Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Técnicas de Adaptación para Funciones de Escalarización usadas en Algoritmos Evolutivos Basados en Descomposición

Tesis que presenta

Miriam Pescador Rojas

Para obtener el Grado de

Doctora en Ciencias

en Computación

Director de la tesis

Dr. Carlos Artemio Coello Coello

Ciudad de México **Marzo, 2019**

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Zacatenco Campus

Computer Science Department

Adaptation Techniques for Scalarizing Functions used in Decomposition-Based Multi-Objective Evolutionary Algorithms

Submitted by

Miriam Pescador Rojas

in partial fulfillment of the requirement for the degree of

Ph.D. in Computer Science

Advisor

Dr. Carlos Artemio Coello Coello

Mexico City. **March, 2019**

Miriam Pescador Rojas

Contents

List of Tables

List of Figures

List of Algorithms

Resumen

En los últimos años, los algoritmos evolutivos multi-objetivo que adoptan funciones de escalarización han mostrado ser una excelente alternativa para dar solución a problemas de optimización multi-objetivo complejos. Sin embargo, estos algoritmos requieren una adecuada configuración de sus parámetros (en particular de la función de escalarización) para tener un buen desempeño.

Las características de un problema multiobjetivo tales como la geometría del frente de Pareto o el número de funciones objetivo son útiles para determinar la mejor configuración de parámetros.

En esta tesis se presenta una revisión de diversas funciones de escalarización que fueron acopladas a algoritmos evolutivos multi-objetivo del estado del arte tales como: MOEA/D, MOMBI-II y MOEA/D-DRA. Se diseñó una metodología experimental de ajuste de parámetros fuera de línea que fue aplicada a diversos casos de estudio entre los que se incluyeron: 1) analizar el comportamiento de las funciones de escalarización para resolver problemas con geometrías convexas, cóncavas, lineales, mixtas y desconectadas del frente de Pareto, 2) identificar a las funciones de escalarización que son capaces de escalar en el número de objetivos (hasta 10 objetivos) y 3) examinar la velocidad de convergencia de las funciones de escalarización.

Además, se emplearon técnicas de adaptación de parámetros en línea para combinar simultáneamente varias funciones de escalarización con el objetivo de resolver eficientemente una mayor diversidad de problemas multi-objetivo.

Los resultados presentados en esta tesis proporcionan pautas para diseñar nuevos algoritmos evolutivos multi-objetivo que sean robustos de tal forma que puedan adaptarse adecuadamente a diversas características de los problemas multi-objetivo.

Abstract

In the last few years, decomposition-based multi-objective evolutionary algorithms, as well as indicator-based multi-objective evolutionary algorithms that adopt scalarizing functions, have shown to be an excellent choice for solving complex multi-objective optimization problems. However, these algorithms require an appropriate setting of their parameters (in particular of the scalarizing function) for them to have a good performance.

The features of a multi-objective problem such as the geometry of the Pareto front or the number of objectives are useful to determine the best parameters settings.

In this thesis, we present a review of several scalarizing functions which were coupled to state-of-the-art multi-objective evolutionary algorithms such as: MOEA/D, MOMBI-II and MOEA/D-DRA. An experimental methodology for offline parameters setting was designed. Such a methodology was applied to several study cases including the following: 1) analyzing the behavior of the scalarizing functions for solving problems having convex, concave, linear, mixed and disconnected Pareto front geometries, 2) identifying the scalarizing functions which are able to scale up with an increasing number of objectives (up to 10 objectives) and 3) examining the speed of convergence of the scalarizing functions.

Furthermore, online parameters setting techniques were adopted for simultaneously combining several objectives with the aim of efficiently solving a greater variety of multi-objective problems.

The results presented in this thesis provide hints for designing new multi-objective evolutionary algorithms that are robust and can properly adapt to different features of multi-objective problems.

Agradecimientos

En primer lugar, quiero expresar mi mayor agradecimiento a mi asesor de tesis, el Dr. Carlos A. Coello Coello, quien desde mi formación en la maestría ha transmitido sus conocimientos en el área de cómputo evolutivo. Siendo un ejemplo a seguir para la generación de investigación de vanguardia y competitiva a nivel internacional. Gracias por los apoyos, consejos y paciencia otorgados.

Agradezco a los revisores de esta tesis por sus enriquecedores consejos y observaciones, doctores: Antonio López Jaimes, Edgar Emmanuel Vallejo Clemente, Amilcar Meneses Viveros, Gerardo de la Fraga y Cuauhtemoc Mancillas López.

Mi especial agradecimiento a los doctores de otras universidades con quienes tuve la oportunidad de colaborar: Elizabeth Montero, María Cristina Riff, Nicolás Emilio Rojas Morales y Denis Pallez.

Agradezco a los doctores de CINVESTAV Sonia G. Mendoza Chapa, Amilcar Meneses Viveros, José Francisco Rambó Rodrígruez, José Guadalupe Rodríguez García, Juan Carlos Ku Cauich por los consejos brindados. Así como al personal de apoyo al departamento: Sofía Reza Cruz, Felipa Rosas López, Erika B. Ríos Hernández, Arcadio A. Morales Vázquez, Santiago Domínguez Domínguez y José Luis Flores Garcilazo.

Agradezco a mis amigos y compañeros del doctorado: Raquel Hernández Gómez por la orientación, apoyo y amistad que me brindó durante nuestras colaboraciones. También agradezco a Carlos Ignacio Hernández Castellanos, Edgar Manoatl López, Jesús Guillermo Falcón Cardona, Oliver Cuate, Lourdes Uribe, Heriberto Cruz Hernández, Israel Buitrón Dámaso, Carlos Martínez Angeles, Angélica Serrano Rubio y especialmente a Víctor Adrián Sosa Hernández por hacer agradable el tiempo de estudio y dedicación al doctorado.

A los profesores Dr. Salvador Godoy Calderón, Dr. Francisco Hiram Calvo, Dr. Ricardo Barrón Fernández por las recomendaciones hechas en los seminarios de investigación del laboratorio de inteligencia artificial en el CIC-IPN.

Quiero hacer un reconocimiento especial a mi madre por ser mi principal pilar y sostén, quien con muchos sacrificios se esforzó todos los días para darnos estudios y motivarnos para ser personas de provecho. De la misma manera agradezco a mis hermanos Carlos Armando y José Alfredo por sus innumerables enseñanzas y apoyo incondicional. Gracias por ayudarme a culminar las metas académicas, profesionales y personales en cada una de las etapas de mi vida.

Agradezco a las instituciones como CONACyT, CINVESTAV y el IPN-ESCOM por los apoyos económicos recibidos, ya que sin ellos sería imposible estudiar un posgrado y participar en eventos internacionales que han sido enriquecedores para mi formación como investigadora.

Esta tesis fue derivada de los proyectos titulados “Esquemas de Selección Alternativos para Algoritmos Evolutivos Multi-Objetivo” (Ref. 1920, Convocatoria Fronteras de la Ciencia 2016 de CONACyT) y “Nuevos Esquemas de Selección para Algoritmos Evolutivos Multi-Objetivo basados en Indicadores de Desempeño” (No. de solicitud 4, Convocatoria 2018 del Fondo SEP-Cinvestav). El responsable técnico de ambos proyectos es el Dr. Carlos A. Coello Coello.

Chapter 1

Introduction

The optimization process is present in several real-world problems. Its solution requires knowing the best set of decision variables to reach the maximum benefit allowed by the constraints imposed on the problem. When a problem needs to optimize more than one objective function at the same time is called a Multi-objective Optimization Problem (MOP).

In recent years, the use of Multi-Objective Evolutionary Algorithms (MOEAs) has allowed to tackle complex MOPs. The first generation of MOEAs were algorithms that used Pareto dominance in their selection mechanism such as NSGA-II [31] and SPEA [179]. However, these MOEAs do not work properly in many-objective optimization (i.e., with problems having more than three objectives). The two most common approaches to deal with many-objective problems are: (1) indicator-based MOEAs and (2) decomposition-based MOEAs. The former refers to the use of methods to assess the quality of the approximation generated by the MOEA using a performance indicator. The best-known indicator-based MOEA is SMS-EMOA [179] which is based on the hypervolume indicator and is, therefore, computationally expensive in many-objective problems. Meanwhile, decomposition-based MOEAs have several advantages such as scalability to many-objective problems (MaOPs) [61,104], a high search ability for combinatorial optimization [74,75,136], and a high compatibility with local search procedures [97, 117, 167]. In this category, MOEA/D [171] is a computationally efficient algorithm that decomposes a MOP into a set of single-objective problems with neighborhood relationships. A typical MOEA based on decomposition uses a type of scalarizing function whose mathematical properties may be appropriate to solve MOPs with certain characteristics related to certain Pareto front shapes or to the number of objectives.

1.1 Motivation

The choice of most suitable scalarizing function in a decomposition-based MOEA plays an important role in its performance to solve MOPs. Moreover, some scalarizing functions need to define a parameter value to which they may be very sensitive. For example, Penalty Boundary Intersection (PBI) [171] has a penalty parameter to balance convergence and uniformity along the true Pareto front. Several studies [36, 75, 136, 137, 159, 161] have provided a sensitivity analysis in scalarizing functions, indicating that the choice of an appropriate parameter value depends on specific MOP's features such as the Pareto front shape [36, 49], the number of decision variables and the number of objective functions.

Two possibilities for dealing with the parameter setting problem in MOEAs are the offline parameter tuning and the online parameter control strategies. The first one refers to establishing *a priori* a set of parameter values and use them in all the iterations of an MOEA with the aim of reaching the best possible performance. In contrast, the second case includes adaptive mechanisms that modify the parameter values based on the information

gathered during the evolutionary search process. Our interest in this work is to explore both options, using the following guidelines: 1) to apply an offline parameter tuning methodology to gain a better understanding of the behavior and robustness of scalarizing functions in decomposition-based MOEAs to solve a set of test problems, and 2) to explore online adaptive mechanisms during the excution of an MOEA to combine the advantages provided by diverse scalarizing functions.

1.2 Problem Statement

Parameter tuning for configuring a MOEA is a combinatorial optimization problem that consists of choosing from among the set of possible components, assembling them and assigning specific values to their free parameters [11] with the aim of achieving the best possible performance. Formally, the tuning problem can be defined as:

Definition 1.2.1 *Given a metaheuristic $\mathcal{M}$, its categorical parameters $P_c = \{c_1, \ldots, c_k\}$ and numerical parameters $P_n = \{n_1, \ldots, n_m\}$ that respectively belong to the domains $D = \{d_1, \ldots, d_{m+k}\}$, a set of problem instances (I) and a performance measure (q), the goal is to find the values for the elements in P_c and P_n such that these values allow the algorithm to obtain the best aggregated performance measure ($c(q)$).*

Offline parameter tuning is computationally expensive, but it is useful for deriving knowledge about the relation among parameters involved in an MOEA. This procedure can tackle the problem from two perspectives: 1) specialization of algorithms which means that for a given algorithm configuration, the aim is to find a subset of optimization problems on which this algorithm obtains good results, and 2) generalization of algorithms in which the aim is to find an algorithm configuration that solves the largest number of problems with different features.

Meanwhile, online parameter tuning has an acceptable computational cost but requires of mechanisms to handle the trade-off between the exploration and the exploitation procedures. Namely, given a pool of algorithmic configurations, the aim is to make a balance between exploring all posible options or exploiting the most promising configurations previously tested. The aforementioned strategies should be limited by our budget of objective function evaluations. Here, another problem arises: we need to establish a fair comparison between the offline or online parameter setting technique and the baseline algorithm.

We can explore the best configuration of a particular component or set of parameters used by MOEAs. In our case, we focus this thesis on the tuning of scalarizing functions to improve the performance of MOEAs.

1.3 General and Particular Objectives

In the following, we state the objectives of this thesis.

General Objective

To design offline and online tuning techniques for adapting the scalarizing functions used by multi-objective evolutionary algorithms with the goal of solving multi-objective optimization problems with different Pareto front shapes and high dimensionality in the objective space.

Particular Objectives

1. To investigate the most promising scalarizing functions and their properties to solve different Pareto front shapes.
2. To design a methodology of parameter tuning with the aim of selecting the most suitable scalarizing function for solving several classes of MOPs.
3. To analyze the performance of diverse scalarizing functions to solve MOPs with complicated Pareto front shapes and high dimensional objective space.
4. To study the convergence speed of scalarizing functions in different types of Pareto front shapes.
5. To study online adaptation techniques in decomposition-based MOEAs.
6. To design online adaptation strategies for different scalarizing functions.

1.4 Current Contributions

Our current contributions related to this thesis are the following:

Conference publications

- **Miriam Pescador-Rojas** and Carlos A. Coello-Coello, *A novel local search mechanism based on reflected-ray tracing coupled with MOEA/D*. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016. ISBN 978-1-5090-4240-1
- **Miriam Pescador-Rojas**, Raquel Hernández Gómez, Elizabeth Montero, Nicolás Rojas-Morales, María-Cristina Riff, and Carlos A. Coello Coello, *An Overview of Weighted and Unconstrained Scalarizing Functions*. In: Trautmann H. et al. (eds) Evolutionary Multi-Criterion Optimization. EMO 2017. Lecture Notes in Computer Science, vol 10173. Springer, Cham, pp. 499-513. ISBN 978-3-319-54156-3
- **Miriam Pescador-Rojas** and Carlos A. Coello-Coello, *Collaborative and Adaptive Strategies of Different Scalarizing Functions in MOEA/D*, 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, 2018, pp. 1-8.

- **Miriam Pescador-Rojas** and Carlos A. Coello-Coello, *Studying the effect of techniques to generate reference vectors in many-objective optimization.*, In Proceedings of The Genetic and Evolutionary Computation Conference, 2018
- **Miriam Pescador-Rojas** and Carlos A. Coello-Coello, *Studying the Effect of Robustness Measures in Offline Parameter Tuning for Estimating the Performance of MOEA/D*, IEEE Symposium Series on Computational Intelligence, 2018

Journal publications

- **Miriam Pescador-Rojas**, Raquel Hernández Gómez, Elizabeth Montero, Nicolás Rojas-Morales, María-Cristina Riff, and Carlos A. Coello Coello, *Performance improvement in Scalarizing Methods based on the Chebyshev function coupled to Multi-objective Evolutionary Algorithms*, 2019 (to be submitted to the IEEE Transactions on Evolutionary Computation journal)

1.5 Organization of this work

The remainder of this thesis is organized as follows: Chapter 2 introduces some theoretical background on multi-objective optimization. Also, we present a general overview of multi-objective evolutionary algorithms (MOEAs). Chapter 3 provides the mathematical definitions of diverse scalarizing functions and their model parameters. Moreover, we analyze the use of these functions in MOEAs. Chapter 4 explains the definition of an offline and online parameter tuning. Besides, we present the most representative methods used for configuring MOEAs. Chapter 5 presents our proposal of offline parameter tuning methodology which is employed in different case studies involving the performance of scalarizing functions and their scalability in objective space. Chapter 6 describes our proposed approaches about the online adaptive strategies to use simultaneously multiple scalarizing functions and techniques to control their parameter model. Finally, Chapter 7 presents our general conclusions and some possible paths for future work. Appendix A describes the test problems used in our comparative studies. Appendix B presents figures and tables related to Chapter 5 for the case study involving the effect on convergence and distribution of nondominated solutions produced by differente types of scalarizing functions. Appendix C shows plots that illustrate the impact on convergence speed of the scalarizing functions presented in the experiments of Chapter 6.

Chapter 2

Background and Related Work

2.1 Multi-objective Optimization

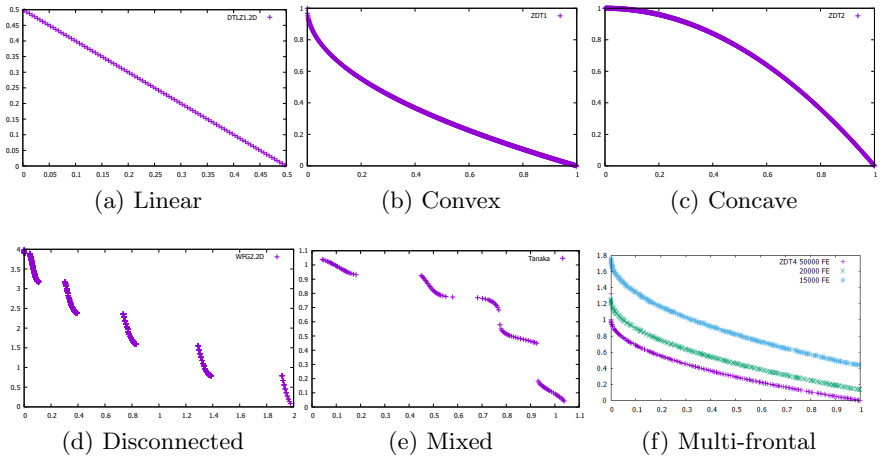
Multi-objective optimization problems (MOPs) are very common in real-world applications, and involve the solution of problems that have two or more (often conflicting) objectives, which we aim to optimize at the same time.

Let us assume that, we have an MOP with m objective functions $(f_i, i = 1, \ldots, m)$ and n decision variables $(x_i, i = 1, \ldots, n)$. The goal of optimization is to minimize simultaneously all objectives. Mathematically, it can be described as follows:

$$\text{minimize } \mathbf{f}(\mathbf{x}) = \big[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\big]^T \text{ subject to } x \in \mathcal{S}$$

where $\mathcal{S}$ is the feasible space of solutions and $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathcal{S}$ is the vector of decision variables. $f_i : \mathbb{R}^n \to \mathcal{Z}^1$, $i \in \{1, \ldots, m\}$ are the objective functions.

Some MOPs have complicated characteristics that cause difficulties for converging to the Pareto optimal solutions. For example, a large number of decision variables (large-scale MOPs) or a large number of objective functions (many-objective problems). Furthermore, there are several Pareto front shapes such as linear, convex, concave, mixed or disconnected geometries (see Fig. 2.1).



(a) Linear (b) Convex (c) Concave

(d) Disconnected (e) Mixed (f) Multi-frontal

Figure 2.1: Examples of Pareto Front shapes

[1] Where Z is the feasible objective region.

2.1.1 Optimality Definitions

Next, we introduce some definitions to describe the concept of optimality in which we are interested.

Pareto optimality. A decision vector $\mathbf{x}^* \in S$ is a Pareto optimal solution if there does not exist another decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all $i = 1, ..., m$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one index j. Equivalently, an objective vector $z^* \in Z$ is a Pareto optimal solution if the decision vector corresponding to it is Pareto optimal.

Weak Pareto optimality. A point $\mathbf{x}^* \in S$, is weak Pareto optimal if there does not exist another point, $x \in S$, such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$. In other words, a point is weakly Pareto optimal if there is no other point that improves all of the objective functions simultaneously.

Pareto dominance.

1. Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, we say that $\mathbf{u}$ dominates $\mathbf{v}$ ($\mathbf{u} \preceq \mathbf{v}$) iff $\mathbf{u}$ is better than $\mathbf{u}$. i.e. $\forall_i \{1, \ldots, m\}$, $u_i \leq v_i \wedge \exists_j \in \{1, \ldots, m\} : u_j < v_j$.

   If we compare two solutions, we have three possibilities: 1) $\mathbf{u}$ dominates $\mathbf{v}$, 2) $\mathbf{u}$ is dominated by $\mathbf{v}$ or 3) $\mathbf{u}$ and $\mathbf{v}$ are incomparable. Figure 2.2 illustrates these three possibilities, in a bi-objective optimization problem.

2. We say that a set of non-dominated solutions (or Pareto optimal solutions) conforms the **Pareto Optimal Set** (POS). Matematically, $\mathcal{POS}^* := \{\mathbf{x} \in \mathcal{S} \mid \nexists \mathbf{y} \in \mathcal{S} : \mathbf{y} \prec \mathbf{x}\}$

3. In objective space, this set is called the **Pareto Optimal Front** (POF). Matematically, $\mathcal{POF}^* := \{\mathbf{F}(\mathbf{x}) \in Z \mid \mathbf{x} \in \mathcal{POS}^*\}$. We thus wish to determine the $\mathcal{POF}$ from the set $\mathcal{POS}$ of all the decision variable vectors that satisfy Equation (2.1).

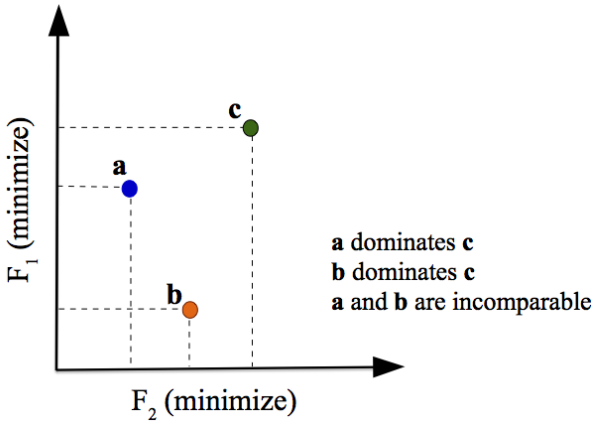


Figure 2.2: A bi-objective problem that illustrates the three possible cases where two solutions are compared according to Pareto dominance.

2.1.2 Reference Points

There exist representative points in the objective space as follows.

Ideal objective vector. Let $\mathbf{z}^* = [z_1^*, \cdots, z_m^*]^T$ be a vector that denotes the optimum value of the *ith* objective function of an MOP. It can be obtained by minimizing each of the objective functions individually. In general, $\mathbf{z}^*$ is unattainable in an MOP where objectives have some degree of conflict among them.

Utopian objective vector. Let $\mathbf{z}^{**} = z_i^* - \epsilon_i$ for all $i = 1, ..., m$, where $\epsilon_i > 0$ is a relatively small but computationally significant scalar.

Nadir objective vector. Let $\mathbf{z}^{nad} \in \mathbb{R}^m$ be a vector defined by the upper bounds for the objective function values of the Pareto optimal solutions.

During the iterative procedure of a Multi-Objective Evolutionary Algorithm (MOEA), an estimation of the ideal and Nadir vectors can be computed by means of the minimum and maximum objective values from the current set of Pareto optimal solutions. This procedure may not obtain an exact computation of the ideal and Nadir objective vectors, mainly because of setting inappropriate values by the influence of weak Pareto solutions. For more details, a review of Nadir point estimation procedures can be consulted in [33]. Figure 2.3 shows an example of the ideal and Nadir vectors in a bi-objective optimization problem.

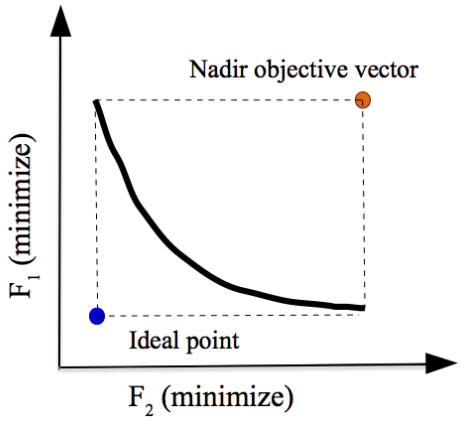


Figure 2.3: Reference points commonly used in multi-objective optimization

2.1.3 Performance Quality Measures

Performance indicators are mathematical models that give us a quantitative measure to assess the quality of an approximate Pareto Front (PF). Let $\mathcal{A} \in \mathcal{S}$ be a set of $m-$dimensional objective vectors. $\mathcal{A}$ is called an approximate PF if any element of $\mathcal{A}$ does not weakly dominate any other vector in $\mathcal{A}$.

$\mathcal{A}$ should accomplish some key features: Firstly, *convergence* towards the POF, namely that all elements in $\mathcal{A}$ are non-dominated and no other solutions exist which dominates them. The second characteristic is the *coverage (spread)* along the PF which means that the candidate solutions are located around all the regions of the PF. If $\mathcal{A}$ has a good coverage, it can provide to the decision maker a wide variety of possibilities. The hypervolume [177] and the unary $R2$ [15] indicators are examples of quality measures that incorporate both aspects, and they are also Pareto compliant. Finally, a *uniform distribution* of the solutions along

the PF is desirable. This refers to having solutions uniformly distributed in the objective space. The three above aspects can be visualized in Figure 2.4. Unfortunately, there does not exist a quality indicator that measures the three characteristics at the same time.

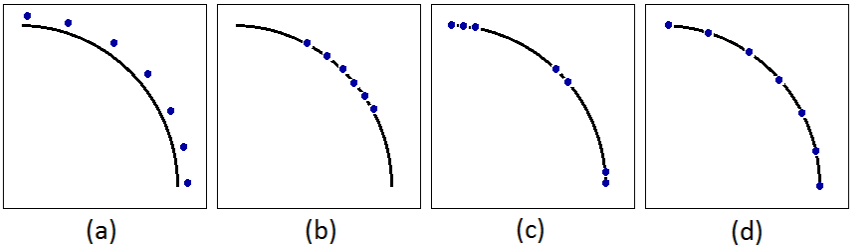


Figure 2.4: The key aspects to assess the quality of the POS (the circles represent the candidate solutions, and the continuous line illustrates a POF with concave shape). (a) this case has a good coverage and uniform distribution but not convergence to the real POF, (b) solutions are located in the optimal PF but do not have a good coverage, (c) good convergence and coverage but the solutions are not uniformly distributed and (d) this is the desired case in which the candidate solutions reach the optimal PF, and have a good coverage and a uniform distribution.

Let $\mathcal{I}(\mathcal{A}) : \mathcal{S}^k \rightarrow \mathbb{R}$, be a $k-$ary quality indicator function which assigns each vector $(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k)$ of k approximate PFs a real value $\mathcal{I}(\mathcal{A}_1, \ldots, \mathcal{A}_k)$. It would be desirable that a quality indicator is Pareto compliant. This means:

Definition 2.1.1 *Assuming that greater indicator values correspond to higher quality, an indicator $I : \mathcal{S} \rightarrow \mathbb{R}$ is Pareto compliant if for all $\mathcal{A}, \mathcal{B} \in \mathcal{S} : \mathcal{A} \preceq \mathcal{B} \Longrightarrow \mathcal{I}(\mathcal{A}) \geq \mathcal{I}(\mathcal{B})$.*

Next, we describe the most important performance measures adopted in this thesis.

2.1.4 Hypervolume Indicator

Zitzler and Thiele [179] proposed the Hypervolume (HV) indicator, also known as Lebesgue measure ($\mathcal{L}$) or S-metric. It measures convergence towards the PF and maximum spread through the union of hypercubes formed by all non-dominated elements in $\mathcal{A}$ with respect to a reference point $\mathbf{r}$ (see Figure 2.5, where it is shown an example of the computation of the hypervolume in a bi-objective optimization case).

HV is Pareto compliant, but it becomes very costly as the number of objectives increases. A high HV value is better. Its mathematical formulation is defined in the following equation:

$$HV(\mathcal{A}, \mathbf{r}) = \mathcal{L}\big(\cup_{i=1}^{\mu} [\mathcal{A}^{(i)}, \mathbf{r}]\big). \tag{2.1}$$

2.1.5 Generational Distance Indicator and its variations

Van Veldhuizen and Lamont [154] introduced the Generational Distance (GD) indicator to measure a relative distance between the obtained set of nondominated solutions and the POF. Its mathematical definition is the following:

Let $\mathcal{Z} = \{z_1, z_2, \ldots z_{|\mathcal{Z}|}\}$ be the best reference point set known, where $|\mathcal{Z}|$ is the cardinality of $\mathcal{Z}$. And $\mathcal{A} = \{a_1, a_2, \ldots a_{|\mathcal{A}|}\}$ be the aproximated set of nondominated solutions.

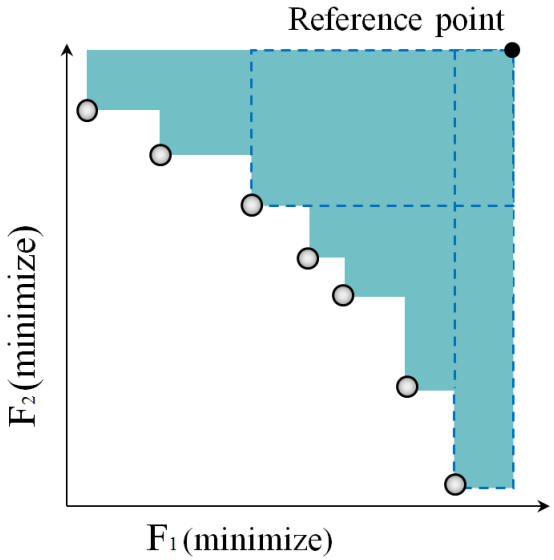


Figure 2.5: A graphical example of computing hypervolume in a bidimensional space.

$$GD(\mathcal{A}, \mathcal{Z}) = \frac{1}{|\mathcal{A}|}\sqrt{\sum_{i=1}^{|\mathcal{A}|} d_i{}^2}, \tag{2.2}$$

where d_i is the Euclidean distance from $\mathbf{a_i} \in \mathcal{A}$ to its nearest member in $\mathcal{Z}$.

Coello and Cortés [22] proposed the **Inverted Generational Distance (IGD)** indicator as a modification of GD. It has the aim of determining how far is located each element of POF ($\mathcal{Z}$) from the estimated set of nondominated solutions ($\mathcal{A}$). Its mathematical definition is:

$$IGD(\mathcal{Z}, \mathcal{A}) = \frac{1}{|\mathcal{Z}|}\Big(\sum_{i=1}^{|\mathcal{Z}|} \hat{d}_i{}^p\Big)^{\frac{1}{p}}, \tag{2.3}$$

where $p \in \mathbb{N}$, usually set to $p = 2$ and $\hat{d}_i$ is the Euclidean distance from $\mathbf{z}_i \in \mathcal{Z}$ to its nearest member in $\mathcal{A}$.

Schütze et al. [140] proposed a variation of IGD in order to make a fair comparison when comparing outcomes of different magnitudes. This slight modification is defined by:

$$IGD_p(\mathcal{Z}, \mathcal{A}) = \Big(\frac{1}{|\mathcal{Z}|}\sum_{i=1}^{|\mathcal{Z}|} \hat{d}_i{}^p\Big)^{\frac{1}{p}}. \tag{2.4}$$

Both indicators, IGD and IGD_p are Pareto non-compliant.

Recently, Ishibuchi [77] proposed the $IGD+$ indicator, which is weakly Pareto compliant. Here, the modified distance and model are defined by the following equations:

$$IGD+(\mathcal{Z},\mathcal{A}) = \frac{1}{|\mathcal{Z}|}\sum_{\mathbf{z}\in\mathcal{Z}} \min_{\mathbf{a}\in\mathcal{A}} d^{+}(\mathbf{z},\mathbf{a}), \tag{2.5}$$

where $d^{+}(\mathbf{z},\mathbf{a})$ is modified distance calculation defined as follows:

$$d^{+}(\mathbf{z},\mathbf{a}) = \sqrt{\sum_{i=1}^{m} (\max\{a_i - z_i, 0\})^2}. \tag{2.6}$$

The aforementioned IGD indicator and its variations can assess convergence and uniform distribution along the PF. The smaller the indicator values the closer the approximation $\mathcal{A}$ is to the reference set $\mathcal{Z}$.

2.1.6 R2 indicator

Hansen and Jaszkiewicz [58] proposed a weakly Pareto complaint indicator, called R2 which maps each objective vector in MOP into a scalar value through a utility function (i.e., a scalarizing function). Given an approximate Pareto point set $\mathcal{A}$ and a set of weight vectors $\boldsymbol{\lambda}$, the R2 indicator is defined as:

$$R2(\mathcal{A}:\Lambda, z^{*}) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda} \min_{\mathbf{a}\in\mathcal{A}}\{ \max_{i\in\{1,\ldots,m\}} \lambda_i |a_i - z_i^{*}| \ \}, \tag{2.7}$$

where z^{*} is a reference point such as the ideal or the utopian point in order to normalize the objectives. More details about scalarizing functions can be seen in Chapter 3. R2 is a good option in many objective problems because of its lower computational cost. Lower values of this indicator are desirable.

2.2 Multi-Objective Evolutionary Algorithms

The following sections present a general overview of the most representative Multi-Objective Evolutionary Algorithms (MOEAs) to solve multi- and many-objective optimization problems. We classify them in two groups: Pareto-based and Non-Pareto-based MOEAs.

The use of Evolutionary Algorithms (EAs) to solve optimization problems has become a trending research area, mainly in recent years [23]. An EA aims to solve complex optimization problems through stochastic mechanisms that provide good approximations to optimal solutions in a reasonable computational time. Evolutionary optimization adopts some specific terminology, such as the following:

- **Population**: a set of candidate solutions used in every generation of an EA. Typically, an EA maintains fixed the population size.
- **Individual**: a member of the EA population that represents a candidate solution of the optimization problem.
- **Generation**: an iteration of an EA.
- **Fitness**: a function derived from equation (2.1) (optimization problem) to obtain the quality of a candidate solution.

- **Evolutionary operators**: the procedures to generate new solutions (children) using the individuals from the current population.
- **Selection mating operator**: a mechanism to choose a mating pool of parent solutions whose information is used to generate new solutions.
- **Elitism**: a procedure to identify at each generation the best solution and to maintain them during the evolutionary process.
- **Survival selection**: a mechanism to determine if a set of solutions survives and continue to the next generation.

Figure 2.6 and Algorithm 1 show a typical structure of an EA. The first step consists of initializing a random population of candidate solutions. Here, it is recommended to use a strategy such as latin hypercubes for generating a uniform distribution of values of the decision variable space to explore all possible regions in search space. The second step is to evaluate each element in the population using the objective(s) function(s) that define the problem. In the case of an MOP there exist strategies to transform a vector optimization problem into a scalar value. The next step is to select the parent solutions which will be used to generate new solutions via an evolutionary procedure. The selection of appropriate evolutionary operators depends on the type of optimization problem to be solved. The following step is to apply a survival selection procedure to choose the solutions that pass to the next generation. These last four steps are repeated until a certain termination criterion is reached. Lines 3, 4 and 6 in Algorithm 1 vary according to the type of MOEA adopted.

Algorithm 1: A general Evolutionary Algorithm procedure

```
   Data: Optimization problem
   Result: Solution of the optimization problem
1  Initialize population.
2  while termination criterion has not been reached do
3      Fitness Assignment.
4      Parent selection.
5      Evolutionary operators.
6      Survival selection.
```

In 1984, David Schaffer [138] adapted a simple genetic algorithm to solve multi-objective optimization problems. The proposal, called Vector Evaluated Genetic Algorithm (VEGA), was the first MOEA reported in the specialized literature and it consisted on dividing the whole population into m subpopulations, at each iteration (m is the number of objectives). Then, each solution in every subpopulation is assigned a fitness value based on one objective function corresponding to their subpopulations (i.e., an individual in subpopulation one gets evaluated with respect to objective one). VEGA uses proportional selection and applies crossover between two solutions from different subpopulations. Then, the mutation operator is applied to each individual. VEGA has two main drawbacks: it does not use an explicit mechanism to maintain diversity and the population tends to converge to solutions which are very superior in one objective, but very poor at others.

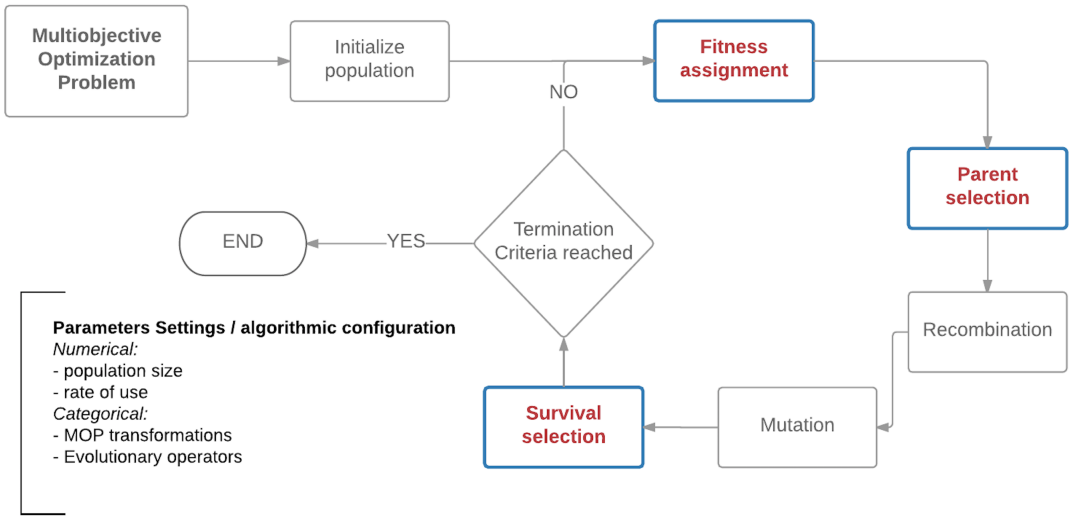


Figure 2.6: General structure of an evolutionary algorithm.

2.2.1 Pareto-based MOEAs

The first generation of MOEAs were algorithms that used Pareto dominance in their selection mechanism such as MOGA [44], NSGA-II [31] and SPEA [179]. MOEAs based on Pareto dominance compare individuals preferring those that are less dominated by other members in the population. When a tie occurs, a secondary selection criterion is applied, usually oriented to improve the diversity of solutions. However, these MOEAs do not work properly in many-objective optimization (i.e., with problems having more than three objectives), because the selection pressure quickly dilutes as we increase the number of objectives.

The next section describes the most representative algorithms based on Pareto dominance.

The Multi-Objective Genetic Algorithm (MOGA) [44] was proposed by Carlos M. Fonseca and Peter J. Fleming in 1993. MOGA implemented a variant of the rank-based fitness assignment method (which was proposed by Goldberg in 1989 [50]) to sort the population. This method assings a rank value to each solution x_i of the current population, using Equation (2.8). Then, the rank value is interpolated considering the best and the worst rank values.

$$\text{rank}(x_i) = \frac{1}{1+n_i}, \tag{2.8}$$

where n_i is the number of solutions that dominate the solution x_i.

Figure 2.7 illustrates this ranking process. In this example, the points a, b and c are nondominated solutions, and therefore the rank value equal to one. Point d is dominated by b and c and has a rank value equal to three, whereas point e is dominated by all the others and it has a rank value equal to five. Algorithm 2 shows all the steps used by MOGA.

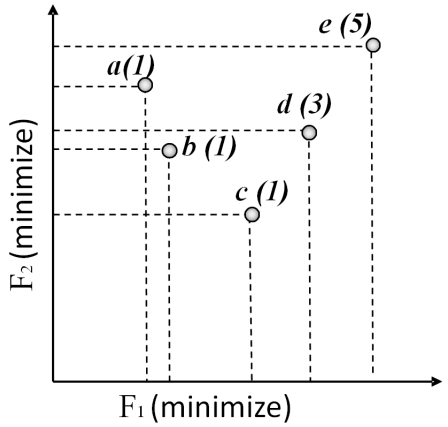


Figure 2.7: An example of the ranking method used by MOGA.

Algorithm 2: The Multi-Objective Genetic Algorithm

```
1  Initialize population.
2  Evaluate objective values.
3  Assign rank based on Pareto dominance.
4  Compute niche count.
5  Assign linearly scaled fitness.
6  Assign Shared Fitness.
7  while termination criterion has not been reached do
8      Selection via stochastic universal sampling.
9      Single point crossover.
10     Mutation.
11     Evaluate objective values.
12     Assign rank based on Pareto dominance.
13     Compute niche count.
14     Assign linearly scaled fitness.
15     Assign shared fitness.
```

Deb et al. [31] proposed the **Non-dominated Sorting Genetic Algorithm II (NSGA-II)**, which eventually became one of the most popular MOEAs used to solve MOPs with two and three objective functions. In NSGA-II, every solution in the population has two associated values which correspond to the two following mechanisms:

1. The *nondominated sorting method* is used to rank the solutions based on Pareto dominance. This is, the population composed by parents and offspring is classified in different levels of dominance. The first step is to identify the nondominated solutions and then, these solutions are removed to obtain a new set of nondominated solutions. So, each subset is associated with different fronts as we can see in Figure 2.8.

2. The crowding distance procedure, which is used to preserve the diversity of solutions. The first step in this procedure is to sort the set of solutions in ascending order according to one objective function. Then, each solution is assigned the average distance of its two neighboring solutions as we can see in Figure 2.8. This procedure is used as a tie-breaker in the selection phase. Namely, if two solutions x and y are in the same nondominated front, the solution with a higher crowding distance wins. Otherwise, the solution with the lowest rank is selected.

Algorithm 3 shows all steps employed by NSGA-II. This MOEA adopts Simulated Binary Crossover and Polynomial-based Mutation, which are described in Algorithms 4 and 5, respectively.

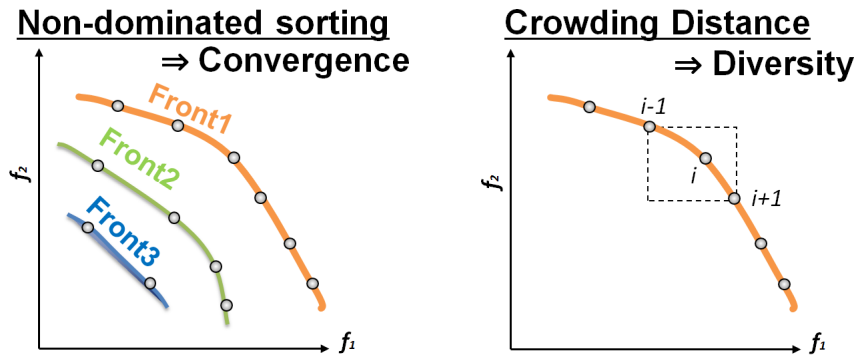


Figure 2.8: Non-dominated sorting and crowding distance methods used by NSGA-II.

Eckart Zitzler [180] proposed the **Strength Pareto Evolutionary Algorithm (SPEA)** which combines procedures from different MOEAs. SPEA works with two population: 1) the whole population P, with dominated and non-dominated solutions and 2) a copy of only non-dominated solutions stored in an external archive E with a limit on size. When the external population size is exceeded, the members in the population are pruned using the average linkage method [113] which is a clustering technique aimed to preserve the characteristics of the nondominated front and to maintain a uniform distribution. At each generation, SPEA employs a fitness assignment mechanism based on fitness sharing using the information of the external population as follows:

- The fitness of each element i in the external population E is assigned using:

$$f_i = \frac{k}{N+1} \tag{2.9}$$

  where k represents the number of solutions dominated by the external solution i and N is the population size of P.

Algorithm 3: The Non-dominated Sorting Genetic Algorithm II

```
Initialize population.
Generate random population (size N).
Evaluate objective values.
Assign rank (level) based on Pareto dominance (sorting mechanism).
Generate offspring population.
Binary tournament selection.
Recombination and mutation.
while termination criterion has not been reached do
    With parent and offspring populations, assign rank (level) based on Pareto
    dominance (sorting mechanism).
    Generate sets of non-dominated fronts.
    for each front do
        Determine crowding distance between points on each front.
        Select points (elitist) on the lower front (with lower rank) and which have the
        highest crowding distance value.
    Binary tournament selection.
    Apply recombination and mutation operators.
```

- The fitness of each element in population P is computed using:

$$f_j = \frac{\sum_{i,j \preceq i} s_i + N + 1}{N + 1} \tag{2.10}$$

  where s_i refers the sum of the solutions dominated by elements j in the external population.

Figure 2.9 illustrates an example of the fitness sharing procedure adopted in SPEA for a bi-objective optimization problem.

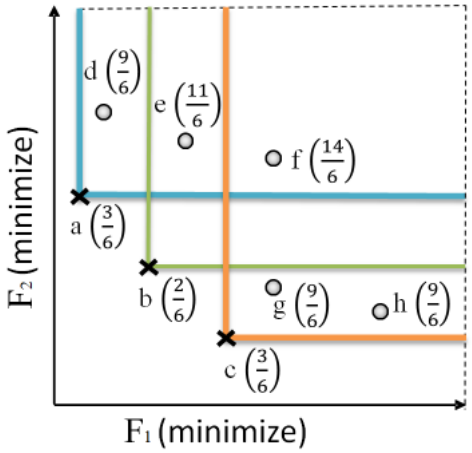


Figure 2.9: An example of the assignment mechanism used by SPEA.

Algorithm 6 shows the details of SPEA.

Algorithm 4: Simulated Binary Crossover

Data: $\mathbf{x}^1$ and $\mathbf{x}^2$: mating parents.
P_c: Crossover rate
η_c: crossover distribution index
n: the number of decision variables
$\mathbf{x}^{upper}$ and $\mathbf{x}^{lower}$: the vector of upper and lower bounds for each decision variable.
Result: $\mathbf{y}^1$ and $\mathbf{y}^2$: a modified vector

$i \leftarrow 0$;
$r \leftarrow U[0,1]$;
if $r \leq P_c$ **then**
  do
    $r_1 \leftarrow U[0,1]$;
    if $r_1 \leq 0.5$ **then**
      $\beta = 1 + (2(x_i^1 - x_i^{lower})/(x_i^2 - x_i^1))$
      $\alpha = 2 - \beta^{-(\eta_c+1)}$
      $r_2 \leftarrow U[0,1]$;
      $$\beta_q \leftarrow \begin{cases} (\alpha \times r_2)^{\frac{1}{\eta_c}+1} & \text{if } r_2 \leq \frac{1}{\alpha} \\ \left(\frac{1}{(2 - \alpha \times r_2)}\right)^{\frac{1}{\eta_c}+1} & \text{otherwise} \end{cases}$$
      $y_i^1 = 0.5(x_i^1 + x_i^2) - \beta_q \times (x_i^2 - x_i^1)$
      $\beta = 1 + (2(x_i^{upper} - x_i^2)/(x_i^2 - x_i^1))$
      $\alpha = 2 - \beta^{-(\eta_c+1)}$
      $$\beta_q \leftarrow \begin{cases} (\alpha \times r_2)^{\frac{1}{\eta_c}+1} & \text{if } r_2 \leq \frac{1}{\alpha} \\ \left(\frac{1}{(2 - \alpha \times r_2)}\right)^{\frac{1}{\eta_c}+1} & \text{otherwise} \end{cases}$$
      $y_i^2 = 0.5(x_i^1 + x_i^2) - \beta_q \times (x_i^2 - x_i^1)$
    else
      $y_i^1 = x_i^1$
      $y_i^2 = x_i^2$
    $i++$
  while $(i \neq n)$;

2.2.2 Non-Pareto-based MOEAs

MOEAs based on Pareto dominance have difficulties to handle many-objective optimization problems. Some of these problems are:

- The number of nondominated solutions increases exponentially with respect to the number of objective functions. In consequence, MOEAs based on Pareto dominance considerably reduce their selection pressure as the number of objectives increases.

- Density estimators are normally computationally expensive with a higher number of objectives.

Algorithm 5: Polynomial-based mutation

Data: $\mathbf{y}$: vector to be modified.
P_m: Rate of use
η_m: mutation distribution index
n: the number of decision variables
$\mathbf{y}^{upper}$ and $\mathbf{y}^{lower}$: the vector of upper and lower bounds for each decision variable.
Result: $\mathbf{y}'$: a modified vector

$i \leftarrow 0$;
do
  $r \leftarrow U[0,1]$;
  if $r \leq P_m$ **then**
    $\delta_1 \leftarrow \frac{y_i - y_i^{Lower}}{y_i^{Upper} - y_i^{Lower}}$;
    $\delta_2 \leftarrow \frac{y_i^{Upper} - y_i}{y_i^{Upper} - y_i^{Lower}}$;
    $r \leftarrow U[0,1]$;
    $\delta_q \leftarrow \begin{cases} [(2r) + (1-2r) * (1-\delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1 & \text{if } r \leq 0.5 \\ 1 - [2(1-r) + 2 \cdot (r-0.5) * (1-\delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & \text{otherwise} \end{cases}$
    $y_i' \leftarrow y_i + \delta_q \cdot \left(y_i^{Upper} - y_i^{Lower}\right)$
    $i++$
while $(i \neq n)$;

Algorithm 6: The Strength Pareto Evolutionary Algorithm

Initialize population P.
Create empty external set E.
while *termination criterion has not been reached* **do**
  Copy nondominated members of P to E.
  Remove elements from E which are covered by any other member of E.
  Prune E (using clustering) when the maximum capacity of E has been exceeded.
  Compute fitness of each individual in P and in E.
  Use binary tournament selection with replacement to select individuals from $P + E$ (multiset union).
  while *the mating pool is not full* **do**
    Apply crossover and mutation operators.

- In a high dimensional objective space, is difficult for the decision maker to establish preferences.

The two main approaches to deal with many-objective problems are: (1) indicator-based MOEAs and (2) decomposition-based MOEAs. The former refers to the use of methods to assess the quality of the approximation generated by a MOEA to select solutions or as a secondary selection criterion, whereas the latter transforms a multi-objective problem into several single-objective problems which are solved simultaneously.

Indicator-based MOEAs favor solutions that highly contribute to a performance indicator, which reflects a quality aspect regarding convergence and diversity of the current population. The indicator-based Evolutionary Algorithm (IBEA) [181] was proposed as a general framework for coupling evolutionary algorithms to a selection mechanism based on a performance indicator. IBEA is an algorithm that incorporates a mating selection using binary tournaments and includes an iterative procedure for removing the worst individual from the population to update the fitness values of the remaining individuals. Another well-known indicator-based MOEA is the S-Metric Selection Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [41] which is a steady-state algorithm that generates at each generation only one new individual with the goal of maximizing the hypervolume indicator. In SMS-EMOA, the primary selection criterion mechanism is the non-dominated sorting algorithm, and the secondary criterion is the hypervolume which is applied only to the last front. SMS-EMOA is a powerful MOEA, but it is too expensive (computationally speaking), specially in many-objective optimization problems.

On the other hand, decomposition-based MOEAs transform a MOP into several single-objective subproblems, and each one is associated with a different target direction that optimizes a particular aggregating function. Decomposition-based MOEAs have several advantages such as scalability to many-objective problems (MaOPs) [61,104], a high search ability for combinatorial optimization [74,75,136], and a high compatibility with local search methods [97,117,167].

In this thesis, we focus on MOEAs that use reference vectors and aggregating functions. The next section provides details of the most representative algorithms within this class.

The framework called **Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D)** proposed by Zhang [171] has been widely used to solve a large number of MOPs. MOEA/D decomposes a MOP into several single-objective subproblems which are solved in a collaborative manner. Each solution from the population is associated with a weight vector to optimize a scalarizing function. Two goals should be accomplished during the search process. First, to minimize the distance between a candidate solution and the reference point (typically, the ideal vector) to achieve convergence towards the Pareto optimal front. The second step is to maintain a diversity of solutions through the target directions of the search previously defined by a uniform distribution around all regions of objective space. A crossover operator is applied only in a neighborhood of solutions predefined according to the nearest neighbors in weight vector space. Figure 2.10 shows an example of the search process applied by MOEA/D. Here, five target directions $(\lambda_1, \ldots, \lambda_5)$ are defined in two-dimensional space. Each solution optimizes one direction to converge to the Pareto optimal front.

Algorithm 7 shows the main steps employed in MOEA/D. The initialization process (lines 1 to 4) generates a set of N weight vectors, a population with N candidate solutions and a reference point. Each solution is evaluated by a scalarizing function $g(f|\lambda^j, \Omega)$. In the iterative process, each element x^s from the population is enhanced by applying evolutionary operators via a selection of parent solutions from its neighborhood. After that, an update process modifies the reference point and the current population according to the scalarizing function values (see Chapter 3, for more details about scalarizing functions). The steps described in lines 6 to 12 are repeated until the stopping criterion is satisfied.

The first version of MOEA/D adopted Simulated Binary Crossover (SBX) [30] and Polynomial-based Mutation (PM). These evolutionary operators are described in Algorithms 4 and 5. The framework of MOEA/D was tested with the Weighted Sum, Chebyshev

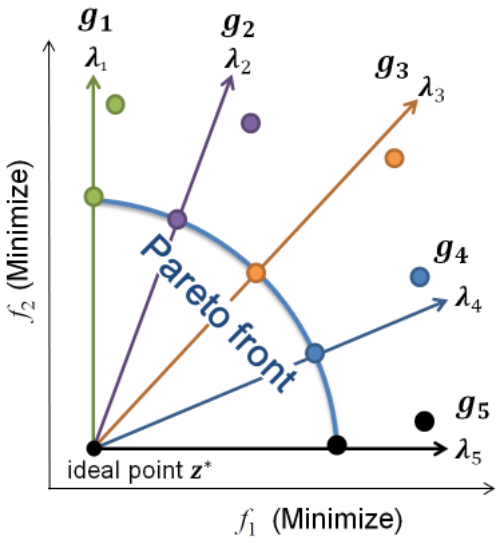


Figure 2.10: Example of the search process in MOEA/D in two dimensions.

Algorithm 7: MOEA/D algorithm

Data: N: population size
T: neighborhood size
Result: Pareto Set Approximation

```
Initialize N weight vectors λ^1, ..., λ^N.
Determine the neighborhood of each weight vector by recording its nearest T
neighbors.
Initialize the population x^1, ..., x^N and evaluate them.
Initialize the reference point z*.
Reproduction and update
while Stopping conditions are not satisfied do
    for each sϵ{1, ..., N} do
        Select mating parents from the neighborhood of x^s.
        Generate a new solution y by using operators of recombination and mutation.
        Update the reference point z*.
        Evaluate new solutions via scalarizing function.
        Update the population.
```

and Penalty Boundary Intersection (PBI) functions (their mathematical definition can be seen in Chapter 3). PBI attained the best performance in problems with more than three objectives. In the other cases, a good option was the Chebyshev function.

There are several enhanced versions of MOEA/D. Some modifications include the following: 1) to replace the genetic operators by other evolutionary approaches [102], 2) to modify the selection and replacement strategies, 3) the use of online tuning techniques or mechanisms based on learning period models to select an operator from a predefined pool. Next, we describe the most representative approaches that have been proposed in the spe-

cialized literature. *MOEA/D-DE* [102] replaces the genetic operators SBX and PM by those of Differential Evolution (DE) [145] and introduces an additional parameter δ to balance the selection of the mating parents between the neighborhood of the target subproblem and the whole population. MOEA/D-DE was able to successfully solve the test instances of the CEC 2009 competition [99]. In [65], Huang and Li presented a comprehensive study of *MOEA/D-DE* using different DE schemes.

Chen et al. [19] replaced the DE operator by a guided mutation technique [64]. Additionally, a replacement mechanism was incorporated via a priority queue to monitor the subproblems that were successfully updated before. This is similar to the process adopted in priority-scheduling used in operating systems. In [172], Zhang proposed the *MOEA/D-DRA* with two crossover operators: Simplex Crossover (SPX) and Center of Mass Crossover (CMX). Every k generations, *MOEA/D-DRA* monitors a relative decrease of the objectives for each subproblem and based on this, a tournament selection strategy is employed. *MOEA/D-AMS* proposed in [21] improved MOEA/D-DE using two strategies: 1) a controlled selection of subproblems with the goal of identifying unsolved subproblems to use more computational effort on these, and 2) an adaptive mating selection mechanism that considered the Euclidean distance between individuals in decision space instead of the distance between weight vectors. Adaptive Differential Evolution for Multiobjective Problems (*ADEMO/D*) [155] incorporated a pool of mutation strategies adaptation inspired by the adaptive SaDE algorithm. Each strategy is associated with a probability of use which is updated based on its success and failure counters. Other proposals include adaptive mechanisms. For more details see Chapter 4.

The **Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization (MOMBI-II)** proposed by Hernández and Coello [61] is an effective algorithm to tackle many-objective problems. MOMBI-II used the Achievement Scalarizing Function (ASF) [162] which outperforms the Chebyshev model adopted in MOEA/D (for more details see Chaper 3). In contrast, MOMBI-II computes the R2 indicator for ranking the current population. If two individuals contribute with the same utility, then the tiebreaker is set by the lower Euclidean distance. MOMBI-II incorporates various reference points: $\mathbf{z}^*$, $\mathbf{z}^{nadir}$, $\mathbf{z}^{min}$ and $\mathbf{z}^{max}$. The last two are updated in an adaptive way according to the variance of the Nadir point. They are also used to normalize each objective in the current population using:

$$f_i' = \frac{f_i(\mathbf{x}) - z_i^{min}}{z_i^{max} - z_i^{min}} \quad \forall_i \in \{1, \ldots, m\}. \tag{2.11}$$

MOMBI-II is more computationally expensive than MOEA/D but is effective in avoiding the generation of weakly Pareto optimal solutions. Algorithm 8 shows the general steps executed by MOMBI-II.

The Non-dominated Sorting Genetic Algorithm-III (NSGA-III) [82] incorporates some procedures from the NSGA-II but it incorporates the use of reference points and a scalarizing function.

NSGA-III uses the general framework of a genetic algorithm: it initializes the population and applies mating selection for creating new solutions via evolutionary operators (SBX and PM). In order to solve a MOP, NSGA-III uses the following operators:

- It establishes a set of reference directions for each objective function (similar to

Algorithm 8: MOMBI-II algorithm

Data: N: population size
Result: Pareto set approximation
Initialize N weight vectors $\lambda^1, \ldots, \lambda^N$.
Initialize the population $x^1, \ldots, x^N$ and evaluate each individual.
Compute the $L_2 - norm$ of objective space for the population.
Initialize the reference points: $\mathbf{z}^{min} \leftarrow \mathbf{z}^*$ and $\mathbf{z}^{max} \leftarrow \mathbf{z}^{nad}$.
Reproduction and update
while *Stopping conditions are not satisfied* **do**
    for *each* $s\epsilon\{1, \ldots, N\}$ **do**
        Select mating parents.
        Generate new solutions y by using operators of recombination and mutation.
        Compute the $L_2 - norm$ of objective space for the population
        Normalize the objective functions.
        Execute the $R2$ ranking procedure (Algorithm 9)).
        Update the population.
        Update the reference point z^* via Algorithm 10.

Algorithm 9: R2 ranking algorithm

Data: Population P, set of weight vectors Λ
Result: Ranked population
Set $p.rank \leftarrow p.\alpha \leftarrow \infty \forall_p \in P$.
for $\lambda \in \Lambda$ **do**
    for $p \in P$ **do**
        $p.\alpha \leftarrow g(p.\mathbf{F}|\mathbf{0}, \lambda)$.
    Sort P w.r.t. the fields α and L_2 in increasing order; $rank \leftarrow 1$.
    for $p \in P$ **do**
        $p.rank \leftarrow min\{p.rank, rank\}$. $rank \leftarrow rank + 1$.

MOEA/D).

- At each generation, nondominated sorting is applied to the population in order to identify levels of nondominance.
- The solutions are transformed using a normalization process that involves the ideal point.
- Each solution optimizes the ASF and includes a second selection mechanism based on the Euclidean distance between the candidate solution and the associated reference point.

Figure 2.11 shows an example related to the operation of NSGA-III. The dotted lines represent the target directions established by the reference point which is shown in red color. The normalized candidate solutions in blue color are associated to each target direction.

Algorithm 11 shows the general structure of NSGA-III.

Algorithm 10: Update Reference Points

Data: $\mathbf{z}^{min}$, $\mathbf{z}^{max}$, population P
Result: updated reference points $\mathbf{z}^{min}$, $\mathbf{z}^{max}$

Update $\mathbf{z}^*$ and $\mathbf{z}^{nadir}$:
$\mathbf{z}^{min} \leftarrow min\{z_i^{min}, z_i^*\}\ \forall \in \{1, \ldots, m\}$
Store $\mathbf{z}^{nad}$ in *record*.
Obtain vector of variances $\mathbf{v} \in \mathbb{R}^m$ for $\mathbf{z}^{nad}$ from *record*.
if $\max_{j \in \{1,\ldots,m\} v_j} > \alpha$ **then**
    $z_i^{max} \leftarrow \max_{j \in \{1,\ldots,m\}} z_j^{nad}\ \forall_i \in \{1, \ldots, m\}$
else
    for *all* $i \in \{1, \ldots, m\}$ **do**
        if $|z_i^{max} - z_i^{min}| < \epsilon$ **then**
            $z_i^{max} \leftarrow \max_{j \in 1,\ldots,m} z_j^{max}$.
            Mark z_i^{max}.
            else if $z_i^{nad} > z_i^{max}$ **then**
                $z_i^{max} \leftarrow 2z_i^{nad} - z_i^{max}$.
                Mark z_i^{max}.
            else if $v_i = 0$ *and* z_i^{max} *has not been marked recently* **then**
                Obtain the maximum value α for z_i^{nad} from *record*.
                $z_i^{max} \leftarrow (z_i^{max} + \alpha)/2$.
                Mark z_i^{max}.
Set $p.rank \leftarrow p.\alpha \leftarrow \infty \forall_p \in P$
for $\lambda \in \Lambda$ **do**
    for $p \in P$ **do**
        $p.\alpha \leftarrow g(p.\mathbf{F}|\mathbf{0}, \lambda)$.
    Sort P w.r.t. the fields α and L_2 in increasing order $rank \leftarrow 1$.
    for $p \in P$ **do**
        $p.rank \leftarrow min\{p.rank, rank\}$.
        $rank \leftarrow rank + 1$.

2.3 Summary

In this chapter, we explained the basic mathematical definitions that are required to understand the rest of the thesis. We also provided a description of the performance indicators that are adopted to validate our results.

We discussed the characteristics of diverse MOEAs based on Pareto dominance and MOEAs that use reference points and scalarizing functions. We provide details of the MOEA/D and MOMBI-II frameworks because our experiments are focused on the behavior and performance of scalarizing functions.

The next chapter provides a review of several scalarizing functions used in the areas of mathematical programming and evolutionary computation. We will explain their mathematical properties based on their model parameters and their effect in different evolutionary

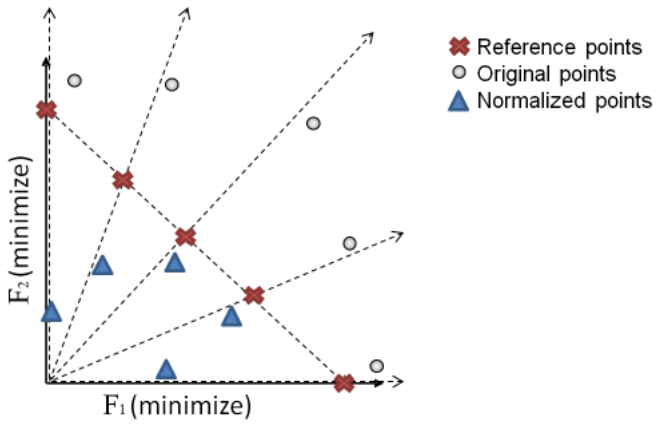


Figure 2.11: A graphical representation of the operation of the NSGA-III algorithm in a problem with two objective functions.

Algorithm 11: The NSGA-III algorithm

Data: A set of structured reference points and a parent population P_t
Result: A new population P_{t+1}

$S_t = \emptyset$, $i = 1$.
Q_t = Recombination + mutation (P_t).
$R_t = P_t \cup Q_t$.
$(F_1, F_2, \ldots)$ = non-dominated-sort(R_t).
while $|S_t| \leq N$ **do**
    $S_t = S_t \cup F_i$ and $i = i + 1$
Last front to be included: $F_l = F_i$.
if $|S_t| = N$ **then**
    $P_{t+1} = S_t$, break.
else
    $P_{t+1} = \cup_{j=1}^{l-1} F_j$ Points to be chosen from $F_l : K = N - |P_{t+1}|$.
    Normalize objectives and create reference set Z^r, using $(f^n, S_t, Z^r, Z^s, Z^a)$
    Associate each member s of S_t with a reference point, $[\pi(s), d(s)]$ associate with $(S_t, Z^r)\%\pi(s)$ to the closest reference point, d : distance between s and $\pi(s)$.
    Compute niche count of reference point $j \in Z^r$: $\rho_j = \Sigma_{s \in S_t/F_l}((\pi(s) = j)$.
    Choose K members one at a time from F_l to construct P_{t+1}.
    Niching$(K, \rho_j, \pi, d, Z_r, F_l, P_{t+1})$.

algorithms of the state-of-the-art as well as the challenges that they face to tackle diverse Pareto front shapes.

Chapter 3

Weighted and Unconstrained Scalarizing Functions

A Scalarizing Function (SF) also known as utility or aggregation function is a useful mathematical programming technique used to transform a multi-objective optimization problem (MOP) into a single-objective one, its goal is to combine a vector of objective functions $\mathbf{y} = [f_1, \ldots, f_m]^T$ to obtain a scalar value $g(\mathbf{y}) : \mathbb{R}^m \rightarrow \mathbb{R}$. There exists a large variety of SFs that can be employed as a priori, a posteriori or progressive preference articulation mechanisms [105]. SFs can be classified according to the way in which they perform the transformation in weighted/unweighted SFs and if they add extra restrictions into the original MOP. Figure 3.1 shows a taxonomy of the SFs based on their mathematical definition.

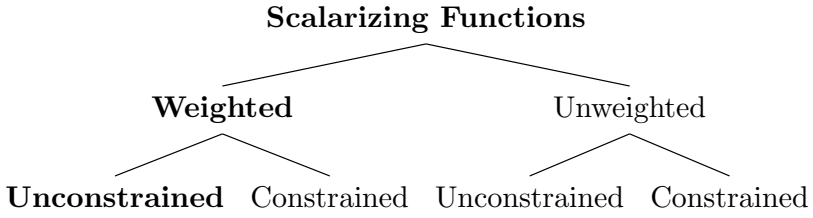


Figure 3.1: A taxonomy of scalarizing functions.

Some examples of weighted and unweighted SFs without constrained functions are the global criterion method [109, p. 67] and the L_p-metric function [109, p. 70]. Unweighted and constrained SFs have been used in methods such as: ϵ-Constraint [109, p. 85], Lexicographic ordering [109, p. 119], Light Beam Search [109, p. 179], Reference Direction [109, p. 191], whereas some instances of weighted and constrained approaches are the Hybrid Method [109, p. 96] and the Reference Direction Approach [109, p. 185].

In this thesis, we focus on weighted and unconstrained SFs which are mathematically defined as follows:

$$\text{minimize} \quad g(\mathbf{f}'(\mathbf{x}); \boldsymbol{\lambda}) \tag{3.1}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{S}, \tag{3.2}$$

where $\boldsymbol{\lambda} = \{\lambda^1, \ldots, \lambda^m\}$ is a weight vector and each component of $\boldsymbol{\lambda}$ must satisfy that $\lambda_i > 0$ for all $i \in \{1, \ldots, m\}$ and $\sum_{i=1}^{m} \lambda^i = 1$ [135]. $\mathbf{f}'(\mathbf{x})$ is a normalization of the objective vector, which allows handling negative or incommensurable objectives, and may adopt one of the forms:

$$\mathbf{f}'(\mathbf{x}) := \mathbf{f}(\mathbf{x}) - \mathbf{z}^*, \tag{3.3}$$

$$\mathbf{f}''(\mathbf{x}) := \frac{\mathbf{f}(\mathbf{x}) - \mathbf{z}^*}{\mathbf{z}^{nad} - \mathbf{z}^*}, \tag{3.4}$$

$$\mathbf{f}'''(\mathbf{x}) := \mathbf{z}^{nad} - \mathbf{f}(\mathbf{x}). \tag{3.5}$$

Other forms to transform the objective functions to dimensionless form are presented in [105, p. 373]. In [120, p. 865], evidence is provided to validate that the lack of a normalization process causes deterioration in the performance of an MOEA based on decomposition.

The use of weight vectors in the SFs can be seen as a representation of order that defines a relative importance of the objective functions in a MOP establishing search directions in the objective space. Typically, the more uniform is the distribution of these search directions, the more the diversity of solutions over the objective space. Setting one or more of the weight components to zero can result in weak Pareto optimality [105].

In general, weighted and unconstrained SFs minimize some distance metric between a candidate optimal solution and a reference point (e.g., the ideal or nadir vectors) using a target direction (weight vector). Other SFs combine two distance metrics and some of them also consider the deviation distance to the weight vector. Some desirable properties for capturing optimal solutions via SFs are described next:

1. *Sufficient condition.* SFs should be monotonically increasing concerning every objective function. Namely, $g(\mathbf{f}'(\mathbf{x}); \boldsymbol{\lambda})$ is strictly increasing if it is nondecreasing and if it presents a positive response to an increase in at least one input value.

   Mathematically, an SF is *Pareto compliant*, if it satisfies:

   $$\forall \mathbf{x}, \mathbf{y} \in \mathcal{S} : \mathbf{x} \prec \mathbf{y} \Rightarrow g(\mathbf{f}'(\mathbf{x}); \boldsymbol{\lambda}) < g(\mathbf{f}'(\mathbf{y}); \boldsymbol{\lambda}), \tag{3.6}$$

   and *weakly Pareto-compliant* if it holds:

   $$\forall \mathbf{x}, \mathbf{y} \in \mathcal{S} : \mathbf{x} \preceq \mathbf{y} \Rightarrow g(\mathbf{f}'(\mathbf{x}); \boldsymbol{\lambda}) \leq g(\mathbf{f}'(\mathbf{y}); \boldsymbol{\lambda}). \tag{3.7}$$

   Otherwise, an SF is Pareto non-compliant.

2. *Necessary condition.* Any Pareto optimal point should be obtainable by an SF by adjusting its model parameters or weight vectors.

3. In addition, the SFs should avoid numerical problems such as overflow and underflow.

During the last decades, SFs have been successfully coupled to MOEAs such as MOEA/D [171], MOMBI-II [61], NSGA-III [82], or RVEA [20] achieving a good performance in terms of convergence and uniform distribution along the Pareto optimal front (especially in MOPs with many objectives) at a low computational cost. Nevertheless, several studies [36, 49, 75, 81, 124] have revealed that their efficiency depends strongly on the Pareto front geometry and the number of objective functions. For example, the weighted sum function [166] has a faster convergence speed [79] but can only solve MOPs with convex Pareto front shapes. Conversely, SFs with additional model parameters such as the Penalty Boundary Intersection (PBI) function are able to handle MOPs with different Pareto front geometries. However, several experimental works [75, 110, 136, 137] have shown a high parameter sensitivity on the penalty parameter of PBI (θ), indicating that the choice of an appropriate value depends on specific features of the MOP.

This chapter is focused on the study of weighted and unconstrained SFs. We analyze their mathematical definition and properties to generate Pareto optimal solutions in MOPs with different features. Moreover, we present a review of the most representative MOEAs in which each SF has been employed.

3.1 Scalarizing Functions based on the L_p metric

The L_p-metric function also known as **the weighted norm (WN)** model [168] is one SFs to that best adapts any Pareto front shape. It has been used recently in [75,81] to deal with different Pareto front curves. Its mathematical definition is described in the following equation:

$$\min \quad g^{lp}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', p) = \Big(\sum_{i=1}^{m} \lambda_i \big|f_i'(\mathbf{x})\big|^p\Big)^{\frac{1}{p}}, \tag{3.8}$$

where $f_i'(\mathbf{x})$ can be defined by equations (3.3), (3.4), (3.5) and the exponent p is a variable parameter that can be set in the range $1 \leq p < \infty$. The basic idea of WN is to minimize the distance from the POF to the reference point, being able to generate Pareto optimal solutions when $p \in [1, \infty)$ [109, p. 98].

A specific $p-$value can represent other classical functions; one example is the so-called weighted sum method [166]. When $p = 1$, this model involves a linear combination of objectives, but does not provide a necessary condition for Pareto optimality, neither captures solutions on non-convex Pareto front shapes. When $p = 2$, the weighted least square model [109, p. 97] is represented. Athan and Papalambros [4] proved that higher p values increase the effectiveness of the method in providing the whole Pareto optimal set. If $p \to \infty$, the model is transformed into the Chebyshev function, also known as weighted min-max method [85] that can achieve Pareto optimal points, especially with a non-convex Pareto front geometry.

In figure 3.2, we can appreciate a contour line plot to visualize the effect of the search performed by the three aforementioned models. In this plot, the same direction vector $\lambda = (0.5, 0.5)$ is used. The L_1 metric finds optimal solutions only in the extremes of the Pareto front, while L_2 can find more than one solution if the curvature coincides with the Pareto front. L_∞ intersects one optimal solution in the same search direction. It shows the influence of the curvature of SFs to obtain candidate solutions, given a specific Pareto front shape.

Considering this effect, Ishibuchi [75, 81] have proposed local estimations in the Pareto front curvature to adjust the $p-$value. In [159], the behavior of WN was studied on MOEA/D, using continuous test problems having up to 7 objectives. The value of p was adaptively fine-tuned based on a local estimation of the Pareto front shape, taking different values from the set $\{1/2, 2/3, 1, 2, 3, \ldots, 10, 1000\}$.

The **Weighted Sum (WS)** [166] is one of the most commonly used SFs, which linearly combines the objectives as follows:

$$\min \quad g^{ws}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}') = \sum_{i} \lambda_i f_i'(\mathbf{x}). \tag{3.9}$$

WS cannot generate solutions in concave regions of the Pareto front [98]. Some attempts have been proposed to alleviate this drawback, such as its combination with other SFs [67, 79, 80], the use of dynamic weights in combination with a secondary population [84], and the use of WS as a local search engine [121, 160]. Additionally, some studies have reported that WS is an effective method for solving many-objective problems [73, 114]. This SF has been integrated into several evolutionary algorithms (see e.g., [78, 171]).

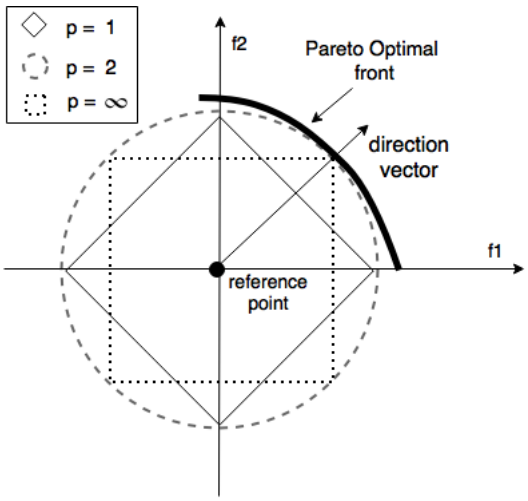


Figure 3.2: A contour line plot for the $Lp-$ metric function when $p = 1$, $p = 2$, $p \to \infty$ and $\lambda = (0.5, 0.5)$ in a bi-objective optimization problem.

Weighted Compromise Programming (WCP) [168] is a variation of the *Global Criterion Method* [23, p. 32] that includes the weight vector for modeling preferences as follows:

$$\min \quad g^{wcp}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', p) = \sum_i \left(\lambda_i f_i'(\mathbf{x})\right)^p. \tag{3.10}$$

A high value of $p \in (1, \infty)$ is preferred to obtain the complete Pareto Optimal Set (POS) [4]. In [98], the authors recommend to use odd values for this parameter and coupled WCP ($p = 9$) with a metaheuristic for solving convex and concave Pareto fronts with 2 and 3 objectives.

The **Weighted Power (WPO)** [100], also known as *the p-power Lagrangian formulation*, is given by:

$$\min \quad g^{wpo}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', p) = \sum_i \lambda_i \left(f_i'(\mathbf{x})\right)^p. \tag{3.11}$$

For a suitable value of $p \in [1, \infty)$, this SF can also find optimal solutions in concave Pareto-fronts [101, 105]. In [34], WPO was coupled with a genetic algorithm, where the weight vectors and the exponent p were updated during the evolutionary process according to predefined rules.

3.2 Scalarizing Functions based on Chebyshev model

The **Chebyshev (CHE) function** [85] also known as the weighted min-max [67, 105] defined in equation (3.12) minimizes the maximum value found by the linear combination between each objective function and a weight vector component. This model is weakly Pareto compatible.

$$\min \quad g^{che}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}') = \max_{i,\ldots,m} \left\{\lambda_i \left| f_i'(\mathbf{x}) \right|\right\}. \tag{3.12}$$

The CHE function was used in the MOEA/D [171] framework showing a clear effectiveness to obtain at least weakly Pareto optimal solutions in non-convex Pareto fronts, in contrast to the weighted sum method. Graphically, the optimal solutions are located in the knee and extremes of the Pareto front (see left column in Table 3.1).

A recent analysis [46] has revealed that the search ability of CHE is equivalent to Pareto-based methods. Thus, in many-objective problems, the probability to obtain non-dominated solutions using CHE is lower than the WN ($0 < p < \infty$) and equivalent to Pareto-based methods. CHE has been adopted by different MOEAs (see e.g., [60, 171]).

Some modified CHE models have been introduced with the goal of avoiding the generation of weakly Pareto solutions. The so-called **Augmented Chebyshev (ACHE)** function [144] given by equation (3.13) considers the CHE function and an augmented term defined by the L_1- metric which is multiplied by a small positive scalar α. In contrast to equation (3.12), the ACHE function generates different slopes for each objective function.

$$\min \quad g^{ache}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha) = \max_{i,\ldots,m} \left\{\lambda_i \left| f_i'(\mathbf{x})\right|\right\} + \alpha \sum_{i=1}^{m} \left| f_i'(\mathbf{x})\right|. \tag{3.13}$$

A too small value of α may result in a loss of significance of the extra term, still leading to the generation of weakly Pareto optimal solutions. However, a too large value of this parameter may cause that some non-dominated points become unreachable [129]. Although the recommendation is to use small values of α, such as $[0.001, 0.01]$ [143], some studies have shown a better performance when using large values, revealing a high sensitivity of this parameter on discrete many-objective problems [80].

Another variation to the CHE model is known as **modified Chebyshev (MCHE)** function [86] defined by equation (3.14). Here, both terms described in the ACHE model are affected by a weight vector. This model generates the same slopes for all objective functions. More details can be consulted in [109, p. 101]. Both, the ACHE and MCHE are compatible with Pareto optimality.

$$\min \quad g^{mche}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha) = \max_{i,\ldots,m} \left\{ \lambda_i \left(\left| f_i'(\mathbf{x})\right| + \alpha \sum_{i=1}^{m} \left| f_i'(\mathbf{x})\right| \right) \right\}, \tag{3.14}$$

where α should be a small positive value. The features of this method are discussed and illustrated in [109]. To the best of our knowledge, this SF has not been exploited in any MOEA.

In the CHE, ACHE and MCHE models, the optimal achievement solutions are located in an opposite direction to the reference line established by each weight vector. The use of reciprocal weight vectors corrects this behavior. The most common approach is the so-called **Achievement Scalarizing Function (ASF)** [162], similar to the CHE model. This function minimizes the maximum value found multiplying each objective function by a reciprocal weight vector. It is given by equation (3.15).

$$\min \quad g^{asf}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}') = \max_{i,\ldots,m} \left\{ \frac{f_i'(\mathbf{x})}{\lambda_i} \right\}. \tag{3.15}$$

The ASF model has been widely employed in solving many-objective optimization problems (more than three objectives) with different decomposition-based MOEAs such as MOMBI-II [61] NSGA-III [82] and MOEA/D-GRA [175]. ASF can find Pareto optimal points parallel to λ, improving diversity in many-objective problems, as was shown in [35, 61, 165].

There are interesting observations about the convergence and distribution of the Pareto optimal solutions found by the CHE function and its reciprocal version (ASF). Table 3.1 shows the median execution of 30 independent runs using the MOEA/D framework [171] on DTLZ1, DTLZ3, and DTLZ3^{-1} which have linear, concave and convex Pareto front shapes, respectively. Also, we computed the mean and standard deviation for two performance indicators: Hypervolume (HV) [179] and a modification of Inverted Generation Distance (IGD+) [75] to measure convergence, spread, and uniformity. Here, we can appreciate that CHE finds solutions in the extremes and the knee of the Pareto front (thus, the HV is greater than ASF). In contrast, a better uniform distribution of the optimal points was reached by ASF in all three Pareto front shapes tested.

In the case of the augmented Chebyshev functions, an appropriate value for the parameter α has an impact on the performance of the MOEA. In [129] and [27] it was mentioned that a small α value can find all nondominated solutions in non-convex problems while, on the other hand, a large α value prevents the generation of weakly Pareto optimal solutions. Moreover, a too small α value may cause an underflow numerical problem which results is a loss of significance in the augmentation term.

The weighted Chebyshev functions and its variations have been commonly used in the improved MOEA/D versions [19, 65, 102, 172]. These proposals have incorporated other changes in their components such as the evolutionary operators or have included strategies of resource allocation based on the SF performance. Similar to the ACHE function, Miettinen [109, p. 111] proposed the **Augmented Achievement Scalarizing Function (AASF)** [109, p. 111] defined in equation (3.16). In this case, the augmented term is equal to the weighted sum function with reciprocal weight vector. It is noted that the formulation of ASF and AASF does not include the absolute value of f'.

$$\min \quad g^{aasf}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha) = \max_{i,\ldots,m}\left\{\frac{f_i'(\mathbf{x})}{\lambda_i}\right\} + \alpha \sum_{i=1}^{m} \frac{f_i'(\mathbf{x})}{\lambda_i}, \tag{3.16}$$

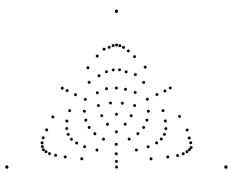
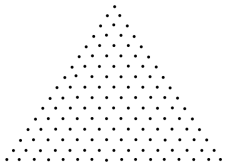
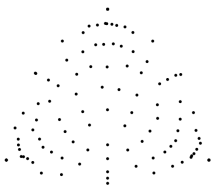
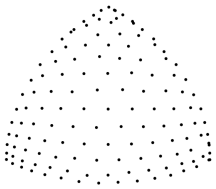
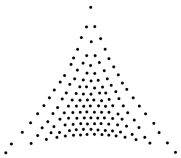
α should take small values. In [151], it was recommended to set $\alpha \approx 10^{-4}$. There are few MOEAs that adopt this scalarizing function [134, 151].

Figure 3.3 shows the parallel coordinates for CHE, ASF and AASF ($\alpha = 0.1$) to appreciate their behavior in the search process for bi-objective problems considering convex, concave and linear Pareto fronts.

In this thesis, we introduce two additional models using the ACHE and MCHE models, defined by equations (3.17) and (3.18), respectively. The aim of these two additional models is to corroborate if reciprocal weight vectors are more beneficial in previous CHE versions.

$$\min \quad g^{rache}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha) = \max_{i,\ldots,m}\left\{\left|\frac{f_i'(\mathbf{x})}{\lambda_i}\right|\right\} + \alpha \sum_{i=1}^{m} \left|f_i'(\mathbf{x})\right|, \tag{3.17}$$

Table 3.1: Distribution of Pareto optimal solutions on linear, convex and concave Pareto front shapes.

CHE	ASF
$DTLZ1$	
	
IHV = **2.6969e+01(1.4894e-04)**	IHV = 2.6900e+01(1.0348e-02)
IGD+ = 1.6334e-02(2.7468e-04)	IGD+ = **1.1585e-02(1.7939e-04)**
$DTLZ3$	
	
IHV = **2.6385e+01(5.3241e-03)**	IHV = 2.6374e+01(1.0869e-02)
IGD+ = 3.1332e-02(2.1360e-03)	IGD+ =**2.3957e-02(2.3843e-03)**
$DTLZ3^{-1}$	
	
IHV = **2.6920e+01(2.4372e-03)**	IHV = 2.6871e+01(2.6856e+01)
IGD+ = **1.5650e-06(8.2029e-07)**	IGD+= 8.8691e-05(5.9436e-05)

$$\min \quad g^{rmche}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha) = \max_{i,\ldots,m} \left\{ \frac{\left|f_i'(\mathbf{x})\right| + \alpha \sum_{i=1}^{m} \left|f_i'(\mathbf{x})\right|}{\lambda_i} \right\}. \tag{3.18}$$

The weighted Chebyshev method and its variations have been commonly used in the improved MOEA/D versions ([102], [19], [65], [172], [21], [155], [175]).

In Figure 3.4, we can appreciate the parallel coordinates for the aforementioned augmented CHE models. In Chapter 6, we present a parameter sensitivity study related to the

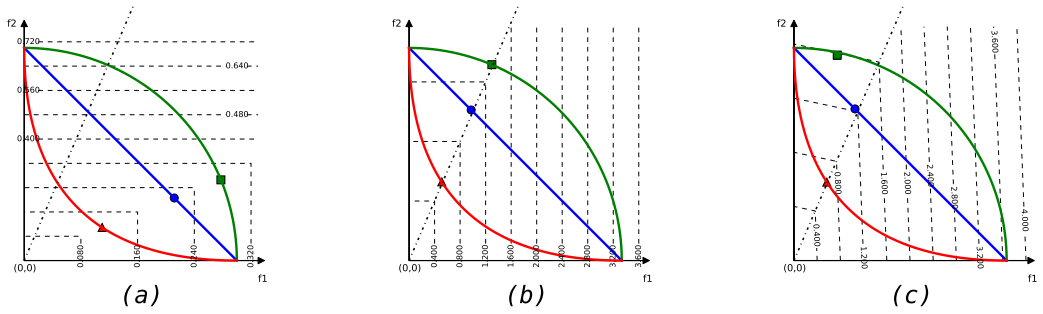


Figure 3.3: (a) The CHE function, (b) the ASF model and (c) the AASF with $\alpha = 0.1$. In the three cases, the target direction is defined by $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$

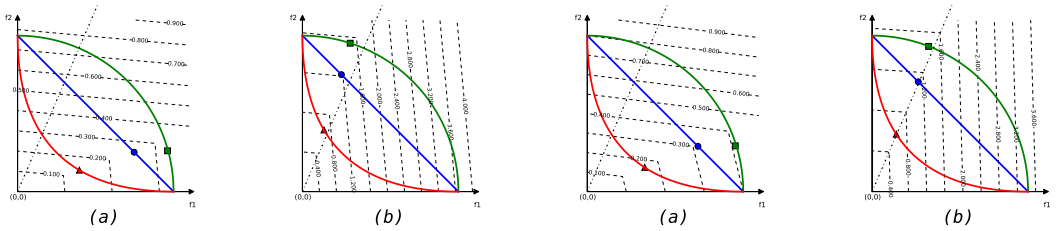


Figure 3.4: Graphical comparison of the contour lines for (a) The MCHE, (b) RMCHE, (c) ACHE, (d) RACHE functions with $\alpha = 0.1$ and target direction is defined by $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$

augmented SF based on the CHE model.

3.3 Scalarizing Functions based on Penalty Boundary Intersection model

The **Penalty Boundary Intersection (PBI)** [171] draws ideas from the Normal-Boundary Intersection (NBI) method [28], defined as follows:

$$\begin{aligned} \min \quad & g^{pbi}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \theta) = d_1 + \theta d_2 \\ \text{where} \quad & d_1 = \left| \mathbf{f}'(\mathbf{x}) \cdot \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} \right| \quad \text{and} \quad d_2 = \left\| \mathbf{f}'(\mathbf{x}) - d_1 \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} \right\|. \end{aligned} \tag{3.19}$$

d_1 represents the distance between the reference point and an optimal solution to assess convergence. Similarly, d_2 defines the perpendicular distance between the reference vector and an optimal point to assess uniformity. θ is a penalty parameter that balances convergence (measured by d_1) and diversity (measured by d_2), both to be minimized [164]. The PBI function can produce uniformly distributed solutions in objective space by setting appropriate values for θ. Some studies [75, 136, 137] have provided a sensitivity analysis of PBI, indicating that the choice of a suitable θ value depends on specific features such as the Pareto front geometry [36, 48, 49], the number of decision variables and the number of objectives. More recently, some attempts have been made to adapt this parameter into MOEA/D [164].

The **Inverted Penalty Boundary Intersection (IPBI)** [136] is an extension of PBI, given by:

$$\begin{aligned} \min \quad & g^{ipbi}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \theta) = \theta d_2 - d_1 \\ \text{where} \quad & d_1 = \left| \mathbf{f}''(\mathbf{x}) \cdot \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} \right| \quad \text{and} \quad d_2 = \left\| \mathbf{f}''(\mathbf{x}) - d_1 \frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} \right\|, \end{aligned}$$

where $\mathbf{f}''$ is defined as:

$$\mathbf{f}''(\mathbf{x}) := \mathbf{z}^{nad} - \mathbf{f}(\mathbf{x}), \tag{3.20}$$

and $\mathbf{z}^{nad} = (z_1^{nad}, \ldots, z_m^{nad})^T$ is the Nadir point, i.e., $z_i^* := \max\{f_i(\mathbf{x}) \mid \mathbf{x} \in POS\}$. IPBI aims to enhance the spread of solutions in objective space and to improve the performance in many-objective problems [136]. As in PBI, θ handles the balance between d_1 and d_2. However, a solution having a large d_1 and a small d_2 is considered as a better solution. When $\theta = 0$, the behavior of IPBI is similar to WS. In [136, 137], a set of different values for θ were tested in MOEA/D for solving many-objective problems with diverse features.

Recently, Ishibuchi proposed two variations of the PBI model. The first variation is called the **Two-level Penalty Boundary Intersection (2LPBI)** [76] given by:

$$\min \quad g^{2lpbi}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \theta_1, \theta_2) := \begin{cases} d_1 + \theta_1 d_2 & \text{if } d_2 \leq d^* \\ d_1 + \theta_1 d^* + \theta_2 (d_2 - d^*) & \text{if } d_2 > d^*, \end{cases} \tag{3.21}$$

where $\theta_1 < \theta_2$ and d^* is a parameter to switch the penalty value between θ_1 and θ_2. If d_2 is smaller than d^*, a small penalty value θ_1 is used. If d_2 is larger than d^*, a larger penalty value θ_2 is used for the amount of the violation: $d_2 - d^*$. Its authors recommend to use: $\theta_1 = 0.1$ and $\theta_2 = 10$.

The value of d^* is specified by solutions in the current population as follows:

$$d^* = \alpha \frac{1}{H} \frac{1}{m} \sum_i z_i^* - z_i, \tag{3.22}$$

where α is a parameter, H is an integer parameter used for generating uniformly distributed weight vectors in MOEA/D, z_i^* and z_i are the Nadir and ideal values of the ith objective in the current population, respectively. In equation (3.22), the average width of the domain of each objective function is divided by H to obtain a rough estimation for the distance between adjacent solutions. The parameter α is used to examine the validity of the formulation (3.22) through computational experiments with various values of α. The idea in (3.21) is to use a small penalty value only when a solution is close to the reference line.

The second variation is the so-called **Quadratic Penalty Boundary Intersection (QPBI)** [76] and is given by:

$$\min \quad g^{qpbi}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \theta) := d_1 + \theta d_2 \frac{d_2}{d^*}, \tag{3.23}$$

where d^* is the same parameter as in (3.21), which is calculated by (3.22). The effect of the penalty parameter θ is decreased by the factor (d_2/d^*) when d_2 is smaller (i.e., $d_2 < d^*$) and increased by (d_2/d^*) when d_2 is large (i.e., $d_2 > d^*$). When $d_2 = d^*$, this formulation

is the same as the PBI function in (3.19). $\theta = 1$ is recommended.

The value of d^* in 2LPBI and QPBI is calculated using information of the current population from the MOEA.

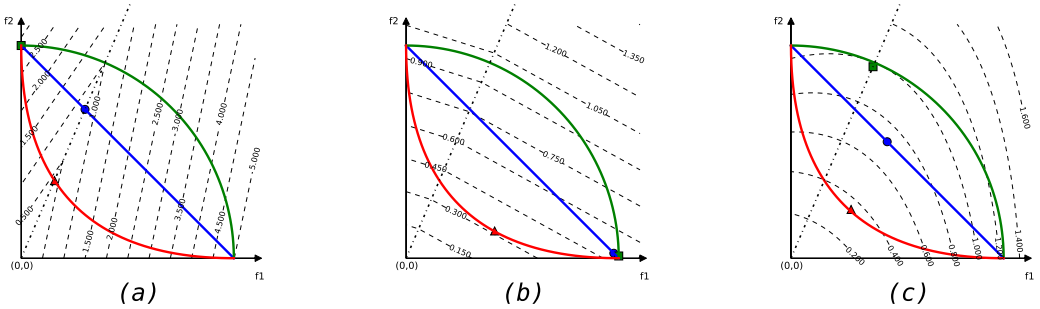


Figure 3.5: (a) The PBI function with $\theta = 1.0$, (b) the PBI2L model with the parameters recommeded by its authors ($\theta_1 = 0.1$, $\theta_2 = 10.0$) and (c) the PBIQ with $\theta = 1.0$. In the three cases, the target direction is defined by $\lambda_1 = 0.3$ and $\lambda_2 = 0.7$

Figure 3.5 shows the effect on parallel coordinates for each of the PBI versions described.

3.4 Exponential Scalarizing Functions

The **Weighted Product(WPR)** [150, p. 9], also called *product of powers*, is defined as follows:

$$\min \quad g^{wpr}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}') = \prod_i \left(f_i'(\mathbf{x})\right)^{\lambda_i}. \tag{3.24}$$

This SF has been integrated with several ant colony optimization algorithms [3], and it has also been applied to solve a network design problem [13]. However, this approach has not been widely used with MOEAs. In Chapter 6, we couple this function into MOEA/D and MOMBI-II to assess its performance.

The **Exponential Weighted Criteria (EWC)** [4] can deal with any Pareto front shape, and is given by:

$$\min \quad g^{ewc}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', p) = \sum_i \left(e^{p\,\lambda_i} - 1\right) e^{p\,f_i'(\mathbf{x})} \tag{3.25}$$

A large value of p is required to achieve Pareto optimality, but this can lead to numerical overflow [105]. In [18], EWC was used to solve a problem related to a voltage distribution network. To the best of our knowledge, this scalarizing function has not been integrated into any MOEA until now.

The **Vector Angle Distance Scaling (VADS)** [67] can discover solutions in concavities that may appear as discontinuities in the Pareto front given by:

$$\min \quad g^{vads}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', p) = \frac{\left\|\mathbf{f}'(\mathbf{x})\right\|}{\left(\frac{\boldsymbol{\lambda}}{\|\boldsymbol{\lambda}\|} \bullet \frac{\mathbf{f}'(\mathbf{x})}{\left\|\mathbf{f}'(\mathbf{x})\right\|}\right)^p}. \tag{3.26}$$

Here, the numerator measures convergence, whereas the denominator measures the deviation of the objective vector from the weight vector. Thus, the final solution should be lying parallel to $\boldsymbol{\lambda}$. Orthogonal vectors require special care. Small values of p hinder the search of sharp concavities. Its authors recommend to use $p = 100$ [67]. This scalarizing function is not compatible with any form of Pareto optimality. VADS has been implemented on an MOEA in combination with CHE [67, 68].

Regarding the parameter models, special care should be taken for a large value of p in WCP, WPO, EWC, and VADS since numerical overflow might occur.

3.4.1 Augmented and hybridized Scalarizing Functions

The **Conic Scalarization (CS)** [89] is a variation of WS, where an extra term is included for dealing with concave regions of the Pareto front, and is defined as:

$$\min \quad g^{cs}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha) = \sum_i \lambda_i f_i'(\mathbf{x}) + \alpha \sum_i \big|f_i'(\mathbf{x})\big|. \tag{3.27}$$

According to [90], CS can generate weakly Pareto optimal solutions if $\alpha \in [0, \lambda_i]$, $\lambda_i > 0$ for all $i \in \{1, \ldots, m\}$ and there exists $k \in \{1, \ldots, m\}$ such that $\lambda_k > \alpha$. Few MOEAs have adopted the CS function (e.g., [43]).

The **Dynamic Interactive Decision Analysis and Support System (DIDASS)** [54] given by:

$$\min \quad g^{didass}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \beta, \Gamma) = \max\Big\{\beta \max_i \lambda_i \big|f_i'(\mathbf{x})\big|, \sum_i \lambda_i \big|f_i'(\mathbf{x})\big|\Big\} + \sum_i \gamma_i \big|f_i'(\mathbf{x})\big|. \tag{3.28}$$

The **General Scalarizing Function (GSF)** [36] is similar to ACHE, given by:

$$\min \quad g^{gsf}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \alpha, \beta) = \beta \max_i \big\{\lambda_i \big|f_i'(\mathbf{x})\big|\big\} + \alpha \sum_i \lambda_i \big|f_i'(\mathbf{x})\big|. \tag{3.29}$$

$\beta \geq 0$ and $\alpha \geq 0$. This scalarizing function covers the special cases of WS ($\beta = 0$, $\alpha = 1$) and CHE ($\beta = 1$, $\alpha = 0$).

The **Normalized Scalarizing Function (NSF)** [36] is derived from GSF, given by:

$$\min \quad g^{nsf}(\mathbf{x}|\boldsymbol{\lambda}, \mathbf{f}', \delta) = (1 - \delta) \max_i \big\{\lambda_i \big|f_i'(\mathbf{x})\big|\big\} + \delta \sum_i \lambda_i \big|f_i'(\mathbf{x})\big|, \tag{3.30}$$

$\delta \in [0, 1]$.

In Table A.1, we summarize the SFs previously described. The column *"support"* indicates if the SF is Pareto compliant ($\prec$) or weakly Pareto-compliant ($\preceq$), and if it can generate solutions along convex (x), concave (c) or linear (l) Pareto fronts. The notation $||$ means that the optimal objective vector $\mathbf{y}^*$ is nearly parallel to the weight vector $\boldsymbol{\lambda}$. The dot product is symbolized as $\bullet$, the absolute value of a real number is denoted by $|\cdot|$, and $\|\cdot\|$ represents the magnitude of a vector. For all the scalarizing functions $g(\mathbf{y}; \boldsymbol{\lambda})$, $\mathbf{y}$ can be one of the transformations in (3.3) or (3.4), except for IPBI that adopts (3.5). In all cases,

their computational complexity is $O(m)$.

In the next chapter, we present an overview of various parameter setting techniques that have been applied to MOEAs. The first part presents statistical methods and tools employed to adapt the MOEA parameters in an offline manner. On the other hand, the second section examines different mechanisms to adapt parameters in an online manner.

Chapter 4

Parameter Setting Techniques

One of the most significant challenges in the evolutionary computing field is to identify the most appropriate parameter settings of a metaheuristic with the goal of solving a large number of problem instances. The search for the best parameter values of an Evolutionary Algorithm (EA) is called *parameter setting* or *tuning* [39]. This task could be done before the execution of the algorithm (*off-line*) or during the execution itself (*on-line*).

The design process of an EA involves the choice of several components such as the encoding, the population size or topology, the selection mechanism, the evolutionary operators and their rates of use. We use the term *configuration of a target algorithm* when these components are instantiated and properly tuned. Each parameter has an important influence on the performance of an EA both in terms of its rate of convergence and on the quality of the solutions obtained. In other words, EAs are sensitive to the value of their parameters to improve their performance in a significant way.

Based on the taxonomy proposed by Eiben et al. in [38], this chapter describes two main possibilities for dealing with the parameter setting problem: the offline parameter tuning and the online parameter control strategies. The former refers to select a set of parameter values which can be established by hand according to the user's experience, by analogy or applying experimental design methods. In these cases, the same set of parameters is used in all the iterations of an EA. Alternatively, the second (the online parameter control) involves the use of *deterministic rules, adaptive or self-adaptive* strategies whose goal is to modify the parameter values during the evolutionary search process.

On the other hand, when a metaheuristic uses deterministic rules, its parameters are changed over time, after a certain number of generations. Such rules do not consider any feedback obtained from the search. On the other hand, adaptive methods use information gathered from the evolutionary search process during a time window, through indicators

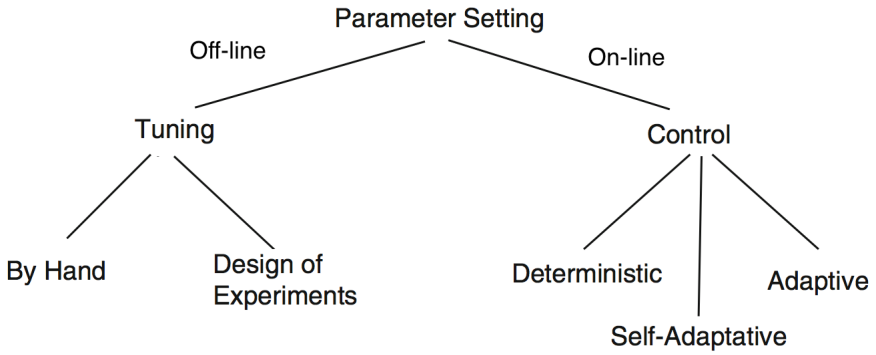


Figure 4.1: A taxonomy of parameter setting strategies.

that monitor the performance of the EA. An example of adaptive control is the so-called $\frac{1}{5}$ success rule proposed by Rechenberg [130] to modify the step-size of the evolution strategies. Finally, in self-adaptive control, the parameter values are encoded within each individual from the population, and evolved at the same time that the optimization problem is solved. In Figure 4.1, we show a graphical representation of the aforementioned taxonomy.

Regardless of the parameter setting strategy adopted, it has to deal with the following difficulties:

- The parameters are sensitive to variations in the problem's features. That is, an algorithmic configuration is able to solve one class of problems, but it does not work with others.
- The parameters may be strongly correlated between them, which means that finding appropriate parameter values requires to solve a combinatorial problem with very high computational effort.
- To estimate the performance for an algorithmic configuration implicates to deal with the noise generated by its stochastic procedures. This means that two independent runs of a metaheuristic will produce different behavior and, consequently, different results [131].

The tuning methods should deal with at least two types of parameters [63]:

1. *Categorical or unordered parameters* are those whose domains constitute a set of discrete values, each one representing an option for the component of a metaheuristic. For example, in MOEAs with continuous variable space, the crossover operators can be of the following types: Simulated Binary (SBX), Differential Evolution (DE), Parent-centric (PCX) or Unimodal Normal Distribution (UNDX).
2. *Numerical parameters* are those whose domains belong to $\mathbb{N}$ or $\mathbb{R}$. In this case, it is possible to define distance measures between any pair of values. This means that it is necessary to determine the lower and upper bounds. Also, when the parameters are in $\mathbb{R}$, it is necessary to define an accuracy level (number of decimal positions).

There is a particular class known as *conditional parameters* which are parameters that are only active when some specific parameters are selected into the EA.

In single-objective optimization, some parameter setting techniques are based on fitness measures or fitness landscape analysis. However, in Multi-Objective Optimization Problems (MOPs), the design of these mechanisms is more complex due to the difficulty to define quality measures for sets of solutions. In this thesis, we are interested in both offline and online parameter settings techniques in the context of multi-objective evolutionary algorithms (MOEAs). Additionally, we are interested in analyzing the impact of the parameter values on the performance of an MOEA based on decomposition with the aim of gaining a better understanding of the behavior and robustness of MOEAs when solving certain types of problems, thus enhancing their applicability. The following sections present an overview of the most representative parameter settings methods that have been developed for metaheuristics.

4.1 Offline parameter tuning

Offline parameter tuning is computationally expensive because of the large number of configurations to be tested. However, it is useful for deriving knowledge about the relationships among parameters involved in an EA. Offline tuning can tackle the problem from two perspectives: 1) *The specialization of algorithms* (given an algorithm's configuration, the aim is to find a subset of optimization problems on which the algorithm obtains good results) and 2) *the generalization of algorithms* (the aim is to find an algorithmic configuration to solve the major quantity of problems with different features [39]). Both approaches are related to the so-called No Free Lunch (NFL) theorem which indicates that, under certain assumptions, no stochastic optimization algorithm can be superior to the others on all possible classes of optimization problems [163].

The offline parameter tuning problem can be formalized as that of finding a 6-tuple $T = (A, P, C, O, \psi, b_{max})$ where:

- A is a target algorithm to be tuned. In this thesis, we concentrate on decomposition-based MOEAs;
- P is the set of parameters of A to be tuned. For instance, if A is an EA, then the population size, the maximum number of generations, the crossover rate and the mutation rate belong to P.
- C is the set of values for each parameter of P. An instance $c \in C$ is called a *configuration* of A;
- O is called a *scenario* and is a set of several problem instances (for example, {DTLZ1, DTLZ3, DTLZ5, DTLZ7} from the *Deb-Thiele-Laumanns-Zitzler (DTLZ)* test suite [32]);
- ψ is a scalar fitness function used by T to assess the quality of a configuration e.g., hypervolume (HV) [179], Inverted Generational Distance (IGD) [22] or the $R2$ [58] indicator.
- b_{max} is the maximum budget of function evaluations available to tune A on O;
- T is a single-objective hyperheuristic whose goal is to find the 'best' $c \in C$, based on ψ, for tuning A on an scenario O with b_{max} function evaluations.

In recent years, several offline tuning methods or automatic algorithm configuration tools have been designed to search the most appropriate configuration in the parameter values of the stochastic search based algorithms. These methods can be classified into four categories: Experimental design, model-based, sequential statistical testing and heuristic optimization. The following sections describe the most representative techniques from each category.

4.1.1 Experimental design

Several approaches based on statistical Design Of Experiments (DOE) such as the analysis of variance (ANOVA), the use of confidence intervals, factorial design, fractional factorial design, linear and nonlinear regression models have been adopted to find effective settings of

the parameter values in an EA. These techniques have two goals: 1) to design an experiment to collect appropriate data and 2) to analyze them using statistical methods to draw valid and objective conclusions. Both topics are closely related because the method of analysis depends directly on the design employed. Moreover, DOE techniques are used to build a predictive model of the performance of a target algorithm over a range of tuning parameter settings and a set of problem instances. For more details about these techniques, see [112] and [29].

In [1], it was proposed an algorithm called CALIBRA that employs Taguchi's methodology in a 2 level factorial design coupled with a local search procedure. In a factorial design, the first step is to select the most influential parameters that affect the performance of the target algorithm. Then, two or three critical values for these parameters are established. CALIBRA was applied in six different algorithms, but this version was limited to handle a maximum of five parameters. Algorithm 12 shows the iterative procedure used by CALIBRA.

Algorithm 12: CALIBRA's algorithm

```
1 CALIBRA (A, P, C, O)
  Input  : Metaheuristic and its parameters to tune and their critical values
  Output: A set of appropriate parameter values
2 Initialization. Perform 2^k experiments and determine the best and worst initial
  combination of levels for each parameter.
3 while not termination_criterion met do
4     Assign three levels for each parameter as the initial point for a local search.
5     while A local optima has not been reached do
6         Apply a Taguchi's fractional factorial experimental design, using the
          parameter levels from the previous iteration.
7         Use the results of the experiment to set the levels for the next iteration.
8     Update the list of solutions with the new local optimum found
```

Coy et al. [25] proposed an algorithm that combines the fractional factorial design and the gradient descent method to find effective values for six tuning parameters on vehicle routing problems. Given a small number of problems from the whole set, Coy's algorithm employs the two-level factorial design to produce parameter settings with a low and high-quality. Then, the response surface methodology is applied over the measurements found to interpolate and obtain a linear approximation that allows achieving effective parameter settings for solving the whole problem set. The procedure in detail is presented in Algorithm 13.

Ridge and Kudenko [131] implemented a predictive model using a DOE approach and the numerical optimization known as Nelder-Mead downhill simplex [125, p. 326]. This proposal used the response surface methodology to tune the parameters involved in the ant colony algorithm to solve the traveling salesman problem. The model was able to tune 9 numerical and 3 categorical parameters.

Algorithm 13: Coy's algorithm

Coy'algorithm (A, P, C, O)
Input : Metaheuristic and its parameters to tune.
Output: A set of appropriate parameter values.
Select a subset of problems to analyze from the entire set of problems.
Select the starting level of each parameter, i.e. the range over which each parameter will be varied, an amount to change each parameter.
Generate a factorial experimental design.
for *each problem in the analysis set* **do**
    Compute the parameter settings vector associated with the factorial experimental design.
    Perform five trials starting from the same five initial solutions for each parameter settings vector calculated in the previous step.
    Fit a linear model using the average distance from each set of five trial as the dependent variable.
    Find the path of steepest descent on the response surface obtained in the previous step.
    while *not all of the statistically significant parameters have reached the limit of the experimental region or a new minimum has not been found* **do**
        Compute the parameter vector associated with the path of steepest descent.
        Perform five trails using the same initial solutions adopted in step 6 and determine the average total length.
Combine the settings obtained to produce high-quality parameter values.

4.1.2 Model-based

The most representative model-based technique is the Sequential Parameter Optimization methodology developed by Bartz-Beielstein [7]. This methodology was divided into three main phases: Experiment construction, parameter optimization, and the rejection or acceptance of the statistical hypothesis. In the first step, the experimenter defines exactly what is to be studied and how the data are to be collected. The second phase includes the design of experiments and the statistical modeling and prediction to employ. Here, we need to define the next elements:

- an optimization problem and their constraints (for example, the maximum number of function evaluations),
- an initialization and termination method,
- an algorithm, and its important factors,
- a measure to assess the performance

Finally, the third step refers to the rejection or acceptance of the statistical hypothesis and to the interpretation of the results.

4.1.3 Sequential Statistical testing

The racing concept was introduced in 1997 [106] as a machine learning technique with the goal of decreasing the computational cost to estimate the quality of a set of parameter configurations. In this procedure, a set of candidate parameter configurations compete in a race over a training set of problem instances. Here, the worst configurations are identified and discarded using statistical evidence. These techniques are useful strategies when we try to tune a large number of parameters. The main idea of a racing approach is to evaluate the performance of a candidate configuration incrementally [122] allocating computational resources only in the promising configurations instead of wasting computational time in the worst configurations. In Algorithm 14, we can see the general steps involved in the race method.

Algorithm 14: The general irace procedure

```
1 The general race procedure
  Input  : P, a set of candidate parameters, O, a set of problem instances.
  Output: The best parameter configuration.
2 Create a race.
3 for each problem instance do
4 |   for each candidate parameter do
5 |   |   Create and execute run.
6 |   Remove worst candidates.
7 Return the best parameter configuration.
```

Birattari et al. [12] designed one of the first racing approaches known as F-race which used the nonparametric Friedman's two-way analysis of variance by ranks. In F-Race, the

statistical test was based on the ranking of the candidate parameter configurations. F-race aims to find evidence that at least one of the configurations is significantly different from others. Otherwise, Friedman tests are applied newly to eliminate the candidate configurations that are significantly worse than the best one. Iterated F-race (also known as irace) was proposed in [6]. In this approach, at each iteration, a set of new candidate configurations is built using a probability model; then, these configurations are evaluated to select the best one via the race procedure. After that, the sampling distribution is updated towards the best configurations. Algorithm 15 shows the steps used by irace.

Algorithm 15: The general irace procedure

```
1 irace procedure
2 while not termination_criterion mets do
3     Generate a set of new configurations according to a particular distribution or
      probability model.
4     Evaluate all candidates.
5     Select the best configurations using racing procedure.
6     Update the sampling distribution considering only the best configurations.
```

4.1.4 Heuristic optimization

In this section, we describe approaches based on metalevel optimization techniques that work in a search space defined by the parameters involved in the target algorithm. Grefenstette [55] proposed one of the first approaches based on metalevel optimization that implements a genetic algorithm to tune six parameters to solve single-objective optimization problems.

The so-called method for Relevance Estimation and Value Calibration of EA parameters (REVAC) proposed in [115] and [116] determines the most appropriate parameter values via a probability density distribution with maximized Shannon entropy, using an Estimation of Distribution Algorithm (EDA). This tool is able to determine the sensitivity of parameters in order to establish the recommended ranges for the target EA.

Hutter et al. [69, 70] proposed the ParamILS method which is based on a steady-state algorithm that uses an iterated local search to improve the performance of algorithmic configurations. ParamILS requires an initial configuration and a set of possible values for each tuning parameter ($\Theta \in \theta$). Algorithms 16 and 17 show the procedures employed by ParamILS. The main loop consists of a solution perturbation to escape from local optima. Then, a randomized local search procedure and an acceptance criterion are adopted to decide whether to keep or reject a newly obtained candidate solution. ParamILS is very sensitive to the initial value used. The use of different values generates different results. For that reason, this tool is used to improve the best configuration previously known.

In this section, we describe in detail a tool called EVOCA [132]. We focus on this tool because it was employed to tune decomposition-based MOEAs in the next chapter. EVOCA consists of a steady-state EA able to work on categorical and numerical parameters at the same time without requiring an in-depth knowledge of parameter tuning methods. EVOCA has been successfully applied to both the design and calibration of a couple of different metaheuristic algorithms such as: a multi-objective immune algorithm for solving ZDT and

Algorithm 16: The paramILS procedure

```
1  ParamILS procedure ()
   Input  : Parameter configuration space Θ, neighborhood relation N.
   Output: The best parameter configuration θ found.
2  Initialization.
3  θ_0 ← default parameter configuration θ ∈ Θ.
4  for i ← 1...R do
5      θ ← random θ ∈ Θ.
6      if θ is better than θ_0 then
7          θ_0 ← θ.
8  θ_ils ← IterativeFirstImprovement(θ_0, N).
9  while not termination_criterion mets do
10     θ ← θ_ils.
11     Perturbation.
12     for i ← 1...s do
13         θ ← random θ' ∈ N(θ)
14     LocalSearch
15     θ ← IterativeFirstImprovement(θ, N).
16     AcceptanceCriterion
17     if θ is better than θ_ils then
18         θ_ils ← θ.
19     with probability p_restart θ_ils ← random θ ∈ Θ.
20     returns the best θ found.
```

Algorithm 17: The paramILS procedure

```
1  Procedure IterativeFirstImprovement (Θ, N)
   Input  : Parameter configuration space Θ, neighborhood relation N.
   Output: The best parameter configuration θ found.
2  repeat
3  θ' ← θ.
4  for θ'' ∈ N(θ') in randomize order do
5      if θ'' is better than θ' then
6          θ ← θ''.
7          break.
8  until θ' == θ.
9  return θ.
```

DTLZ problems [111], the multi-objective SMS-EMOA for solving the DTLZ and WFG problems [108], component selection of MOEA/D and MOMBI-II for solving the Lamé supersphere problems [124]. In the following, the EVOCA's components are described.

Representation. The chromosome is represented by a string where each element corresponds to a parameter, and its value is taken within the parameter domain. Thus, the string length is the number of parameters to be tuned.

Initial Population. The initial domains for categorical parameters as well as for parameters belonging to discrete domains are directly their domains themselves. The population size is computed considering the number of parameters to be tuned and their initial domain sizes. The key idea is to include all the values allown for each parameter, in an independent way, on the first population. This is, in most cases, possible for categorical parameters and for parameters belonging to reduced discrete domains. In other cases, the interval is divided according to their precision value. For instance, to tune a classical crossover probability, we can decide to include all values in $\{0.0, 0.1, \ldots, 0.9, 1.0\}$ on the initial population. The size of the population is equal to the largest parameter domain restricted by a maximum MP. Algorithm 18 shows the procedure.

Algorithm 18: Population initialization

```
1 Generate Initial Population (P_c, P_n, precision_n, IDS, MP)
  Input  : Categorical and numerical parameters, precision and initial domain sizes
  Output: Initial population
2 for each parameter j do
3     L_j = List of parameter j values using IDS_{P_j}.
4 Population size = min{max_{j=1,...,m+k}{IDS_{P_j}}, MP}.
5 for each gen j do
6     for each chromosome do
7         Take one value in a cyclic order from the list L_j
```

Operators. The algorithm uses two operators: a wheel-crossover operator that constructs one child from the whole population. It uses a roulette wheel procedure [50] to select the value of the gene of each offspring, as shown in Algorithm 19. Thus, this operator is focused on inheriting good parameter values among all the values that are available in the population. The child generated replaces the worst individual on the current population. The crossover procedure is performed at each iteration, thus it does not have an associated probability. The mutation operator is a hill climbing first improvement procedure, which takes a copy of the child generated by the crossover operator and tries to improve it by modifying one of its parameter values as shown in Algorithm 20. The mutation operator is always applied. When a numerical parameter is selected, it tries to randomly take a new value within a continuous range that represents the parameter domain. In the example in the above section, related to the crossover probability, the mutation operator considers its interval from 0 to 1 as a continuous one. Thus, using mutation, the algorithm can include a new value within this range, which has not been involved in the initial population. The child generated by applying mutation replaces the second worst individual on the current population, in case it obtains a better performance. At each iteration, at most two individuals are changed: One by the crossover operator, and the second one by the mutation

operator. Thus, there are no probability values to be defined for these operators.

Algorithm 19: Crossover operator

```
1 Wheel-crossover (Population)
  Input  : Population
  Output: Child
2 Construct roulette wheel using fitness of the Population.
3 for j ← 0 to n + k do
4     Child[j] = Randomly select a gene value using roulette wheel.
```

Algorithm 20: Mutation operator

```
1 Mutation (Child)
  Input  : Child
  Output: Mutated-Child
2 Mutated-Child=Child.
3 m = int_rand(1, m + k)
4 if P_j is real then
5     Mutated-Child[j] = float_rand(d_il, d_iu)
6 else
7     Mutated-Child[j] = int_rand(d_il, d_iu)
```

The Algorithm. EVOCA uses a Latin hypercube design instead of a uniform random sampling for generating an initial population. Most of the well-known tuners require to make decisions about how to discretize the domain of the numerical values to be used as input or about the data range, as well as, the definition of many parameter values for themselves. Moreover, their performance strongly depends on these set-up decisions. Therefore we pay special attention to both the definition of the input data required by EVOCA, and its initialization step. EVOCA includes a local search procedure used by the mutation operator. Crossover allows the combination of the parameter values on the population and is focused on the target algorithm performance. Algorithm 21 shows the procedure. In our algorithm, the parameter R is used to define the number of seeds used for evaluating the parameter configurations. Given the stochastic nature of the metaheuristics, a value bigger than one for R is recommended in order to have more reliable information about the algorithm's behavior. Moreover, the parameter MP restricts the maximum population size.

This thesis uses the EVOCA tool because it provides several advantages that are described in Chapter 5.

Algorithm 21: EVOCA

```
1  EVOCA (M, P_c, P_n, precision_n, D, MP, R)
   Input  : Metaheuristic and its parameters to tune, their domains and their precision
   Output: Population
2  for j ← 0 to n + k do
3      Compute IDS[j] from D_j.
4  Population = Generate Initial Population (P_c, P_n, precision_n, IDS, MP).
5  Evaluate each configuration in Population on M using R random seeds.
6  while not termination_criterion met do
7      New-Population = Population.
8      Child = Wheel-crossover(Population).
9      Evaluate Child on M using R random seeds.
10     Replace the worst chromosome by Child in New-Population.
11     Mutated-child = Mutation(Child).
12     Evaluate Mutated-child on M using R random seeds.
13     if Mutated-Child is better than child then
14         Replace the second worst chromosome by Mutated-child in New-Population.
15     Population=New-Population.
```

4.2 Online parameter control

Since the origins of evolutionary computing, several parameter control techniques have been proposed to establish appropriate configurations at different stages of the search process. [169] and [88] present comprehensive overviews of the trends and challenges for employing parameter adaptation mechanisms in EAs. Both articles consider the following main aspects that should be taken into account:

1. To define *the adaptation objects*. This means to define which components or parameters from the EA will be adapted. For example, the evolutionary operators, population structure or selection procedures.

2. The type of *adaptation methods* to employ. For instance, the use of deterministic rules, co-evolution strategies or machine learning methods.

3. To determine which *adaptation evidences* will be used to feedback the adaptation methods. Some examples are the fitness values or population distribution. Moreover, the adaptive strategy can focus on a specific scope or level of change. That is, to monitor evidence of changes in the population- or individual-level. In the case of MOEAs, we can involve performance indicators.

The following sections describe proposals that adopt diverse adaptation methods to tune low- and high-level components into MOEAs. We present a general overview of two classes of adaptive strategies: those based on statistical rules and those based on machine learning methods. Moreover, we focus on the use of adaptation techniques coupled to MOEAs based on decomposition because this thesis provides techniques to adapt their scalarizing function component.

4.2.1 Adaptation in low- and high-level components of MOEAs

We can classify the parameters in MOEAs as low- and high-level components. The low-level components refer to a specific parameter such as the population size, the stopping termination criteria, the percentage of use for the evolutionary operators or their model parameters which commonly present a sensitivity influence in the performance of algorithms. For example, the Differential Evolution (DE) operators involve the factors F and Cr for controlling the mutation and crossover rates, respectively. Whereas the high-level components are defined in a certain stage of the evolutionary algorithms and commonly include more than one parameter. Examples of high-level components are the variation operators, the selection or replacement mechanisms, the local search procedures, the techniques of archiving, or the performance indicators to guide the search. Even, we can consider the parameter tuning using whole algorithms; this is also known as the design of hyper-heuristics. This section presents methods to control low and high-level components into MOEAs.

Online stopping termination criteria. Typically, the stopping termination criteria for MOEAs are determined previously by establishing a maximum number of function evaluations or a desired performance indicator level. Nevertheless, the use of online stopping criteria mechanisms can avoid wasting computational effort. Furthermore, they can be used to determine the application of the local search procedures. Because the majority of MOPs do not have available the gradient information and considering the stochastic nature of MOEAs, in the last years, several online stopping criteria proposals have been developed. Basically, the methods for convergence detection adopt the use of multiple performance indicators and nonparametric statistics (with confidence intervals). Tobias Wagner et al. [158] presented an analysis of several strategies of convergence detection using a formal taxonomy definition that compares aspects such as progress indicators and evidence gathering processes. Guerrero et al. [56] proposed an online stopping criteria mechanism for MOEAs combining three performance indicators: the hypervolume, the ϵ indicator and Mutual Domination Rate (MDR), where at least 2 of them must generate a convergence value in the Kalman filter to conclude that the stopping criterion has been reached. This proposal was performed on the DTLZ test functions and different versions of algorithms such as NSGA-II [31], SPEA2 [178] and PESA [24]. Roudenko and Schoenauer [133] introduced a measure based on the density of the non-dominated solutions as online stopping criteria for the NSGA-II algorithm. This method used the difference between the maximum and minimum value of the crowding distance during several generations. The drawback of this mechanism is the definition of the number of generations to detect stability.

The dynamic population size in MOEAs. The effectiveness and efficiency of evolutionary algorithms also depend on selecting an appropriate population size at each generation. The main advantages of incorporating an adaptive population size method is that it reduces the computational effort due to a large number of individuals and that it prevents premature convergence. There are many proposals of this sort in single-objective optimization but almost none in multiobjective optimization. Tan et al. [147] proposed the incrementing multiobjective evolutionary algorithm (IMOEA), which adapts the population size based on the distribution density discovered at each generation of the Pareto front. IMOEA involves the next components:

- a predefined set of preferences,

- the lower and upper bounds for the desired population size,
- the desired population size per unit volume, and
- a fuzzy boundary local perturbation procedure.

The population size is increased or decreased dynamically through a fuzzy local procedure that perturbs the set of nondominated solutions according to the distribution density in the discovered hyper-area and the desired population size. The preliminary results indicated that is beneficial to initialize the process of evolution with small population size. Then, this size can be adaptively increased or decreased in subregions of objective space via the computation of a progress measure that considers the distance between the discovered nondominated solutions and the Pareto optimal set.

Adaptation of multiple search operators. There exists diverse variation operators whose efficiency strongly depends on the MOP's characteristics. Vrugt and Robinson [157] and Vrugt et al. [156] introduced an adaptive multi-operator hybrid called AMALGAM. This proposal allocates resources dynamically to each search operator based on their individual performances. The first version of the AMALGAM for multi-objective optimization employed simultaneously the NSGA-II, particle swarm optimization (PSO), adaptive metropolis search (AMS), and differential evolution (DE). Each strategy has a probability to generate offspring solutions, which is updated according to the ranking method used by the fast nondominated sorting algorithm [31]. In 2015, Mashwani et al. [107] developed an enhanced version of AMALGAM that combines DE, PSO, SBX, SPX, and the Pareto archived evolution strategy (PAES [91]). This work introduced a new adaptive resource allocation procedure based on a reward method that calculates the number of successful solutions produced at every generation of the evolutionary process.

Adaptation techniques in various components. Toscano and Coello [149] presented the μGA^2, one of the first proposals that adapts all its components automatically in an online manner, which does not require any parameter fine-tuning. The μGA^2 adopts three micro-genetic algorithms (each of them with five individuals) associated with different crossover operators (SBX, two-point crossover and a crossover operator proposed by its authors). The three micro-genetic algorithms are executed in parallel to monitor their performance via the number of non-dominated solutions generated. The micro-genetic algorithm with the worst performance replaces its crossover operator by the one with the best performance. The μGA^2 includes an adaptive grid and two population memories: one which is replaceable and one which is not. The first one is not modified to provide diversity to the algorithm while the second one is evolved during the search process. μGA^2 modifies the percentages of crossover and mutation through two phases: *an exploration process* that uses a higher percentage of mutation than the crossover operator (50%) and *an exploitation process* that decreases the percentage of mutation while the percentage of crossover is set to 100%. Finally, μGA^2 adopts an automatic stopping criterion reached when none of the internal micro-GAs can improve the solutions previously attained.

David Hadka and Patrick Reed [57] designed the so-called Borg MOEA that employs three different adaptive strategies:

1. The $\epsilon-$progress measure to monitor the progression and stagnation of the algorithm.

2. An adaptive tournament size to maintain elitist selection.

3. An auto-adaptive multi-operator recombination composed by SBX, DE, PCX, SPX, Unimodal Normal Distribution crossover (UNDX), and Uniform Mutation (UM).

Borg adopts a similar mechanism as the AMALGAM algorithm [157]. Given K operators, each operator has a probability of producing the next offspring. These probabilities are updated by counting the number of solutions produced by each operator in the $\epsilon-$box dominance archive [93].

In recent years, the concept of *auto-configuration of MOEAs* is related to the combination of high-level components in MOEAs to design automatically new algorithms. Bezerra et al. [10] proposed the automatic component-wise design method for solving diverse scenarios using the irace tool. This work combines the components of MOEA/D and SMS-EMOA. In 2016, Bezerra et al. [9] used MOGA, NSGA-II, SPEA-2, IBEA, HypE and SMS-EMOA. Both proposals adopt the following methodology:

- identify individual algorithmic components in different MOEAs that have the same function.
- To adapt the design of MOEAs to their particular application scenario.
- To instantiate a larger number of MOEAs from the same algorithmic template.
- Evaluate each new algorithmic instance in a particular scenario.
- Obtain the most appropriate configurations.

Table 4.1 shows an example of the classification of common components.

Table 4.1: An example of the common components.

Component	**Parameters**
Variation operators	<SBX, PM, DE, SPX, CMX>
Selection mechanism	<random, priority_structure, tournament>
Replacement mechanism	<neighborhood, entire_population, resource_a>
Scalarizing functions	<WS, PBI, CHE, ASF, AASF>

4.2.2 Adaptation based on rules and statistics

Adaptation based on entropy measures. In adaptive parameter control for EAs, the entropy-based measures are useful to discretize parameter value ranges. In this case, at each iteration, the information gathered from the search process is classified into groups of parameter values that were found to be successful or unsuccessful. In [2], entropy-based clustering was used to identify the parameter ranges of an EA used to solve the quadratic assignment problem with constraints maps. In this work, given a set of parameters, $p_1, ..., p_n$, where each parameter p_i has $v_{i1}, ..., v_{im}$ values that can be discrete numbers or intervals of continuous numbers, parameter control based on entropy has the task of deriving the optimal next value v_{ij} to optimize the influence of p_i on the performance of the algorithm.

The cross-entropy method employed for solving MOPs involves the estimation of parameters for a number of probability distributions. This method was used in [152] and [47] where the general idea is twofold. Firstly, to cluster the nondominated solutions on the Pareto front to adapt the probability distribution parameters. Secondly, the probability distribution functions of the parameters are updated on the basis of a performance metric on the generated sample, to produce a better sample at the next iteration.

The **self-adaptive control** techniques in EAs consist in encoding the parameter values directly into the representation of an individual. These methods have been widely studied in single-objective optimization [94, 142] but scarcely in MOPs. In 2015, Qiu et al. [128] presented one of the most representative self-adaptive methods coupled to a Differential Evolution-based MOEA to adapt the scaling factor (F) in an objective-wise manner. This proposal implemented independently the adaptive DE variants previously used in the tuning methods applied in jDE [128], JADE [170], and DESAP [26]. This proposal was coupled to NSGA-II and MOEA/D, following the next procedures:

1. The use of a ranking method for each individual using an estimation of groups.
2. A self-adaptative scheme, where each solution stores its corresponding successful F values.
3. An estimation of the arithmetic mean to control parameter values associated with the members in one estimation group.

The **Covariance Matrix Adaptation (CMA)** algorithm [118], [59] and its variants [5], [14], have been widely used to control various aspects of the variation operators, mainly the mutation step size in Evolution Strategies (ES), because the performance of ES depends on a suitable choice of the internal strategy for parameter control. In these proposals, CMA-ES is a technique that records the mutation step size values to monitor correlations at each generation during the evolutionary search.

One of the first CMA-ES employed to solve MOPs was proposed by Igel in [71]. The selection mechanism of this approach was based on nondominated sorting and the hypervolume contribution was used as a second sorting criterion coupled to an elitist variant of the single-objective CMA-ES. Here, the main contribution was that the algorithm's performance is preserved when the search space of an MOP is affected by variants such as translation, rotation, or rescaling, which is important to generalize the results obtained on benchmark functions to real-world problems. This pioneering work has been taken as a basis to develop other multi-objective CMA-ES variants, see for example [72] and [95].

Coevolutionary methods for parameter tuning use a procedure to generate new parameter values and incorporate them into the evolutionary process. Coevolutionary algorithms can be seen as some sort of reinforcement learning approaches inspired on the inter-species interactions. Based on this, a coevolutionary algorithm manages two or more populations that evolve simultaneously to solve a sequential decision problem. In other words, coevolution occurs when the fitness function of one individual depends on that of other individuals. There are at least two types of coevolutionary algorithms: competitive and cooperative [169]. Both have the goal of improving the convergence towards the global optimum. A coevolutionary algorithm used as a high-level evolutionary method was presented

in [141], where the use of multi-populations allows to implement different evolutionary operations into a multi-objective immune algorithm.

4.2.3 Adaptation using machine learning techniques

Reinforcement Learning (RL) is a simulation-based method, which is suitable for optimization through online learning. RL is employed as the main part of the Adaptive Optimization Algorithm [37].
In Evolutionary methods, RL can involve an agent in a certain environment and receive reward or punishment for certain behavior [153]. The principal structure for RL is a Markov Decision Process, where its basic elements include states, actions, and rewards. These elements can be defined as follows:

- A **state** is constructed as values of a set of parameters attached to each objective function.
- An **action** is considered as the direction for tuning a set of parameters related to each objective function.
- The definition of **reward** refers to credit assignment according to the performance reached by the parameter settings.

Definition 4.2.1 *Mathematically, let a set* $S = \{s_1, ..., s_N\}$ *be the state space of a finite Markov chain and* $A = \{a_1, \ldots, a_r\}$ *the action set available. Each combination of starting state* s_i*, an action choice* $a_i \in A_i$ *and next state* s_j *has an associated transition probability* $T(s_j, s_i, a_i)$ *and reward* $R(s_i, a_i)$*. RL has the goal of learning a policy* π*, which maps each state to an action so that the expected discounted reward is maximized.*

In [153], two RL techniques: Q-learning and the SARSA algorithm [146] were employed in a MOEA. The hypervolume indicator is adopted as an action selection strategy to decide in an online manner between the use of two different scalarizing functions: a weighted sum and Chebyshev functions.

The **Fuzzy Logic Controller** (FLC) is a technique that has been used as a tool to control in an online manner the parameters in an EA. There exist many FLC proposals in single-objective optimization [62] to control the evolutionary operator rates and population size with the aim of avoiding premature convergence and improving the algorithm's performance. The steps needed to design an FLC for parameter settings in EAs are the following:

1. To define the inputs and outputs. The first one refers to the performance measures that describe the EA's behavior, and the second indicates the values or changes in the control parameters.
2. To determine a *data base* which includes the membership functions of the fuzzy sets to specify the meaning of the linguistic terms.
3. To obtain a *rule base* to represent the expert's knowledge. Here, we have two possibilities: using the experience of the experts in EAs or using an automatic learning technique.

[83] presented a proposal that combines FLC into the Multi-Objective Differential Evolution (MODE) algorithm to tune the parameters such as the greediness and perturbation factor involved in MODE. The performance measures used by the fuzzy membership function were population diversity and the percentage of the total number of generations that have already passed.

The Self-Organizing Maps (SOM) [92] are a class of artificial neural networks which have proven to be a valuable tool in analysis and visualization of high-dimensional data. SOM is an unsupervised method which uses competitive and cooperative training for clustering, and performs a nonlinear mapping from a high-dimensional input space onto a two-dimensional grid. In [16], SOM was employed to learn from the evolutionary path to adapt the mutation step size. That is, to focus on areas that had promising solutions in order to speed up convergence. This approach approximates the Pareto front with a SOM, by modifying its training algorithm. The topology of the SOM defines the selection of solutions.

The **Adaptive Operator Selection** (AOS), is a paradigm to select adaptively which evolutionary operator should be applied at each instant of the search process. The AOS procedure includes two main components:

1. The credit assignment scheme that refers to assessing the performance of each operator based on the progress of its application in the target algorithm. The goal is to reward the best ones.

2. The operator selection rule which is responsible for selecting from among the best operators according to their rewards.

In AOS techniques, a continuous observation of the performance of the operators is monitored to balance exploration and exploitation actions during the search. The exploration concept means to evaluate new solutions in a large search space. On the other hand, exploitation means to focus the search to promising regions where the global optimum could be located. Formally, we have a set of k operators $O = \{o_1, \ldots, o_k\}$, and a probability distribution vector $P(t) = \{p_1(t), \ldots, p_k(t)\}$ $(\forall t : 0 \leq p_i(t) \leq 1; \sum_{i=1}^{k} p_i(t) = 1)$, with associated expected values $\{\mu_1, \ldots, \mu_k\}$ and variances $\{\gamma_1^2, \ldots, \gamma_k^2\}$. The adaptive allocation rule selects an operator to be executed in proportion to the probability values specified in $P(t)$. When an operator o is applied to the environment at time t, a reward $R_o(t)$ is returned. During the evolutionary process, the values of the operator probabilities are adapted following learning rules according to the quality of the new solutions created by the operators. Next, three types of AOS are briefly described:

- *Probability Matching (PM)*, where each operator is selected by a roulette-wheel process. In this technique, simple rules called classifiers are used to identify the variance-sensitivity in the operator's performance.

- The *Adaptive Pursuit (AP)* strategy proposed in the learning automata area has been used to determine the operator's quality. Given Δt iterations of the target algorithm, this rule modifies the operator's probabilities in such a way that they match the reward distribution. The AP algorithm increases the selection probability of the best operator and decreases all other probabilities based on a learning rate in time $t > 0$. This strategy is sensitive to changes in the reward distribution.

- The *bandit-based AOS* is an upper confidence model that selects a strategy according to some determinist rule based on *Multi-Armed Bandit (MAB)* algorithms as an alternative to exploit the best operators. MAB is inspired on the next problem: given a slot machine with n arms (bandits), a gambler has to collect as much money as possible pulling these arms over many turns. This is a mathematical formalism used to study the convergence properties of Reinforcement Learning with a single state. MAB algorithms consist of two processes: the first is to assign a reward to a strategy based on its recent performance in the search process. The second is the choice of the best strategy based on these current reward values. In [17, 96, 176], there are several models based on MAB algorithms such as the $\epsilon-$greedy, Boltzmann exploration or Upper Confidence Bound (UCB).

4.2.4 Parameter Adaptation in Decomposition-based algorithms

Our interest in this thesis is to incorporate adaptive strategies in MOEAs that use SFs as their transformation function, following the guidelines employed for tuning the high-level components and involving the MAB algorithms (only used in AOS). In this section, we review adaptive techniques used to improve the performance of MOEA/D and then we concentrate on works that only adapt the SF.

During the last few years, several enhanced MOEA/D versions have adopted adaptive strategies to deal with more complicated MOPs. In this section, we describe the most relevant proposals, and we focus on the related works that use either one or multiple Scalarizing Functions (SFs) in their search process.

Zhang proposed the *MOEA/D-DRA* [172] with two crossover operators: Simplex Crossover (SPX) and Center of Mass Crossover (CMX). Every k generations, *MOEA/D-DRA* monitors a relative decrease of the objectives for each subproblem and based on this, a tournament selection strategy is employed.

Chiang and Lai proposed the *MOEA/D-AMS* [21] that improved MOEA/D-DE using two strategies:

- a controlled selection of subproblems with the goal of identifying unsolved subproblems to assign a computational effort on these, and
- an adaptive mating selection mechanism that considers the Euclidean distance between individuals in decision space instead of the distance between weight vectors.

Venske et al. [155] introduced an Adaptive Differential Evolution for Multiobjective Problems (*ADEMO/D*) incorporating a pool of adaptive mutation strategies based on the SaDE algorithm [127]. Each strategy is associated with a probability of use which is updated based on its success and failure counters.

A recent approach called *MOEA/D-GRA* [175] presents an experimental study to show the importance of resource allocation. In this work, two strategies were implemented to assign a probability for each subproblem to be computed. It works as follows:

1. *Offline Resource Allocation.* The probability of use is computed according with the number of function evaluations per subproblem using Equation (4.1).

$$p_i^{off} = \frac{\overline{FE}^i}{\max_{j=1,\ldots,N} \overline{FE}^j}. \tag{4.1}$$

where $\overline{FE}^i$ denotes the average cost consumed by subproblem i.

2. *Online Resource Allocation.* Let x_t^i and $x_{t-\Delta_T}^i$ denote the solutions of subproblem i at generation t and the $(t-\Delta T)$th generation, respectively. u^i is the utility function as the relative improvement in the last ΔT generations.

$$u^i = \frac{g^i(x_{t-\Delta_T}^i) - g^i(x_t^i)}{g^i(x_{t-\Delta_T}^i)}. \tag{4.2}$$

The probability is assigned by:

$$p_i^{on} = \frac{u^i + \epsilon}{\max_{j=1,...,N} u^j + \epsilon}, \tag{4.3}$$

where $i = 1, ..., N$, $\epsilon = 1.0 \times 10^{-50}$ is a small value to avoid dividing by zero.

In this work, the online resource allocation strategy was found to be the best option.

Zhao et al. [174] coupled MOEA/D with an Ensemble of Neighborhood Sizes, called ENS-MOEA/D. This algorithm employs a Learning Period (LP) which stores (during a certain number of generations) a success probability associated with each Neighborhood Size (NS) option. At the generation $G > LP - 1$, the probability of choosing the kth ($k = 1, 2, ..., K$) NS is updated by:

$$p_{k,G} = \frac{R_{k,G}}{\sum_{k=1}^{K} R_{k,G}}, \tag{4.4}$$

where

$$R_{k,G} = \frac{\sum_{g=G-LP}^{G-1} FEs\_sucess_{k,g}}{\sum_{g=G-LP}^{G-1} FEs_{k,g}} + \epsilon, \tag{4.5}$$

$(k = 1, 2, .., K; G > LP)$ and $R_{k,G}$ represents the proportion of improved solutions generated with the kth NS within the previous LP generations. FEs_sucess is the number of successful function evaluations.

The mechanisms based on Adaptive Operator Selection (AOS) have been promising techniques for improving the performance of MOEA/D. AOS considers an upper confidence model that selects the most suitable operator from a pool of options according to some deterministic rules based on multi-armed bandit algorithms. Bandit-based AOS establish mainly two processes: one is to assign a reward to a strategy based on its recent performance (its record of fitness improvement rates) in the search process. The second is the choice of the best strategy based on these current reward values. Li et al. [103] proposed an adaptive mechanism to select a mutation operator from differential evolution versions based on the multi-armed bandit. Qi et al. [126] proposed an adaptive mechanism to select two components of MOEA/D: a candidate operator to generate offspring and a variation of the neighborhood size for each subproblem. A bandit-based AOS is an upper confidence model that selects a strategy according to some deterministic rule based on multi-armed bandit algorithms.

Table 4.2 summarizes the general components employed by the enhanced MOEA/D versions previously described. We can notice that these algorithms implemented adaptive

strategies on their evolutionary operators, selection and replacement mechanisms, maintaining fixed the SF procedure. This thesis focuses on adaptive techniques to select an appropriate SF during the evolutionary search process of MOEA/D. Next, we present related work that adapts the SF component into the MOEA/D framework.

Table 4.2: Some improved versions of MOEA/D

MOEAD	**Variation operators**	**Selection and replacement mechanism**	**Scalarization Functions**
MOEA/D-DE [102]	DE (rand/1/bin) Polynomial based-Mutation (PM)	Neighborhood or Entire population (delta probability) A maximum number of replacements nr to control the population's diversity and the convergence speed.	Tchebycheff
MOEA/D-GM [19]	Simulated Binary Crossover SBX Guided mutation	Introduced a priority structure for subproblem indexes Q	Tchebycheff
MOEA/D-DE [65]	DE/rand/1/bin DE/rand/2/bin DE/best/1 DE/best/2 DE/rand-best/1	The same as the original MOEA/D.	Tchebycheff
MOEA/D-DRA [172]	Simplex Crossover (SPX) Center of Mass Crossover (CMX) Polynomial Mutation (PM)	Neighborhood or Entire population (delta probability) 10-tournament selection based on (π) the utility of subproblems	Tchebycheff
MOEA/D-AMS [21]	MOEA/D-DE [102]	An adaptive mating selection: Neighborhood 1: weight vectors Neighborhood 2: decision space uses a 5-tournament	Tchebycheff
ADEMO/D [155]	DE/rand/1/bin DE/rand/2/bin DE/current-to-neighborhood Polynomial mutation	Neighborhood or Entire population. (delta probability)	Tchebycheff
MOEA/D-GRA [175]	MOEA/D-DE, (Li, 2009)	Online Resource Allocation. Offline Resource Allocation. Set p (the probability that a subproblem is inverted) according to a hardness measure of subproblems in advance. Information collected from the previous search.	ASF Hybrid utility function.

Ishibuchi et al. [79, 80] presented some mechanisms to use several SFs simultaneously in MOEA/D as an option to maintain diversity and to solve many-objective problems. Ishibuchi et al. [79] presented an early work that combined the WS and CHE functions in MOEA/D. At each generation, a selection strategy monitors if k or more solutions in the neighborhood of an individual have the same objective vector. If that's the case, then CHE is adopted; otherwise, WS is adopted. In [80], two alternatives were proposed to combine the WS and CHE functions. Given a set of uniformly distributed weight vectors, two subpopulations focused on a particular SF. The second alternative is to assign, alternately, one SF to each vector. Recently, Hernández and Coello [52] proposed a hyper-heuristic where each individual in the population minimizes a different SF assigned by a heuristic selection mechanism based on the quality indicator called s-energy with the purpose of maintaining a uniform distribution of solutions. Here, seven different SFs were employed.

4.3 Summary

This chapter provided a general description of different parameter tuning techniques in the context of multi-objective optimization. We can see that there are relatively few proposals in this area, which was the main motivation for the research reported in this thesis. We have provided here an overview of these proposals, as a way of positioning the work proposed in this thesis.

Our contributions in this work are related to the implementation of offline and online tuning for SFs. Chapter 5 proposes a methodology for offline tuning of decomposition-based MOEAs with the goal of providing knowledge about the use of different SFs through a characterization of MOPs according to the Pareto front and the number of objective functions. Chapter 6 present two proposals: the first one uses a multi-armed bandit algorithm to select the most suitable SF in different stages of the evolutionary search process of MOEA/D-DRA. The second one adopts an online strategy to determine the resource allocation in the use of several SFs based on the Chebyshev model.

Chapter 5

Offline Parameter Tuning for Scalarizing Functions

Scalarizing Functions (SFs) play a crucial role in multi-objective evolutionary algorithms (MOEAs) based on decomposition and the R2 indicator. These algorithms have shown scalability to any number of objectives while having a low computational cost. The performance of MOEAs based on decomposition and R2 indicator, two key points should be considered: 1)the setting of the SF and 2)the generation or adaptation of weight vectors. Regarding the choice of the weight vectors, several attempts have been proposed for adapting them in order to get a high-quality approximation of the Pareto front (see for example [40, 151]). Several studies [36, 49, 75, 81, 124, 136, 137] have tried to provide guidelines to establish the most suitable SFs concerning the characterization of a type of MOP. However, none of these studies analyzes their ability to scale to any number of objectives. In fact, only three SFs (the Weighted Sum, Chebyshev and Penalty Boundary Intersection function) have been exhaustively researched so far [73, 74, 137], neglecting other approaches that may be able to handle many-objective problems (i.e. problems with more than three objective functions).

This chapter proposes a general methodology of offline parameter tuning for MOEAs. Then, we study weighted and unconstrained SFs which have been proposed not only within evolutionary multi-objective optimization but also in the mathematical programming literature. Additionally, we investigate their scalability up to 10 objectives, solving different test problems using the Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [171] and the Many-Objective Metaheuristic Based on the $R2$ Indicator II (MOMBI-II) [61] frameworks. For this purpose, the best suited SFs and their model parameters are determined through our proposed experimental methodology in the following three case studies:

1. To study the relationships between SFs and three basic Pareto front shapes (linear, convex and concave).

2. To analyze the convergence and distribution reached by the Chebyshev functions on various Pareto front geometries varying the number of objective functions.

3. To examine the behavior of the SFs in scenarios with more complex features such as degenerated or multifrontal Pareto fronts and many-objective optimization problems.

Moreover, we present some modifications for estimating the performance of the algorithmic configurations in the EVOCA tool using diverse robustness measures taken from the uncertainty handling area (see Section 5.3).

5.1 Proposed experimental methodology

We propose a methodology for applying an offline parameter tuning process to evolutionary multiobjective algorithms. It is an optimization strategy focused on setting the parameter values of a target algorithm in order to reach its best performance. Our proposed methodology requires the definition of at least three stages as shown in Figure 5.1. In the following, all steps involved in this methodology are described.

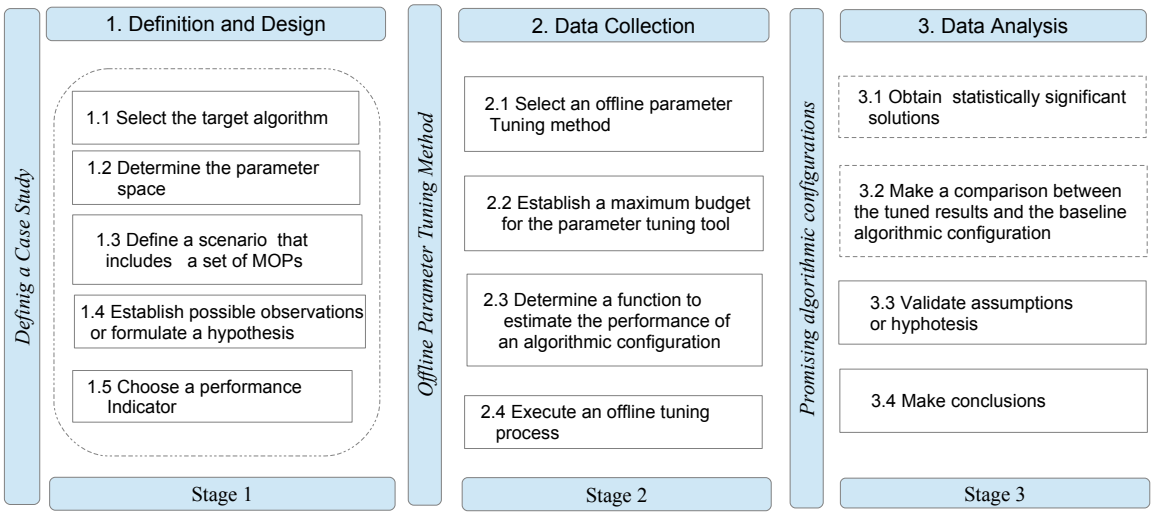


Figure 5.1: Our experimental methodology for the offline tuning process

1. *Definition and Design.* In the first stage, the goal is to define all the components related to the offline parameter tuning problem (the formal definition was explained in Section 4.1 as a 6-tuple):

   - Selection of the target algorithm, its parameters to be tuned and their respective domains. A correct analysis of parameter space plays a key role in selecting an appropriate parameter tuning method (e.g. one of the aforementioned in Chapter 4).
   - Definition of scenarios by selecting MOPs of interest. Here, it is useful to apply a classification of problems regarding particular characteristics. If we choose a set of problems having common features, we can say that a specialized configuration will be obtained to solve this type of problems. Otherwise, if we have a set of instances with various features, the calibration process is more difficult according to the No Free Lunch Theorem presented in Section 4.1.
   - Formulating a hypothesis or establishing assumptions that are relevant to extract useful knowledge about the correlations between parameter values and the characteristics of the MOPs. An example of this could be when we assume that the decision variable space influences the selection of a suitable evolutionary operator.

- Multiobjective optimization requires to establish performance indicators to assess the quality of the best solutions reached by the algorithmic configurations. Depending on the performance indicator, different aspects are evaluated such as convergence, uniform distribution, and coverage along the Pareto front. It is important to determine what are the aspects that we are interested on.

2. *Data collection.* Once we established the knowledge of our interest, we can collect the information, via the following steps:

   - Select a correct parameter tuning method according to expected observations. We need to consider the type and number of parameters to tune. For example, if we expect to improve the baseline version of a target algorithm, or we want to obtain more than one algorithmic configurations per scenario.
   - Establishing a maximum budget requires to consider different aspects: a) the number of function evaluations for the target MOEA, b) regarding the stochastic nature in MOEAs, we need to establish the number of seeds to compute descriptive statistics, and c) the budget of function evaluations available for the offline parameter tuning tool.
   - Determine a function to estimate the performance of the algorithmic configuration. The calibration process typically computes the mean of R executions using different random seeds. Moreover, it is possible to use other statistics like the median or variance. Section 5.2.3 presents other alternatives to compute this aggregating function.

3. *Data analysis.* From a set of promising algorithmic configurations, we need to filter out the most significant one according to our hypothesis or assumptions. We recommend to apply the following two optional steps:

   - To obtain statistically significant solutions from a set of algorithmic configurations using tests of nonparametric statistics such as Wilcoxon signed rank, Kruskal-Wallis, Mann-Whitney, Spearman Rank Correlation or Friedman's test.
   - Many times, the goal of applying the offline parameter tuning is to improve a baseline version previous proposed. If this is the case, we recommend making a comparison between the algorithmic configurations obtained by the calibration tool and the baseline version.
   - To validate our assumptions or hypothesis, we need to make independent runs of the algorithmic configurations with different random seeds. Then, we compute the statistics of the performance indicators.
   - Finally, we report the conclusions reached by the parameter tuning process.

We employ this methodology to perform an experimental study about the SFs adopted by the MOEA/D and the MOMBI-II frameworks as follows.

5.1.1 Definition and design

We want to identify the following observations:

- Firstly, to determine which are the SFs that are the most suitable for both the MOEA/D and MOMBI-II frameworks on different scenarios defined by varying the Pareto Front (PF) shapes and the number of objective functions.

- We want to verify if this robust set of SFs can significantly improve the performance of the baseline versions of these MOEAs.

- Our main interest is to identify the SFs that can solve a wider variety of test instances, or at least to know in which instances these SFs perform well.

All the experiments were implemented in the EMO Project framework [1] developed by Hernández and Coello [51], which is implemented in C language. Table 5.1 shows the parameters which adopted the same values in MOEA/D and MOMBI-II according to the number of objective functions (m). The weight vectors were generated using the Simplex Lattice Design (SLD) method described in [28] where H is a factor looking for a cardinality analogous to the population size (*popsize*). The stopping criterion consisted of reaching a maximum number of function evaluations of the MOP. The first study case used 50,000 evaluations. Other cases were fixed by the values presented in the column NFE. T is the neighborhood size adopted by MOEA/D. We use the same number of weight vectors as the population size. The number of decision variables is represented by n_1 for the Lamé Supersphere problems and $DTLZ1$, n_2 for $DTLZ3$ and $DTLZ3^{-1}$ and n_3 for $WFG(1, 2, 3)$.

Table 5.1: The parameters settings used by MOEA/D for each dimension m. The mark p in column H means that the original set of weight vectors generated by SLD was pruned in order to obtain the desirable population size.

m	n_1	n_2	n_3	H	*popsize*	NFE	T
2	6	11	24	99	100	40,000	20
3	7	12	26	15	136	60,000	27
4	8	13	28	8	166	70,000	33
5	9	14	30	6p	180	80,000	36
6	10	15	32	5p	200	80,000	40
7	11	16	34	4	210	90,000	42
8	12	17	36	3, 4p	230	100,000	46
9	13	18	38	3, 4p	250	100,000	50
10	14	19	40	2, 3	266	110,000	53

For MOEA/D, the probability of Polynomial-based mutation and Simulated Binary Crossover (SBX) were set to $\frac{1}{n}$ and 0.9 respectively; in both cases, the distribution index was set to 20. The parameter values employed for MOMBI-II were: record = 5, tolerance threshold = 1×10^{-3} and 0.5 for the variance threshold. The baseline version adopted in MOEA/D was CHE for 2 objectives and PBI with $\theta = 5$ for the remaining objectives [82, 171]. The baseline of MOMBI-II was ASF [61].

In summary, we used the next elements in our parameter tuning procedure:

- A : MOEA/D or MOMBI-II frameworks.

[1] Available at `http://computacion.cs.cinvestav.mx/~rhernandez`

- $P :=$ { Weighted Compromise Programming (WCP), Weighted Sum (WS), Exponential Weighted Criteria (EWC), Weighted Power (WPO), Weighted Product (WPR), Weighted Norm (WN), Chebyshev(CHE), Augmented CHE (ACHE), Modified CHE (MCHE), Achievement Scalarizing Function (ASF), Augmented ASF (AASF), Penalty Boundary Intersection (PBI), Inverted PBI (IPBI), Two-level PBI (2LPBI), Quadratic PBI (QPBI), Conic Scalarization (CS), Vector Angle Distance Scaling (VADS), General Scalarizing Function(GSF), Normalized Scalarizing Function (NSF) and Dynamic Interactive Decision Analysis and Support System (DIDASS) }with their corresponding model parameter values defined in the ranges $\alpha \in [0, 10.0]$, $p \in [0.1, 10.0]$, $t \in [1, 100]$, and $\theta \in [0.1, 50.0]$}.

- $O :=$ DTLZ1, DTLZ3, DTLZ3^{-1} from the *Deb-Thiele-Laumanns-Zitzler (DTLZ)* test suite [66] and the Lamé Superspheres (LS) test problems [42] varying the parameter $\gamma = \{0.3, 0.5, 1.0, 2.0, 4.0\}$ with the goal of achieving diverse Pareto front shapes. We classify our test MOPs in different scenarios according to their Pareto front geometry and number of obejective functions.

- $\psi :=$ the *normalized hypervolume* (NHV), defined as:

$$NHV(A) := \frac{HV}{\prod_i r_i}, \tag{5.1}$$

  where $HV(\mathcal{A}, \mathbf{r}) = \mathcal{L}\big(\cup_{i=1}^{\mu} [\mathcal{A}^{(i)}, \mathbf{r}]\big)$, is the hypervolume indicator [177]. $\mathcal{L}$ measures convergence towards the PF and maximum spread through the union of hypercubes formed by all non-dominated elements in $\mathcal{A}$ and a reference point $\mathbf{r} := (r_1, \ldots, r_m)^T$. A high NHV value is better.

  We adopted the NHV performance indicator to make a fair comparison among the problems predefined in each scenario. NHV requires to calculate the exact HV. For the Lamé Superspheres problems, it was computed by the formulation given in [42] and [148]. In the same manner, we consider that DTLZ1 has the same hypervolume to Lamé with $\gamma = 1.0$ divided by two, DTLZ3 is equal to Lamé with $\gamma = 2.0$ and DTLZ3 convex is Lamé with $\gamma = 0.5$. For the WFG instances, the maximum hypervolume was computed using a sampling of many Pareto optimal solutions.

5.1.2 Data collection

The EVOCA tool was selected as our offline parameter method because it offers the next advantages:

1. It does not require additional specialized information to adapt any stochastic algorithm.

2. It can deal with numerical, categorical and conditional parameters. The last type of parameters was very important for the experimental design adopted in this thesis.

3. Regarding numerical parameters, EVOCA can include a zero value for tuned parameters. This is useful to identify the parameters that do not provide an improvement in the target algorithm. In the next chapter, we show a case in which we adopt scalarizing functions with model parameters equal to zero.

4. We have had collaborative projects with EVOCA's authors, which implies that we can easily modify and adapt its source code as many times as required.

In EVOCA, an individual represents a calibration that involves one of the SFs described in Chapter 3, and the set of all the model parameters. In order to achieve accurate results, we consider that all the model parameters are real values.

The maximum budget ($b_{max} :=$) used by EVOCA is $10,000$ function evaluations and ten seeds per test problem at each generation. The tuning process was independent for each scenario.

5.1.3 Data Analysis

At the end of a tuning process, EVOCA returns a set of algorithmic configurations, corresponding to the best found. These configurations are executed 30 times independently. Then, these are filtered using the Wilcoxon rank sum test (one-tailed) (with a confidence level of 99%) and a ranking process for each configuration is applied. We compute a ranking process by pairs, considering three possibilities: a) if a calibration significantly outperforms another one, we add 1, b) if the calibration is outperformed by another one, we subtract -1, c) otherwise this is a tie that does not affect the rank value.

Finally, the most significant configurations are compared with the baseline algorithm applying the Wilcoxon test with a confidence interval of 99%.

Figure 5.2 shows the interaction among the components defined at the three stages of our methodology. A target algorithm with its respective parameter space and a performance indicator are inputs required for the EVOCA tool. The interaction between the target MOEAs and EVOCA is established by a translator which interprets the performance of algorithmic configurations. Finally, a statistical analysis is performed to the most promising configurations.

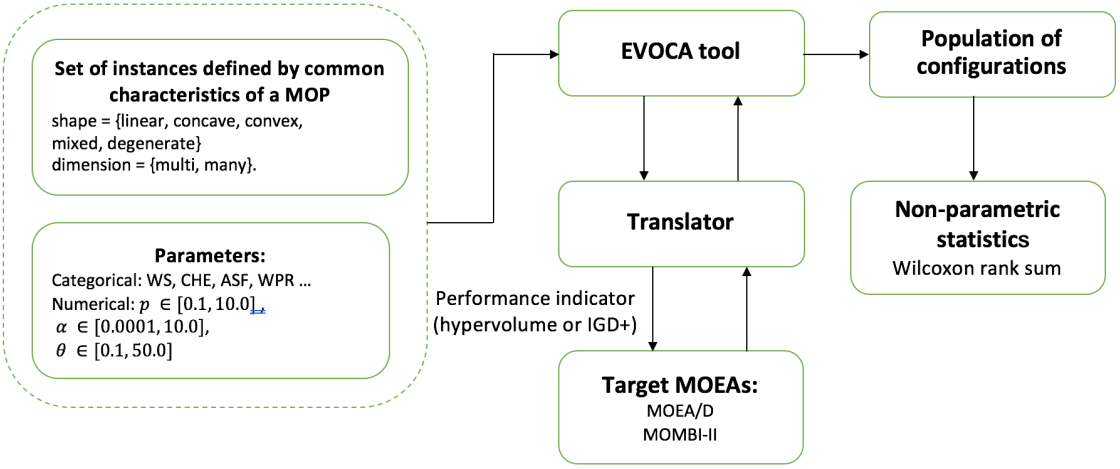


Figure 5.2: Offline tuning parameter process.

The next section explains three case studies that employ our methodology and the components previously indicated.

5.2 Case Studies

5.2.1 Performance of the Scalarizing Functions according to Pareto front shapes

In the first case study, our main interest was to analyze the behavior of fifteen SFs (WCP, WS, EWC, WPO, WPR, WN, CHE, ACHE, MCHE, ASF, AASF, PBI, IPBI, CS and VADS) and their model parameters to solve three basic Pareto front shapes. Moreover, we are interested in analyzing their scalability with respect to the number of objective functions.

Figure 5.3 gives an example to illustrate the effect of the parameter sensitivity in SFs. We present boxplots that show the behavior of the AASF and PBI functions on the MOEA/D framework with the same parameters settings. We compute the normalized hypervolume on 30 independent runs for solving the DTLZ1, DTLZ2, and DTLZ2^{-1}. We can see that the model parameters (α for AASF and θ for PBI) are very sensitive. Therefore, one parameter value can be appropriate to a particular Pareto front shape but can work poorly for others. This effect occurs in the same way if we vary the MOP's dimensionality.

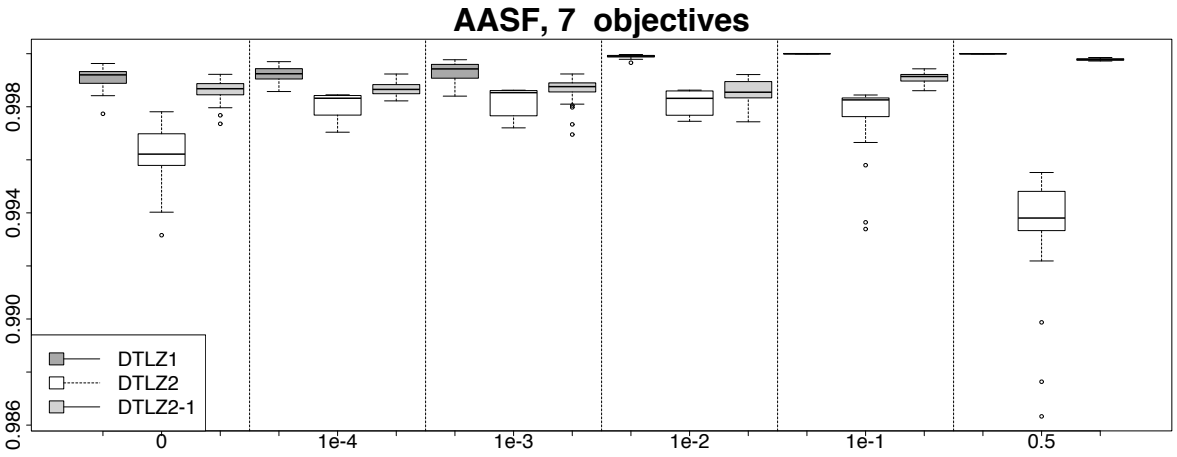
We selected the Lamé Superspheres test problems [42] since they encompass the three basic Pareto front geometries (linear, convex and concave) that can challenge SFs. Moreover, this benchmark is scalable to any number of variables and objective functions. Hence, we tested them for $2, 3, 5, 7$ and 10 objectives (m). We fixed the parameter $\gamma \in \{0.5, 1.0, 2.0\}$ to achieve Pareto fronts with convex, linear and concave geometries, respectively. Only unimodal problems were considered since we aimed to determine if the SFs can handle different shapes of the Pareto front for multi- and many-objective problems. Thus, adding difficulties in the MOPs would introduce noise to the selection process of an MOEA. We designed the following scenarios:

- Six scenarios that considered all the combinations of the cartesian product between the Pareto front geometries, given by $\gamma \in \{0.5, 1.0, 2.0\}$, and the {multi, many} cases. The "multi" case groups 2 and 3 objectives, whereas the "many" considers 5, 7 and 10 objectives.
- A global scenario which includes all the previous combinations.

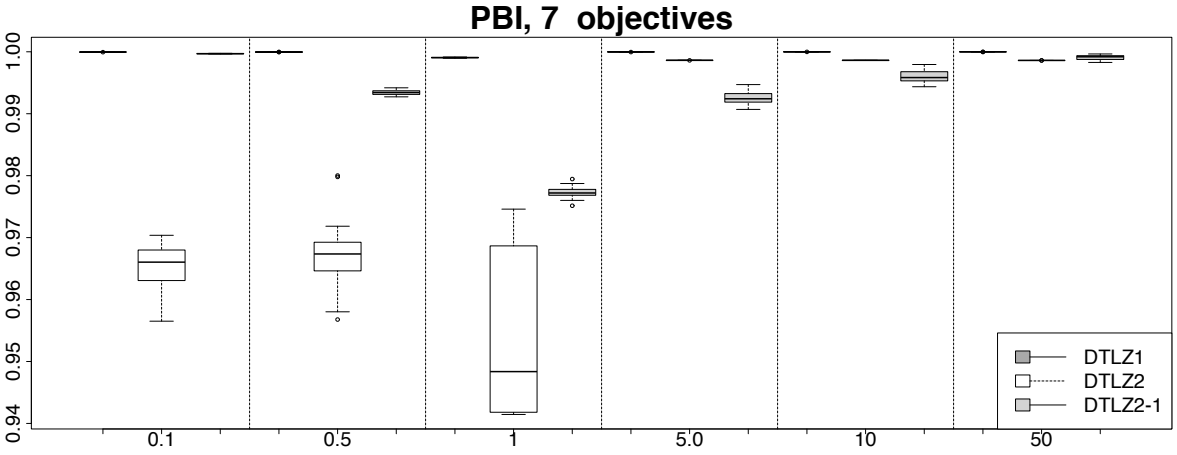
The best-ranked calibrations obtained via the filter process that uses Wilcoxon test are presented in Table 5.2.

Table 5.2: EVOCA's recommendation for each possible scenario. In every calibration, it is shown the scalarization function and its model parameter (in parentheses).

γ	**MOEA/D**		**MOMBI-II**	
	Multi-objective	**Many-objective**	**Multi-objective**	**Many-objective**
0.5	AASF (0.8001)	EWC (7.6)	EWC (8.6) AASF (1.0)	CHE
1.0	PBI (15.1) PBI (20.5639)	AASF (0.3001)	PBI (21.4123)	AASF (0.3001)
2.0	PBI (2.6) PBI (7.6) VADS (8.6)	PBI (2.6)	PBI (2.4026)	VADS (5.6) VADS (2.1)
Global	ASF CHE AASF (0.2423)		AASF (0.0501) CHE ASF	



(a) The AASF function with $\alpha = \{0.0, 0.0001, 0.001, 0.01, 0.1, 0.5\}$



(b) The PBI function with $\theta = \{0.1, 0.5, 1.0, 5.0, 10.0, 50.0\}$

Figure 5.3: Examples of parameter sensitivity for the AASF and PBI functions. Figures a) and b) show the normalized hypervolume indicator for linear (DTLZ1), concave (DTLZ3) and concave (DTLZ3^{-1}) Pareto fronts.

To summarize the results, in the multi-objective case more than one calibration was obtained, but in the many-objective case, we found only one recommended SF. EVOCA found that 6 of the 15 SF (EWC, CHE, ASF, AASF, PBI, and VADS) had an outstanding performance in the particular scenarios that we studied. In the global scenario, EVOCA determined that CHE, ASF, and AASF had the best results, emphasizing that ASF forms part of the original version of MOMBI-II, while CHE is used by MOEA/D for 2 objectives. Furthermore, the best options to solve problems with convex shape ($\gamma = 0.5$) were AASF, EWC, and CHE. For linear shapes ($\gamma = 1.0$) PBI and AASF. For concave shapes ($\gamma = 2.0$) the best choices were PBI and VADS. Finally, the tuning process obtained a greater accuracy for the corresponding model parameters than that provided by the values recommended in the literature.

For all scenarios, our experimental results are shown in Tables 5.3, 5.4 and 5.5. Here, the best value between the calibrated versions and the baseline MOEA is shown in grayscale. A line above the median ($\overline{\cdot}$) implies that the calibrated version outperformed in a significantly better way the baseline algorithm. Conversely, a line under the median ($\underline{\cdot}$) means that the calibrated version was significantly outperformed. In the multi-objective scenarios of Table 5.3, we can observe a clear performance improvement over the calibrated versions for MOEA/D with 2 objectives and MOMBI-II with 2 and 3 objectives. Only the version of MOMBI-II using EWC (8.6) was outperformed by the baseline MOMBI-II. In the particular case of MOEA/D, the major gains were achieved for the convex MOPs, and the best suited SFs were AASF (0.8001) and PBI (15.1, 20.5639, 2.6, 7.6). In MOMBI-II, the major gains were in the convex MOPs and the remaining problems with 3 objectives. The best SFs for this optimizer were EWC (8.6), AASF (1.0) and PBI (21.4123, 2.4026). As can be noticed, AASF worked very well for both MOEAs in the convex problems, while PBI performed best in the linear and concave problems. However, this SF is sensitive to its parameter value.

Table 5.3: Median ($\times 10^{-1}$) and standard deviation of the normalized hypervolume indicator on the **multi-objective scenarios**.

γ	**MOEA/D**					**MOMBI-II**				
	Config.	m				**Config.**	m			
		2		**3**			**2**		**3**	
0.5	Baseline	9.570400	5.0e-07	9.906204	3.4e-04	Baseline	9.570404	6.0e-07	9.917509	1.2e-04
	AASF (0.8001)	$\overline{9.573699}$	4.5e-08	$\overline{9.974616}$	1.2e-05	EWC (8.6)	$\overline{9.575178}$	5.4e-08	$\underline{9.902034}$	6.8e-04
						AASF (1.0)	$\overline{9.573940}$	9.4e-08	$\overline{9.975483}$	1.3e-05
1.0	Baseline	8.737370	8.2e-08	9.744895	2.5e-07	Baseline	8.737369	8.0e-08	9.636011	6.3e-04
	PBI (15.1)	$\overline{8.737374}$	4.5e-08	9.744894	7.0e-07	PBI (21.4123)	$\overline{8.737374}$	4.5e-07	$\overline{9.744881}$	1.8e-06
	PBI (20.5639)	$\overline{8.737374}$	3.0e-07	9.744894	1.0e-06					
2.0	Baseline	8.025324	1.3e-07	9.277872	1.6e-06	Baseline	8.025323	9.6e-08	9.209490	1.0e-03
	PBI (2.6)	$\overline{8.025325}$	1.7e-07	9.277875	1.1e-06	PBI (2.4026)	$\overline{8.025325}$	1.1e-07	$\overline{9.277874}$	1.0e-06
	PBI (7.6)	$\overline{8.025326}$	5.1e-07	9.277867	1.4e-06					
	VADS (8.6)	8.025324	1.4e-06	9.277872	1.5e-06					

In the many-objective scenarios of Table 5.4, we can notice a clear performance improvement over the calibrated versions of both optimizers. In MOEA/D, there were only 2 ties for the concave problems, and the major gains were in the convex MOPs. The best SFs for MOEA/D were: EWC (7.6), AASF (0.3001) and PBI (2.6). In MOMBI-II, the major

gains were in 5 and 7 objectives. The best SFs for this optimizer were CHE, AASF (0.3001) and VADS (5.6, 2.1). In both optimizers, AASF (0.3001) worked very well on the linear problems.

Table 5.4: Median ($\times 10^{-1}$) and standard deviation of the normalized hypervolume indicator on the **many-objective scenarios**.

γ	Config.	m					
		5		**7**		**10**	
		MOEA/D					
0.5	Baseline	9.961741	8.0e-04	9.908978	1.4e-03	9.831210	1.1e-03
	EWC (7.6)	9.999756	6.1e-08	9.999988	1.8e-08	9.999985	1.1e-16
1.0	Baseline	9.989021	2.1e-07	9.999387	1.9e-06	9.999864	1.5e-05
	AASF (0.3001)	9.989041	5.7e-07	9.999427	3.8e-07	9.999990	5.9e-07
2.0	Baseline	9.904560	2.3e-06	9.986141	4.5e-06	9.999186	4.2e-06
	PBI (2.6)	9.904578	1.8e-06	9.986145	5.4e-06	9.999160	3.8e-06
	MOMBI-II						
0.5	Baseline	9.970399	3.9e-04	9.987786	3.3e-04	9.991083	2.8e-04
	CHE	9.999357	3.2e-06	9.999964	1.5e-06	9.999942	4.1e-06
1.0	Baseline	9.937457	9.2e-04	9.977353	8.9e-04	9.993752	4.5e-04
	AASF (0.3001)	9.989052	5.6e-07	9.999427	4.8e-08	9.999991	0.0e+0
2.0	Baseline	9.884399	3.9e-04	9.976128	2.2e-04	9.995263	1.5e-04
	VADS (5.6)	9.904381	6.7e-06	9.985715	1.9e-05	9.998632	1.7e-05
	VADS (2.1)	9.904350	8.6e-06	9.985683	2.4e-05	9.998591	2.0e-05

In the global scenario of Table 5.5 a different pattern is observed. In the case of MOEA/D, ASF and CHE worked very well in the convex problems from 3 up to 10 objectives. However, they worsened their behavior in the linear and concave MOPs. Similarly, AASF (0.2423) performed well in the convex problems and the linear problems for 5, 7 and 10 objectives. However, its performance deteriorated in the concave MOPs. On the other hand, for MOMBI-II, the 2 SFs were complementary to each other. For example, for 2 objectives, CHE was competitive with respect to the baseline version, while in the concave problems for 3 to 10 objectives AASF (0.0501) performed best.

These results suggest that no SF can solve effectively all the problems. Instead, there is a subset of them that can tackle in an effective manner some specific problems.

Table 5.5: Median ($\times 10^{-1}$) and standard deviation of the normalized hypervolume indicator on the **global scenario**.

γ	Config.	m = 2		3		5		7		10	
					MOEA/D						
0.5	Baseline	9.570400	5.0e-07	9.906204	3.4e-04	9.961741	8.0e-04	9.908978	1.4e-03	9.831210	1.1e-03
	ASF	9.570399	3.2e-07	9.917019	1.5e-04	9.971177	5.7e-04	9.987843	2.6e-04	9.994325	2.8e-04
	CHE	9.570400	5.0e-07	9.953880	2.6e-06	9.999339	1.6e-06	9.999972	5.5e-06	9.999967	6.3e-06
	AASF (0.2423)	9.571983	1.9e-07	9.966615	7.4e-05	9.995188	5.0e-05	9.997490	4.8e-05	9.998646	5.4e-05
1.0	Baseline	8.737370	8.2e-08	9.744895	2.5e-07	9.989021	2.1e-07	9.999387	1.9e-06	9.999864	1.5e-05
	ASF	8.737369	7.2e-08	9.638966	6.2e-04	9.914455	7.2e-04	9.963243	8.9e-04	9.988689	4.1e-04
	CHE	8.737370	8.2e-08	9.689283	1.8e-05	9.967928	2.4e-05	9.983080	4.6e-04	9.932911	2.2e-03
	AASF (0.2423)	8.668011	1.3e-06	9.720175	4.1e-05	9.989038	2.8e-07	9.999426	2.8e-07	9.999991	4.9e-07
2.0	Baseline	8.025324	1.3e-07	9.277872	1.6e-06	9.904560	2.3e-06	9.986141	4.5e-06	9.999186	4.2e-06
	ASF	8.025323	1.0e-07	9.207519	7.9e-04	9.870073	6.6e-04	9.970228	5.6e-04	9.985247	1.6e-03
	CHE	8.025324	1.3e-07	9.228448	5.7e-05	9.854362	1.3e-05	9.807065	2.3e-03	9.561658	6.5e-03
	AASF (0.2423)	7.876793	2.0e-06	9.152424	2.5e-03	9.847468	2.3e-03	9.969067	5.6e-04	9.996733	1.3e-04
					MOMBI-II						
0.5	Baseline	9.570404	6.0e-07	9.917509	1.2e-04	9.970399	3.9e-04	9.987786	3.3e-04	9.991083	2.8e-04
	CHE	9.570404	6.2e-07	9.953905	3.6e-06	9.999357	3.2e-06	9.999964	1.5e-06	9.999942	4.1e-06
	AASF (0.0501)	9.570326	4.4e-07	9.924333	2.7e-04	9.979430	3.4e-04	9.989421	2.1e-04	9.993698	2.2e-04
1.0	Baseline	8.737369	8.0e-08	9.636011	6.3e-04	9.937457	9.2e-04	9.977353	8.9e-04	9.993752	4.5e-04
	CHE	8.737369	7.2e-08	9.689161	1.2e-05	9.968534	1.1e-04	9.984665	6.2e-04	9.931828	2.4e-03
	AASF (0.0501)	8.732288	1.2e-05	9.744887	6.7e-07	9.989033	2.2e-07	9.999427	1.8e-08	9.999991	0.0e+0
2.0	Baseline	8.025323	9.6e-08	9.209490	1.0e-03	9.884399	3.9e-04	9.976128	2.2e-04	9.995263	1.5e-04
	CHE	8.025323	9.6e-08	9.229191	1.5e-04	9.855497	7.7e-04	9.851832	2.2e-03	9.563050	6.3e-03
	AASF (0.0501)	8.003164	2.5e-05	9.269135	1.0e-04	9.898159	1.1e-04	9.985653	1.7e-05	9.999221	5.8e-06

5.2.2 Scalability of the Scalarizing Functions on different scenarios

We designed this case study as an extension of the first one with the aim of accomplishing the following goals:

- Identify the most suitable SFs to deal with complicated MOPs that involve characteristics such as disconnected, mixed[2], degenerated Pareto front shapes or multifrontal difficulty.
- Perform a study of scalability including scenarios in which one test problem is adopted with a different number of objective functions.
- Determine if there exists a significant improvement in the use of reciprocal SF[3] or not.
- Explore the performance of alternative SFs which have not been used in the MOEA/D framework (CS, EWC, VADS, GSF, NSF, DIDASS).
- Find configurations of SFs that outperform the baseline version of MOEA/D.

[2]Pareto front with convex and concave regions.
[3]i.e. $\frac{1}{\lambda}$, where λ is a weight vector.

This study case considers 32 variants of SFs: WCP, rWCP, WS, rWS, rEWC, EWC, WPO, rWPO, WPR, rWPR, WN, rWN, CHE, ACHE, rACHE, MCHE, rMCHE, ASF, AASF, PBI, QPBI, 2LPBI, CS, rCS, VADS, rVADS, GSF, rGSF, NSF, rNSF, DIDASS, rDIDASS[4] which are tuned for solving scenarios built by selecting test function from three different benchmarks: the Lamé Supersphere functions, the Deb-Thiele-Laumanns-Zitzler (DTLZ) [32] and the Walking-Fish-Group (WFG) test suites [66]. These MOPs were classified based on their Pareto front geometry in the following scenarios:

- Convex: $DTLZ3^{-1}$, Lamé Supersphere functions with $\gamma = 0.3$ and $\gamma = 0.5$.
- Linear: $DTLZ1$ and Lamé Supersphere functions with $\gamma = 1.0$.
- Concave: $DTLZ3$, Lamé Supersphere functions with $\gamma = 2.0$, $\gamma = 4.0$.
- Mixed: $WFG1$ and $WFG2$.
- Degenerated[5] : $WFG3$.

We separate two subsets of MOPs considering the number of objectives: *multi-objective* case as $m = \{2, 3\}$, and the *many-objective* case using $m = \{4, 5, 6, 7, 8, 9, 10\}$. We define ten scenarios combining the {covex, concave, linear, mixed and degenerated} Pareto front shapes versus the {multi and many} cases. Additionally, we include the global case that involves all test problems with the goal of finding out if there are robust SFs. The performance indicator used by offline tuning was the normalized hypervolume (NHV).

Table 5.6 summarizes the best results obtained for each scenario. Following the same nomenclature used in the previous case study, the best value between the calibrated versions and the baseline MOEA is shown in grayscale. A line above ($\overline{\cdot}$) implies that the calibrated version outperformed in a significantly better way the baseline algorithm. Conversely, a line under ($\underline{\cdot}$) means that the calibrated version was significantly outperformed. In general 10 of the 32 SFs were selected by the EVOCA tool as the most suitable in the predefined scenarios. The ASF and EWC functions have the most repetitions, emphasizing that the EWC function is appropriate in complicated Pareto front geometries (degenerated and mixed cases) and it was also the best option for the global scenario. However, the p value for EWC is sensitive to the type of problems being solved.

Tables 5.7 and 5.8 show the results for multi- and many-objective optimization, respectively. In the first case, the EVOCA's recommendations only outperform the baseline version in at least 20% of the cases. Besides, other recommendations are competitive in the majority of cases. Thus in multi-objective optimization (2 and 3 objectives), we can find a variety of SFs that are suitable for the MOEA/D framework. EVOCA was able to find similar values with respect to the baseline version in a multi-linear scenario (PBI with $\theta = 5.1$).

On the other hand, in the many-objective scenarios we can highlight the next observations:

- **Many-concave**: the baseline version outperforms all the configurations proposed by EVOCA. One alternative for finding a better configuration is to increase the number of function evaluations for EVOCA for this scenario. But this would not allow a fair comparison with other scenarios.

[4]The prefix r used in the acronym refers to the reciprocal version of the SF.

[5]The degenerated subset refers to MOPs with $m - 1$ dimensionality in their Pareto front.

Table 5.6: EVOCA's recommendation according to the hypervolume indicator.

Scenario	**multi**		**many**	
concave	ASF		ASF	
	WCP	$p = 10$	rNSF (a)	$\alpha = 0.3$
convex			**ASF**	
		ASF	r**ACHE**	$\alpha = 2.0468$
degenerated	GSF	$\alpha = 1.5$, $\beta = 6.5$	**EWC** (a)	$p = 9.1$
	rEWC	$p = 8.1$	**EWC** (b)	$p = 4.6$
linear	rACHE	$\alpha = 6.0001$	**2LPBI**	$\alpha = 3.4, \theta_1 = 37, \theta_2 = 20.2$
	PBI	$\theta = 5.1$	EWC	$p = 0.6$
mixed	r**ACHE**	$\alpha = 0.0001$	**EWC**	$p = 6.1$
	EWC	$p = 4.1$		
global		**EWC** $p = 6.1$	**AASF** $\alpha = 7.5001$	

- **Many-convex**: Both EVOCA's recommendations, ASF and rACHE functions outperform the baseline version in more than 90% of cases. According to the results, we conclude that ASF is scalable in MOPs with a convex Pareto front.
- **Many-degenerated**: EWC with two ρ values (9.1 and 4.6) outperforms the baseline version, and both cases are scalable for any number of objectives. The best results were obtained by EWC with $\rho = 9.1$
- **Many-linear**: the 2LPBI function outperforms the baseline version only when using from 6 to 10 objectives.
- **Many-mixed**: EWC outperforms the baseline version in all problem instances into the many-mixed scenario. The same configuration (EWC with $p = 6.1$) was obtained in the global scenario. Therefore, we can say that this configuration is scalable.

Table 5.7: Median and standard deviation of the normalized hypervolume indicator on the **multi-objective scenarios**.

MOEA/D

Problem	**Config.**	**Objectives**			
		2		**3**	
multi-concave					
LAME γ 2.0	Baseline	9.969345e-01	1.8e-07	9.877527e-01	5.3e-05
	ASF	9.969346e-01	2.6e-07	9.853744e-01	1.1e-03
	WCP (p 10)	9.966721e-01	7.6e-06	9.866191e-01	1.5e-04
LAME γ 4.0	Baseline	9.966626e-01	3.2e-05	9.914845e-01	9.1e-05
	ASF	9.966620e-01	3.2e-05	9.889950e-01	9.7e-04
	WCP (p 10)	9.960594e-01	2.1e-06	9.895413e-01	1.8e-05
DTLZ3	Baseline	9.944036e-01	1.7e-03	9.871988e-01	7.7e-04
	ASF	9.935713e-01	4.4e-03	9.884571e-01	1.1e-03
	WCP (p 10)	9.940339e-01	2.0e-03	9.868133e-01	6.6e-04
multi-convex					
LAME γ 0.3	Baseline	9.980796e-01	5.6e-07	9.996877e-01	6.2e-07
	ASF	9.980796e-01	7.2e-07	9.981039e-01	1.2e-05
LAME γ 0.5	Baseline	9.969164e-01	3.0e-07	9.968244e-01	4.0e-06
	ASF	9.969163e-01	5.6e-07	9.929787e-01	1.5e-04
DTLZ3	Baseline	9.493454e-01	2.7e-03	9.914186e-01	4.4e-04
CONVEX	ASF	9.492604e-01	6.7e-03	9.904018e-01	8.5e-04
multi-degenerate					
WFG3	Baseline	9.816058e-01	8.5e-03	9.691653e-01	7.8e-03
	GSF (α 1.5 β 6.5)	9.707367e-01	7.6e-03	9.556639e-01	5.5e-03
	rEWC (p 8.1)	9.746220e-01	1.2e-02	8.662606e-01	9.8e-03
multi-linear					
LAME γ 1.0	Baseline	9.985565e-01	1.1e-07	9.893772e-01	1.8e-05
	rACHE (α 6.0001)	9.985565e-01	7.3e-08	9.840810e-01	6.9e-04
	PBI (θ 5.1)	9.985570e-01	1.6e-07	9.947087e-01	1.9e-07
DTLZ1	Baseline	9.983356e-01	5.0e-05	9.986995e-01	1.6e-05
	rACHE (α 6.0001)	9.983484e-01	3.9e-05	9.955623e-01	6.0e-04
	PBI (θ 5.1)	9.982488e-01	1.9e-04	9.993214e-01	2.7e-05
multi-mixed					
WFG1	Baseline	5.201770e-01	2.4e-02	5.421981e-01	1.3e-02
	rACHE (α 0.0001)	5.436472e-01	2.5e-02	5.504687e-01	1.6e-02
	EWC (p 4.1)	4.974010e-01	1.6e-02	5.157478e-01	1.6e-02
WFG2	Baseline	9.182027e-01	2.9e-02	9.701562e-01	9.3e-03
	rACHE (α 0.0001)	9.118927e-01	4.7e-02	9.798285e-01	9.2e-03
	EWC (p 4.1)	9.009642e-01	4.0e-02	9.907056e-01	6.6e-03

Table 5.8: Median and standard deviation of the normalized hypervolume indicator on the **many-objective scenarios**.

MOEA/D

Problem	Config.	Objectives						
		4	5	6	7	8	9	10
many-concave								
LAME γ 2.0	Baseline	9.925562e-01 1.5e-06	9.951465e-01 9.3e-07	9.974913e-01 3.7e-06	9.988698e-01 1.1e-06	9.994953e-01 4.9e-07	9.997971e-01 1.0e-06	9.999371e-01 2.0e-07
	ASF	9.876243e-01 6.9e-04	9.906960e-01 8.1e-04	9.952222e-01 4.6e-04	9.960805e-01 1.2e-03	9.973939e-01 9.6e-04	9.979544e-01 1.9e-03	9.978943e-01 1.5e-03
	rNSF (α 0.3)	9.647418e-01 4.8e-03	9.801227e-01 3.2e-03	9.908602e-01 1.5e-03	9.943737e-01 6.3e-03	9.976813e-01 7.2e-04	9.989850e-01 4.5e-04	9.990061e-01 4.3e-04
LAME γ 4.0	Baseline	9.929665e-01 1.6e-05	9.943133e-01 1.1e-05	9.954232e-01 5.0e-05	9.973841e-01 4.0e-05	9.985008e-01 1.1e-04	9.989675e-01 1.1e-04	9.993414e-01 2.0e-04
	ASF	9.894687e-01 7.2e-04	9.909984e-01 5.5e-04	9.908904e-01 6.2e-03	9.902206e-01 6.7e-03	9.931062e-01 6.2e-03	9.939219e-01 7.2e-03	9.926051e-01 8.3e-03
	rNSF (α 0.3)	9.784736e-01 0.0e+00	9.853783e-01 0.0e+00	9.907218e-01 1.2e-02	9.868155e-01 7.4e-03	9.938669e-01 4.9e-02	9.933535e-01 2.4e-02	9.962380e-01 1.0e-01
DTLZ3	Baseline	9.882697e-01 1.4e-01	9.928695e-01 2.0e-01	9.937282e-01 3.6e-01	9.982696e-01 1.3e-03	9.992847e-01 6.7e-05	9.996409e-01 8.4e-02	9.998860e-01 8.4e-02
	ASF	9.896359e-01 5.8e-04	9.931205e-01 4.6e-04	9.930334e-01 2.6e-03	9.965651e-01 5.7e-03	9.943435e-01 7.9e-03	9.951717e-01 5.0e-03	9.939067e-01 6.2e-03
	rNSF (α 0.3)	9.552270e-01 4.3e-03	9.781834e-01 5.0e-03	9.883960e-01 9.0e-03	9.893900e-01 4.1e-03	9.918828e-01 3.4e-03	9.944961e-01 2.7e-03	9.950253e-01 2.8e-03
many-convex								
LAME γ 0.3	Baseline	9.984063e-01 1.2e-04	9.988913e-01 8.4e-05	9.991883e-01 5.2e-05	9.987610e-01 7.5e-05	9.987765e-01 6.2e-05	9.989282e-01 3.9e-05	9.987983e-01 2.6e-05
	ASF	9.989846e-01 3.1e-05	9.993536e-01 4.4e-05	9.996678e-01 6.4e-05	9.997291e-01 6.3e-05	9.997899e-01 4.8e-05	9.998575e-01 2.4e-05	9.998533e-01 2.3e-05
	rACHE (α 2.0468)	9.989105e-01 5.5e-06	9.992142e-01 1.3e-05	9.994533e-01 9.9e-06	9.994054e-01 9.7e-06	9.994143e-01 2.5e-05	9.994706e-01 2.5e-05	9.993744e-01 4.6e-05
LAME γ 0.5	Baseline	9.956296e-01 5.4e-04	9.954438e-01 8.3e-04	9.953600e-01 6.0e-04	9.914424e-01 1.0e-03	9.882683e-01 1.1e-03	9.870667e-01 1.2e-03	9.829351e-01 5.8e-04
	ASF	9.947577e-01 4.6e-04	9.966345e-01 4.5e-04	9.980127e-01 2.7e-04	9.984035e-01 3.5e-04	9.987172e-01 2.3e-04	9.989549e-01 2.4e-04	9.987793e-01 3.5e-04
	rACHE (α 2.0468)	9.938734e-01 4.0e-05	9.941115e-01 2.4e-04	9.950108e-01 5.3e-04	9.940720e-01 5.2e-04	9.946271e-01 8.9e-04	9.945870e-01 6.6e-04	9.939221e-01 8.5e-04
DTLZ3	Baseline	9.930689e-01 1.2e-03	9.935250e-01 1.2e-03	9.927509e-01 1.2e-03	9.907361e-01 1.2e-03	9.865057e-01 1.4e-03	9.862367e-01 1.2e-03	9.823582e-01 1.4e-03
	ASF	9.970675e-01 5.7e-04	9.982056e-01 4.4e-04	9.989476e-01 4.3e-04	9.991261e-01 2.2e-04	9.991172e-01 4.6e-04	9.990511e-01 4.5e-04	9.981220e-01 1.0e-03
	rACHE (α 2.0468)	9.946176e-01 6.2e-04	9.947823e-01 6.1e-04	9.954141e-01 7.8e-04	9.944618e-01 4.7e-04	9.943914e-01 7.2e-04	9.930459e-01 1.2e-03	9.911028e-01 5.8e-04
many-degenerate								
WFG3	Baseline	9.467935e-01 8.6e-03	9.194890e-01 1.0e-02	8.727428e-01 1.5e-02	8.272288e-01 1.8e-02	8.137638e-01 1.9e-02	7.532558e-01 1.5e-02	7.759282e-01 1.9e-02
	EWC (p 9.1)	9.899487e-01 7.9e-03	9.735388e-01 1.1e-02	9.669307e-01 1.2e-02	9.355234e-01 2.0e-02	9.499984e-01 1.6e-02	9.290373e-01 1.4e-02	9.211612e-01 2.3e-02
	EWC (p 4.6)	9.765538e-01 8.8e-03	9.237277e-01 8.5e-03	9.392740e-01 1.4e-02	9.258923e-01 1.6e-02	9.444254e-01 2.1e-02	9.215792e-01 1.4e-02	9.106359e-01 2.3e-02
many-linear								
LAME γ 1.0	Baseline	9.974575e-01 6.0e-07	9.988628e-01 1.9e-07	9.997368e-01 2.4e-06	9.999413e-01 8.3e-07	9.999836e-01 3.3e-07	9.999941e-01 1.0e-06	9.999985e-01 4.9e-07
	2LPBI (α 3.4 θ_1 37 θ_2 20.2)	9.974574e-01 6.1e-07	9.988627e-01 1.6e-07	9.997375e-01 3.5e-07	9.999425e-01 2.0e-07	9.999841e-01 5.3e-08	9.999961e-01 4.2e-08	9.999991e-01 4.9e-08
	EWC (p 0.6)	9.880296e-01 5.9e-07	9.967538e-01 3.0e-07	9.993732e-01 4.4e-07	9.988052e-01 2.2e-04	9.970301e-01 1.6e-03	9.972289e-01 4.8e-04	9.964991e-01 9.0e-04
DTLZ1	Baseline	9.996650e-01 1.0e-05	9.999563e-01 5.1e-06	9.999689e-01 9.3e-06	9.999836e-01 6.8e-06	9.999763e-01 1.4e-05	9.999309e-01 3.4e-05	9.998910e-01 5.1e-05
	2LPBI (α 3.4 θ_1 37 θ_2 20.2)	9.996392e-01 1.5e-03	9.999559e-01 2.5e-03	9.999850e-01 7.8e-06	9.999954e-01 4.5e-06	9.999968e-01 1.7e-06	9.999916e-01 4.2e-06	9.999888e-01 4.6e-06
	EWC (p 0.6)	9.990971e-01 2.6e-06	9.998983e-01 4.8e-07	9.999903e-01 1.2e-06	9.999487e-01 5.4e-05	9.998297e-01 8.2e-05	9.998445e-01 4.0e-05	9.997459e-01 3.7e-05
many-mixed								
WFG1	Baseline	4.790568e-01 2.2e-02	4.520302e-01 2.0e-02	3.846597e-01 1.4e-02	3.601013e-01 1.5e-02	3.252665e-01 1.1e-02	3.131757e-01 1.5e-02	2.881070e-01 1.2e-02
	EWC (p 6.1)	5.193374e-01 1.2e-02	4.930685e-01 7.6e-03	4.639335e-01 8.0e-03	4.481889e-01 5.8e-03	4.229312e-01 7.5e-03	4.062602e-01 6.8e-03	3.807265e-01 8.0e-03
WFG2	Baseline	9.449912e-01 1.9e-02	9.429711e-01 2.4e-02	9.033079e-01 2.3e-02	8.962356e-01 2.4e-02	8.928489e-01 2.1e-02	8.754769e-01 2.7e-02	8.703044e-01 3.1e-02
	EWC (p 6.1)	9.881093e-01 1.0e-02	9.849093e-01 1.1e-02	9.671404e-01 2.2e-02	9.608640e-01 1.5e-02	9.761252e-01 1.3e-02	9.707780e-01 2.4e-02	9.594243e-01 2.5e-02

The results in a so-called global scenario (see Table 5.9) confirm that there does not exist a single configuration that optimizes the 11 MOPs at the same time. However, the two configurations EWC ($p = 6.1$) and AASF ($\alpha = 7.5001$) outperform the MOEA/D baseline in more than 55% of the cases. We can observe a certain degree of conflict among the characteristics of the problems WFG1 and WFG2. This means that one configuration is better for WFG1 but not for WG2 and viceversa. This requires an additional analysis of the MOP's characteristics. AASF with $\alpha = 7.5001$ outperforms EWC with $p = 6.1$ in multifrontal problems such as DTLZ1 and DTLZ3 (convex). According to our results, there is evidence that the reciprocal versions of SF such as the ASF, AASF and ACHE-1 present the best performance in most cases.

Table 5.9: Median and standard deviation of the normalized hypervolume indicator on the **global scenario**.

MOEA/D

Problem	Config.	Objectives								
		2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
		global								
LAME γ 0.3	Baseline	9.980796e-01 8.6e-07	9.996877e-01 6.2e-07	9.983878e-01 1.3e-04	9.989230e-01 1.1e-04	9.992095e-01 6.4e-05	9.987942e-01 1.1e-04	9.987744e-01 7.0e-05	9.989237e-01 5.1e-05	9.988091e-01 1.8e-05
	EWC (p 6.1)	9.989225e-01 6.9e-08	9.999426e-01 5.0e-08	9.999993e-01 6.8e-08	9.999999e-01 5.1e-08	9.999999e-01 6.4e-08	9.999999e-01 5.7e-08	1.000000e+00 5.7e-08	1.000000e+00 2.4e-08	1.000000e+00 4.2e-08
	AASF (α 7.5001)	9.988212e-01 1.3e-07	9.995033e-01 5.7e-06	9.997146e-01 3.4e-06	9.997818e-01 3.4e-06	9.998333e-01 4.4e-06	9.998205e-01 8.7e-06	9.998356e-01 6.9e-06	9.998593e-01 8.2e-06	9.998580e-01 9.9e-06
LAME γ 0.5	Baseline	9.969165e-01 3.8e-07	9.968250e-01 4.4e-06	9.959666e-01 4.9e-04	9.954714e-01 5.8e-04	9.954406e-01 6.3e-04	9.914835e-01 6.6e-04	9.884854e-01 8.3e-04	9.872914e-01 1.2e-03	9.826802e-01 5.8e-04
	EWC (p 6.1)	9.974221e-01 3.0e-08	9.989210e-01 3.8e-07	9.998606e-01 6.3e-08	9.999772e-01 5.8e-08	9.999975e-01 5.7e-08	9.999987e-01 0.0e+00	9.999962e-01 4.8e-08	9.999979e-01 0.0e+00	9.999985e-01 0.0e+00
	AASF (α 7.5001)	9.973750e-01 0.0e+00	9.990449e-01 3.6e-06	9.997047e-01 5.9e-06	9.997647e-01 9.0e-06	9.998317e-01 1.9e-05	9.998160e-01 2.0e-05	9.998261e-01 2.4e-05	9.998751e-01 2.6e-05	9.998848e-01 2.5e-05
DTLZ3 CONVEX	Baseline	9.492573e-01 3.4e-03	9.914824e-01 2.6e-04	9.928667e-01 1.4e-03	9.931869e-01 1.2e-03	9.925022e-01 1.2e-03	9.900792e-01 1.2e-03	9.862972e-01 1.4e-03	9.854287e-01 1.3e-03	9.819201e-01 1.0e-03
	EWC (p 6.1)	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00
	AASF (α 7.5001)	9.458294e-01 3.4e-03	9.944519e-01 1.1e-04	9.995630e-01 1.2e-05	9.999137e-01 1.7e-05	9.999743e-01 7.9e-06	9.999789e-01 9.1e-06	9.999643e-01 1.4e-05	9.999890e-01 8.5e-06	9.999777e-01 1.2e-05
LAME γ 1.0	Baseline	9.985565e-01 7.3e-08	9.893746e-01 2.6e-05	9.974575e-01 3.1e-07	9.988627e-01 1.7e-07	9.997361e-01 3.8e-06	9.999416e-01 5.4e-07	9.999837e-01 2.2e-07	9.999944e-01 9.7e-07	9.999985e-01 3.6e-07
	EWC (p 6.1)	9.982656e-01 4.2e-08	9.931184e-01 1.3e-06	9.956297e-01 2.1e-06	9.965154e-01 1.7e-06	9.987855e-01 9.8e-07	9.989159e-01 3.9e-06	9.969166e-01 1.2e-04	9.973082e-01 1.8e-04	9.965168e-01 1.7e-04
	AASF (α 7.5001)	9.355383e-01 7.0e-06	9.625377e-01 2.8e-06	9.851280e-01 1.4e-04	9.963348e-01 8.8e-06	9.988290e-01 1.6e-05	9.998184e-01 7.7e-06	9.999406e-01 1.5e-06	9.999868e-01 5.0e-07	9.999963e-01 1.8e-06
DTLZ1	Baseline	9.983389e-01 4.7e-05	9.987001e-01 1.4e-05	9.996624e-01 9.8e-06	9.999564e-01 5.1e-06	9.999663e-01 1.2e-05	9.999856e-01 6.4e-06	9.999721e-01 1.2e-05	9.999222e-01 5.7e-05	9.998694e-01 4.5e-05
	EWC (p 6.1)	9.979841e-01 3.5e-05	9.988662e-01 3.5e-06	9.994399e-01 7.1e-07	9.998983e-01 2.8e-07	9.999902e-01 6.2e-08	9.999611e-01 3.9e-06	9.998098e-01 2.3e-05	9.998057e-01 2.2e-05	9.997446e-01 4.4e-05
	AASF (α 7.5001)	9.841566e-01 7.6e-05	9.953951e-01 4.3e-05	9.989147e-01 1.1e-05	9.998847e-01 8.5e-07	9.999817e-01 2.4e-07	9.999985e-01 5.6e-08	9.999997e-01 3.6e-07	9.999999e-01 2.2e-07	1.000000e+00 4.2e-08
LAME γ 2.0	Baseline	9.969346e-01 6.5e-07	9.877482e-01 6.4e-05	9.925565e-01 1.2e-06	9.951463e-01 8.5e-07	9.974917e-01 3.4e-06	9.988704e-01 1.1e-06	9.994954e-01 4.7e-07	9.997973e-01 7.3e-07	9.999371e-01 1.7e-07
	EWC (p 6.1)	9.948469e-01 4.3e-07	9.917105e-01 1.2e-06	9.873410e-01 1.1e-04	9.903658e-01 3.4e-07	9.960221e-01 7.8e-07	9.795977e-01 1.0e-03	9.454341e-01 5.4e-03	9.453165e-01 5.3e-03	9.266335e-01 1.2e-02
	AASF (α 7.5001)	9.316770e-01 0.0e+00	9.358289e-01 0.0e+00	9.554140e-01 0.0e+00	9.736181e-01 2.5e-03	9.856070e-01 8.4e-03	9.857694e-01 7.4e-03	9.961327e-01 1.3e-03	9.962102e-01 1.1e-03	9.979555e-01 2.2e-02
LAME γ 4.0	Baseline	9.966627e-01 5.3e-05	9.914988e-01 7.7e-05	9.929701e-01 1.5e-05	9.943152e-01 9.0e-06	9.954352e-01 3.8e-05	9.973813e-01 3.5e-05	9.985182e-01 6.7e-05	9.989755e-01 6.9e-05	9.993369e-01 2.8e-04
	EWC (p 6.1)	9.927484e-01 1.0e-06	9.910310e-01 2.0e-06	9.872021e-01 8.7e-07	9.919570e-01 4.7e-07	9.952329e-01 1.2e-06	9.423000e-01 8.8e-03	8.830983e-01 3.1e-02	8.729300e-01 1.9e-02	8.672018e-01 2.8e-02
	AASF (α 7.5001)	9.740260e-01 3.9e-08	9.735744e-01 0.0e+00	9.784736e-01 1.9e-02	9.853783e-01 8.9e-03	9.907218e-01 1.3e-02	9.716829e-01 5.9e-02	9.870850e-01 1.3e-01	9.846821e-01 1.8e-01	9.947090e-01 9.4e-02
DTLZ3	Baseline	9.933834e-01 2.5e-03	9.874109e-01 1.1e-03	9.869861e-01 1.8e-01	9.932291e-01 1.2e-03	9.951178e-01 2.2e-01	9.982092e-01 2.4e-01	9.992331e-01 1.1e-04	9.996550e-01 7.7e-05	9.998726e-01 2.0e-05
	EWC (p 6.1)	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 3.0e-01	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00	0.000000e+00 0.0e+00
	AASF (α 7.5001)	9.289130e-01 3.2e-03	9.347853e-01 8.1e-04	9.549998e-01 3.3e-04	9.734147e-01 8.5e-03	9.698804e-01 8.1e-03	9.844006e-01 5.3e-03	9.860411e-01 4.0e-03	9.868702e-01 3.7e-03	9.892454e-01 3.4e-03
WFG1	Baseline	5.318888e-01 2.7e-02	5.414383e-01 1.3e-02	4.980034e-01 2.2e-02	4.530968e-01 2.1e-02	3.793992e-01 1.5e-02	3.599532e-01 1.2e-02	3.360926e-01 1.4e-02	3.106197e-01 1.4e-02	2.904074e-01 1.2e-02
	EWC (p 6.1)	4.951364e-01 2.2e-02	5.363181e-01 1.6e-02	5.193374e-01 1.2e-02	4.930685e-01 7.6e-03	4.639335e-01 8.0e-03	4.481889e-01 5.8e-03	4.229312e-01 7.5e-03	4.062602e-01 6.8e-03	3.807265e-01 8.0e-03
	AASF (α 7.5001)	4.585346e-01 1.9e-02	5.235461e-01 1.2e-02	5.135119e-01 1.1e-02	4.877982e-01 9.3e-03	4.598161e-01 8.7e-03	4.431439e-01 8.5e-03	4.166216e-01 6.9e-03	4.021527e-01 8.5e-03	3.782518e-01 7.2e-03
WFG2	Baseline	9.106644e-01 5.3e-02	9.666343e-01 5.4e-03	9.377309e-01 1.9e-02	9.371860e-01 2.2e-02	8.947709e-01 3.0e-02	8.885380e-01 3.0e-02	8.928254e-01 2.4e-02	8.748844e-01 2.2e-02	8.753256e-01 2.9e-02
	EWC (p 6.1)	9.074237e-01 3.2e-02	9.893847e-01 7.5e-03	9.881093e-01 1.0e-02	9.849093e-01 1.1e-02	9.671404e-01 2.2e-02	9.608640e-01 1.5e-02	9.761252e-01 1.3e-02	9.707780e-01 2.4e-02	9.594243e-01 2.5e-02
	AASF (α 7.5001)	8.776587e-01 5.2e-02	9.765476e-01 1.1e-02	9.885827e-01 9.3e-03	9.851713e-01 1.2e-02	9.802246e-01 1.7e-02	9.778885e-01 1.7e-02	9.793166e-01 1.9e-02	9.647487e-01 2.2e-02	9.611644e-01 2.0e-02
WFG3	Baseline	9.790526e-01 1.1e-02	9.690774e-01 8.4e-03	9.444232e-01 8.6e-03	9.097807e-01 9.5e-03	8.794383e-01 1.2e-02	8.234520e-01 1.5e-02	8.247546e-01 1.7e-02	7.624484e-01 1.7e-02	7.754245e-01 2.0e-02
	EWC (p 6.1)	9.875105e-01 1.5e-02	9.828129e-01 3.9e-03	9.868404e-01 7.0e-03	9.655437e-01 8.1e-03	9.595693e-01 1.4e-02	9.410464e-01 1.2e-02	9.515970e-01 1.5e-02	9.240387e-01 1.6e-02	9.134445e-01 1.7e-02
	AASF (α 7.5001)	8.308122e-01 9.7e-03	8.368212e-01 1.4e-02	7.940841e-01 1.1e-02	6.726645e-01 5.0e-03	7.054622e-01 1.2e-02	6.860270e-01 1.2e-02	7.141976e-01 1.3e-02	6.784168e-01 1.0e-02	6.858779e-01 2.1e-02

5.2.3 Studying the effect of convergence and distribution of the Augmented Chebyshev functions

The Chebyshev (CHE) functions have interesting mathematical properties that allow them to properly handle MOPs with diverse Pareto front shapes and any number of objectives. Moreover, the augmented CHE functions are useful to avoid the generation of weak Pareto solutions. However, they require the definition of a parameter value which is very sensitive and plays an important role in their performance. This section focuses on a comparative study of five variants of weighted SFs based on the CHE model that includes the next procedures: 1) a sensitivity analysis testing different parameter values. We compute the Spearman's correlation to determine a monotonic relationship between the performance indicator value and the variant of CHE function coupled to MOEA/D, 2) a parameter tuning scheme (proposed in Section 5.1) to determine the best ranges that improve the convergence and distribution of solutions along the Pareto front. Additionally, we also investigate their scalability up to 10 objectives and provide guidelines to apply the CHE functions in an online adaptive algorithm.

Sensitivity Analysis

We compute the sensitivity degree of the model parameter for the augmented CHE functions (see equations 3.13, 3.14, 3.16, 3.17 and 3.18 in Chapter 3) testing 10 different values of $\alpha = \{0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 3.0, 5.0, 7.0, 10.0\}$ on $DTLZ1$, $DTLZ3$ and $DTLZ3^{(-1)6}$ which have linear, convex and concave Pareto front shapes. These test problems were adopted with 3, 5, 7 and 10 objectives using the MOEA/D framework. Appendix B shows Tables C.1 and C.4 with the results corresponding to the 30 independent runs performed for HV. The reference point used in all problems was set to $(3, 3, \ldots, 3)^T$. The best performance indicator for each augmented CHE function is shown in **boldface** and the best global value found are highlighted with a gray tone.

For a better understanding of the effect of the α value, we compute the Spearman correlation coefficient (ρ). This method is a nonparametric statistical technique that aims to establish the relationship between two variables. In our case, these variables are set by different α values and a performance measure. The coefficient ρ is set in the range $[-1, +1]$. When the absolute ρ value is close to 1, it indicates a monotonic relation. Otherwise, a near-zero value suggests no association. We executed 30 independent runs to compute descriptive statistics of the Spearman's correlation for the HV and IGD+ indicators concerning a specific Pareto front geometry. Table 5.10 shows these results. When the absolute ρ value is greater than 0.5 is highlighted with a gray tone. In general, we can say that the HV results do not show big differences among the augmented CHE functions. Regarding scalability in objective space, we can identify the CHE functions with a monotonic relation. These cases are: $DTLZ1$ with RMCHE and $\alpha = 0.0001$ is the best option in all dimensions. $DTLZ3$ with ACHE (with $\alpha < 1$), RCHE ($\alpha = 0.1$ in 3,5 and 7 objectives or $\alpha = 0.01$ in 10 objectives) RMCHE ($\alpha < 1.0$). $DTLZ3^{-1}$ with ACHE ($\alpha = 0.01$ with $m = 3, 5, 10$ and $\alpha = 0.1$ with 7 objectives), RCHE ($\alpha = 0.1$ with $m = 3, 5, 7$ and $\alpha = 0.01$ in 10 objectives) and RMCHE ($\alpha < 1.0$). Although RACHE is the reciprocal version of ACHE, both obtain similar results with the same α values. However, RACHE has a more significant number of successes than the other variants. We can observe cases where the α value increases

[6]This is a projection of DTLZ3 function to generate a Pareto front with a convex shape.

while the HV value decreases. This occurs in the ACHE function solving $DTLZ1$ with 10 objectives and $DTLZ3$ with 7 objectives.

On the other hand, Table 5.10 shows in almost all cases significant differences regarding the IGD+ indicator. The following CHE functions present a high correlation in the IGD+ indicator. $DTLZ1$ with ACHE ($\alpha > 1.0$) and AASF ($\alpha \leq 0.01$), $DTLZ3$ with ACHE, RACHE, RMCHE and AASF, $DTLZ3^{-1}$ with ACHE, RACHE and RMCHE. In all cases, the best HV value is obtained using $\alpha \leq 0.1$. We can observe that the IGD+ value is improved in several cases where the α value increases or decreases. For example: ACHE in $DTLZ1$ with $m = \{5, 7, 10\}$, RACHE in $DTLZ1$ with $m = \{5, 7, 10\}$, AASF in $DTLZ1$ with $m = \{7, 10\}$ and RMCHE in $DTLZ3^{-1}$ with 10 objectives.

Table 5.10: Descriptive statistics of Spearman correlation coefficients (ρ) with respect to the HV indicator on $DTLZ1$, $DTLZ3$ and $DTLZ3^{-1}$.

MOP	Stats	**HV**					**IGD+**				
		ACHE	RACHE	MCHE	RMCHE	AASF	ACHE	RACHE	MCHE	RMCHE	AASF
DTLZ1	min	-0.7697	-0.7939	0.0749	-0.9515	-0.9483	-0.9515	-0.8667	-0.7091	-0.8303	0.9515
	max	0.2128	-0.1636	0.4369	-0.7455	0.5222	-0.8545	0.9273	0.1273	0.9758	0.9879
	mean	-0.2483	-0.5030	0.2138	-0.8697	-0.2062	-0.9152	-0.0182	-0.1667	0.1061	0.9697
	med	-0.2182	-0.5273	0.1716	-0.8909	-0.1993	-0.9273	-0.0667	-0.0424	0.1394	0.9697
	std	0.3701	0.2543	0.1413	0.0801	0.7285	0.0364	0.8234	0.3207	0.7296	0.0136
DTLZ3	min	-0.8788	-0.9394	-0.503	-0.9515	-0.7212	0.8424	0.8424	-0.3697	0.8909	0.4667
	max	-0.8424	-0.6606	0.6	-0.8545	-0.4424	0.9273	0.9394	0.8545	0.9758	0.697
	mean	-0.8697	-0.8455	0.0727	-0.8970	-0.5515	0.8879	0.8879	0.2273	0.9303	0.6030
	med	-0.8788	-0.8909	0.097	-0.8909	-0.5212	0.8909	0.8848	0.2121	0.9273	0.6242
	std	0.0157	0.1099	0.4608	0.0359	0.1134	0.0313	0.0347	0.4372	0.0347	0.0880
DTLZ3^{-1}	min	-0.9515	-0.9515	-0.7576	-0.9879	0.0303	0.8788	0.7818	-0.5273	0.6485	-0.7576
	max	-0.9273	-0.8545	-0.0667	-0.9273	0.3576	0.9273	0.8909	0.3576	0.9879	0.8667
	mean	-0.9455	-0.8939	-0.4364	-0.9485	0.1848	0.9030	0.8455	0.0727	0.8606	0.1939
	med	-0.9515	-0.8848	-0.4606	-0.9394	0.1758	0.9030	0.8545	0.2303	0.9030	0.3333
	std	0.0105	0.0357	0.2485	0.0248	0.1269	0.0192	0.0432	0.3505	0.1284	0.6706

To illustrate the effect of the distribution of the optimal solutions, Table 5.11 shows figures with the three basic Pareto fronts shapes (linear, concave and convex). Here, a small α value can achieve a better distribution. When we use the range $0.0001 < \alpha < 0.1$, the solutions are distributed in the extreme and the knee of the Pareto front. This effect increases the HV values. However, for values close to 1.0, the augmented CHE function cannot correctly generate Pareto optimal solutions (this is more evident in concave Pareto front shapes).

Parameter tuning process

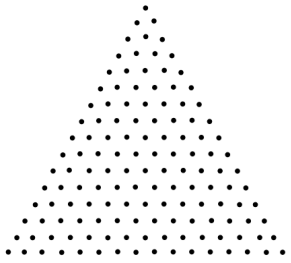
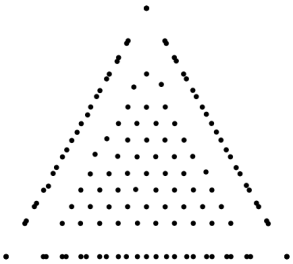
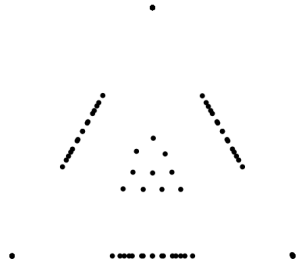
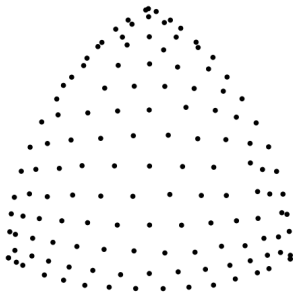
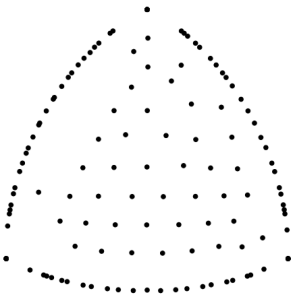
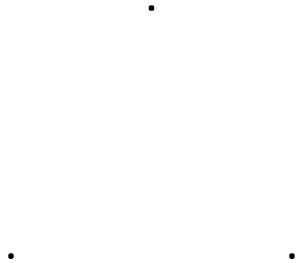
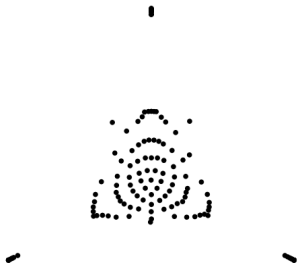
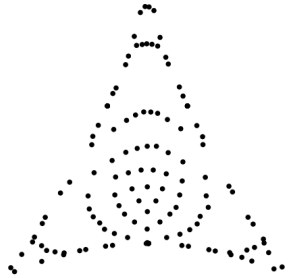
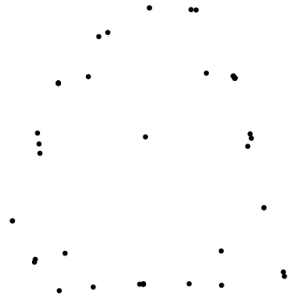
We use the same scenarios of the previous case study: convex, linear, concave, mixed and degenerated in multi- and many-objective cases.

The quality of a Pareto front approximation obtained by a particular configuration is assessed independently using two different performance indicators: the normalized hypervolume indicator and the modified Inverted Generational Distance (IGD+) [75]. A random sampling of Pareto optimal points establishes the reference set used by the IGD+ indicator.

The parameter α was set in the range $[0.0001, 10.0]$ according to the guidelines provided in the specialized literature.

The best-ranked calibrations are presented in Table 5.12. The most commonly recommended α value (0.0001) in the specialized literature is highlighted with a gray tone. We

Table 5.11: Effect of the model parameter of the augmented CHE functions.

$\alpha = 0.0001$	$\alpha = 0.1$	$\alpha = 1.0$
AASF, $DTLZ1$		
		
RACHE, $DTLZ3$		
		
MCHE, $DTLZ3^{-1}$		
		

can observe that 24% of EVOCA's recommendations according to the HV indicator and 50% according to the IGD+ indicator correspond to $\alpha = 0.0001$. Next, we mention the most relevant observations in the results obtained by EVOCA. ACHE and RACHE improve its performance in both indicators (HV and IGD+) with α values greater than 3 in multi- and many-objective optimization problems. In MOPs with a concave Pareto front, the majority of augmented CHE functions improve HV and IGD+ with values near to or equal to 0.0001, except for RACHE and AASF in the HV indicator. In the case of MOPs with convex and mixed Pareto fronts, AASF prefers a large value concerning others. RACHE recommends

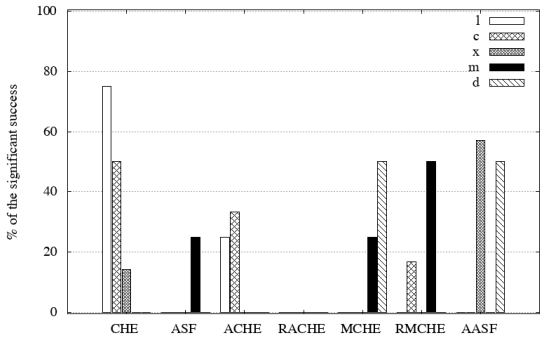
α values greater than 0.5 in degenerated Pareto fronts for multi- and many-objective problems. Here, the set of values recommendations reported is more diverse than in other cases.

Table 5.12: EVOCA's recommendations according to the HV and IGD+ indicators.

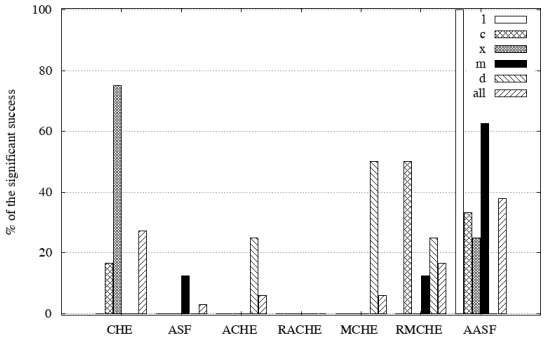
Dim	shape	HV					IGD+				
		ACHE	RACHE	MCHE	RMCHE	AASF	ACHE	RACHE	MCHE	RMCHE	AASF
multi-case	l	3.3335	6.7491	1.0001	0.0001	1.5868	3.3335	6.5001	0.0001	0.0001	0.0001
	c	0.0001	1.0001	0.0001	0.0001	0.8784	0.0001	0.0001	0.0001	0.0001	0.0001
	x	0.5534	1.0001	0.5534	1.0001	9.6872	0.0001	0.0001	0.0001	0.0596	9.5357
	m	0.0001	0.5001	0.1343	0.0001	0.2312	0.0224	0.0547	0.0224	0.0001	0.0288
	d	0.5001	1.7368	0.0253	0.0253	1.0001	0.2774	2.8820	0.3514	0.2312	0.0001
many-case	l	7.5001	9.2402	0.5001	0.5001	0.5001	7.5001	1.5868	0.3565	0.0001	0.0001
	c	0.0001	2.0001	0.0001	0.0001	0.7567	0.0001	0.0547	0.0001	0.0001	0.0001
	x	0.5402	0.3545	0.5402	0.5402	9.8990	0.0001	0.0001	0.0090	0.0001	10.000
	m	0.0143	0.0143	0.3565	0.3565	2.5357	0.0001	0.0090	0.0001	0.0001	4.5001
	d	0.0001	0.5534	0.4174	0.0001	0.0001	0.2774	9.0001	0.2895	0.2312	0.1392

We compute both HV and IGD+, for each test problem using EVOCA's recommendations. Tables C.7, C.8, C.9 and C.10 show the mean and standard deviation corresponding to 30 independent runs. Also, we applied the Wilcoxon rank test with a 99% of confidence to corroborate that the best SF found is statistically significant. The symbol (↑) means that the CHE SF was outperformed by the best option highlighted with a gray tone. Conversely, a (↓) symbol means that the difference between the best option and another CHE model is not significant. We count only the best significant indicator values per each SF and compute a percentage of success. Table 5.13 shows bar plots to compare the SF's performance per scenario and the global cases that consider all the test problem instances. For the whole set of test problems (11 functions and 6 different dimensions), the most promising SFs considering the HV indicator are AASF, CHE and RMCHE with a 37.88%, 27.27% and 16.66% of statistically significant values, respectively. While the IGD+ indicator presents a wider variety of options being the best: AASF(19.7%), ASF(18.18%), RACHE and MCHE (both 13.64%). Considering the HV indicator, the best options for linear Pareto front shapes are the CHE and AASF models in multi- and many-objective problems, respectively. CHE is better than ASF in concave shapes but in the mixed cases, ASF outperforms CHE. MCHE properly solves WFG3, which has a degenerated Pareto front in multi-objective problems. ACHE and MCHE work better for many-objective problems. Regarding the IGD+ indicator, any EVOCA's recommendations can solve more than 50% of the test instances per scenario. The most promising function to improve the diversity in the many-objective case is MCHE for convex shapes, while AASF can properly handle mixed shapes.

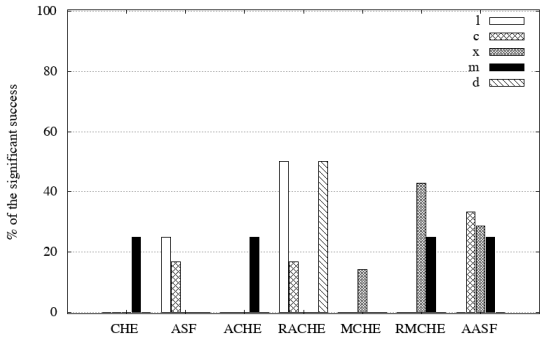
Table 5.13: Percentage of success regarding the performance indicators.



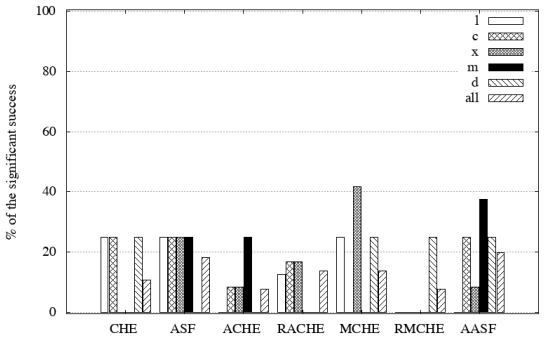
HV indicator in the multi-case scenario



HV indicator in the many-case scenario



IGD+ indicator in the multi-case scenario



IGD+ indicator in the many-case scenario

5.3 Robustness measure in Offline parameter tuning

The offline parameter tuning applied to MOEAs has the goal of finding *the best* algorithmic configuration that successfully solves a large number of MOPs. According to The No Free Lunch Theorem [163], no stochastic search algorithm can have the best performance in all classes of optimization problems. However, it is possible to find an appropriate set of parameters of an algorithm for solving a particular class of problems. For that sake, we need to study how to estimate the aggregate quality function for an algorithmic configuration assessed on a set of optimization problems. In MOEAs, each parameter has a significant influence on their performance such as its rate of convergence and on the quality of the solutions obtained. Offline parameter tuning is subject to uncertainty according to two aspects: 1) the stochastic procedures involved in MOEAs, 2) the optimal configuration for an MOEA optimizing a set of problem instances at the same time.

The goal of this section is to study robustness measures used in the area of uncertainty [8,45] to compute the performance of an algorithmic configuration used in the offline tuning methods. Our main contribution is to provide a set of guidelines on the use of robustness measures for solving multi-objective optimization problems (MOPs).

The typical goal in optimization is to identify optimal solutions. In the case of parameter settings, we want to optimize the configuration $c \in C$ of an MOEA on a scenario O. As defined in the previous chapter, O contains several problems $o_{1..k}$ to be optimized in their turn. However, the optimal configuration for solving one problem o_i can be poor for another

problem o_j of one scenario O. Thus, the additional challenge is to have a set of configurations that have a 'good' performance in a set of scenarios. The main issue is how to define this performance. To do so, we propose to use *robustness* measures, by incorporating some of the concepts of *robust* optimization [53], to solve the offline parameter tuning problem represented by Equation (5.2).

$$\max_{c \in C, o \in O} \psi_A(c, o). \tag{5.2}$$

In order to do this, we substitute the nominal objective for a robust measure that aggregates on the results from the different problems measured by scalar fitness function ψ. The robust measures that we use in this work are the following:

- Mean: it is one of the most commonly used in the literature. With this measure, we would be looking for the configuration that works best in the mean of the cases. However, this measure is not appropiate when results contain outliers since an outlier value has a significant impact on the mean value itself.

- Median: it is quite useful since it removes outliers. However, removing outliers would mean that we are not interested in all cases but rather in most of them. Thus, using this measure would mean that we are interested in the configuration that works well for most scenarios but it could fail completely in the others.

- Worst case: in this case, the fitness would be represented by the worst result in the given scenario. This would optimize over worst cases and thus ensure that the configuration would work at least with that quality. However, this measure can be over-conservative, since the worst case could never happen in practice.

Each of those measures has some advantages and drawbacks. The decision to use them should include the preference of the decision maker as well as the aim of the algorithm that we are tuning.

Moreover, since the MOEA has uncertainty itself (the same configuration can give different quality), it is also required to treat it in the same form as for the scenario. We first approximate the fitness for the MOEA and afterwards for the scenario. Namely, we combine: mean-mean, median-median, best-worst, mean-worst, median-worst and worst-worst. We make an emphasis on the worst-case since we are aiming for one configuration that works in all problem instances.

Our two particular goals for these experiments are: 1) to verify which measures are the most useful in three scenarios characterized according to the Pareto front geometry, and 2) to identify the most robust configurations when scaling up the number of objective functions in a MOP.

We use the next elements in the offline parameter tuning process:

- MOEA/D framework.

- $P := \{CHE, ASF, AASF, EWC, VADS, PBI\}$ with their corresponding model parameter values defined in the ranges $\alpha \in [0, 10.0]$, $p \in [0.1, 10.0]$, $t \in [1, 100]$, and $\theta \in [0.1, 50.0]$.

- Scenarios:

Table 5.14: EVOCA's recommendation for different scenarios using 3,5,7 objectives

Robustness measures	**Geometry**		
	Convex	**Concave**	**Linear**
mean-mean	AASF ($\alpha = 5.3727$)	VADS ($p = 11.9$)	PBI ($\theta = 15.9$)
median-median	EWC ($p = 7.2$)	PBI ($\theta = 8.2$)	PBI ($\theta = 10.4$)
best-worst	CHE	PBI ($\theta = 2.6$)	AASF ($\alpha = 0.0469$)
worst-worst	AASF ($\alpha = 1.5305$)	VADS ($p = 6.3$)	PBI ($\theta = 4.2$)
mean-worst	AASF ($\alpha = 0.6977$)	VADS ($p = 11.9$)	PBI ($\theta = 8.4$)
median-worst	AASF ($\alpha = 1.4065$)	VADS ($p = 12.1$)	PBI ($\theta = 11.3$)

1. *convex* = {DTLZ3^{-1}, LS ($\gamma = 0.3$), LS ($\gamma = 0.5$)}
2. *linear* = {DTLZ1, LS ($\gamma = 1.0$)}
3. *concave* = {DTLZ3, LS ($\gamma = 2.0$), LS ($\gamma = 4.0$)}

We tested this approach for $m = 3, 5, 7$ objectives.

- $\psi :=$ the *normalized hypervolume* (NHV).

The general results reached on this section are presented in Table 5.14 that shows the SFs obtained by EVOCA for each of the approaches in the different scenarios related to the Pareto front shapes. It is interesting to see that the approaches select not only different parameters for the scalarizing function but also different functions in several cases. This is an interesting result since it shows the impact of changing the robustness measure that is adopted. For all scenarios, the experimental results are shown in Table 5.15. In gray, we show the best result among the different robustness measures. An arrow upwards indicates that the approach is outperformed in a significant way by the baseline algorithm (EVOCA with mean-mean). An arrow downwards means that the baseline algorithm outperforms the robustness measure. Furthermore, Figures C.1, C.2 and C.3 show the box plots for the different scenarios using the hypervolume indicator without applying a normalization process. Marks a, b, c, d, e, f correspond to each robustness measures presented in Table 5.15. We can see in Figure C.1 that EVOCA's recommendation for a convex scenario outperforms the baseline version only in Lamé supersphere with $\alpha = 0.3$ and $\alpha = 0.5$ using *median-median* and *best-worst* measures. But in multifrontal MOPs *median-median* fails and other measures obtain similar results. This is because the median statistics discard the outlier results. Figure C.2 shows the linear scenario, where there is an evident tendency: the *best-worst* measure outperforms the baseline version in MOPs with more than 3 objectives. Finally, we obtain similar results in Figure C.3, where the EVOCA's recommendation improve the baseline version only in unimodal MOPs.

From the results, it is possible to observe that the best-worst approach, is capable of outperforming the base algorithm in most cases when the problems are unimodal. A similar case occurs with the median. However, for the multi-modal problems we can observe a deterioration of the quality of the results. This suggests that such problems should be in a different scenario and have their own configuration to find good solutions.

Also, we can notice that the configurations found work well while increasing the number of objectives. Further, we can observe that the different robustness measures outperform the baseline algorithm in most of the problems (aside from the multi-modal problems). This suggests that alternative measures to treat uncertainty can have a positive impact while searching for configurations for a set of scenarios.

Table 5.15: Results reported by EVOCA with different robust measures for different scenarios

Sc	Geo	Dim	mean-mean	median-median	best-worst	worst-worst	mean-worst	med-worst
Convex	L $\gamma = 0.3$	03D	0.99949(7.6989e-06)	↑ 0.99994(2.4944e-08)	↑0.9968(7.6713e-07)	↓0.99937(1.6849e-05)	↓0.99916(2.2555e-05)	↓0.99935(1.4162e-05)
		05D	0.99978(3.6113e-06)	↑ 1.0000(4.0000e-08)	↑0.99999(4.4408e-16)	↓0.99976(5.9568e-06)	↓0.99973(1.1476e-05)	↓0.99976(6.4822e-06)
		07D	0.99982(8.5306e-06)	↑ 1.0000(1.7950e-08)	↑ 1.0000(0.0000e+00)	↓0.99981(8.6050e-06)	↓0.99979(1.0696e-05)	↓0.99981(9.7633e-06)
	L $\gamma = 0.5$	03D	0.99778(3.6635e-06)	↓0.99758(4.4021e-07)	↓0.99557(3.0983e-06)	↓0.99768(9.3282e-06)	↓0.99748(1.5878e-05)	↓0.99767(8.3159e-06)
		05D	0.99976(9.1288e-06)	↑ 0.99997(5.1207e-08)	↑0.99993(2.5377e-07)	↓0.99973(1.0707e-05)	↓0.99968(1.9492e-05)	↓0.99973(1.0752e-05)
		07D	0.99981(2.0659e-05)	↑ 0.99999(3.3306e-16)	↑0.99999(4.4457e-06)	↓0.99978(2.2967e-05)	↓0.99975(2.4902e-05)	↓0.99977(2.8233e-05)
DTLZ3^{-1}		03D	0.99335(2.0098e-04)	↓0.0000(0.0000e+00)	↓0.99010(3.0459e-04)	0.99334(1.6041e-04)	0.99340(1.3607e-04)	0.99342(7.1454e-05)
		05D	0.99991(1.4226e-05)	↓0.0000(0.0000e+00)	↓0.99978(1.0693e-05)	↓0.99989(2.1022e-05)	↓0.99983(4.3120e-05)	↓0.99989(2.0928e-05)
		07D	0.99996(1.4068e-05)	↓0.0000(0.0000e+00)	↓0.99983(2.1933e-04)	↓0.99995(2.5509e-05)	↓0.99994(2.3587e-05)	↓0.99995(2.0053e-05)
Linear	L $\gamma = 1.0$	03D	0.97481(9.1966e-07)	0.97481(3.1155e-07)	↓0.97481(1.1706e-06)	0.97481(2.3795e-07)	0.97481(2.6042e-07)	0.97481(3.3757e-07)
		05D	0.99886(1.4922e-07)	0.99886(1.4083e-07)	↑ 0.99886(4.1831e-07)	0.99886(2.8511e-07)	0.99886(1.5205e-07)	0.99886(1.3743e-07)
		07D	0.99994(2.8481e-07)	0.99994(4.0573e-07)	↑ 0.99994(3.3306e-16)	↓0.99994(9.7262e-07)	↓0.99994(4.9379e-07)	↑0.99994(2.5289e-07)
	DTLZ1	03D	0.99681(3.2364e-05)	0.99681(2.5263e-05)	↑ 0.99683(1.1418e-05)	0.99681(2.3043e-05)	0.99680(2.7473e-05)	0.99681(2.3141e-05)
		05D	0.99995(3.9159e-06)	0.99995(3.7555e-06)	↑ 0.99996(5.0619e-07)	↓0.99995(3.9932e-06)	0.99995(2.3360e-06)	↑0.99995(3.7648e-06)
		07D	0.99999(2.5451e-06)	0.99999(3.7277e-06)	↑ 0.99999(4.3969e-07)	↓0.99997(1.1753e-05)	↓0.99998(4.1686e-06)	0.99999(2.9046e-06)
Concave	L $\gamma = 2.0$	03D	0.92821(2.0129e-06)	0.92821(1.2526e-06)	↑ 0.92821(1.3764e-06)	↑ 0.92821(1.0095e-06)	↑0.92821(2.0129e-06)	0.92821(2.0137e-06)
		05D	0.99017(1.2045e-06)	0.99017(9.6148e-07)	↑ 0.99017(6.5248e-07)	0.99017(1.0589e-06)	↑0.99017(1.2045e-06)	0.99017(1.1618e-06)
		07D	0.99863(1.5886e-06)	↑0.99863(1.0217e-06)	↑ 0.99863(7.1644e-07)	0.99863(2.2111e-06)	↑0.99863(1.5886e-06)	0.99863(1.9602e-06)
	L $\gamma = 4.0$	03D	0.89376(1.8281e-05)	↑ 0.89419(1.3943e-05)	↑0.89419(1.2420e-05)	↓0.89328(1.2611e-05)	↑0.89376(1.8281e-05)	0.89377(1.8035e-05)
		05D	0.97715(1.1496e-05)	↑ 0.97754(8.2635e-06)	↑0.97753(1.0646e-05)	↓0.97682(1.0701e-05)	↑0.97715(1.1496e-05)	↑0.97716(1.2953e-05)
		07D	0.99482(5.2873e-05)	↑ 0.99500(3.4305e-05)	↑0.99494(6.3347e-05)	↓0.99479(4.2571e-05)	↑0.99482(5.2873e-05)	0.99483(4.2165e-05)
DTLZ3		03D	0.92623(8.8163e-04)	↓0.91765(6.6721e-03)	↓0.92330(1.9209e-03)	0.92632(6.7448e-04)	↑0.92623(8.8163e-04)	0.92643(1.0158e-03)
		05D	0.98925(3.7990e-04)	↓0.66321(4.5688e-01)	↓0.98887(4.2865e-04)	0.98921(3.2035e-04)	↑ 0.98925(3.7990e-04)	0.98907(4.3676e-04)
		07D	0.99051(3.6548e-02)	↓0.67700(4.3885e-01)	↓0.96755(1.1470e-01)	↑ 0.99781(3.7914e-04)	↑0.99051(3.6548e-02)	0.93313(2.0597e-01)

5.4 Summary

In this chapter, we presented our experimental methodology for parameter tuning with the goal of determining the weighted and unconstrained SFs that are the most suitable for the MOEA/D and the MOMBI-II frameworks. For this purpose, we designed several test scenarios considering different Pareto front shapes and objectives. We used the tuning tool EVOCA to determine the best calibration for each of these scenarios. In almost all cases, EVOCA recommendations outperform the baseline version of these MOEAs. Our most important conclusion is that no unique scalarizing function performs best in all the scenarios but a set of them, regarding the normalized hypervolume indicator. In general, we obtained good results with AASF, PBI, EWC, and VADS. These SFs deserve further research attention.

We examined the choice of an SF based on the Chebyshev model in MOEA/D with the aim of characterizing test problems via their Pareto front shape and number of objective functions. Good results were obtained from the AASF over various settings of test problems. Regarding the specification of the α parameter in the augmented CHE functions, we derived the following interesting observations:

- We identified a correlation between the increase of the α value and the distribution of optimal solution for the augmented SF. This relation depends on the Pareto front geometry.
- The EVOCA framework found that a small α value is useful for concave Pareto fronts in almost all augmented CHE functions. However, some large values work better for solving linear, convex and degenerated Pareto front shapes.
- Regarding the IGD+ indicator, the seven CHE functions have a similar performance, and none of them is better than the others.
- One of the most promising versions of the CHE function is AASF which is able to improve both the HV and IGD+ values.

In the next chapter, we study and design adaptive models that combine various SFs regarding the most promising results presented here. We propose an adaptive strategy for the α value in the SFs based on the CHE model. Also, we propose a co-evolutionary scheme to adapt the weight vectors, according to the performance reached by the PBI function.

Chapter 6

Adaptive Strategies for Scalarizing Functions

In decomposition based MOEAs, we can examine three different spaces: 1) decision variable space, 2) objective function space, and 3) weight vector space. In this chapter, we focus our interest in the last two to adopt adaptive strategies in an online manner to select an appropriate Scalarizing Function (SF) during the evolutionary search process of MOEA/D with the goal of improving the quality of Pareto optimal solutions in multi- and many-objective optimization problems (MOPs and MaOPs).

The objective and the weight vector spaces are defined in $\mathbb{R}^m$, where m is the number of functions involved in the original MOP (see Equation (2.1)). The weight vector space is composed by a set of convex vectors defined in an $m-$simplex (see Chapter 3) to establish the search directions for each scalar subproblem. Methods based on low discrepancy sequences or the Simplex Lattice Design (SLD) [123] are usually employed to generate weight vectors with a uniform distribution around this $m-$dimensional space. However, a uniform distribution on weight vector space does not guarantee obtaining a uniform distribution in the approximation of Pareto solutions because it depends on several issues, in cases such as the following:

- MOPs with complicated Pareto front geometries such as mixed shapes (i.e. linear, convex or concave subregions in the same Pareto front), disconnected regions or degenerated fronts (see Figures 6.1 and 6.2).
- In Many-objective Optimization Problems (MaOPs) ($m > 3$) is complicated to obtain a good population diversity. A visualization tool such as the parallel coordinates plot can give us a general idea about the distribution of the Pareto front solutions (see Figure 6.3).
- The use of non-reciprocal scalarizing functions generates candidate solutions in different locations of the target directions. For example, the Chebyshev (CHE) function generates solutions in the extremes and the knee of the Pareto front, see Table 3.1.
- The augmented scalarizing functions such as AASF, MCHE or RASF (see Chapter 5) involve model parameters that form angles between target directions and the generated candidate solutions (see Figures 6.4 and 6.5).

Therefore, MOEAs strongly depends on the choice of the Scalarizing Function (SF). Methods such as the Modified Weighted Chebyshev (MCHE), Penalty Boundary Intersection (PBI) and Augmented Achievement Scalarizing Function (AASF) have been found to be very effective for achieving both convergence to the true Pareto front and a uniform distribution of solutions along it. However, the choice of an appropriate model parameter is

required for these SFs. In this chapter, we want to combine the strengths and compensate for the weaknesses of different SFs by employing an online method. Our proposed approach uses collaborative subpopulations to establish the best model parameter for each MOP. We also investigate the scalability of our proposed approach using up to 10 objectives, adopting several benchmark problems. Our preliminary results give rise to some interesting observations regarding the way in which different SFs are combined and adapted during the evolutionary process of MOEA/D.

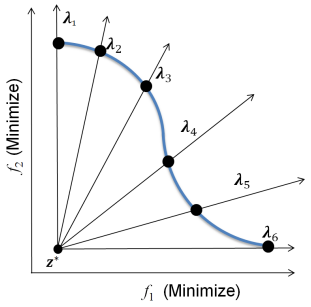


Figure 6.1: Mixed Pareto front shape.

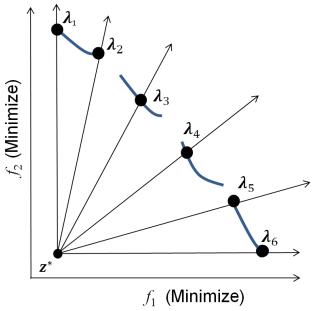


Figure 6.2: Disconnected Pareto front shape.

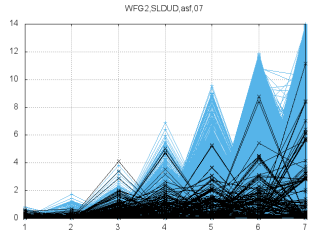


Figure 6.3: Parallel coordinates plot in a 7-objective MOP.

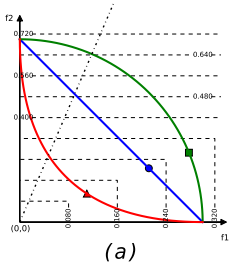


Figure 6.4: The CHE function in a bi-objective MOP.

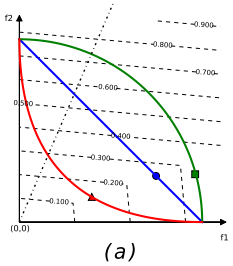


Figure 6.5: The MCHE function in a bi-objective MOP.

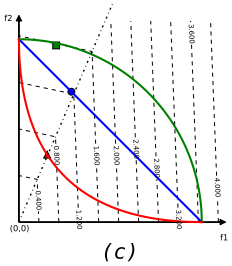


Figure 6.6: The AASF function in a bi-objective MOP.

6.1 Analysis of Convergence Speed in Scalarizing Functions

In the original MOEA/D all subproblems are treated equally assigning the same SF and model parameter value. However, recent studies such as [21], [175], and [87] have shown that some parts of the PF in a MOP can be more difficult to approximate than others. These works have employed dynamic resource allocation strategies to assign different computational resources to each subproblem. In this section, we present experiments to analyze how the convergence speed can vary for each subproblem depending on the SF and its parameter model. We tested diverse SFs and their model parameters to establish guidelines that allow us to combine more than one SFs during the search process of an MOEA and we can reduce the computational cost required to solve the original MOP.

We employed the Differential Evolution (DE) algorithm in its rand/1/bin version to

optimize each scalar subproblem in an independent manner. Given a set of uniform weight vector generated via the SLD method, the DE algorithm optimizes the following equation:

$$\text{minimize} \quad g(\mathbf{f}(\mathbf{x}); \boldsymbol{\lambda}), \tag{6.1}$$

where g is a SF, $\mathbf{f}$ represents the original MOP, $\mathbf{x}$ is the vector of decision variables and $\boldsymbol{\lambda}$ is a weight vector that establishes a target direction.

We evaluated the minimum number of evaluation functions required to converge to the true Pareto optimal solutions corresponding to each subproblem. We tested the SFs based on the Chebyshev model, Penalty Boundary Intersection (PBI) and Weighted Norm (WN) functions wiht the following parameter values:

- ACHE, RACHE, AASF, MCHE, RMCHE with $\alpha \in \{0, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 3.0, 5.0, 7.0, 10.0\}$
- PBI with $\theta = \{0.1, 1.0, 2.0, 5.0, 10.0, 50.0\}$ and
- WN with $p = \{0.5, 1.0, 2.0, 3.0, 5.0, 10.0, 100.0\}$

We tested $DTLZ1$, $DTLZ3$ and $DTLZ3^{-}1$ which have linear, concave and convex Pareto front shapes, respectively. We adopted 2, 3, and 5 objectives. The number of weight vectors was 100, 120 and 210 for each dimension.

Figures 6.7, 6.8, 6.9, 6.10, 6.11 and 6.12 show the convergence plots of the most relevant results. In these figures, we can see that ACHE and MCHE are very sensitive to their model parameter. In these cases the most efficient α values are very small. Also, in convex and concave Pareto shapes (DTLZ3 and DTLZ3^{-}1), when $\alpha \geq 1.0$, the convergence speed drastically decreases. Appendix C describes the remainder of the experiments.

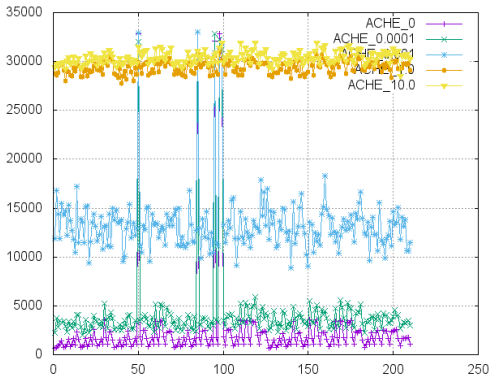


Figure 6.7: ACHE in DTLZ1 with 5 objectives.

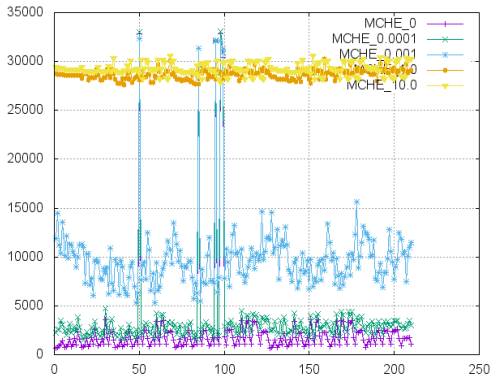


Figure 6.8: MCHE in DTLZ1 with 5 objectives.

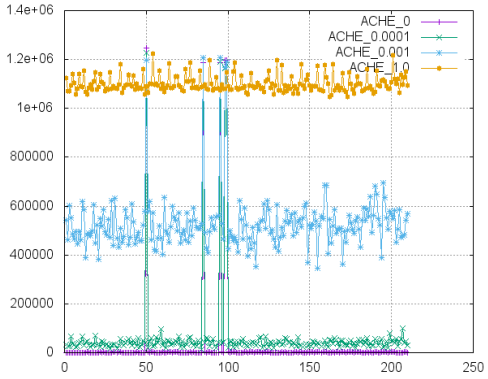


Figure 6.9: ACHE in DTLZ3 with 5 objectives.

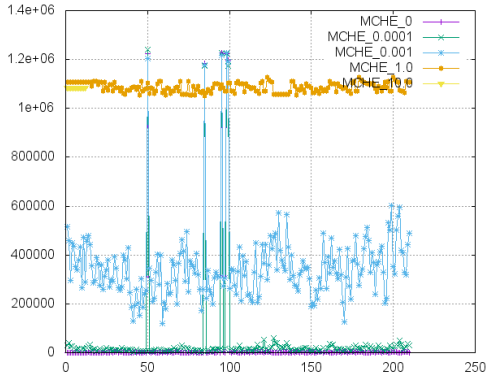


Figure 6.10: MCHE in DTLZ3 with 5 objectives.

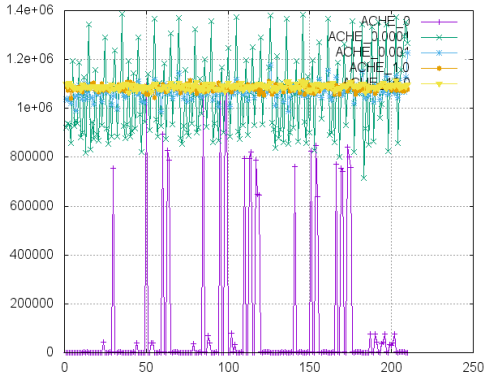


Figure 6.11: ACHE in DTLZ3^{-}1 with 5 objectives.

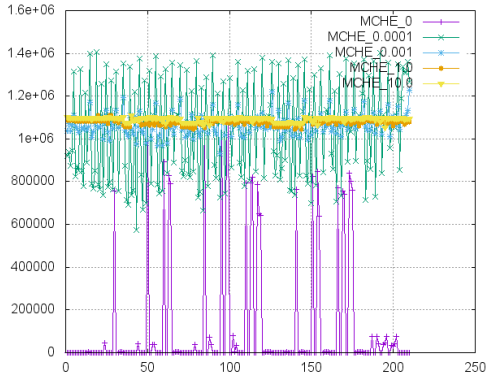


Figure 6.12: MCHE in DTLZ3^{-}1 with 5 objectives.

The results presented in this section show the importance of combining several parameter values or even more than one SF. This is particularly useful in cases where the MOPs present complicated Pareto front geometries, such as mixed or disconnected shapes.

6.2 Collaborative and adaptive strategy of different Scalarizing Functions

In this section, we propose a strategy based on collaborative populations combining different SFs and model parameter values via an adaptive operator selection based on the Multi-Objective Multi-Armed Bandit (MOMAB) technique, which consists on the following definition:

Definition 6.2.1 *Given n slot machines called "arms" with random rewards, the goals of MOMAB are:*

- *to maximize the returned reward,*

- *to minimize the regreat of pulling suboptimal arms and*
- *to identify the set of optimal arms*

In our case, we associate the term *arm* to a parameter setting in an MOEA, whereas the *reward* concept corresponds to the performance quality reached by this configuration. The bandit problem is formally equivalent to a one-state Markov decision process as an extension of Markov chains, where there are actions and rewards [17, 176]. This problem is also called *exploration and exploitation trade-off*, where our interest is to identify how many times to play each machine and in which order to play them. When a machine is played we require to establish a reward from a specific probability distribution. In other words, MOMAB is a useful adaptive allocation strategy to map states and actions to obtain a maximum reward [96].

Some MOEA/D improvements [103, 119] have employed MOMAB techniques as an Adaptive Operator Selection (AOS) mechanism to select the most suitable operator from a pool of options according to some determinist rule based on multi-armed bandit algorithms. Bandit-based AOS establish two tasks: 1) to determine a reward for each operator via its record of fitness improvement rates, and 2) to select an operator for being used according to the current reward value. For more details see Chapter 4.

The aim of this section is to adopt a MOMAB strategy, only used in evolutionary operators, to select an appropriate SF during the evolutionary search process of MOEA/D. Our proposed approach uses collaborative subpopulations to establish the best model parameter for each MOP. We also investigate the scalability of our proposed approach using up to 10 objectives, adopting several benchmark problems.

6.2.1 Our proposed approach

Next, we define some guidelines to combine more than one SF simultaneously, using an adaptive strategy selection and collaborative subpopulations. As indicated before, the mechanism adopted is a bandit-based AOS algorithm coupled to MOEA/D-DRA [170]. We analyze the effect that our proposed approach has on the performance of MOEA/D-DRA when solving MOPs with up to 10 objectives and complicated Pareto front shapes.

Pool of Strategies

As mentioned in Section 6, the CHE and ASF methods find optimal solutions in opposite target directions. Based on this and our prior experience, only the SFs with similar target directions will be combined simultaneously. Otherwise, the algorithm can not generate well-distributed solutions along the Pareto front. We propose the next pool of strategies for combining multiple SFs:

- $S_1 = \{ACHE, MCHE, WN\}$,
- $S_2 = \{AASF, MASF, PBI\}$,

The model parameters were suggested based on the values proposed by different authors:

- $\alpha = \{0.0, 0.0001, 0.001, 0.01, 0.5\}$,
- $p = \{0.5, 1.0, 2.0, 3.0, 5.0, 10.0, 100.0\}$,

- $\theta = \{0.1, 1.0, 2.0, 5.0, 10.0, 50.0\}$

These values consider three special cases: $\alpha = 0$ to include the ASF method, $p = 1$ for the WS function and $p = 100$ which is similar to the CHE function. Thus, $|S_1| = 17$ and $|S_2| = 16$ is the number of different configurations of SFs.

Collaborative populations

We pre-defined a set of uniformly distributed weight vectors using the simplex lattice design technique [139], to maintain a good diversity in MOEA/D-DRA. These weight vectors are divided into k subsets used by a subpopulation which optimizes one of the types of SF selected from a pool of options (S_1 or S_2). Each weight vector is assigned alternately to each subpopulation in the same manner as was mentioned in [80] (single grid implementation technique).

Our proposed approach uses a MOMAB method for deciding which SF should be employed at a time point in MOEA/D-DRA to solve different MOPs. Figure 6.13 shows the general idea of our proposal. Here, we can see an example where two subpopulations A and B are associated to the PBI ($\theta = 1.0$) and AASF ($\alpha = 0.1$) functions. These options were taken from a pool of strategies previously established.

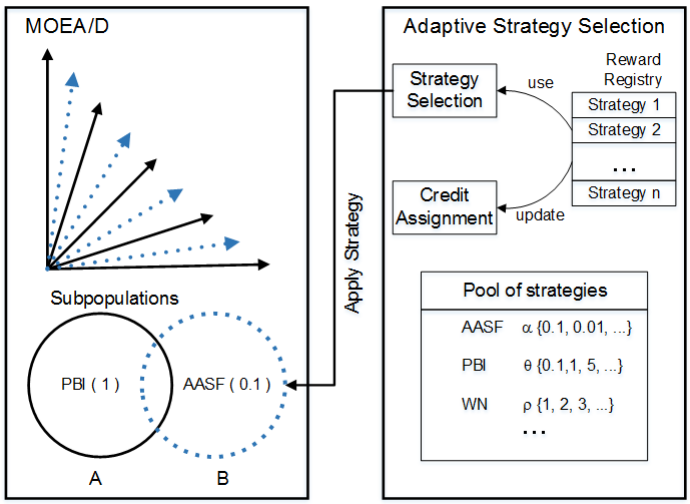


Figure 6.13: A general scheme of our proposed approach.

6.2.2 Adaptive strategy selection

As we can see in Section 6, the performance of a decomposition-based algorithm strongly depends on the selection of its SF. Some types of SF are more beneficial for a particular Pareto front shape or a certain number of objectives. We employ a MOMAB technique to select the most appropriate SF for each subproblem in a decomposition-based MOEA. Here, MOMAB considers two main aspects: to assign a credit value to each operator and to select one operator based on its historical performance. In our case, we used the term "operator" to refer to a SF and its model parameter. We adopted MOEA/D-DRA coupled to AOS based on the fitness improvement rates (FIR) [103], which is computed by each subproblem i at time t, as defined by equation (6.2).

$$FIR_{i,t} = \frac{g(\mathbf{x}^i|\boldsymbol{\lambda}^i, \mathbf{z}^*, \mathbf{f}') - g(\mathbf{y}|\boldsymbol{\lambda}^i, \mathbf{z}^*, \mathbf{f}')}{g(\mathbf{x}^i|\boldsymbol{\lambda}^i, \mathbf{z}^*, \mathbf{f}')}, \tag{6.2}$$

where $\mathbf{x}^i$ is the current solution and $\mathbf{y}$ is its generated offspring solution after applying the genetic operators. Function g is a specific SF selected from a pool of strategies. The aim of the FIR technique is to deal with the largest ranges of raw fitness values at different stages of the evolutionary search process [103]. We used a sliding window with a fixed size W and a first-in, first-out (FIFO) queue structure in order to store the FIR values of the recently used SF and its model parameter.

The reward value assigned to each strategy i is given by:

$$FRR_{i,t} = \frac{Decay_i}{\sum_{j=1}^{k} Decay_j}, \tag{6.3}$$

where $Decay_i = D^{rank_i} \times R_i$, $D \in [0, 1]$ is a decaying factor to increase the probability of selecting the best strategies, $rank_i$ is the rank assigned to each strategy (in descending order) and R_i (or reward) is the sum of all FIR values for each strategy i in the current sliding window.

We select the best SF using equation (6.4).

$$S_i = \arg\max \left\{ \frac{R_{k,t}}{\sum_{i=1}^{k} R_i} + C \times \sqrt{2 \times \frac{ln(\sum_{j=1}^{m} \eta_j)}{\eta_i}} \right\}, \tag{6.4}$$

where C is a weight factor to control the trade-off between exploration and exploration. η_i is the number of times that the strategy i was used.

6.2.3 Collaborative and adaptive strategies coupled to MOEA/D-DRA

MOEA/D with Dynamical Resource Allocation (MOEA/D-DRA) [170] is an improved version of MOEA/D [171], which was the winning algorithm in the CEC 2009 MOEA contest. MOEA/D-DRA incorporates a mechanism to compute the relative decrease of fitness improvement for each subproblem in order to assign computational effort according to the obtained benefits.

Next, we describe how to couple the adaptive strategy selection to MOEA/D-DRA. The first step is an initialization process. Then, we split the weight vectors into each subpopulation in order to assign a type of SF. Next, we use the adaptive strategy selection and associate one scalarizing function to each subpopulation in order to assign a different type of SF to each of them. At each generation, we monitor the FIR value for each subproblem of each subpopulation. Next, a dynamical resource allocation mechanism used in the original MOEA/D-DRA is applied. After that, a generation of new solutions via Differential Evolution and Polynomial-based mutation is employed. Finally, we update the reward subpopulation based on the FIR and Algorithm 23, while Algorithm 24 illustrates in more detail the steps described before.

Experimental settings

We divide our experiments in two parts. The first is focused on variations of the Pareto front geometry and the number of objectives using some MOPs defined in the *Deb-Thiele-*

Algorithm 22: Our proposed bandit-based operator selection mechanism

Input: P: A pool of scalarizing functions
Output: S_{new}: The new selected strategy

```
if There are scalarizing functions that have not been selected in P then
    S_i = one scalarizing function, which is selected randomly from the pool of strategies
else
    for S_i ∈ P do
        Update S_i using Equation (6.4)
S_new ← select the best strategy
return S_new
```

Algorithm 23: Credit assignment algorithm

Input: D: decay factor
Output: R: List of k reward values

```
Initialization:
for i ← 1 to k do
    Initialize each R[i] = 0
k = slidingWindow.length for i ← 1 to k do
    s = slidingWindow.GetIndexOp(i)
    FIR = slidingWindow.GetFIR(i)
    R[s] = R[s] + FRR
    for j ← 1 to s do
        Update FRR_{j,t} using Equation (6.3)
Rank R in descending order
return R
```

Laumanns-Zitzler (DTLZ) test suite [66]: DTLZ1 for linear, DTLZ3 for non-convex, DTLZ5 for degenerate and DTLZ7 for mixed Pareto front geometries. Additionally, we transformed DTLZ3 so that it had a convex shape. To test the scalability of our proposal, each problem was tried with 2,3,5,7 and 10 objectives. For the second set of experiments, we adopted complicated MOPs presented in the CEC 2009 contest. For a fair comparison, we used the same MOEA/D-DRA parameters in all the MOP instances. The neighborhood size T was set to 20% of the population size (p_size). The crossover and mutation parameters were set as: $F = 0.5$, $Cr = 1$. For the DTLZ test problems, $H = \{99, 14, 6, 4, 3\}$ was used by the simplex lattice design method. The population size was set to: $pop_s = \{100, 120, 210, 210, 220\}$ for $m = \{2, 3, 5, 7, 10\}$. The number of objective function evaluations was set to 40,000 for $m = 2$ and $m = 3$ and it was 50,000 for all the other cases. In the case of the UF test functions (i.e., those from the CEC 2009 contest), we adopted a population size of 600 for $m = 2$ and of 1000 for $m = 3$. The number of function evaluations was set to 300,000.

The parameters for the adaptive strategy selection were set as suggested in [103]. We used the decay factor $D = 1.0$ and a factor to control the exploitation and the exploration of $C = 5.0$ and $W = 0.5 \times p\_size$.

Algorithm 24: MOEA/D-DRA-MSF

Input: A stopping criterion, ns: number of subpopulations, pop_size: population size, $\{\lambda_1, \ldots, \lambda_N\}$: A well-distributed set of weight vectors, $pool$: A pre-defined pool of strategies
Output: Pareto front estimation

Step 1. Initialization
$E_p \leftarrow \emptyset$
Compute the Euclidean distance between any two weighted vectors and then work out the T closest weighted vectors to each weighted vector.
Generate an initial set of subpopulations $P_i = \{x_1, \ldots x_{N/ns}\}$.
Evaluate individuals in the initial subpopulations.
Set the ideal point $z_i^* = \arg\min\{f_i(x)\}$.
while *the stopping criterion is not satisfied* **do**
    Step 2. Adaptive Strategy selection.
    Select ns scalarizing functions g and model parameter from the pool of strategies, using Algorithm 22
    Associate one scalarizing function to each subpopulation, according to the partition strategy.
    for *each subpopulation* **do**
        Set $gen = 0$ and $\pi^i = 1$ for all $i = 1, \ldots, N$.
        Selection of subproblems for searching: the indexes of the subproblems whose objectives correspond to the MOP's individual objectives f_i are selected to form the initial I. By using a 10-tournament selection strategy based on π^i, select other $[\frac{N}{5}]m$ indexes and add them to I.
        for *each* $i \in I$, *do* **do**
            Step 3. Selection of Mating/Update Range:
            Uniformly randomly generate a number $rand$ in the range $(0, 1)$.
            Then set
            $$P = \begin{cases} B(i) & \text{if } rand < \delta, \\ \{1, \ldots N\} & \text{otherwise} \end{cases}$$
            Step 4 Reproduction: Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from P, and then generate a solution $\mathbf{y}$ from xr_1, xr_2 and xr_3 by a Differential Evolution (DE) operator, and then perform mutation on $\mathbf{y}$.
            Step 5. Update.
            Update $z*$:
            Update subproblem i, $g(\mathbf{x} : \boldsymbol{\lambda}, \mathbf{z}^*, \mathbf{f}')$
            Update Neighboring Solutions
            for $j \in B(i)$ **do**
                if $g(y'|\lambda_j, \mathbf{z}^*, \mathbf{f}') \leq g(x|\lambda_j, \mathbf{z}^*, \mathbf{f}')$ **then**
                    compute FIR based on equation (6.2)
                    set $x_j = y'$
            Step 6. Update Reward registry.
            Update credit assignment using Algorithm 23
            $gen = gen + 1$.
            If gen is a multiple of 50, then compute Δ_i, the relative decrease of the objective for each subproblem i during the last 50 generations, update
            $$\pi^i = \begin{cases} 1 & \text{if } \Delta^i > 0.001, \\ (0.95 + 0.05) * \frac{\Delta_i}{0.001}\pi^i & \text{otherwise} \end{cases}$$
$P \leftarrow$ non-dominated solutions from the population
return P

We performed 30 independent runs for each MOEA and problem instance. For comparing our results, we adopted the hypervolume indicator (HV) [179] to assess both conver-

gence and maximum spread. We established the following reference points: $(1, 1, \ldots 1)^T$ for DTLZ1, $(7, 7, \ldots 7)^T$ for DTLZ3 and DTLZ3_*convex*, $(4, 4, \ldots 4)^T$ for DTLZ5, $(21, 21, \ldots 21)^T$ for DTLZ7. $(2.0, \ldots, 2.0)^T$ for all the CEC 2009 test problems (UF1-10).

Discussion of results

In this subsection, we compare our proposed approach using multiple SFs (considering the two pools of strategies previously discussed) with respect to the original MOEA/D-DRA adopting only one SF (in our experiments, we used CHE, ASF, WN ($p = 2$) and PBI ($\theta = 5$)). Table 6.1 presents the hypervolume indicator for each DTLZ test problem. The best values are highlighted with a darker gray tone and the second best with a lighter tone. We applied the Wilcoxon rank sum test with a 95% of confidence level to corroborate that the best result found is statistically significant with respect to the others. The symbol ($\uparrow$) means that the best case (algorithmic configuration per problem) outperformed another algorithm in a significantly better way. The symbol ($\downarrow$) indicates that the difference between the best option and another algorithm is not significant.

We analyzed the results according to each proposed strategy (i.e., S_1 and S_2). In the same way, AASF, MAASF and PBI outperformed MOEA/D-DRA with ASF. One interesting observation is that the main improvements were obtained in the many-objective problems. For two objectives, the results were very similar among themselves. However, with 3 or more objectives there were some significant differences, especially in the seven- and ten-objective MOPs. We can notice that the standard deviation values increase when the MOP has multimodality such as in dtlz1, dtlz3 and dtlz3_convex. In general, the strategy S_1 is better than S_2 in the DTLZ test problems adopted.

In a second experiment, we compare our proposal with respect to state-of-the-art MOEAs such as the original version of MOEA/D-DRA [172], and with respect to MOEA/D-DRA-MAB [103] which used the same multi-armed bandit algorithm but applied to a pool of Differential Evolution operators. Moreover, we also compared results with respect to ADEMO/D [155] which includes a learning period strategy to select from a pool of DE operators.[1]

For assessing performance, we computed the IGD+ [77] and HV indicators (see Table 6.2), then we applied the Wilcoxon test in the same manner as done in our previous experiments. In almost all cases, MOEA/D-DRA with multiple SFs outperformed the other approaches with respect to which it was compared, but the most evident improvement was observed in UF5, UF6 and UF10 in both indicators (HV and IGD+).

[1]The source code for these algorithms was obtained from `https://coda-group.github.io/publications.html`

Table 6.1: Statistical results for strategies S_1 and S_2. We show the mean and the standard deviations (in parentheses)

m	MOP	CHE	WN($p=2$)	MOEA/D-DRA-MSF1	ASF	PBI($\theta=5$)	MOEA/D-DRA-MSF2
2	dtlz1	↑8.7122e-01(7.1742e-04)	8.7296e-01(3.4337e-01)	↑8.7034e-01(1.3440e-01)	↓8.7140e-01(5.2209e-04)	↑8.7189e-01(4.6566e-04)	↓8.7114e-01(5.4564e-02)
	dtlz3	4.8138e+01(1.3129e-02)	↑4.2977e+01(1.2991e+00)	↑4.8125e+01(1.1017e+02)	↑4.8134e+01(1.4122e-02)	↑4.3496e+01(1.4836e+00)	↓4.8131e+01(1.4154e-02)
	dtlz3*	↑4.8585e+01(1.4334e-02)	↑4.8008e+01(2.3724e+02)	↑3.9507e+01(1.6363e+00)	4.8622e+01(9.7710e-03)	↑4.8076e+01(1.1369e-01)	↓4.8570e+01(1.9431e-02)
	dtlz5	↑1.5209e+01(5.3291e-15)	↑1.5000e+01(0.0000e+00)	1.5210e+01(3.7268e-04)	↑1.4107e+01(2.7241e-05)	↑1.3207e+01(1.3291e-11)	↑1.5210e+01(2.0331e-03)
	dtlz7	↑3.6431e+02(1.2806e+01)	↑3.6581e+02(1.2689e+01)	3.6591e+02(1.2713e+00)	↑3.6591e+02(8.7256e+00)	↑3.6587e+02(9.5238e+00)	↑3.6589e+02(8.7220e+00)
3	dtlz1	4.9629e-01(1.1656e-02)	↑3.3465e-01(1.3951e+01)	↑3.3000e-03(9.3354e-02)	↓4.9626e-01(7.1114e-02)	↑3.2531e-03(1.2794e-01)	↓4.3143e-03(1.0891e-01)
	dtlz3	↑5.1501e+01(5.5516e+01)	↑1.9355e-01(1.7713e+03)	1.4713e+02(9.5746e+02)	↑1.4584e+02(4.2952e+01)	↑9.4666e+00(1.9882e+02)	↑9.5988e+01(5.4480e+01)
	3*	↑1.7266e+00(3.7764e+02)	9.0919e+01(1.4094e+03)	↓9.0237e+01(3.2449e+02)	↑8.5693e+01(3.6189e+02)	↑1.3596e+01(1.1116e+02)	↑3.3232e+01(8.4332e+01)
	dtlz5	4.3524e+01(3.3929e-02)	↑4.3373e+01(8.3057e-01)	↑4.3517e+01(1.9373e-02)	↑4.3224e+01(1.3812e-02)	↑4.2307e+01(4.4186e+00)	↑4.0095e+01(1.7415e+00)
	dtlz7	↑6.7217e+03(1.9440e+02)	↑6.4022e+03(2.9728e+02)	↑6.7218e+03(2.4975e+02)	↑6.7977e+03(1.9758e+02)	↑6.8171e+03(2.2749e+02)	6.8291e+03(1.6706e+02)
5	dtlz1	↑4.9929e-01(1.5924e-01)	↑2.5117e-01(6.1529e+02)	↑4.9972e-01(1.7369e-01)	8.5079e-01(7.3003e-02)	↑2.9759e-01(2.3140e-01)	↑8.1317e-01(9.2416e-02)
	dtlz3	↑5.8410e+03(3.1219e+03)	↑5.6629e+03(1.9979e+05)	↑4.4650e+03(3.2435e+03)	1.3322e+04(1.2718e+03)	↑2.3804e+03(4.8386e+04)	↑1.1450e+04(2.0792e+03)
	3*	↑1.2414e+04(5.1761e+04)	1.6408e+04(4.0656e+10)	↑8.0325e+03(2.0862e+03)	↑6.8641e+03(2.5797e+03)	↓1.1697e+04(1.7868e+03)	↑1.2897e+04(1.2760e+03)
	dtlz5	↑4.0830e+02(1.4979e+02)	↓6.0898e+02(2.3057e+01)	↑6.1298e+02(5.5498e+01)	↑2.8630e+02(8.1132e+01)	6.1298e+02(1.8570e+01)	↑6.2335e+02(1.1285e+02)
	dtlz7	↑2.3765e+06(1.6408e+05)	↑2.4342e+06(2.1796e+05)	2.5201e+06(1.3996e+05)	↑2.3798e+06(1.6727e+05)	↑2.3684e+06(1.5960e+05)	↑2.3782e+06(1.3541e+05)
7	dtlz1	↑3.8993e-01(8.9484e-02)	1.3453e+00(2.8435e+01)	↑4.9979e-01(1.3579e-01)	↑6.9847e-01(5.0892e-02)	↑7.3862e-01(4.2553e-02)	↓1.2867e-00(8.1133e-02)
	dtlz3	↑1.2494e+05(1.6655e+05)	↑1.1803e+05(7.6051e+06)	5.8770e+05(6.9250e+04)	↑4.7643e+05(7.2851e+04)	↑7.7523e+04(1.8189e+06)	↑5.3980e+05(6.6100e+04)
	3*	↑4.2864e+05(9.2783e+05)	↑6.6023e+05(3.6630e+04)	1.1480e+06(2.3005e+09)	↑4.5671e+05(9.6351e+04)	↑2.4514e+05(1.1749e+05)	↑4.5453e+05(9.9030e+04)
	dtlz5	↑2.3702e+03(4.3165e+03)	↑3.6759e+03(2.3452e+03)	↑9.2144e+03(9.7608e+02)	↑4.2746e+03(1.0812e+03)	↑4.2830e+03(1.1001e+03)	9.5339e+03(1.4573e+03)
	dtlz7	↑7.6908e+08(7.9234e+07)	↑7.6876e+08(1.3667e+08)	↑9.0892e+08(5.8621e+07)	↑7.7546e+08(9.7418e+07)	9.1073e+08(6.6888e+07)	↓8.4123e+08(7.5738e+07)
10	dtlz1	↑3.7893e-01(9.4529e-02)	↓5.0146e-01(8.2056e-02)	3.0936e+00(9.6209e+01)	↑3.5268e-01(9.9343e-02)	↑7.2013e-01(3.1656e-02)	↑6.5351e-01(5.9505e-02)
	dtlz3	↑5.7924e+07(5.0050e+07)	↑8.1510e+04(2.3712e+09)	1.6133e+08(2.8166e+07)	↑2.7653e+07(5.1446e+07)	↑8.1303e+07(1.8458e+08)	↑8.2516e+07(4.4397e+07)
	3*	↑4.7820e+07(1.6563e+08)	1.0000e+09(2.3990e+12)	↑2.1945e+08(1.5715e+07)	↑1.6077e+08(8.0149e+07)	↑1.0528e+08(4.3691e+07)	↑2.3384e+08(4.9907e+07)
	dtlz5	↑1.3934e+05(2.8259e+05)	↓2.3465e+05(1.4312e+05)	↓3.7277e+05(1.0007e+05)	↑3.9270e+05(4.3071e+04)	5.6638e+05(6.6277e+04)	↑5.5157e+05(2.7411e+04)
	dtlz7	1.0000e+09(6.9947e+11)	1.0000e+09(6.0816e+12)	1.0000e+09(1.1297e+12)	1.0000e+09(9.3084e+11)	1.0000e+09(7.3536e+11)	1.0000e+09(6.8013e+11)

Table 6.2: Statistical results for strategies S_1 and S_2. We show the mean and the standard deviations (in parentheses)

MOP	MOEA/D-DRA	MOEA/D-DRA-MAB	ADEMO/D	MOEA/D-DRA-MSF1	MOEA/D-DRA-MSF2
Hypervolume indicator					
UF1	↑3.5847(0.0587)	↓3.4397(0.0895)	↑3.5791(0.0373)	3.6611(0.0015)	↑3.6335(0.0204)
UF2	↑3.6026(0.0344)	↑3.5965(0.0193)	↑3.6325(0.0137)	3.6532(0.0112)	↓3.6396(0.0346)
UF3	↑3.4353(0.1568)	↑3.2093(0.1340)	3.4164(0.1265)	3.6551(0.0259)	↑3.3769(0.0696)
UF4	↑3.1783(0.0135)	↑3.1978(0.0107)	↓3.2337(0.0126)	↑3.2566(0.0101)	3.2878(0.0030)
UF5	↑0.7428(0.8705)	↑1.7446(0.2659)	↑1.8762(0.2609)	↑2.7135(0.3376)	3.0424(0.1003)
UF6	↑2.5232(0.2259)	↑2.6236(0.2125)	↑2.7444(0.1814)	↑2.8315(0.4389)	3.1201(0.1323)
UF7	↑3.4408(0.0391)	↑3.2580(0.3566)	↑3.3883(0.2606)	3.4870(0.0316)	↑3.4850(0.0194)
UF8	↑6.9568(0.3853)	↑6.9779(0.3591)	↓7.3229(0.0242)	7.4003(0.0226)	↑7.3145(0.0017)
UF9	↑6.9542(0.3346)	↑7.2106(0.2974)	↓7.2993(0.2163)	↓7.3362(0.2213)	7.6577(0.0348)
UF10	↑0.7238(1.0449)	↑4.4226(0.7018)	↑4.7172(0.9373)	↑4.0287(0.6905)	6.1128(0.1576)
IGD+ indicator					
UF1	↑0.0389(0.0291)	↑0.0392(0.0126)	↑0.0142(0.0033)	0.0013(0.0001)	↑0.0063(0.0023)
UF2	↑0.0271(0.0131)	↑0.0077(0.0011)	↑0.0035(0.0006)	0.0034(0.0012)	↑0.0040(0.0028)
UF3	↑0.1476(0.0986)	↑0.0712(0.0240)	↑0.0531(0.0256)	0.0037(0.0051)	↑0.0500(0.0092)
UF4	↑0.0559(0.0045)	↑0.0449(0.0015)	↑0.0333(0.0014)	↑0.0271(0.0019)	0.0172(0.0003)
UF5	↑1.2284(0.4610)	↑0.5648(0.1253)	↑0.5132(0.1058)	↓0.2564(0.1017)	0.2167(0.0385)
UF6	↑0.4302(0.1407)	↑0.1466(0.0741)	0.1211(0.0413)	↑0.1686(0.1105)	↑0.1428(0.0717)
UF7	↑0.0217(0.0152)	↑0.0603(0.0836)	↑0.0293(0.0638)	0.0019(0.0018)	↑0.0044(0.0011)
UF8	↑0.0693(0.0258)	↓0.0690(0.0213)	↑0.0348(0.0098)	0.0115(0.0108)	↑0.0526(0.0004)
UF9	↑0.2455(0.0462)	↑0.2217(0.0637)	↑0.2098(0.0494)	↑0.2123(0.0487)	0.1372(0.0048)
UF10	↑1.7270(0.7275)	↑0.2542(0.0528)	↓0.2394(0.0724)	↑0.2571(0.0757)	0.1544(0.0288)

6.3 Summary

In this chapter, we have presented a comparative study to determine some guidelines to combine, in a simultaneous way, several scalarizing functions. One of our most important contributions is that we have identified that the pool of scalarizing functions should establish the same target directions in order to generate well-distributed (i.e., uniform) solutions along the Pareto front. Also, we noticed that the appropriate parameters settings depend directly on the number of objectives and on the Pareto front geometry. We claim that the bandit-based AOS adopted in our proposed approach is a good option to detect appropriate parameters settings in SFs, while requiring a lower computational effort than the use of static parameter tuning strategies (offline tuning methods).

Chapter 7

Conclusions and Future Work

This thesis presented the importance of the use of Scalarizing Functions (SF) in the Multi-Objective Evolutionary Algorithms (MOEAs) based on decomposition and the R2 indicator to transform the original Multi-Objective Optimization Problem (MOP) into several single objective problems with the aim of maintaining a good balance between convergence and diversity of the Pareto optimal solutions.

We studied different families of weighted and unconstrained SFs, providing an analysis of their mathematical properties such as: 1) if an SF is Pareto compliant or weakly Pareto-compliant, 2) if it can generate solutions in convex, concave and linear Pareto fronts, 3) its ability to scale up to any number of objectives, 4) its convergence speed and capability to provide a uniform distribution when tackling different Pareto front shapes. Moreover, we presented the use of these SFs in state-of-the-art MOEAs such as MOEA/D, MOMBI-II and MOEA/D-DRA. We can remark that some SFs proposed in the mathematical programming area and scarcely employed in MOEAs show an interesting behavior to solve MOPs. For example, the Exponential Weighted Criteria (EWC), the Vector Angle Distance Scaling (VADS) and the Dynamic Interactive Decision Analysis Support System (DIDASS) obtained competitive results with respect to the Penalty Boundary Intersection (PBI) function and with respect to a SF based on the Chebyshev model both of which are commonly employed in decomposition based-MOEAs.

On the other hand, this thesis presented a classification of different parameter tuning and parameter control methodologies to configure MOEAs. We contrasted their advantages and disadvantages depending on different cases of applicability, for instance, their adaptation in low- and high-level components of MOEAs.

The main contributions of this thesis are divided into two aspects: 1) the design of a novel methodology of offline parameter tuning for MOEAs and its applicability to configure SFs and, 2) the use of adaptive strategies to combine diverse SFs in the same MOEA. In the following, we describe the main conclusions related to these contributions:

- Our parameter tuning methodology defines three general steps, remarking for each one, some important recommendations as follows:

    1. The first step requires to formulate a correct definition of assumptions or hypotheses related to the correlations between parameter settings and the best performance in MOEAs in order to establish the observations we are interested on. At the same time, we need to define a class of problems to tackle, the space of parameter components to configure and the quality indicator used to assess the MOEA's performance.
    2. The second step defines the form of collecting the data for our observations. Here, the selected tuning tool and its configuration play an important role, because, considering the stochastic nature of MOEAs, we require to establish the minimum

budget for the number of executions of the target MOEAs. In addition, we should consider the robustness measures and nonparametric statistical analysis to obtain confidence results to guarantee that a configuration is better than another one.

3. The third step is related to statistical analysis to obtain the most promising configurations. Here, we can validate our initial assumptions or hypotheses, or otherwise, we should reformulate our analysis in the previous steps. However, the last option is costly, and, therefore, it is recommended to try to extract all obtained knowledge from the statistical results.

- We applied our offline parameter tuning methodology to identify which SFs are more appropriate for solving specific test problems, concluding that the results depend on the type to goal to be reached, i.e., the design of experiments can focus on generalizing or specializing the algorithmic configuration. In the first case, we can observe a certain degree of conflict among the characteristics of different types of MOPs, and therefore, there is no unique SF that performs best in all problems at the same time. On the other hand, the level of specialization of an algorithmic configuration depends on selecting MOPs with similar features, for instance, the Pareto front geometry or the number of objective functions. In such cases, we can obtain appropriate SFs that solve a particular type of MOP.

- We employed our parameter tuning methodology to configure only the Chebyshev SFs. Here, some interesting observations were derived: 1) we found diverse values that had never been used in the literature, 2) in the case of the augmented Chebyshev functions, we identified parameter sensitivity when the α value is directly correlated to the obtained distribution of Pareto optimal solutions.

- Based on the knowledge gathered from the offline parameter tuning, we can establish guidelines to combine several scalarizing functions. Here, the most important conclusions are the following:

  1. The most robust SFs in tackling diverse Pareto front geometries and a different number of objective functions are: Chebyshev SFs, PBI, EWC, VADS, and the Weighted Norm SF. In all these SFs, we should configure their parameter model because there exists a correlation between the MOP's characterization (i.e. the Pareto front geometry or the number of objective functions) and the value of the parameter model.
  2. MOPs with complicated Pareto fronts (i.e., MOPs with mixed or disconnected subregions, and degenerate shapes), require to combine more than one configuration of an SF or even different SFs.
  3. The pool of SFs that can be combined should have the same target directions with the aim of exploring all the search space around of Pareto front.
  4. We claim that the Multi-Objective Multi-Armed Bandit techniques adopted in our proposed approach is a good option to detect appropriate parameters settings in SFs, while requiring a lower computational effort than the use of static parameter tuning strategies (i.e., offline tuning methods). Moreover using this approach, we were able to outperform state-of-the-art MOEAs, mainly in MOPs with complicated Pareto fronts.

7.1 Future work

As part of our future work, we are interested in studying the following topics:

1. To explore other types of SFs regarding taxonomy presented in Figure 3.1. We want to establish guidelines to create new Pareto compliant SFs able to generate optimal solutions in different Pareto front shapes. To obtain theoretical and experimental foundations to speed up convergence towards different regions of the Pareto front.

2. To employ the parameter tuning techniques (in online and offline manner) revised in Chapter 4 to tune various MOEA's components at the same time (for example: the evolutionary operators, the selection mechanism, the termination criterion, or their combination), with the aim of deriving significant knowledge to understand the correlation among parameters.

3. To explore other adaptive techniques based on the multi-armed bandit model or on learning period strategies and to compare them.

4. To study the relationship between the weight vectors and the corresponding SFs to adapt in online manner the distribution of weight vectors and improve the converge and uniformity along the Pareto front.

5. To combine the offline and online parameter tuning methods in the same algorithm. This means that we would like to gather as much knowledge as possible about a MOP and the parameter space in an MOEA and then use it to train online adaptive algorithms. In this sense, we can use the machine learning techniques used in classification problems to optimize the parameters of an MOEA.

Appendices

Appendix A

Contour lines of Scalarizing Functions

This appendix presents plots of contour lines to explain the behavior of the weighted and unconstrained Scalarizing Functions (SFs) studied in this thesis. In general, these SFs minimize some distance metric between a candidate optimal solution and a reference point using a target direction (weight vector), which is a representation of order that defines the relative importance of the objective functions in a Multiobjective Optimization Problem (MOP).The contour lines is a graphical technique for representing the manner of each SF directs the search in a bidimensional MOP.

Table A.1 summarize all SFs studied in this thesis. Moreover, we include the mathematical definition, the range of values for parameter models and the Pareto front geometries (represented by x: convex, c: concave and l: linear) that each SF can be solved.

Figures A.1 and A.1 show the intersection between contour lines and three basic Pareto fronts shapes. In all cases, we used the same target direction $\lambda = (0.65, 0.35)$. The biggest non-filled shapes (circle, triangle, and square) denote the obtained optimal solutions for each of the Pareto-front shapes. We can see that the functions such as WCP, LS, WN, CHE, ACHE, MCHE, CS, GSF, NSF and DIDASS attain incongruent optimal solutions respect to the target directions. It means that these solutions are located in different positions respect to target directions. The WS, WPR and CS functions cannot solve linear and concave Pareto fronts. Even if we vary the weight vector value, all solutions are located in the extremes of the Pareto front. On the other, the ASF, AASF, PBI, IPBI, 2LPBI, QPBI and VADS functions can solve the three Pareto front shapes tested. Furthermore, the optimal solutions are located in the same target direction.

Table A.1: Summary of weighted and unconstrained scalarizing functions.

Acronym	Full Name	Minimize $g(\mathbf{y};\boldsymbol{\lambda}) :=$	Support	Model Parameters	Reference
WCP	weighted compromise programming	$\sum_i (\lambda_i y_i)^p$	x, c, l $\prec$	$p \in (1,\infty)$	[168]
WS	weighted sum	$\sum_i \lambda_i y_i$	x $\prec$	-	[166]
LS	least squares	$\sqrt{\sum_i \lambda_i \lvert y_i\rvert^2}$	x, c, l $\prec$	-	[109, p. 97]
EWC	exponential weighted criteria	$\sum_i \left(e^{p\,\lambda_i} - 1\right) e^{p\,y_i}$	x, c, l $\prec$	$p \in [1,\infty)$	[4]
WPO	weighted power	$\sum_i \lambda_i\,(y_i)^p$	x, c, l $\prec$	$p \in [1,\infty)$	[100]
WPR	weighted product	$\prod_i (y_i)^{\lambda_i}$		-	[150, p. 9]
WN	weighted norm	$\left(\sum_i \lambda_i \lvert y_i\rvert^p\right)^{\frac{1}{p}}$	x, c, l $\prec$	$p \in [1,\infty)$	[168]
CHE	Chebyshev	$\max_i \{\lambda_i \lvert y_i\rvert\}$	x, c, l $\preceq$	-	[85]
ACHE	augmented Chebyshev	$\max_i \{\lambda_i \lvert y_i\rvert\} + \alpha \sum_i \lvert y_i\rvert$	x, c, l $\prec$	$\alpha \in [0.001, 0.01]$	[144]
MCHE	modified Chebyshev	$\max_i \left\{\lambda_i \left(\lvert y_i\rvert + \alpha \sum_i \lvert y_i\rvert\right)\right\}$	x, c, l $\prec$	small $\alpha > 0$	[86]
ASF	achievement scalarizing function	$\max\left\{\frac{y_i}{\lambda_i}\right\}$	x, c, l $\preceq$ ||	-	[162]
AASF	augmented ASF	$\max\left\{\frac{y_i}{\lambda_i}\right\} + \alpha \sum_i \frac{y_i}{\lambda_i}$	x, c, l $\prec$ ||	small $\alpha > 0$	[109, p. 111]
PBI	penalty boundary intersection	$d_1 + \theta d_2$, where $d_1 := \left\lvert \mathbf{y} \cdot \frac{\boldsymbol{\lambda}}{\lVert\boldsymbol{\lambda}\rVert}\right\rvert$ and $d_2 := \left\lVert \mathbf{y} - d_1 \frac{\boldsymbol{\lambda}}{\lVert\boldsymbol{\lambda}\rVert}\right\rVert$	x, c, l ||	$\theta \in (0,\infty)$ suggested $\theta = 5$	[171]
IPBI	inverted PBI	$\theta d_2 - d_1$, where d_1 and d_2 are the same as PBI	x, c, l ||	$\theta \in (0,\infty)$	[136]
2LPBI	two-level PBI	$d_1 + \theta_1 d_2$ if $d_2 \le d^*$, $d_1 + \theta_1 d^* + \theta_2 (d_2 - d^*)$ otherwise, where d_1 and d_2 are the same as PBI	x, c, l ||	$\theta_1 < \theta_2$ suggested $\theta_1 = 0.1$ $\theta_2 = 10$	[76]
QPBI	quadratic PBI	$d_1 + \theta d_2 \frac{d_2}{d^*}$, where d_1 and d_2 are the same as PBI	x, c, l ||	suggested $\theta = 1$	[76]
CS	conic scalarization	$\sum_i \lambda_i y_i + \alpha \sum_i \lvert y_i\rvert$	x	$\alpha \in [0, \lambda_i]$	[89]
VADS	vector angle distance scaling	$\frac{\lVert\mathbf{y}\rVert}{\left(\frac{\boldsymbol{\lambda}}{\lVert\boldsymbol{\lambda}\rVert} \cdot \frac{\mathbf{y}}{\lVert\mathbf{y}\rVert}\right)^p}$	x, c, l ||	$p > 0$ suggested $p = 100$	[67]
GSF	general scalarizing function	$\beta \max_i \{\lambda_i \lvert y_i\rvert\} + \alpha \sum_i \lambda_i \lvert y_i\rvert$	x, c, l	$\beta \ge 0$ $\alpha \ge 0$	[36]
NSF	normalized scalarizing function	$(1-\delta) \max_i \{\lambda_i \lvert y_i\rvert\} + \delta \sum_i \lambda_i \lvert y_i\rvert$	x, c, l	$\delta \in [0,1]$	[36]
DIDASS	dynamic interactive decision analysis and support system	$\max\left\{\beta \max_i \lambda_i \lvert y_i\rvert, \sum_i \lambda_i \lvert y_i\rvert\right\} + \sum_i \gamma_i \lvert y_i\rvert$	x, c, l	$\beta \in [0,1]$ $\gamma_i \in [0,1]$	[54]

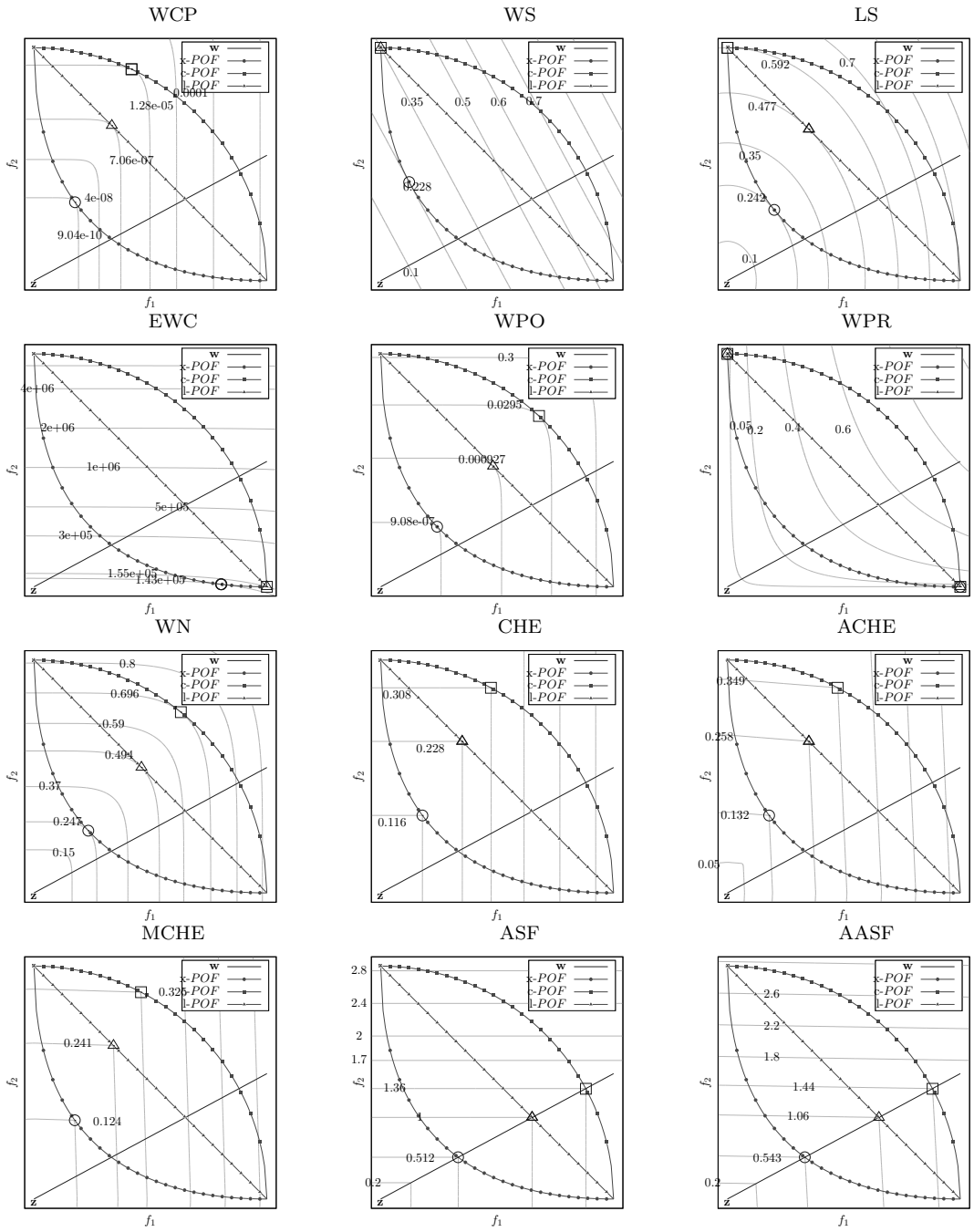


Figure A.1: Contour lines of the scalarizing functions for the weight vector $\lambda = (0.65, 0.35)$.

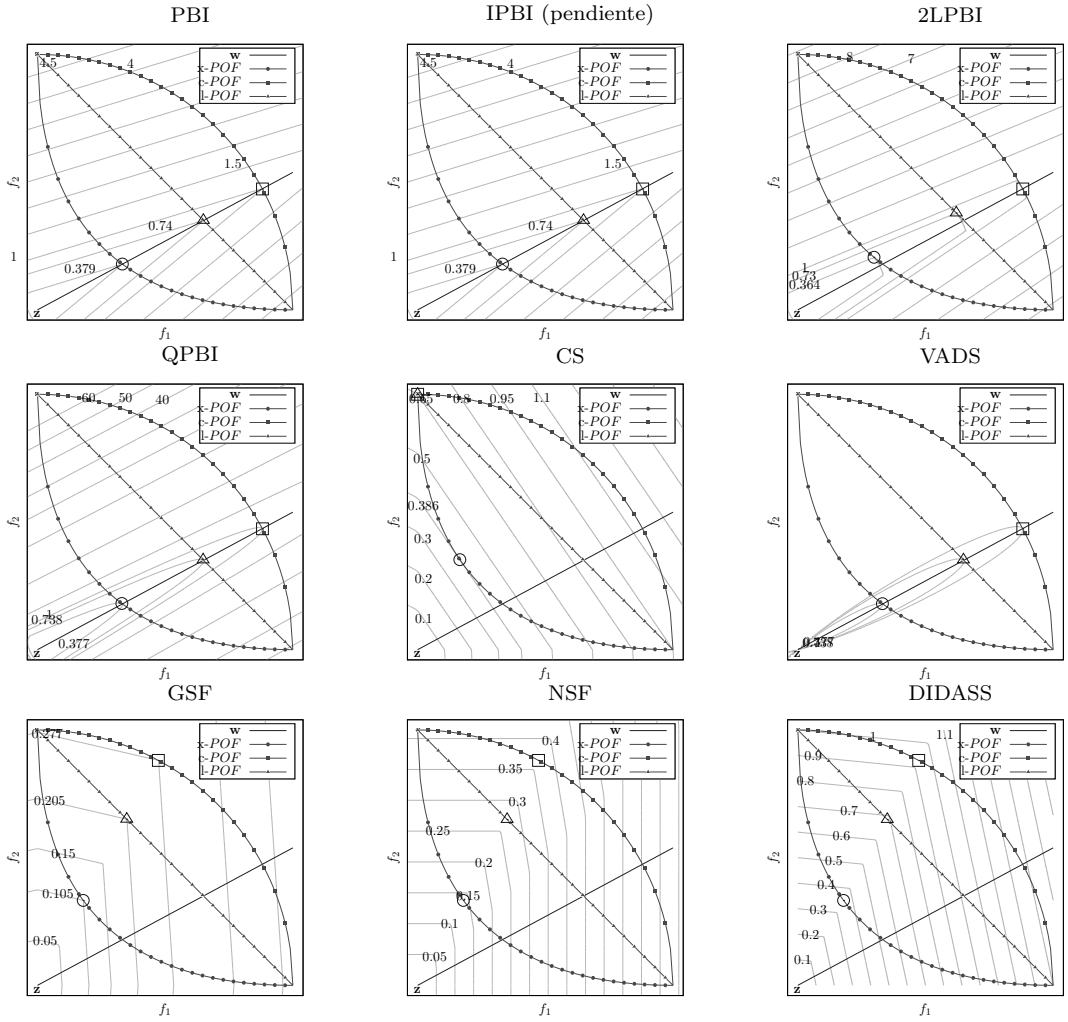


Figure A.2: Contour lines of the scalarizing functions for the weight vector $\lambda = (0.65, 0.35)$ (continuation).

Appendix B

Test Suites Adopted

In this Appendix, we present the test suites of multi-objective optimization problems (MOPs) used in the experiments performed in this thesis. We describe the mathematical definition for real-valued and unconstrained MOPs which are commonly employed in the specialized literature such as the Deb-Thiele-Laumanns-Zitzler (DTLZ) [32], the Walking Fish Group (WFG) [66] and the CEC 2009 Special Session and Competition MOPs (UF) [173]. These test suites are categorized by different properties such as bias, multimodality, non-separability and the Pareto front shapes (convex, concave, disconnected, degenerate and mixed). DTLZ and WFG are scalable in the number of decision variables and objective functions. The UF test suite considers a class of more complex problems with two and three objective functions.

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

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = \big[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\big]^T \\ &\text{subject to } x \in \mathcal{S} \end{aligned} \tag{B.1}$$

where $\mathcal{S}$ is the feasible space of solutions and $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathcal{S}$ is the vector of decision variables. The number of objectives is represented by m.

In MOPs, the aim is to find the set of optimal solutions known as the Pareto optimal set (see Chapter 2). For the DTLZ, WFG and UF test suites, the exact location of the Pareto optimal sets is known.

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

The Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [32] includes seven unconstrained MOPs which are scalable to any number of decision variables and objectives. This test suite include characteristics such as nonseparable problems, deceptive problems, a degenerate problem, a mixed shape Pareto front problem, problems scalable in the number of position related parameters, and problems with dependencies between position- and distance-related parameters. DTLZ considers that the total number of decision variables is given by $n = m + k - 1$, where k is the number of distance parameters. These distance parameters are defined as $\mathbf{y} = [x_m, x_{m+1}, \ldots, x_n]^T$ considering the decision vector $\mathbf{x} = [x_1, \ldots, x_{m-1}, x_m, \ldots, x_n]^T$ In [32], the following k-values are recommended: 5 for DTLZ1, 10 for DTLZ2-6, and 20 for DTLZ7.

The following sections present the mathematical definition of each of the DTLZ test problems.

B.1.1 DTLZ1

This MOP is separable and multimodal and is defined as follows:

$$\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = 0.5(1+g(\mathbf{y}))\prod_{i=1}^{m-1} x_i, \\
&f_{j=2:m-1}(\mathbf{x}) = 0.5(1+g(\mathbf{y}))(1-x_{m-j+1})\prod_{i=1}^{m-j} x_i, \\
&f_m(\mathbf{x}) = 0.5(1+g(\mathbf{y}))(1-x_1). \\
&\text{Where} \\
&g(\mathbf{y}) = 100\left[k+\sum_{i=1}^{k}(y_i-0.5)^2-\cos(20\pi(y_i-05))\right].
\end{aligned} \tag{B.2}$$

Its Pareto optimal front is linear and all objective function values lie on the linear hyperplane $\sum_{i=1}^{m} f_i = 0.5$.

B.1.2 DTLZ2

This problem is separable and unimodal and is defined as follows:

$$\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = (1+g(\mathbf{y}))\prod_{i=1}^{m-1}\cos\left(\frac{x_i\pi}{2}\right), \\
&f_{j=2:m-1}(\mathbf{x}) = (1+g(\mathbf{y}))\left(\prod_{i=1}^{m-j}\cos\left(\frac{x_i\pi}{2}\right)\right)\sin\left(\frac{x_{m-j+1}\pi}{2}\right), \\
&f_m(\mathbf{x}) = (1+g(\mathbf{y}))\sin\left(\frac{x_1\pi}{2}\right). \\
&\text{Where} \\
&g(\mathbf{y}) = \sum_{i=1}^{k}(y_i-0.5)^2.
\end{aligned} \tag{B.3}$$

The Pareto optimal solutions are produced when $\mathbf{y} = (0.5, 0.5, \dots)^T$ and all objective functions values must satisfy that $\sum_{i=1}^{m}(f_i)^2 = 1$. Its Pareto front shape is concave.

B.1.3 DTLZ3

This problem is similar to DTLZ2 but includes multifrontal difficulty. Its definition is given as follows:

$$\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = (1 + g(\mathbf{y})) \prod_{i=1}^{m-1} \cos\left(\frac{x_i\pi}{2}\right) \\
&f_{j=2:m-1}(\mathbf{x}) = (1 + g(\mathbf{y})) \left(\prod_{i=1}^{m-j} \cos\left(\frac{x_i\pi}{2}\right)\right) \sin\left(\frac{x_{m-j+1}\pi}{2}\right) \\
&f_m(\mathbf{x}) = (1 + g(\mathbf{y})) \sin\left(\frac{x_1\pi}{2}\right) \\
&\text{Where} \\
&g(\mathbf{y}) = 100 \left[k + \sum_{i=1}^{k} (y_i - 0.5)^2 - \cos(20\pi(y_i - 05))\right]
\end{aligned} \tag{B.4}$$

The Pareto optimal front corresponds to $\mathbf{y} = (0.5, 0.5, \dots)^T$.

B.1.4 DTLZ4

This problem is concave, separable and unimodal, and it is defined as follows:

Minimize

$$\begin{aligned}
f_1(\mathbf{x}) &= (1 + g(\mathbf{y})) \prod_{i=1}^{m-1} \cos\left(\frac{x_i^\alpha\pi}{2}\right) \\
f_{j=2:m-1}(\mathbf{x}) &= (1 + g(\mathbf{y})) \left(\prod_{i=1}^{m-j} \cos\left(\frac{x_i^\alpha\pi}{2}\right)\right) \sin\left(\frac{x_{m-j+1}^\alpha\pi}{2}\right) \\
f_m(\mathbf{x}) &= (1 + g(\mathbf{y})) \sin\left(\frac{x_1^\alpha\pi}{2}\right)
\end{aligned}$$

Where

$$g(\mathbf{y}) = 100 \left[k + \sum_{i=1}^{k} (y_i - 0.5)^2\right] \tag{B.5}$$

The parameters $\alpha = 100$ is suggested by its authors.

B.1.5 DTLZ5

This problem is unimodal and degenerated and its mathematical definition is:

Minimize

$$
\begin{aligned}
f_1(\mathbf{x}) &= (1+g(\mathbf{y}))\prod_{i=1}^{m-1}\cos\left(\frac{\theta_i\pi}{2}\right)\\
f_{j=2:m-1}(\mathbf{x}) &= (1+g(\mathbf{y}))\left(\prod_{i=1}^{m-j}\cos\left(\frac{\theta_i\pi}{2}\right)\right)\sin\left(\frac{\theta_{m-j+1}\pi}{2}\right)\\
f_m(\mathbf{x}) &= (1+g(\mathbf{y}))\sin\left(\frac{\theta_1\pi}{2}\right)
\end{aligned}
$$

Where

$$
\begin{aligned}
\theta_i &= \begin{cases} x_i & \text{if } i=1 \\ \frac{1+2g(\mathbf{y})}{2(1+g(\mathbf{y}))}x_i & \forall i\in\{2,3,\ldots,m-1\}\end{cases}\\
g(\mathbf{y}) &= \sum_{i=1}^{k}(y_i-0.5)^2
\end{aligned}
\tag{B.6}
$$

The Pareto optimal front corresponds to $\mathbf{y}=(0.5,\ldots,0.5)^T$.

B.1.6 DTLZ6

This problem is unimodal and degenerated and is a modified version of DTLZ5. Its mathematical definition is:

Minimize

$$\begin{aligned} f_1(\mathbf{x}) &= (1+g(\mathbf{y}))\prod_{i=1}^{m-1}\cos\left(\frac{\theta_i\pi}{2}\right) \\ f_{j=2:m-1}(\mathbf{x}) &= (1+g(\mathbf{y}))\left(\prod_{i=1}^{m-j}\cos\left(\frac{\theta_i\pi}{2}\right)\right)\sin\left(\frac{\theta_{m-j+1}\pi}{2}\right) \\ f_m(\mathbf{x}) &= (1+g(\mathbf{y}))\sin\left(\frac{\theta_1\pi}{2}\right) \end{aligned}$$

Where

$$\begin{aligned} \theta_i &= \begin{cases} x_i & \text{if } i=1 \\ \frac{1+2g(\mathbf{y})}{2(1+g(\mathbf{y}))}x_i & \forall i\in\{2,3,\ldots,m-1\} \end{cases} \\ g(\mathbf{y}) &= \sum_{i=1}^{k} y_i^{0.1} \end{aligned} \tag{B.7}$$

The Pareto optimal front is located at $\mathbf{y}=(0,\ldots,0)^T$.

B.1.7 DTLZ7

This problem has a disconnected Pareto front shape and is defined as follows:

Minimize

$$\begin{aligned} f_{j=1:m-1}(\mathbf{x}) &= x_j \\ f_m(\mathbf{x}) &= (1+g(\mathbf{y}))\left(m-\sum_{i=1}^{m-1}\left(\frac{f_i}{1+g(\mathbf{y})}(1+\sin(3\pi f_i))\right)\right) \end{aligned}$$

Where

$$g(\mathbf{y}) = 1+\frac{9}{k}\sum_{i=1}^{k} y_i \tag{B.8}$$

The Pareto optimal solutions correspond to $\mathbf{y}=(0,\ldots,0)^T$.

B.2 The Walking Fish Group (WFG) Test Suite (WFG)

Huband et al. [66] proposed the Walking-Fish-Group (WFG) test suite. This test suite suggests nine multi-objective test problems scalable with respect to the number of variables and objectives. The number of decision variables (x_i) is given by $n = k + l$, where $k \in \{m-1, 2(m-1), 3(m-1), \dots\}$ is the position related-parameter, l is the distance-related parameter and it should be satisfied that $n \leq m$. All $x_i \in \mathbf{x}$ have the domain $[0, 1]$. The vector $\mathbf{x}$ is derived from a vector of parameters $\mathbf{z} \in \mathbb{R}^n$. The domain of all $z_i \in \mathbf{z}$ is $[0, 2i]$. These problems include diverse Pareto front shapes and characteristics such as bias, multi-modality, and non-separability which vary their degree of difficulty via a set of transformation functions, which map parameters with domain $[0, 1]$ onto the range $[0, 2i]$.

In the following, we describe this test suite.

B.2.1 WFG1

WFG1 is separable and unimodal, but it has a polynomial and flat region. It is strongly biased toward small values of the variables. Its definition is given as:

$$\begin{aligned} &\text{Given} \\ &\quad \mathbf{z} = \{z_1, \dots, z_k, z_{k+1}, \dots, z_n\} \end{aligned} \tag{B.9}$$

Minimize

$$\begin{aligned} f_1(\mathbf{x}) &= x_m + 2 \prod_{i=1}^{m-1} \left(1 - \cos\left(\frac{x_i \pi}{2}\right)\right) \\ f_{j=2:m-1}(\mathbf{x}) &= x_m + 2j \left(\prod_{i=1}^{m-j} \left(1 - \cos\left(\frac{x_i \pi}{2}\right)\right)\right) \left(1 - \sin\left(\frac{x_{m-j+1} \pi}{2}\right)\right) \\ f_m(\mathbf{x}) &= x_m + 2m \left(1 - x_1 - \frac{\cos\left(\frac{10\pi x_1}{2}\right)}{10\pi}\right) \end{aligned}$$

Where

$$\begin{aligned} x_{i=1:m-1} &= \text{r\_sum}\left(\{y_{(i-1)k/(m-1)+1}, \dots, y_{ik/(m-1)}\}, \left\{\frac{2(i-1)k}{(m-1)} + 1, \dots, \frac{2ik}{(m-1)}\right\}\right) \\ x_m &= \text{r\_sum}\left(\{y_{k+1}, \dots, y_n\}, \{2(k+1), \dots, 2n\}\right) \\ y_{i=1:n} &= \text{b\_poly}(y'_i, 0.02) \\ y'_{i=1:k} &= y''_i \\ y'_{i=k+1:n} &= \text{b\_flat}(y''_i, 0.8, 0.75, 0.85) \\ y''_{i=1:k} &= \frac{z_i}{2i} \\ y''_{i=k+1:n} &= \text{s\_linear}\left(\frac{z_i}{2i}, 0.35\right) \end{aligned} \tag{B.10}$$

B.2.2 WFG2

This problem is non-separable and multimodal and its Pareto optimal front shape is disconnected. It is defined as follows:

Minimize

$$\begin{aligned}
f_1(\mathbf{x}) &= x_m + 2\prod_{i=1}^{m-1}\left(1-\cos\left(\frac{x_i\pi}{2}\right)\right)\\
f_{j=2:m-1}(\mathbf{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}\left(1-\cos\left(\frac{x_i\pi}{2}\right)\right)\right)\left(1-\sin\left(\frac{x_{m-j+1}\pi}{2}\right)\right)\\
f_m(\mathbf{x}) &= x_m + 2m\left(1-x_1\cos^2(5x_1\pi)\right)
\end{aligned}$$

Where

$$\begin{aligned}
x_{i=1:m-1} &= \text{r_sum}\left(\{y_{(i-1)k/(m-1)+1},\ldots,y_{ik/(m-1)}\},\{1,\ldots,1\}\right)\\
x_m &= \text{r_sum}\left(\{y_{k+1},\ldots,y_{k+l/2}\},\{1,\ldots,1\}\right)\\
y'_{i=1:k} &= y'_i\\
y'_{i=k+1:k+l/2} &= \text{r_nonsep}(\{y'_{k+2(i-k)-1},y'_{k+2(i-k)}\},2)\\
y'_{i=1:k} &= \frac{z_i}{2i}\\
y'_{i=k+1:n} &= \text{s_linear}\left(\frac{z_i}{2i},0.35\right)
\end{aligned} \tag{B.11}$$

B.2.3 WFG3

This problem is non-separable but unimodal with a linear and degenerate Pareto front shape. It is given by the following expression:

$$
\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = x_m + 2\prod_{i=1}^{m-1}(x_i) \\
&f_{j=2:m-1}(\mathbf{x}) = x_m + 2j\left(\prod_{i=1}^{m-j} x_i\right)(1 - x_{m-j+1}) \\
&f_m(\mathbf{x}) = x_m + 2m(1 - x_1) \\
&\text{Where } x_{i=1} = u_i \\
&x_{i=2:m-1} = x_m(u_i - 0.5) + 0.5 \\
&x_m = \text{r\_sum}\left(\{y_{k+1}, \ldots, y_{k+l/2}\}, \{1, \ldots, 1\}\right) \\
&u_i = \text{r\_sum}\left(\{y_{(i-1)k/(m-1)+1}, \ldots, y_{ik/(m-1)}\}, \{1, \ldots, 1\}\right) \\
&y'_{i=1:k} = y'_i \\
&y'_{i=k+1:k+l/2} = \text{r\_nonsep}(\{y'_{k+2(i-k)-1}, y'_{k+2(i-k)}\}, 2) \\
&y'_{i=1:k} = \frac{z_i}{2i} \\
&y'_{i=k+1:n} = \text{s\_linear}\left(\frac{z_i}{2i}, 0.35\right)
\end{aligned}
\tag{B.12}
$$

B.2.4 WFG4

WFG4 is separable, but highly multimodal. Its Pareto front shape is concave. It is defined as follows:

$$
\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = x_m + 2\prod_{i=1}^{m-1}\sin\left(x_i\pi/2\right) \\
&f_{j=2:m-1}(\mathbf{x}) = x_m + 2j\left(\prod_{i=1}^{m-j}\sin(x_i\pi/2)\right)\cos(x_{m-j+1}\pi/2) \\
&f_m(\mathbf{x}) = x_m + 2m\cos(x_1\pi/2) \\
&\text{Where} \\
&x_{i=1:m-1} = \text{r\_sum}\left(\{y_{k+1}, \ldots, y_{ik/(m-1)}\}, \{1, \ldots, 1\}\right) \\
&x_m = \text{r\_sum}\left(\{y_{k+1}, \ldots, y_n\}, \{1, \ldots, 1\}\right) \\
&y_{i=1:n} = \text{s\_multi}\left(\frac{z_i}{2i}, 30, 10, 0.35\right)
\end{aligned}
\tag{B.13}
$$

B.2.5 WFG5

This problem is deceptive, separable, and its Pareto front shape is concave. It is defined as follows:

$$\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = x_m + 2\prod_{i=1}^{m-1}\sin\left(x_i\pi/2\right) \\
&f_{j=2:m-1}(\mathbf{x}) = x_m + 2j\left(\prod_{i=1}^{m-j}\sin(x_i\pi/2)\right)\cos(x_{m-j+1}\pi/2) \\
&f_m(\mathbf{x}) = x_m + 2m\cos(x_1\pi/2) \\
&\text{Where} \\
&x_{i=1:m-1} = \text{r_sum}\left(\{y_{k+1},\ldots,y_{ik/(m-1)}\},\{1,\ldots,1\}\right) \\
&x_m = \text{r_sum}\left(\{y_{k+1},\ldots,y_n\},\{1,\ldots,1\}\right) \\
&y_{i=1:n} = \text{s_decept}\left(\frac{z_i}{2i},0.35,0.001,0.05\right)
\end{aligned} \tag{B.14}$$

B.2.6 WFG6

This problem is deceptive, separable, and has a concave Pareto front shape. It is defined by the following expression:

$$\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = x_m + 2\prod_{i=1}^{m-1}\sin\left(x_i\pi/2\right) \\
&f_{j=2:m-1}(\mathbf{x}) = x_m + 2j\left(\prod_{i=1}^{m-j}\sin(x_i\pi/2)\right)\cos(x_{m-j+1}\pi/2) \\
&f_m(\mathbf{x}) = x_m + 2m\cos(x_1\pi/2) \\
&\text{Where} \\
&x_{i=1:m-1} = \text{r_nonsep}\left(\{y_{(i-1)k/(m-1)+1},\ldots,y_{ik/(m-1)}\},\{1,\ldots,1\}\right) \\
&x_m = \text{r_nonsep}\left(\{y_{k+1},\ldots,y_n\},l\right) \\
&y_{i=1:k} = \frac{z_i}{2i} \\
&y_{i=k+1:n} = \text{s_decept}\left(\frac{z_i}{2i},0.35\right)
\end{aligned} \tag{B.15}$$

B.2.7 WFG7

This problem is separable and unimodal, with a concave Pareto optimal front. It is defined as follows:

$$
\begin{aligned}
&\text{Minimize} \\
f_1(\mathbf{x}) &= x_m + 2\prod_{i=1}^{m-1}\sin\left(x_i\pi/2\right) \\
f_{j=2:m-1}(\mathbf{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}\sin(x_i\pi/2)\right)\cos(x_{m-j+1}\pi/2) \\
f_m(\mathbf{x}) &= x_m + 2m\cos(x_1\pi/2) \\
&\text{Where} \\
x_{i=1:m-1} &= \text{r\_sum}\left(\{y_{(i-1)k/(m-1)+1},\ldots,y_{ik/(m-1)}\},\{1,\ldots,1\}\right) \\
x_m &= \text{r\_sum}\left(\{y_{k+1},\ldots,y_n\},\{1,\ldots,1\}\right) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:n} &= \text{s\_linear}(y'_i, 0.35) \\
y'_{i=1:k} &= \text{b\_param}(z_i/(2i), \text{r\_sum}(\{z_{i+1}/(2(i+1)),\ldots,z_n/2n\}, \\
&\qquad \{1,\ldots,1\}), \frac{0.98}{49.98}, 0.02, 50) \\
y_{i=k+1:n} &= \frac{z_i}{2i}
\end{aligned}
\tag{B.16}
$$

B.2.8 WFG8

This problem has a parameter-dependent bias, and is non-separable and unimodal. Its Pareto front shape is concave. It is defined as follows:

$$
\begin{aligned}
&\text{Minimize} \\
f_1(\mathbf{x}) &= x_m + 2\prod_{i=1}^{m-1}\sin\left(x_i\pi/2\right) \\
f_{j=2:m-1}(\mathbf{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}\sin(x_i\pi/2)\right)\cos(x_{m-j+1}\pi/2) \\
f_m(\mathbf{x}) &= x_m + 2m\cos(x_1\pi/2) \\
&\text{Where} \\
x_{i=1:m-1} &= \text{r\_sum}\left(\{y_{(i-1)k/(m-1)+1},\ldots,y_{ik/(m-1)}\},\{1,\ldots,1\}\right) \\
x_m &= \text{r\_sum}\left(\{y_{k+1},\ldots,y_n\},\{1,\ldots,1\}\right) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:n} &= \text{s\_linear}(y'_i, 0.35) \\
y_{i=1:k} &= \frac{z_i}{2i} \\
y'_{i=k+1:n} &= \text{b\_param}(z_i/(2i), \text{r\_sum}(\{z_1/2,\ldots,z_{i-1}/(2(i-1))\}, \\
&\qquad \{1,\ldots,1\}), \frac{0.98}{49.98}, 0.02, 50)
\end{aligned}
\tag{B.17}
$$

B.2.9 WFG9

WFG9 is non-separable, multimodal, deceptive, and has a parameter-dependent bias. Its Pareto front shape is concave, but in the edge of the shape is flat. This problem is defined as:

$$
\begin{aligned}
&\text{Minimize} \\
&f_1(\mathbf{x}) = x_m + 2\prod_{i=1}^{m-1}\sin\left(x_i\pi/2\right) \\
&f_{j=2:m-1}(\mathbf{x}) = x_m + 2j\left(\prod_{i=1}^{m-j}\sin(x_i\pi/2)\right)\cos(x_{m-j+1}\pi/2) \\
&f_m(\mathbf{x}) = x_m + 2m\cos(x_1\pi/2) \\
&\text{Where} \\
&x_{i=1:m-1} = \text{r\_nonsep}\left(\{y_{(i-1)k/(m-1)+1},\ldots,y_{ik/(m-1)}\}, k/(m-1)\right) \\
&x_m = \text{r\_nonsep}\left(\{y_{k+1},\ldots,y_n\}, l\right) \\
&y_{i=1:k} = \text{s\_decept}(y'_i, 0.35, 0.001, 0.05) \\
&y_{i=k+1:n} = \text{s\_multi}(y'_i, 30, 95, 0.35) \\
&y'_{i=1:n-1} = \text{b\_param}(z_i/(2i), \text{r\_sum}(\{z_{i+1}/(2(i-1),\ldots,z_n/2n)\}, \\
&\qquad \{1,\ldots,1\}), \frac{0.98}{49.98}, 0.02, 50) \\
&y'_n = \frac{z_n}{2n}
\end{aligned}
\tag{B.18}
$$

B.3 The CEC 2009 Special Session and Competition Test Suite

The problems defined at the special session on multi-objective optimization held at the 2001 IEEE Congress on Evolutionary Computation (CEC'2009) [173] includes 10 MOPs with two and three objective functions. In general, these problems have similar difficulties related to real-world problems. Moreover, the set of Pareto solutions in the decision variable space presents a complicated distribution.

B.3.1 UF1

This problem has two objectives and a convex Pareto front shape. It is given by the following expression:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}[x_j - sin(6\pi x_1 + \frac{j\pi}{n})]^2 \\ f_2 &= 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}[x_j - sin(6\pi x_1 + \frac{j\pi}{n})]^2 \end{aligned} \tag{B.19}$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$ The search space is $[0,1] \times [-1,1]^{n-1}$

B.3.2 UF2

This problem has two objectives and a convex Pareto front shape. It is defined as follows:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2 \\ f_2 &= 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2 \end{aligned} \tag{B.20}$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$ and

$$y_j = \begin{cases} x_j - [0.3x_1^2 cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]cos(6\pi x_1 + \frac{j\pi}{n}) j \in J_1, \\ x_j - [0.3x_1^2 cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]sin(6\pi x_1 + \frac{j\pi}{n}) j \in J_2 \end{cases}$$

The search space is $[0,1] \times [-1,1]^{n-1}$

B.3.3 UF3

This problem has two objectives and a convex Pareto front shape. It is given by the following expression:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1} cos(\frac{20y_j\pi}{\sqrt{j}}) + 2) \\ f_2 &= 1 - \sqrt{x_1} + \frac{2}{|J_2|}(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2} cos(\frac{20y_j\pi}{\sqrt{j}}) + 2) \end{aligned} \tag{B.21}$$

where J_1 and J_2 are the same of those of $F1$, and $y_j = x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}$, $j = 2, \ldots, n$. The search space is $[0,1]^n$

B.3.4 UF4

This problem has two objectives and concave Pareto front shape. It is defined as follows:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}\sum_{j\in J_1} h(y_j) \\ f_2 &= 1 - x_1^2 + \frac{2}{|J_2|}\sum_{j\in J_2} h(y_j) \end{aligned} \tag{B.22}$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$, $y_j = x_j - sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, \ldots, n$ and $h(t) = \frac{|t|}{1+e^{2|t|}}$. Its search space $[0,1] \times [-2,2]^{n-1}$

B.3.5 UF5

This problem has two objectives with a linear and disconnected Pareto front shape. It is given by the following expression:

$$\begin{aligned} f_1 &= x_1 + (\frac{1}{2N} + \epsilon)|sin(2N\pi x_1)| + \frac{2}{|J_1|}\sum_{j\in J_1} h(y_j) \\ f_2 &= 1 - x_1 + (\frac{1}{2N} + \epsilon)|sin(2N\pi x_1)| + \frac{2}{|J_2|}\sum_{j\in J_2} h(y_j) \end{aligned} \tag{B.23}$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$. N is an integer, $\epsilon > 0$, $y_j = x_j - sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, \ldots, n$ and $h(t) = 2t^2 cos(4\pi t) + 1$. The search space $[0,1] \times [-1,1]^{n-1}$

B.3.6 UF6

This problem has two objectives with a linear and disconnected Pareto front shape. It is defined as follows:

$$\begin{aligned} f_1 &= x_1 + \max\{0, 2(\frac{1}{2N} + \epsilon)\sin(2N\pi x_1)\} + \frac{2}{|J_1|}(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2) \\ f_2 &= 1 - x_1 + \max\{0, 2(\frac{1}{2N} + \epsilon)\sin(2N\pi x_1)\} + \frac{2}{|J_2|}(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2) \end{aligned} \tag{B.24}$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$, and $y_j = x_j - sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, \ldots, n$. The search space $[0,1] \times [-1,1]^{n-1}$

B.3.7 UF7

This problem has two objectives with a linear Pareto front shape. It is given by the following expression:

$$\begin{aligned} f_1 &= \sqrt[5]{x_1} + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2 \\ f_2 &= 1 - \sqrt[5]{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2 \end{aligned} \tag{B.25}$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$, and $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, \ldots, n$. The search space is $[0,1] \times [-1,1]^{n-1}$.

B.3.8 UF8

This problem has three objectives and a concave Pareto front shape. It is given by the following expression:

$$\begin{aligned} f_1 &= \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2 \\ f_2 &= \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_2}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2 \\ f_3 &= \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2 \end{aligned} \tag{B.26}$$

where
$J_1 = \{j|3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$,
$J_2 = \{j|3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$,
$J_3 = \{j|3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}$. The search space is $[0,1]^2 \times [-2,2]^{n-2}$.

B.3.9 UF9

This problem has three objectives with a linear and disconnected Pareto front shape. It is defined as follows:

$$
\begin{aligned}
f_1 &= 0.5[\max\{0,(1+\epsilon)(1-4(2x_1-1)^2)\}+2x_1]x_2+\frac{2}{|J_1|}\sum_{j\in J_1}(x_j-2x_2\sin(2\pi x_1+\frac{j\pi}{n}))^2\\
f_2 &= 0.5[\max\{0,(1+\epsilon)(1-4(2x_1-1)^2)\}-2x_1+2]x_2+\frac{2}{|J_2|}\sum_{j\in J_2}(x_j-2x_2\sin(2\pi x_1+\frac{j\pi}{n}))^2\\
f_3 &= 1-x_2+\frac{2}{|J_3|}\sum_{j\in J_3}(x_j-2x_2\sin(2\pi x_1+\frac{j\pi}{n}))^2
\end{aligned}
\tag{B.27}
$$

where
$J_1=\{j|3\le j\le n$, and $j-1$ is a multiplication of $3\}$,
$J_2=\{j|3\le j\le n$, and $j-2$ is a multiplication of $3\}$,
$J_3=\{j|3\le j\le n$, and j is a multiplication of $3\}$ and $\epsilon=0.1$, (ϵ can take any other positive values). The search space is $[0,1]^2\times[-2,2]^{n-2}$.

B.3.10 UF10

This problem has three objectives and a concave Pareto front shape. It is given by the following expression:

$$
\begin{aligned}
f_1 &= \cos(0.5x_1\pi)\cos(0.5x_2\pi)+\frac{2}{|J_1|}\sum_{j\in J_1}[4y_j^2-\cos(8\pi y_j)+1]\\
f_2 &= \cos(0.5x_1\pi)\sin(0.5x_2\pi)+\frac{2}{|J_2|}\sum_{j\in J_2}[4y_j^2-\cos(8\pi y_j)+1]\\
f_3 &= \sin(0.5x_1\pi)+\frac{2}{|J_3|}\sum_{j\in J_3}[4y_j^2-\cos(8\pi y_j)+1]
\end{aligned}
\tag{B.28}
$$

where
$J_1=\{j|3\le j\le n$, and $j-1$ is a multiplication of $3\}$,
$J_2=\{j|3\le j\le n$, and $j-2$ is a multiplication of $3\}$,
$J_3=\{j|3\le j\le n$, and j is a multiplication of $3\}$, and $y_j=x_j-2x_2\sin(2\pi x_1+\frac{j\pi}{n})$, $j=3,\ldots,n$. The search space is $[0,1]^2\times[-2,2]^{n-2}$.

Appendix C

Additional results of Offline Parameter Tuning experiments

In this Appendix, we present figures and tables related to Chapter 5 for the case study regarding the effect of convergence and distribution of several scalarizing functions based on the Augmented Chebyshev model.

Tables C.1, C.2 and C.3 present the mean hypervolume indicator values of 30 independent runs on $DTLZ1$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives. Whereas tables C.4, C.5 and C.6 show the inverted generational distance for the same experiment.

Tables C.7, C.8, C.9 and C.10 show the mean and standard deviation corresponding to the EVOCA's recommendations for 2, 3, 4, 6, 8 and 10 objective functions.

Finally, Figures C.1, C.2 and C.3 show the box plots for the different scenarios using the hypervolume indicator without applying a normalization process. Marks a, b, c, d, e, f correspond to each robustness measure: mean-mean, median-median, best-worst, mean-worst, median-worst and worst-worst.

Table C.1: Mean HV indicator values (the standard deviation is shown in parentheses) of 30 independent runs on $DTLZ1$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives.

Function, m	α	ACHE	RACHE	MCHE	RMCHE	AASF
$DTLZ1$, 3	0.0001	2.6968e+01(3.01e-02)	2.6943e+01(3.32e-02)	2.6974e+01(2.72e-04)	**2.6974e+01(5.33e-05)**	2.6974e+01(6.45e-05)
	0.01	2.6979e+01(2.44e-02)	2.6979e+01(1.45e-02)	2.6974e+01(5.45e-05)	2.6946e+01(3.91e-02)	**2.6974e+01(4.26e-05)**
	0.1	2.6983e+01(2.12e-02)	**2.6997e+01(8.52e-03)**	2.6974e+01(1.01e-04)	2.6973e+01(3.82e-03)	2.6973e+01(1.95e-03)
	1.0	2.6912e+01(2.84e-02)	2.6975e+01(1.90e-02)	2.6974e+01(4.31e-05)	2.6959e+01(1.02e-02)	2.6962e+01(5.84e-04)
	5.0	**2.6994e+01(1.22e-02)**	2.6959e+01(6.94e-02)	2.6974e+01(1.20e-04)	2.6927e+01(2.90e-02)	2.6948e+01(9.82e-04)
	10.0	2.6977e+01(1.49e-02)	2.6978e+01(4.98e-02)	2.6974e+01(1.12e-03)	2.6934e+01(5.50e-02)	2.6948e+01(1.73e-02)
$DTLZ1$, 5	0.0001	2.4298e+02(1.52e-02)	2.4298e+02(1.18e-02)	2.4299e+02(4.95e-04)	**2.4299e+02(1.53e-03)**	2.4299e+02(9.33e-05)
	0.01	2.4295e+02(5.09e-02)	2.4299e+02(4.90e-03)	2.4299e+02(1.31e-03)	2.4298e+02(1.31e-02)	2.4299e+02(8.46e-04)
	0.1	2.4299e+02(5.54e-03)	**2.4299e+02(4.31e-03)**	2.4299e+02(1.34e-03)	2.4298e+02(1.59e-02)	2.4299e+02(1.35e-04)
	1.0	2.4298e+02(1.95e-02)	2.4281e+02(2.16e-01)	2.4299e+02(1.35e-03)	2.4295e+02(5.64e-02)	2.4299e+02(7.98e-04)
	5.0	**2.4299e+02(5.50e-03)**	2.4295e+02(1.20e-01)	2.4299e+02(1.36e-03)	2.4294e+02(8.89e-02)	2.4299e+02(1.89e-03)
	10.0	2.4299e+02(1.26e-02)	2.4290e+02(1.67e-01)	2.4299e+02(6.62e-04)	2.4291e+02(1.23e-01)	2.4299e+02(1.90e-03)
$DTLZ1$, 7	0.0001	**2.1869e+03(4.03e-02)**	2.1863e+03(6.32e-01)	2.1869e+03(2.13e-03)	**2.1870e+03(4.00e-04)**	2.1869e+03(9.89e-04)
	0.01	2.1866e+03(4.57e-01)	**2.1869e+03(6.06e-02)**	2.1869e+03(1.44e-03)	2.1869e+03(2.15e-02)	2.1870e+03(1.26e-03)
	0.1	2.1868e+03(3.16e-01)	2.1869e+03(7.13e-02)	**2.1870e+03(5.25e-04)**	2.1869e+03(1.63e-02)	2.1870e+03(3.39e-04)
	1.0	2.1866e+03(5.43e-01)	2.1863e+03(9.34e-01)	2.1869e+03(2.14e-03)	2.1869e+03(3.76e-02)	2.1870e+03(4.26e-04)
	5.0	2.1858e+03(1.00e+00)	2.1866e+03(1.06e+00)	2.1869e+03(2.22e-03)	2.1869e+03(1.48e-01)	**2.1870e+03(2.49e-04)**
	10.0	2.18e+03(1.1285e+00)	2.1865e+03(9.95e-01)	2.1869e+03(1.63e-03)	2.1869e+03(1.48e-01)	2.1870e+03(4.53e-04)
$DTLZ1$, 10	0.0001	**5.9020e+04(1.59e+01)**	**5.9015e+04(1.32e+01)**	5.9048e+04(8.64e-02)	**5.9048e+04(1.62e-02)**	5.9048e+04(1.50e-02)
	0.01	5.9018e+04(1.61e+01)	5.9013e+04(2.83e+01)	**5.9049e+04(3.39e-03)**	5.9048e+04(1.43e-01)	**5.9049e+04(0.00e+00)**
	0.1	5.9016e+04(2.17e+01)	5.9014e+04(2.46e+01)	5.9048e+04(1.52e-02)	5.9048e+04(5.37e-01)	**5.9049e+04(0.00e+00)**
	1.0	5.9015e+04(3.03e+01)	5.8997e+04(4.63e+01)	5.9048e+04(1.22e-02)	5.9048e+04(8.45e-01)	**5.9049e+04(0.00e+00)**
	5.0	5.9009e+04(4.43e+01)	5.9000e+04(4.16e+01)	5.9049e+04(1.02e-02)	5.9048e+04(8.00e-01)	**5.90e+04(0.0000e+00)**
	10.0	5.8995e+04(4.34e+01)	5.8988e+04(6.35e+01)	5.9048e+04(2.54e-02)	5.9047e+04(1.77e+00)	**5.9049e+04(0.00e+00)**

Table C.2: Mean HV indicator values of 30 independent runs on $DTLZ1$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives (continued).

Function, m	α	ACHE	RACHE	MCHE	RMCHE	AASF
$DTLZ3$, 3	0.0001	2.6827e+01(2.86e-01)	2.6225e+01(8.45e-01)	2.6067e+01(1.04e+00)	2.6022e+01(1.76e+00)	2.5704e+01(1.52e+00)
	0.01	**2.6898e+01(2.26e-01)**	**2.6774e+01(2.96e-01)**	**2.6305e+01(5.28e-01)**	2.6149e+01(8.10e-01)	**2.6273e+01(7.23e-01)**
	0.1	2.6572e+01(4.90e-01)	2.6612e+01(8.92e-01)	2.5665e+01(2.11e+00)	**2.6150e+01(9.35e-01)**	2.6031e+01(1.18e+00)
	1.0	2.6022e+01(1.19e+00)	2.6226e+01(7.64e-01)	2.5965e+01(1.35e+00)	2.5773e+01(6.76e-01)	2.5622e+01(1.79e+00)
	5.0	1.8082e+01(2.25e+00)	1.7080e+01(2.69e+00)	2.5217e+01(2.11e+00)	1.8007e+01(3.01e+00)	2.5485e+01(1.75e+00)
	10.0	1.7983e+01(9.85e-03)	1.7990e+01(2.30e+00)	2.5825e+01(1.29e+00)	1.8184e+01(1.07e+00)	2.5621e+01(1.81e+00)
$DTLZ3$, 5	0.0001	2.4163e+02(5.51e+00)	2.4277e+02(4.02e-01)	2.4033e+02(6.52e+00)	**2.4249e+02(3.47e-01)**	2.4247e+02(5.00e-01)
	0.01	2.4202e+02(6.13e-01)	**2.4290e+02(2.74e-01)**	2.4104e+02(4.76e+00)	2.4203e+02(8.21e-01)	**2.4262e+02(8.15e-02)**
	0.1	**2.4271e+02(5.64e-01)**	2.4253e+02(6.12e-01)	2.4016e+02(5.24e+00)	2.4124e+02(8.98e-01)	2.4248e+02(2.51e-01)
	1.0	1.8173e+02(2.60e+01)	2.3044e+02(1.21e+01)	**2.4233e+02(4.37e-01)**	2.3343e+02(1.69e+01)	2.4260e+02(6.18e-01)
	5.0	1.6191e+02(5.38e-02)	1.6553e+02(1.34e+01)	2.4224e+02(5.58e-01)	1.6193e+02(4.69e-02)	2.4212e+02(3.44e-01)
	10.0	1.6192e+02(5.80e-02)	1.6194e+02(4.10e-02)	2.4198e+02(8.68e-01)	1.5923e+02(1.45e+01)	2.4215e+02(4.38e-01)
$DTLZ3$, 7	0.0001	**2.1783e+03(1.09e+01)**	2.1433e+03(7.36e+01)	2.1193e+03(2.99e+02)	2.1699e+03(5.07e+01)	2.1804e+03(1.80e+01)
	0.01	2.1668e+03(5.52e+01)	2.1701e+03(1.08e+01)	2.1319e+03(2.04e+02)	**2.1852e+03(1.50e+00)**	**2.1850e+03(6.88e+00)**
	0.1	2.1604e+03(3.17e+01)	**2.1761e+03(9.92e+00)**	2.1726e+03(4.68e+01)	2.1807e+03(2.31e+01)	2.1709e+03(3.46e+01)
	1.0	1.4738e+03(8.71e+01)	2.0086e+03(2.65e+02)	2.1749e+03(3.64e+01)	2.0703e+03(1.52e+02)	2.1811e+03(2.53e+01)
	5.0	1.4575e+03(2.57e-01)	1.4576e+03(4.69e-01)	**2.1749e+03(3.56e+01)**	1.4575e+03(3.29e-01)	2.1753e+03(3.71e+01)
	10.0	1.4575e+03(3.62e-01)	1.4575e+03(3.77e-01)	2.1652e+03(5.33e+01)	1.4574e+03(5.07e-01)	2.1786e+03(2.55e+01)
$DTLZ3$, 10	0.0001	5.8143e+04(3.62e+02)	5.8246e+04(2.28e+02)	5.8206e+04(4.23e+03)	5.8974e+04(2.45e+02)	5.8959e+04(2.53e+02)
	0.01	**5.8439e+04(5.04e+02)**	**5.8275e+04(7.71e+01)**	**5.9040e+04(1.10e+01)**	**5.9043e+04(1.46e+01)**	**5.9047e+04(2.19e+00)**
	0.1	5.8038e+04(7.80e+02)	5.8022e+04(5.31e+02)	5.9003e+04(2.08e+02)	5.8979e+04(1.39e+02)	5.8944e+04(5.58e+02)
	1.0	4.2858e+04(5.80e+03)	5.4690e+04(2.79e+03)	5.8545e+04(1.84e+03)	5.4768e+04(5.75e+03)	5.8998e+04(1.28e+02)
	5.0	3.9358e+04(5.14e+00)	3.9795e+04(2.35e+03)	5.8961e+04(4.26e+02)	3.9359e+04(4.04e+00)	5.8557e+04(1.49e+03)
	10.0	3.9798e+04(2.35e+03)	3.9355e+04(1.08e+01)	5.7080e+04(9.97e+03)	3.9356e+04(7.12e+00)	5.8870e+04(4.88e+02)

Table C.3: Mean HV indicator values of 30 independent runs on $DTLZ1$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives. (continued)

Function, m	α	ACHE	RACHE	MCHE	RMCHE	AASF
$DTLZ3^{-1}$, 3	0.0001	2.4998e+01(3.29e+00)	2.4022e+01(4.48e+00)	2.4537e+01(6.87e-01)	2.4021e+01(3.30e+00)	2.4576e+01(3.47e-01)
	0.01	**2.5489e+01(1.01e+00)**	2.4495e+01(3.14e+00)	2.4472e+01(8.65e-01)	**2.4103e+01(3.27e+00)**	2.2776e+01(5.42e+00)
	0.1	2.4201e+01(4.45e+00)	**2.5061e+01(3.38e+00)**	**2.4638e+01(7.42e-02)**	2.3517e+01(4.58e+00)	2.4058e+01(3.25e+00)
	1.0	2.0806e+01(3.48e+00)	2.0713e+01(4.01e+00)	2.3795e+01(4.36e+00)	2.0464e+01(4.00e+00)	1.7064e+01(4.00e-01)
	5.0	1.8018e+01(2.52e+00)	1.6727e+01(2.40e+00)	1.9143e+01(2.85e+00)	1.8065e+01(2.53e+00)	**2.4849e+01(7.15e-02)**
	10.0	1.7688e+01(8.19e-01)	1.7413e+01(2.45e+00)	2.2812e+01(4.42e+00)	1.7467e+01(2.47e+00)	2.4825e+01(6.86e-02)
$DTLZ3^{-1}$, 5	0.0001	1.8254e+02(1.90e+00)	1.8207e+02(9.43e-01)	**1.6011e+02(4.85e-01)**	1.5998e+02(5.14e-01)	1.5992e+02(6.47e-01)
	0.01	**1.8270e+02(2.36e+00)**	1.8344e+02(2.44e+00)	1.6004e+02(7.92e-01)	1.5595e+02(2.65e+01)	1.6009e+02(6.12e-01)
	0.1	1.7503e+02(2.13e+00)	**1.8840e+02(8.97e+00)**	1.5998e+02(5.89e-01)	**1.6311e+02(1.25e+00)**	1.5929e+02(1.59e+00)
	1.0	1.2964e+02(2.54e+00)	1.3179e+02(2.57e+00)	1.5997e+02(8.20e-01)	1.2781e+02(1.64e+01)	1.5991e+02(8.34e-01)
	5.0	9.8355e+01(5.45e-01)	1.0036e+02(6.56e-01)	1.5404e+02(2.69e+01)	9.9842e+01(1.01e+00)	1.5937e+02(1.95e+00)
	10.0	9.3519e+01(1.60e+00)	9.4250e+01(5.75e-01)	1.5971e+02(1.17e+00)	9.3948e+01(2.76e-01)	**1.6026e+02(3.58e-01)**
$DTLZ3^{-1}$, 7	0.0001	1.2462e+03(2.48e+02)	1.1804e+03(8.30e+00)	7.3448e+02(1.13e+01)	7.0584e+02(1.18e+02)	7.3182e+02(1.29e+01)
	0.01	1.1668e+03(2.18e+02)	1.2247e+03(1.76e+02)	7.3192e+02(1.22e+01)	**7.0942e+02(8.92e+00)**	7.4052e+02(9.57e+00)
	0.1	**1.3491e+03(3.84e+02)**	**1.3791e+03(4.17e+02)**	7.2994e+02(1.27e+01)	6.5895e+02(1.35e+01)	7.4607e+02(1.03e+01)
	1.0	8.1306e+02(8.22e+01)	7.9828e+02(7.03e+01)	7.3081e+02(1.28e+01)	4.9260e+02(6.16e+00)	7.2194e+02(1.30e+02)
	5.0	5.0828e+02(1.39e+01)	5.1406e+02(4.47e+00)	**7.3466e+02(1.12e+01)**	4.2967e+02(7.80e+01)	**7.4724e+02(7.02e+00)**
	10.0	4.6056e+02(2.07e+00)	4.7154e+02(2.36e+00)	7.2829e+02(1.23e+01)	4.3870e+02(4.90e+00)	7.4664e+02(5.56e+00)
$DTLZ3^{-1}$, 10	0.0001	1.6417e+04(2.34e+02)	1.6478e+04(1.29e+02)	**6.2833e+03(1.41e+02)**	**6.2098e+03(1.37e+02)**	6.3170e+03(2.11e+02)
	0.01	**1.6990e+04(1.94e+02)**	**1.7062e+04(1.52e+02)**	6.1959e+03(1.54e+02)	5.9322e+03(1.20e+02)	6.6243e+03(1.39e+02)
	0.1	1.5017e+04(3.45e+02)	1.4842e+04(2.11e+03)	6.2199e+03(1.62e+02)	5.4010e+03(9.26e+01)	**6.8636e+03(9.75e+01)**
	1.0	8.1398e+03(1.21e+02)	8.8112e+03(4.15e+01)	6.1891e+03(1.37e+02)	4.6390e+03(2.34e+01)	6.6495e+03(8.44e+01)
	5.0	5.3241e+03(4.72e+01)	5.4890e+03(2.50e+01)	6.1977e+03(1.74e+02)	4.5218e+03(8.12e+01)	6.5255e+03(1.49e+02)
	10.0	4.8960e+03(4.88e+01)	4.9702e+03(2.51e+01)	6.2587e+03(1.98e+02)	4.3757e+03(8.17e+02)	6.5044e+03(1.24e+02)

Table C.4: Mean IGD+ indicator values (the standard deviation is shown in parentheses) of 30 independent runs on $DTLZ$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives.

Function, m	α	ACHE	RACHE	MCHE	RMCHE	AASF
$DTLZ1$, 3	0.0001	1.6168e-02(1.49e-04)	1.6196e-02(1.73e-04)	1.1964e-02(1.62e-04)	**1.1901e-02(1.39e-04)**	1.1920e-02(1.67e-04)
	0.01	1.6178e-02(1.36e-04)	**1.5303e-02(1.74e-04)**	1.1912e-02(1.29e-04)	1.3351e-02(1.02e-04)	**1.1868e-02(1.14e-04)**
	0.1	1.6093e-02(1.36e-04)	1.7017e-02(2.17e-04)	1.1899e-02(1.32e-04)	1.5724e-02(1.46e-04)	1.5072e-02(1.02e-04)
	1.0	1.6040e-02(9.61e-05)	4.2501e-02(1.62e-03)	**1.1852e-02(1.05e-04)**	4.2389e-02(7.25e-04)	4.1997e-02(1.04e-03)
	5.0	**1.5994e-02(8.60e-05)**	8.6195e-02(9.94e-03)	1.1863e-02(9.72e-05)	9.1254e-02(6.93e-03)	7.1034e-02(1.71e-03)
	10.0	1.6004e-02(1.08e-04)	8.9001e-02(9.97e-03)	1.1836e-02(1.25e-04)	9.1812e-02(7.34e-03)	8.0725e-02(2.71e-03)
$DTLZ1$, 5	0.0001	1.0859e-01(8.01e-03)	1.0815e-01(7.12e-03)	**5.1632e-02(2.52e-03)**	**4.8545e-02(1.82e-03)**	4.6916e-02(8.54e-04)
	0.01	1.0554e-01(6.57e-03)	1.0514e-01(6.56e-03)	5.1904e-02(3.19e-03)	1.1939e-01(2.40e-02)	**4.7825e-02(1.64e-03)**
	0.1	1.0156e-01(6.41e-03)	**1.0064e-01(5.45e-03)**	5.2689e-02(2.45e-03)	1.2558e-01(2.16e-02)	4.8464e-02(1.23e-03)
	1.0	9.8483e-02(5.74e-03)	1.7777e-01(3.02e-02)	5.2599e-02(2.51e-03)	1.8085e-01(5.16e-02)	8.4589e-02(8.42e-04)
	5.0	9.9175e-02(6.55e-03)	1.6304e-01(4.63e-02)	5.1902e-02(3.10e-03)	1.3791e-01(4.36e-02)	1.0127e-01(3.57e-03)
	10.0	**9.7398e-02(9.83e-03)**	1.7016e-01(4.72e-02)	5.2149e-02(2.45e-03)	1.3694e-01(4.03e-02)	1.0170e-01(4.25e-03)
$DTLZ1$, 7	0.0001	2.3000e-01(4.85e-02)	2.3748e-01(4.36e-02)	9.8280e-02(6.27e-03)	**8.6123e-02(3.35e-03)**	**8.4865e-02(3.14e-03)**
	0.01	2.1872e-01(3.91e-02)	2.2651e-01(4.65e-02)	9.6883e-02(7.29e-03)	2.2097e-01(6.02e-02)	8.5355e-02(2.92e-03)
	0.1	2.1538e-01(5.01e-02)	2.0494e-01(4.16e-02)	9.7712e-02(5.48e-03)	1.7163e-01(6.63e-02)	8.7053e-02(2.87e-03)
	1.0	1.8256e-01(5.60e-02)	1.7612e-01(4.99e-02)	9.6985e-02(4.40e-03)	1.3652e-01(5.39e-02)	1.0478e-01(3.82e-03)
	5.0	1.7691e-01(4.79e-02)	1.5151e-01(4.27e-02)	**9.6391e-02(7.64e-03)**	1.3278e-01(5.53e-02)	1.0872e-01(5.48e-03)
	10.0	**1.6912e-01(5.21e-02)**	**1.4244e-01(3.84e-02)**	1.0085e-01(9.06e-03)	1.1241e-01(4.25e-02)	1.1101e-01(6.10e-03)
$DTLZ1$, 10	0.0001	3.4613e-01(3.36e-02)	3.4673e-01(3.94e-02)	**9.7480e-02(2.47e-02)**	1.0624e-01(3.06e-02)	**1.1001e-01(3.17e-02)**
	0.01	3.3680e-01(5.35e-02)	3.2185e-01(5.38e-02)	1.1680e-01(4.31e-02)	1.2452e-01(8.02e-02)	1.2486e-01(2.94e-02)
	0.1	3.1826e-01(7.24e-02)	3.0719e-01(6.95e-02)	1.0017e-01(2.79e-02)	1.0320e-01(4.38e-02)	1.3540e-01(2.33e-02)
	1.0	2.9043e-01(7.14e-02)	2.8322e-01(7.09e-02)	1.0311e-01(3.19e-02)	9.9685e-02(3.98e-02)	1.5543e-01(3.95e-03)
	5.0	2.8126e-01(7.21e-02)	2.8320e-01(7.70e-02)	1.1861e-01(3.96e-02)	**8.5544e-02(2.37e-02)**	1.6168e-01(3.39e-03)
	10.0	**2.5839e-01(7.23e-02)**	**2.5623e-01(7.27e-02)**	1.0358e-01(3.65e-02)	8.9865e-02(1.77e-02)	1.6258e-01(3.74e-03)

Table C.5: Mean IGD+ indicator values of 30 independent runs on $DTLZ$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives. (continued)

$DTLZ3$, 3	0.0001	1.4139e-01(2.96e-01)	9.8713e-02(2.39e-01)	1.0039e-01(2.45e-01)	**7.1118e-02(2.08e-01)**	1.9521e-01(3.66e-01)
	0.01	**6.6053e-02(1.75e-01)**	**6.5968e-02(1.74e-01)**	**3.9601e-02(7.28e-02)**	8.0434e-02(1.75e-01)	**5.8526e-02(1.76e-01)**
	0.1	1.5861e-01(8.19e-02)	7.7554e-02(1.74e-01)	1.6100e-01(3.45e-01)	8.9508e-02(1.98e-01)	1.1609e-01(2.50e-01)
	1.0	2.6931e-01(1.24e-01)	1.9256e-01(9.47e-02)	1.7750e-01(1.79e-01)	1.5830e-01(2.83e-02)	2.1343e-01(2.48e-01)
	5.0	6.0650e-01(2.05e-01)	6.5515e-01(3.32e-01)	2.4731e-01(2.99e-01)	6.3307e-01(2.75e-01)	2.1343e-01(2.49e-01)
	10.0	5.8955e-01(4.13e-02)	6.2323e-01(2.02e-01)	1.7924e-01(1.79e-01)	5.8489e-01(6.67e-02)	2.1305e-01(2.49e-01)
$DTLZ3$, 5	0.0001	2.2573e-01(2.46e-01)	2.4280e-01(2.51e-01)	2.0867e-01(2.97e-01)	**9.5297e-02(5.71e-02)**	9.1953e-02(5.90e-02)
	0.01	**1.6537e-01(4.435-02)**	2.5390e-01(2.93e-01)	1.9643e-01(2.4936e-01)	2.1658e-01(9.19e-02)	**7.7814e-02(1.53e-02)**
	0.1	2.7937e-01(1.80e-01)	**2.0346e-01(1.82e-01)**	2.5007e-01(3.37e-01)	2.5066e-01(1.09e-01)	1.1741e-01(3.57e-02)
	1.0	7.8051e-01(1.07e-01)	7.2943e-01(2.34e-01)	**1.5721e-01(1.52e-02)**	6.4036e-01(2.13e-01)	1.7052e-01(7.74e-03)
	5.0	8.3710e-01(6.68e-04)	8.3243e-01(1.68e-02)	1.8882e-01(5.20e-02)	8.3694e-01(5.79e-04)	1.6948e-01(1.51e-03)
	10.0	8.3697e-01(7.17e-04)	8.3679e-01(5.07e-04)	2.0192e-01(8.50e-02)	8.7020e-01(1.79e-01)	1.8732e-01(9.50e-02)
$DTLZ3$, 7	0.0001	6.3517e-01(2.50e-01)	6.5300e-01(1.89e-01)	4.8117e-01(4.37e-01)	3.3805e-01(4.30e-01)	2.5726e-01(2.90e-01)
	0.01	**5.9895e-01(2.30e-01)**	**5.7363e-01(1.29e-01)**	4.9661e-01(5.07e-01)	**2.2834e-01(1.01e-01)**	**1.8876e-01(1.83e-01)**
	0.1	7.1206e-01(1.83e-01)	5.7615e-01(1.68e-01)	4.2465e-01(3.54e-01)	2.8465e-01(2.02e-01)	3.2453e-01(3.54e-01)
	1.0	9.3266e-01(4.79e-03)	9.0309e-01(2.51e-02)	**3.2369e-01(2.78e-01)**	8.7176e-01(8.02e-02)	2.5109e-01(2.22e-01)
	5.0	9.3366e-01(2.96e-04)	9.3360e-01(6.43e-04)	4.6966e-01(3.37e-01)	9.3364e-01(4.52e-04)	3.1567e-01(2.78e-01)
	10.0	9.3372e-01(4.76e-04)	9.3376e-01(5.12e-04)	4.6039e-01(3.26e-01)	9.3382e-01(6.96e-04)	2.8311e-01(2.19e-01)
$DTLZ3$, 10	0.0001	9.4537e-01(8.62e-03)	9.4292e-01(5.46e-03)	6.3341e-01(4.29e-01)	4.9467e-01(2.85e-01)	5.4923e-01(2.95e-01)
	0.01	**9.3492e-01(4.36e-02)**	9.3887e-01(1.37e-02)	5.3701e-01(3.19e-01)	**3.9958e-01(2.55e-01)**	**4.3223e-01(2.44e-01)**
	0.1	9.4615e-01(1.75e-02)	**9.3349e-01(3.24e-02)**	**5.3135e-01(3.66e-01)**	4.9357e-01(2.96e-01)	5.2972e-01(3.63e-01)
	1.0	9.8119e-01(3.15e-03)	9.7275e-01(7.27e-03)	6.7937e-01(4.68e-01)	9.7108e-01(9.60e-03)	5.0938e-01(2.48e-01)
	5.0	9.8312e-01(2.61e-04)	9.8292e-01(1.35e-03)	7.1167e-01(5.05e-01)	9.8307e-01(2.05e-04)	7.2472e-01(3.67e-01)
	10.0	9.8277e-01(1.26e-03)	9.8325e-01(5.51e-04)	5.3719e-01(5.37e-01)	9.8321e-01(3.61e-04)	6.7530e-01(2.65e-01)

Table C.6: Mean IGD+ indicator values of 30 independent runs on $DTLZ$, $DTLZ3$ and $DTLZ3^{-1}$ with 3, 5, 7 and 10 objectives. (continued)

$DTLZ3^{-1}$, 3	0.0001	9.9152e-02(3.17e-01)	1.4611e-01(3.96e-01)	5.5794e-02(5.79e-02)	1.0670e-01(2.94e-01)	5.4174e-02(3.55e-02)
	0.01	**4.5947e-02(2.07e-02)**	**9.7476e-02(2.92e-01)**	6.1075e-02(5.80e-02)	**9.8929e-02(2.89e-01)**	2.1079e-01(4.80e-01)
	0.1	1.3878e-01(4.04e-01)	1.0026e-01(2.71e-01)	**4.7600e-02(4.43e-03)**	1.4771e-01(4.08e-01)	9.6858e-02(2.86e-01)
	1.0	1.9192e-01(3.19e-01)	2.0183e-01(3.83e-01)	1.5423e-01(6.00e-01)	2.1243e-01(3.82e-01)	4.6272e-02(6.28e-02)
	5.0	2.7304e-01(2.55e-01)	2.6708e-01(2.56e-01)	1.1662e-01(2.89e-01)	2.6540e-01(2.55e-01)	**3.1858e-02(5.18e-03)**
	10.0	2.6007e-01(4.90e-02)	2.9436e-01(2.51e-01)	1.7172e-01(3.90e-01)	2.9234e-01(2.52e-01)	3.2781e-02(7.05e-03)
$DTLZ3^{-1}$, 5	0.0001	1.2523e-01(4.91e-03)	1.2649e-01(6.10e-03)	2.2709e-01(2.59e-03)	2.2633e-01(4.52e-03)	2.2729e-01(3.79e-03)
	0.01	1.0148e-01(8.96e-03)	1.1505e-01(4.56e-03)	2.2735e-01(3.37e-03)	2.8230e-01(3.48e-01)	2.2712e-01(3.04e-03)
	0.1	**8.6413e-02(4.11e-03)**	**1.1395e-01(3.63e-03)**	2.2588e-01(4.06e-03)	**1.5700e-01(3.16e-03)**	2.2714e-01(3.77e-03)
	1.0	2.2090e-01(5.39e-03)	1.9896e-01(2.16e-03)	2.2380e-01(3.19e-03)	2.2434e-01(1.17e-01)	2.2435e-01(2.32e-03)
	5.0	3.4808e-01(3.49e-03)	3.3835e-01(1.97e-03)	2.9722e-01(3.44e-01)	3.3997e-01(3.05e-03)	**2.2356e-01(3.32e-03)**
	10.0	3.7135e-01(6.94e-03)	3.6648e-01(3.55e-03)	**2.2343e-01(2.70e-03)**	3.6834e-01(2.22e-03)	2.2513e-01(3.27e-03)
$DTLZ3^{-1}$, 7	0.0001	1.7793e-01(7.86e-03)	1.8122e-01(8.09e-03)	2.9807e-01(6.00e-03)	3.3394e-01(1.97e-01)	2.9808e-01(7.54e-03)
	0.01	2.1813e-01(4.16e-01)	1.5507e-01(8.75e-03)	**2.9645e-01(6.81e-03)**	**2.9820e-01(6.47e-03)**	**2.9639e-01(9.98e-03)**
	0.1	**1.2977e-01(6.82e-03)**	**1.3309e-01(3.84e-03)**	2.9698e-01(9.31e-03)	3.0084e-01(5.17e-03)	3.0442e-01(8.41e-03)
	1.0	3.0324e-01(1.02e-02)	2.7259e-01(2.13e-03)	2.9707e-01(7.46e-03)	4.1608e-01(6.74e-03)	3.8303e-01(3.89e-01)
	5.0	4.3108e-01(3.05e-03)	4.1882e-01(2.45e-03)	2.9882e-01(6.04e-03)	5.2545e-01(3.49e-01)	3.1200e-01(2.60e-03)
	10.0	4.5304e-01(1.83e-03)	4.4578e-01(1.83e-03)	2.9872e-01(7.82e-03)	4.6558e-01(4.21e-03)	3.1196e-01(2.42e-03)
$DTLZ3^{-1}$, 10	0.0001	2.1387e-01(1.44e-02)	2.1189e-01(1.16e-02)	**4.4093e-01(1.01e-02)**	**4.4638e-01(1.01e-02)**	4.4046e-01(1.08e-02)
	0.01	**1.7074e-01(1.28e-02)**	**1.9048e-01(1.72e-02)**	4.4757e-01(9.99e-03)	4.6165e-01(7.87e-03)	**4.2695e-01(9.41e-03)**
	0.1	1.8530e-01(1.07e-02)	1.9253e-01(1.01e-01)	4.4513e-01(9.33e-03)	4.8981e-01(4.71e-03)	4.2826e-01(8.88e-03)
	1.0	3.9843e-01(6.17e-03)	3.5646e-01(2.46e-03)	4.4533e-01(9.33e-03)	5.3550e-01(1.61e-03)	4.5263e-01(5.22e-03)
	5.0	5.1526e-01(3.09e-03)	5.0173e-01(2.29e-03)	4.4392e-01(1.10e-02)	5.4434e-01(3.29e-03)	4.5759e-01(6.07e-03)
	10.0	5.3264e-01(2.18e-03)	5.2579e-01(2.38e-03)	4.4285e-01(8.70e-03)	6.2075e-01(4.11e-01)	4.5913e-01(4.84e-03)

Table C.7: Statistics (mean and standard deviation) of the HV indicator for EVOCA's recommendations.

m	CHE	ASF	ACHE	RACHE	MCHE	RMCHE	AASF
				LAME $\gamma = 1.0$			
2	↑ 8.4949e+00(4.42e-07)	↑ 8.4949e+00(4.26e-07)	8.4949e+00(3.55e-15)	↑ 8.4949e+00(5.12e-07)	↑ 8.4848e+00(0.00e+00)	↑ 8.4949e+00(3.55e-15)	↑ 8.3515e+00(1.60e-05)
3	2.6756e+01(1.69e-04)	↑ 2.6595e+01(1.67e-02)	↑ 2.6743e+01(7.65e-03)	↑ 2.6601e+01(1.72e-02)	↑ 2.6658e+01(9.13e-03)	↑ 2.6597e+01(1.38e-02)	↑ 2.6645e+01(2.12e-05)
4	↑ 8.0829e+01(2.13e-04)	↑ 8.0392e+01(5.78e-02)	↑ 8.0761e+01(1.93e-02)	↑ 8.0411e+01(6.44e-02)	↑ 8.0538e+01(8.27e-02)	↑ 8.0506e+01(7.15e-02)	8.0884e+01(5.32e-05)
6	↑ 7.2895e+02(6.99e-03)	↑ 7.2675e+02(4.48e-01)	↑ 7.2792e+02(7.18e-01)	↑ 7.2709e+02(5.14e-01)	↑ 7.2811e+02(7.07e-01)	↑ 7.2688e+02(3.30e+00)	7.2896e+02(4.31e-04)
8	↑ 6.5557e+03(3.24e+00)	↑ 6.5447e+03(5.72e+00)	↑ 6.5056e+03(3.49e+01)	↑ 6.5459e+03(5.85e+00)	↑ 6.5229e+03(2.17e+01)	↑ 6.5569e+03(5.01e+00)	6.5609e+03(1.67e-03)
10	↑ 5.8956e+04(2.15e+01)	↑ 5.8953e+04(4.18e+01)	↑ 5.8316e+04(4.37e+02)	↑ 5.8960e+04(4.30e+01)	↑ 5.8487e+04(3.62e+02)	↑ 5.9037e+04(2.34e+01)	5.9049e+04(3.00e-03)
				DTLZ1			
2	8.8734e+00(2.18e-04)	↑ 8.8734e+00(2.43e-04)	↑ 8.8729e+00(7.84e-04)	↑ 8.8734e+00(5.56e-04)	↑ 8.8710e+00(1.87e-04)	↑ 8.8734e+00(2.74e-04)	↑ 8.8379e+00(3.44e-04)
3	2.6969e+01(1.48e-04)	↑ 2.6900e+01(1.03e-02)	↑ 2.6967e+01(6.68e-04)	↑ 2.6906e+01(1.59e-02)	↑ 2.6958e+01(1.32e-03)	↑ 2.6899e+01(1.17e-02)	↑ 2.6955e+01(5.57e-05)
4	↑ 8.0989e+01(5.74e-05)	↑ 8.0857e+01(2.97e-02)	↑ 8.0982e+01(1.16e-03)	↑ 8.0884e+01(3.33e-02)	↑ 8.0977e+01(2.41e-03)	↑ 8.0972e+01(4.60e-03)	8.0992e+01(4.28e-05)
6	↑ 7.2899e+02(1.07e-02)	↑ 7.2861e+02(1.59e-01)	↑ 7.2898e+02(1.26e-02)	↑ 7.2870e+02(1.77e-01)	↑ 7.2891e+02(5.84e-02)	↑ 7.2884e+02(3.40e-01)	7.2899e+02(4.25e-04)
8	↑ 6.5601e+03(6.25e-01)	↑ 6.5578e+03(1.78e+00)	↑ 6.5569e+03(1.23e+00)	↑ 6.5571e+03(2.41e+00)	↑ 6.5577e+03(2.19e+00)	↑ 6.5609e+03(7.66e-02)	6.5610e+03(5.61e-04)
10	↑ 5.9029e+04(1.89e+01)	↑ 5.8997e+04(2.34e+01)	↑ 5.8940e+04(6.15e+01)	↑ 5.8990e+04(3.55e+01)	↑ 5.8994e+04(4.21e+01)	↑ 5.9048e+04(3.63e-02)	5.9049e+04(0.00e+00)
				LAME $\gamma = 2.0$			
2	↑ 8.2101e+00(9.43e-07)	↑ 8.2101e+00(7.18e-06)	8.2101e+00(3.55e-15)	↑ 8.0079e+00(2.99e-02)	↑ 8.2101e+00(6.90e-07)	↑ 8.2101e+00(1.57e-06)	↑ 8.0000e+00(0.00e+00)
3	2.6388e+01(5.30e-04)	↑ 2.6294e+01(1.95e-02)	↑ 2.6379e+01(2.13e-03)	↑ 2.6183e+01(1.22e-01)	↑ 2.6379e+01(2.69e-03)	↑ 2.6328e+01(1.42e-02)	↑ 2.6000e+01(0.00e+00)
4	8.0443e+01(1.00e-03)	↑ 8.0361e+01(5.09e-02)	↑ 8.0341e+01(8.82e-03)	↑ 7.9918e+01(4.18e-01)	↑ 8.0346e+01(9.00e-03)	↑ 8.0397e+01(4.62e-02)	↑ 8.0000e+01(0.00e+00)
6	7.2855e+02(2.21e-01)	↑ 7.2805e+02(2.46e-01)	↑ 7.2787e+02(4.57e-01)	↑ 7.2513e+02(4.99e+00)	↑ 7.2764e+02(7.10e-01)	↑ 7.2831e+02(1.38e-01)	↑ 7.2787e+02(4.74e-01)
8	↑ 6.5169e+03(1.11e+01)	↑ 6.5554e+03(2.02e+00)	↑ 6.4758e+03(2.24e+01)	↑ 6.5478e+03(3.49e+01)	↑ 6.4737e+03(3.20e+01)	↑ 6.5566e+03(1.87e+00)	6.5594e+03(8.02e-01)
10	↑ 5.8494e+04(1.02e+02)	↑ 5.8999e+04(2.10e+01)	↑ 5.8260e+04(2.31e+02)	↑ 5.8827e+04(4.20e+02)	↑ 5.8309e+04(4.05e+01)	↑ 5.9011e+04(2.92e+01)	5.9044e+04(2.25e+00)
				LAME $\gamma = 4.0$			
2	↑ 8.0696e+00(1.61e-04)	↑ 8.0696e+00(1.73e-04)	8.0697e+00(3.55e-15)	↑ 8.0000e+00(0.00e+00)	↑ 8.0697e+00(4.58e-07)	↑ 8.0697e+00(3.72e-07)	↑ 8.0000e+00(0.00e+00)
3	↑ 2.6129e+01(5.02e-04)	↑ 2.6051e+01(1.81e-02)	↑ 2.6133e+01(9.56e-04)	↑ 2.5698e+01(6.81e-01)	↑ 2.6130e+01(1.15e-03)	2.6154e+01(3.16e-04)	↑ 2.6000e+01(0.00e+00)
4	↑ 8.0151e+01(7.55e-04)	↑ 8.0052e+01(4.07e-02)	↑ 8.0122e+01(2.63e-03)	↑ 7.6854e+01(2.87e+00)	↑ 8.0119e+01(3.29e-03)	8.0221e+01(9.43e-04)	↑ 7.9934e+01(3.53e-01)
6	↑ 7.2751e+02(7.91e-01)	↑ 7.2714e+02(3.59e-01)	↑ 7.2746e+02(1.49e+00)	↑ 7.1208e+02(2.27e+01)	↑ 7.2691e+02(1.54e+00)	7.2816e+02(3.36e-01)	↑ 7.2720e+02(1.53e+00)
8	↑ 6.4747e+03(1.78e+01)	↑ 6.5429e+03(1.06e+01)	↑ 6.4766e+03(2.95e+01)	↑ 6.4699e+03(1.79e+02)	↑ 6.4604e+03(5.28e+01)	6.5586e+03(2.16e+00)	↑ 6.5483e+03(4.30e+01)
10	↑ 5.7683e+04(4.54e+02)	↑ 5.8845e+04(8.15e+01)	↑ 5.8273e+04(1.47e+02)	↑ 5.7231e+04(2.28e+03)	↑ 5.8080e+04(5.30e+02)	5.9022e+04(3.17e+01)	↑ 5.8798e+04(1.17e+03)
				DTLZ3			
2	8.1979e+00(9.16e-03)	↑ 8.1949e+00(1.12e-02)	↑ 8.1175e+00(4.28e-01)	↑ 7.9914e+00(3.54e-02)	↑ 8.1938e+00(9.74e-03)	↑ 8.1962e+00(6.83e-03)	↑ 7.9900e+00(7.28e-03)
3	2.6385e+01(5.32e-03)	↑ 2.6374e+01(1.08e-02)	↑ 2.6379e+01(6.68e-03)	↑ 2.6037e+01(9.61e-02)	↑ 2.6378e+01(8.55e-03)	↑ 2.6373e+01(1.29e-02)	↑ 2.5993e+01(3.20e-03)
4	↑ 8.0445e+01(6.66e-03)	↑ 8.0503e+01(1.60e-02)	↑ 8.0361e+01(1.22e-02)	↑ 7.8536e+01(1.26e+00)	↑ 8.0373e+01(1.70e-02)	8.0505e+01(1.69e-02)	↑ 7.9992e+01(5.18e-03)
6	↑ 7.2722e+02(1.61e+00)	↑ 7.2756e+02(9.89e-01)	↑ 7.2713e+02(1.32e+00)	↑ 7.2124e+02(6.45e+00)	↑ 7.2597e+02(3.81e+00)	7.2788e+02(7.05e-01)	↑ 7.2651e+02(1.46e+00)
8	↑ 6.4924e+03(3.82e+01)	↑ 6.5510e+03(1.41e+01)	↑ 6.4797e+03(2.01e-01)	↑ 6.5016e+03(6.09e+01)	↑ 6.4793e+03(2.03e+00)	↑ 6.5499e+03(1.28e+01)	6.5565e+03(3.19e+00)
10	↑ 5.8150e+04(3.87e+02)	↑ 5.8939e+04(1.20e+02)	↑ 5.8283e+04(7.85e+01)	↑ 5.8580e+04(5.86e+02)	↑ 5.8310e+04(1.82e+01)	↑ 5.8935e+04(1.31e+02)	5.9030e+04(1.24e+01)

Table C.8: Statistics (mean and standard deviation) of the HV indicator for EVOCA's recommendations (continued).

m	CHE	ASF	ACHE	RACHE	MCHE	RMCHE	AASF
				LAME $\gamma = 0.3$			
2	↑ 8.9623e+00(2.76e-06)	↑ 8.9623e+00(2.76e-06)	↑ 8.7317e+00(4.25e-06)	↑ 8.9623e+00(2.61e-06)	↑ 8.7315e+00(1.71e-06)	↑ 8.6495e+00(8.19e-07)	8.9653e+00(5.73e-07)
3	2.6997e+01(6.79e-06)	↑ 2.6966e+01(2.78e-04)	↑ 2.6754e+01(5.57e-04)	↑ 2.6966e+01(9.38e-05)	↑ 2.6755e+01(4.50e-05)	↑ 2.6651e+01(6.97e-05)	↑ 2.6991e+01(8.67e-05)
4	8.0999e+01(0.00e+00)	↑ 8.0946e+01(1.82e-03)	↑ 8.0666e+01(2.85e-03)	↑ 8.0942e+01(7.81e-04)	↑ 8.0671e+01(1.00e-04)	↑ 8.0671e+01(7.56e-05)	↑ 8.0984e+01(1.61e-04)
6	7.2900e+02(0.00e+00)	↑ 7.2883e+02(2.27e-02)	↑ 7.2792e+02(1.79e-02)	↑ 7.2876e+02(2.03e-02)	↑ 7.2797e+02(1.64e-04)	↑ 7.2797e+02(3.54e-03)	↑ 7.2891e+02(1.63e-03)
8	6.5610e+03(0.00e+00)	↑ 6.5600e+03(1.63e-01)	↑ 6.5563e+03(6.90e-02)	↑ 6.5592e+03(1.78e-01)	↑ 6.5565e+03(7.52e-04)	↑ 6.5474e+03(3.79e-01)	↑ 6.5602e+03(2.83e-02)
10	5.9049e+04(0.00e+00)	↑ 5.9043e+04(9.66e-01)	↑ 5.9024e+04(7.07e-01)	↑ 5.9037e+04(1.81e+00)	↑ 5.9025e+04(7.54e-02)	↑ 5.8966e+04(3.59e-01)	↑ 5.9042e+04(2.74e-01)
				LAME $\gamma = 0.5$			
2	↑ 8.8281e+00(1.62e-06)	↑ 8.8281e+00(1.22e-06)	↑ 8.5124e+00(1.18e-05)	↑ 8.8281e+00(1.69e-06)	↑ 8.5108e+00(6.22e-06)	↑ 8.2997e+00(4.57e-06)	8.8299e+00(3.00e-07)
3	↑ 2.6964e+01(2.49e-05)	↑ 2.6869e+01(1.74e-03)	↑ 2.6302e+01(1.73e-03)	↑ 2.6867e+01(1.02e-03)	↑ 2.6332e+01(9.24e-04)	↑ 2.5872e+01(1.50e-03)	2.6980e+01(5.12e-05)
4	8.0990e+01(2.03e-05)	↑ 8.0737e+01(2.22e-02)	↑ 7.9679e+01(9.34e-03)	↑ 8.0688e+01(1.68e-02)	↑ 7.9722e+01(3.56e-03)	↑ 7.9690e+01(2.85e-03)	↑ 8.0987e+01(3.16e-04)
6	7.2899e+02(1.13e-13)	↑ 7.2815e+02(1.39e-01)	↑ 7.2194e+02(2.08e-01)	↑ 7.2764e+02(1.99e-01)	↑ 7.2236e+02(1.91e-02)	↑ 7.2231e+02(3.18e-02)	↑ 7.2892e+02(1.08e-02)
8	6.5609e+03(8.54e-03)	↑ 6.5559e+03(1.41e+00)	↑ 6.5166e+03(1.18e+00)	↑ 6.5503e+03(2.51e+00)	↑ 6.5197e+03(1.17e-01)	↑ 6.3701e+03(4.49e+00)	↑ 6.5602e+03(1.12e-01)
10	5.9048e+04(2.23e-01)	↑ 5.9006e+04(1.42e+01)	↑ 5.8735e+04(1.68e+01)	↑ 5.8946e+04(2.94e+01)	↑ 5.8760e+04(1.66e+00)	↑ 5.7371e+04(1.16e+01)	↑ 5.9043e+04(1.13e+00)
				DTLZ3 CONVEX			
2	8.6428e+00(1.08e-02)	↑ 8.6386e+00(1.16e-02)	↑ 8.0329e+00(2.00e+00)	8.6426e+00(1.15e-02) ↓	↑ 8.5689e+00(2.03e-02)	↑ 8.4797e+00(2.85e-02)	↑ 8.6247e+00(2.06e-02)
3	↑ 2.6920e+01(2.43e-03)	↑ 2.6871e+01(1.88e-02)	↑ 2.6199e+01(4.13e-02)	↑ 2.6850e+01(1.58e-02)	↑ 2.6391e+01(2.77e-02)	↑ 2.5759e+01(2.34e-02)	2.6943e+01(1.35e-03)
4	↑ 8.0974e+01(8.96e-04)	↑ 8.0861e+01(3.15e-02)	↑ 7.8804e+01(9.02e-02)	↑ 8.0789e+01(3.92e-02)	↑ 7.9007e+01(1.85e-02)	↑ 7.8880e+01(2.43e-02)	8.0991e+01(3.85e-04)
6	7.2899e+02(2.69e-03)	↑ 7.2852e+02(1.58e-01)	↑ 7.1840e+02(8.30e-01)	↑ 7.2803e+02(2.71e-01)	↑ 7.2048e+02(7.66e-02)	↑ 7.2028e+02(2.27e-01)	↑ 7.2899e+02(3.98e-03)
8	↑ 6.5607e+03(2.29e-01)	↑ 6.5564e+03(2.32e+00)	↑ 6.5010e+03(3.76e+00)	↑ 6.5496e+03(2.75e+00)	↑ 6.5124e+03(6.64e-01)	↑ 6.3636e+03(9.74e+00)	6.5609e+03(5.28e-02)
10	↑ 5.9041e+04(5.29e+00)	↑ 5.8996e+04(2.91e+01)	↑ 5.8635e+04(3.62e+01)	↑ 5.8794e+04(5.22e+01)	↑ 5.8719e+04(9.68e+00)	↑ 5.7217e+04(2.02e+01)	5.9048e+04(4.24e-01)
				WFG1			
2	↑ 6.3436e+00(3.30e-01)	↑ 6.4494e+00(3.59e-01)	↑ 6.3612e+00(2.64e-01)	↑ 6.2494e+00(3.29e-01)	↑ 6.3774e+00(2.79e-01)	6.4650e+00(2.27e-01)	↑ 6.2169e+00(3.01e-01)
3	↑ 5.4557e+01(1.31e+00)	5.5924e+01(1.21e+00)	↑ 5.4631e+01(1.29e+00)	↑ 5.5321e+01(1.57e+00)	↑ 5.5154e+01(1.46e+00)	↑ 5.4685e+01(1.45e+00)	↑ 5.4932e+01(1.61e+00)
4	↑ 4.7297e+02(7.32e+00)	4.8950e+02(5.94e+00)	↑ 4.8774e+02(1.02e+01)	↑ 4.8657e+02(1.06e+01)	↑ 3.8174e+02(1.61e+01)	↑ 3.5287e+02(1.48e+01)	↑ 4.8755e+02(1.02e+01)
6	↑ 5.9250e+04(1.46e+03)	↑ 6.1194e+04(1.22e+03)	↑ 6.1555e+04(1.20e+03)	↑ 6.1029e+04(1.10e+03)	↑ 4.5435e+04(1.28e+03)	↑ 4.5451e+04(2.11e+03)	6.1957e+04(9.73e+02)
8	↑ 1.3657e+07(7.85e+05)	↑ 1.2722e+07(8.77e+05)	↑ 1.4296e+07(2.12e+05)	↑ 1.3400e+07(4.59e+05)	↑ 1.0917e+07(4.74e+05)	9.3898e+06(3.46e+05)	↑ 1.4576e+07(2.72e+05)
10	↑ 5.0939e+09(1.56e+08)	↑ 4.4472e+09(2.65e+08)	↑ 5.1956e+09(1.08e+08)	↑ 4.6085e+09(2.77e+08)	↑ 4.0507e+09(1.04e+08)	↑ 3.4198e+09(8.73e+07)	5.2623e+09(1.08e+08)
				WFG2			
2	↑ 1.0295e+01(4.60e-01)	↑ 1.0451e+01(5.16e-01)	↑ 1.0242e+01(5.29e-01)	↑ 1.0415e+01(5.04e-01)	↑ 1.0380e+01(6.09e-01)	1.0533e+01(4.39e-01)	↑ 1.0422e+01(5.22e-01)
3	↑ 9.8140e+01(7.42e-01)	↑ 9.9337e+01(9.86e-01)	↑ 9.8137e+01(9.66e-01)	↑ 9.9320e+01(8.03e-01)	9.9986e+01(7.49e-01)	↑ 9.9500e+01(5.97e-01)	↑ 1.0019e+02(6.23e-01)
4	↑ 8.9326e+02(7.72e+00)	↑ 9.1573e+02(9.51e+00)	↑ 9.1828e+02(9.06e+00)	↑ 9.1465e+02(8.95e+00)	↑ 9.0093e+02(9.01e+00)	↑ 8.7569e+02(1.39e+01)	9.2773e+02(9.04e+00)
6	↑ 1.1898e+05(6.03e+03)	↑ 1.2888e+05(1.19e+03)	↑ 1.3019e+05(2.41e+03)	↑ 1.3032e+05(1.73e+03)	↑ 1.2711e+05(2.61e+03)	↑ 1.2577e+05(3.56e+03)	1.3172e+05(2.09e+03)
8	↑ 2.0349e+07(3.85e+06)	↑ 3.3077e+07(6.63e+05)	↑ 3.2995e+07(7.58e+05)	↑ 3.3081e+07(4.49e+05)	↑ 3.2532e+07(5.21e+05)	↑ 3.0382e+07(1.03e+06)	3.3496e+07(6.14e+05)
10	↑ 7.5623e+09(2.07e+09)	↑ 1.2845e+10(4.61e+08)	↑ 1.3101e+10(2.89e+08)	↑ 1.2682e+10(6.46e+08)	↑ 1.2658e+10(3.90e+08)	↑ 1.1669e+10(6.05e+08)	↑ 1.3100e+10(3.93e+08)
				WFG3			
2	↑ 1.0790e+01(7.44e-02)	↑ 1.0787e+01(9.35e-02)	↑ 1.0825e+01(6.84e-02)	↑ 1.0798e+01(7.70e-02)	↑ 1.0784e+01(1.05e-01)	↑ 1.0772e+01(7.72e-02)	9.9910e+00(9.67e-02)
3	↑ 7.4665e+01(5.46e-01)	↑ 7.4697e+01(5.46e-01)	↑ 7.4541e+01(5.78e-01)	↑ 7.4466e+01(2.88e-01)	7.4920e+01(5.56e-01)	↑ 7.4813e+01(5.83e-01)	↑ 6.9662e+01(3.35e-01)
4	↑ 6.4318e+02(8.58e+00)	↑ 6.3988e+02(7.92e+00)	6.4612e+02(7.19e+00)	↑ 6.3743e+02(8.80e+00)	↑ 6.3735e+02(6.93e+00)	↑ 6.3807e+02(6.59e+00)	↑ 6.3810e+02(8.65e+00)
6	↑ 8.0624e+04(1.22e+03)	↑ 8.1288e+04(2.11e+03)	↑ 8.0781e+04(1.38e+03)	↑ 8.1557e+04(2.93e+03)	↑ 8.1663e+04(2.30e+03)	8.2974e+04(2.22e+03)	↑ 8.2717e+04(2.71e+03)
8	↑ 1.9265e+07(2.94e+05)	↑ 1.5923e+07(5.25e+05)	↑ 1.9505e+07(2.64e+05)	↑ 1.6155e+07(5.33e+05)	2.0236e+07(6.91e+05)	↑ 1.6028e+07(4.27e+05)	↑ 1.5872e+07(4.12e+05)
10	↑ 7.1474e+09(1.20e+08)	↑ 5.5793e+09(2.18e+08)	↑ 7.1465e+09(1.13e+08)	↑ 5.5510e+09(2.53e+08)	7.7003e+09(2.33e+08)	↑ 5.5851e+09(2.22e+08)	↑ 5.5289e+09(2.12e+08)

Table C.9: Statistics (mean and standard deviation) of the IGD+ indicator for EVOCA's recommendations.

m	CHE	ASF	ACHE	RACHE	MCHE	RMCHE	AASF
				LAME $\gamma = 1.0$			
2	↑ 2.5132e-03(7.27e-08)	↑ 2.5131e-03(7.88e-08)	↑ 2.5131e-03(8.69e-08)	2.5131e-03(6.66e-08)	↑ 2.5135e-03(8.59e-08)	↑ 2.5134e-03(7.58e-08)	↑ 2.5132e-03(1.02e-07)
3	↑ 3.2407e-02(1.54e-05)	2.2445e-02(1.21e-05)	↑ 3.2334e-02(1.09e-04)	↑ 2.2450e-02(9.91e-06)	↑ 3.2309e-02(1.23e-04)	↑ 2.2455e-02(9.59e-06)	↑ 2.2484e-02(1.74e-05)
4	↑ 9.7143e-02(1.10e-04)	↑ 5.0280e-02(3.06e-05)	↑ 9.5863e-02(1.22e-03)	↑ 5.0280e-02(2.71e-05)	1.3777e-01(4.02e-03)	↑ 5.0320e-02(3.65e-05)	↑ 5.0529e-02(4.77e-05)
6	↑ 1.5900e-01(3.61e-03)	1.0566e-01(1.27e-04)	↑ 3.0270e-01(1.16e-01)	↑ 1.0581e-01(4.94e-04)	↑ 2.4558e-01(6.02e-02)	↑ 1.0599e-01(1.55e-03)	↑ 1.0638e-01(1.43e-04)
8	↑ 1.9028e-01(2.75e-02)	1.6648e-01(2.93e-03)	↑ 4.2513e-01(1.32e-01)	1.6655e-01(3.41e-03) ↓	↑ 4.1865e-01(1.01e-01)	↑ 1.6699e-01(3.40e-03)	↑ 1.6738e-01(9.36e-04)
10	↑ 2.3367e-01(3.73e-02)	1.9454e-01(2.80e-02)	↑ 4.8224e-01(1.14e-01)	↑ 1.9913e-01(2.63e-02)	↑ 5.2689e-01(1.57e-01)	↑ 1.9554e-01(2.60e-02)	↑ 2.1413e-01(2.28e-02)
				DTLZ1			
2	↑ 1.5119e-03(2.24e-04)	↑ 1.5690e-03(2.58e-04)	↑ 1.4599e-03(1.19e-04)	1.4503e-03(1.21e-04)	1.4716e-03(1.88e-04)↓	↑ 1.5179e-03(2.59e-04)	↑ 1.5991e-03(2.90e-04)
3	↑ 1.6334e-02(2.74e-04)	↑ 1.1585e-02(1.79e-04)	↑ 1.6096e-02(9.42e-05)	1.1443e-02(8.54e-05)	↑ 1.6233e-02(1.61e-04)	↑ 1.1646e-02(2.99e-04)	↑ 1.1683e-02(3.05e-04)
4	↑ 4.8325e-02(2.92e-04)	↑ 2.5756e-02(2.93e-04)	↑ 4.8340e-02(5.11e-04)	2.5721e-02(2.39e-04)	↑ 6.6917e-02(2.17e-03)	↑ 2.5883e-02(3.99e-04)	↑ 2.5912e-02(3.62e-04)
6	↑ 8.6295e-02(6.26e-03)	↑ 5.6554e-02(2.48e-03)	↑ 2.2741e-01(5.83e-02)	↑ 5.5706e-02(2.21e-03)	1.0770e-01(1.88e-02)	↑ 5.5668e-02(2.57e-03)	↑ 5.5177e-02(1.37e-03)
8	1.1802e-01(1.84e-02)	↑ 8.4483e-02(4.31e-03)	↑ 3.0655e-01(7.02e-02)	↑ 8.5854e-02(6.14e-03)	↑ 2.3690e-01(3.51e-02)	↑ 8.4109e-02(6.20e-03)	↑ 8.6014e-02(3.83e-03)
10	1.3592e-01(2.18e-02)	↑ 7.6054e-02(7.98e-03)	↑ 2.3239e-01(8.08e-02)	↑ 7.6255e-02(6.60e-03)	↑ 2.8464e-01(8.75e-02)	↑ 7.6947e-02(6.87e-03)	↑ 8.1577e-02(5.89e-03)
				LAME $\gamma = 2.0$			
2	2.1178e-03(7.01e-08)	↑ 2.1184e-03(2.09e-06)	↑ 2.1185e-03(6.39e-08)	2.1189e-03(5.79e-06)↓	↑ 2.1185e-03(6.77e-08)	↑ 2.1187e-03(9.03e-07)	↑ 2.1185e-03(9.36e-08)
3	↑ 2.8408e-02(1.13e-04)	↑ 1.8794e-02(1.84e-05)	↑ 2.8210e-02(1.58e-04)	↑ 1.8781e-02(2.59e-05)	↑ 2.8179e-02(1.78e-04)	↑ 1.8783e-02(2.55e-05)	1.8667e-02(2.58e-05)
4	↑ 7.9768e-02(3.97e-04)	↑ 3.8937e-02(5.59e-05)	↑ 8.8182e-02(1.15e-03)	3.8647e-02(9.24e-05)	↑ 8.7979e-02(1.10e-03)	↑ 3.8912e-02(4.30e-05)	↑ 3.8725e-02(3.30e-05)
6	1.0498e-01(2.18e-03)	↑ 7.4781e-02(2.87e-03)	↑ 1.6936e-01(5.63e-04)	↑ 7.2813e-02(4.70e-03)	↑ 1.6933e-01(1.07e-03)	↑ 7.5638e-02(8.23e-04)	↑ 7.5605e-02(7.46e-04)
8	1.1951e-01(6.59e-03)	↑ 7.7226e-02(4.00e-03)	↑ 1.7784e-01(8.57e-03)	↑ 7.7154e-02(3.22e-03)	↑ 1.7507e-01(7.19e-03)	↑ 7.9670e-02(3.53e-03)	↑ 7.9160e-02(3.76e-03)
10	1.2453e-01(9.21e-03)	↑ 8.3229e-02(5.56e-04)	↑ 1.8758e-01(2.81e-02)	↑ 8.1577e-02(7.85e-04)	↑ 1.7854e-01(5.51e-03)	↑ 8.2808e-02(6.13e-04)	↑ 8.1374e-02(3.25e-04)
				LAME $\gamma = 4.0$			
2	↑ 2.1969e-03(1.75e-07)	↑ 2.1969e-03(2.85e-07)	↑ 2.1978e-03(6.60e-08)	2.1968e-03(5.94e-08)	↑ 2.1972e-03(5.08e-08)	↑ 2.1972e-03(5.89e-08)	↑ 2.1978e-03(9.05e-08)
3	↑ 2.1728e-02(6.08e-05)	↑ 1.6165e-02(1.51e-05)	↑ 2.1514e-02(8.85e-05)	↑ 1.6187e-02(2.83e-05)	↑ 2.1626e-02(7.14e-05)	↑ 1.6178e-02(3.81e-05)	1.6079e-02(2.98e-05)
4	↑ 4.8427e-02(1.46e-04)	↑ 3.1548e-02(5.30e-05)	↑ 5.0653e-02(2.09e-04)	↑ 3.2171e-02(3.39e-04)	↑ 5.0649e-02(2.09e-04)	↑ 3.1492e-02(1.10e-04)	3.1422e-02(8.31e-05)
6	↑ 9.1879e-02(2.38e-02)	↑ 5.4899e-02(1.98e-03)	1.0174e-01(2.51e-02)	↑ 5.8926e-02(1.21e-02)	↑ 1.1456e-01(2.85e-02)	↑ 5.7744e-02(1.16e-02)	↑ 5.6075e-02(5.98e-03)
8	↑ 2.9463e-01(1.69e-02)	1.0119e-01(4.30e-02)	↑ 2.9137e-01(1.84e-02)	↑ 9.8117e-02(3.47e-02)	↑ 3.0038e-01(2.75e-02)	↑ 9.0913e-02(3.47e-02)	↑ 9.1733e-02(3.02e-02)
10	↑ 4.0132e-01(3.02e-02)	↑ 1.4688e-01(4.18e-02)	↑ 3.6337e-01(1.27e-02)	↑ 1.8496e-01(4.48e-02)	↑ 3.7297e-01(3.01e-02)	↑ 1.6210e-01(5.32e-02)	1.3526e-01(3.58e-02)
				DTLZ3			
2	↑ 8.4694e-03(4.56e-03)	1.0068e-02(6.43e-03)	4.2645e-02(1.79e-01) ↓	↑ 4.2712e-02(1.78e-01)	↑ 1.0849e-02(5.72e-03)	↑ 9.5995e-03(3.77e-03)	↑ 8.5550e-03(4.35e-03)
3	↑ 3.1332e-02(2.13e-03)	2.3957e-02(2.38e-03)	↑ 3.2142e-02(2.45e-03)	↑ 2.5399e-02(3.34e-03)	↑ 3.2293e-02(3.58e-03)	↑ 2.4624e-02(3.05e-03)	↑ 2.4574e-02(2.97e-03)
4	↑ 8.1297e-02(2.27e-03)	↑ 4.7369e-02(3.13e-03)	↑ 9.1781e-02(3.39e-03)	4.4490e-02(2.75e-03)	↑ 8.9890e-02(3.10e-03)	↑ 4.6776e-02(3.51e-03)	↑ 4.5986e-02(3.33e-03)
6	↑ 2.4262e-01(1.54e-01)	↑ 1.5055e-01(8.57e-02)	↑ 2.4356e-01(1.26e-01)	↑ 1.5194e-01(9.22e-02)	↑ 3.2302e-01(1.71e-01)	↑ 1.5170e-01(8.61e-02)	1.2133e-01(6.33e-02)
8	↑ 7.3374e-01(1.05e-01)	3.1936e-01(2.18e-01)	↑ 8.5581e-01(1.83e-02)	↑ 4.2319e-01(2.17e-01)	↑ 8.2124e-01(9.93e-02)	↑ 4.1000e-01(2.55e-01)	↑ 4.6581e-01(2.28e-01)
10	↑ 8.9599e-01(7.01e-02)	5.5026e-01(2.27e-01)	↑ 9.1422e-01(6.80e-02)	↑ 5.6497e-01(2.47e-01)	↑ 9.2788e-01(2.74e-02)	↑ 5.7560e-01(2.84e-01)	↑ 5.8610e-01(2.53e-01)

Table C.10: Statistics (mean and standard deviation) of the IGD+ indicator for EVOCA's recommendations. (continued)

m	CHE	ASF	ACHE	RACHE	MCHE	RMCHE	AASF
				LAME $\gamma = 0.3$			
2	↑ 2.1022e-03(1.83e-06)	↑ 2.1022e-03(1.89e-06)	↑ 2.0217e-03(1.42e-06)	↑ 2.1021e-03(1.48e-06)	2.0075e-03(1.59e-06)	↑ 4.1063e-03(1.92e-07)	↑ 7.7436e-04(4.47e-07)
3	↑ 8.4967e-03(6.77e-06)	↑ 2.6365e-03(1.74e-05)	↑ 7.0284e-03(3.74e-05)	↑ 2.6429e-03(1.48e-05)	↑ 5.9311e-03(4.14e-05)	↑ 5.4597e-03(2.15e-04)	1.2817e-03(2.35e-06)
4	↑ 7.7256e-03(1.05e-05)	↑ 1.9618e-03(2.30e-05)	↑ 5.9018e-03(6.19e-05)	↑ 1.9624e-03(1.70e-05)	↑ 1.8854e-03(3.86e-05)	↑ 1.9611e-03(2.65e-05)	1.3825e-03(3.49e-06)
6	↑ 5.7241e-03(2.01e-04)	1.1275e-03(2.41e-05)	↑ 3.6513e-03(8.94e-05)	↑ 1.1275e-03(3.38e-05)	↑ 1.3625e-03(5.32e-05)	↑ 1.1354e-03(3.33e-05)	↑ 9.7371e-04(1.53e-05)
8	↑ 3.3891e-03(2.51e-04)	↑ 6.9041e-04(1.40e-05)	↑ 3.2391e-03(2.32e-05)	↑ 6.9669e-04(1.57e-05)	3.0623e-03(1.74e-05)	↑ 6.9689e-04(1.89e-05)	↑ 5.1361e-04(6.01e-06)
10	↑ 1.9833e-03(3.10e-04)	↑ 4.8837e-04(3.77e-05)	↑ 1.6584e-03(7.95e-06)	↑ 4.8321e-04(1.71e-05)	1.5868e-03(6.93e-06)	↑ 4.8681e-04(2.12e-05)	↑ 3.3306e-04(4.31e-06)
				LAME $\gamma = 0.5$			
2	↑ 8.5414e-04(1.17e-06)	↑ 8.5464e-04(8.27e-07)	↑ 8.7307e-04(1.03e-06)	↑ 8.5464e-04(8.27e-07)	↑ 8.6150e-04(9.26e-07)	2.1453e-04(8.83e-08)	↑ 3.5583e-06(6.71e-09)
3	↑ 3.0512e-10(2.48e-11)	↑ 1.7000e-04(2.36e-06)	↑ 2.0804e-11(1.14e-12)	↑ 1.7102e-04(1.46e-06)	↑ 1.3798e-09(2.39e-11)	1.1996e-05(1.91e-07)	↑ 5.6953e-07(1.37e-08)
4	↑ 5.1308e-10(7.47e-12)	↑ 9.7796e-05(2.63e-06)	1.1500e-10(6.81e-12)	↑ 9.8850e-05(1.47e-06)	↑ 4.0698e-07(6.33e-09)	↑ 9.6560e-05(1.90e-06)	↑ 5.8471e-07(2.22e-08)
6	↑ 9.0260e-10(3.47e-11)	↑ 4.4097e-05(5.04e-06)	↑ 7.3291e-10(8.43e-11)	↑ 4.2840e-05(5.72e-06)	2.5599e-07(4.87e-08)	↑ 4.0719e-05(8.27e-06)	↑ 4.0157e-07(9.80e-08)
8	↑ 4.9842e-06(4.56e-07)	↑ 2.6841e-05(1.10e-05)	↑ 4.9378e-06(1.38e-08)	2.5510e-05(1.03e-05)	↑ 4.9179e-06(1.12e-08)	↑ 2.7134e-05(1.02e-05)	↑ 3.5841e-07(8.55e-08)
10	↑ 4.3187e-06(7.64e-07)	1.7756e-05(1.00e-05)	↑ 3.4628e-06(5.79e-09)	↑ 1.9975e-05(9.91e-06)	↑ 3.4518e-06(7.61e-09)	↑ 1.8816e-05(9.24e-06)	↑ 2.6183e-07(6.51e-08)
				DTLZ3 CONVEX			
2	↑ 9.0629e-02(6.53e-03)	↑ 9.2829e-02(6.57e-03)	↑ 9.0277e-02(5.83e-03)	↑ 9.0937e-02(6.32e-03)	↑ 9.0852e-02(7.71e-03)	↑ 9.1409e-02(7.59e-03)	1.7972e-01(4.69e-01)
3	↑ 1.5650e-06(1.01e-06)	↑ 8.8691e-05(5.94e-05)	↑ 1.4622e-06(1.19e-06)	↑ 1.1753e-04(5.63e-05)	↑ 1.8017e-07(5.93e-07)	1.1607e-05(1.31e-06)	↑ 7.1272e-08(6.84e-08)
4	↑ 5.5177e-07(7.72e-07)	↑ 2.7423e-05(1.81e-05)	↑ 7.1213e-07(9.34e-07)	↑ 3.5926e-05(2.26e-05)	2.3609e-07(8.30e-08)	↑ 2.6263e-05(1.85e-05)	↑ 4.3536e-08(2.92e-08)
6	↑ 7.5753e-07(9.29e-07)	↑ 3.7044e-06(5.55e-06)	↑ 3.8995e-07(4.83e-07)	↑ 4.0642e-06(6.29e-06)	1.5391e-07(8.35e-08)	↑ 2.4299e-06(3.31e-06)	↑ 7.3875e-08(6.21e-08)
8	↑ 6.0899e-06(1.71e-06)	2.2307e-06(2.84e-06)	↑ 5.1909e-06(1.72e-07)	↑ 4.4270e-06(4.48e-06)	↑ 5.3326e-06(1.29e-07)	↑ 2.5648e-06(2.40e-06)	↑ 4.5291e-08(2.44e-08)
10	↑ 6.7298e-06(1.67e-06)	↑ 3.1304e-06(3.05e-06)	↑ 3.6405e-06(7.96e-08)	1.0899e-05(8.38e-06)	↑ 3.6716e-06(5.61e-08)	↑ 1.1202e-05(9.03e-06)	↑ 6.4705e-08(3.05e-08)
				WFG1			
2	↑ 9.9642e-01(4.76e-02)	↑ 9.7745e-01(5.79e-02)	↑ 9.9703e-01(4.19e-02)	↑ 9.9429e-01(4.79e-02)	↑ 9.9379e-01(4.68e-02)	9.7441e-01(3.24e-02)	↑ 9.8404e-01(4.50e-02)
3	↑ 1.1625e+00(1.19e-02)	↑ 1.1648e+00(1.19e-02)	1.1596e+00(1.22e-02)	↑ 1.1650e+00(1.12e-02)	↑ 1.1641e+00(1.35e-02)	↑ 1.1691e+00(9.94e-03)	↑ 1.1670e+00(1.24e-02)
4	↑ 1.4174e+00(5.93e-02)	1.3527e+00(1.83e-02)	↑ 1.4227e+00(5.80e-02)	↑ 1.3657e+00(2.74e-02)	↑ 1.4140e+00(6.18e-02)	↑ 1.3631e+00(2.59e-02)	↑ 1.3839e+00(2.95e-02)
6	↑ 1.8401e+00(9.60e-02)	1.7331e+00(4.02e-02)	↑ 1.8735e+00(1.21e-01)	↑ 1.7381e+00(4.87e-02)	↑ 1.8391e+00(9.32e-02)	↑ 1.7519e+00(3.89e-02)	↑ 1.7666e+00(5.82e-02)
8	↑ 2.2443e+00(1.26e-01)	↑ 2.2146e+00(1.66e-01)	↑ 2.1784e+00(8.39e-02)	↑ 2.0876e+00(6.56e-02)	↑ 2.1741e+00(7.78e-02)	↑ 2.1803e+00(7.38e-02)	2.0434e+00(8.84e-02)
10	↑ 2.5047e+00(1.07e-01)	↑ 2.5115e+00(1.27e-01)	↑ 2.4242e+00(5.28e-02)	↑ 2.5451e+00(1.32e-01)	↑ 2.3836e+00(5.90e-02)	↑ 3.3780e+00(3.58e-01)	2.3636e+00(8.22e-02)
				WFG2			
2	↑ 9.7826e-02(4.96e-02)	↑ 8.3896e-02(5.08e-02)	↑ 8.8277e-02(5.05e-02)	↑ 8.8200e-02(5.38e-02)	↑ 8.7446e-02(4.45e-02)	↑ 8.0228e-02(4.15e-02)	7.6507e-02(4.05e-02)
3	1.1888e-01(1.47e-02)	↑ 6.7075e-02(2.41e-02)	↑ 9.1332e-02(1.44e-02)	↑ 6.1411e-02(2.24e-02)	↑ 8.0628e-02(2.53e-02)	↑ 6.1367e-02(1.94e-02)	↑ 5.4650e-02(1.45e-02)
4	↑ 3.1454e-01(1.86e-02)	↑ 1.3561e-01(3.59e-02)	↑ 3.0305e-01(3.09e-02)	↑ 1.4303e-01(4.46e-02)	↑ 2.9836e-01(3.93e-02)	↑ 1.5011e-01(3.17e-02)	1.3343e-01(4.73e-02)
6	↑ 9.3362e-01(3.67e-01)	↑ 2.6324e-01(5.18e-02)	↑ 5.7302e-01(2.09e-01)	2.3746e-01(4.00e-02)	↑ 7.2255e-01(2.44e-01)	2.4313e-01(5.35e-02) ↓	↑ 3.4164e-01(7.71e-02)
8	↑ 3.9348e+00(6.78e-01)	↑ 4.3669e-01(1.59e-01)	1.0874e+00(4.49e-01)	↑ 3.0205e-01(1.30e-01)	↑ 1.1738e+00(3.94e-01)	↑ 3.0686e-01(1.03e-01)	↑ 4.7149e-01(1.27e-01)
10	↑ 4.6236e+00(1.48e+00)	↑ 7.1061e-01(2.25e-01)	1.0225e+00(3.24e-01)	↑ 5.5109e-01(2.27e-01)	↑ 1.3046e+00(3.20e-01)	↑ 5.8136e-01(2.21e-01)	↑ 6.8507e-01(1.07e-01)
				WFG3			
2	↑ 2.9775e-02(1.17e-02)	↑ 3.0277e-02(1.48e-02)	2.7334e-02(1.44e-02) ↓	2.6580e-02(1.34e-02)	↑ 3.3798e-02(1.79e-02)	↑ 3.3286e-02(1.47e-02)	↑ 2.8751e-02(1.38e-02)
3	↑ 7.6939e-02(2.70e-02)	↑ 6.7119e-02(3.02e-02)	↑ 4.3499e-02(1.85e-02)	3.1529e-02(1.23e-02)	↑ 4.1095e-02(1.70e-02)	↑ 5.8486e-02(2.79e-02)	↑ 5.8060e-02(2.21e-02)
4	↑ 2.0309e-01(4.84e-02)	↑ 1.6782e-01(3.85e-02)	↑ 1.0678e-01(3.57e-02)	↑ 1.4203e-01(2.32e-02)	1.0004e-01(3.71e-02)	↑ 1.4729e-01(3.21e-02)	↑ 1.4711e-01(3.43e-02)
6	1.1603e+00(1.19e-01)	↑ 7.7433e-01(1.16e-01)	↑ 3.2373e-01(7.55e-02)	↑ 7.3914e-01(1.84e-01)	↑ 2.3845e-01(8.18e-02)	↑ 8.5643e-01(5.65e-02)	↑ 7.4802e-01(1.24e-01)
8	↑ 1.7036e+00(1.30e-01)	↑ 1.4224e+00(1.03e-01)	↑ 5.3335e-01(1.53e-01)	↑ 1.3483e+00(6.90e-02)	↑ 2.9756e-01(8.71e-02)	1.3069e+00(6.12e-02)	↑ 1.3370e+00(4.33e-02)
10	↑ 2.4274e+00(7.00e-02)	↑ 2.0559e+00(1.40e-01)	↑ 8.8578e-01(1.06e-01)	↑ 1.7471e+00(7.20e-02)	↑ 5.6732e-01(9.88e-02)	↑ 1.7521e+00(8.67e-02)	1.6629e+00(5.67e-02)

Figure C.1: Scenario with convex geometry and 3, 5, 7 objectives

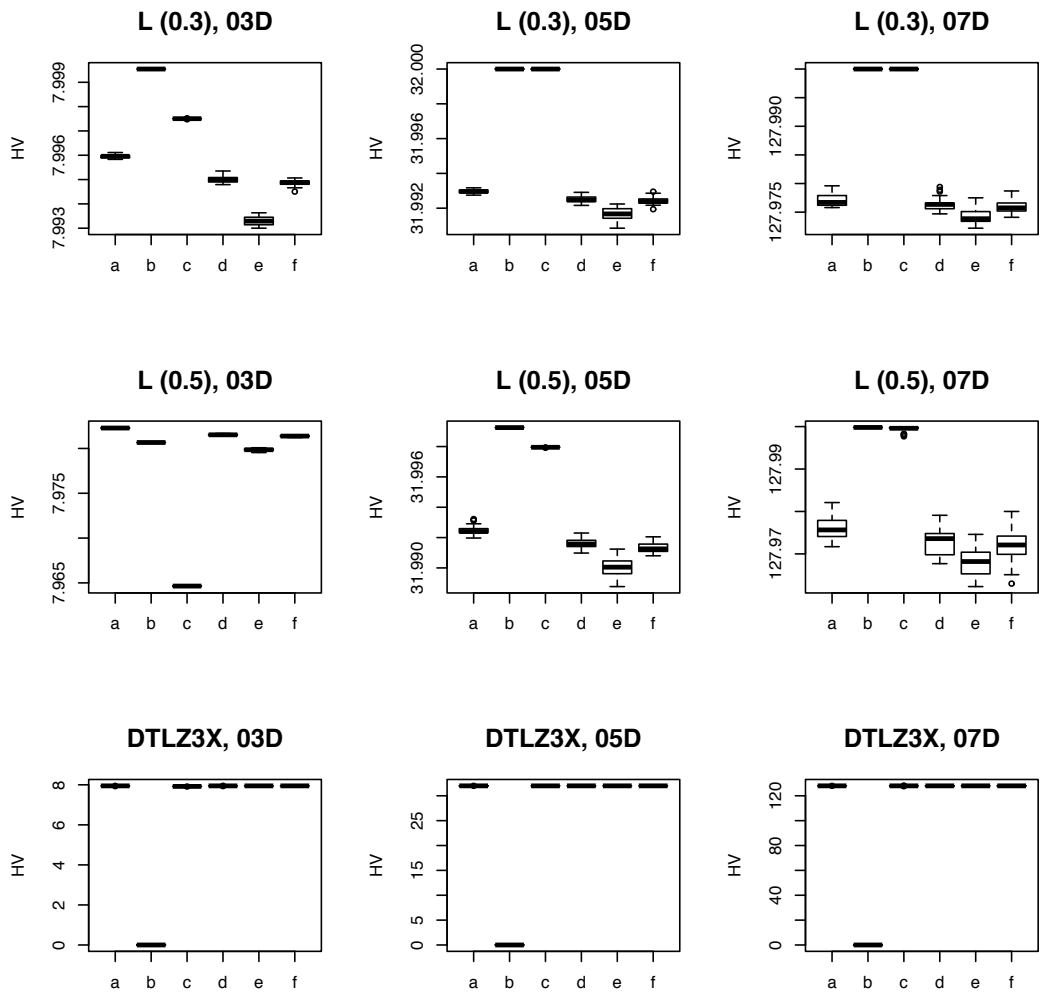


Figure C.2: Scenario with linear geometry and 3, 5, 7 objectives

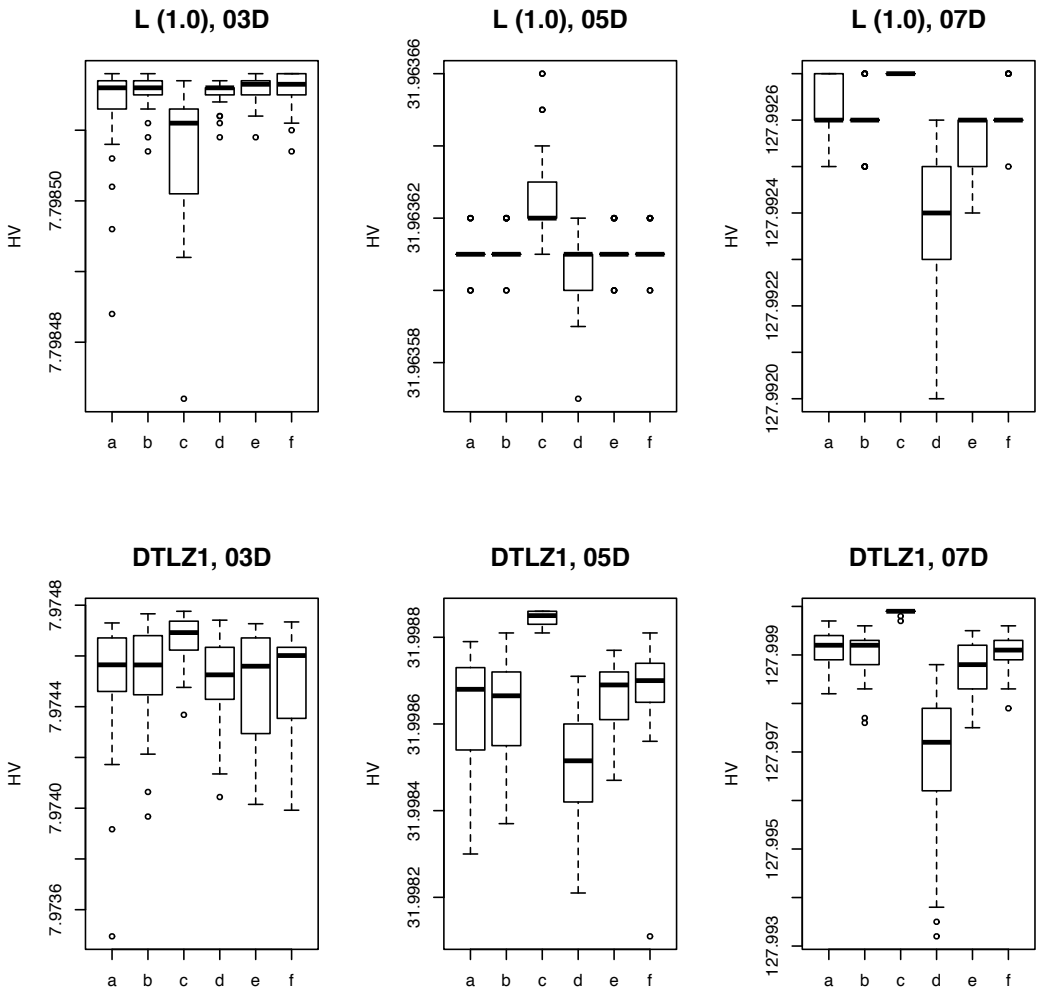
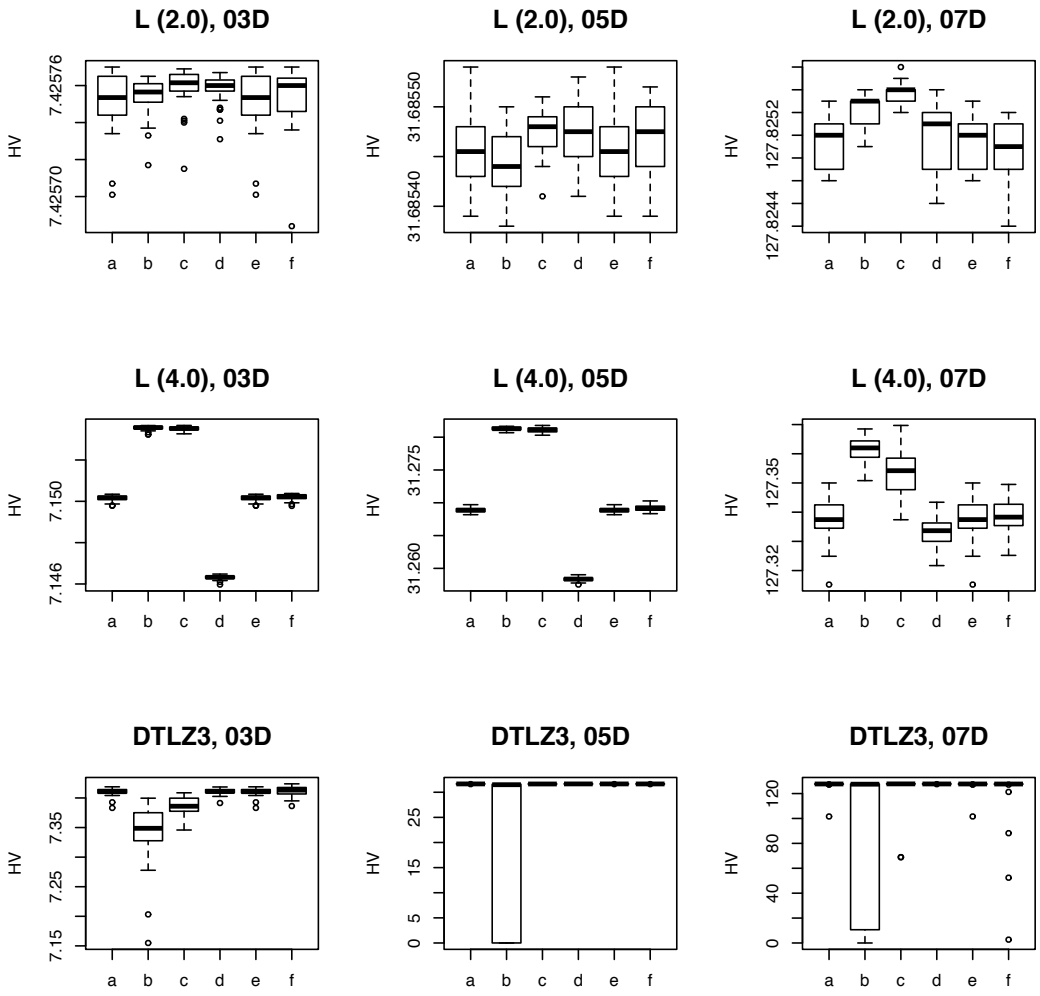


Figure C.3: Scenario with concave geometry and 3, 5, 7 objectives



Appendix D

Convergence Speed Caused by the Scalarizing Functions based on the Chebyshev Model

In this Appendix, we present figures that correspond the case study regarding the effect on convergence speed of the Scalarizing Functions based on the Chebyshev model such as the Penalty Boundary Intersection (PBI) and the Weighted Norm (WN) functions. This material complements the results presented in Chapter 6

Figures D.11 to D.63 show the convergence plots for different scenarios. We ran tests on $DTLZ1$, $DTLZ3$ and $DTLZ3^{-}1$ which have linear, concave and convex Pareto front shapes, respectively. We adopted 2, 3, and 5 objectives. The number of weight vectors was 100, 120 and 210 for each dimension. The parameter values employed were the following:

- ACHE, RACHE, AASF, MCHE, RMCHE with $\alpha \in \{0, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 3.0, 5.0, 7.0, 10.0\}$
- PBI with $\theta = \{0.1, 1.0, 2.0, 5.0, 10.0, 50.0\}$ and
- WN with $p = \{0.5, 1.0, 2.0, 3.0, 5.0, 10.0, 100.0\}$

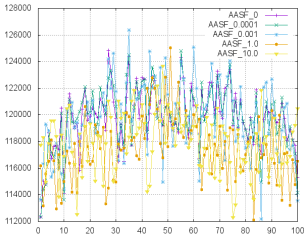


Figure D.1: AASF in DTLZ1 with 2 objectives.

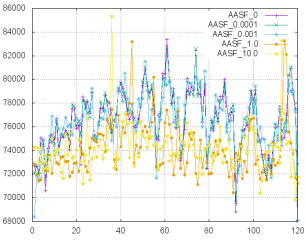


Figure D.2: AASF in DTLZ1 with 3 objectives.

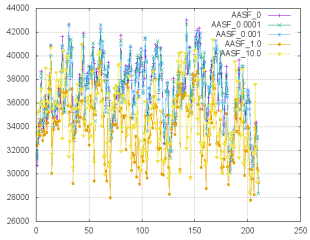


Figure D.3: AASF in DTLZ1 with 5 objectives.

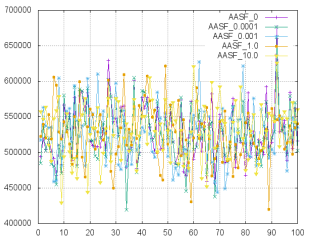


Figure D.4: AASF in DTLZ3 with 2 objectives.

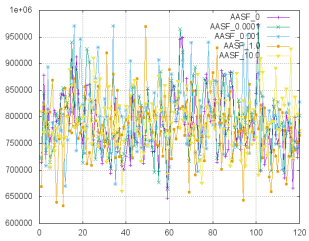


Figure D.5: AASF in DTLZ3 with 3 objectives.

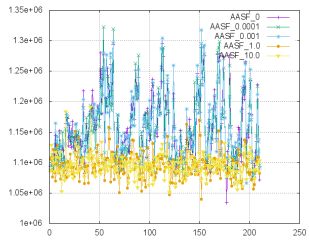


Figure D.6: AASF in DTLZ3 with 5 objectives.

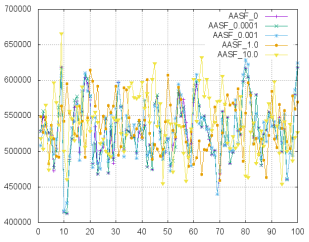


Figure D.7: AASF in DTLZ3 convex with 2 objectives.

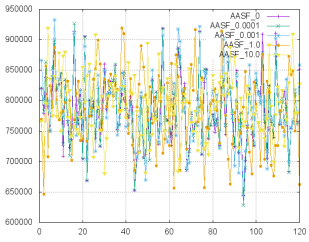


Figure D.8: AASF in DTLZ3 convex with 3 objectives.

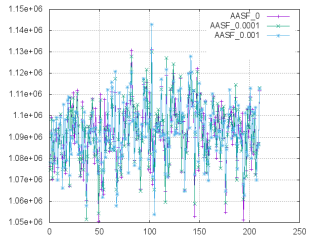


Figure D.9: AASF in DTLZ3 convex with 5 objectives.

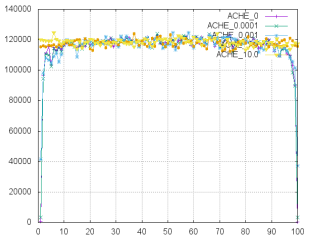


Figure D.10: ACHE in DTLZ1 with 2 objectives.

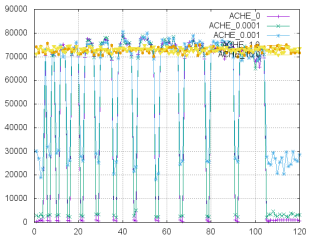


Figure D.11: ACHE in DTLZ1 with 3 objectives.

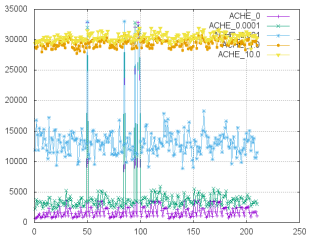


Figure D.12: ACHE in DTLZ1 with 5 objectives.

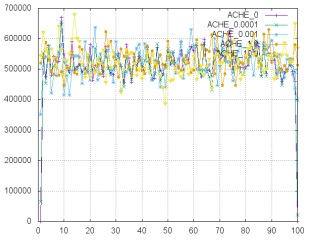


Figure D.13: ACHE in DTLZ3 with 2 objectives.

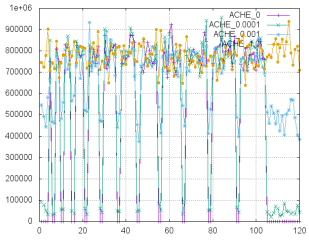


Figure D.14: ACHE in DTLZ3 with 3 objectives.

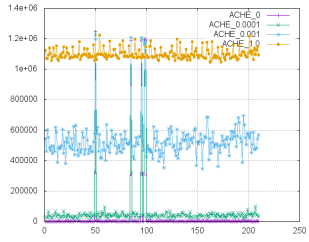


Figure D.15: ACHE in DTLZ3 with 5 objectives.

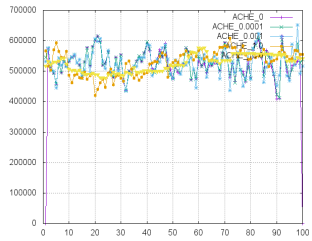


Figure D.16: ACHE in DTLZ3 convex with 2 objectives.

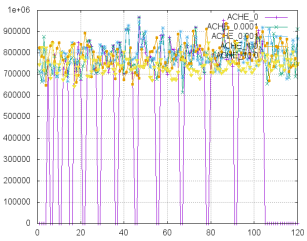


Figure D.17: ACHE in DTLZ3 convex with 3 objectives.

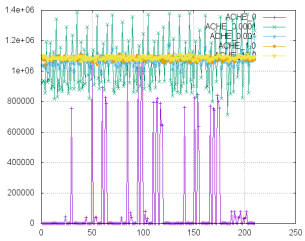


Figure D.18: ACHE in DTLZ3 convex with 5 objectives.

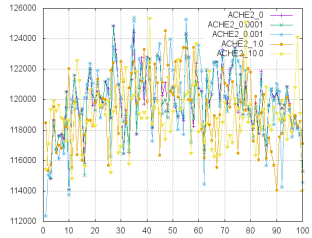


Figure D.19: RACHE in DTLZ1 with 2 objectives.

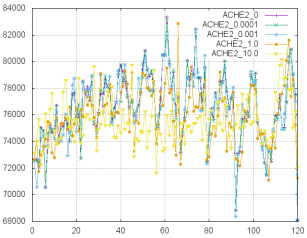


Figure D.20: RACHE in DTLZ1 with 3 objectives.

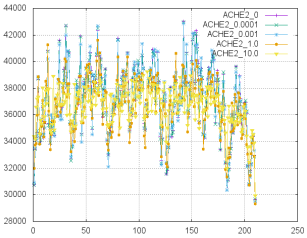


Figure D.21: RACHE in DTLZ1 with 5 objectives.

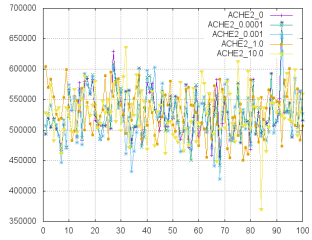


Figure D.22: RACHE in DTLZ3 with 2 objectives.

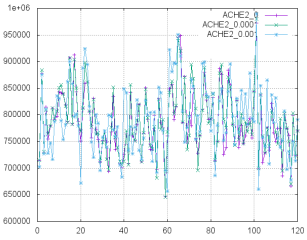


Figure D.23: RACHE in DTLZ3 with 3 objectives.

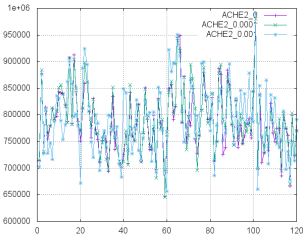


Figure D.24: RACHE in DTLZ3 with 5 objectives.

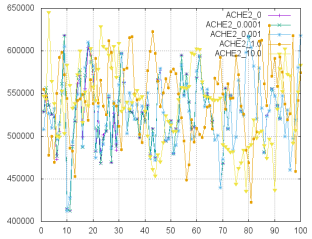


Figure D.25: RACHE in DTLZ3 convex with 2 objectives.

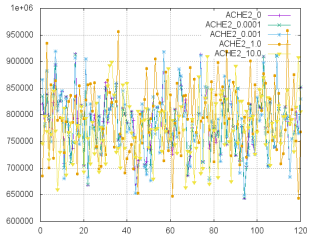


Figure D.26: RACHE in DTLZ3 convex with 3 objectives.

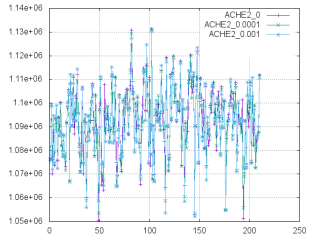


Figure D.27: RACHE in DTLZ3 convex with 5 objectives.

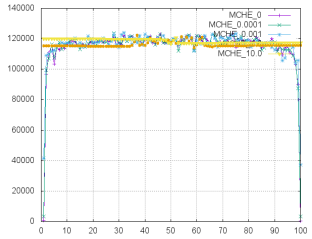


Figure D.28: MCHE in DTLZ1 with 2 objectives.

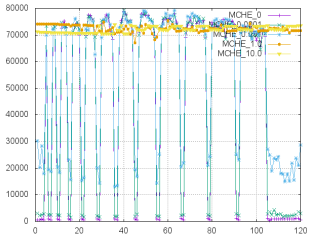


Figure D.29: MCHE in DTLZ1 with 3 objectives.

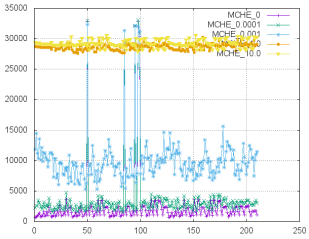


Figure D.30: MCHE in DTLZ1 with 5 objectives.

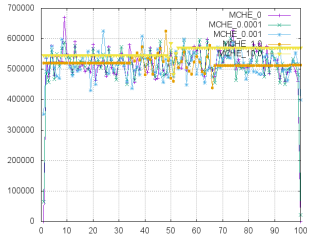


Figure D.31: MCHE in DTLZ3 with 2 objectives.

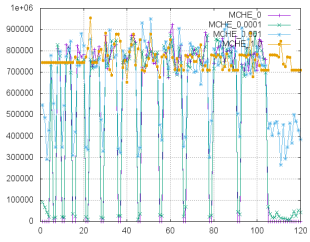


Figure D.32: MCHE in DTLZ3 with 3 objectives.

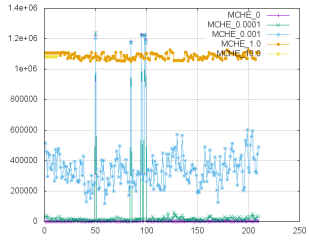


Figure D.33: MCHE in DTLZ3 with 5 objectives.

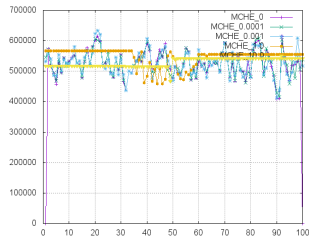


Figure D.34: MCHE in DTLZ3 convex with 2 objectives.

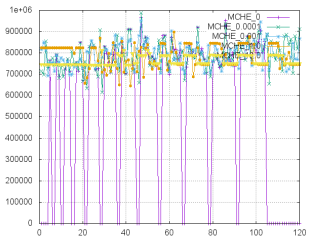


Figure D.35: MCHE in DTLZ3 convex with 3 objectives.

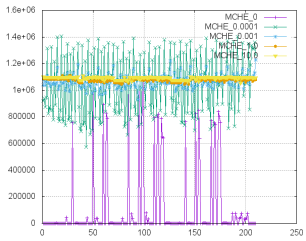


Figure D.36: MCHE in DTLZ3 convex with 5 objectives.

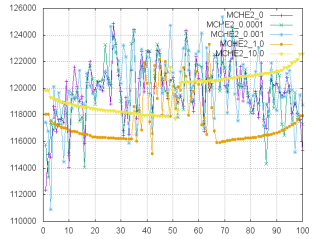


Figure D.37: RMCHE in DTLZ1 with 2 objectives.

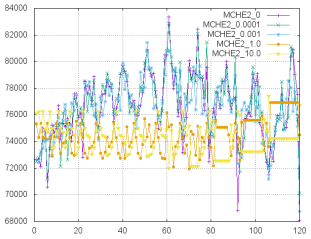


Figure D.38: RMCHE in DTLZ1 with 3 objectives.

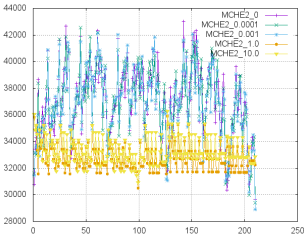


Figure D.39: RMCHE in DTLZ1 with 5 objectives.

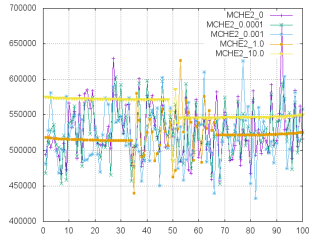


Figure D.40: RMCHE in DTLZ3 with 2 objectives.

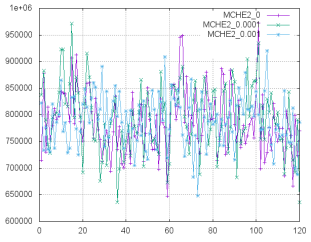


Figure D.41: RMCHE in DTLZ3 with 3 objectives.

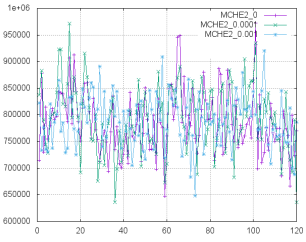


Figure D.42: RMCHE in DTLZ3 with 5 objectives.

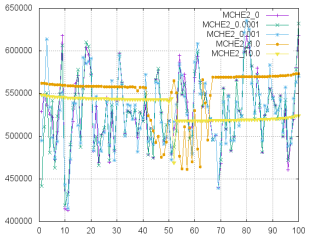


Figure D.43: RMCHE in DTLZ3 convex with 2 objectives.

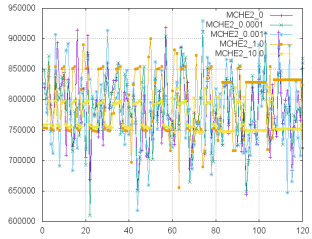


Figure D.44: RMCHE in DTLZ3 convex with 3 objectives.

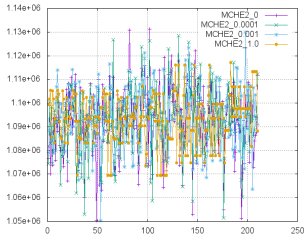


Figure D.45: RMCHE in DTLZ3 convex with 5 objectives.

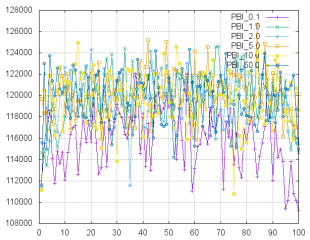


Figure D.46: PBI in DTLZ1 with 2 objectives.

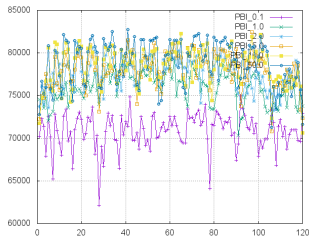


Figure D.47: PBI in DTLZ1 with 3 objectives.

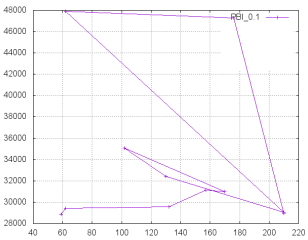


Figure D.48: PBI in DTLZ1 with 5 objectives.

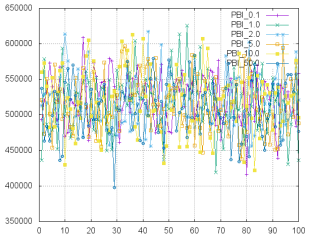


Figure D.49: PBI in DTLZ3 with 2 objectives.

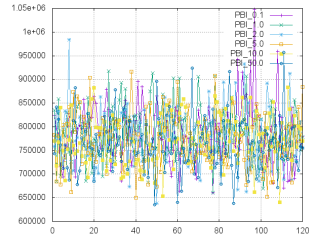


Figure D.50: PBI in DTLZ3 with 3 objectives.

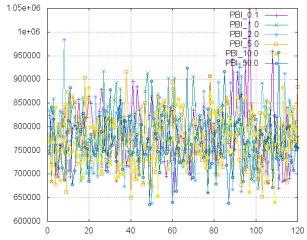


Figure D.51: PBI in DTLZ3 with 5 objectives.

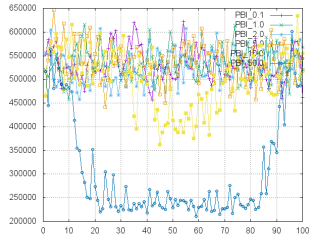


Figure D.52: PBI in DTLZ3 convex with 2 objectives.

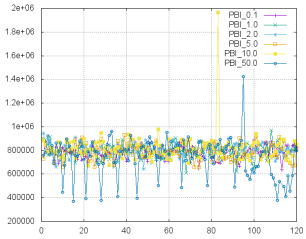


Figure D.53: PBI in DTLZ3 convex with 3 objectives.

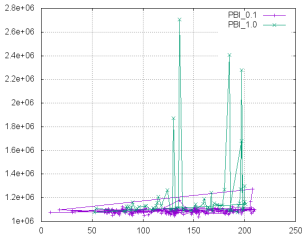


Figure D.54: PBI in DTLZ3 convex with 5 objectives.

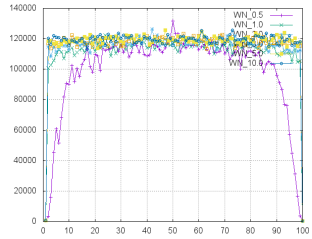


Figure D.55: WN in DTLZ1 with 2 objectives.

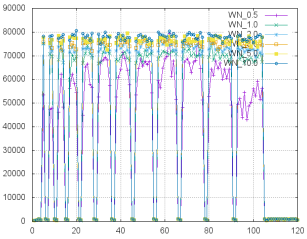


Figure D.56: WN in DTLZ1 with 3 objectives.

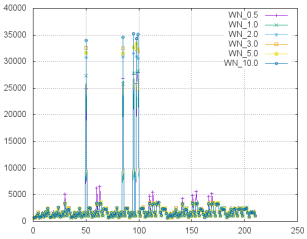


Figure D.57: WN in DTLZ1 with 5 objectives.

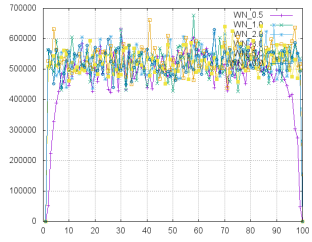


Figure D.58: WN in DTLZ3 with 2 objectives.

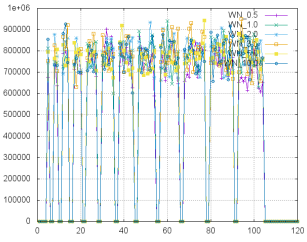


Figure D.59: WN in DTLZ3 with 3 objectives.

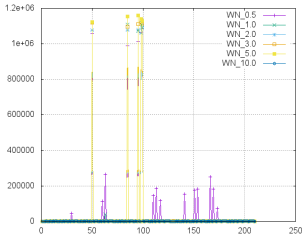


Figure D.60: WN in DTLZ3 with 5 objectives.

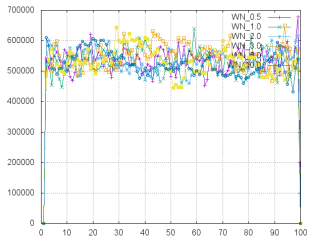


Figure D.61: WN in DTLZ3 convex with 2 objectives.

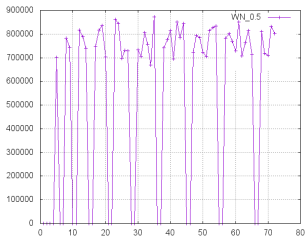


Figure D.62: WN in DTLZ3 convex with 3 objectives.

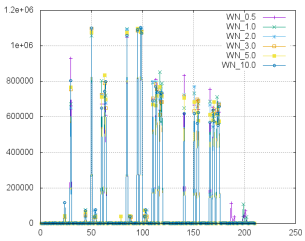


Figure D.63: WN in DTLZ3 convex with 5 objectives.

Bibliography

[1] Adenso-Diaz, B., and Laguna, M. Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. *Operations Research 54*, 1 (Jan. 2006), 99–114.

[2] Aleti, A., and Moser, I. Entropy-based Adaptive Range Parameter Control for Evolutionary Algorithms. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2013), GECCO '13, ACM, pp. 1501–1508.

[3] Angelo, J. S., and Barbosa, H. J. On Ant Colony Optimization Algorithms for Multiobjective Problems. In *Ant Colony Optimization – Methods and Applications*, A. Ostfeld, Ed. InTech, Cham, 2011, pp. 53–74.

[4] Athan, T. W., and Papalambros, P. Y. A Note on Weighted Criteria Methods for Compromise Solutions in Multi-objective Optimization. *Engineering Optimization 27*, 2 (1996), 155–176.

[5] Auger, A., Schoenauer, M., and Vanhaecke, N. LS-CMA-ES: A Second-Order Algorithm for Covariance Matrix Adaptation. In *Parallel Problem Solving from Nature - PPSN VIII* (Berlin, Heidelberg, 2004), X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, Eds., Springer Berlin Heidelberg, pp. 182–191.

[6] Balaprakash, P., Birattari, M., and Stützle, T. Improvement Strategies for the F-Race Algorithm: Sampling Design and Iterative Refinement. In *Hybrid Metaheuristics* (Berlin, Heidelberg, 2007), T. Bartz-Beielstein, M. J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, Eds., Springer Berlin Heidelberg, pp. 108–122.

[7] Bartz-Beielstein, T. *Experimental Research in Evolutionary Computation: The New Experimentalism (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] Beyer, H.-G., and Sendhoff, B. Robust optimization A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering 196*, 3334 (2007), 3190 – 3218.

[9] Bezerra, L. C. T., López-Ibáñez, M., and Stützle, T. Automatic Design of Evolutionary Algorithms for Multi-Objective Combinatorial Optimization. In *Parallel Problem Solving from Nature – PPSN XIII* (Cham, 2014), T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, Eds., Springer International Publishing, pp. 508–517.

[10] Bezerra, L. C. T., López-Ibáñez, M., and Stützle, T. Comparing Decomposition-Based and Automatically Component-Wise Designed Multi-Objective Evolutionary Algorithms. In *Evolutionary Multi-Criterion Optimization* (Cham, 2015), A. Gaspar-Cunha, C. Henggeler Antunes, and C. Coello Coello, Eds., Springer International Publishing, pp. 396–410.

[11] Birattari, M. *Tuning Metaheuristics: A Machine Learning Perspective*, 1st ed. 2005. 2nd printing ed. Springer Publishing Company, Incorporated, 2009.

[12] Birattari, M., Stützle, T., Paquete, L., and Varrentrapp, K. A Racing Algorithm for Configuring Metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference* (San Francisco, CA, USA, 2002), GECCO '02, Morgan Kaufmann Publishers Inc., pp. 11–18.

[13] Bjornson, E., Jorswieck, E. A., Debbah, M., and Ottersten, B. Multiobjective Signal Processing Optimization: The Way to Balance Conflicting Metrics in 5G Systems. *IEEE Signal Processing Magazine 31*, 6 (Nov 2014), 14–23.

[14] Böttcher, S., Doerr, B., and Neumann, F. Optimal Fixed and Adaptive Mutation Rates for the LeadingOnes Problem. In *Parallel Problem Solving from Nature, PPSN XI* (Berlin, Heidelberg, 2010), R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., Springer Berlin Heidelberg, pp. 1–10.

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

[16] Büche, D., Milano, M., and Koumoutsakos, P. Self-Organizing Maps for Multi-Objective Optimization. In *GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference* (2002), Morgan Kaufmann Publishers, pp. 152–155.

[17] Burtini, G., Loeppky, J. L., and Lawrence, R. A survey of online experiment design with the stochastic multi-armed bandit. *CoRR abs/1510.00757* (2015).

[18] Carpinelli, G., Caramia, P., Mottola, F., and Proto, D. Exponential Weighted Method and a Compromise Programming Method for Multi-objective Operation of plug-in Vehicle Aggregators in Microgrids. *International Journal of Electrical Power & Energy Systems 56* (2014), 374 – 384.

[19] Chen, C.-M., ping Chen, Y., and Zhang, Q. Enhancing MOEA/D with Guided Mutation and Priority Update for Multi-Objective Optimization. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)* (Trondheim, Norway, May 2009), IEEE Press, pp. 209–216.

[20] Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation 20*, 5 (October 2016), 773–791.

[21] Chiang, T. C., and Lai, Y. P. MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (June 2011), pp. 1473–1480.

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

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

[24] Corne, D. W., Knowles, J. D., and Oates, M. J. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In *Parallel Problem Solving from Nature PPSN VI* (Berlin, Heidelberg, 2000), M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds., Springer Berlin Heidelberg, pp. 839–848.

[25] Coy, S. P., Golden, B. L., Runger, G. C., and Wasil, E. A. Using Experimental Design to Find Effective Parameter Settings for Heuristics. *Journal of Heuristics 7*, 1 (Jan 2001), 77–97.

[26] Cui, L., Li, G., Lin, Q., Chen, J., and Lu, N. Adaptive Differential Evolution Algorithm with Novel Mutation Strategies in Multiple Sub-populations. *Computers & Operation Research 67*, C (Mar. 2016), 155–173.

[27] Dächert, K., Gorski, J., and Klamroth, K. An Augmented Weighted Tchebycheff Method with Adaptively Chosen Parameters for Discrete Bicriteria Optimization Problems. *Computers & Operation Research 39*, 12 (Dec. 2012), 2929–2943.

[28] Das, I., and Dennis, J. E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization 8*, 3 (1998), 631–657.

[29] Dean Angela, Voss Daniel, D. D. *Design and Analysis of Experiments*, 2nd ed. 2017 ed. Springer International Publishing, 2017.

[30] Deb, K., and Agrawal, R. B. Simulated Binary Crossover for Continuous Search Space. Tech. rep., Indian Institute of Technology, 1994.

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

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

[33] Deb K., M. K. A Review of Nadir Point Estimation Procedures Using Evolutionary Approaches: A Tale of Dimensionality Reduction. Tech. Rep. KanGAL Report 2008004, Indian Institute of Technology, 2008.

[34] Dellino, G., Fedele, M., and Meloni, C. Dynamic Objectives Aggregation Methods for Evolutionary Portfolio Optimisation. A Computational Study. *International Journal of Bio-Inspired Computation 4*, 4 (jul 2012), 258–270.

[35] Derbel, B., Brockhoff, D., and Liefooghe, A. Force-Based Cooperative Search Directions in Evolutionary Multi-objective Optimization. In *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds., vol. 7811. Lecture Notes in Computer Sciences, Sheffield, UK, March 19-22 2013, pp. 383–397.

[36] Derbel, B., Brockhoff, D., Liefooghe, A., and Verel, S. On the Impact of Multiobjective Scalarizing Functions. In *Parallel Problem Solving from Nature – PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings*, T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, Eds. Springer International Publishing, Cham, 2014, pp. 548–558.

[37] Dou, L., Zong, Q., Ji, Y., and Zeng, F. Adaptive multi-objective optimization based on feedback design. *Transactions of Tianjin University 16*, 5 (2010), 359–365.

[38] Eiben, A. E., Hinterding, R., and Michalewicz, Z. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation 3*, 2 (July 1999), 124–141.

[39] Eiben, A. E., and Smit, S. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation 1*, 1 (2011), 19 – 31.

[40] Eichfelder, G. An Adaptive Scalarization Method in Multiobjective Optimization. *SIAM Journal on Optimization 19*, 4 (Jan 2009), 1694–1718.

[41] Emmerich, M., Beume, N., and Naujoks, B. An EMO Algorithm Using the Hypervolume Measure As Selection Criterion. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization* (Berlin, Heidelberg, 2005), EMO'05, Springer-Verlag, pp. 62–76.

[42] Emmerich, M. T., and Deutz, A. H. Test Problems Based on Lamé Superspheres. In *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007* (Matshushima, Japan, March 2007), S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4033, Lecture Notes in Computer Sciences, pp. 922–936.

[43] Erozan, İ., Torkul, O., and Ustun, O. Proposal of a Nonlinear Multi-objective Genetic Algorithm Using Conic Scalarization to the Design of Cellular Manufacturing Systems. *Flexible Services and Manufacturing Journal 27*, 1 (2015), 30–57.

[44] Fonseca, C., and Fleming, P. Multiobjective Genetic Algorithms. In *IEE Colloquium on Genetic Algorithms for Control Systems Engineering* (1993), IEE, pp. 6/1–6/5.

[45] Gaspar-Cunha, A., and Covas, J. A. Robustness in multi-objective optimization using evolutionary algorithms. *Computational Optimization and Applications 39*, 1 (Jan 2008), 75–96.

[46] Giagkiozis, I., and Fleming, P. J. Methods for Multi-objective Optimization: An analysis. *Information Sciences 293* (February 1 2015), 338–350.

[47] Giagkiozis, I., Purshouse, R., and Fleming, P. Generalized decomposition and cross entropy methods for many-objective optimization. *Information Sciences 282* (2014), 363 – 387.

[48] Giagkiozis, I., Purshouse, R. C., and Fleming, P. J. Generalized Decomposition. In *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds. Springer. Lecture Notes in Computer Science Vol. 7811, Sheffield, UK, March 19-22 2013, pp. 428–442.

[49] Giagkiozis, I., Purshouse, R. C., and Fleming, P. J. Towards Understanding the Cost of Adaptation in Decomposition-Based Optimization Algorithms. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (Oct 2013), pp. 615–620.

[50] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[51] Gómez, R. H. *Parallel Hyper-Heuristics for Multi-Objective Optimization.* PhD thesis, CINVESTAV-IPN, Mexico City, March 2018.

[52] Gómez, R. H., and Coello, C. A. C. A Hyper-heuristic of Scalarizing Functions. In *Proceedings of the Genetic and Evolutionary Computation Conference* (New York, NY, USA, 2017), GECCO '17, ACM, pp. 577–584.

[53] Gorissen, B. L., hsan Yankolu, and den Hertog, D. A practical guide to robust optimization. *Omega 53* (2015), 124 – 137.

[54] Grauer, M. A Dynamic Interactive Decision Analysis and Support System (DIDASS). Users Guide (May 1983). IIASA Working Paper, -, IIASA, Laxenburg, Austria, June 1983.

[55] Grefenstette, J. J. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics 16*, 1 (Jan 1986), 122–128.

[56] Guerrero, J. L., Garcia, J., Marti, L., Molina, J. M., and Berlanga, A. A Stopping Criterion Based on Kalman Estimation Techniques with Several Progress Indicators. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2009), GECCO '09, ACM, pp. 587–594.

[57] Hadka, D., and Reed, P. Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework. *Evolutionary Computation 21* (2013), 231–259.

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

[59] Hansen, N., and Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation* (May 1996), pp. 312–317.

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

[61] Hernández Gómez, R., and Coello Coello, C. A. Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2015), GECCO '15, ACM, pp. 679–686.

[62] Herrera, F., and Lozano, M. Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers. In *Genetic Algorithms and Soft Computing* (1996), Physica-Verlag, pp. 95–125.

[63] Hoos, H. H. Automated Algorithm Configuration and Parameter Tuning. In *Autonomous Search*, Y. Hamadi, E. Monfroy, and F. Saubion, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 37–71.

[64] Hsieh, C.-T., Chen, C.-M., and Chen, Y.-p. Particle Swarm Guided Evolution Strategy. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2007), GECCO '07, ACM, pp. 650–657.

[65] Huang, W., and Li, H. On the differential evolution schemes in MOEA/D. In *2010 Sixth International Conference on Natural Computation* (Aug 2010), vol. 6, pp. 2788–2792.

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

[67] Hughes, E. J. Multiple Single Objective Pareto Sampling. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)* (Canberra, Australia, December 2003), vol. 4, IEEE Press, pp. 2678–2684.

[68] Hughes, E. J. MSOPS-II: A General-Purpose Many-Objective Optimiser. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)* (Singapore, September 2007), IEEE Press, pp. 3944–3951.

[69] Hutter, F., Hoos, H. H., Leyton-Brown, K., and Stützle, T. ParamILS: An Automatic Algorithm Configuration Framework. *Journal of Artificial Intelligence Research 36*, 1 (Sept. 2009), 267–306.

[70] Hutter, F., Hoos, H. H., and Stützle, T. Automatic Algorithm Configuration Based on Local Search. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2* (2007), AAAI'07, AAAI Press, pp. 1152–1157.

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

[72] Igel, C., Suttorp, T., and Hansen, N. Steady-State Selection and Efficient Covariance Matrix Update in the Multi-objective CMA-ES. In *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata,

Eds., vol. 4403 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 171–185.

[73] Ishibuchi, H., Akedo, N., and Nojima, Y. A Study on the Specification of a Scalarizing Function in MOEA/D for Many-Objective Knapsack Problems. In *Learning and Intelligent Optimization, 7th International Conference, LION 7*, G. Nicosia and P. Pardalos, Eds., vol. 7997. Lecture Notes in Computer Sciences, Catania, Italy, January 7-11 2013, pp. 231–246.

[74] Ishibuchi, H., Akedo, N., and Nojima, Y. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. *IEEE Transactions on Evolutionary Computation 19*, 2 (April 2015), 264–283.

[75] Ishibuchi, H., Doi, K., Masuda, H., and Nojima, Y. Relation Between Weight Vectors and Solutions in MOEA/D. In *2015 IEEE Symposium Series on Computational Intelligence* (Dec 2015), pp. 861–868.

[76] Ishibuchi, H., Doi, K., and Nojima, Y. Use of Piecewise Linear and Nonlinear Scalarizing Functions in MOEA/D. In *Parallel Problem Solving from Nature – PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter, Eds. Springer International Publishing, Cham, 2016, pp. 503–513.

[77] Ishibuchi, H., Masuda, H., Tanigaki, Y., and Nojima, Y. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In *Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 –April 1, 2015. Proceedings, Part II*, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Springer International Publishing, Cham, 2015, pp. 110–125.

[78] Ishibuchi, H., and Nojima, Y. Optimization of Scalarizing Functions Through Evolutionary Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007* (Matshushima, Japan, March 2007), S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4403, Lecture Notes in Computer Sciences, pp. 51–65.

[79] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. Adaptation of Scalarizing Functions in MOEA/D: An Adaptive Scalarizing Function-Based Multiobjective Evolutionary Algorithm. In *Evolutionary Multi-Criterion Optimization: 5th International Conference, EMO 2009, Nantes, France, April 7-10, 2009. Proceedings*, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 438–452.

[80] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. Simultaneous Use of Different Scalarizing Functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2010), GECCO '10, ACM, pp. 519–526.

[81] Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Transactions on Evolutionary Computation 21*, 2 (Apr. 2017), 169–190.

[82] Jain, H., and Deb, K. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation 18*, 4 (August 2014), 602–622.

[83] Jariyatantiwait, C., and Yen, G. G. Multiobjective Differential Evolution Based on Fuzzy Performance Feedback. *International Journal on Swarm Intelligence Research 5*, 4 (Oct. 2014), 45–64.

[84] Jin, Y., Okabe, T., and Sendho, B. Adapting Weighted Aggregation for Multiobjective Evolution Strategies. In *Evolutionary Multi-Criterion Optimization* (Berlin, Heidelberg, 2001), E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, Eds., Springer Berlin Heidelberg, pp. 96–110.

[85] Jr, V. B. On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-criteria Objectives. In *Multiple Criteria Decision Making: Proceedings of a Conference Jouy-en-Josas, France May 21–23, 1975*, H. Thiriez and S. Zionts, Eds., vol. 130. Springer, Berlin, Germany, 1973, pp. 76–86.

[86] Kaliszewski, I. A modified weighted Tchebycheff metric for multiple objective programming. *Computer & Operations Research 14*, 4 (1987), 315–323.

[87] Kang, Q., Song, X., Zhou, M., and Li, L. A Collaborative Resource Allocation Strategy for Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2018), 1–8.

[88] Karafotias, G., Hoogendoorn, M., and Eiben, A. E. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE Transactions on Evolutionary Computation 19*, 2 (April 2015), 167–187.

[89] Kasimbeyli, R. A Nonlinear Cone Separation Theorem and Scalarization in Nonconvex Vector Optimization. *SIAM Journal on Optimization 20*, 3 (Jan 2010), 1591–1619.

[90] Kasimbeyli, R., Ozturk, Z. K., Kasimbeyli, N., Yalcin, G. D., and Icmen, B. Conic Scalarization Method in Multiobjective Optimization and Relations with Other Scalarization Methods. In *Modelling, Computation and Optimization in Information Systems and Management Sciences: Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences - MCO 2015 - Part I*, H. A. L. Thi, T. P. Dinh, and N. T. Nguyen, Eds. Springer, Cham, Switzerland, 2015, pp. 319–329.

[91] Knowles, J., and Corne, D. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* (July 1999), vol. 1, pp. 98–105 Vol. 1.

[92] Kohonen, T., Schroeder, M. R., and Huang, T. S., Eds. *Self-Organizing Maps*, 3rd ed. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[93] Kollat, J. B., and Reed, P. M. The Value of Online Adaptive Search: A Performance Comparison of NSGAII, ϵ-NSGAII and ϵMOEA. In *Proceedings of the Third*

International Conference on Evolutionary Multi-Criterion Optimization (Berlin, Heidelberg, 2005), EMO'05, Springer-Verlag, pp. 386–398.

[94] Kramer, O. Evolutionary self-adaptation: a survey of operators and strategy parameters. *Evolutionary Intelligence 3*, 2 (Aug 2010), 51–65.

[95] Krimpmann, C., Braun, J., Hoffmann, F., and Bertram, T. Active covariance matrix adaptation for multi-objective CMA-ES. In *2013 Sixth International Conference on Advanced Computational Intelligence (ICACI'2013),* (Oct 2013), pp. 189–194.

[96] Kuleshov, V., and Precup, D. Algorithms for multi-armed bandit problems. *CoRR abs/1402.6028* (2014).

[97] Lara, A., Alvarado, S., Salomon, S., Avigad, G., Coello Coello, C. A., and Schütze, O. The Gradient Free Directed Search Method as Local Search within Multi-Objective Evolutionary Algorithms. In *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, O. Schütze, C. A. Coello Coello, A.-A. Tantar, E. Tantar, P. Bouvry, P. Del Moral, and P. Legrand, Eds. Springer, Advances in Intelligent Systems and Computing Vol. 175, Berlin, Germany, 2012, pp. 153–168. ISBN 978-3-642-31519-0.

[98] Lee, L. H., Chew, E. P., Yu, Q., Li, H., and Liu, Y. A Study on Multi-objective Particle Swarm Optimization with Weighted Scalarizing Functions. In *Proceedings of the 2014 Winter Simulation Conference* (Piscataway, NJ, USA, 2014), WSC '14, IEEE Press, pp. 3718–3729.

[99] Li, C., Yang, S., Nguyen, T. T., Yu, E. L., Yao, X., Jin, Y., g. Beyer, H., and Suganthan, P. N. Benchmark Generator for CEC2009 Competition on Dynamic Optimization, 2008.

[100] Li, D. Convexification of a Noninferior Frontier. *Journal of Optimization Theory and Applications 88*, 1 (1996), 177–196.

[101] Li, D., Yang, J.-B., and Biswal, M. Quantitative Parametric Connections Between Methods for Generating Noninferior Solutions in Multiobjective Optimization. *European Journal of Operational Research 117*, 1 (1999), 84–99.

[102] Li, H., and Zhang, Q. Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation 13*, 2 (April 2009), 284–302.

[103] Li, K., Fialho, ., Kwong, S., and Zhang, Q. Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation 18*, 1 (Feb 2014), 114–130.

[104] Liu, H. L., Chen, L., Zhang, Q., and Deb, K. An evolutionary many-objective optimisation algorithm with adaptive region decomposition. In *2016 IEEE Congress on Evolutionary Computation (CEC'2016)* (July 2016), pp. 4763–4769.

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

[106] Maron, O., and Moore, A. W. The Racing Algorithm: Model Selection for Lazy Learners. *Artificial Intelligence Review 11*, 1 (Feb 1997), 193–225.

[107] Mashwani, W. K., Salhi, A., jan, M. A., Khanum, R. A., and Sulaiman, M. Enhanced Version of Multi-algorithm Genetically Adaptive for Multiobjective optimization. *International Journal of Advanced Computer Science and Applications 6*, 12 (2015).

[108] Menchaca-Mendez, A., Montero, E., Riff, M.-C., and Coello Coello, C. A. A More Efficient Selection Scheme in iSMS-EMOA. In *Advances in Artificial Intelligence – IBERAMIA 2014: 14th Ibero-American Conference on AI, Santiago de Chile, Chile, November 24-27, 2014, Proceedings*, A. L. Bazzan and K. Pichara, Eds. Springer International Publishing, Cham, 2014, pp. 371–380.

[109] Miettinen, K. *Nonlinear Multiobjective Optimization*, vol. 12. International Series in Operations Research & Management Science, 1998.

[110] Mohammadi, A., Omidvar, M. N., Li, X., and Deb, K. Sensitivity analysis of Penalty-based Boundary Intersection on aggregation-based EMO algorithms. In *2015 IEEE Congress on Evolutionary Computation (CEC'2015)* (May 2015), pp. 2891–2898.

[111] Montero, E., and Riff, M.-C. Towards a Method for Automatic Algorithm Configuration: A Design Evaluation Using Tuners. In *Parallel Problem Solving from Nature – PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings*, T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, Eds. Springer International Publishing, Cham, 2014, pp. 90–99.

[112] Montgomery, D. C. *Design and Analysis of Experiments*. Student solutions manual. John Wiley & Sons, USA, 2008.

[113] Morse, J. Reducing the size of the nondominated set: Pruning by clustering. *Computers and Operations Research 7*, 1 (1980), 55–66.

[114] Murata, T., and Taki, A. Many-Objective Optimization for Knapsack Problems Using Correlation-Based Weighted Sum Approach. In *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Lecture Notes in Computer Science, Nantes, France, April 2009, pp. 468–480.

[115] Nannen, V., and Eiben, A. A Method for Parameter Calibration and Relevance Estimation in Evolutionary Algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2006), GECCO '06, ACM, pp. 183–190.

[116] Nannen, V., and Eiben, A. E. Relevance Estimation and Value Calibration of Evolutionary Algorithm Parameters. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (San Francisco, CA, USA, 2007), IJCAI'07, Morgan Kaufmann Publishers Inc., pp. 975–980.

[117] no, A. A. M., Coello, C. A. C., and Mezura-Montes, E. MODE-LD+SS: A Novel Differential Evolution Algorithm Incorporating Local Dominance and Scalar Selection Mechanisms for Multi-Objective Optimization. In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)* (Barcelona, Spain, July 18–23 2010), IEEE Press, pp. 3284–3291.

[118] Ostermeier, A., Gawelczyk, A., and Hansen, N. Step-size adaptation based on non-local use of selection information. In *Parallel Problem Solving from Nature — PPSN III* (Berlin, Heidelberg, 1994), Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., Springer Berlin Heidelberg, pp. 189–198.

[119] Pacula, M., Ansel, J., Amarasinghe, S., and O'Reilly, U.-M. Hyperparameter Tuning in Bandit-Based Adaptive Operator Selection. In *Applications of Evolutionary Computing, EvoApplications2012: EvoCOMNET, EvoCOMPLEX, EvoFIN, EvoGAMES* (Malaga, Spain, 11-13 April 2012), C. Di Chio, A. Agapitos, S. Cagnoni, C. Cotta, F. Fernandez de Vega, G. A. Di Caro, and R. Drechsler, Eds., vol. 7248 of *LNCS*, Springer Verlag, pp. 71–80.

[120] Palar, P. S., Tsuchiya, T., and Parks, G. Comparison of scalarization functions within a local surrogate assisted multi-objective memetic algorithm framework for expensive problems. In *2015 IEEE Congress on Evolutionary Computation (CEC'2015)* (May 2015), pp. 862–869.

[121] Paquete, L., and Stützle, T. A Two-Phase Local Search for the Biobjective Traveling Salesman Problem. In *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003* (Faro, Portugal, April 2003), C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds., vol. 2632, Lecture Notes in Computer Sciences, pp. 479–493.

[122] Pereira, I., and Madureira, A. Racing based approach for Metaheuristics parameter tuning. In *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)* (June 2015), pp. 1–6.

[123] Pescador-Rojas, M., and Coello Coello, C. A. Studying the effect of techniques to generate reference vectors in many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2018, Kyoto, Japan, July 15-19, 2018* (2018), pp. 193–194.

[124] Pescador-Rojas, M., Hernández Gómez, R., Montero, E., Rojas-Morales, N., Riff, M.-C., and Coello Coello, C. A. An Overview of Weighted and Unconstrained Scalarizing Functions. In *Evolutionary Multi-Criterion Optimization: 9th International Conference, EMO 2017, Münster, Germany, March 19-22, 2017, Proceedings*, H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. Wiecek, Y. Jin, and C. Grimme, Eds. Springer International Publishing, Cham, 2017, pp. 499–513.

[125] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, New York, NY, USA, 1986.

[126] Qi, Y., Bao, L., Ma, X., Miao, Q., and Li, X. Self-adaptive Multi-objective Evolutionary Algorithm Based on Decomposition for Large-scale Problems. *Information Sciences 367*, C (Nov. 2016), 529–549.

[127] Qin, A. K., and Suganthan, P. N. Self-adaptive differential evolution algorithm for numerical optimization. In *2005 IEEE Congress on Evolutionary Computation* (Sep. 2005), vol. 2, pp. 1785–1791.

[128] Qiu, X., Xu, W., Xu, J.-X., and Tan, K. C. A New Framework for Self-adapting Control Parameters in Multi-objective Optimization. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference* (New York, NY, USA, 2015), GECCO '15, ACM, pp. 743–750.

[129] Ralphs, T. K., Saltzman, M. J., and Wiecek, M. M. An Improved Algorithm for Solving Biobjective Integer Programs. *Annals of Operations Research 147*, 1 (2006), 43–70.

[130] Rechenberg, I. *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.

[131] Ridge, E., and Kudenko, D. Analyzing Heuristic Performance with Response Surface Models: Prediction, Optimization and Robustness. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2007), GECCO '07, ACM, pp. 150–157.

[132] Riff, M. C., and Montero, E. A New Algorithm for Reducing Metaheuristic Design Effort. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)* (June 2013), pp. 3283–3290.

[133] Roudenko, O., and Schoenauer, M. A Steady Performance Stopping Criterion for Pareto-based Evolutionary Algorithms. In *In Proceedings of the 6th International Multi-Objective Programming and Goal Programming Conference* (2004), pp. 14–16.

[134] Ruiz, A. B., Saborido, R., and Luque, M. A Preference-based Evolutionary Algorithm for Multiobjective Optimization: The Weighting Achievement Scalarizing Function Genetic Algorithm. *Journal of Global Optimization 62*, 1 (may 2015), 101–129.

[135] Santiago, A., Huacuja, H. J. F., Dorronsoro, B., Pecero, J. E., Santillan, C. G., Barbosa, J. J. G., and Monterrubio, J. C. S. A Survey of Decomposition Methods for Multi-objective Optimization. In *Recent Advances on Hybrid Approaches for Designing Intelligent Systems* (2014), O. Castillo, P. Melin, W. Pedrycz, and J. Kacprzyk, Eds., Springer, pp. 453–465. ISBN 978-3-319-05170-3.

[136] Sato, H. Inverted PBI in MOEA/D and its Impact on the Search Performance on Multi and Many-Objective Optimization. In *2014 Genetic and Evolutionary Computation Conference (GECCO 2014)* (Vancouver, Canada, July 12-16 2014), ACM Press, pp. 645–652. ISBN 978-1-4503-2662-9.

[137] Sato, H. Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs. *Journal of Heuristics 21*, 6 (2015), 819–849.

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

[139] Scheffé, H. Experiments with Mixtures. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 20* (1958), 344360.

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

[141] Shi, J., Gong, M., Ma, W., and Jiao, L. A Multipopulation Coevolutionary Strategy for Multiobjective Immune Algorithm. *The Scientific World Journal 2014* (Feb 2013).

[142] Smith, J. E. Self-Adaptation in Evolutionary Algorithms for Combinatorial Optimisation. In *Adaptive and Multilevel Metaheuristics*, C. Cotta, M. Sevaux, and K. Sörensen, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 31–57.

[143] Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, New York, 1986.

[144] Steuer, R. E., and Choo, E.-U. An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming. *Mathematical Programming 26*, 3 (1983), 326–344.

[145] Storn, R., and Price, K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization 11*, 4 (Dec. 1997), 341–359.

[146] Sutton, R. S., and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[147] Tan, K., Lee, T., and Khor, E. Evolutionary Algorithms with Dynamic Population Size and Local Exploration for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation 5*, 6 (December 2001), 565–588.

[148] Toral, R. The volume enclosed by an n-dimensional Lamé curve. Tech. rep., CSIC-UIB, Campus UIB, E-07122 Palma de Mallorca, Spain, 11 2013.

[149] Toscano Pulido, G., and Coello Coello, C. A. The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization*, C. Fonseca, P. Fleming, E. Zitzler, L. Thiele, and K. Deb, Eds., vol. 2632 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2003, pp. 252–266.

[150] Triantaphyllou, E. *Multi-criteria Decision Making Methods: A Comparative Study*, vol. 44. Springer USA, 2000.

[151] Tutum, C. C., and Deb, K. A Multimodal Approach for Evolutionary Multiobjective Optimization (MEMO): Proof-of-Principle Results. In *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, A. Gaspar-Cunha,

C. H. Antunes, and C. Coello Coello, Eds. Springer. Lecture Notes in Computer Science Vol. 9018, Guimarães, Portugal, March 29 - April 1 2015, pp. 3–18.

[152] Unveren, A., and Acan, A. Multi-objective optimization with cross entropy method: Stochastic learning with clustered pareto fronts. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007).* (Sept 2007), pp. 3065–3071.

[153] Van Moffaert, K., Drugan, M. M., and Nowé, A. Hypervolume-Based Multi-Objective Reinforcement Learning. In *Evolutionary Multi-Criterion Optimization* (Berlin, Heidelberg, 2013), R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, Eds., Springer Berlin Heidelberg, pp. 352–366.

[154] Veldhuizen, D. A. V., and Lamont, G. B. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Tech. Rep. TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA, 1998.

[155] Venske, S. M., Goncalves, R. A., and Delgado, M. R. ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm. *Neurocomputing 127* (March 15 2014), 65–77.

[156] Vrugt, J., Robinson, B., and Hyman, J. Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces. *IEEE Transactions on Evolutionary Computation, IEEE Transactions on 13*, 2 (April 2009), 243–259.

[157] Vrugt, J. A., and Robinson, B. A. Improved evolutionary optimization from genetically adaptive multimethod search. *Proceedings of the National Academy of Sciences 104*, 3 (2007), 708–711.

[158] Wagner, T., Trautmann, H., and Martí, L. A Taxonomy of Online Stopping Criteria for Multi-objective Evolutionary Algorithms. In *Proceedings of the 6th International Conference on Evolutionary Multi-criterion Optimization* (Berlin, Heidelberg, 2011), EMO'11, Springer-Verlag, pp. 16–30.

[159] Wang, R., Zhang, Q., and Zhang, T. Pareto Adaptive Scalarising Functions for Decomposition Based Algorithms. In *Evolutionary Multi-Criterion Optimization: 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 –April 1, 2015. Proceedings, Part I*, A. Gaspar-Cunha, C. Henggeler Antunes, and C. Coello Coello, Eds. Springer International Publishing, Cham, 2015, pp. 248–262.

[160] Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., and Zhang, T. Localized Weighted Sum Method for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation 22*, 1 (Feb 2018), 3–18.

[161] Wang, Z., Zhang, Q., Li, H., Ishibuchi, H., and Jiao, L. On the use of two reference points in decomposition based multiobjective evolutionary algorithms. *Swarm and Evolutionary Computation 34* (2017), 89 – 102.

[162] Wierzbicki, A. P. The Use of Reference Objectives in Multiobjective Optimization. In *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter*, G. Fandel and T. Gal, Eds. Springer, Berlin, Germany, August 20-24 1980, pp. 468–486.

[163] Wolpert, D. H., and Macready, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation 1*, 1 (Apr 1997), 67–82.

[164] Yang, S., Jiang, S., and Jiang, Y. Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes. *Soft Computing* (2016), 1–15.

[165] Yuan, Y., Xu, H., Wang, B., Zhang, B., and Yao, X. Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers. *IEEE Transactions on Evolutionary Computation 20*, 2 (April 2016), 180–198.

[166] Zadeh, L. Optimality and Non-scalar-valued Performance Criteria. *IEEE Transactions on Automatic Control 8*, 1 (Jan 1963), 59–60.

[167] Zapotecas Martínez, S., and Coello Coello, C. A. A Direct Local Search Mechanism for Decomposition-based Multi-Objective Evolutionary Algorithms. In *2012 IEEE Congress on Evolutionary Computation (CEC'2012)* (Brisbane, Australia, June 10-15 2012), IEEE Press, pp. 3431–3438.

[168] Zeleny, M. Compromise Programming. In *Multiple Criteria Decision Making*, L. Cochrane and M. Zeleny, Eds. University of South Carolina Press, Columbia, USA, 1973, pp. 262–301.

[169] Zhang, J., Chen, W.-N., Zhan, Z.-H., Yu, W.-J., Li, Y.-L., Chen, N., and Zhou, Q. A survey on algorithm adaptation in evolutionary computation. *Frontiers of Electrical and Electronic Engineering 7*, 1 (March 2012), 16–31.

[170] Zhang, J., and Sanderson, A. C. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionar Computation 13*, 5 (Oct. 2009), 945–958.

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

[172] Zhang, Q., Liu, W., and Li, H. The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances. In *2009 IEEE Congress on Evolutionary Computation (CEC'2009)* (Trondheim, Norway, May 2009), IEEE Press, pp. 203–208.

[173] Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. Multiobjective optimization test instances for the CEC 2009 special session and competition. Tech. Rep. CES-487, University of Essex and Nanyang Technological University, 2008.

[174] Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. Decomposition-Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes. *IEEE Transactions on Evolutionary Computation 16*, 3 (June 2012), 442–446.

[175] Zhou, A., and Zhang, Q. Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation 20*, 1 (Feb 2016), 52–64.

[176] Zhou, L. A Survey on Contextual Multi-armed Bandits. *CoRR abs/1508.03326* (2015).

[177] Zitzler, E. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

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

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

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

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