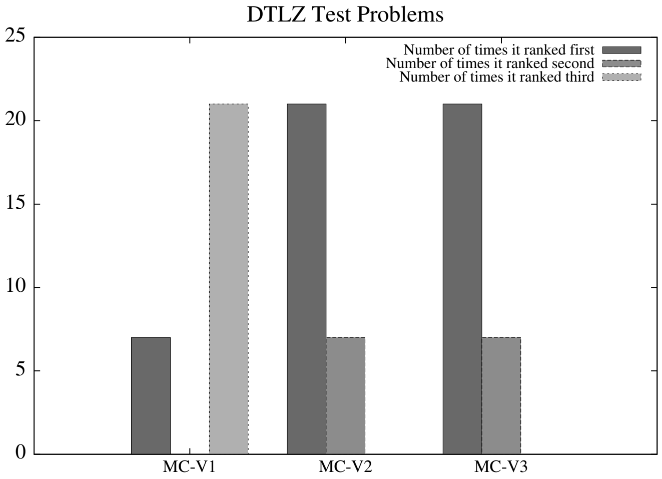
Since these four MOEAs use the same operators to create new individuals (they use the same crossover and mutation operators adopted by *NSGA-II*), the comparison of selection mechanisms is fair. For *MOEA/D* and *SMS-EMOA*, we also adopted the parameters suggested by the authors of *NSGA-II*: $p_c = 0.9$, $p_m = 1/n$, where n is the number of decision variables, $\eta_c = 15$ and $\eta_m = 20$. In the case of *MOEA/D*, we used a neighborhood with size equal to 20 and in the case of *SMS-EMOA* we used 10^4 as our number of samples.

Before we perform the comparison, it is important to mention that both *MOEA/D* and *SMS-EMOA* have important disadvantages. *SMS-EMOA* is impractical to solve MOPs with many objective function because calculating I_H or its contribution involves a very high computational cost. In this work, we use a version that approximates the contribution to I_H. However, as we will see later on, the competition scheme used by *SMS-EMOA* is not efficient and therefore, the running time of this version of *SMS-EMOA* is also high. On the other hand, *MOEA/D* needs to generate a set of well-distributed convex weights and this task becomes more difficult as we increase the number of objective functions.

Regarding I_H and considering the DTLZ test problems, *MD-MOEA* ranked second in eleven cases, third in nine cases, fourth in five cases and first in three cases. *MAH-MOEA* ranked first in fourteen cases, second in ten cases, third in three cases and fourth in one case. *MOEA/D* ranked fourth in nineteen cases, third in five cases, second in three cases and first in one case. Finally, *SMS-EMOA* ranked third in eleven cases, first in ten cases, second in four cases and fourth in three cases. See Table 6.12. Table 6.16 shows the results of the statistical analysis that we conducted to validate our experiments, for which we used Wilcoxon's rank sum and I_H. In this case, we decided to compare the MOEAs based on MFF (*MD-MOEA* and *MAH-MOEA*) with respect to *MOEA/D* and *SMS-EMOA*. For *MD-MOEA* and *MOEA/D*, we can say that only in one problem they have a similar behavior and in the twenty-seven remaining problems the null hypothesis ("medians are equal") can be rejected. The same occurs with *MAH-MOEA* and *MOEA/D*, since in only one problem they have a similar behavior. In the case of *MD-MOEA* and *SMS-EMOA* only in two problems both algorithms have a similar behavior and in the twenty-six remaining problems the null hypothesis can be rejected. Finally, with respect to *MAH-MOEA* and *SMS-EMOA* only in two cases they have a similar behavior and in the twenty-six remaining problems the null hypothesis can be rejected. From these results, we can say that *MAH-MOEA* is the best algorithm, followed by *SMS-EMOA* in the second place, *MD-MOEA* in the third place and *MOEA/D* in the fourth place. Another interesting thing is that *MAH-MOEA* is much faster than *SMS-EMOA*. It is also worth noticing that *MD-MOEA* is ranked second with respect to the running time but it is not much slower than *MOEA/D* which is in the first place. See Table 6.13.

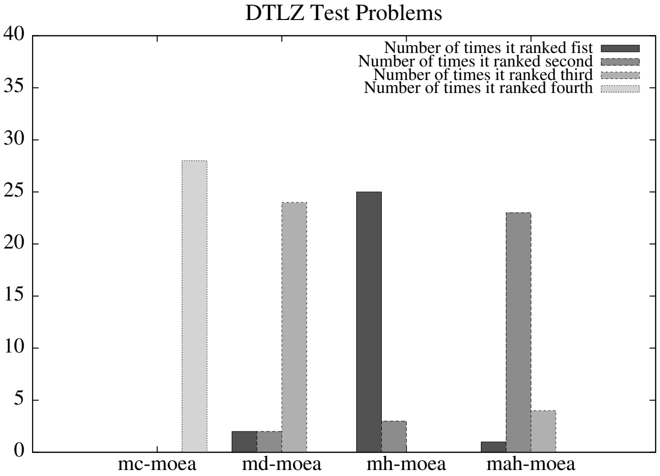
With respect to the WFG test problems, we can see in Table 6.14 that *MD-MOEA* ranked third in seventeen cases, first in 6 cases, second in three cases and fourth in two cases. *MAH-MOEA* ranked second in twenty-five cases and first in three cases. *MOEA/D* ranked fourth in twenty-three cases and third in five cases. Finally, *SMS-EMOA* ranked first in nineteen cases, third in six cases and fourth in three cases. We applied a statistical analysis to validate our experiments in the same way as done with the DTLZ test problems, see Table 6.17. For *MD-MOEA* and *MOEA/D* in all twenty-eight problems, the null hypothesis can be rejected. And the same occurs for *MAH-MOEA* and *MOEA/D*. For *MD-MOEA* and *SMS-EMOA*, in two problems these two algorithms have a similar behavior and in the twenty-six remaining problems we can reject the null hypothesis. And, for *MAH-MOEA* and *SMS-EMOA*, we can see that in two cases both algorithms have a similar behavior and for all twenty-six problems the null hypothesis can be rejected. Then, for these problems *SMS-EMOA* is the best algorithm, followed by *MAH-MOEA* in the second place, *MD-MOEA* in the third place and *MOEA/D* in the fourth place. However, also in these problems *MAH-MOEA* is much faster than *SMS-EMOA* and *MD-MOEA* is not much slower than *MOEA/D*.

As a conclusion of this section, we can say that MOEAs based on MFF are a good option to solve MOPs with low and high dimensionality because they can outperform well-known MOEAs such as *SMS-EMOA* and *MOEA/D*, e.g., both *MD-MOEA* and *MAH-MOEA* outperformed *MOEA/D* in the two sets of test problems adopted (DTLZ and WFG) and *MAH-MOEA* outperformed *SMS-EMOA* in the DTLZ test problems. In addition, both *MD-MOEA*, *MAH-MOEA* are much faster than *SMS-EMOA* and *MD-MOEA* is not much slower than *MOEA/D*.



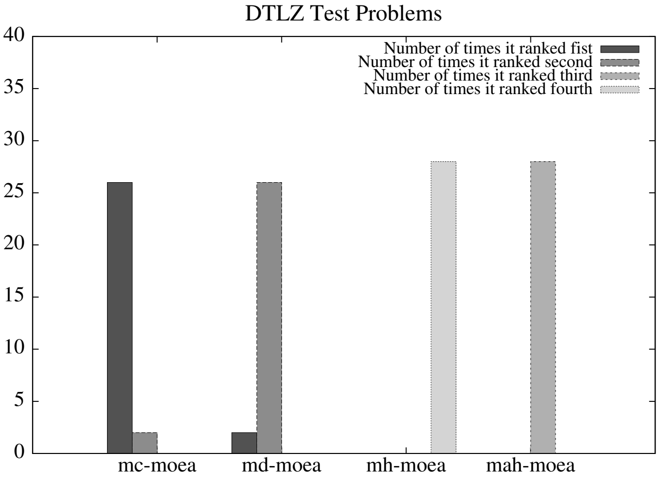
$\vec{f}$	mc-v1 I_H	mc-v2 I_H	mc-v3 I_H
DTLZ1 (3)	1.2560 (0.089)	**1.2714 (0.080)**	1.2714 (0.080)
DTLZ2 (3)	0.7209 (0.014)	**0.7291 (0.011)**	0.7291 (0.011)
DTLZ3 (3)	**0.9044 (0.323)**	0.8759 (0.366)	0.8759 (0.366)
DTLZ4 (3)	0.7162 (0.014)	**0.7319 (0.010)**	0.7319 (0.010)
DTLZ5 (3)	0.2860 (0.019)	**0.2991 (0.007)**	0.2991 (0.007)
DTLZ6 (3)	0.3992 (0.064)	**0.4589 (0.035)**	0.4589 (0.035)
DTLZ7 (3)	0.7402 (0.007)	**0.7430 (0.007)**	0.7430 (0.007)
DTLZ1 (4)	1.2676 (0.364)	**1.3576 (0.254)**	1.3576 (0.254)
DTLZ2 (4)	**1.0024 (0.022)**	0.9978 (0.019)	0.9978 (0.019)
DTLZ3 (4)	1.4621 (0.002)	**1.4623 (0.002)**	1.4623 (0.002)
DTLZ4 (4)	1.0556 (0.019)	**1.0668 (0.015)**	1.0668 (0.015)
DTLZ5 (4)	0.6936 (0.099)	**0.7784 (0.107)**	0.7784 (0.107)
DTLZ6 (4)	1.0759 (0.058)	**1.1089 (0.038)**	1.1089 (0.038)
DTLZ7 (4)	0.7034 (0.014)	**0.7072 (0.013)**	0.7072 (0.013)
DTLZ1 (5)	**1.5935 (0.025)**	1.5860 (0.036)	1.5860 (0.036)
DTLZ2 (5)	1.2681 (0.030)	**1.2750 (0.033)**	1.2750 (0.033)
DTLZ3 (5)	**1.6085 (0.002)**	1.6083 (0.003)	1.6083 (0.003)
DTLZ4 (5)	1.3698 (0.026)	**1.3827 (0.024)**	1.3827 (0.024)
DTLZ5 (5)	**0.8852 (0.094)**	0.8847 (0.084)	0.8847 (0.084)
DTLZ6 (5)	**0.9683 (0.091)**	0.9506 (0.098)	0.9506 (0.098)
DTLZ7 (5)	0.7861 (0.028)	**0.7941 (0.026)**	0.7941 (0.026)
DTLZ1 (6)	1.7482 (0.025)	**1.7574 (0.022)**	1.7574 (0.022)
DTLZ2 (6)	1.5556 (0.050)	**1.5688 (0.037)**	1.5688 (0.037)
DTLZ3 (6)	1.7685 (0.003)	**1.7690 (0.003)**	1.7690 (0.003)
DTLZ4 (6)	1.6894 (0.042)	**1.6943 (0.040)**	1.6943 (0.040)
DTLZ5 (6)	0.9326 (0.095)	**0.9328 (0.077)**	0.9328 (0.077)
DTLZ6 (6)	0.9559 (0.035)	**0.9700 (0.050)**	0.9700 (0.050)
DTLZ7 (6)	**0.7230 (0.053)**	0.7116 (0.059)	0.7116 (0.059)

Table 6.1: Results obtained in the DTLZ test problems with up to six objective functions. We compare *MC-V1*, *MC-V2* and *MC-V3* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



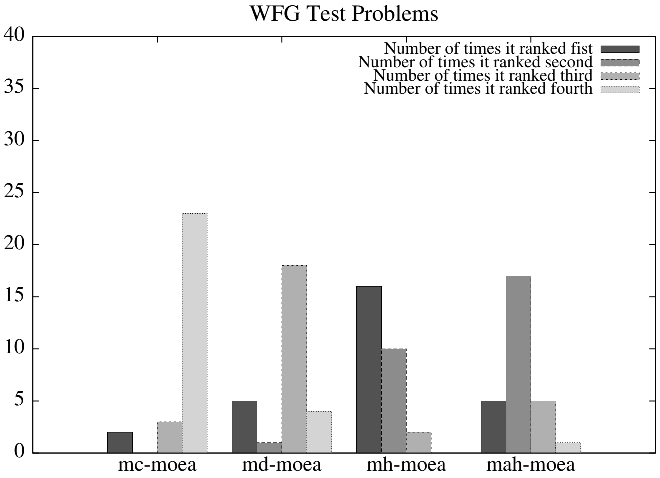
$\vec{f}$	mc-moea	md-moea	mh-moea	mah-moea
	I_H	I_H	I_H	I_H
DTLZ1 (3)	1.1593 (0.013)	1.1823 (0.003)	**1.1897 (0.001)**	1.1896 (0.001)
DTLZ2 (3)	0.6964 (0.008)	0.7319 (0.003)	**0.7535 (0.002)**	0.7525 (0.002)
DTLZ3 (3)	1.3043 (0.021)	**1.3281 (0.000)**	1.3265 (0.000)	1.3263 (0.000)
DTLZ4 (3)	0.7015 (0.008)	0.7304 (0.006)	0.7491 (0.004)	**0.7492 (0.004)**
DTLZ5 (3)	0.2552 (0.005)	0.2651 (0.000)	**0.2661 (0.001)**	0.2661 (0.000)
DTLZ6 (3)	1.0453 (0.067)	1.0989 (0.008)	**1.0998 (0.007)**	1.0975 (0.009)
DTLZ7 (3)	0.5377 (0.040)	0.5519 (0.041)	**0.5645 (0.043)**	0.5638 (0.043)
DTLZ1 (4)	1.4241 (0.016)	1.4375 (0.005)	**1.4471 (0.002)**	1.4444 (0.002)
DTLZ2 (4)	1.0480 (0.019)	1.1249 (0.010)	**1.1672 (0.003)**	1.1647 (0.005)
DTLZ3 (4)	1.4630 (0.001)	**1.4638 (0.000)**	1.4638 (0.000)	1.4638 (0.000)
DTLZ4 (4)	1.1455 (0.016)	1.1912 (0.010)	**1.2261 (0.006)**	1.2252 (0.006)
DTLZ5 (4)	0.3928 (0.017)	0.4668 (0.023)	**0.5492 (0.005)**	0.5469 (0.004)
DTLZ6 (4)	1.2518 (0.045)	1.2742 (0.024)	**1.3617 (0.003)**	1.3600 (0.003)
DTLZ7 (4)	0.6659 (0.014)	0.7062 (0.016)	**0.7305 (0.019)**	0.7262 (0.018)
DTLZ1 (5)	1.5861 (0.069)	1.6011 (0.003)	**1.6061 (0.001)**	1.6031 (0.001)
DTLZ2 (5)	1.3004 (0.025)	1.4063 (0.015)	**1.4680 (0.004)**	1.4519 (0.006)
DTLZ3 (5)	1.6089 (0.002)	1.6098 (0.000)	**1.6098 (0.000)**	1.6098 (0.000)
DTLZ4 (5)	1.4341 (0.018)	1.4801 (0.011)	**1.5182 (0.006)**	1.5096 (0.007)
DTLZ5 (5)	0.7294 (0.019)	0.8122 (0.033)	**0.9249 (0.007)**	0.9180 (0.015)
DTLZ6 (5)	0.4890 (0.074)	1.3396 (0.063)	**1.5522 (0.002)**	1.5450 (0.005)
DTLZ7 (5)	0.7562 (0.026)	0.8534 (0.009)	**0.8716 (0.015)**	0.8699 (0.010)
DTLZ1 (6)	1.7002 (0.088)	1.7652 (0.004)	**1.7688 (0.001)**	1.7669 (0.001)
DTLZ2 (6)	1.6061 (0.027)	1.6757 (0.014)	**1.7264 (0.004)**	1.7114 (0.006)
DTLZ3 (6)	1.7707 (0.001)	1.7710 (0.000)	**1.7713 (0.000)**	1.7712 (0.000)
DTLZ4 (6)	1.6978 (0.024)	1.7234 (0.008)	**1.7495 (0.002)**	1.7407 (0.003)
DTLZ5 (6)	0.8283 (0.030)	0.9563 (0.067)	**1.1347 (0.019)**	1.1170 (0.041)
DTLZ6 (6)	0.5754 (0.050)	1.0454 (0.174)	**1.7039 (0.004)**	1.6984 (0.004)
DTLZ7 (6)	0.7382 (0.054)	0.9319 (0.022)	**0.9438 (0.020)**	0.9380 (0.026)

Table 6.2: Results obtained in the DTLZ test problems with up to six objective functions. We compare *MC-MOEA*, *MD-MOEA*, *MH-MOEA* and *MAH-MOEA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



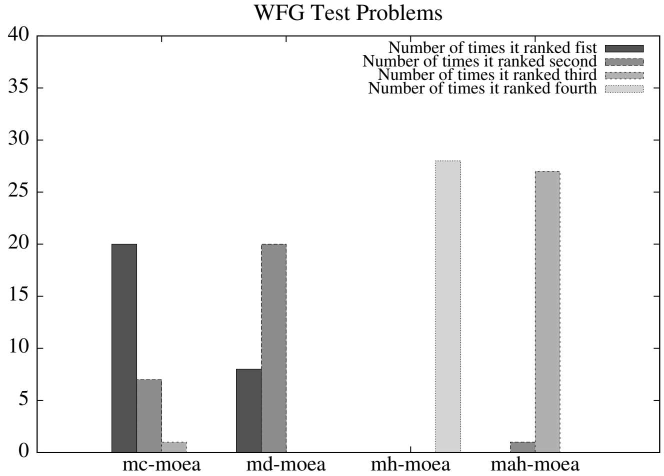
$\vec{f}$	mc-moea time	md-moea time	mh-moea time	mah-moea time
DTLZ1 (3)	**0.6277 (0.010)**	0.8317 (0.025)	341.8550 (48.468)	26.5683 (6.888)
DTLZ2 (3)	**0.9857 (0.011)**	1.7247 (0.016)	362.5860 (46.285)	80.2447 (11.191)
DTLZ3 (3)	**0.6150 (0.016)**	0.7897 (0.028)	410.8940 (51.899)	26.4793 (5.165)
DTLZ4 (3)	**1.0137 (0.009)**	1.7187 (0.018)	441.8840 (25.976)	77.7587 (9.585)
DTLZ5 (3)	**0.8377 (0.009)**	1.2467 (0.019)	362.4503 (36.636)	9.6463 (2.308)
DTLZ6 (3)	**0.7157 (0.032)**	1.0763 (0.020)	358.8433 (17.170)	26.3760 (12.040)
DTLZ7 (3)	**0.9210 (0.018)**	1.0917 (0.036)	370.5477 (18.106)	28.6033 (7.032)
DTLZ1 (4)	**0.8150 (0.013)**	1.0227 (0.015)	350.6833 (13.917)	17.1963 (2.158)
DTLZ2 (4)	**1.2033 (0.015)**	2.0807 (0.009)	380.9720 (12.960)	56.9457 (9.412)
DTLZ3 (4)	**0.8083 (0.013)**	1.0830 (0.029)	403.1007 (26.826)	23.3403 (2.858)
DTLZ4 (4)	**1.2520 (0.011)**	2.0533 (0.020)	424.2623 (37.408)	52.1267 (9.269)
DTLZ5 (4)	**1.2420 (0.017)**	2.3523 (0.028)	418.8673 (56.864)	50.0157 (6.454)
DTLZ6 (4)	**0.9953 (0.011)**	1.6473 (0.020)	386.8503 (13.442)	49.7007 (6.381)
DTLZ7 (4)	**1.1347 (0.019)**	1.1903 (0.026)	354.2237 (11.950)	16.5777 (2.984)
DTLZ1 (5)	**0.9823 (0.015)**	1.1693 (0.037)	495.2057 (40.617)	12.5180 (1.662)
DTLZ2 (5)	**1.3807 (0.017)**	2.3043 (0.014)	893.2087 (23.893)	29.8747 (3.466)
DTLZ3 (5)	**0.9920 (0.009)**	1.2967 (0.028)	558.4480 (23.573)	17.2507 (2.455)
DTLZ4 (5)	**1.4673 (0.022)**	2.2767 (0.021)	843.5720 (16.920)	28.0337 (3.063)
DTLZ5 (5)	**1.4577 (0.011)**	2.4603 (0.021)	1516.0527 (91.143)	44.2683 (5.359)
DTLZ6 (5)	**1.3483 (0.016)**	2.2447 (0.044)	980.2290 (39.726)	46.6733 (5.942)
DTLZ7 (5)	1.3380 (0.016)	**1.3070 (0.035)**	324.5770 (35.547)	7.3807 (1.137)
DTLZ1 (6)	**1.1350 (0.017)**	1.3033 (0.070)	3137.1263 (406.976)	10.5600 (1.946)
DTLZ2 (6)	**1.5407 (0.046)**	2.4600 (0.023)	7760.6740 (199.641)	19.9033 (1.739)
DTLZ3 (6)	**1.1487 (0.010)**	1.4473 (0.030)	3859.2973 (340.664)	15.7413 (2.575)
DTLZ4 (6)	**1.6273 (0.014)**	2.4343 (0.026)	6662.2467 (190.364)	18.3560 (2.040)
DTLZ5 (6)	**1.6177 (0.025)**	2.6093 (0.028)	7964.5500 (1610.847)	37.3323 (5.122)
DTLZ6 (6)	**1.5197 (0.018)**	2.8480 (0.049)	5459.3457 (126.423)	44.0187 (5.048)
DTLZ7 (6)	1.5173 (0.017)	**1.4347 (0.047)**	618.1323 (159.834)	3.7347 (0.632)

Table 6.3: Results obtained in the DTLZ test problems with up to six objective functions. We compare *MC-MOEA*, *MD-MOEA*, *MH-MOEA* and *MAH-MOEA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	mc-moea I_H	md-moea I_H	mh-moea I_H	mah-moea I_H
WFG1 (3)	1.0315 (0.061)	1.0043 (0.057)	**1.1414 (0.046)**	1.1274 (0.049)
WFG2 (3)	0.6980 (0.091)	**0.7559 (0.091)**	0.7559 (0.091)	0.7559 (0.091)
WFG3 (3)	0.5476 (0.011)	0.6156 (0.004)	**0.6320 (0.003)**	0.6312 (0.002)
WFG4 (3)	0.5549 (0.015)	0.6584 (0.008)	**0.7430 (0.002)**	0.7414 (0.002)
WFG5 (3)	0.4479 (0.012)	0.5325 (0.005)	**0.5531 (0.001)**	0.5529 (0.001)
WFG6 (3)	0.5081 (0.008)	0.5483 (0.003)	**0.5615 (0.003)**	0.5602 (0.004)
WFG7 (3)	0.5324 (0.012)	0.6099 (0.013)	**0.7354 (0.004)**	0.7342 (0.005)
WFG1 (4)	1.1074 (0.045)	1.0396 (0.027)	1.2004 (0.049)	**1.2045 (0.045)**
WFG2 (4)	**0.6485 (0.132)**	0.6338 (0.134)	0.6338 (0.134)	0.6338 (0.134)
WFG3 (4)	0.1807 (0.057)	0.5674 (0.009)	**0.5865 (0.009)**	0.5801 (0.009)
WFG4 (4)	0.4842 (0.045)	0.8193 (0.017)	**1.0055 (0.005)**	0.9955 (0.007)
WFG5 (4)	0.3934 (0.019)	0.5501 (0.008)	**0.5901 (0.002)**	0.5870 (0.003)
WFG6 (4)	0.3125 (0.055)	0.5510 (0.016)	**0.5714 (0.012)**	0.5712 (0.010)
WFG7 (4)	0.5361 (0.018)	0.6163 (0.018)	**0.8721 (0.008)**	0.8623 (0.008)
WFG1 (5)	1.1905 (0.036)	1.0896 (0.030)	**1.2266 (0.031)**	1.2095 (0.031)
WFG2 (5)	0.6027 (0.183)	**0.6349 (0.203)**	0.6349 (0.203)	0.6349 (0.203)
WFG3 (5)	0.0000 (0.000)	**0.5753 (0.018)**	0.5724 (0.022)	0.5742 (0.018)
WFG4 (5)	0.5167 (0.044)	0.9373 (0.022)	**1.2261 (0.011)**	1.2076 (0.009)
WFG5 (5)	0.3200 (0.028)	0.5744 (0.014)	0.6136 (0.012)	**0.6142 (0.008)**
WFG6 (5)	0.0020 (0.007)	0.5557 (0.023)	0.5697 (0.031)	**0.5735 (0.025)**
WFG7 (5)	0.5125 (0.018)	0.6113 (0.023)	**0.9623 (0.019)**	0.9482 (0.019)
WFG1 (6)	**1.3749 (0.023)**	1.2711 (0.036)	1.3562 (0.029)	1.3288 (0.031)
WFG2 (6)	0.6486 (0.198)	**0.7530 (0.164)**	0.7530 (0.164)	0.7530 (0.164)
WFG3 (6)	0.0000 (0.000)	**0.5635 (0.027)**	0.5630 (0.037)	0.5575 (0.032)
WFG4 (6)	0.6202 (0.053)	1.0276 (0.032)	**1.4082 (0.012)**	1.4025 (0.015)
WFG5 (6)	0.1658 (0.016)	0.5893 (0.023)	0.6005 (0.022)	**0.6099 (0.024)**
WFG6 (6)	0.0016 (0.006)	0.6030 (0.034)	**0.6140 (0.031)**	0.6134 (0.032)
WFG7 (6)	0.4872 (0.020)	0.6100 (0.023)	0.8517 (0.092)	**0.9036 (0.071)**

Table 6.4: Results obtained in the WFG test problems with up to six objective functions. We compare *MC-MOEA*, *MD-MOEA*, *MH-MOEA* and *MAH-MOEA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	mc-moea	md-moea	mh-moea	mah-moea
	time	time	time	time
WFG1 (3)	**2.0607 (0.016)**	3.0037 (0.036)	407.7103 (16.770)	97.5580 (11.196)
WFG2 (3)	1.4210 (0.197)	**1.3063 (0.081)**	384.3690 (16.681)	1.5803 (0.314)
WFG3 (3)	**1.7350 (0.025)**	2.7167 (0.036)	412.2397 (17.685)	87.2683 (15.644)
WFG4 (3)	**1.7633 (0.010)**	2.9810 (0.040)	404.2377 (16.539)	78.2380 (14.106)
WFG5 (3)	**1.7807 (0.014)**	3.1817 (0.119)	402.1483 (16.008)	122.2553 (17.817)
WFG6 (3)	**1.6670 (0.013)**	2.6367 (0.115)	406.3600 (18.357)	95.7627 (14.819)
WFG7 (3)	**2.1950 (0.011)**	3.7967 (0.053)	404.4447 (17.979)	138.2930 (26.767)
WFG1 (4)	**2.2793 (0.016)**	3.1263 (0.028)	446.1657 (25.163)	55.2243 (6.841)
WFG2 (4)	1.5113 (0.188)	**1.3700 (0.040)**	361.7037 (13.118)	1.7497 (0.150)
WFG3 (4)	**1.8777 (0.026)**	2.3457 (0.082)	387.6740 (22.302)	37.7127 (6.090)
WFG4 (4)	**2.0153 (0.021)**	3.5377 (0.070)	452.5403 (25.090)	66.3593 (11.242)
WFG5 (4)	**1.9810 (0.010)**	3.3533 (0.029)	391.5327 (19.290)	75.2213 (11.970)
WFG6 (4)	**1.8187 (0.009)**	2.2033 (0.062)	380.4550 (20.296)	39.5240 (5.911)
WFG7 (4)	**2.5327 (0.016)**	4.1817 (0.043)	455.2317 (27.525)	85.7517 (9.854)
WFG1 (5)	**2.4997 (0.018)**	3.1180 (0.048)	1514.0497 (79.687)	25.0433 (3.385)
WFG2 (5)	1.7283 (0.237)	**1.5483 (0.055)**	269.0003 (38.875)	2.0567 (0.346)
WFG3 (5)	2.0820 (0.019)	**1.8530 (0.023)**	266.8000 (28.414)	2.7177 (0.374)
WFG4 (5)	**2.2467 (0.011)**	3.8390 (0.030)	1651.3397 (74.059)	52.2173 (8.366)
WFG5 (5)	**2.1750 (0.011)**	3.1130 (0.119)	336.6403 (25.775)	45.3483 (6.833)
WFG6 (5)	2.0200 (0.012)	**1.8200 (0.014)**	283.3267 (61.769)	3.6643 (0.604)
WFG7 (5)	**2.9070 (0.013)**	4.5740 (0.035)	1779.0997 (149.456)	49.3623 (6.280)
WFG1 (6)	**2.6130 (0.015)**	3.0127 (0.039)	13890.5980 (909.094)	13.4993 (1.680)
WFG2 (6)	1.9440 (0.201)	**1.6703 (0.060)**	231.7187 (18.578)	1.9280 (0.205)
WFG3 (6)	2.2267 (0.017)	**1.9417 (0.018)**	233.8433 (16.113)	2.9970 (0.399)
WFG4 (6)	**2.4050 (0.019)**	4.0547 (0.035)	14471.2413 (18.960)	47.5760 (7.722)
WFG5 (6)	**2.2907 (0.029)**	2.5957 (0.090)	331.3267 (23.342)	18.0147 (3.537)
WFG6 (6)	2.1507 (0.011)	**1.9097 (0.022)**	234.4510 (12.421)	2.7093 (0.111)
WFG7 (6)	**3.0050 (0.012)**	4.7223 (0.048)	14228.5130 (998.589)	44.0367 (6.551)

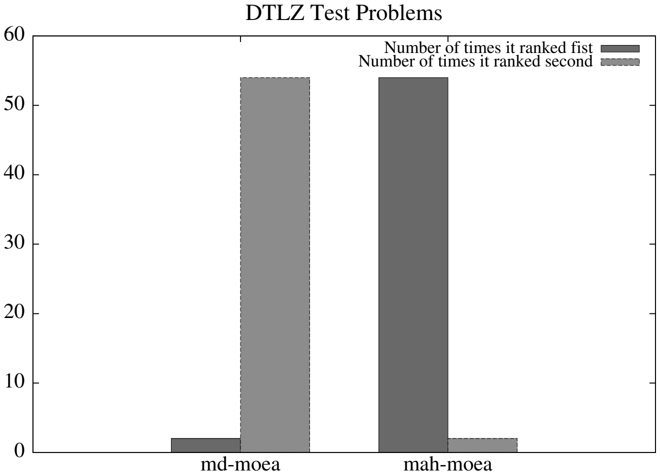
Table 6.5: Results obtained in the WFG test problems with up to six objective functions. We compare *MC-MOEA*, *MD-MOEA*, *MH-MOEA* and *MAH-MOEA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.

$\vec{f}$	mc-moea & md-moea $P(H)$	md-moea & mah-moea $P(H)$	mah-moea & mh-moea $P(H)$
DTLZ1 (3)	0.000000 (1)	0.000000 (1)	0.464273 (0)
DTLZ2 (3)	0.000000 (1)	0.000000 (1)	0.203559 (0)
DTLZ3 (3)	0.000000 (1)	0.000000 (1)	0.100764 (0)
DTLZ4 (3)	0.000000 (1)	0.000000 (1)	0.958731 (0)
DTLZ5 (3)	0.000000 (1)	0.000000 (1)	0.125965 (0)
DTLZ6 (3)	0.000000 (1)	0.784460 (0)	0.180900 (0)
DTLZ7 (3)	0.000002 (1)	0.000001 (1)	0.510598 (0)
DTLZ1 (4)	0.000168 (1)	0.000000 (1)	0.000002 (1)
DTLZ2 (4)	0.000000 (1)	0.000000 (1)	0.048413 (1)
DTLZ3 (4)	0.000000 (1)	0.000586 (1)	0.428630 (0)
DTLZ4 (4)	0.000000 (1)	0.000000 (1)	0.355472 (0)
DTLZ5 (4)	0.000000 (1)	0.000000 (1)	0.005570 (1)
DTLZ6 (4)	0.067869 (0)	0.000000 (1)	0.046756 (1)
DTLZ7 (4)	0.000000 (1)	0.000001 (1)	0.010763 (1)
DTLZ1 (5)	0.055546 (0)	0.002624 (1)	0.000000 (1)
DTLZ2 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ3 (5)	0.000007 (1)	0.111927 (0)	0.183242 (0)
DTLZ4 (5)	0.000000 (1)	0.000000 (1)	0.000015 (1)
DTLZ5 (5)	0.000000 (1)	0.000000 (1)	0.010315 (1)
DTLZ6 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ7 (5)	0.000000 (1)	0.000001 (1)	0.464273 (0)
DTLZ1 (6)	0.000038 (1)	0.006096 (1)	0.000000 (1)
DTLZ2 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ3 (6)	0.039873 (1)	0.003337 (1)	0.000004 (1)
DTLZ4 (6)	0.000001 (1)	0.000000 (1)	0.000000 (1)
DTLZ5 (6)	0.000000 (1)	0.000000 (1)	0.005570 (1)
DTLZ6 (6)	0.000000 (1)	0.000000 (1)	0.000001 (1)
DTLZ7 (6)	0.000000 (1)	0.153667 (0)	0.428963 (0)

Table 6.6: Statistical analysis using Wilcoxon's rank sum for the DTLZ test problems. For this, we used I_H, see Table 6.2. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

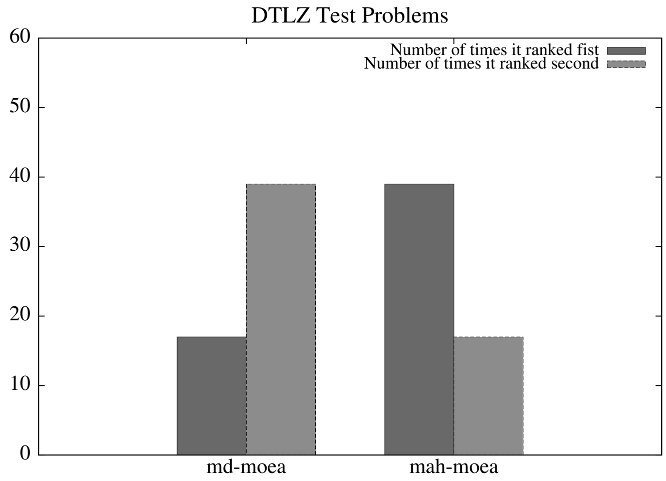
$\vec{f}$	mc-moea & md-moea $P(H)$	md-moea & mah-moea $P(H)$	mah-moea & mh-moea $P(H)$
WFG1 (3)	0.129670 (0)	0.000000 (1)	0.277189 (0)
WFG2 (3)	0.007959 (1)	1.000000 (0)	1.000000 (0)
WFG3 (3)	0.000000 (1)	0.000000 (1)	0.087710 (0)
WFG4 (3)	0.000000 (1)	0.000000 (1)	0.004033 (1)
WFG5 (3)	0.000000 (1)	0.000000 (1)	0.446413 (0)
WFG6 (3)	0.000000 (1)	0.000000 (1)	0.332841 (0)
WFG7 (3)	0.000000 (1)	0.000000 (1)	0.420386 (0)
WFG1 (4)	0.000000 (1)	0.000000 (1)	0.728265 (0)
WFG2 (4)	0.946956 (0)	1.000000 (0)	1.000000 (0)
WFG3 (4)	0.000000 (1)	0.000003 (1)	0.005322 (1)
WFG4 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG5 (4)	0.000000 (1)	0.000000 (1)	0.000006 (1)
WFG6 (4)	0.000000 (1)	0.000004 (1)	0.813003 (0)
WFG7 (4)	0.000000 (1)	0.000000 (1)	0.000077 (1)
WFG1 (5)	0.000000 (1)	0.000000 (1)	0.033874 (1)
WFG2 (5)	0.403538 (0)	1.000000 (0)	1.000000 (0)
WFG3 (5)	0.000000 (1)	0.841801 (0)	0.888303 (0)
WFG4 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG5 (5)	0.000000 (1)	0.000000 (1)	0.876635 (0)
WFG6 (5)	0.000000 (1)	0.006669 (1)	0.888303 (0)
WFG7 (5)	0.000000 (1)	0.000000 (1)	0.009883 (1)
WFG1 (6)	0.000000 (1)	0.000000 (1)	0.001767 (1)
WFG2 (6)	0.055546 (0)	1.000000 (0)	1.000000 (0)
WFG3 (6)	0.000000 (1)	0.515261 (0)	0.374931 (0)
WFG4 (6)	0.000000 (1)	0.000000 (1)	0.264326 (0)
WFG5 (6)	0.000000 (1)	0.001680 (1)	0.074827 (0)
WFG6 (6)	0.000000 (1)	0.283778 (0)	0.807275 (0)
WFG7 (6)	0.000000 (1)	0.000000 (1)	0.024157 (1)

Table 6.7: Statistical analysis using Wilcoxon's rank sum for the WFG test problems. For this, we used I_H, see Table 6.4. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.



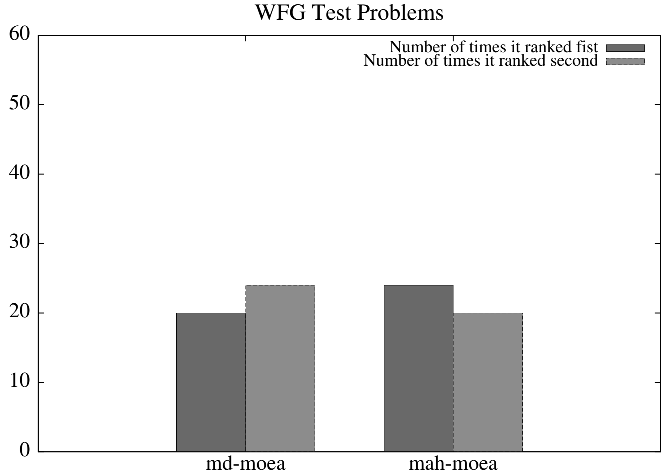
$\vec{f}$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{A})$	$\vec{f}$	$I_{SC}(\mathcal{A},\mathcal{B})$	$I_{SC}(\mathcal{B},\mathcal{B})$
DTLZ1 (3)	0.0040	**0.0577**	DTLZ1 (7)	0.0090	**0.3713**
DTLZ2 (3)	0.0457	**0.4613**	DTLZ2 (7)	0.0013	**0.1653**
DTLZ3 (3)	0.0320	**0.1180**	DTLZ3 (7)	0.0097	**0.2893**
DTLZ4 (3)	0.0453	**0.4700**	DTLZ4 (7)	0.0023	**0.0840**
DTLZ5 (3)	0.1610	**0.4800**	DTLZ5 (7)	0.0260	**0.5730**
DTLZ6 (3)	**0.6693**	0.6390	DTLZ6 (7)	0.0000	**0.9887**
DTLZ7 (3)	0.0903	**0.2870**	DTLZ7 (7)	0.0287	**0.0373**
DTLZ1 (4)	0.0077	**0.0283**	DTLZ1 (8)	0.0253	**0.8963**
DTLZ2 (4)	0.0227	**0.2827**	DTLZ2 (8)	0.0000	**0.1813**
DTLZ3 (4)	0.0153	**0.0773**	DTLZ3 (8)	0.0060	**0.7860**
DTLZ4 (4)	0.0213	**0.2647**	DTLZ4 (8)	0.0013	**0.0617**
DTLZ5 (4)	0.0317	**0.4537**	DTLZ5 (8)	0.0227	**0.5470**
DTLZ6 (4)	0.0137	**0.9117**	DTLZ6 (8)	0.0000	**0.9867**
DTLZ7 (4)	0.0640	**0.2723**	DTLZ7 (8)	**0.0173**	0.0140
DTLZ1 (5)	0.0110	**0.0277**	DTLZ1 (9)	0.1243	**0.9667**
DTLZ2 (5)	0.0153	**0.1567**	DTLZ2 (9)	0.0000	**0.3167**
DTLZ3 (5)	0.0063	**0.0893**	DTLZ3 (9)	0.0033	**0.9230**
DTLZ4 (5)	0.0160	**0.1377**	DTLZ4 (9)	0.0023	**0.0337**
DTLZ5 (5)	0.0193	**0.5630**	DTLZ5 (9)	0.0303	**0.4677**
DTLZ6 (5)	0.0040	**0.9790**	DTLZ6 (9)	0.0000	**0.9887**
DTLZ7 (5)	0.0433	**0.1357**	DTLZ7 (9)	0.0070	**0.0073**
DTLZ1 (6)	0.0103	**0.0553**	DTLZ1 (10)	0.0773	**0.9837**
DTLZ2 (6)	0.0067	**0.1380**	DTLZ2 (10)	0.0000	**0.7277**
DTLZ3 (6)	0.0090	**0.0880**	DTLZ3 (10)	0.0017	**0.9673**
DTLZ4 (6)	0.0067	**0.1143**	DTLZ4 (10)	0.0000	**0.0370**
DTLZ5 (6)	0.0270	**0.5767**	DTLZ5 (10)	0.0387	**0.2883**
DTLZ6 (6)	0.0000	**0.9867**	DTLZ6 (10)	0.0000	**0.9817**
DTLZ7 (6)	0.0280	**0.0650**	DTLZ7 (10)	0.0033	**0.0037**

Table 6.8: Results obtained in the DTLZ test problems with up to ten objective functions. We compare *MD-MOEA* and *MAH-MOEA* with respect to I_{SC}. In this case, $\mathcal{A}$ is the set composed by all solutions found by *MD-MOEA* considering all 30 independent runs and $\mathcal{B}$ is the set composed by all solutions found by *MAH-MOEA* considering all 30 independent runs.



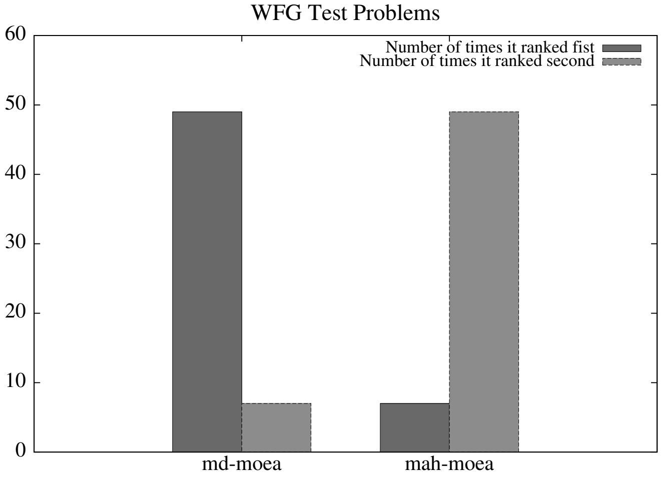
$\vec{f}$	md-moea	mah-moea	$\vec{f}$	md-moea	mah-moea
	I_S	I_S		I_S	I_S
DTLZ1(3)	0.0383(0.051)	**0.0227(0.014)**	DTLZ1(7)	4.7387(8.298)	**0.1142(0.170)**
DTLZ2(3)	**0.0494(0.003)**	0.0519(0.005)	DTLZ2(7)	**0.1752(0.018)**	0.2028(0.027)
DTLZ3(3)	39.7739(16.843)	**0.0835(0.114)**	DTLZ3(7)	17.6441(27.499)	**0.6413(1.399)**
DTLZ4(3)	**0.0485(0.004)**	0.0534(0.005)	DTLZ4(7)	**0.1736(0.021)**	0.2011(0.018)
DTLZ5(3)	**0.0107(0.001)**	0.0149(0.002)	DTLZ5(7)	0.3055(0.034)	**0.2296(0.046)**
DTLZ6(3)	0.0477(0.050)	**0.0244(0.015)**	DTLZ6(7)	0.7690(0.102)	**0.3620(0.061)**
DTLZ7(3)	**0.0515(0.012)**	0.0548(0.011)	DTLZ7(7)	0.2959(0.081)	**0.2948(0.077)**
DTLZ1(4)	0.3313(1.026)	**0.0716(0.139)**	DTLZ1(8)	44.9282(15.636)	**3.1587(3.930)**
DTLZ2(4)	**0.0821(0.010)**	0.0976(0.007)	DTLZ2(8)	**0.2151(0.023)**	0.2337(0.021)
DTLZ3(4)	30.1241(24.519)	**0.1443(0.152)**	DTLZ3(8)	121.0272(70.941)	**2.0211(4.517)**
DTLZ4(4)	**0.0852(0.008)**	0.0979(0.007)	DTLZ4(8)	**0.1982(0.024)**	0.2265(0.020)
DTLZ5(4)	0.1303(0.028)	**0.0871(0.009)**	DTLZ5(8)	0.3297(0.035)	**0.2416(0.040)**
DTLZ6(4)	0.2187(0.036)	**0.1289(0.020)**	DTLZ6(8)	0.8535(0.075)	**0.3994(0.049)**
DTLZ7(4)	0.1223(0.015)	**0.1222(0.018)**	DTLZ7(8)	0.3472(0.080)	**0.3425(0.078)**
DTLZ1(5)	0.1476(0.311)	**0.1434(0.437)**	DTLZ1(9)	53.2934(10.620)	**9.9927(4.748)**
DTLZ2(5)	**0.1128(0.011)**	0.1406(0.012)	DTLZ2(9)	0.2740(0.076)	**0.2477(0.026)**
DTLZ3(5)	5.9726(12.907)	**0.2318(0.270)**	DTLZ3(9)	163.9581(73.838)	**0.8544(0.974)**
DTLZ4(5)	**0.1160(0.013)**	0.1379(0.011)	DTLZ4(9)	**0.2206(0.023)**	0.2521(0.024)
DTLZ5(5)	0.2209(0.029)	**0.1379(0.021)**	DTLZ5(9)	0.3472(0.040)	**0.2594(0.045)**
DTLZ6(5)	0.4020(0.044)	**0.2472(0.058)**	DTLZ6(9)	0.9396(0.103)	**0.4521(0.052)**
DTLZ7(5)	0.1837(0.033)	**0.1735(0.030)**	DTLZ7(9)	0.5877(0.906)	**0.5839(0.907)**
DTLZ1(6)	0.5147(1.209)	**0.1094(0.109)**	DTLZ1(10)	56.1186(11.811)	**13.1724(5.369)**
DTLZ2(6)	**0.1471(0.015)**	0.1782(0.016)	DTLZ2(10)	0.4853(0.102)	**0.2714(0.028)**
DTLZ3(6)	8.5028(15.487)	**0.6253(2.037)**	DTLZ3(10)	186.9971(41.579)	**3.5598(8.554)**
DTLZ4(6)	**0.1416(0.012)**	0.1777(0.018)	DTLZ4(10)	**0.2555(0.034)**	0.2882(0.033)
DTLZ5(6)	0.2670(0.033)	**0.1933(0.026)**	DTLZ5(10)	0.3324(0.049)	**0.2802(0.052)**
DTLZ6(6)	0.6232(0.066)	**0.3207(0.070)**	DTLZ6(10)	0.9776(0.088)	**0.4803(0.053)**
DTLZ7(6)	**0.2478(0.047)**	0.2520(0.055)	DTLZ7(10)	0.9527(1.374)	**0.8445(1.303)**

Table 6.9: Results obtained in the DTLZ test problems with up to ten objective functions. We compare *MD-MOEA* and *MAH-MOEA* with respect to I_S. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



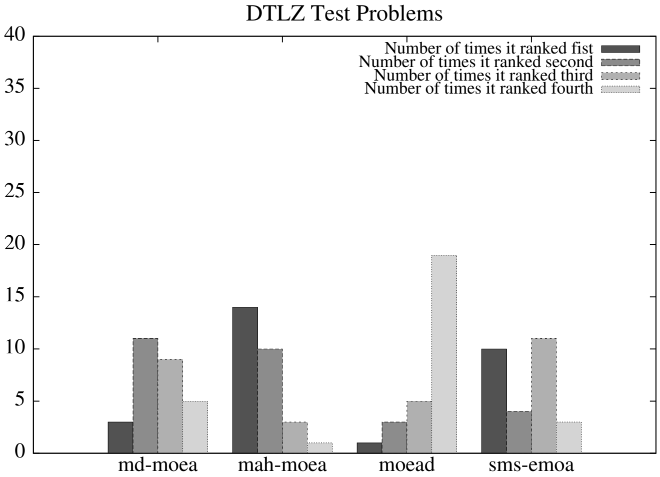
$\vec{f}$	md-moea	mah-moea	$\vec{f}$	md-moea	mah-moea
	I_{SC}	I_{SC}		I_{SC}	I_{SC}
WFG1 (3)	0.0000	**0.0017**	WFG1 (7)	0.0000	0.0000
WFG2 (3)	**0.6703**	0.6703	WFG2 (7)	**0.7737**	0.7737
WFG3 (3)	0.0140	**0.5907**	WFG3 (7)	0.2647	**0.2687**
WFG4 (3)	0.0003	**0.9870**	WFG4 (7)	0.0007	**0.0543**
WFG5 (3)	0.0000	**0.2183**	WFG5 (7)	0.0013	**0.0020**
WFG6 (3)	0.0157	**0.6220**	WFG6 (7)	**0.3477**	0.3410
WFG7 (3)	0.0153	**0.1930**	WFG7 (7)	0.0000	0.0000
WFG1 (4)	0.0000	0.0000	WFG1 (8)	0.0000	0.0000
WFG2 (4)	**0.8647**	0.8647	WFG2 (8)	**0.7787**	0.7787
WFG3 (4)	0.0217	**0.1677**	WFG3 (8)	**0.1923**	0.1923
WFG4 (4)	0.0000	**0.6527**	WFG4 (8)	0.0000	**0.0267**
WFG5 (4)	0.0000	**0.1337**	WFG5 (8)	0.0013	**0.0030**
WFG6 (4)	0.0300	**0.2420**	WFG6 (8)	0.2877	**0.3213**
WFG7 (4)	**0.0030**	0.0010	WFG7 (8)	0.0000	0.0000
WFG1 (5)	0.0000	0.0000	WFG1 (9)	0.0000	0.0000
WFG2 (5)	**0.8357**	0.8357	WFG2 (9)	**0.7750**	0.7750
WFG3 (5)	**0.1807**	0.1787	WFG3 (9)	**0.1383**	0.1383
WFG4 (5)	0.0000	**0.3083**	WFG4 (9)	0.0000	**0.0177**
WFG5 (5)	0.0000	**0.0317**	WFG5 (9)	0.0093	**0.0103**
WFG6 (5)	0.3310	**0.3520**	WFG6 (9)	**0.2753**	0.2383
WFG7 (5)	0.0000	0.0000	WFG7 (9)	0.0000	0.0000
WFG1 (6)	0.0000	0.0000	WFG1 (10)	0.0000	0.0000
WFG2 (6)	**0.8113**	0.8113	WFG2 (10)	**0.7377**	0.7377
WFG3 (6)	**0.3297**	0.3210	WFG3 (10)	**0.1050**	0.1050
WFG4 (6)	0.0000	**0.1350**	WFG4 (10)	0.0000	**0.0057**
WFG5 (6)	0.0000	**0.0020**	WFG5 (10)	**0.0147**	0.0147
WFG6 (6)	**0.4010**	0.3943	WFG6 (10)	**0.2247**	0.2097
WFG7 (6)	0.0000	0.0000	WFG7 (10)	0.0000	0.0000

Table 6.10: Results obtained in the WFG test problems with up to ten objective functions. We compare *MD-MOEA* and *MAH-MOEA* with respect to I_{SC}. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



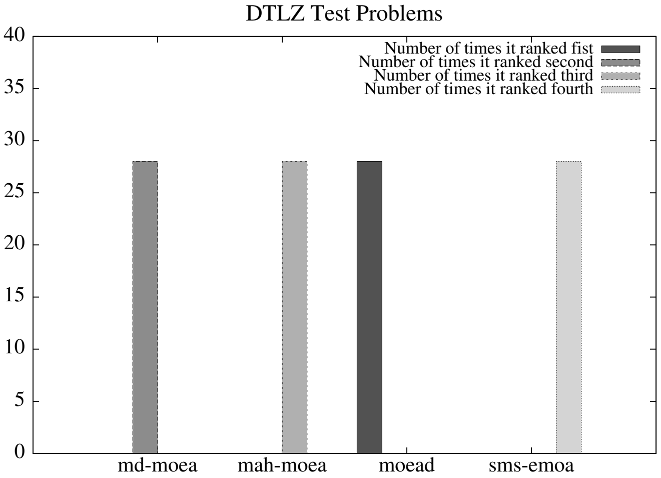
$\vec{f}$	md-moea I_S	mah-moea I_S	$\vec{f}$	md-moea I_S	mah-moea I_S
WFG1(3)	**0.0545(0.009)**	0.0815(0.022)	WFG1(7)	**0.0807(0.011)**	0.0824(0.009)
WFG2(3)	**0.0224(0.009)**	0.0224(0.009)	WFG2(7)	**0.0516(0.018)**	0.0516(0.018)
WFG3(3)	**0.0181(0.002)**	0.0242(0.002)	WFG3(7)	**0.0900(0.024)**	0.0906(0.024)
WFG4(3)	**0.1586(0.015)**	0.1821(0.016)	WFG4(7)	**0.9768(0.112)**	1.6140(0.152)
WFG5(3)	**0.0885(0.010)**	0.1049(0.010)	WFG5(7)	**0.5441(0.072)**	0.5752(0.083)
WFG6(3)	**0.0380(0.006)**	0.0453(0.005)	WFG6(7)	**0.1831(0.039)**	0.1833(0.042)
WFG7(3)	**0.1226(0.011)**	0.1511(0.013)	WFG7(7)	**0.5718(0.056)**	0.6547(0.141)
WFG1(4)	**0.0643(0.008)**	0.0979(0.012)	WFG1(8)	**0.0877(0.013)**	0.0908(0.016)
WFG2(4)	**0.0349(0.013)**	0.0349(0.013)	WFG2(8)	**0.0614(0.017)**	0.0614(0.017)
WFG3(4)	**0.0286(0.003)**	0.0356(0.003)	WFG3(8)	**0.0950(0.016)**	0.0950(0.016)
WFG4(4)	**0.3332(0.028)**	0.4306(0.035)	WFG4(8)	**1.2455(0.117)**	1.9606(0.270)
WFG5(4)	**0.1580(0.015)**	0.1813(0.014)	WFG5(8)	**0.6427(0.061)**	0.6614(0.072)
WFG6(4)	**0.0837(0.020)**	0.0846(0.016)	WFG6(8)	**0.2072(0.046)**	0.2139(0.041)
WFG7(4)	**0.2277(0.023)**	0.3390(0.024)	WFG7(8)	0.6796(0.066)	**0.6442(0.171)**
WFG1(5)	**0.0703(0.007)**	0.0971(0.009)	WFG1(9)	0.1019(0.014)	**0.1016(0.013)**
WFG2(5)	**0.0390(0.017)**	0.0390(0.017)	WFG2(9)	**0.0754(0.033)**	0.0754(0.033)
WFG3(5)	0.0505(0.004)	**0.0500(0.004)**	WFG3(9)	**0.1036(0.020)**	0.1036(0.020)
WFG4(5)	**0.5066(0.065)**	0.7625(0.076)	WFG4(9)	**1.5718(0.213)**	2.1582(0.318)
WFG5(5)	**0.2592(0.040)**	0.2649(0.035)	WFG5(9)	0.7540(0.080)	**0.7505(0.081)**
WFG6(5)	**0.1152(0.020)**	0.1307(0.027)	WFG6(9)	**0.2350(0.039)**	0.2381(0.045)
WFG7(5)	**0.3302(0.038)**	0.5677(0.050)	WFG7(9)	0.8251(0.106)	**0.6846(0.161)**
WFG1(6)	**0.0736(0.009)**	0.0887(0.012)	WFG1(10)	**0.1149(0.014)**	0.1149(0.014)
WFG2(6)	**0.0521(0.025)**	0.0521(0.025)	WFG2(10)	**0.0761(0.030)**	0.0761(0.030)
WFG3(6)	**0.0744(0.014)**	0.0755(0.014)	WFG3(10)	**0.1129(0.026)**	0.1129(0.026)
WFG4(6)	**0.7314(0.086)**	1.1253(0.095)	WFG4(10)	**1.8951(0.216)**	2.3724(0.330)
WFG5(6)	**0.4318(0.062)**	0.4621(0.059)	WFG5(10)	**0.8311(0.100)**	0.8390(0.115)
WFG6(6)	**0.1510(0.031)**	0.1605(0.031)	WFG6(10)	0.2894(0.052)	**0.2653(0.041)**
WFG7(6)	**0.4422(0.058)**	0.6964(0.129)	WFG7(10)	0.9441(0.105)	**0.7823(0.177)**

Table 6.11: Results obtained in the WFG test problems with up to ten objective functions. We compare *MD-MOEA* and *MAH-MOEA* with respect to I_S. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



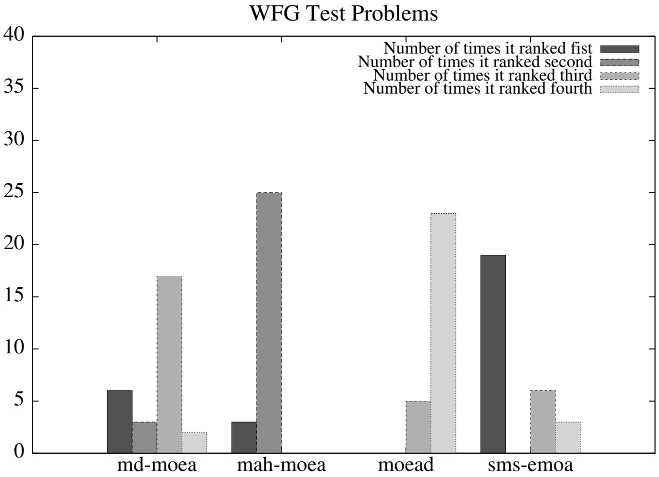
$\vec{f}$	md-moea	mah-moea	moead	sms-emoa
	I_H	I_H	I_H	I_H
DTLZ1 (3)	1.1823 (0.003)	1.1896 (0.001)	1.2683 (0.001)	**1.3128 (0.001)**
DTLZ2 (3)	0.7319 (0.003)	0.7525 (0.002)	0.7203 (0.000)	**0.7645 (0.002)**
DTLZ3 (3)	1.3281 (0.000)	1.3263 (0.000)	1.3307 (0.000)	**1.3309 (0.000)**
DTLZ4 (3)	0.7304 (0.006)	0.7492 (0.004)	0.7255 (0.000)	**0.7699 (0.002)**
DTLZ5 (3)	0.2651 (0.000)	0.2661 (0.000)	0.2467 (0.001)	**0.2662 (0.000)**
DTLZ6 (3)	**1.0989 (0.008)**	1.0975 (0.009)	0.5129 (0.022)	0.5979 (0.012)
DTLZ7 (3)	0.5519 (0.041)	**0.5638 (0.043)**	0.4509 (0.027)	0.5366 (0.060)
DTLZ1 (4)	1.4375 (0.005)	1.4444 (0.002)	1.4134 (0.003)	**1.4621 (0.001)**
DTLZ2 (4)	1.1249 (0.010)	**1.1647 (0.005)**	0.9237 (0.001)	1.0700 (0.004)
DTLZ3 (4)	**1.4638 (0.000)**	1.4638 (0.000)	1.4638 (0.000)	1.4630 (0.001)
DTLZ4 (4)	1.1912 (0.010)	**1.2252 (0.006)**	0.9169 (0.001)	1.0645 (0.005)
DTLZ5 (4)	0.4668 (0.023)	0.5469 (0.004)	0.4921 (0.004)	**0.5618 (0.004)**
DTLZ6 (4)	1.2742 (0.024)	**1.3600 (0.003)**	1.2083 (0.006)	1.3085 (0.007)
DTLZ7 (4)	0.7062 (0.016)	**0.7262 (0.018)**	0.3875 (0.022)	0.6922 (0.043)
DTLZ1 (5)	1.6011 (0.003)	1.6031 (0.001)	1.6095 (0.000)	**1.6103 (0.000)**
DTLZ2 (5)	1.4063 (0.015)	**1.4519 (0.006)**	1.0345 (0.004)	1.2717 (0.017)
DTLZ3 (5)	**1.6098 (0.000)**	1.6098 (0.000)	1.6096 (0.000)	1.5998 (0.007)
DTLZ4 (5)	1.4801 (0.011)	**1.5096 (0.007)**	1.0317 (0.005)	1.2778 (0.016)
DTLZ5 (5)	0.8122 (0.033)	0.9180 (0.015)	0.7215 (0.022)	**0.9540 (0.006)**
DTLZ6 (5)	1.3396 (0.063)	**1.5450 (0.005)**	1.3358 (0.012)	1.4298 (0.009)
DTLZ7 (5)	0.8534 (0.009)	**0.8699 (0.010)**	0.1285 (0.089)	0.7547 (0.021)
DTLZ1 (6)	1.7652 (0.004)	1.7669 (0.001)	**1.7696 (0.000)**	1.7675 (0.009)
DTLZ2 (6)	1.6757 (0.014)	**1.7114 (0.006)**	1.0435 (0.010)	1.4695 (0.020)
DTLZ3 (6)	1.7710 (0.000)	**1.7712 (0.000)**	1.7702 (0.000)	1.7640 (0.007)
DTLZ4 (6)	1.7234 (0.008)	**1.7407 (0.003)**	1.0187 (0.007)	1.4427 (0.018)
DTLZ5 (6)	0.9563 (0.067)	1.1170 (0.041)	0.8622 (0.014)	**1.2291 (0.006)**
DTLZ6 (6)	1.0454 (0.174)	**1.6984 (0.004)**	1.4940 (0.021)	1.6855 (0.005)
DTLZ7 (6)	0.9319 (0.022)	**0.9380 (0.026)**	0.0255 (0.005)	0.7375 (0.009)

Table 6.12: Results obtained in the DTLZ test problems with up to six objective functions. We compare *MD-MOEA*, *MAH-MOEA*, *MOEA/D* and *SMS-EMOA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



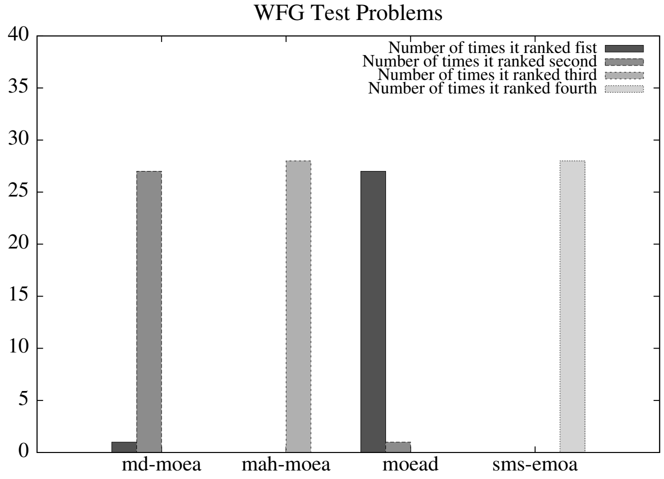
$\vec{f}$	md-moea	mah-moea	moead	sms-emoa
	time	time	time	time
DTLZ1 (3)	0.8317 (0.025)	26.5683 (6.888)	**0.5180 (0.009)**	1368.8483 (56.345)
DTLZ2 (3)	1.7247 (0.016)	80.2447 (11.191)	**0.5732 (0.008)**	3186.9297 (33.957)
DTLZ3 (3)	0.7897 (0.028)	26.4793 (5.165)	**0.5092 (0.008)**	2516.9973 (131.555)
DTLZ4 (3)	1.7187 (0.018)	77.7587 (9.585)	**0.5839 (0.003)**	3209.5767 (28.945)
DTLZ5 (3)	1.2467 (0.019)	9.6463 (2.308)	**0.5773 (0.023)**	485.1490 (13.942)
DTLZ6 (3)	1.0763 (0.020)	26.3760 (12.040)	**0.5055 (0.025)**	1872.8680 (494.697)
DTLZ7 (3)	1.0917 (0.036)	28.6033 (7.032)	**0.5600 (0.034)**	3047.7933 (245.354)
DTLZ1 (4)	1.0227 (0.015)	17.1963 (2.158)	**0.5386 (0.013)**	953.9040 (72.115)
DTLZ2 (4)	2.0807 (0.009)	56.9457 (9.412)	**0.6048 (0.010)**	2696.8443 (28.356)
DTLZ3 (4)	1.0830 (0.029)	23.3403 (2.858)	**0.5312 (0.010)**	1652.6900 (191.963)
DTLZ4 (4)	2.0533 (0.020)	52.1267 (9.269)	**0.6281 (0.007)**	3186.0707 (23.889)
DTLZ5 (4)	2.3523 (0.028)	50.0157 (6.454)	**0.5903 (0.001)**	1559.1663 (56.480)
DTLZ6 (4)	1.6473 (0.020)	49.7007 (6.381)	**0.5942 (0.008)**	1758.4097 (30.011)
DTLZ7 (4)	1.1903 (0.026)	16.5777 (2.984)	**0.6512 (0.008)**	3240.0063 (293.247)
DTLZ1 (5)	1.1693 (0.037)	12.5180 (1.662)	**0.5706 (0.005)**	1029.0453 (92.775)
DTLZ2 (5)	2.3043 (0.014)	29.8747 (3.466)	**0.6248 (0.001)**	1879.0867 (19.569)
DTLZ3 (5)	1.2967 (0.028)	17.2507 (2.455)	**0.5796 (0.016)**	1617.6280 (75.710)
DTLZ4 (5)	2.2767 (0.021)	28.0337 (3.063)	**0.6812 (0.001)**	2191.7757 (21.185)
DTLZ5 (5)	2.4603 (0.021)	44.2683 (5.359)	**0.6527 (0.040)**	1788.9270 (49.232)
DTLZ6 (5)	2.2447 (0.044)	46.6733 (5.942)	**0.6370 (0.006)**	1957.4123 (25.557)
DTLZ7 (5)	1.3070 (0.035)	7.3807 (1.137)	**0.6935 (0.005)**	2822.8597 (171.838)
DTLZ1 (6)	1.3033 (0.070)	10.5600 (1.946)	**0.6036 (0.016)**	1523.3607 (158.557)
DTLZ2 (6)	2.4600 (0.023)	19.9033 (1.739)	**0.6551 (0.005)**	1605.2567 (13.806)
DTLZ3 (6)	1.4473 (0.030)	15.7413 (2.575)	**0.5999 (0.013)**	1751.4160 (130.305)
DTLZ4 (6)	2.4343 (0.026)	18.3560 (2.040)	**0.7307 (0.001)**	1827.7093 (18.135)
DTLZ5 (6)	2.6093 (0.028)	37.3323 (5.122)	**0.7436 (0.001)**	1815.6567 (68.869)
DTLZ6 (6)	2.8480 (0.049)	44.0187 (5.048)	**0.6892 (0.005)**	2128.5127 (27.407)
DTLZ7 (6)	1.4347 (0.047)	3.7347 (0.632)	**0.7247 (0.004)**	2522.0583 (54.170)

Table 6.13: Results obtained in the DTLZ test problems with up to six objective functions. We compare *MD-MOEA*, *MAH-MOEA*, *MOEA/D* and *SMS-EMOA* with respect to the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	md-moea	mah-moea	moead	sms-emoa
	I_H	I_H	I_H	I_H
WFG1 (3)	1.0043 (0.057)	1.1274 (0.049)	0.9178 (0.017)	**1.2126 (0.024)**
WFG2 (3)	**0.7559 (0.091)**	0.7559 (0.091)	0.1593 (0.204)	0.7460 (0.132)
WFG3 (3)	0.6156 (0.004)	0.6312 (0.002)	0.5001 (0.026)	**0.6357 (0.002)**
WFG4 (3)	0.6584 (0.008)	0.7414 (0.002)	0.5952 (0.013)	**0.7515 (0.002)**
WFG5 (3)	0.5325 (0.005)	0.5529 (0.001)	0.4715 (0.010)	**0.5573 (0.002)**
WFG6 (3)	0.5483 (0.003)	0.5602 (0.004)	0.4538 (0.007)	**0.5639 (0.002)**
WFG7 (3)	0.6099 (0.013)	0.7342 (0.005)	0.4908 (0.056)	**0.7495 (0.004)**
WFG1 (4)	1.0396 (0.027)	1.2045 (0.045)	1.1014 (0.061)	**1.4089 (0.008)**
WFG2 (4)	**0.6338 (0.134)**	0.6338 (0.134)	0.0577 (0.111)	0.1102 (0.139)
WFG3 (4)	0.5674 (0.009)	**0.5801 (0.009)**	0.2813 (0.035)	0.5665 (0.013)
WFG4 (4)	0.8193 (0.017)	0.9955 (0.007)	0.6576 (0.026)	**1.0160 (0.004)**
WFG5 (4)	0.5501 (0.008)	0.5870 (0.003)	0.3677 (0.015)	**0.5913 (0.002)**
WFG6 (4)	0.5510 (0.016)	0.5712 (0.010)	0.2890 (0.016)	**0.5817 (0.008)**
WFG7 (4)	0.6163 (0.018)	0.8623 (0.008)	0.2920 (0.036)	**0.9224 (0.007)**
WFG1 (5)	1.0896 (0.030)	1.2095 (0.031)	1.1550 (0.069)	**1.4687 (0.010)**
WFG2 (5)	**0.6349 (0.203)**	0.6349 (0.203)	0.1387 (0.148)	0.0001 (0.000)
WFG3 (5)	**0.5753 (0.018)**	0.5742 (0.018)	0.1566 (0.037)	0.2116 (0.092)
WFG4 (5)	0.9373 (0.022)	1.2076 (0.009)	0.6592 (0.024)	**1.2305 (0.008)**
WFG5 (5)	0.5744 (0.014)	0.6142 (0.008)	0.2357 (0.014)	**0.6457 (0.003)**
WFG6 (5)	0.5557 (0.023)	**0.5735 (0.025)**	0.2536 (0.015)	0.4463 (0.052)
WFG7 (5)	0.6113 (0.023)	0.9482 (0.019)	0.2163 (0.014)	**1.0149 (0.012)**
WFG1 (6)	1.2711 (0.036)	1.3288 (0.031)	1.0911 (0.030)	**1.5042 (0.024)**
WFG2 (6)	**0.7530 (0.164)**	0.7530 (0.164)	0.0839 (0.107)	0.0000 (0.000)
WFG3 (6)	**0.5635 (0.027)**	0.5575 (0.032)	0.1615 (0.050)	0.1057 (0.056)
WFG4 (6)	1.0276 (0.032)	1.4025 (0.015)	0.5982 (0.029)	**1.4082 (0.009)**
WFG5 (6)	0.5893 (0.023)	0.6099 (0.024)	0.1555 (0.017)	**0.6664 (0.014)**
WFG6 (6)	0.6030 (0.034)	**0.6134 (0.032)**	0.2319 (0.019)	0.4075 (0.057)
WFG7 (6)	0.6100 (0.023)	0.9036 (0.071)	0.1842 (0.014)	**1.0398 (0.051)**

Table 6.14: Results obtained in the WFG test problems with up to six objective functions. We compare *MD-MOEA*, *MAH-MOEA*, *MOEA/D* and *SMS-EMOA* using the hypervolume indicator I_H. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.



$\vec{f}$	md-moea time	mah-moea time	moead time	sms-emoa time
WFG1 (3)	3.0037 (0.036)	97.5580 (11.196)	**1.3196 (0.214)**	4308.4217 (25.228)
WFG2 (3)	1.3063 (0.081)	1.5803 (0.314)	**1.1146 (0.176)**	1513.2233 (155.007)
WFG3 (3)	2.7167 (0.036)	87.2683 (15.644)	**0.9307 (0.007)**	2084.2703 (17.679)
WFG4 (3)	2.9810 (0.040)	78.2380 (14.106)	**1.0456 (0.120)**	4153.4260 (30.095)
WFG5 (3)	3.1817 (0.119)	122.2553 (17.817)	**0.9530 (0.014)**	3551.4573 (41.543)
WFG6 (3)	2.6367 (0.115)	95.7627 (14.819)	**0.9624 (0.014)**	3064.4163 (27.785)
WFG7 (3)	3.7967 (0.053)	138.2930 (26.767)	**1.1576 (0.005)**	5333.1087 (47.859)
WFG1 (4)	3.1263 (0.028)	55.2243 (6.841)	**1.6046 (0.012)**	2830.8050 (29.269)
WFG2 (4)	1.3700 (0.040)	1.7497 (0.150)	**1.2374 (0.144)**	1471.5220 (222.983)
WFG3 (4)	2.3457 (0.082)	37.7127 (6.090)	**1.4797 (0.384)**	1690.0347 (18.244)
WFG4 (4)	3.5377 (0.070)	66.3593 (11.242)	**1.4224 (0.004)**	3919.7280 (52.832)
WFG5 (4)	3.3533 (0.029)	75.2213 (11.970)	**0.9704 (0.004)**	3103.7243 (30.165)
WFG6 (4)	2.2033 (0.062)	39.5240 (5.911)	**1.0423 (0.059)**	2524.6620 (34.021)
WFG7 (4)	4.1817 (0.043)	85.7517 (9.854)	**1.2276 (0.009)**	3888.7363 (56.772)
WFG1 (5)	3.1180 (0.048)	25.0433 (3.385)	**1.7252 (0.011)**	1695.3017 (19.921)
WFG2 (5)	1.5483 (0.055)	2.0567 (0.346)	**1.2578 (0.184)**	1803.4930 (258.393)
WFG3 (5)	**1.8530 (0.023)**	2.7177 (0.374)	1.9463 (0.057)	1598.1797 (31.162)
WFG4 (5)	3.8390 (0.030)	52.2173 (8.366)	**1.5348 (0.006)**	2923.8940 (48.865)
WFG5 (5)	3.1130 (0.119)	45.3483 (6.833)	**1.0555 (0.011)**	3149.2743 (70.538)
WFG6 (5)	1.8200 (0.014)	3.6643 (0.604)	**1.1502 (0.013)**	2355.6050 (41.596)
WFG7 (5)	4.5740 (0.035)	49.3623 (6.280)	**1.4021 (0.037)**	2869.6150 (55.364)
WFG1 (6)	3.0127 (0.039)	13.4993 (1.680)	**1.8216 (0.007)**	1263.6903 (21.408)
WFG2 (6)	1.6703 (0.060)	1.9280 (0.205)	**1.0977 (0.103)**	2278.2863 (262.298)
WFG3 (6)	1.9417 (0.018)	2.9970 (0.399)	**1.7925 (0.307)**	1892.1680 (25.837)
WFG4 (6)	4.0547 (0.035)	47.5760 (7.722)	**1.6597 (0.007)**	2553.0097 (49.627)
WFG5 (6)	2.5957 (0.090)	18.0147 (3.537)	**1.2348 (0.061)**	2927.7660 (222.317)
WFG6 (6)	1.9097 (0.022)	2.7093 (0.111)	**1.1737 (0.012)**	2463.2440 (27.182)
WFG7 (6)	4.7223 (0.048)	44.0367 (6.551)	**1.7574 (0.008)**	2381.4430 (59.016)

Table 6.15: Results obtained in the WFG test problems with up to six objective functions. We compare *MD-MOEA*, *MAH-MOEA*, *MOEA/D* and *SMS-EMOA* in terms of the running time required by each MOEA to obtain the approximation of the Pareto optimal set. The results are in seconds. We show average values over 30 independent runs. The values in parentheses correspond to the standard deviations.

$\vec{f}$	md-moea & moead $P(H)$	md-moea & sms-emoa $P(H)$	mah-moea & moead $P(H)$	mah-moea & sms-emoa $P(H)$
DTLZ1 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ2 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ3 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ4 (3)	0.000068 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ5 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.332841 (0)
DTLZ6 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ7 (3)	0.000000 (1)	0.239850 (0)	0.000000 (1)	0.239850 (0)
DTLZ1 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ2 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ3 (4)	0.000252 (1)	0.000000 (1)	0.135171 (0)	0.000001 (1)
DTLZ4 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ5 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ6 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ7 (4)	0.000000 (1)	0.491783 (0)	0.000000 (1)	0.000000 (1)
DTLZ1 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ2 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ3 (5)	0.000232 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ4 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ5 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ6 (5)	0.264326 (0)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ7 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ1 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000030 (1)
DTLZ2 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ3 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ4 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ5 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ6 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
DTLZ7 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)

Table 6.16: Statistical analysis using Wilcoxon's rank sum for the DTLZ test problems. For this, we used I_H, see Table 6.12. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

$\vec{f}$	md-moea & moead $P(H)$	md-moea & sms-emoa $P(H)$	mah-moea & moead $P(H)$	mah-moea & sms-emoa $P(H)$
WFG1 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG2 (3)	0.000000 (1)	0.599689 (0)	0.000000 (1)	0.599689 (0)
WFG3 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG4 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG5 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG6 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000008 (1)
WFG7 (3)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG1 (4)	0.000000 (1)	0.000000 (1)	0.000001 (1)	0.000000 (1)
WFG2 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG3 (4)	0.000000 (1)	0.923442 (0)	0.000000 (1)	0.000077 (1)
WFG4 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG5 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG6 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000111 (1)
WFG7 (4)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG1 (5)	0.000081 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG2 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG3 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG4 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG5 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG6 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG7 (5)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG1 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG2 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG3 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG4 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.141278 (0)
WFG5 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG6 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)
WFG7 (6)	0.000000 (1)	0.000000 (1)	0.000000 (1)	0.000000 (1)

Table 6.17: Statistical analysis using Wilcoxon's rank sum for the WFG test problems. For this, we used I_H, see Table 6.14. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

Chapter 7

Conclusions and Future Work

In this work, we have studied in depth several selection mechanisms for MOEAs, see Chapter 4. Some of their disadvantages that we can highlight are the following:

- MOEAs based on Pareto dominance have difficulties when we want to solve MOPs with many objective functions, with more than four objective functions. The quick increase in the number of nondominated solutions as we increase the number of objective functions, rapidly dilutes the effect of the selection mechanism of Pareto-based MOEAs [38].

- MOEAs based on the hypervolume indicator (I_H) are computationally expensive, because computing I_H is an NP-hard problem with respect to the number of objective functions [12]; therefore, such MOEAs are impractical to solve MOPs with more than five objective functions. Although some techniques for approximating the contribution to I_H have been proposed, when we use them to solve MOPs, the quality of the approximate Pareto front generated by the MOEA significantly decreases.

- MOEAs based on decomposition need to generate a set of well-distributed weighted vectors and this task becomes harder as we increase the number of objective functions.

- The Maximin Fitness Function (MFF) appears to be a good option because it has interesting properties as well as a linear complexity with respect to the number of objective functions. However, it had only been used in problems with low dimensionality (two or three objective functions).

In spite of these disadvantages, we decided to work with selection mechanisms based on I_H because this indicator is the only unary indicator which is strictly Pareto complaint. And also, we decided to work with selection mechanisms based on MFF because MFF has interesting properties and it is also computationally inexpensive to calculate it.

7.1 Conclusions

First, we studied MOEAs based on I_H, finding that in most cases, they use the traditional competition scheme which operates as follows: If we have a population $\mathcal{P}$, then it generates only one solution per iteration. After that, it applies Pareto ranking. When the last front has more than one solution, it calculates the contribution to I_H of each individual in the last front and it eliminates the individual with the worst contribution. Therefore, when all the individuals are nondominated, we need to calculate $|\mathcal{P}| + 1$ contributions to I_H. Furthermore, in our study about I_H, we found one interesting property of it: the locality property. We proposed a new competition scheme that exploits such property. Our scheme only needs to calculate the contribution of three individuals of the population: the new individual, its nearest neighbor and a randomly selected individual. In this way, we can significantly reduce the running time required by MOEAs based on I_H to obtain the approximate Pareto front. We incorporate this competition scheme into the original SMS-EMOA giving rise to the improved SMS-EMOA (iSMS-EMOA). From our experimental results, we could see that iSMS-EMOA was able to decrease the running time required by SMS-EMOA to generate the approximate Pareto front without losing quality in its solutions. For all our experiments, we used seven problems taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite and seven problems taken from the Walking Fish Group (WFG) toolkit. We executed thirty independent runs for each test problem and we conducted an statistical analysis of results using Wilcoxon's rank sum to validate our experiments.

Since iSMS-EMOA needs to calculate contributions to I_H and this task is a **NP-hard** problem, this MOEA remains impractical to solve MOPs with more than five objective functions. For this reason, we have studied different techniques to approximate the hypervolume, and we found out that the technique proposed by Bringmann and Friedrich [12] is an excellent choice to be incorporated into our competition scheme that exploits the locality property of I_H. This assumption was based on the following hypothesis: Since our competition scheme needs to calculate the contribution of only three individuals, we can reduce the error of the approximation in two ways: (i) by increasing the number of samples without excessively increasing the running time and (ii) by considering that the probability of deleting the individual with the lowest contribution is greater than if we use the traditional competition scheme, because in this case we only deal with three errors and not with P errors (where P is the population size). This hypothesis has been empirically validated in this thesis.

To validate our hypothesis, we proposed the approximate version of the improved SMS-EMOA (aviSMS-EMOA). We compared our proposed aviSMS-EMOA with respect to different versions of the original SMS-EMOA: avoSMS-EMOA (which uses the technique proposed by Bringmann and Friedrich [12] in a traditional competition scheme), hypeSMS-EMOA (which assigns fitness to each individual in the population, using the technique proposed by Bader and Zitzler [3]) and SMS-EMOA. Also, we compared our aviSMS-EMOA with respect to MOEA/D using PBI. We showed that aviSMS-EMOA outperforms avoSMS-EMOA, hypeSMS-EMOA and MOEA/D. Moreover, we can say that our aviSMS-EMOA outperforms SMS-EMOA, if we consider both the quality of the approximation of the Pareto front and the computational cost required to obtain that approximation. This is because our proposed aviSMS-EMOA obtains competitive results with respect to SMS-EMOA but at a much lower computational cost. Although with respect to I_H, SMS-EMOA was better than aviSMS-EMOA in most problems, regarding I_{SC}, in 73% of the total number of test problems used, we saw that aviSMS-EMOA was able to generate solutions that no solution found by SMS-EMOA can dominate. Since we cannot say if the nondominated solutions found by SMS-EMOA are better than the nondominated solutions found by aviSMS-EMOA, we claim that they are both competitive. Finally, we also conducted a study about the number of samples that our aviSMS-EMOA should use to increase the quality of the solutions but without exceeding our maximum running time of four hours and we concluded that 10^5 is a good choice. However, we should not forget that aviSMS-EMOA allows us to balance the quality of the solutions and the running time required to obtain them.

With respect to MFF, we identified two important disadvantages of it:

1. It cannot work well in MOPs in which one (or some) of the objective function(s) is (are) easier to solve than the others. This is because MFF prefers weakly dominated solutions instead of dominated solutions, and then, we quickly obtain weakly dominated solutions in some extremes of the Pareto front. This causes a loss of diversity and also makes difficult to solve MOPs with many objective functions.

2. Although MFF penalizes clustering between nondominated solutions, there are some cases in which we cannot decide which solution is better to choose in order to improve the distribution along of the nondominated front. In those cases, we also lose diversity in the population.

In this thesis, we propose to use the following constraint to address the first disadvantage: Any individual that we want to select must not be similar (in objective space) to another (selected) individual. With respect to the second disadvantage, we proposed some techniques to improve the diversity in the population, which gave rise to the following MOEAs:

- Maximin-Clustering Multi-Objective Evolutionary Algorithm (MC-MOEA)
- Maximin-Distances Multi-Objective Evolutionary Algorithm (MD-MOEA)
- Maximin-Hypervolume Multi-Objective Evolutionary Algorithm (MH-MOEA)
- Maximin-Approximated Hypervolume Multi-Objective Evolutionary Algorithm (MAH-MOEA)

According to our experimental results, the best algorithm is MAH-MOEA because it obtains results with a high quality and it can also be used in MOPs with many objective functions (in this work we tested it with up to ten objective functions). MAH-MOEA is followed by MD-MOEA, in terms of performance. MD-MOEA obtained good results in most problems, but it has difficulties in MOPs with degenerate Pareto fronts. We think that this is due to the fact that the aim of the selection mechanism used by MD-MOEA is to obtain a uniform distribution. Consequently, it is hard for MD-MOEA to converge to a Pareto front with a dimensionality lower than the dimensionality of the MOP. If the time to obtain the approximate Pareto optimal set is an important factor, MD-MOEA is the best option because it obtains competitive results with respect to MAH-MOEA but at a much lower computational cost.

Besides, in this thesis we compared MD-MOEA and MAH-MOEA, with respect to two well-known MOEAs: MOEA/D and SMS-EMOA (in a version that approximates the contribution to I_H). Our experimental results showed that both MD-MOEA and MAH-MOEA outperformed MOEA/D in the two sets of test problems adopted (DTLZ and WFG) and MAH-MOEA outperformed SMS-EMOA in the DTLZ test problems. With respect to the running time, both algorithms (MD-MOEA and MAH-MOEA) are efficient because MAH-MOEA is much faster than SMS-EMOA and it also obtained good results (it outperformed SMS-EMOA in the DTLZ test problems) and MD-MOEA is not much slower than MOEA/D, while obtaining better results. Therefore, we can say that MD-MOEA and MAH-MOEA are a good option to solve MOPs with low and high dimensionality because they obtain approximations of the Pareto optimal set with a high quality and the computational cost of both MOEAs is affordable. Additionally, the running time of MD-MOEA is quite good. Also, these MOEAs do not need additional information such as MOEA/D that requires a set of well-distributed convex weights.

Another interesting feature of MOEAs based on MFF is that they can be used to solve MOPs in an interactive way when the decision maker defines his/her preferences. For example, a MOEA based on MFF can present at each generation a set of non-dominated solutions to the user and then he/she chooses the solutions which will be considered when calculating the maximin fitness of each indidividual.

At this point, let us return to the question that was formulated at the beginning of this work: *Is it possible to propose a new selection scheme for Multi-Objective Evolutionary Algorithms (MOEAs) which has the following features? It requires little additional information, it is computationally efficient, it has a good performance in terms of the two following aspects: (i) It allows to find solutions that are, as close as possible, to the true Pareto front and, (ii) it allows to produce solutions that are spread along the Pareto front as uniformly as possible. And, it is scalable with respect to the number of objective functions.* We can conclude that the answer to this question is YES. As part of the work reported in this thesis, we were able to design six new MOEAs: iSMS-EMOA, aviSMS-EMOA, MC-MOEA, MD-MOEA, MH-MOEA and MAH-MOEA. None of them requires additional information. Although iSMS-EMOA and MH-MOEA reduce the running time required by the well-known SMS-EMOA, they are impractical to solve MOPs with more than six objective functions. However, aviSMS-EMOA, MC-MOEA, MD-MOEA and MAH-MOEA are computationally efficient and they are scalable with respect to the number of objective functions. Moreover, all of them are competitive with respect to SMS-EMOA and MOEA/D (two well-known MOEAs of the state of the art), and, in some cases, they are better than these two well-known MOEAs.

7.2 Future Work

As part of our future work, we plan to study other techniques to approximate the contribution of the hypervolume with the aim of reducing even more the running time of our proposed scheme, as well as its approximation error. Also, we plan to design a version of aviSMS-EMOA which is able to use large population sizes. The idea is to start the search using a small population size and to increase it over time. In this way, we should be able to obtain more accurate knowledge about the Pareto front in many-objective problems but saving both evaluations of the objective functions and calculations of the contributions to I_H.

Also, we want to continue studying and designing new selection mechanisms based on performance indicators. For example:

- MOEAs based on the generational distance indicator (I_{GD}). Our idea is to use I_{GD} as a convergence strategy and to use some of the techniques proposed in this thesis to improve the diversity in the population.

- MOEAs based on the Δ_p-indicator. Our idea is to design a technique based on ϵ-dominance to generate the reference set that we require to calculate the Δ_p-indicator.

- MOEAs based on the $R2$-indicator. Since the $R2$-indicator also has a locality property, we want to design a new selection scheme that exploits such property.

Appendix A

Test Problems

A.1 Deb-Thiele-Laumanns-Zitzler Test Suite

The Deb-Thiele-Laumanns-Zitzler (DTLZ) Test Suite [30] contains nine test problems, which are scalable to any number of decision variables and objectives. In this thesis, we use the first seven test problems from this test suite. Table A.1 shows a summary of the features of these test problems which were studied in [48] and we present their definitions in the following. It is important to mention that Huband et al. proved that in the case of DTLZ5 and DTLZ6 with four or more objectives it is untrue that their Pareto optimal fronts are meant to be an arc embedded in a M-objective space.

MOP	Separability	Modality	Geometry
DTLZ1	separable	multimodal	linear
DTLZ2	separable	unimodal	concave
DTLZ3	separable	multimodal	concave
DTLZ4	separable	unimodal	concave
DTLZ5	N/A	unimodal	degenerate
DTLZ6	N/A	unimodal	degenerate
DTLZ7	separable	unimodal	disconnected

Table A.1: Features of the first seven DTLZ test problems. An objective function is **separable** if it can be optimized by considering each parameter in turn, independently of one another, and the resultant set of globally optimal parameter vectors is the cross-product of the optimal sets for each individually optimized parameter. In the multi-objective sense, this means that the ideal points for separable objectives can be determined considering only one parameter at a time. N/A means that the information is not available. An objective function is **multimodal** when it has multiple local optima and it is **unimodal** when it has a single optimum. We consider that a problem is multimodal if it has at least one multimodal objective function.

DTLZ1

This test problem has a linear Pareto front, it is separable and multimodal. DTLZ1 consists in minimizing:

$$\begin{aligned}
f_1(\vec{x}) &= \frac{1}{2}x_1x_2\cdots x_{M-1}[1+g(\vec{x}_M)]\\
f_2(\vec{x}) &= \frac{1}{2}x_1x_2\cdots(1-x_{M-1})[1+g(\vec{x}_M)]\\
&\vdots\\
f_{M-1}(\vec{x}) &= \frac{1}{2}x_1(1-x_2)[1+g(\vec{x}_M)]\\
f_M(\vec{x}) &= \frac{1}{2}(1-x_1)[1+g(\vec{x}_M)]
\end{aligned}$$

such as:

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

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M+k-1$ is the number of decision variables. $x_i \in [0,1]$ for all $i = 1,2,\cdots,n$. This problem has $11^k - 1$ local Pareto fronts. A value of $k=5$ is suggested and the problem can be made more difficult by using a larger k. The Pareto optimal solutions correspond to $\vec{x}_M = \vec{0.5}$ and all objective function values lie on the linear hyperplane: $\sum_{i=1}^{M} f_i = 0.5$. Figure A.1 shows the Pareto optimal front of this test problem with three objective functions.

DTLZ2

This test problem has a concave Pareto front, it is separable and unimodal. DTLZ2 consists in minimizing:

$$
\begin{aligned}
f_1(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}\right)\cos\left(\frac{\pi}{2}x_{M-1}\right)\\
f_2(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}\right)\sin\left(\frac{\pi}{2}x_{M-1}\right)\\
f_3(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cdots\sin\left(\frac{\pi}{2}x_{M-2}\right)\\
&\vdots\\
f_{M-1}(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right)\\
f_M(\vec{x}) &= [1+g(\vec{x}_M)]\sin\left(\frac{\pi}{2}x_1\right)
\end{aligned}
$$

such as:

$$g(\vec{x}) = \sum_{x_i\in\vec{x}_M}(x_i-0.5)^2$$

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M + k - 1$ is the number of decision variables. $x_i \in [0, 1]$ for all $i = 1, 2, \cdots, n$. It is recommended to use $k = 10$. The Pareto optimal solutions correspond to $\vec{x}_M = \vec{0.5}$ and all objective function values must satisfy the equation $\sum_{i=1}^{M}(f_i)^2 = 1$. Figure A.2 shows the Pareto optimal front of this test problem with three objective functions.

DTLZ3

This test problem is defined in the same way as DTLZ1 except for a new g function that makes it multimodal. DTLZ3 consists in minimizing:

$$\begin{aligned}
f_1(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}\right)\cos\left(\frac{\pi}{2}x_{M-1}\right)\\
f_2(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}\right)\sin\left(\frac{\pi}{2}x_{M-1}\right)\\
f_3(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cdots\sin\left(\frac{\pi}{2}x_{M-2}\right)\\
&\vdots\\
f_{M-1}(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right)\\
f_M(\vec{x}) &= [1+g(\vec{x}_M)]\sin\left(\frac{\pi}{2}x_1\right)
\end{aligned}$$

such as:

$$g(\vec{x}) = 100\left\{k + \sum_{x_i\in\vec{x}_M}(x_i-0.5)^2 - \cos\left(20\pi(x_i-0.5)\right)\right\}$$

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M + k - 1$ is the number of decision variables. $x_i \in [0,1]$ for all $i = 1, 2, \cdots, n$. It is recommended to use $k = 10$. DTLZ3 has $3^k - 1$ local Pareto fronts. All local Pareto fronts are parallel to the global Pareto optimal front and a MOEA can get stuck at any of these local Pareto optimal fronts, before converging to the global Pareto optimal front at $g^* = 0$. The global Pareto optimal front corresponds to $\vec{x}_M = \vec{0.5}$. The next local Pareto optimal front is at $g = 1$. Figure A.3 shows the Pareto optimal front of this test problem with three objective functions.

DTLZ4

This test problem has a concave Pareto front, it is separable and unimodal. It is a variation of DTLZ2 with a modified meta-variable mapping $\vec{x} \mapsto \vec{x}^{\alpha}$ ($\alpha > 0$). This tests a MOEA's ability to maintain a good distribution of solutions. DTLZ4 consists in minimizing:

$$\begin{aligned}
f_1(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1^{\alpha}\right)\cos\left(\frac{\pi}{2}x_2^{\alpha}\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}^{\alpha}\right)\cos\left(\frac{\pi}{2}x_{M-1}^{\alpha}\right)\\
f_2(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1^{\alpha}\right)\cos\left(\frac{\pi}{2}x_2^{\alpha}\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}^{\alpha}\right)\sin\left(\frac{\pi}{2}x_{M-1}^{\alpha}\right)\\
f_3(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1^{\alpha}\right)\cos\left(\frac{\pi}{2}x_2^{\alpha}\right)\cdots\sin\left(\frac{\pi}{2}x_{M-2}^{\alpha}\right)\\
&\vdots\\
f_{M-1}(\vec{x}) &= [1+g(\vec{x}_M)]\cos\left(\frac{\pi}{2}x_1^{\alpha}\right)\sin\left(\frac{\pi}{2}x_2^{\alpha}\right)\\
f_M(\vec{x}) &= [1+g(\vec{x}_M)]\sin\left(\frac{\pi}{2}x_1^{\alpha}\right)
\end{aligned}$$

such as:

$$g(\vec{x}) = \sum_{x_i \in \vec{x}_M} (x_i - 0.5)^2$$

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M + k - 1$ is the number of decision variables. $x_i \in [0,1]$ for all $i = 1, 2, \cdots, n$. It is recommended to use $k = 10$ and $\alpha = 100$. Figure A.4 shows the Pareto optimal front of this test problem with three objective functions.

DTLZ5

This test problem is unimodal and it has a degenerate Pareto front. DTLZ5 consists in minimizing:

$$\begin{aligned}
f_1(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\cos(\theta_2)\cdots\cos(\theta_{M-2})\cos(\theta_{M-1})\\
f_2(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\cos(\theta_2)\cdots\cos(\theta_{M-2})\sin(\theta_{M-1})\\
f_3(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\cos(\theta_2)\cdots\sin(\theta_{M-2})\\
&\vdots\\
f_{M-1}(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\sin(\theta_2)\\
f_M(\vec{x}) &= [1+g(\vec{x}_M)]\sin(\theta_1)
\end{aligned}$$

such as:

$$\begin{aligned}
\theta_1 &= \frac{\pi}{2}x_1\\
\theta_i &= \frac{\pi}{4(1+g(\vec{x}_M))}(1+2g(\vec{x}_M))x_i, \forall i \in \{2,3,\ldots,M-1\}\\
g(\vec{x}) &= \sum_{x_i\in\vec{x}_M}(x_i-0.5)^2
\end{aligned}$$

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M + k - 1$ is the number of decision variables. $x_i \in [0, 1]$ for all $i = 1, 2, \cdots, n$. $k = 10$ is suggested. The Pareto optimal solutions correspond to $\vec{x}_M = \vec{0.5}$ and all objective function values must satisfy $\sum_{i=1}^{M}(f_i)^2 = 1$. Figure A.5 shows the Pareto optimal front of this test problem with three objective functions.

DTLZ6

The previous test problem can be made harder by making a similar modification to the g function in DTLZ5, as done in DTLZ3. DTLZ6 consists in minimizing:

$$\begin{aligned}
f_1(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\cos(\theta_2)\cdots\cos(\theta_{M-2})\cos(\theta_{M-1}) \\
f_2(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\cos(\theta_2)\cdots\cos(\theta_{M-2})\sin(\theta_{M-1}) \\
f_3(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\cos(\theta_2)\cdots\sin(\theta_{M-2}) \\
&\vdots \\
f_{M-1}(\vec{x}) &= [1+g(\vec{x}_M)]\cos(\theta_1)\sin(\theta_2) \\
f_M(\vec{x}) &= [1+g(\vec{x}_M)]\sin(\theta_1)
\end{aligned}$$

such as:

$$\begin{aligned}
\theta_1 &= \frac{\pi}{2}x_1 \\
\theta_i &= \frac{\pi}{4(1+g(\vec{x}_M))}(1+2g(\vec{x}_M))x_i, \forall i \in \{2,3,\ldots,M-1\} \\
g(\vec{x}) &= \sum_{x_i \in \vec{x}_M} x_i^{0.1}
\end{aligned}$$

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M + k - 1$ is the number of decision variables. $x_i \in [0,1]$ for all $i = 1, 2, \cdots, n$. $k = 10$ is suggested. The Pareto optimal solutions correspond to $\vec{x}_M = \vec{0.5}$ and all objective function values must satisfy $\sum_{i=1}^{M}(f_i)^2 = 1$. Figure A.6 shows the Pareto optimal front of this test problem with three objective functions.

DTLZ7

This test problem has a disconnected set of 2^{M-1} Pareto optimal regions in the search space. DTLZ7 consists in minimizing:

$$\begin{aligned}
f_1(\vec{x}) &= x_1 \\
f_2(\vec{x}) &= x_2 \\
&\vdots \\
f_{M-1}(\vec{x}) &= x_{M-1} \\
f_M(\vec{x}) &= \Big(1 + g(\vec{x}_M)\Big)\, h(f_1, f_2, \cdots, f_{M-1}, g)
\end{aligned}$$

such as:

$$\begin{aligned}
g(\vec{x}_M) &= 1 + \frac{9}{k} \sum_{x_i \in \vec{x}_M} x_i \\
h(f_1, f_2, \cdots, f_{M-1}, g) &= \left(M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1 + g(\vec{x}_M)} \left(1 + \sin(3\pi f_i)\right)\right]\right)
\end{aligned}$$

where M is the number of objective functions, $\vec{x}_M$ represents the last k variables of the decision vector $\vec{x}$, $n = M + k - 1$ is the number of decision variables. $x_i \in [0, 1]$ for all $i = 1, 2, \cdots, n$. A value of $k = 20$ is suggested. The Pareto optimal solutions correspond to $\vec{x}_M = \vec{0}$. Figure A.7 shows the Pareto optimal front of this test problem with three objective functions.

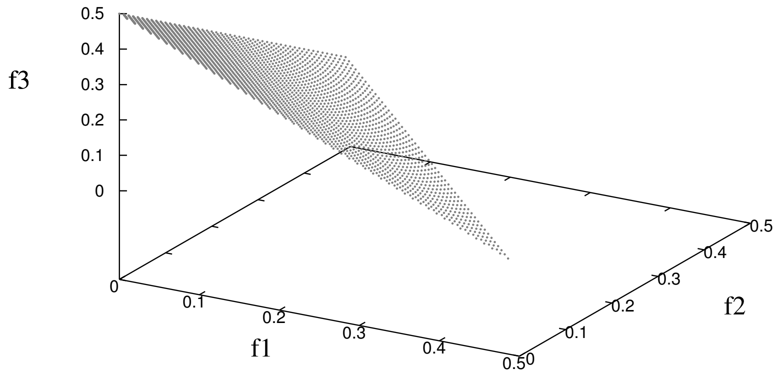


Figure A.1: Pareto optimal front for the DTLZ1 test problem with three objective functions.

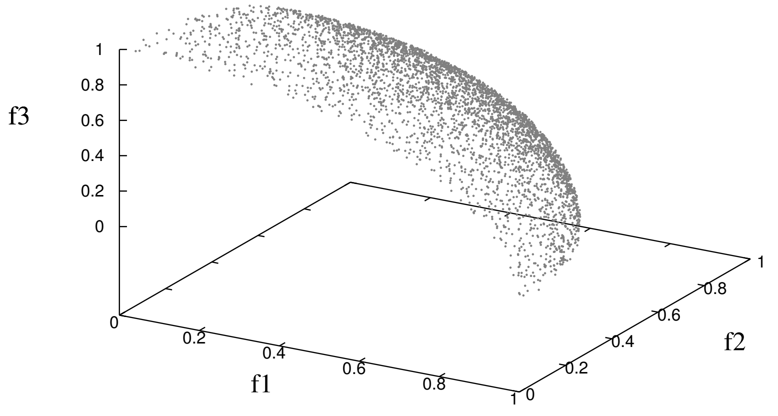


Figure A.2: Pareto optimal front for the DTLZ2 test problem with three objective functions.

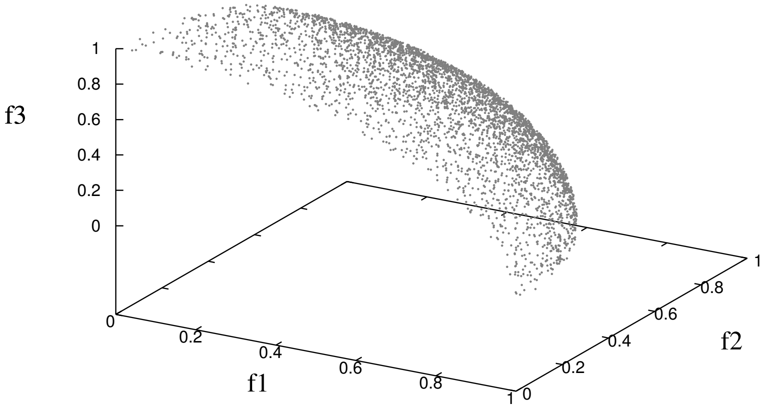


Figure A.3: Pareto optimal front for the DTLZ3 test problem with three objective functions.

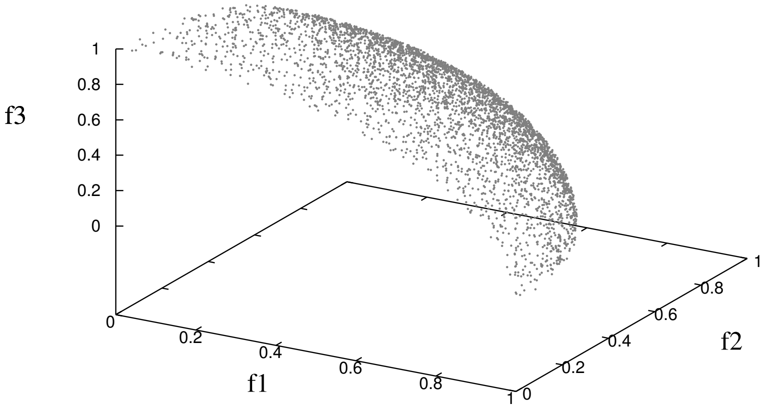


Figure A.4: Pareto optimal front for the DTLZ4 test problem with three objective functions.

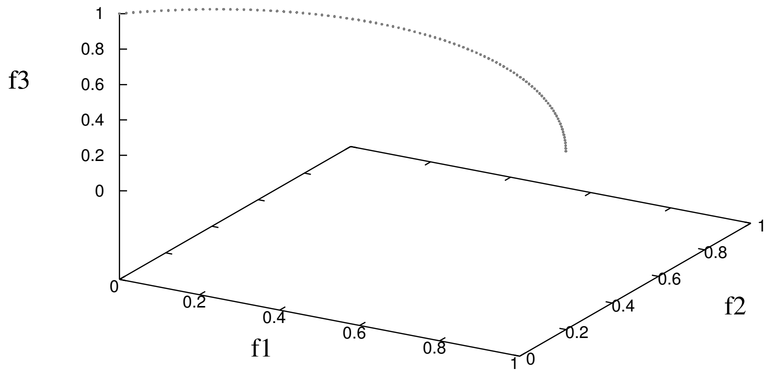


Figure A.5: Pareto optimal front for the DTLZ5 test problem with three objective functions.

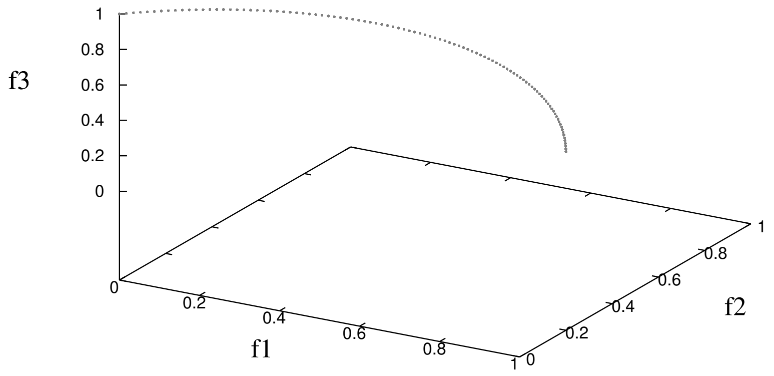


Figure A.6: Pareto optimal front for the DTLZ6 test problem with three objective functions.

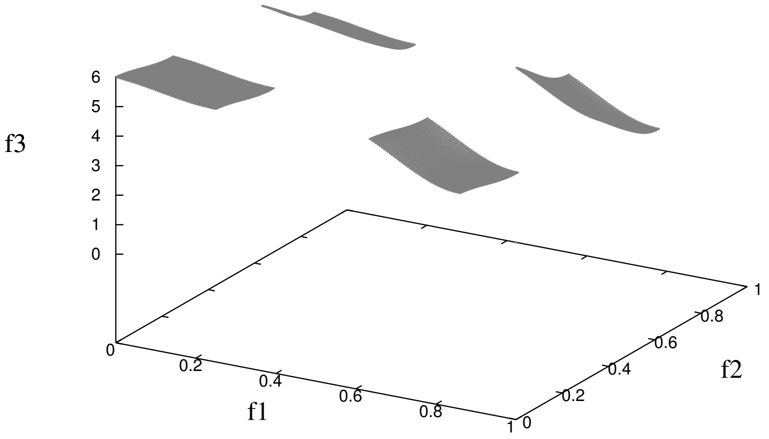


Figure A.7: Pareto optimal front for the DTLZ7 test problem with three objective functions.

A.2 Walking-Fish-Group Test Suite

The Walking-Fish-Group test suite [48] suggests nine multi-objective test problems which are scalable with respect to both objective functions and decision variables. In this thesis, we used the first seven test problems. Table A.2 shows a summary of the features of these seven test problems which were studied in [48].

MOP	Separability	Modality	Geometry
WFG1	separable	unimodal	convex, mixed
WFG2	nonseparable	multimodal	convex, disconnected
WFG3	nonseparable	unimodal	linear, degenerate
WFG4	separable	multimodal	concave
WFG5	separable	deceptive	concave
WFG6	nonseparable	unimodal	concave
WFG7	separable	unimodal	concave

Table A.2: Features of the first seven WFG test problems. An objective function is **separable** if it can be optimized by considering each parameter in turn, independently of one another, and the resultant set of globally optimal parameter vectors is the cross-product of the optimal sets for each individually optimized parameter. In the multi-objective sense, this means that the ideal points for separable objectives can be determined considering only one parameter at a time. An objective function is **multimodal** when it has multiple local optima and it is **unimodal** when it has a single optimum. We consider that a problem is multimodal if it has at least one multimodal objective function. A **deceptive** objective function has a special kind of multimodality (it must have at least two optima: a true optimum and a deceptive optimum). We consider that a problem is deceptive if it has at least one deceptive objective function.

The WFG test problems apply a set of sequential transformations to the vector of decision variables. Each transformation adds a feature to the MOP. The transformations used in the WFG test problems to define the shape of the Pareto front are the following:

$$linear_1(x_1, \cdots, x_{M-1}) = \prod_{i=1}^{M-1} x_i$$

$$linear_{m=2:M-1}(x_1, \cdots, x_{M-1}) = \left(\prod_{i=1}^{M-m} x_i \right) (1 - x_{M-m-1})$$

$$linear_M(x_1, \cdots, x_{M-1}) = 1 - x_1$$

$$convex_1(x_1, \cdots, x_{M-1}) = \prod_{i=1}^{M-1} \left(1 - \cos\left(\frac{\pi}{2} x_i \right) \right)$$

$$convex_{m=2:M-1}(x_1, \cdots, x_{M-1}) = \left(\prod_{i=1}^{M-m} \left(1 - \cos\left(\frac{\pi}{2} x_i \right) \right) \right) \left(1 - \sin\left(\frac{\pi}{2} x_{M-m+1} \right) \right)$$

$$convex_M(x_1, \cdots, x_{M-1}) = 1 - \sin\left(\frac{\pi}{2} x_1 \right)$$

$$concave_1(x_1, \cdots, x_{M-1}) = \prod_{i=1}^{M-1} \sin\left(\frac{\pi}{2} x_i \right)$$

$$concave_{m=2:M-1}(x_1, \cdots, x_{M-1}) = \left(\prod_{i=1}^{M-m} \sin\left(\frac{\pi}{2} x_i \right) \right) \cos\left(\frac{\pi}{2} x_{M-m+1} \right)$$

$$concave_M(x_1, \cdots, x_{M-1}) = \cos\left(\frac{\pi}{2} x_1 \right)$$

$$mixed_M(x_1, \cdots, x_{M-1}) = \left(1 - x_1 - \frac{\cos(2A\pi x_1 + \pi/2)}{2A\pi} \right)^{\alpha}$$

$$disc_M(x_1, \cdots, x_{M-1}) = 1 - x_1^{\alpha} \cos^2(A x_1^{\beta} \pi)$$

It is possible to add more features to increase the difficulty to of a problem. These features are the following:

$$
\begin{aligned}
&b\_poly(y,\alpha) = y^{\alpha}\\
&b\_flat(y,A,B,C) = A + \min(0, \lfloor y - B \rfloor)\frac{A(B-y)}{B}\\
&-\min(0, \lfloor C - y \rfloor)\frac{(1-A)(y-C)}{1-C}\\
&b\_param(y,u(y'),A,B,C) = y^{B+(C-B)(A-(1-2u(y'))|\lfloor 0.5-u(y')+A|)}\\
&s\_linear(y,A) = \frac{|y-A|}{|\lfloor A-y \rfloor + A|}\\
&s\_decept(y,A,B,C) = 1 + (|y-A| - B)\\
&\left(\frac{\lfloor y-A+B \rfloor(1-C+\frac{A-B}{B})}{A-B}\right.\\
&\left.+\frac{\lfloor A+B-y \rfloor(1-C+\frac{1-A-B}{B})}{1-A-B} + \frac{1}{B}\right)\\
&s\_multi(y,A,B,C) = \left(1+\cos\left((4A+2)\pi\left(0.5-\frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)\right)\right.\\
&\left.+4B\left(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)^2\right)/(b+2)\\
&r\_sum(\vec{y},\vec{w}) = \frac{\sum\limits_{i=1}^{|y|} w_1 y_i}{\sum\limits_{i=1}^{|y|} w_i}\\
&r\_nonsep(\vec{y},A) = \frac{\sum\limits_{j=1}^{|y|}\left(y_j + \sum\limits_{k=0}^{A-2}|y_j - y_{1+(j+k)mod|y|}|\right)}{\frac{|y|}{A}\lceil \frac{A}{2} \rceil\left(1+2A-2\lceil \frac{A}{2} \rceil\right)}
\end{aligned}
$$

Considering the above transformations, the first seven problems are described as follows.

WFG1

This problem is separable and unimodal, but it has a polynomial and flat region. It is strongly biased toward small values of the variables, which makes it very difficult for some MOEAs. WFG1 consists in minimizing:

$$f_{m=1:M-1}(\vec{x}) = x_M + S_m convex_M\,(x_1, \cdots, x_{M-1})$$
$$f_M(\vec{x}) = x_M + S_m mixed_M\,(x_1, \cdots, x_{M-1}) \text{ with } \alpha = 1 \text{ and } A = 5$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned}
&z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 1,\ S_{m=1:M} = 2m\\
&x_{i=1:M-1} = r\_sum\left([y'_{(i-1)k/(M-1)+1}, \cdots, y'_{ik/(M-1)}],\right.\\
&\ \left.[2((i-1)k/(M-1)+1), \cdots, 2ik/(M-1)]\right)\\
&x_M = r\_sum\left([y'_{k+1}, \cdots, y'_n], [2(k+1), \cdots, 2n]\right)\\
&y'_{i=1:n} = b\_poly\,(y''_i, 0.02)\\
&y''_{i=1:k} = y'''_i\\
&y''_{i=k+1:n} = b\_flat\,(y'''_i, 0.8, 0.75, 0.85)\\
&y'''_{i=1:k} = z_i/(2i)\\
&y'''_{i=k+1:n} = s\_linear\,(z_i/(2i), 0.35)
\end{aligned}$$

Figure A.8 shows the Pareto optimal front of this test problem with three objective functions.

WFG2

This problem is nonseparable and multimodal. It has a disconnected Pareto front. WFG2 consists in minimizing:

$$\begin{gathered} f_{m=1:M-1}(\vec{x}) = x_M + S_m convex_M\left(x_1, \cdots, x_{M-1}\right) \\ f_M(\vec{x}) = x_M + S_m disc_M\left(x_1, \cdots, x_{M-1}\right) \text{ with } \alpha = \beta = 1 \text{ and } A = 5 \end{gathered}$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned} & z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 1,\ S_{m=1:M} = 2m \\ & x_{i=1:M-1} = r\_sum\left([y'_{(i-1)k/(M-1)+1}, \cdots, y'_{ik/(M-1)}], [1, \cdots, 1]\right) \\ & x_M = r\_sum\left([y'_{k+1}, \cdots, y'_{k+l/2}], [1, \cdots, 1]\right) \\ & y'_{i=1:k} = y''_i \\ & y'_{i=k+1:k+l/2} = r\_nonsep\left([y''_{k+2(i-k)-1}, y''_{k+2(i-k)}], 2\right) \\ & y''_{i=1:k} = z_i/(2i) \\ & y''_{i=k+1:n} = s\_linear\left(z_i/(2i), 0.35\right) \end{aligned}$$

Figure A.9 shows the Pareto optimal front of this test problem with three objective functions.

WFG3

This problem is nonseparable and unimodal. It has a linear and degenerated Pareto front. WFG3 consists in minimizing:

$$f_{m=1:M}(\vec{x}) = x_M + S_m linear_M\left(x_1, \cdots, x_{M-1}\right)$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned}
&z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 0,\ S_{m=1:M} = 2m \\
&x_{i=1:M-1} = r\_sum\left([y'_{(i-1)k/(M-1)+1}, \cdots, y'_{ik/(M-1)}], [1, \cdots, 1]\right) \\
&x_M = r\_sum\left([y'_{k+1}, \cdots, y'_{k+l/2}], [1, \cdots, 1]\right) \\
&y'_{i=1:k} = y''_i \\
&y'_{i=k+1:k+l/2} = r\_nonsep\left([y''_{k+2(i-k)-1}, y''_{k+2(i-k)}], 2\right) \\
&y''_{i=1:k} = z_i/(2i) \\
&y''_{i=k+1:n} = s\_linear\left(z_i/(2i), 0.35\right)
\end{aligned}$$

Figure A.10 shows the Pareto optimal front of this test problem with three objective functions.

WFG4

This problem is separable, but highly multimodal. It has a concave Pareto front. WFG4 consists in minimizing:

$$f_{m=1:M}(\vec{x}) = x_M + S_m concave_m\left(x_1, \cdots, x_{M-1}\right)$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned}
&z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 1,\ S_{m=1:M} = 2m \\
&x_{i=1:M-1} = r\_sum\left([y'_{(i-1)k/(M-1)+1}, \cdots, y'_{ik/(M-1)}], [1, \cdots, 1]\right) \\
&x_M = r\_sum\left([y'_{k+1}, \cdots, y'_n], [1, \cdots, 1]\right) \\
&y''_{i=1:n} = s\_multi\left(z_i/(2i), 30, 10, 0.35\right)
\end{aligned}$$

Figure A.11 shows the Pareto optimal front of this test problem with three objective functions.

WFG5

This problem is deceptive and separable. It has a concave Pareto front. WFG5 consists in minimizing:

$$f_{m=1:M}(\vec{x}) = x_M + S_m concave_m\left(x_1, \cdots, x_{M-1}\right)$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned}
&z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 1,\ S_{m=1:M} = 2m\\
&x_{i=1:M-1} = r\_sum\left([y'_{(i-1)k/(M-1)+1}, \cdots, y'_{ik/(M-1)}], [1, \cdots, 1]\right)\\
&x_M = r\_sum\left([y'_{k+1}, \cdots, y'_n], [1, \cdots, 1]\right)\\
&y''_{i=1:n} = s\_decept\left(z_i/(2i), 0.35, 0.001, 0.05\right)
\end{aligned}$$

Figure A.12 shows the Pareto optimal front of this test problem with three objective functions.

WFG6

This problem is nonseparable and unimodal. It has a concave Pareto front. WFG6 consists in minimizing:

$$f_{m=1:M}(\vec{x}) = x_M + S_m concave_m\left(x_1, \cdots, x_{M-1}\right)$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned}
& z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 1,\ S_{m=1:M} = 2m \\
& x_{i=1:M-1} = r\_nonsep([y'_{(i-1)k/(M-1)+1}, \ldots, y'_{ik/(M-1)}], k/(M-1)) \\
& x_M = r\_nonsep([y'_{k+1}, \ldots, y'_n], l) \\
& y'_{i=1:k} = z_i/(2i) \\
& y'_{i=k+1:n} = s\_linear\left(z_i/(2i), 0.35\right)
\end{aligned}$$

Figure A.13 shows the Pareto optimal front of this test problem with three objective functions.

WFG7

This problem is nonseparable and unimodal. It has a concave Pareto front. WFG7 consists in minimizing:

$$f_{m=1:M}(\vec{x}) = x_M + S_m concave_m \left(x_1, \cdots, x_{M-1}\right)$$

given

$$\vec{z} = [z_1, \cdots, z_k, z_{k+1}, \cdots, z_n]$$

where

$$\begin{aligned}
&z_i \in [0, 2i],\ D = 1,\ A_1 = 1,\ A_{2:M-1} = 1,\ S_{m=1:M} = 2m \\
&x_{i=1:M-1} = r\_sum \left([y'_{(i-1)k/(M-1)+1}, \cdots, y'_{ik/(M-1)}], [1, \cdots, 1]\right) \\
&x_M = r\_sum \left([y'_{k+1}, \cdots, y'_n], [1, \cdots, 1]\right) \\
&y'_{i=1:k} = y''_i \\
&y'_{i=k+1:n} = s\_linear \left(y''_i, 0.35\right) \\
&y''_{i=1:k} = b\_param(z_i/(2i), r\_sum([z_{i+1}/(2(i+1)), \ldots, z_n/(2n)], \\
&[1, \ldots, 1]), \frac{0.98}{49.98}, 0.02, 50) \\
&y''_{i=k+1:n} = z_i/(2i)
\end{aligned}$$

Figure A.14 shows the Pareto optimal front of this test problem with three objective functions.

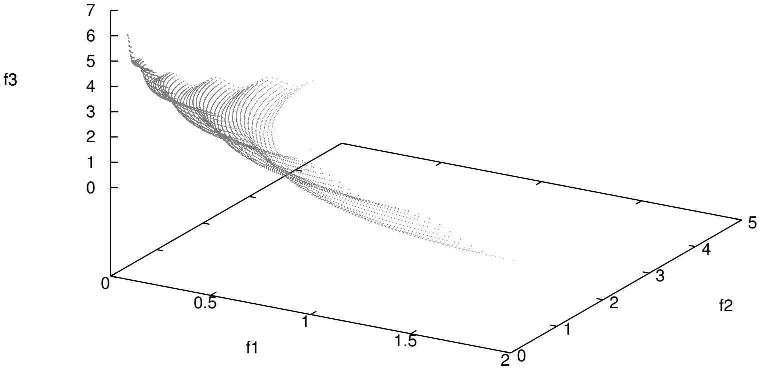


Figure A.8: Pareto optimal front for the WFG1 test problem with three objective functions.

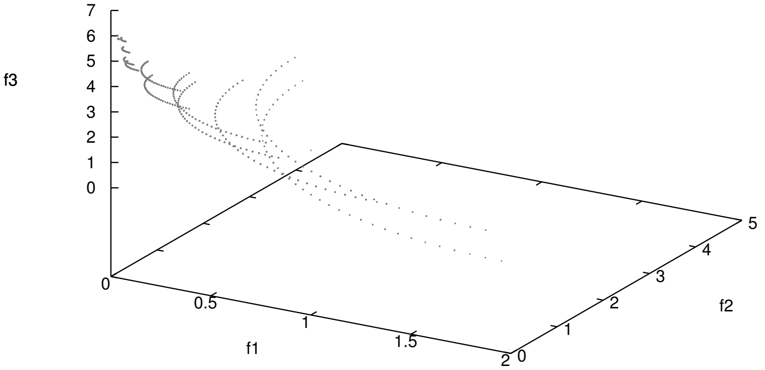


Figure A.9: Pareto optimal front for the WFG2 test problem with three objective functions.

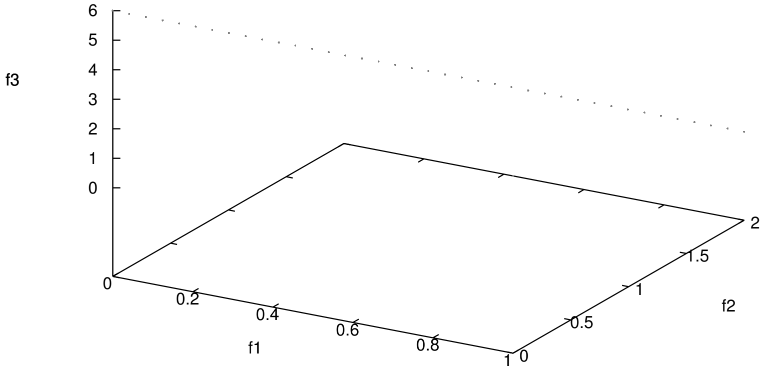


Figure A.10: Pareto optimal front for the WFG3 test problem with three objective functions.

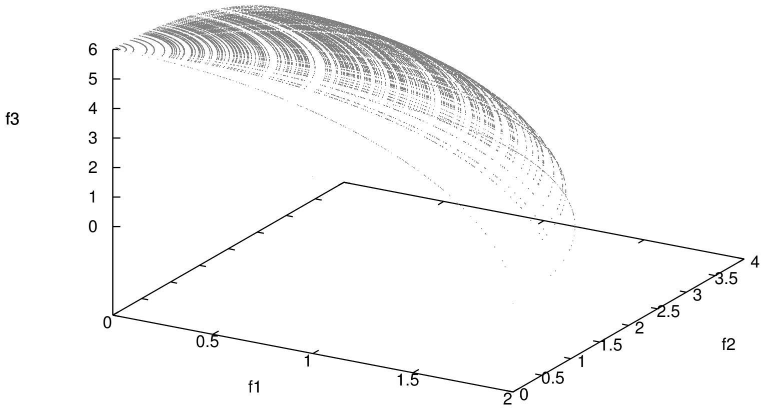


Figure A.11: Pareto optimal front for the WFG4 test problem with three objective functions.

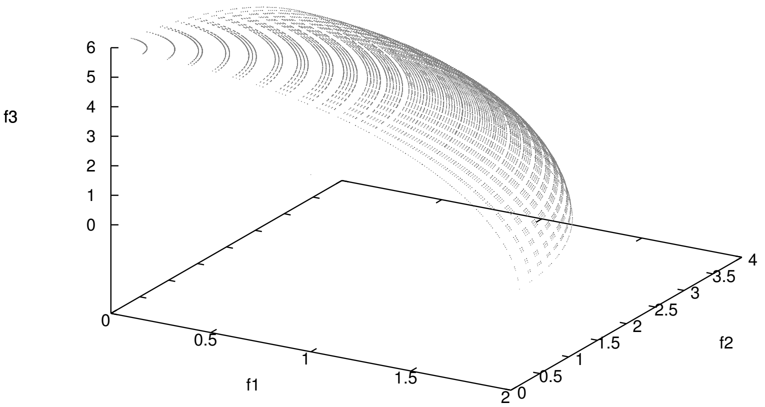


Figure A.12: Pareto optimal front for the WFG5 test problem with three objective functions.

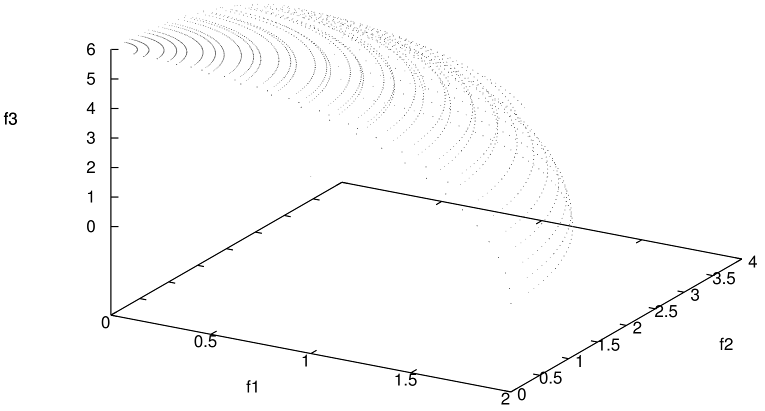


Figure A.13: Pareto optimal front for the WFG6 test problem with three objective functions.

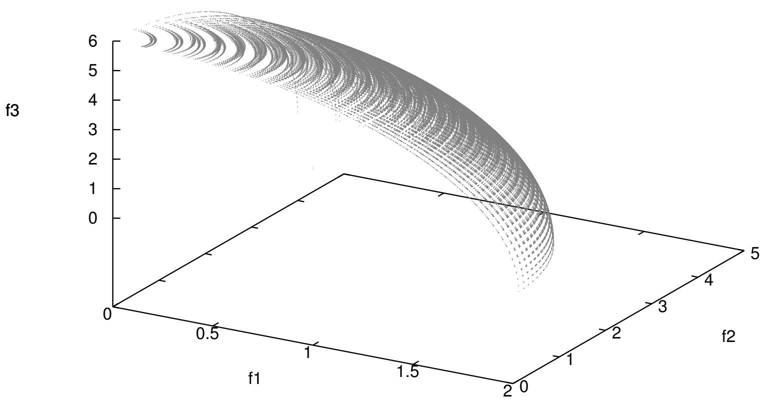


Figure A.14: Pareto optimal front for the WFG7 test problem with three objective functions.

Bibliography

[1] A. Auger, J. Bader, and D. Brockhoff. Theoretically Investigating Optimal μ-Distributions for the Hypervolume Indicator: First Results for Three Objectives. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature–PPSN XI, 11th International Conference, Proceedings, Part I*, pages 586–596. Springer, Lecture Notes in Computer Science Vol. 6238, Kraków, Poland, September 2010.

[2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the Hypervolume Indicator: Optimal $\{\mu\}$-Distributions and the Choice of The Reference Point. In *FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102, Orlando, Florida, USA, January 2009. ACM.

[3] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, Spring, 2011.

[4] R. Balling. Pareto sets in decision-based design. *Journal of Engineering Valuation and Cost Analysis*, 3:189–198, 2000.

[5] R. Balling. The Maximin Fitness Function; Multiobjective City and Regional Planning. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 1–15, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[6] R. Balling and S. Wilson. The Maximin Fitness Function for Multiobjective Evolutionary Computation: Application to City Planning. In

L. S. et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 1079–1084, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[7] P. J. Bentley and J. P. Wakefield. Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. In P. K. Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, Part 5, pages 231–240, London, June 1997. Springer Verlag London Limited. (Presented at the 2nd On-line World Conference on Soft Computing in Design and Manufacturing (WSC2)).

[8] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.

[9] J. Bowman, V.Joseph. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thiriez and S. Zionts, editors, *Multiple Criteria Decision Making*, volume 130 of *Lecture Notes in Economics and Mathematical Systems*, pages 76–86. Springer Berlin Heidelberg, 1976.

[10] L. Bradstreet, L. Barone, and L. While. Updating Exclusive Hypervolume Contributions Cheaply. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 538–544, Trondheim, Norway, May 2009. IEEE Press.

[11] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry*, 43(6–7):601 – 610, 2010.

[12] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*, 425:104–116, March 30 2012.

[13] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do Additional Objectives Make a Problem Harder? In D. Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 765–772, London, UK, July 2007. ACM Press.

[14] D. Brockhoff, T. Wagner, and H. Trautmann. On the Properties of the $R2$ Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 465–472, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.

[15] D. Brockhoff and E. Zitzler. Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 2086–2093, Singapore, September 2007. IEEE Press.

[16] J. Buchanan. A naïve approach for solving mcdm problems: The guess method. *Journal of the Operational Research Society*, 48(2):202–206, 1997.

[17] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley & Sons Inc, New York, December 1961.

[18] A. Charnes, W. W. Cooper, and R. O. Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1(2):138–151, 1955.

[19] C. A. Coello Coello and N. Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2):163–190, June 2005.

[20] C. A. Coello Coello and G. B. Lamont, editors. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, 2004. ISBN 981-256-106-4.

[21] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[22] C. A. Coello Coello and G. Toscano Pulido. A Micro-Genetic Algorithm for Multiobjective Optimization. In E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 126–140. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[23] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.

[24] J. L. Cohon and D. H. Marks. A review and evaluation of multiobjective programing techniques. *Water Resources Research*, 11(2):208–220, 1975.

[25] P. Czyzak and A. Jaszkiewicz. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

[26] V. G. da Fonseca, C. M. Fonseca, and A. O. Hall. Inferential performance assessment of stochastic optimisers and the attainment function. In *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 213–225. Springer-Verlag, 2001.

[27] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, March 1998.

[28] K. Deb and D. E. Goldberg. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 42–50, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

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

[30] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[31] F. di Pierro, S.-T. Khu, and D. A. Savić. An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, February 2007.

[32] A. Díaz-Manríquez, G. Toscano-Pulido, C. A. Coello Coello, and R. Landa-Becerra. A Ranking Method Based on the $R2$ Indicator for Many-Objective Optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 1523–1530, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

[33] C. Domínguez-Medina, G. Rudolph, O. Schüetze, and H. Trautmann. Evenly Spaced Pareto Fronts of Quad-objective Problems using PSA Partitioning Technique. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 3190–3197, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

[34] F. Y. Edgeworth. *Mathematical Psychics.* P. Keagan, London, England, 1881.

[35] M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 62–76, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[36] M. T. Emmerich and C. M. Fonseca. Computing Hypervolume Contributions in Low Dimensions: Asymptotically Optimal Algorithm and Complexity Results. In R. H. Takahashi, K. Deb, E. F. Wanner, and S. Grecco, editors, *Evolutionary Multi-Criterion Optimization, 6th International Conference, EMO 2011*, pages 121–135, Ouro Preto, Brazil, April 2011. Springer. Lecture Notes in Computer Science Vol. 6576.

[37] M. Farina. A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 956–961, Piscataway, New Jersey, May 2002. IEEE Service Center.

[38] M. Farina and P. Amato. On the Optimal Solution Definition for Many-criteria Optimization Problems. In *Proceedings of the NAFIPS-FLINT International Conference'2002*, pages 233–238, Piscataway, New Jersey, June 2002. IEEE Service Center.

[39] C. M. Fonseca and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[40] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1-2):39–45, 1955.

[41] K. Gerstl, G. Rudolph, O. Schütze, and H. Trautmann. Finding Evenly Spaced Fronts for Multiobjective Control via Averaging Hausdorff-Measure. In *The 2011 8th International Conference on Electrical Engineering, Computer Science and Automatic Control (CCE'2011)*, pages 975–980, Mérida, Yucatán, México, October 2011. IEEE Press.

[42] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[43] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 41–49, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[44] M. P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report Technical Report IMM-REP-1998-7, Technical University of Denmark, March 1998.

[45] R. Hernández Gómez and C. A. Coello Coello. MOMBI: A New Metaheuristic for Many-Objective Optimization Based on the $R2$ Indicator. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 2488–2495, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

[46] J. Horn and N. Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.

[47] J. Horn, N. Nafpliotis, and D. E. Goldberg. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center.

[48] S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.

[49] S. Huband, P. Hingston, L. White, and L. Barone. An Evolution Strategy with Probabilistic Mutation for Multi-Objective Optimisation. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 3, pages 2284–2291, Canberra, Australia, December 2003. IEEE Press.

[50] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, Spring 2007.

[51] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima. Hypervolume Approximation Using Achievement Scalarizing Functions for Evolutionary Many-Objective Optimization. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*, pages 530–537, Trondheim, Norway, May 2009. IEEE Press.

[52] J. Knowles and D. Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.

[53] J. Knowles, L. Thiele, and E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, feb 2006. revised version.

[54] J. D. Knowles and D. W. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In *1999 Congress on Evolutionary Computation*, pages 98–105, Washington, D.C., July 1999. IEEE Service Center.

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

[56] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.

[57] H. Li, X. Huang, and Q. Feng. Optimizing expressway maintenance planning by coupling ant algorithm and geography information system transportation in hubei province, china. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 2977 –2979, july 2011.

[58] X. Li. Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function. In Kalyanmoy Deb et al., editor, *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I*, pages 117–128, Seattle, Washington, USA, June 2004. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.

[59] X. Li, J. Branke, and M. Kirley. On Performance Metrics and Particle Swarm Methods for Dynamic Multiobjective Optimization Problems. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 576–583, Singapore, September 2007. IEEE Press.

[60] A. López Jaimes, C. A. Coello Coello, and D. Chakraborty. Objective Reduction Using a Feature Selection Technique. In *2008 Genetic and Evolutionary Computation Conference (GECCO'2008)*, pages 674–680, Atlanta, USA, July 2008. ACM Press. ISBN 978-1-60558-131-6.

[61] A. López Jaimes, L. V. Santana Quintero, and C. A. Coello Coello. Ranking methods in many-objective evolutionary algorithms. In R. Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, pages 413–434. Springer, Berlin, 2009. ISBN 978-3-642-00266-3.

[62] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[63] S. Mostaghim, J. Branke, and H. Schmeck. Multi-Objective Particle Swarm Optimization on Computer Grids. In D. Thierens, editor, *2007*

Genetic and Evolutionary Computation Conference (GECCO'2007), volume 1, pages 869–875, London, UK, July 2007. ACM Press.

[64] K. Nowak, M. Märtens, and D. Izzo. Empirical Performance of the Approximation of the Least Hypervolume Contributor. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature - PPSN XIII, 13th International Conference*, pages 662–671. Springer. Lecture Notes in Computer Science Vol. 8672, Ljubljana, Slovenia, September 13-17 2014.

[65] V. Pareto. *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne, 1896.

[66] D. H. Phan and J. Suzuki. R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 1836–1845, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.

[67] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin, 2005. ISBN 3-540-20950-6.

[68] C. A. Rodríguez Villalobos and C. A. Coello Coello. A New Multi-Objective Evolutionary Algorithm Based on a Performance Assessment Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 505–512, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.

[69] R. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, Department of Communication Sciences, University of Michigan, Ann Arbor, Michigan, USA, June 1967.

[70] H. Sato, H. E. Aguirre, and K. Tanaka. Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 5–20, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[71] J. D. Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, Tennessee, USA, 1984.

[72] J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, May 1995.

[73] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, August 2012.

[74] N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[75] R. Steuer. The tchebycheff procedure of interactive multiple objective programming. In B. Karpak and S. Zionts, editors, *Multiple Criteria Decision Making and Risk Analysis Using Microcomputers*, volume 56 of *NATO ASI Series*, pages 235–249. Springer Berlin Heidelberg, 1989.

[76] A. Sülflow, N. Drechsler, and R. Drechsler. Robust Multi-objective Optimization in High Dimensional Spaces. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 715–726, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[77] H. Trautmann, G. Rudolph, C. Dominguez-Medina, and O. Schütze. Finding Evenly Spaced Pareto Fronts for Three-Objective Optimization Problems. In O. Schütze, C. A. Coello Coello, A.-A. Tantar, E. Tantar, P. Bouvry, P. Del Moral, and P. Legrand, editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, pages 89–105. Springer, Advances in Intelligent Systems and Computing Vol. 175, Berlin, Germany, 2012. ISBN 978-3-642-31519-0.

[78] H. Trautmann, T. Wagner, and D. Brockhoff. R2-EMOA: Focused Multiobjective Search Using R2-Indicator-Based Selection. In G. Nicosia

and P. Pardalos, editors, *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 70–74. Springer Berlin Heidelberg, 2013.

[79] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Payton, Ohio, USA, May 1999.

[80] D. A. Van Veldhuizen and G. B. Lamont. Evolutionary Computation and Convergence to a Pareto Front. In J. R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, Stanford University, California, July 1998. Stanford University Bookstore.

[81] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective Evolutionary Algorithm Test Suites. In J. Carroll, H. Haddad, D. Oppenheim, B. Bryant, and G. B. Lamont, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, San Antonio, Texas, 1999. ACM.

[82] D. A. Van Veldhuizen and G. B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.

[83] T. Wagner, H. Trautmann, and D. Brockhoff. Preference Articulation by Means of the $R2$ Indicator. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, editors, *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, pages 81–95. Springer. Lecture Notes in Computer Science Vol. 7811, Sheffield, UK, March 19-22 2013.

[84] Yan-Yan Tan and Yong-Chang Jiao and Hong Li and Xin-Kuan Wang. MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & Operations Research*, 40(6):1648 – 1660, 2013.

[85] D. W. Y.Y. Haimes, L.S. Lasdon. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 1:296–297, 1971.

[86] L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, Jan 1963.

[87] S. Zapotecas Martínez, V. A. Sosa Hernández, H. Aguirre, K. Tanaka, and C. A. Coello Coello. Using a Family of Curves to Approximate the Pareto Front of a Multi-Objective Optimization Problem. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature - PPSN XIII, 13th International Conference*, pages 682–691. Springer. Lecture Notes in Computer Science Vol. 8672, Ljubljana, Slovenia, September 13-17 2014.

[88] M. Zeleny. Compromise programming. In J. Cochrane and M. Zeleny, editors, *Multiple Criteria Decision Making*, pages 262–301. University of South Carolina Press, Columbia, 1973.

[89] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[90] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[91] E. Zitzler and S. Künzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[92] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[93] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Pe-

riaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.

[94] E. Zitzler and L. Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998.

[95] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[96] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

[97] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.