Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

"Nuevos Esquemas de Selección para Algoritmos Evolutivos Multiobjetivo"

T E S I S

Que presenta

Diana Cristina Valencia Rodríguez

Para obtener el grado de

Doctora en Ciencias en Computación

Director de la tesis:

Dr. Carlos Artemio Coello Coello

Ciudad de México Diciembre, 2023

Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico Nacional

Zacatenco Campus

Computer Science Department

"Novel Selection Schemes for Multi-Objective Evolutionary Algorithms"

T H E S I S

Submitted by

Diana Cristina Valencia Rodríguez

as the fulfillment of the requirement for the degree of

Ph.D. in Computer Science

Advisor:

Dr. Carlos Artemio Coello Coello

Mexico City December, 2023

Resumen

En la industria, normalmente encontramos problemas donde la meta es optimizar simultáneamente dos o más funciones objetivo que usualmente están en conflicto entre sí, tales como minimizar el tiempo y el costo de producción. A éstos se les conoce como Problemas de Optimización Multi-objetivo (POMs). Diversas técnicas de programación matemática han sido propuestas para resolver esta clase de problemas. Sin embargo, dichas técnicas no se desempeñan bien en instancias con muchos óptimos locales y usualmente están diseñadas para POMs con características particulares.

Los Algoritmos Evolutivos Multi-Objetivo (AEMOs) son una alternativa que se ha vuelto popular en años recientes para resolver POMs. Los AEMOs operan con soluciones potenciales (llamadas individuos) que exploran el espacio de búsqueda usando operadores inspirados en la evolución natural (selección, recombinación y mutación).

La mayor parte de los AEMOs pueden clasificarse en tres categorías de acuerdo a su mecanismo de selección: basados en Pareto, basados en indicadores y basados en descomposición. A pesar de que estos mecanismos son predominantes en la literatura especializada, tienen varias desventajas como un desempeño pobre o costoso en problemas con muchos objetivos o sensibilidad a los parámetros adoptados. Por lo tanto, hay una necesidad de crear mecanismos de selección más eficaces que puedan superar estas limitaciones y proporcionar soluciones robustas y eficientes.

En esta tesis se proponen dos nuevos mecanismos de selección alternativos que no se encuentran en ninguna de las tres clases indicadas previamente. Estos mecanismos se incorporan en AEMOs modernos y los algoritmos resultantes son validados con respecto a AEMOs del estado del arte. Los resultados experimentales obtenidos de dicha comparación indican que los mecanismos propuestos permiten resolver problemas con una amplia variedad de características y con más de tres objetivos.

Empleando el conocimiento obtenido de los mecanismos de selección propuestos, se proponen posteriormente dos nuevos indicadores para evaluar el desempeño de los AEMOs. El análisis experimental de dichos indicadores muestra que son capaces de evaluar correctamente algunas características cruciales de las aproximaciones generados por un AEMO.

Abstract

In industry, we normally face problems in which the goal is to simultaneously optimize two or more objective functions which are often in conflict with each other, such as minizing both production time and cost. These are the so-called Multi-objective Optimization Problems (MOPs). A variety of mathematical programming techniques have been proposed to solve such problems. However, such techniques do not have a good performance in instances with many local optimal and are usually designed to deal with MOPs having particular features.

Multi-Objective Evolutionary Algorithms (MOEAs) are a choice for solving MOPs that has become increasingly popular in recent years. MOEAs operate with potential solutions (called individuals) which explore the search space using operators inspired on natural evolution (selection, recombination and mutation).

Most MOEAs can be classified into one of the three following categories based on their selection mechanism: Pareto-based, indicator-based and decomposition-based. Although these mechanisms are predominantly used in the specialized literature, they have several disadvantages, such as a poor or computationally expensive performance in problems having many objectives or high sensitivity to the values of their parameters. Therefore, there is a need to propose new selection mechanisms with a higher efficacy and that can overcome these limitations, thus providing robust and efficient solutions.

In this thesis, we propose two new alternative selection mechanisms that do not belong to any of the three previously indicated classes. Such mechanisms are incorporated into modern MOEAs and the resulting algorithms are validated with respect to state-of-the-art MOEAs. Our experimental results indicate that the proposed mechanisms can properly deal with a wide variety of features and with problems having more than three objectives.

Based on the knowledge obtained from the proposed selection mechanisms, we also propose two new quality indicators to assess performance of MOEAs. The experimental analysis of such indicators shows that they can correctly assess some crucial features of the approximations generated by an MOEA.

Agradecimientos

Primero que nada, quiero agradecer a mi esposo, Ramsés Martínez por todo su apoyo y amor a lo largo de este camino. Así mismo, quiero agradecer a mis padres (Rocio Rodríguez y Daniel Valencia) por impulsarme desde pequeña a seguir mis metas y apoyarme para continuar con mis estudios.

Agradezco al Dr. Carlos Coello Coello por su guía y paciencia durante estos cuatro años. Sin duda, sus enseñanzas me permitieron desarrollarme como investigadora y como persona.

Agradezco al comité sinodal conformado por la Dra. Adriana Lara, el Dr. Luis Gerardo de la Fraga, el Dr. Amilcar Menses, y el Dr. Antonio López, cuyos comentarios ayudaron a mejorar el contenido de esta tesis.

Agradezco al CONACyT y al CINVESTAV por los apoyos económicos otorgados para la realización de esta tesis doctoral.

Esta tesis fue derivada del proyecto titulado "Esquemas de Selección Alternativos para Algoritmos Evolutivos Multi-Objetivo" (Ref. 1920, Convocatoria Fronteras de la Ciencia 2016 de CONACyT). El Responsable Técnico de este proyecto es el Dr. Carlos A. Coello Coello.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Multi-objective optimization refers to minimizing or maximizing two or more objective functions that are usually in conflict (i.e., improving one objective causes deterioration of another). Due to this conflict, the goal of a multi-objective optimization problem (MOP) is to find the solutions that offer the best compromises among objective functions. The set of solutions representing these compromises is called the *Pareto Optimal set*, and its image (i.e., the corresponding objective function values) is called the *Pareto Optimal Front.*

We can find numerous MOPs in the real world. For example, when designing a product line, we seek to minimize the time and cost of production. Alternatively, when planning water networks, we aim to minimize contaminants and freshwater use [2]. Plenty of mathematical programming techniques have been proposed to solve MOPs. However, most techniques can only find one Pareto-optimal solution that usually depends on the chosen starting point. Moreover, they do not perform well on problems with multiple local optima or are designed for problems with particular characteristics, e.g., problems with linear or quadratic objective functions [3].

Multi-Objective Evolutionary Algorithms (MOEAs) are powerful alternative techniques to solve MOPs. MOEAs are population-based approaches that try to find optimal solutions by mimicking natural evolution. A general MOEA operates with a set of *individuals* (called *population*) representing potential solutions to the given problem. The population moves through the search space by employing evolutionary operators such as selection, recombination, and mutation. MOEAs have become popular in recent years due to their ability to explore large and hard-to-navigate search spaces, to their ease of use, and to the lack of need for gradient information. Furthermore, MOEAs can generate several elements of the Pareto Optimal Set in a single run.

Most MOEAs are classified into three categories according to their selection mechanism [4]: Pareto-based, indicator-based, and decomposition-based approaches. Pareto-based algorithms incorporate the concept of Pareto dominance in their selection

process and are one of the most common types of MOEAs. However, one disadvantage of this sort of selection mechanism is that it does not scale properly with the number of objectives.

In indicator-based algorithms, the selection process is guided by a performance indicator that measures the quality of an approximation set. The performance indicator allows us to assess the convergence behavior of the algorithm during the execution. Moreover, many indicators hold mathematical properties that make their theoretical analysis easier. However, the computational cost often increases rapidly with the number of dimensions, and the solutions' distributions might depend on the indicator's settings.

Decomposition-based algorithms split the MOP into multiple single-objective problems. This selection process offers a flexible algorithmic framework and is usually not expensive to compute. Nevertheless, they require a scalarizing function, which makes assumptions about the geometrical shape of the Pareto front.

As we saw, these three types of algorithms present some limitations, such as poor scaling with the number of objectives, sensitivity to their parameters, or the need of prior knowledge. This work aims to study alternative selection techniques that overcome these limitations. In particular, we will examine the properties of Molinet Berenguer and Coello Coello's scheme [5], which performs well in MOPs with diverse characteristics. Moreover, using this knowledge, we will propose new competitive selection schemes that can scale properly with the number of objectives and are computationally efficient.

1.2 Research hypothesis

The hypothesis under which we will work is that it is possible to propose a novel selection scheme for MOEAs that requires minimal extra information for the search, is computationally efficient, has good performance (with respect to state-of-the-art algorithms in the area), and can scale properly as the number of objective functions increases. This new scheme could be based on the method reported in [5].

1.3 Objectives

General Objective

The general objective of this work is to contribute to extending the state-of-the-art of alternative selection schemes that are not based on Pareto optimality, performance indicators, or decomposition.

Specific Objectives

1. To study state-of-the-art alternative selection schemes of multi-objective evolutionary algorithms to identify their advantages and disadvantages.

2. To propose at least one multi-objective evolutionary algorithm with a new alternative selection scheme.

3. To evaluate the proposed algorithm's performance against state-of-the-art multi-objective evolutionary algorithms using problems and performance indicators reported in the specialized literature.

1.4 Publications

The following publications were produced during the development of the thesis:

Conference publications

- D. C. Valencia-Rodríguez and C. A. Coello Coello. A novel performance indicator based on the linear assignment problem. In Michael Emmerich, André Deutz, Hao Wang, Anna V. Kononova, Boris Naujoks, Ke Li, Kaisa Miettinen, and Iryna Yevseyeva, editors, *Evolutionary Multi-Criterion Optimization: 12th International Conference, EMO 2023*, pages 348–360. Springer Nature Switzerland, 2023.

- D. C. Valencia-Rodríguez and C. A. Coello Coello. Multi-Objective Evolutionary Algorithm Based on the Linear Assignment Problem and the Hypervolume Approximation Using Polar Coordinates (MOEA-LAPCO). In G. Rudolph, A. V. Kononova, H. Aguirre, P. Kerschke, G. Ochoa, and T. Tušar, editors, *Parallel Problem Solving from Nature – PPSN XVII*, pages 221–233, Cham, 2022. Springer International Publishing.

- D. C. Valencia-Rodríguez and C. A. Coello Coello. An Ensemble of Scalarizing Functions and Weight Vectors for Evolutionary Multi-Objective Optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC'2021)*, pages 2459–2467. IEEE Press, 2021.

Journal publications

- Diana Cristina Valencia-Rodríguez and Carlos A. Coello Coello. Influence of the Number of Connections Between Particles in the Performance of a Multi-Objective Particle Swarm Optimizer. *Swarm and Evolutionary Computation*, 77(101231), March 2023.

1.5 Thesis structure

The rest of this thesis is organized as follows. Chapter 2 provides basic mathematical concepts associated with multi-objective optimization and a brief introduction to MOEAs.

Chapter 3 describes the selection scheme proposed by Molinet Berenguer and Coello Coello [5], which is the basis of this thesis. Additionally, this chapter examines the related work linked to this scheme.

Chapter 4 focuses on the combination of different scalarizing functions and weight vectors in the Molinet Berenguer and Coello Coello selection scheme. For this purpose, a novel ensemble algorithm is proposed, which combines the best-performing scalarizing functions and weight vectors. Furthermore, a performance evaluation of this algorithm is presented. This chapter also proposes an integration of the hypervolume indicator in the Molinet Berenguer and Coello Coello selection scheme to address some of its disadvantages. The resulting selection mechanism is incorporated into an MOEA, and its performance is evaluated experimentally.

Chapter 5 introduces a new performance indicator based on the selection scheme studied in this thesis. This new indicator is experimentally evaluated, and some detailed examples of its behavior are presented. In addition, a diversity indicator is proposed using an equivalent procedure. This new indicator is experimentally evaluated in artificial Pareto fronts and approximation sets.

Chapter 6 presents the conclusions of this thesis and some possible paths for future research.

Appendix A describes the test suites used in this thesis for the performance evaluation. Furthermore, Appendix B presents some additional experimental results of the thesis.

Chapter 2

Background

In this chapter, we introduce the necessary background to understand this thesis. Section 2.1 describes the main concepts associated with multi-objective optimization. Section 2.2 presents the characteristics of multi-objective evolutionary algorithms, how to evaluate them, and some examples. Finally, Section 2.3 summarizes the contents of this chapter.

2.1 Multi-objective Optimization

Optimization involves finding feasible solutions corresponding to the highest or lowest values of one or more objectives [3]. If there is only one objective function to consider, the process of finding the optimal solution is known as single-objective optimization. On the other hand, if two or more objective functions are involved, it is referred to as multi-objective optimization. The main difference between single- and multi-objective optimization is that, in the latter, different solutions may produce conflicting scenarios among objectives. A solution that is the best for one objective could be the worst for another. As a result, multi-objective optimization aims to identify the best possible trade-offs among the objective functions.

We commonly find numerous multi-objective optimization problems in the real world, such as industrial applications, transport engineering, and medicine [10]. Therefore, the study of multi-objective optimization problems and how to solve them has become a very relevant research area.

This section will introduce the most essential theory associated with multi-objective optimization. We start by presenting the formal definition of an optimization problem (Subsection 2.1.1). Then, we introduce the optimality notion for multi-objective optimization problems (Subsection 2.1.2) and the concept of Pareto dominance (Subsection 2.1.3). After that, we introduce the most commonly used reference points (Subsection 2.1.4). Finally, we present scalarizing functions (Subsection 2.1.5) and weight vectors (Subsection 2.1.6).

2.1.1 Multi-objective optimization problem

A multi-objective optimization problem involves optimizing two or more objective functions that are usually in conflict (i.e., improving one objective causes the deterioration of another). Due to this conflict, an optimal solution for one objective may not be optimal for the others. Therefore, multi-objective optimization aims to find solutions representing the best possible trade-offs among the objective functions. The general form of a multi-objective optimization problem (MOP) can be stated as follows (without loss of generality, we assume a minimization problem) [10]:

$$\begin{aligned} \text{Minimize} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0 \quad i = 1, \ldots, p; \\ & h_j(\boldsymbol{x}) = 0 \quad j = 1, \ldots, q. \end{aligned} \tag{2.1}$$

An MOP involves numerical quantities that are both controllable and unknown, named *decision variables*. To solve the problem, the value of these variables must be determined. A vector of n decision variables is represented by

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T,$$

where $\boldsymbol{x}$ belongs to the *decision variable space* ($\mathcal{D}$).

Moreover, MOPs are often limited by unique features of the environment or available resources. These limitations are called *constraints*, which describe the relationships among constants and decision variables involved in the problem. The MOP comprises p inequality constraints (expressed as $g_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, p$) and q equality constraints (expressed as $h_j(\boldsymbol{x}) = 0,\ j = 1, \ldots, q$). When a solution satisfies all constraints, it is named a feasible solution. Otherwise, it is called an infeasible solution. Additionally, the set of all feasible solutions is known as the *feasible region* ($\mathcal{S}$).

An MOP also consists of m objective functions, which are computable functions of the decision variables and help to determine the solutions' quality. The objective functions form a multi-dimensional space, called *objective function space* ($\mathcal{Z}$). Each value in the decision space has a corresponding point in the objective function space. Therefore, the mapping is from an n-dimensional decision vector to an m-dimensional objective vector (see Fig. 2.1). A vector of m objective functions is represented by

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T.$$

2.1.2 Optimality notion

Multiple optimal solutions may arise in an MOP due to conflicting objective functions. Therefore, the notion of optimum changes as we seek solutions that offer the best possible compromises. The most accepted notion of optimum is the so-called *Pareto Optimality*, originally proposed by Francis Ysidro Edgeworth and independently proposed in a more general way by Vilfredo Pareto [10]. The formal definition is as follows:

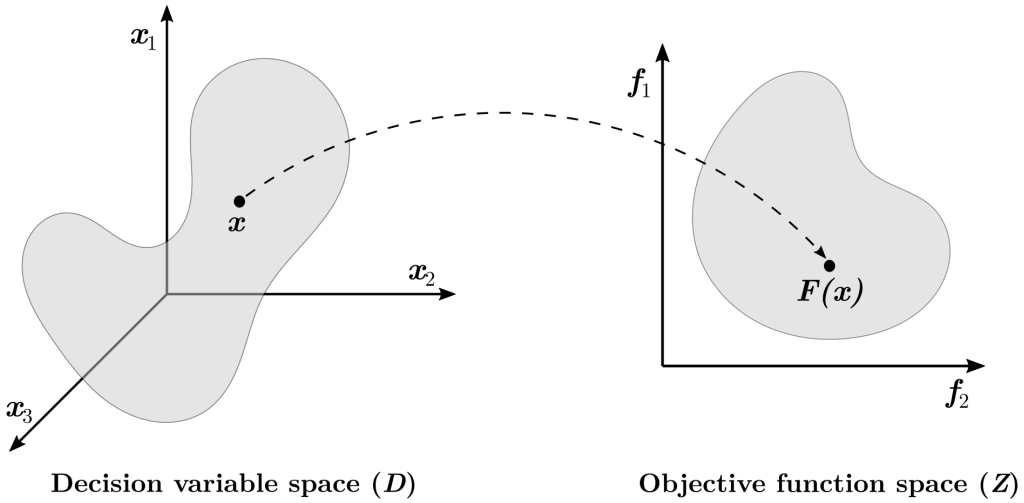


Figure 2.1: Mapping of a multi-objective problem

Definition 2.1 ([10])**.** A solution $\boldsymbol{x} \in \mathcal{S}$ is said to be *Pareto Optimal* with respect to $\mathcal{S}$ if and only if there is no $\boldsymbol{y} \in \mathcal{S}$, such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ for all $i = 1, \ldots, m$, and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$ for at least one $j \in \{1, \ldots, m\}$.

In other words, a solution is Pareto Optimal if no solution in the feasible region is better in at least one objective and not worse in all objectives. When addressing a MOP, the main goal is to identify the Pareto Optimal solutions, which form the so-called *Pareto Optimal Set*. Its formal definition is the following:

Definition 2.2. The *Pareto Optimal Set* $\mathcal{PS}^*$ is defined by:

$$\mathcal{PS}^* = \{\boldsymbol{x} \in \mathcal{S} \mid \boldsymbol{x} \text{ is Pareto Optimal}\}$$

Moreover, the image of the Pareto Optimal Set is called *Pareto Front*, and is formally defined as follows:

Definition 2.3. The *Pareto Front* $\mathcal{PF}^*$ is defined by:

$$\mathcal{PF}^* = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^m \mid \boldsymbol{x} \in \mathcal{PS}^*\}$$

2.1.3 Concept of Pareto dominance

Most multi-objective optimization algorithms use the concept of Pareto dominance to establish a partial order among solutions. Its formal definition is the following:

Definition 2.4 ([3])**.** A solution $\boldsymbol{x}_1$ is said to *dominate* another solution $\boldsymbol{x}_2$ (denoted as $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$), if both conditions 1 and 2 are true:

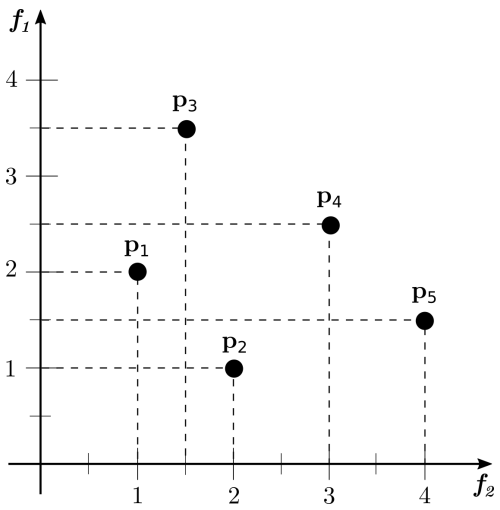


Figure 2.2: Example of dominance

1. The solution $\boldsymbol{x}_1$ is not worse than $\boldsymbol{x}_2$ in all objectives, ($f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ for all $i = 1, 2, \ldots, m$).

2. The solution $\boldsymbol{x}_1$ is strictly better than $\boldsymbol{x}_2$ in at least one objective, ($f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$ for at least one $j \in \{1, 2, \ldots m\}$).

In other words, a solution $\boldsymbol{x}_1$ is said to dominate another solution $\boldsymbol{x}_2$ if it is at least as good as $\boldsymbol{x}_2$ in all objectives and better than $\boldsymbol{x}_2$ in at least one objective. When comparing the dominance relation between two solutions, $\boldsymbol{x}$, and $\boldsymbol{y}$, there are three possible outcomes:

1. Solution $\boldsymbol{x}$ dominates solution $\boldsymbol{y}$.

2. Solution $\boldsymbol{x}$ is dominated by solution $\boldsymbol{y}$.

3. Solutions $\boldsymbol{x}$ and $\boldsymbol{y}$ do not dominate each other.

The previous three cases are illustrated in Figure 2.2, where we consider a two-objective minimization problem with five solutions. Point p_1 dominates points p_3 and p_4, as its objective values are better. Similarly, point p_5 is dominated by point p_2 because the former's objectives have a higher value. Lastly, points p_1 and p_2 do not dominate each other.

Other relationships among solutions used in multi-objective optimization are weak dominance and strict dominance. Their formal definitions are the following:

Definition 2.5 ([1])**.** A solution $\boldsymbol{x}_1$ is said to *weakly dominate* another solution $\boldsymbol{x}_2$ (denoted as $\boldsymbol{x}_1 \preceq \boldsymbol{x}_2$), if $\boldsymbol{x}_1$ is not worse than $\boldsymbol{x}_2$ in all objectives, or $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ for all $i = 1, 2, \ldots, m$.

Definition 2.6 ([1])**.** A solution $\boldsymbol{x}_1$ is said to *strictly dominate* another solution $\boldsymbol{x}_2$ (denoted as $\boldsymbol{x}_1 \prec\prec \boldsymbol{x}_2$), if $\boldsymbol{x}_1$ is better than $\boldsymbol{x}_2$ in all objectives, or $f_i(\boldsymbol{x}_1) < f_i(\boldsymbol{x}_2)$ for all $i = 1, 2, \ldots, m$.

2.1.4 Reference points

When dealing with MOPs, two reference points are crucial in determining the boundaries of the Pareto front: the ideal and Nadir objective vectors. The ideal objective vector sets the lower limit for each objective function within the feasible region. Moreover, this vector is a non-existent solution as its existence would indicate the absence of conflict among objective functions. Its formal definition is as follows:

Definition 2.7 ([11])**.** The components of the *ideal objective vector* $\boldsymbol{z}^* \in \mathbb{R}^m$ are obtained by minimizing each of the objective functions individually subject to the feasible region. That is, each component z_i^* is defined as:

$$z_i^* = \min_{\boldsymbol{x} \in \mathcal{S}} f_i(\boldsymbol{x}),$$

for all $i = 1, \ldots, m$.

On the other hand, the Nadir objective vector represents the upper limit in the Pareto Optimal Set. The formal definition of this vector is the following:

Definition 2.8. The components of the *Nadir objective vector* $\boldsymbol{z}^{nad} \in \mathbb{R}^m$ are obtained by maximizing each of the objective functions individually subject to the Pareto Optimal Set. That is, each component z_i^{nad} is defined as:

$$z_i^{nad} = \max_{\boldsymbol{x} \in PS^*} f_i(\boldsymbol{x}),$$

for all $i = 1, \ldots, m$.

The ideal and Nadir objectives vectors can be used to normalize the objectives using the following expression [3]:

$$f_i^{norm} = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*}. \tag{2.2}$$

2.1.5 Scalarizing functions

A scalarizing function $s : \mathbb{R}^n \to \mathbb{R}$ transforms a multi-objective problem into a single-objective problem of the following form [12]:

$$\begin{aligned} \text{Minimize} \quad & s(\boldsymbol{f'}(\boldsymbol{x}); \boldsymbol{w}) \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{S}, \end{aligned} \tag{2.3}$$

where $\boldsymbol{w} \in \mathbb{R}^m$ is a predefined *weight vector*, $\boldsymbol{f'}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}$, and $\boldsymbol{z} \in \mathbb{R}^m$ is a reference point. The weight vector must satisfy that $w_i \geq 0$ for all $i = 1, \ldots, m$, and $\sum_{i=1}^m w_i = 1$. Moreover, we assume that the ideal point is used as the reference point. Table 2.1 displays the information of the scalarizing functions used in this work where $\theta \in \mathbb{R}$ and $\alpha \in \mathbb{R}^+$.

Table 2.1: Scalarizing functions used in this work

Acronym	Name	Formulation	Suggested parameters	Reference
TCH	Tchebycheff function	$\max_i\{w_i\|f_i'\|\}$	-	[13]
ATCH	Augmented Tchebycheff	$\max_i\{w_i\|f_i'\|\} + \alpha\sum_i\|f_i'\|$	$\alpha \in [0.001, 0.01]$	[14]
ASF	Achievement Scalarizing Function	$\max\left\{\frac{f_i'}{w_i}\right\}$	-	[11]
AASF	Augmented Achievement Scalarizing Function	$\max\left\{\frac{f_i'}{w_i}\right\} + \alpha\sum_i\frac{f_i'}{w_i}$	$\alpha \approx 10^{-4}$	[11]
PBI	Penalty Boundary Intersection	$d_1 + \theta d_2$ where $d_1 := \left\|\boldsymbol{f'}\cdot\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right\|$ and $d_2 := \left\|\boldsymbol{f'} - d_1\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right\|$	$\theta = 5$	[15]
AGSF2	Artificially Generated Scalarizing Function 2	$\max_i\left\{\left\|w_i - \frac{f_i'}{w_i} - f_i'\right\|\right\}$	-	[16]
WS	Weighted Sum	$\sum_i w_i f_i'$	-	[17]

2.1.6 Weight vectors

In many cases and under certain assumptions, solving problem (2.3) leads to a Pareto Optimal solution. Therefore, we could approximate the Pareto Optimal Set if we have a set of uniformly distributed weight vectors. A popular technique for creating weight vectors is the Simplex Lattice Design (SLD) [18]. This method generates a series of vectors evenly spaced at intervals of $\delta = 1/H$ within a unit simplex, where the H parameter represents the number of divisions along each objective. Therefore, the number of vectors generated with SLD is $N = C_{m-1}^{H+m-1}$ for a problem with m objectives. Figure 2.3 shows an example of a set of weight vectors generated with the SLD method.

One drawback of the SLD method is that it produces many vectors at the boundary, i.e., vectors with $w_i = 0$ for at least one element w_i [19]. Additionally, the number of vectors increases non-linearly with the number of objectives. For example, a problem with eight objectives and eight divisions requires 6435 points. To address these disadvantages, other approaches to generate weight vectors have been proposed. Deb and Jain proposed a method that employs two layers of vectors: boundary and inside

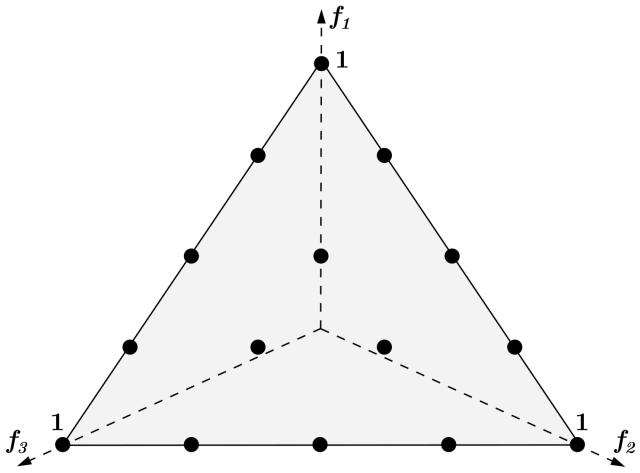


Figure 2.3: Weight vectors generated using the Simplex Lattice Design for a three-objective problem with $H = 4$.

layers [20]. The boundary is generated using the traditional SLD method. While the vectors of the inside layer are generated as follows [19, 20]:

$$w_i = (w_i + 1/m)/2 \text{ for } i = 1, 2, \ldots, m. \tag{2.4}$$

This transformation modifies the limits of the vectors to $1/2m \leq w_i \leq 1/2+1/2m$ for $i = 1, 2 \ldots, m$, pushing the vectors to the center of the simplex. Figure 2.4 shows an example of a set of weight vectors generated using the two-layered approach. For this illustration, we employ a three-objective problem. However, the two-layered method is commonly used in problems with eight or more objectives.

In 2015, Molinet Berenguer and Coello Coello proposed a method that overcomes the drawbacks mentioned earlier [5]. Their approach generates weight vectors by combining uniform design with the Hammersley method. The Hammersley method computes a set of points with minimal *discrepancy* (a numerical measure of scatteredness) at a low computational cost. This method relies on the p-adic representation of natural numbers, where every positive integer m can be uniquely expressed using a prime base $p \geq 2$ as follows:

$$m = \sum_{i=0}^{r} b_i \times p^i, \quad 0 \leq b_i \leq p-1, \quad i = 0, \ldots, r, \tag{2.5}$$

where $p^r \leq m \leq p^{r+1}$. Therefore, any integer $m \geq 1$, whose representation is given by (2.5), can be expressed as:

$$y_p(m) = \sum_{i=0}^{r} b_i \times p^{-(i+1)}, \tag{2.6}$$

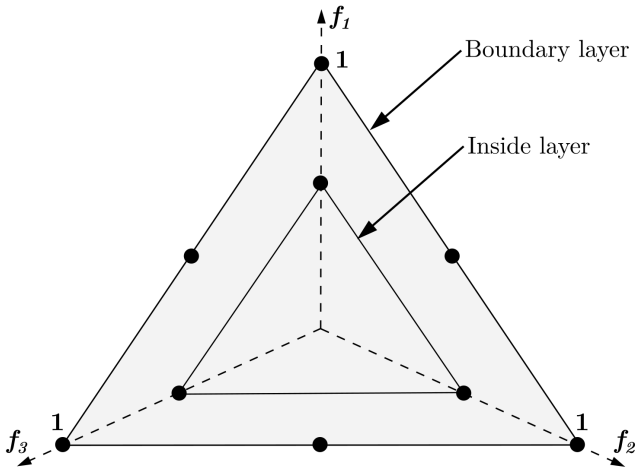


Figure 2.4: Weight vectors generated using a two-layered approach for a three-objective problem. The number of divisions is set to $H = 2$ for the boundary layer and $H = 1$ for the inside layer.

where $y_p(m) \in (0, 1)$ is referred to as the radical inverse of m base p. Accordingly, the set of points generated with the Hammersley method consists of n points given by

$$\boldsymbol{x}_i = \left[\frac{2i-1}{2n}, y_{p_1}(i), \ldots, y_{p_{m-1}}(i)\right]^T \quad i = 1, \ldots, n \tag{2.7}$$

where $m \geq 2$ and $p_1, \ldots, p_{m-1}$ are distinct prime numbers.

On the other hand, uniform design is a technique that selects a set of points to cover a region of interest efficiently. This method requires a set of points with slight discrepancy. Hence, Molinet Berenguer and Coello Coello suggested using the Hammersley method to generate low-discrepancy points $U = \{\boldsymbol{u}_i = [u_{i1}, \ldots, u_{i(m-1)}]^T \mid i = 1, \ldots, n\} \subset [0, 1]^{m-1}$ and then applying the following uniform design transformation:

$$w_{ti} = (1 - u_{ti}^{\frac{1}{m-i}}) \prod_{j=1}^{i-1} u_{tj}^{\frac{1}{m-j}}, \quad i = 1, \ldots, m-1, \tag{2.8}$$

$$w_{tm} = \prod_{j=1}^{m-1} u_{tj}^{\frac{1}{m-j}}, \quad t = 1, \ldots, n. \tag{2.9}$$

Figure 2.5 illustrates a set of points generated using uniform design with Hammersly's method.

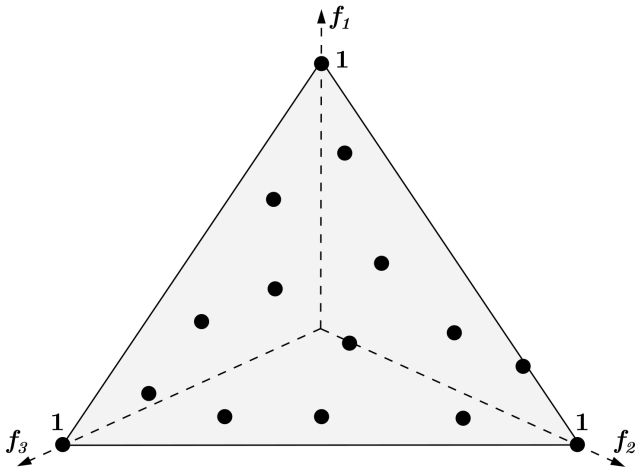


Figure 2.5: Weight vectors generated using the uniform design with Hammersly's method.

2.2 Multi-objective evolutionary algorithms

Evolutionary Algorithms (EAs) are search heuristics that imitate neo-Darwinian evolutionary theory [21] to solve optimization problems. EAs have gained popularity in tackling complex multi-objective optimization problems (the so-called Multi-Objective Evolutionary Algorithms (MOEAs)) that exact methods can't properly handle. The reason for their popularity is that MOEAs have inherent parallelism, the ability to explore intractable large spaces, and a framework that is easy to understand. Furthermore, these algorithms can generate several elements of the Pareto Optimal Set in a single run.

MOEAs use a population (set) of individuals (decision vectors) to search for optimal solutions adopting natural evolution operators such as selection, recombination, and mutation. In addition, many MOEAs use an external archive to store non-dominated solutions found in the search process. Also, they adopt a mechanism called *density estimator* which is responsible for maintaining diversity in the population. The behavior of an MOEA is determined by the choice of each operator, the effectivity of the density estimator, the use of an external archive, and the way in which each solution is inserted into the archive.

In the remainder of this section, we will explain the general framework of MOEAs (Subsection 2.2.1). Then, we will describe different performance indicators to evaluate MOEAs (Subsection 2.2.2) and some commonly used density estimators (Subsection 2.2.3). Finally, we will introduce different selection mechanisms adopted by MOEAs (Subsection 2.2.4).

2.2.1 General framework

An MOEA involves a population of individuals (or potential solutions to a specific problem) that evolve through the use of different evolutionary operators to produce individuals with better characteristics. The three principal evolutionary operators of MOEAs are the following [22, 21]:

1. **Mutation operator**. It corresponds to an erroneous self-replication of individuals. Thus, it operates on a single individual to produce a new one. The main purpose of this operator is to introduce diversity into the population while preserving some of the individual's information. An example of a mutation operator is polynomial-based mutation (PM) [23].

2. **Recombination operator**. It combines two or more individuals (named parents) and produces one or more individuals (named offspring). This operator aims to exchange information between individuals of a population and uses this knowledge to generate better solutions. Simulated binary crossover (SBX) [23] is an example of this operator.

3. **Selection operator**. It plays a crucial role in evolution by picking individuals with superior fitness values to survive and reproduce. The goal of this operator is to guide the population towards better solutions. In MOEAs, this selection operator is essential because more than one solution is optimal. Therefore, we need to employ alternative methods to evaluate the fitness of individuals instead of relying solely on objective function values.

We denote $P(t) = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\mu\}$ as a population at generation t, and A as the external archive that contains the non-dominated solutions found so far. Therefore, the general framework of an MOEA is shown in Algorithm 1. First, the population $P(0)$ is initialized and assessed. Next, the external archive A is initialized with the non-dominated solutions in $P(0)$. In the main loop of the MOEA, a set of new offspring $P''(t)$ is created through recombination and mutation of the parent population $P(t)$. The resulting population $P''(t)$ is evaluated in the subsequent step, and the external archive A is updated. Then, the fittest solutions are selected from both the parent and offspring populations (i.e, from $P''(t) \cup P(t)$). Finally, after the loop, either the last population or the external archive is returned.

The result of an MOEA is called an *Approximation Set*, which is formally defined as follows:

Definition 2.9 ([24])**.** Let $A \subseteq \mathcal{Z}$ be a set of objective vectors. A is called an *approximation set* if any element of A does not weakly dominate any other objective vector in A.

Algorithm 1 General framework of an MOEA

```
1: t := 0
2: Initialize P(0)
3: Evaluate P(0)
4: Initialize A with the non-dominated solutions from P(0)
5: while stopping criterion is not fulfilled do
6:     P'(t) := recombine P(t)
7:     P''(t) := mutate P'(t)
8:     Evaluate P''(t)
9:     Update A using P''(t)
10:    P(t + 1) := select from P''(t) ∪ P(t)
11:    t := t + 1
12: end while
13: return P_t or A
```

2.2.2 Quality indicators

The performance of an MOEA is determined by the quality of the approximation set and the computational resources needed to generate it [24]. Regarding the computational resources, monitoring the number of fitness evaluations or the overall run-time is common in single- and multi-objective optimization. However, this situation changes when it comes to quality. In a single-objective context, the quality of a solution is determined by the objective function: a smaller or larger value indicates a better solution. On the other hand, in a multi-objective context, we rely on the dominance criterion to compare solutions. However, this criterion is ineffective when there are incomparable solutions. As a result, it is unclear what quality means concerning approximations of the Pareto optimal set.

We can make some statements about the quality of approximation sets compared to others. In particular, we can distinguish five relations between two approximation sets shown in Table 2.2. On the other hand, according to Li and Yao [25], we can break down the quality of an approximation set into four aspects:

1. **Convergence**. It refers to the closeness of the set to the Pareto front.

2. **Coverage**. It considers the region covered by the approximation set.

3. **Uniformity**. It refers to how even the solution distribution is in the set. An equidistant spacing among solutions is desirable.

4. **Cardinality**. It refers to the number of solutions in the set.

We can use mathematical expressions (called *quality indicators*) to evaluate these four aspects in an approximation set. The formal definition of a quality indicator is the following:

Table 2.2: Preference relations on Pareto front approximations [1]

Relation	Symbol	Description
Strictly dominates	$A \prec\prec B$	Every $\boldsymbol{b} \in B$ is strictly dominated by at least one $\boldsymbol{a} \in A$
Dominates	$A \prec B$	Every $\boldsymbol{b} \in B$ is dominated by at least one $\boldsymbol{a} \in A$
Better	$A \lhd B$	Every $\boldsymbol{b} \in B$ is weakly dominated by at least one $\boldsymbol{a} \in A$ and $A \neq B$
Weakly dominates	$A \preceq B$	Every $\boldsymbol{b} \in B$ is weakly dominated by at least one $\boldsymbol{a} \in A$
Incomparable	$A \| B$	Neither A weakly dominates B nor B weakly dominates A

Definition 2.10 ([24])**.** An m-ary quality indicator I is a function $I : \Omega^m \rightarrow \mathbb{R}$, which assigns each vector $(A_1, A_2, \ldots, A_m)$ of m approximation sets a real value $I(A_1, \ldots, A_m)$.

In this work, we are interested in *unary quality indicators*, i.e., when $m = 1$. These indicators can be *Pareto-compliant*, *weakly Pareto-compliant*, or none of the above. The formal definitions of the first two concepts are shown below, assuming that a greater indicator value corresponds to a higher quality.

Definition 2.11 ([26])**.** Given two approximation sets A and B, a unary indicator I is $\lhd$-compliant (Pareto-compliant) if $A \lhd B \Rightarrow I(A) > I(B)$.

Definition 2.12 ([26])**.** Given two approximation sets A and B, a unary indicator I is weakly $\lhd$-compliant (weakly Pareto-compliant) if $A \lhd B \Rightarrow I(A) \geq I(B)$.

In the rest of the section, we will introduce some of the most important unary indicators reported in the literature.

Hypervolume

The hypervolume (HV) [27] measures the size of the objective space dominated by a solution set A given a reference point $\boldsymbol{z}$. The HV indicator is defined as follows:

$$HV(A, \boldsymbol{z}) = \Lambda \left(\bigcup_{\boldsymbol{a} \in A} \{\boldsymbol{x} | \boldsymbol{a} \prec \boldsymbol{x} \prec \boldsymbol{z}\} \right)$$

where Λ denotes the Lebesgue measure and $\boldsymbol{z} \in \mathbb{R}^m$ might be dominated by all the elements in A. Higher values of this indicator are preferred.

An advantage of the HV is that its maximization leads to the Pareto-optimal set. Moreover, it is the only unary indicator that is Pareto-compliant. However, a disadvantage of the HV is that its computational cost grows exponentially with the number of objectives.

S-energy

S-energy is a performance indicator that measures the uniform distribution of a set in a d-dimensional manifold. Given an approximation set $A \subset \mathbb{R}^m$, s-energy is defined as [28]:

$$E_s(A) = \sum_{\boldsymbol{x} \in A} \sum_{\substack{\boldsymbol{y} \in A \\ \boldsymbol{y} \neq \boldsymbol{x}}} k_s(\boldsymbol{x}, \boldsymbol{y}), \tag{2.10}$$

where

$$k_s(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} ||\boldsymbol{x} - \boldsymbol{y}||^{-s}, & s > 0 \\ -\log ||\boldsymbol{x} - \boldsymbol{y}||, & s = 0 \end{cases} \tag{2.11}$$

An advantage of the s-energy indicator is that its minimization leads to asymptotically uniformly distributed solutions [29]. Therefore, lower values of this indicator are preferred.

R2 indicator

Given an approximation set $A \subset \mathbb{R}^m$ and a set of utility functions U, the $R2$ indicator is defined as follows [30]:

$$R2(A, U) = \frac{1}{|U|} \sum_{u \in U} \min_{\boldsymbol{a} \in A} \{u(\boldsymbol{a})\}. \tag{2.12}$$

The R2 indicator is weakly Pareo-compliant, scaling-dependent, and has a low computational cost [31]. Moreover, lower values of this indicator are preferred. Many possibilities exist regarding the choice of utility functions, such as TCH, ASF, or PBI (see Table 2.1).

Inverted generational distance

Given an approximation set $A \subset \mathbb{R}^m$ and a reference set $Z \subset \mathbb{R}^m$, the inverted generational distance (IGD) is defined as follows [32]:

$$IGD(A, Z) = \frac{1}{|Z|} \left(\sum_{\boldsymbol{z} \in Z} d(\boldsymbol{z}, A)^p \right)^{1/p} \tag{2.13}$$

where $p > 0$ and $d(\boldsymbol{z}, A) = \min_{\boldsymbol{a} \in \boldsymbol{A}} \sqrt{\sum_{i=1}^{m} (z_i - a_i)^2}$. In other words, the IGD indicator measures the average distance between each reference point and its nearest solution. A small value of IGD indicates good distribution and convergence of an approximation set.

Modified inverted generational distance

The modified inverted generational distance (IGD^+) [33] measures the average distance between a reference set and a Pareto front approximation. Given an approximation set $A \subset \mathbb{R}^m$ and a reference set $Z \subset \mathbb{R}^m$, the IGD^+ is defined as follows:

$$IGD^+(A, Z) = \frac{1}{|Z|} \sum_{\boldsymbol{z} \in Z} \min_{\boldsymbol{a} \in A} d^+(\boldsymbol{a}, \boldsymbol{z}) \tag{2.14}$$

where $d^+(\boldsymbol{a}, \boldsymbol{z}) = \sqrt{\sum_{i=1}^{m} (\max\{a_i - z_i, 0\})^2}$. The IGD^+ indicator is weakly Pareto-compliant and allows measuring diversity and convergence of an approximation set. Lower values of this indicator are preferred.

2.2.3 Density estimators

Density estimators keep MOEAs from converging to a single solution by maintaining the diversity of solutions in a population. Some of the most common density estimators are the following:

- **Fitness sharing** [34, 35]: The fitness sharing (or niching) estimator was introduced by Goldberg in 1987 to promote the solutions' distribution in single-objective optimization and was later adopted by Fonseca and Fleming for multi-objective optimization [34, 36]. This estimator uses a radius (ϕ_{share}) to define the niche of each individual. Then, an individual's density is calculated using a sharing function that considers the sum of distances from the reference individual to each solution within the niche.

- **Adaptive grid** [37, 38]. This estimator places the solutions in a grid based on its objective values and computes the number of elements in each grid location. Accordingly, a solution in a grid location with fewer elements is preferred. The Pareto Archived Evolution Strategy (PAES) was the first MOEA that adopted the adaptive grid [37].

- **Crowding** [39]. The crowding estimator was proposed by Deb et al. in 2000 to assess the density of solutions around a particular individual [39]. It is set as the estimation of the perimeter of the cuboid formed by the two nearest neighbors of a solution.

- **Clustering** [40, 41]. This estimator uses algorithms (usually deterministic) to create a fixed number of solution groups or clusters. The density of a solution is determined by its distance from other solutions in the same cluster. An example of an MOEA that employs the clustering method is the Strength Pareto Evolutionary Algorithm (SPEA) [40].

- **Parallel Coordinates** [42]. This indicator represents a graph in a digital image, where a pixel specifies the level of overlapping line segments. Hence, the

individuals that cover a larger area of the image have a higher probability of survival. The Multi-objective Optimizer based on Value Path (MOVAP) [42] is an example of an MOEA that uses this density estimator.

- **Entropy** [43, 44]. In this estimator, the concept of entropy is taken from information theory to measure the distribution of solutions based on the amount of neighborhood information.

2.2.4 Classification based on the selection scheme

In 1984, David Schaffer presented the first implementation of a MOEA named Vector Evaluated Genetic Algorithm (VEGA) [45]. Since then, a wide variety of MOEAs with distinct characteristics have been proposed. They were initially classified as Pareto-based and non-Pareto-based methods [46]. However, more recent variants are classified into three prominent families based on their selection scheme [46, 4]:

1. **Pareto-based.** The algorithms in this class incorporate the concept of Pareto dominance to select solutions, usually through non-dominated sorting and density estimators. For several years, many MOEAs have employed the Pareto-based selection mechanism and have effectively addressed MOPs with two or three objective functions [41, 39]. However, when solving problems with four or more objectives, the MOEAs' performance deteriorates since the selection pressure dilutes. This dilution is caused by the quick growth of non-dominated solutions as the number of objectives increases, making Pareto-based selection equivalent to choosing solutions randomly [47, 48]. Two of the most representative algorithms in this class are NSGA-II [39] and SPEA2 [41].

2. **Indicator-based.** These MOEAs use a performance indicator to guide the selection process. The underlying idea is to optimize the indicator value of the population throughout the evolutionary process, transforming the MOP into a single-objective optimization problem [49]. The hypervolume is the most commonly used performance indicator due to its mathematical properties, such as Pareto compliance [50, 51]. However, its computational cost increases exponentially with the number of objectives. For this reason, other indicators like R2 have been employed to guide MOEAs despite their theoretical limitations [52]. In addition to the high computational cost, the approximation sets found by indicator-based MOEAs are strongly related to the indicators' preferences. IBEA [50] and SMS-EMOA [51] are two examples of this class of MOEAs.

3. **Decomposition-based.** This class of MOEAs transforms the MOP into multiple single-objective problems, each targeting different segments of the Pareto front. A distinct parametrization (or weighting) of a scalarizing function defines every subproblem. Therefore, a single run of a decomposition-based MOEA could lead to the subproblems' solutions and, consequently, to approximating

the Pareto front [53]. However, the effectiveness of these algorithms depends heavily on the adopted scalarizing function and the method used to generate weights [46]. Moreover, a uniform distribution of weight vectors does not guarantee a uniformly distributed Pareto front [53]. MOGLS [54] and MOEA/D [15] are two remarkable examples of this class of MOEAs.

Many drawbacks of the three previous selection schemes have been addressed. For instance, low-cost approximations of the hypervolume have been suggested [55, 56]. Nevertheless, the problems have not been fully solved. This thesis explores alternative selection schemes that are not part of the three existing families previously described. In particular, we study here the work proposed by Molinet Berenguer and Coello Coello [5] that transforms the selection process into a Linear Assignment Problem (see Chapter 3).

2.3 Summary

This chapter presented the most essential aspects of multi-objective optimization and MOEAs. First, the formal definition of an MOP was presented. Then, we introduced the most common optimality notion of an MOP (Pareto optimality) and described the concept of Pareto dominance. After that, we also described the Nadir and ideal objective vectors, two crucial reference points in an MOP. In addition, we presented the scalarizing functions that transform an MOP into a single-objective problem using a weight vector. The methods for generating these vectors were described later on.

Regarding MOEAs, we first introduced their general framework in conjunction with their principal evolutionary operators: mutation, recombination, and selection. Then, we presented some quality indicators that determine the quality of an approximation set. Additionally, we introduced the most frequently used density estimators. Finally, we showed the three most common families of MOEAs according to their selection scheme: Pareto-based, Indicator-based, and Decomposition-based.

Chapter 3

A selection scheme based on the Linear Assignment Problem

The selection mechanism based on the linear assignment problem is considered an alternative selection scheme because it does not use Pareto optimality information to select new individuals. Its core idea is transforming the MOEA's selection process into a linear assignment problem. The algorithms that incorporate this selection scheme have been found to obtain good approximations of the Pareto Optimal Front without relying on any extra mechanism. This selection mechanism it is the object of study of this thesis.

This chapter aims to explain in detail the linear assignment problem selection scheme. For this purpose, Section 3.1 introduces what an assignment is, and Section 3.2 formally defines a linear assignment problem. Section 3.3 describes a method to solve this problem called the Kuhn-Munkres algorithm. Then, Section 3.4 introduces the selection scheme, and Section 3.5 discusses some algorithms that incorporate it. Finally, Section 3.6 provides a summary of the chapter.

3.1 Assignment problem

Assignment problems aim to determine the optimal way to assign n items (jobs or students) to n other items (machines or tasks). Formally, an assignment can be described as a bijective mapping φ between two finite sets U and V of n elements [57]. The mapping can be represented as a *permutation* φ with the form

$$\begin{pmatrix} 1 & 2 & \dots & n \\ \varphi(1) & \varphi(2) & \dots & \varphi(n) \end{pmatrix},$$

where 1 is mapped to $\varphi(1)$, 2 is mapped to $\varphi(2)$, ..., and n is mapped to $\varphi(n)$.

Every permutation φ of the set $\{1, 2, \dots, n\}$ corresponds to an $n \times n$ *permutation*

matrix $X_\varphi = (x_{ij})$ with

$$x_{ij} = \begin{cases} 1 & \text{if } j = \varphi(i), \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

which satisfies the linear system of equations

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1, 2, \ldots, n), \tag{3.2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1, 2, \ldots, n). \tag{3.3}$$

The set of equations (3.2) and (3.3) show that each permutation matrix must have one element assigned to each row and column, resulting in a sum of 1 for each. For instance, the assignment represented by the permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}, \tag{3.4}$$

corresponds to the following permutation matrix

$$X_\varphi = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Bipartite graphs can also be used to represent assignments. A graph $G = (U, V; E)$ is called *bipartite* if its vertex set E can be partitioned into two non-empty subsets, U and V, such that each edge of G has one end in U and the other in V [58]. A *matching* M in G is a subset of the edges E such that every vertex of G meets at most one edge of the matching. Moreover, a matching M is called a *perfect matching* if every vertex of graph G corresponds to an edge in M, and the number of vertices in U and V equals n [57]. Therefore, any assignment problem can be depicted as a perfect matching. Figure 3.1 displays the bipartite graph of the assignment in (3.4).

3.2 Linear sum assignment problem

The linear sum assignment problem (or linear assignment problem) involves assigning n agents to n tasks based on a cost matrix $C = (c_{ij})$, where each cell c_{ij} represents the cost of assigning the i^{th} agent to the j^{th} task. The goal is to find the assignment that results in the lowest sum of costs. For the rest of this thesis, we will use the acronym LAP to refer to the linear sum assignment problem.

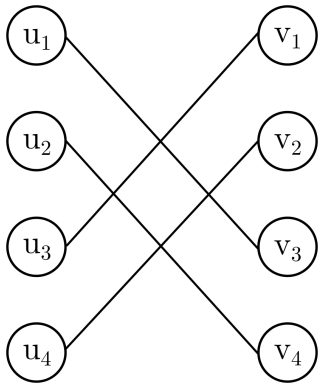


Figure 3.1: Assignment represented by a bipartite graph

Given a cost matrix $C = (c_{ij})$ and a binary matrix $X = (x_{ij})$ such that

$$x_{ij} = \begin{cases} 1 & \text{if row } i \text{ is assigned to column } j, \\ 0 & \text{otherwise.} \end{cases}$$

A LAP can be modeled as

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{3.5}$$

$$s.t. \quad \sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1, 2, \ldots, n), \tag{3.6}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1, 2, \ldots, n), \tag{3.7}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j = 1, 2, \ldots, n). \tag{3.8}$$

Note that the resulting matrix X is a permutation matrix.

3.3 Kuhn-Munkres algorithm

The Kuhn-Munkres algorithm, or the Hungarian algorithm, is a well-known method for solving LAPs. Kuhn initially introduced it in 1955 [59] , and Munkres later enhanced it in 1957 [60]. Although its initial computational complexity was $O(n^4)$, the most efficient implementation is now $O(n^3)$ [57]. The Kuhn-Munkres algorithm is described below.

Input:

– A cost matrix of size $n \times n$.

Output:

– The optimal assignment.

Procedure:

Step 0) Initialization:

Step 0.1) For each row of the matrix, find the smallest element and subtract it from every component of the row.

Step 0.2) For each column of the resulting matrix, find the smallest element and subtract it from every component of the column.

Step 1) Find a partial feasible solution:

Step 1.1) Find a zero in the resulting matrix.

Step 1.2) If there is no starred zero in its row or column, mark that zero with a star (*).

Step 1.3) Repeat for each zero in the matrix.

Step 1.4) Go to Step 2.

Step 2) Check for optimality:

Step 2.1) Cover every column containing a starred zero (0*).

Step 2.2) If all columns are covered, the starred zeros are the optimal assignment; stop the process.

Step 2.3) Otherwise, go to Step 3.

Step 3) Prime zeros:

Step 3.1) Choose an uncovered zero and prime it. Let r be the row containing it.

Step 3.2) If there is no starred zero z in row r, go to Step 4.

Step 3.3) Otherwise, cover row r and uncover the column of z.

Step 3.4) Repeat until all zeros are covered.

Step 3.5) Go to Step 5.

Step 4) Construct an alternate path:

Step 4.1) Construct a sequence of alternating starred and primed zeros as follows:

a) The uncovered 0' is the first element Z_0.

b) The 0* in Z_0's column (if any) is the second element Z_1.

c) The 0' in $Z_1's$ row is the third element Z_2.

d) Continue similarly until the sequence stops at a 0' with no 0* in its column.

Step 4.2) Unstar each starred zero of the sequence, and star each primed zero of the sequence.

Step 4.3) Erase all primes and uncover every line.

Step 4.4) Return to Step 2.

Step 5) Create additional zeros:

Step 5.1) Find the smallest uncovered element (h) of the matrix.

Step 5.2) Increase each twice-covered element by h.

Step 5.3) Decrease each uncovered element by h.

Step 5.4) Return to Step 3.

Figure 3.2 illustrates the manual execution of the Kuhn-Munkres algorithm to assign three agents to three tasks.

	t_1	t_2	t_3
a_1	10	15	9
a_2	9	18	5
a_3	6	14	3

Cost matrix

	t_1	t_2	t_3
a_1	0	0	0
a_2	3	7	0
a_3	2	5	0

Step 0

	t_1	t_2	t_3
a_1	0*	0	0
a_2	3	7	0*
a_3	2	5	0

Step 1

	t_1	t_2	t_3
a_1	0*	0	0
a_2	3	7	0
a_3	2	5	0*

Step 2

	t_1	t_2	t_3
a_1	0*	0'	0
a_2	3	7	0*
a_3	2	5	0

Step 3

	t_1	t_2	t_3
a_1	0*	0'	2
a_2	1	5	0*
a_3	0	3	0

Step 5 ($h = 2$)

	t_1	t_2	t_3
a_1	0*	0'	2
a_2	1	5	0*
a_3	0'	3	0

Step 3

	t_1	t_2	t_3
a_1	$z_1 =$ 0*	$z_2 =$0'	2
a_2	1	5	0*
a_3	$z_0 =$0'	3	0

Step 4.1

	t_1	t_2	t_3
a_1	0	0*	2
a_2	1	5	0*
a_3	0*	3	0

Step 4.2-4.4

	t_1	t_2	t_3
a_1	0	0*	2
a_2	1	5	0*
a_3	0*	3	0

Step 2

Figure 3.2: Execution of the Kuhn-Munkres algorithm. The resulting assignment is a_1 with t_2, a_2 with t_3, and a_3 with t_1. Moreover, the minimum assignment cost is 26.

3.4 Transforming the selection process

In 2015, Molinet Berenguer and Coello Coello [5] proposed transforming a MOEA's selection process into a LAP. In this transformation, we consider a set of individuals (the parents and their offspring) and a set of weight vectors representing regions of the Pareto Front. The cost of assigning an individual to a weight vector measures how suited this individual is to the part of the Pareto Front that the vector represents. Hence, we can identify which individuals better characterize the Pareto Front (i.e., we can identify the best individuals in the population) by finding the assignment with minimal cost.

The assignment cost can be computed using a scalarizing function (though other methods can also be adopted [61]). Therefore, the elements c_{ij} of the cost matrix C is computed as follows:

$$c_{ij} = s(\tilde{\boldsymbol{f}}(\boldsymbol{x_j}), \boldsymbol{w_i}) \quad i = 1, \ldots, n, \quad j = 1, \ldots, 2n \tag{3.9}$$

where s is the scalarizing function, $2n$ is the size of the population considering parents and offspring, n the number of weight vectors, and $\tilde{\boldsymbol{f}}(\boldsymbol{x_j})$ is the normalized objective vector. This vector is defined as:

$$\tilde{\boldsymbol{f}}(\boldsymbol{x_j}) = [\tilde{f}_1(\boldsymbol{x_j}), \ldots, \tilde{f}_k(\boldsymbol{x_j})] \tag{3.10}$$

$$s.t. \quad \tilde{f}_i(\boldsymbol{x_j}) = \frac{f_i(\boldsymbol{x_j}) - z_i^{\min}}{z_i^{\max} - z_i^{\min}}, \qquad i = 1, \ldots, k \tag{3.11}$$

$$z_i^{\min} = \min_{l=1,\ldots,2n} f_i(\boldsymbol{x_l}), \qquad i = 1, \ldots, k \tag{3.12}$$

$$z_i^{\max} = \max_{l=1,\ldots,2n} f_i(\boldsymbol{x_l}), \qquad i = 1, \ldots, k \tag{3.13}$$

where $f_i(\boldsymbol{x_j})$ is the i^{th} function value of the j^{th} solution. We used in this work the Hungarian algorithm to solve the LAP in which the number of agents $2n$ must be equal to the number of tasks. Hence, we added dummy costs (where $c_{ij} = 0$ for $i = n+1, \ldots, 2n$ and $j = 1, \ldots, 2n$) to match the values, as recommended in [57].

To illustrate the selection process, let's assume that we have a population of six individuals (shown in Figure 3.3) and we must select three. Moreover, we use the weight vectors in Figure 3.3 to describe the different regions of the Pareto front. Therefore, we first compute the cost matrix using the ASF scalarizing function (see Table 3.1). Then, we apply the Kuhn-Munkres algorithm and we obtain the following assignment:

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 6 & 1 \end{pmatrix},$$

that is w_1 with I_4, w_2 with I_6, and w_3 with I_1. Finally, the selected individuals are I_1, I_4, and I_6. We can observe that the chosen individuals are non-dominated solutions, and each of them is assigned to the closest weight vector.

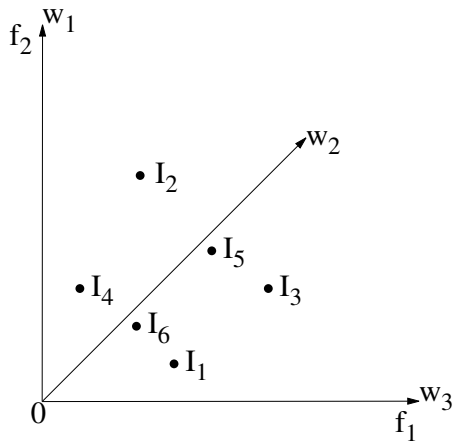


Figure 3.3: Example of a population of six individuals and three weight vectors

C	I_1	I_2	I_3	I_4	I_5	I_6
w_1	500.0	320.0	1000.0	0.4	700.0	300.0
w_2	0.71	1.43	1.43	0.57	1.0	0.43
w_3	0.5	1000.0	400.0	400.0	600.0	200.0

Table 3.1: Example of a cost matrix for six individuals and three weight vectors. Grey cells show the assignment elements.

3.5 Algorithms that incorporate the linear assignment problem transformation

Molinet Berenguer and Coello Coello proposed the first algorithm that incorporates the LAP transformation in its selection process: the Hungarian Differential Evolution (HDE) approach [5]. The algorithm works as follows. First, it randomly initializes the population and evaluates it. After that, it generates the weight vectors using the UDH method. In the main loop of the algorithm, it creates new solutions using DE/*rand*/1/*bin* [62] and evaluates them. Then, it merges the parents with their offspring and normalizes their objective values. With the resulting information, the algorithm constructs the cost matrix using a scalarizing function and a weight vector set. After that, the algorithm solves the LAP using the Hungarian method, and the resulting assignment determines which individual proceeds to the next generation. Algorithm 2 summarizes the above process.

In order to evaluate its performance, the HDE was compared to the MOEA/D [15], appSMS-EMOA [5], and SMS-EMOA [51]. The adopted problems were selected from the ZDT [63] and DTLZ [64] (using two to ten objectives) test suites. Moreover, the hypervolume indicator was used as the performance measure. The experimental results showed that the HDE outperforms MOEA/D and appSMS-EMOA, and it outperforms SMS-EMOA in many test instances but while having a lower computational

Algorithm 2 Hungarian Differential Evolution (HDE)

Require: Multi-objective problem, population size (n), maximum number of generations (g_{max}), parameters C_r and F for DE/$rand$/1/bin
Ensure: Last population (P_{gmax})
1: Generate initial population P_1 randomly
2: Evaluate each individual in P_1
3: $W \leftarrow$ Generate n weight vectors using UDH
4: **for** $i \leftarrow 0$ to g_{max} **do**
5: $P_g^* \leftarrow$ Generate offspring from P_g using DE/$rand$/1/bin
6: Evaluate each individual in P_g^*
7: $Q_g \leftarrow P_g \cup P_g^*$
8: $NQ_g \leftarrow$Normalize objectives of each individual in Q_g
9: $C \leftarrow$ Construct a cost matrix using NQ_g, W, and a scalarizing function
10: $I \leftarrow$Obtain the best assignment in C using the Hungarian method
11: $P_{g+1} \leftarrow \{\boldsymbol{x}_i | i \in I, \boldsymbol{x}_i \in Q_g\}$
12: **end for**

cost.

In 2015, Miguel Antonio and Coello Coello incorporated the LAP selection into a Multi-Objective Particle Swarm Optimizer (MOPSO) [65]. The core idea was to merge the current and new particles and select the best ones using the LAP transformation. The resulting algorithm was called LAP based PSO (LAPSO). The experimental results showed that LAPSO offers competitive results compared to state-of-the-art MOPSOs in problems with three to ten objectives.

In 2016, Manoatl Lopez and Coello Coello proposed the IGD$^+$-MOEA [61]. In this case, the problem consisted of assigning a reference set representing the Pareto front to individuals created with the SBX and PM [23] operators. For this purpose, they used the modified distance calculation from IGD$^+$ [33] as the assignment cost. Therefore, the aim was to minimize the sum of the modified distances between each reference point and each individual. In order to compute the assignment cost, the objective function values are normalized using the expression

$$f_i'(\boldsymbol{x}_i) = \frac{f_i(\boldsymbol{x}_i)}{u_i} \tag{3.14}$$

$$s.t. \quad u_i = \max_{j=1,\ldots,N} f_i(\boldsymbol{x}_j) \tag{3.15}$$

for all $i = 1, \ldots, m$, where N is the population size and m the number of objective functions. Hence, the assignment cost between a solution $\boldsymbol{x}_i$ and a vector $\boldsymbol{v}_j$ is defined as follows

$$C_{i,j} = \sqrt{\sum_{k=1}^{m} \left(\max\{f_{i,k}'(\boldsymbol{x}_i) - v_{j,k}, 0\} \right)^2} \tag{3.16}$$

The authors compared the performance of IGD$^+$-MOEA with the MOEA/D [15] and a version of the SMS-EMOA that employs the Monte Carlo sampling to approximate the hypervolume [66]. Moreover, they adopted problems from the DTLZ [64] and WFG [67] test suites with two to eight objectives. The experiments showed that the IGD$^+$-EMOA can produce better approximations than the other algorithms in many test instances.

In 2017, Sun et al. proposed an algorithm called global view-based selection NSGA-III (GS-NSGA-III) [68], which incorporated the LAP transformation into the selection process of NSGA-III [20]. However, the assignment cost between a solution s_i and a vector v_j is determined by the rank of s_i and its normalized perpendicular distance to v_j. Algorithm 3 presents the survival selection of GS-NSGA-III. First, the individuals are ranked using non-dominated sorting. Then, the perpendicular distance to each solution is computed for each reference vector, and the distance's limits are obtained. After that, the assignment cost between each solution and each vector is computed. Finally, the LAP is solved, and the assigned individuals are selected. The GS-NSGA-III algorithm was compared to NSGA-III [20], MOEA/D [15], GrEA [69], and Hype [66] using the DTLZ [64] test suite with 8, 10, and 15 objectives. The experimental results showed that GS-NSGA-III is competitive in tackling many-objective optimization problems.

Algorithm 3 Global View-based Selection

Require: Reference vectors ($V = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$), and population ($P_t = \{s_1, \ldots, s_N\}$).
Ensure: Selected solutions (P_{t+1})

```
Non-dominated sort of P_t
Assign the corresponding rank r(s) to each solution in P_t
Initialize the cost matrix C ∈ ℝ^{k×N} with zeros
for i ← 1 to k do
    for j ← 1 to N do
        C_{i,j} ← ||s_j − v_i s_j v_i^T/(v_i v_i^T)||
    end for
    a ← min{C_{i,1}, ..., C_{i,N}}
    b ← min{C_{i,1}, ..., C_{i,N}}
    for j ← 1 to N do
        C_{i,j} ← 0.9 × (C_{i,j} − a)/(b − a) + r(s_j)
    end for
end for
P_{t+1} ← Solve the LAP with cost matrix C
```

In 2018, Miguel Antonio et al. [70] proposed an improved version of the HDE called MOEA-LAPS. The main difference with respect to the HDE was that the assignment cost was computed in this case using the ASF scalarizing function. Furthermore, the performance of the MOEA-LAPS was compared with θ-DEA [71], CMA-PAES-HAGA [72], NSGA-III [20], and HDE. They adopted the DTLZ [64] and WFG [67]

test suites, with two to ten objective functions, and the hypervolume indicator was employed as the performance measure. In this case, the experimental results showed that MOEA-LAPS produced the best overall results. Moreover, it provided good approximations in terms of convergence and distribution of solutions.

We can observe that none of the previous algorithms uses additional mechanisms besides the LAP transformation to achieve accurate approximations. Moreover, these algorithms are able to improve the performance of state-of-the-art algorithms in problems with two to ten objectives. As a result, the LAP transformation has proven to be an exceptional alternative selection method that deserves further investigation.

3.6 Summary

This chapter introduced all the details of transforming the MOEA's selection process into a LAP. First, we described the definition of an assignment problem and some ways to represent an assignment. After that, we presented the formal definition of a LAP and the best well-known algorithm to solve it, the Kuhn-Munkres algorithm. Then, we described the LAP transformation and presented an example of its behavior. Finally, we discussed five algorithms that implemented the LAP transformation and showed that this selection scheme is a promising alternative selection method that deserves further studies.

Chapter 4

On the design of selection schemes using the LAP

This chapter introduces two novel selection schemes that employ the LAP transformation. Section 4.1 explores the simultaneous use of different scalarizing functions and weight vectors in the LAP selection scheme, where the resulting algorithm is called ESW. Section 4.2 incorporates the hypervolume indicator in the LAP selection scheme to address their drawbacks. Moreover, the resulting scheme is incorporated into a novel algorithm called MOEA-LAPCO. Finally, a summary of this chapter is presented in Section 4.3.

4.1 Simultaneous use of scalarizing functions and weight vectors in the LAP selection scheme

The famous No Free Lunch Theorem (NFL) for search [73] states that no single search algorithm can outperform all the others in all types of problems. Therefore, some attempts have been made to combine techniques as a mechanism to generalize a search engine. The ensembles are examples of these algorithms. They contain several populations that evolve simultaneously with different types of techniques. Within each population, parents compete with their offspring and the offspring of other populations to improve diversity [74].

Some researchers have provided evidence about the benefits of adopting several scalarizing functions simultaneously within an MOEA. In 2010, Ishibuchi et al. [75] proposed a variant of the MOEA/D that uses the WS and the TCH scalarizing function at the same time. They evaluated this new algorithm in variants of the knapsack problem and concluded that the simultaneous use of both scalarizing functions outperforms their individual use. On the other hand, in 2017, Hernández Gómez and Coello Coello [76] proposed a new algorithm called MOMBI-III that incorporated seven different scalarizing functions. The core idea was to use these functions and the s-energy indicator to rank individuals for survival selection. The experiments showed

that MOMBI-III was able to outperform the individual use of each function as well as state-of-the-art algorithms in many-objective problems. In spite of this evidence, this sort of mechanism has not been explored so far for the LAP selection scheme.

This section presents an ensemble of scalarizing functions and weight vectors using the LAP transformation for the survival selection mechanism of an MOEA. For this purpose, we choose an MOEA that uses this selection scheme (the HDE) and evaluate its performance using six different scalarizing functions and two weight vector generators (Subsection 4.1.1). Then, we propose an ensemble of the pairs (scalarizing function + weight vectors) with the best performance to obtain a more powerful MOEA (Subsection 4.1.2). Then, we assess the performance of the new MOEA and experimentally show that our new approach outperforms individual functions and state-of-the-art algorithms (Subsection 4.1.3). Finally, we present a summary of the section (Subsection 4.1.4).

4.1.1 Influence of the scalarizing functions and weight vectors in the HDE

Initially, the HDE was implemented with the TCH scalarizing function [5], and later on, it was found that the use of the ASF function was able to improve its performance [70]. Hence, it is evident that the type of scalarizing function adopted in HDE has an impact on its performance. Furthermore, only the UDH method has been tested in HDE, and we do not know if other methods could improve its performance. In this section, we present an experimental study that analyzes the behavior of HDE using different scalarizing functions and weight vectors.

Experimental setup

We tested the HDE algorithm using six different scalarizing functions: TCH, ATCH, ASF, AASF, PBI, and AGSF2. We selected $\theta = 5$ for PBI, $\alpha = 10^{-4}$ for AASF and $\alpha = 0.005$ for ATCH. Besides the scalarizing functions, we tested the UDH and the SLD methods for generating weight vectors. For the case of the SLD vectors, we adopted $H = 14$. We performed 30 independent runs of each scalarizing function with each weight vector set. The parameters used in HDE in all cases were $F = 1.0$, $C_r = 0.4$, $g_{max} = 300$ and a population size of 120.

We adopted the DTLZ1 - DTLZ7 problems from the DTLZ test suite [64], the WFG1 - WFG9 problems from the WFG test suite [67], and the Minus-DTLZ and Minus-WFG test problems from [77]. The number of variables in DTLZ is defined by $n = 3 + k - 1$ where $k = 5$ for DTLZ1, $k = 10$ for DTLZ2-DTLZ6 and $k = 20$ for DTLZ7. The same was applied to Minus-DTLZ. Regarding the WFG and Minus-WFG, we set the position parameters to $k = 4$ and the distance parameters to $l = 20$. All of these problems were adopted with three objectives.

We want to analyze which pair of scalarizing functions and weight vector sets have the best overall performance. Hence, we employed the hypervolume indicator [27] for the evaluation of the algorithms. This indicator measures the size of the objective

space covered by the solution set, providing us with a way to measure the convergence and diversity of an approximation.

Experimental results and discussion

The average and standard deviation of the 30 independent runs are shown in Tables B.1 and B.2. The two best values of each problem are highlighted in gray, where the darker tone corresponds to the best value. In addition, the "*" symbol means that the result is statistically significant using Wilcoxon's rank-sum test with a significance level of 5%.

In 81.25% of the conventional test problems (WFG and DTLZ), the best performance scalarizing functions were the ones that adopted the SLD method. However, in the minus problems (WFG-Minus and DTLZ-Minus), the UDH method's functions had the best performance in 93.75% of the problems.

The function with the best performance in the conventional problems was AGSF2, which won in 56.25% of the problems. It was followed by the AASF which won in 25% of the problems. Regarding the minus problems, AGSF2 had the best performance with 62.5%, followed by AASF with 18.75%.

From these experiments, we can conclude that the SLD method is the best choice for the conventional test suites (WFG and DTLZ) and that the UDH method is a better option for the minus test suites (Minus-WFG and Minus-DTLZ). These results also suggest that the UDH set is less sensitive to the Pareto Front shape than SLD. Moreover, we concluded that AGSF2 and AASF are the recommended scalarizing functions for HDE.

4.1.2 Our proposed approach

We saw in the previous subsection that the performance of HDE depends on the scalarizing function and weight vectors adopted. In particular, the SLD weight vectors had the best performance in conventional problems, while in the minus problems, the UDH weight vectors had the best performance. Furthermore, the best scalarizing functions for all the problems were the AASF and the AGSF2. Therefore, we hypothesized that if we could design an ensemble of different scalarizing functions and weight vectors, we could improve the performance of HDE in all the test suites, turning it into a more general multi-objective optimizer.

Based on our previous discussion, we propose here an *Ensemble of Scalarizing functions and Weight vectors* (ESW). Our proposed approach consists of four different pairs of scalarizing functions and weight vectors. Particularly, we selected AGSF2 with SLD, AASF with UDH, ASF with SLD, and AASF with UDH since they had the best performance in our previous experiment. However, it is evident that other pairs can also be adopted.

Each pair has a subpopulation that generates offspring independently using $DE/rand/1/bin$. Moreover, every population will use its corresponding pair to compute the LAP assignment cost and to select its individuals using the Hungarian algo-

rithm (as in the original HDE). Nevertheless, the parent, its offspring, and the other subpopulation's offspring will be considered in all the selection processes. Therefore, the parents can be replaced by the offspring of other populations. We argue that this information exchange can improve diversity and allows populations to help each other.

The solutions' distribution and convergence speed between subpopulations may differ because each has a different scalarizing function and weight vectors. Therefore, we include an external archive that stores the non-dominated solutions for merging the information collected by the subpopulations. If the archive exceeds a predefined size, the solution with the worst contribution of s-energy [28] is deleted. Algorithm 4 displays the pseudocode of ESW, and its flowchart is presented in Figure 4.1.

Algorithm 4 Ensemble of Scalarizing functions and Weight vectors (ESW)

Require: Max archive size (A_{max}), stopping condition, subpopulation size (n), parameters C_r and F for DE/$rand$/1/bin
Ensure: External Archive (A)

$w_{\text{SLD}} \leftarrow$ Generate weight vectors of size n using SLD
$w_{\text{UDH}} \leftarrow$ Generate weight vectors of size n using UDH
$W = \{w_{\text{SLD}}, w_{\text{UDH}}, w_{\text{SLD}}, w_{\text{UDH}}\}$
$SF = \{\text{AGSF2}, \text{AGSF2}, \text{AASF}, \text{AASF}\}$
Generate initial population POP_i randomly, $\forall i = \{1, ..., 4\}$
Evaluate populations
$A \leftarrow$Obtain the non-dominated solutions in the populations
while the stopping condition is not fulfilled **do**
    for $i \leftarrow 1$ to 4 **do**
        $\text{OFF}_i \leftarrow$ Generate offspring from POP_i using DE/$rand$/1/bin
    end for
    Evaluate offspring
    Insert offspring into A using Pareto dominance
    if $|A| > A_{max}$ **then**
        Remove the individual with worst s-energy contribution until $|A| = A_{max}$
    end if
    for $i \leftarrow 1$ to 4 **do**
        $Q_i \leftarrow \bigcup_{j=1}^{4} \text{OFF}_j \cup \text{POP}_i$
        $NQ_i \leftarrow$Normalize objectives of Q_i
        $C \leftarrow$ Construct a cost matrix using NQ_i, SF_i and W_i
        $I_i \leftarrow$ Obtain the best assignment in C_i using the Hungarian method
        $\text{POP}_i \leftarrow \{\boldsymbol{x}_j | j \in I, \boldsymbol{x}_j \in Q_i\}$
    end for
end while

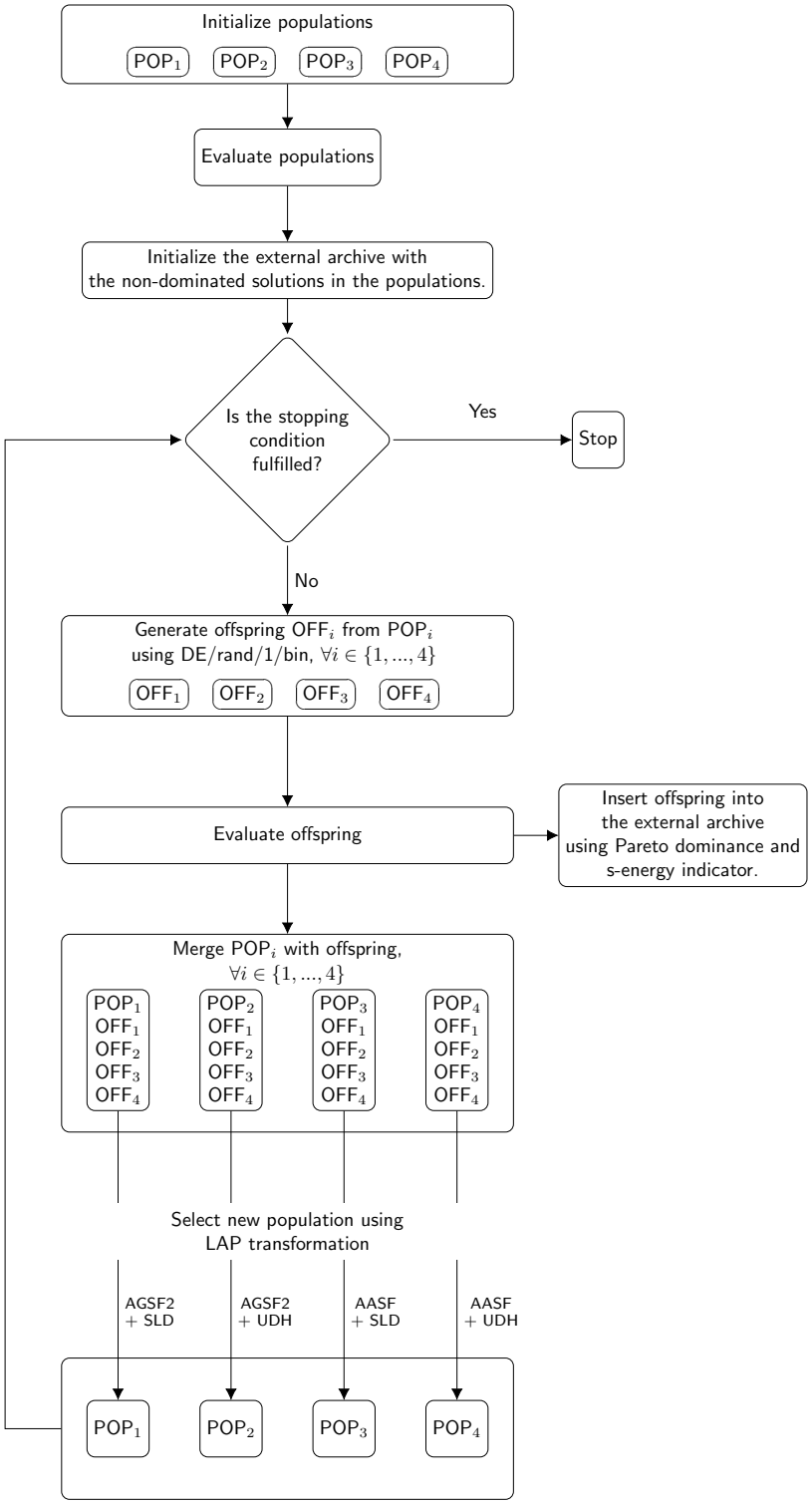


Figure 4.1: General flowchart of ESW

Table 4.1: Parameters of the algorithms used in experiment 1

Algorithms	Parameters settings
HDE (in all versions)	$F = 1.0$, $C_r = 0.4$, $g_{max} = 1100$, $n = 120$
ESW	$F = 1.0$, $C_r = 0.4$, max function evaluations = 132000, $n = 55$, max archive size = 120

4.1.3 Experimental analysis

In this subsection, we validate the performance of ESW using two experiments. The first experiment compares ESW with HDE using the pairs (scalarizing function + weight vectors) separately. We present this experiment in subsection 4.1.3. The second experiment compares ESW with state-of-the-art algorithms, and it is presented in subsection 4.1.3. In both cases, we use the hypervolume and the s-energy indicator for performance assessment. In all the tables, the two best values of each problem are highlighted in gray, where the darker tone indicates the best value. In addition, the "*" symbol means that the result is statistically significant using Wilcoxon's rank-sum test with a significance level of 5%.

Comparison with standalone pairs

For this comparison, we performed 30 independent runs of ESW and HDE with the following separate pairs: AGSF2 with SLD, AGSF2 with UDH, AASF with SLD, and AASF with UDH. We adopted the benchmark problems (including their configuration), the scalarizing functions' parameters and the weight vectors' parameters from Section 4.1.1. Furthermore, Table 4.1 displays the parameters of the algorithms used in this experiment.

We present the results of the experiment in Tables B.3 and B.4. Regarding the hypervolume indicator, our proposed approach outperforms the other algorithms in 84% of the problems. Regarding the s-energy indicator, our approach has the best performance in 93% of the problems. Therefore, we can conclude that ESW outperforms HDE in almost all the test problems adopted.

Comparison with respect to state-of-the-art algorithms

We selected three well-known algorithms for our experimental study at the second stage: NSGA-III [20], MOEA/DD [78], and SMS-EMOA [50]. In this case, we tested many-objective problems. For this sake, we adopted a version of SMS-EMOA that uses the algorithm of HYPE [66] to approximate the hypervolume contributions when dealing with problems having more than three objectives.

We used the DTLZ1-DTLZ4 and DTLZ7 problems from the DTLZ test suite.

Table 4.2: Number of SLD partitions used by NSGA-III and MOEA/DD

Objectives (m)	3	5	7	10
Number of partitions (H)	14	6	5	2,3

Table 4.3: Number of SLD partitions and subpopulation sizes used by ESW

Objectives (m)	3	5	7	10
Number of partitions (H)	9	4	3	2,3
Subpopulation sizes (n)	55	70	84	275

Moreover, we adopted the minus versions of the same problems. We set the number of objectives (m) to: 3, 5, 7, and 10. The number of variables was set to $n = k+m-1$, where k takes the same values as indicated in Section 4.1.1.

We selected the SLD method to generate the weight vectors of MOEA/DD and NSGA-III. For problems having ten objectives, we used the two-layer approach proposed in [20] to generate the weight vectors. Table 4.2 displays the corresponding H values for each objective. In the case of ESW, Table 4.3 displays the subpopulation sizes given the H values. We also used the two-layer approach for ten objectives. Table 4.4 shows the population size (or maximum archive size) and the maximum number of function evaluations that the algorithms used.

The parameters of SBX and polynomial-based mutation were set to $pc = 1.0$, $pm = 1/n$, $\eta_c = 30$ and $\eta_m = 20$. We set the parameters of the DE operator to $F = 1.0$ and $Cr = 0.4$. MOEA/DD also used a neighborhood size $T = 20$, a neighborhood selection probability $\delta = 0.9$, and the PBI scalarizing function with $\theta = 5$.

Table B.5 shows the average and standard deviation of the hypervolume values. Regarding the DTLZ problems, MOEA/DD had the best performance since it is ranked first place in 10 out of 20 instances. The algorithm with the second-best performance was SMS-EMOA with nine instances, and the third-best algorithm was ESW with seven instances. In the case of the Minus-DTLZ problems, the best algorithm is SMS-EMOA that obtained the first place in 13 out of 20 instances. The second-best was ESW, with seven instances. NSGA-III and MOEA/DD did not obtain first places in this case. We can observe from these results that the performance of SMS-EMOA and ESW does not depend on the Pareto front shape since both can obtain first places in all test suites. This is not the case of MOEA/DD that had the best performance in almost all the DTLZ problems. However, it could not obtain the

Table 4.4: General parameters used in experiment 2

Number of objectives (m)	3	5	7	10
Population size (or archive size)	120	210	210	276
Max function evaluations	132000	231000	231000	303600

first place in the remaining problems. In general, the best algorithm regarding the hypervolume indicator was SMS-EMOA with 22 out of 40 instances in the first place. The second best algorithm was our proposed ESW, which ranked first in 14 out of 40 instances.

On the other hand, Table B.6 shows the average and standard deviation of the s-energy values. For the DTLZ problems, ESW had the best performance, with 13 out of 20 instances in the first place. MOEA/DD had the second-best performance since it obtained the first place in 7 out of 20 instances. In the Minus-DTLZ problems, ESW also outperforms the other algorithms with 18 out of 20 instances in the first place, followed by MOEA/DD with two instances. We can see that ESW outperforms the other algorithms in almost all the problem instances regarding the s-energy indicator.

In summary, we can conclude that ESW is a competitive approach with respect to state-of-the-art algorithms.

4.1.4 Discussion

In this section, we evaluate the influence of scalarizing functions and weight vectors on the performance of HDE. We conclude that the SLD weight vectors perform better in conventional problems (DTLZ and WFG) and the UDH weight vectors perform better in minus problems (Minus-WFG and Minus-DTLZ). These results suggest that the UDH problems are less sensitive to the Pareto Front shape than the SLD vectors. Moreover, the recommended scalarizing functions for HDE are the AGSF2 and the AASF. In addition, we proposed a new ensemble algorithm (called ESW) using the best pairs of scalarizing functions and weight vectors that had the best performance in the previous experiment. The experimental results show that ESW outperforms HDE and is competitive with respect to state-of-the-art algorithms.

4.2 Integrating a performance indicator into the LAP selection scheme

It has been shown that the LAP selection scheme is an excellent alternative to the standard selection schemes adopted by Multi-Objective Evolutionary Algorithms (MOEAs). However, we identified two critical issues in its operation: it occasionally selects duplicated solutions and it does not always prefer non-dominated solutions. This chapter introduces a novel selection scheme that combines the LAP transformation with the hypervolume indicator in order to overcome the previous drawbacks. However, since the computation of the hypervolume indicator is expensive, we adopted an approximation scheme that uses a polar coordinates transformation [55]. In addition, we incorporate our proposed selection mechanism into an MOEA, called Multi-Objective Evolutionary Algorithm Based on the Linear Assignment Problem and the Hypervolume Approximation using Polar Coordinates (MOEA-LAPCO). Our experimental results show that MOEA-LAPCO outperforms HDE and state-of-the-art al-

Table 4.5: Example of a case where the Hungarian algorithm selects duplicated solutions. The individuals I_1 and I_2 have the same value and the best assignment cost (highlighted in gray). Therefore, these solutions will be assigned and selected for the next generation.

C	I_1	I_2	I_3	I_4
w_1	0	0	333333.33	1000000.0
w_2	0	0	1000000.0	333333.33

gorithms.

This section is organized as follows. Subsection 4.2.1 explains the drawbacks of the LAP selection scheme. Subsection 4.2.2 introduces our proposed approach, and Subsection 4.2.3 evaluates it through an experimental study. Finally, Subsection 4.2.4 presents a summary of this section.

4.2.1 Drawbacks of the LAP selection scheme

Although it has been shown that the LAP transformation is a promising selection scheme [5, 70], we have found two critical drawbacks in its operation. First, we identified that it selects duplicated individuals. This problem arises because the assignment costs are the same for repeated solutions. Therefore, if the solutions have the best cost for different weight vectors, the Hungarian algorithm will prefer them. For instance, let's assume that we want to select two elements from a set of four individuals such that $\boldsymbol{I_1} = \boldsymbol{I_2}$, $\boldsymbol{f}(\boldsymbol{I_1}) = \boldsymbol{f}(\boldsymbol{I_2}) = [1, 2]^T$, $\boldsymbol{f}(\boldsymbol{I_3}) = [4, 3]^T$, and $\boldsymbol{f}(\boldsymbol{I_4}) = [2, 5]^T$. Moreover, we consider two weight vectors, $\boldsymbol{w_1} = [1, 0]^T$ and $\boldsymbol{w_2} = [0, 1]^T$. Accordingly, their assignment costs using the ASF function are displayed in Table 4.5. We can observe that individuals 1 and 2 have the best assignment costs for all the weight vectors. Hence, the Hungarian algorithm will assign either $\boldsymbol{w_1}$ to $\boldsymbol{I_1}$ and $\boldsymbol{w_2}$ to $\boldsymbol{I_2}$ or vice versa.

The second drawback is that the LAP transformation occasionally prefers weakly-dominated solutions over non-dominated ones. To illustrate this fact, we executed HDE over 100 generations with a population size of 120 individuals, using the WFG7 problem [67]. Figure 4.2 displays, for each generation, the number of non-dominated solutions available in the population of parents and their offspring. Moreover, it shows the number of weakly dominated solutions selected by the LAP selection scheme. We can see in Figure 4.2 that even though, on some occasions, there are more than 120 non-dominated solutions available, HDE still selects 20 or more weakly dominated solutions.

This issue occurs when weakly dominated (or even dominated) solutions are closer to non-covered vectors, while some non-dominated solutions are in crowded vectors. Therefore, the costs of assigning the non-covered vectors to the former solutions are lower than those of the latter. To illustrate this situation, let us suppose we want to assign the four weight vectors to the five individuals shown in Figure 4.3. Further-

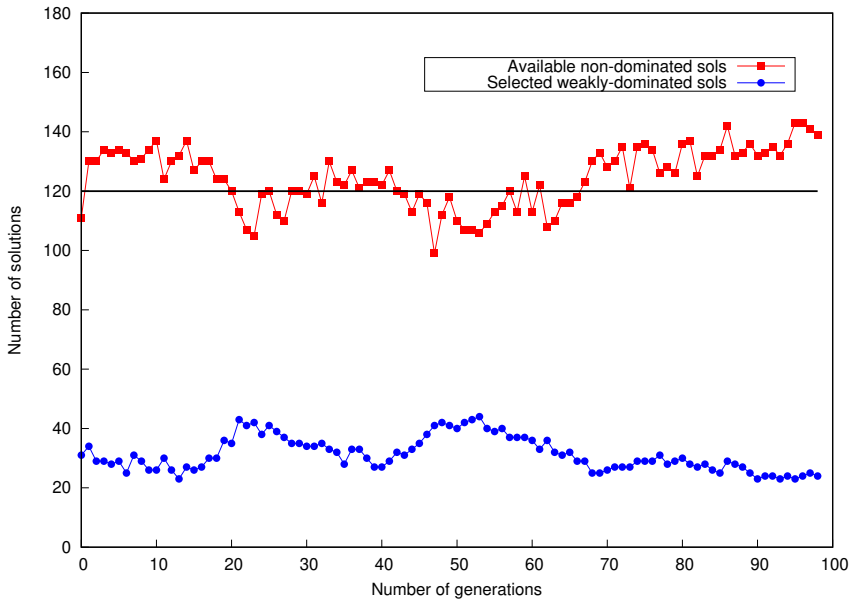


Figure 4.2: Execution of the HDE during 100 generations using the WFG7 problem. Squares represent the number of available non-dominated solutions, and circles represent the number of selected weakly-dominated solutions.

more, Table 4.6 shows the assignment cost using the ASF function for this problem. We can observe that all the individuals are non-dominated except for individual I_5, which is also the closest to vector w_3. Moreover, individual I_4 is located in a crowded area far from vector w_3. Consequently, the cost of assigning vector w_3 to individual I_5 is lower than that of assigning w_3 to individual I_4. Hence, in the best assignment (which is gray-colored in Table 4.6), the individual I_5 is preferred over I_4.

Table 4.6: Example of a cost matrix where a dominated individual (I_5) is preferred over a non-dominated one (I_4). The best assignment is highlighted in gray.

C	I_1	I_2	I_3	I_4	I_5
w_1	1.0	428571.43	857142.86	285714.29	1000000.0
w_2	1.5	1.29	2.57	1.38	3.0
w_3	3.0	2.08	1.29	2.77	1.5
w_4	1000000.0	692307.69	0.86	923076.92	307692.31

4.2.2 Our proposed approach

This subsection proposes incorporating a performance indicator in the LAP selection scheme to solve its disadvantages. For this purpose, we selected the hypervolume indicator because it can differentiate dominated and non-dominated solutions (due to its Pareto compliance). However, we decided to use a polar coordinate approxi-

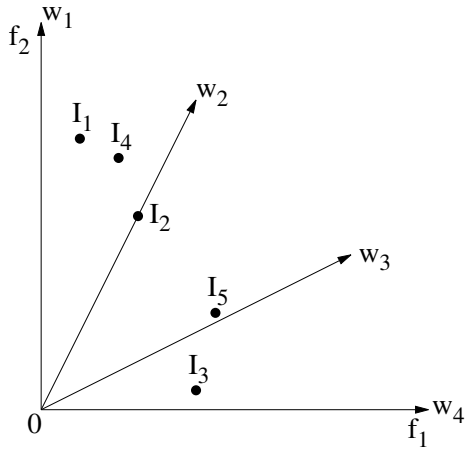


Figure 4.3: Example of a set of individuals where a dominated individual is preferred over a non-dominated one using the LAP selection mechanism.

mation of the hypervolume [55] because of its high computational cost which grows exponentially with the number of objectives.

For the remainder of this section, we first introduce the polar coordinate approximation of the hypervolume (see Section 4.2.2). Then, we describe our proposed selection scheme that combines the hypervolume indicator and the LAP transformation (see Section 4.2.2). Finally, we present the Multi-Objective Evolutionary Algorithm Based on the Linear Assignment Problem and the Hypervolume Approximation using Polar Coordinates (MOEA-LAPCO), which is an MOEA that incorporates the proposed selection scheme (see Section 4.2.2).

Approximating the hypervolume contribution using polar coordinates

The hypervolume indicator (denoted by I_H) measures the size of the objective space covered by a set given a reference point. Let $A \subset \mathbb{R}^k$ and $\boldsymbol{z}^u \in \mathbb{R}^k$ be a reference point dominated by every point in A. Therefore, the I_H of A and $\boldsymbol{z^u}$ can be written as [55]:

$$I_H(A, \boldsymbol{z^u}) = \int_D I_\Omega(\boldsymbol{z}) dz \tag{4.1}$$

where $\boldsymbol{z}^l = (z_1^l, ..., z_k^l)^T$ s.t. $z_i^l = \min\{y_i \mid \boldsymbol{y} = (y_1, ..., y_k)^T \in A\}$, $D = \{\boldsymbol{z} \in \mathbb{R}^k \mid \boldsymbol{z^l} \prec \boldsymbol{z} \prec \boldsymbol{z^u}\}$, $\Omega = \{\boldsymbol{z} \in \mathbb{R}^k \mid \exists \boldsymbol{y} \in A$ such that $\boldsymbol{y} \prec \boldsymbol{z} \prec \boldsymbol{z^u}\}$ and $I_\Omega(\boldsymbol{z})$ is the characteristic function of Ω. Moreover, the hypervolume contribution of a vector $\boldsymbol{y} \in A$ considering A and $\boldsymbol{z^u}$ is defined as $V(\boldsymbol{y}, A, \boldsymbol{z^u}) = I_H(A, \boldsymbol{z^u}) - I_H(A \backslash \{\boldsymbol{y}\}, \boldsymbol{z^u})$.

The computational cost of the hypervolume is prohibitive when the number of objectives is larger than six. To deal with this problem, Deng and Zhang [55] proposed a new method to approximate the hypervolume using polar coordinates. Their idea is to express the hypervolume (displayed in (4.1)) as a $(k-1)-D$ integral using the polar

coordinate system. Deng and Zhang proposed different methods to approximate the hypervolume contribution using the polar coordinates transformation. In this work, we selected the most stable method according to the experimental results reported in [55].

Let $A \subset \mathbb{R}^k$ and $\boldsymbol{z}^u \in \mathbb{R}^k$ be a reference point dominated by every point in A. To compute the hypervolume contribution, this method first generates n uniformly distributed points $\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(n)}\}$ on the $(k-1)-D$ unit sphere in R^m_+ using the Unit Normal Vector Approach. Therefore, each point $\boldsymbol{\theta}^i$ is generated as follows:

$$\boldsymbol{\theta}^{(i)} = \frac{\mid \boldsymbol{x} \mid}{\mid\mid \boldsymbol{x} \mid\mid_2} \text{ where } \boldsymbol{x} \sim \mathcal{N}(0, I_k). \tag{4.2}$$

In addition, a matrix M is constructed, whose (i,j)-entry is the j^{th} largest value in $\{l_{\bar{\boldsymbol{y}}}(\boldsymbol{\theta}^{(i)}) \mid \bar{\boldsymbol{y}} \in A\}$ where

$$l_{\bar{\boldsymbol{y}}}(\boldsymbol{\theta}) = \min_{1 \leq m \leq k} (1/\theta_m)(z^u_m - \bar{y}_m). \tag{4.3}$$

Finally, the contribution $V(\boldsymbol{y}, A, \boldsymbol{z^u})$ is approximated using the following expression:

$$\tilde{V}(\boldsymbol{y}, A, \boldsymbol{z^u}) = \frac{\Phi}{2^k} \frac{1}{kn} \sum_{i=1}^{n} \begin{cases} M^k_{i1} - M^k_{i2} & \text{if } l_{\boldsymbol{y}}(\boldsymbol{\theta}^{(i)}) = M_{i1} \\ 0, & otherwise, \end{cases} \tag{4.4}$$

where $\Phi = [2\pi^{(k/2)}/\Gamma(k/2)]$ is the area of the $(k-1)-D$ unit sphere and $\Gamma(x) = \int_0^\infty z^{x-1}e^{-z}dz$ is the analytic continuation of the factorial function.

Selection process

Our proposed selection scheme (summarized in Algorithm 5) splits the selection process into two phases. During the first phase, we eliminate a percentage $p \in [0, 50]$ of the population by utilizing the LAP transformation. In the second phase, we remove the remaining $50 - p$ percentage using the polar-coordinate approximation of the hypervolume. The idea is to use the hypervolume indicator to increase the preference for the non-dominated solutions.

We have to take some considerations to apply the above mechanism. First, we observed that the approximation of the hypervolume contribution using polar coordinates is extremely sensitive to the reference point adopted (as pointed out by Deng and Zhang [55]). In particular, the algorithm's distribution was poor when dominated solutions were considered for the reference point computation. Therefore, in the selection process, we first obtain the non-dominated solutions on the population (see line 1). If the number of non-dominated solutions is less than n_{pop}, we only employ the LAP transformation to select the individuals for the next generation (see lines 5 to 7). Otherwise, we employ our proposed two-phase process for selecting the individuals (see lines 8 to 18).

Algorithm 5 Select_individuals

Require: Set of weight vectors (w_1), set of weight vectors (w_2), population (Q), uniformly distributed points (θ), population size (n_{pop}), reference point factor (λ), number of objectives (k)

1: $ND \leftarrow$ Obtain the non-dominated solutions from Q
2: $\boldsymbol{z^{max}} = [z_1^{max}, \ldots, z_k^{max}] \quad s.t. \quad z_j^{max} = \max_{x \in ND} f_j(\boldsymbol{x}), \quad j = \{1, \ldots, k\}$
3: $\boldsymbol{z^{min}} = [z_1^{min}, \ldots, z_k^{min}] \quad s.t. \quad z_j^{min} = \min_{x \in ND} f_j(\boldsymbol{x}), \quad j = \{1, \ldots, k\}$
4: $Q' \leftarrow$ Normalize the objective functions of Q using $\boldsymbol{z^{max}}$ and $\boldsymbol{z^{min}}$
5: **if** $|ND| \leq n_{pop}$ **then**
6: $\quad C \leftarrow$ Compute the assignment cost using Q' and w_1
7: $\quad I \leftarrow$ Obtain the best assignment in C using the Hungarian algorithm
8: **else**
9: $\quad C \leftarrow$ Compute the assignment cost using Q' and w_2
10: $\quad I_H \leftarrow$ Obtain the best assignment in C using the Hungarian algorithm
11: $\quad I_{ND} \leftarrow$ Obtain the indices of the non-dominated solutions from $A := \{x_i \mid I_H[i] = 1, x_i \in Q'\}$
12: $\quad nd\_size \leftarrow |\{x_i \mid I_{ND}[i] = 1, x_i \in Q'\}|$
13: $\quad$ **if** $nd\_size < n_{pop}$ **then**
14: $\quad\quad I \leftarrow$ Prune_population_with_polar_coordinates(Q', I_H, θ, λ, k)
15: $\quad$ **else**
16: $\quad\quad I \leftarrow$ Prune_population_with_polar_coordinates(Q', I_{ND}, θ, λ, k)
17: $\quad$ **end if**
18: **end if**
19: $P \leftarrow \{x_i \mid I[i] = 1, x_i \in Q\}$
20: return P

In addition, it is essential to note that the LAP transformation does not always prefer non-dominated solutions, even if we have enough solutions of this type. Therefore, in the two-phase process, we compute the number of non-dominated solutions returned by the LAP before initiating the polar-coordinate discarding mechanism (see line 11). If the number of non-dominated solutions is less than n_{pop}, the pruning process will be carried out over the solutions provided by the LAP (see line 14). Otherwise, the pruning process will only be performed on the selected non-dominated solutions, discarding the rest (see line 16). We assume that the duplicated solutions are dominated (except for the first solution to appear). Consequently, the algorithm also removes the duplicated solutions in this phase.

On the other hand, we changed the normalization limits adopted in the original LAP selection scheme. Instead of obtaining the maximum and minimum values from the whole population, we only consider the non-dominated solutions of the population. Moreover, the normalization is performed using these limits in the expression (3.10). See lines 2 to 4.

Regarding the approximation of the hypervolume, we preprocess some information to avoid the recurrence of expensive operations. First, the set θ does not depend on

the population to prune and, therefore, it can be the same for all generations. Hence, we compute this set at the beginning of the MOEA-LAPCO algorithm. Second, the values $l_{\boldsymbol{y}}(\boldsymbol{\theta^{(i)}})$ do not change when a solution is removed. The only information that changes is the matrix ranking. Therefore, at the beginning of the pruning procedure, we compute the $l_{\boldsymbol{y}}(\boldsymbol{\theta^{(i)}})^k$ values and store them in a matrix M. We raise the $l_{\boldsymbol{y}}(\boldsymbol{\theta^{(i)}})$ values to the power k because it does not affect the contribution order and avoids an extra computational cost. Furthermore, we obtain the indices that sort in descending order each row of M. The above procedure is displayed in Algorithm 6.

Algorithm 6 Compute_and_sort_M

Require: Normalized objective vectors (Q'), number of elements in Q' (m), list that handles currently selected solutions (I_S), uniformly distributed points (θ), number of points in θ (n), reference point ($\boldsymbol{z^u}$), number of objectives (k)

1: Initialize matrix M of size $n \times m$ with zeros
2: Initialize matrix I_S of size $n \times m$ with zeros
3: **for** $i = 1$ to n **do**
4: $\quad c_{im} \leftarrow 0$
5: $\quad$ **for** $j = 1$ to m **do**
6: $\quad\quad$ **if** $I_S[j] = 1$ **then**
7: $\quad\quad\quad \bar{y} \leftarrow Q'[j]$
8: $\quad\quad\quad l_{\bar{y}}(\boldsymbol{\theta^{(i)}}) \leftarrow \min_{1 \leq l \leq k}(1/\theta_l^{(i)})(z_l^u - \bar{y}_l)$
9: $\quad\quad\quad M[i][j] = (l_{\bar{y}}(\boldsymbol{\theta^{(i)}}))^k$
10: $\quad\quad\quad c_{im} \leftarrow c_{im} + 1$
11: $\quad\quad\quad I_M[i][c_{im}] = j$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad$ Sort the first c_{im} elements of $I_M[i]$ in descending order such that $I_M[i][x]$ is bigger than $I_M[i][y]$ when $M[i][I_M[i][x]] > M[i][I_M[i][y]]$.
15: **end for**
16: return M, I_M, c_{im}

Adopting the previous considerations, Algorithm 7 displays the procedure to approximate the hypervolume contribution. First, we initialize the array C of contributions with zeros. Then, we find the best and second-best solution indices for each i^{th}-point in θ. Since we have the matrix I_M, we only have to go through the list $I_M[i]$ to find the first two still selected individuals (see lines 3 and 4). After that, we obtain the best and second-best elements from M and compute their difference (see lines 5 and 6). Then, we go through the $I_M[i]$ list starting from the index of the best individual, add the difference to the currently selected individuals, and stop when the value $M[i][I_M[j]]$ is different from the best individual (see lines 7 to 18). At the end of the iterations, we multiply the contributions by $\frac{\Phi}{2^k}\frac{1}{kn}$ as in equation (4.4).

Finally, the pruning procedure using the polar coordinate approximation is displayed in Algorithm 8. The first step is to obtain the reference point from the currently

Algorithm 7 Compute_contribution

Require: matrix (M), indices that sort the solutions in M (I_M), cols of I_M (c_{IM}), list that handles currently selected solutions (I_S), number of objectives (k), number of rows in M (r), number of cols in M (c)

$C \leftarrow$Initialize array of size c with zeros
for $i = 1$ to r **do**
  $jbest \leftarrow \min_{j=1,\ldots,c_{IM}} j \quad s.t. \quad I_S[I_M[i][j]] = 1$
  $jsbest \leftarrow \min_{j=1,\ldots,c_{IM}} j \quad s.t. \quad I_S[I_M[i][j]] = 1 \wedge j! = jbest$
  $best \leftarrow M[i][I_M[i][jbest]]$
  $diff \leftarrow best - M[i][I_M[i][jsbest]]$
  $j \leftarrow jbest$
  while $j < c$ **do**
    $idx \leftarrow I_M[j]$
    if $best = M[i][idx]$ **then**
      if $I_S[idx] = 1$ **then**
        $C[idx] = C[idx] + diff$
      end if
    else
      break
    end if
    $j \leftarrow j + 1$
  end while
end for
for $j = 1$ to c **do**
  $C[j] = \frac{\Phi}{2^k} \frac{1}{kn} * C[j]$
end for
return C

selected individuals. Then, we compute and sort the matrix M using Algorithm 6. Then, we compute the contribution of each individual in the population using Algorithm 7 and remove the one with the lower contribution. We repeat the above procedure until the population size is equal to n_{pop}.

Algorithm 8 Prune_population_with_polar_coordinates

Require: Normalized objective vectors (Q'), list that handles currently selected solutions (I_S), uniformly distributed points (θ), reference point factor(λ), number of objectives (k)

$\boldsymbol{z} \leftarrow$ Find the maximum value of each objective from $A := \{x_i \mid I_S[i] = 1, x_i \in Q'\}$
$\boldsymbol{z^u} \leftarrow \lambda * \boldsymbol{z}$
$r_M \leftarrow |\theta|$
$c_M \leftarrow |Q'|$
$M, I_M, c_{IM} \leftarrow$ Compute_and_sort_M(Q', c_M, I_S, θ, r_M, $\boldsymbol{z^u}$, k)
$num\_sel = c_{IM}$
while $num\_sel > n_{pop}$ **do**
    $C \leftarrow$ Compute_contribution(M, I_M, c_{IM}, I_S, k, r_M, c_M)
    $idx \leftarrow \arg\min_{i=1,\cdots,c_M} C[i] \quad s.t. \quad I_S[i] = 1$
    $I_S[idx] = 0$
    $num\_sel = num\_sel - 1$
end while
return I_S

MOEA-LAPCO

In this section, we incorporate our proposed selection scheme into an MOEA. The resulting approach was called MOEA-LAPCO and is displayed in Algorithm 9. First, two sets of weight vectors (w_1 and w_2) are generated using the UDH method. The set w_1 is used when only the LAP transformation is applied, and the set w_2 is used when the two-phase selection is performed. Thus, the size of w_2 depends on the parameter p and is calculated as $|w_2| = \frac{(100-p)*n_{pop}*2}{100}$. Then, the set θ of uniformly distributed points is created using equation (4.2). Then, the initial population is generated and evaluated. In the main loop of the algorithm, a new population is generated using the SBX and PM operators [23]. After that, the individuals that persist for the next generation are selected using Algorithm 5, and this procedure continues until a maximum number of evaluations is reached.

4.2.3 Experimental analysis

We evaluated the performance of MOEA-LAPCO with respect to state-of-the-art algorithms. For this purpose, we performed 30 independent runs of HDE [5], HDE with SBX and PM, MOEA/DD [78], NSGA-III [20], and our proposed algorithm. We adopted the WFG1-WFG9 problems from the WFG test suite [67] using 3, 5, 8, and

Algorithm 9 MOEA-LAPCO

Require: Multi-objective problem, population size (n_{pop}), number of uniformly distributed points (n_{hv}), maximum number of evaluations, variation operators' parameters, reference point factor (λ), percentage of solutions to be discarded using the LAP transformation (p)

Ensure: P

1: $w1 \leftarrow$ Generate n_{pop} weights vectors using UDH
2: $w2 \leftarrow$ Generate $\frac{(100-p)*n_{pop}*2}{100}$ weights vectors using UDH
3: $\boldsymbol{\theta} \leftarrow$ Generate n_{hv} uniformly distributed points using equation (4.2)
4: Generate initial population P
5: Evaluate population P
6: **while** the maximum number of evaluations is not reached **do**
7: $\quad P' \leftarrow$ Generate from P the new population using variation operators
8: $\quad$ Evaluate population P'
9: $\quad Q \leftarrow P \cup P'$
10: $\quad P \leftarrow$ Select_individuals(w_1, w_2, Q, θ, n_{pop}, λ, p)
11: **end while**
12: return P

10 objectives. The position-related parameters were set to $m = 2 \times (k-1)$ where k is the number of objectives, the distance-related parameters were set to $l = 20$, and the number of variables to $n = m + l$. Finally, we used the hypervolume indicator for the performance assessment.

Regarding the variation operators, SBX and PM, we set $pc = 0.9$, $pm = 1/n$, $\eta_c = 20$ and $\eta_m = 20$. Furthermore, we set the parameters of DE to $F = 1.0$ and $Cr = 0.4$. In the case of the weight vectors of the MOEA/DD and NSGA-III, we adopted Das and Dennis' approach with the two-layer technique used in the NSGA-III for more than five objectives [20]. Concerning the MOEA/DD's parameters, we set $T = 20$, $\delta = 0.9$, and we used the PBI function with $\theta = 5$. In the case of MOEA-LAPCO, we set $n_{hv} = 10000$, $\delta = 1.5$, and $p = 25$. Furthermore, we used the AASF function with $\alpha = 0.0001$ in both versions of HDE and in the MOEA-LAPCO. Considering the parameters for all the algorithms, we used a population size of 120 for three objectives, 210 for five, 156 for eight, and 276 for ten. In the case of the maximum number of evaluations, we used the population size times 1000 in all cases regardless of the number of objectives.

Table 4.7 shows the average and the standard deviation of the hypervolume's values over 30 generations of each algorithm. The best averages are highlighted in dark gray, and the second-bests are highlighted in light gray. Moreover, the symbol "*" indicates that the algorithm is statistically better than the others employing the Wilcoxon rank-sum test with a significance level of 5%.

We can observe that MOEA-LAPCO is better than HDE with DE and SBX+PM in almost all the problems, indicating that the new mechanism improves the original

versions. On the other hand, the MOEA-LAPCO is better than all the algorithms in 20 out of 36 problems. Remarkably, it is the best in problems WFG3, WFG5, WFG6, WFG8, and WFG9 using 3, 5, and 8 objectives. Moreover, it is the best in the WFG4 and WFG7 problems using 3 and 5 objectives. However, we can notice that it is not the best in any of the problems with ten objectives, suggesting that the algorithm's performance degrades with more than eight objectives. We believe that this happens because the reference point selection mechanism is not good enough for many-objective problems.

Table 4.7: Average and standard deviation of the hypervolume indicator over 30 generations of MOEA-LAPCO and state-of-the-art algorithms. The best values are highlighted in dark gray, and the second-best values are shown in light gray. The symbol "*" indicates that the algorithm is better than the others in a statistically significant.

	k	HDE_DE	HDE_SBX+PM	MOEA-LAPCO	MOEA/DD	NSGA-III
WFG1	3	8.031e-1 (9.0e-3)	8.963e-1 (2.2e-2)	9.214e-1 (2.5e-2)	*1.218e+0 (3.2e-2)	7.749e-1 (3.7e-2)
	5	9.095e-1 (9.5e-3)	1.103e+0 (1.6e-2)	1.157e+0 (1.5e-2)	*1.454e+0 (5.4e-2)	8.978e-1 (3.5e-2)
	8	1.245e+0 (1.7e-2)	1.691e+0 (2.7e-2)	1.746e+0 (1.8e-2)	*1.915e+0 (1.1e-1)	1.384e+0 (1.1e-1)
	10	1.446e+0 (1.8e-2)	2.029e+0 (2.2e-2)	2.077e+0 (2.7e-2)	*2.297e+0 (1.1e-1)	1.958e+0 (1.5e-1)
WFG2	3	1.234e+0 (2.9e-3)	1.189e+0 (8.7e-2)	*1.194e+0 (8.8e-2)	1.168e+0 (8.9e-2)	1.153e+0 (8.9e-2)
	5	1.548e+0 (4.2e-3)	1.572e+0 (6.9e-2)	*1.586e+0 (7.1e-2)	1.537e+0 (6.6e-2)	1.523e+0 (8.2e-2)
	8	*2.134e+0 (1.8e-2)	1.977e+0 (1.7e-1)	2.065e+0 (1.2e-1)	1.957e+0 (9.9e-2)	1.939e+0 (1.5e-1)
	10	*2.591e+0 (1.2e-3)	2.525e+0 (1.3e-1)	2.546e+0 (2.5e-2)	2.292e+0 (3.2e-2)	2.428e+0 (1.1e-1)
WFG3	3	9.105e-1 (3.9e-3)	9.387e-1 (2.1e-3)	*9.466e-1 (1.5e-3)	8.932e-1 (6.0e-3)	9.014e-1 (5.4e-3)
	5	1.109e+0 (5.9e-3)	1.163e+0 (6.5e-3)	*1.186e+0 (6.0e-3)	1.039e+0 (9.5e-3)	1.059e+0 (1.1e-2)
	8	1.426e+0 (7.7e-3)	1.457e+0 (1.2e-2)	*1.494e+0 (2.3e-2)	1.16e+0 (2.2e-2)	1.263e+0 (2.6e-2)
	10	1.694e+0 (8.5e-3)	*1.724e+0 (1.1e-2)	1.703e+0 (3.1e-2)	1.279e+0 (2.0e-2)	1.576e+0 (2.8e-2)
WFG4	3	7.303e-1 (3.9e-3)	7.79e-1 (2.1e-3)	*7.983e-1 (9.6e-4)	7.784e-1 (1.6e-3)	7.565e-1 (3.0e-3)
	5	1.191e+0 (6.3e-3)	1.267e+0 (3.7e-3)	*1.334e+0 (2.4e-3)	1.286e+0 (4.3e-3)	1.210e+0 (8.7e-3)
	8	1.706e+0 (1.5e-2)	1.468e+0 (1.1e-1)	1.54e+0 (4.2e-2)	*1.736e+0 (2.5e-2)	1.595e+0 (4.1e-2)
	10	*2.180e+0 (1.6e-2)	1.972e+0 (9.3e-2)	1.893e+0 (3.6e-2)	2.097e+0 (4.1e-2)	1.959e+0 (3.8e-2)
WFG5	3	7.368e-1 (2.3e-3)	7.419e-1 (4.7e-3)	*7.667e-1 (3.2e-3)	7.434e-1 (3.8e-3)	7.321e-1 (5.1e-3)
	5	1.251e+0 (3.2e-3)	1.237e+0 (3.4e-3)	*1.319e+0 (2.8e-3)	1.258e+0 (3.8e-3)	1.228e+0 (5.3e-3)
	8	1.443e+0 (2.4e-1)	1.640e+0 (1.1e-1)	*1.853e+0 (9.4e-2)	1.681e+0 (2.8e-2)	1.726e+0 (2.5e-2)
	10	1.821e+0 (2.1e-2)	1.964e+0 (3.3e-2)	1.845e+0 (5.7e-2)	2.046e+0 (4.3e-2)	*2.137e+0 (2.6e-2)
WFG6	3	7.056e-1 (6.9e-4)	7.59e-1 (6.4e-3)	*7.783e-1 (5.7e-3)	7.541e-1 (6.2e-3)	7.377e-1 (8.4e-3)
	5	1.206e+0 (1.4e-3)	1.25e+0 (9.5e-3)	*1.314e+0 (8.7e-3)	1.254e+0 (1.1e-2)	1.217e+0 (1.2e-2)
	8	1.802e+0 (2.6e-3)	1.790e+0 (2.5e-2)	*1.934e+0 (3.1e-2)	1.765e+0 (2.8e-2)	1.737e+0 (3.5e-2)
	10	*2.285e+0 (1.4e-3)	2.068e+0 (5.5e-2)	2.143e+0 (9.4e-2)	2.140e+0 (3.9e-2)	2.172e+0 (3.4e-2)
WFG7	3	7.507e-1 (2.2e-3)	7.711e-1 (1.1e-3)	*7.886e-1 (5.7e-4)	7.727e-1 (1.3e-3)	7.599e-1 (2.7e-3)
	5	1.207e+0 (6.1e-3)	1.266e+0 (3.3e-3)	*1.336e+0 (1.3e-3)	1.299e+0 (3.6e-3)	1.245e+0 (1.1e-2)
	8	1.747e+0 (1.9e-2)	1.542e+0 (9.3e-2)	1.764e+0 (1.6e-1)	*1.867e+0 (1.3e-2)	1.709e+0 (3.5e-2)
	10	2.242e+0 (1.6e-2)	2.08e+0 (5.4e-2)	1.95e+0 (8.e-2)	*2.263e+0 (1.7e-1)	2.169e+0 (3.4e-2)
WFG8	3	8.525e-1 (4.8e-3)	8.979e-1 (2.1e-3)	*9.195e-1 (1.3e-3)	9.007e-1 (2.1e-3)	8.717e-1 (5.3e-3)
	5	1.140e+0 (7.1e-3)	1.33e+0 (1.2e-2)	*1.375e+0 (2.e-2)	1.268e+0 (1.2e-2)	1.175e+0 (9.2e-3)
	8	1.598e+0 (1.8e-2)	1.596e+0 (8.1e-2)	*1.874e+0 (9.5e-2)	1.755e+0 (6.4e-2)	1.528e+0 (3.3e-2)
	10	2.116e+0 (1.3e-2)	1.988e+0 (4.3e-2)	1.990e+0 (7.4e-2)	*2.216e+0 (6.1e-2)	1.951e+0 (4.6e-2)
WFG9	3	8.626e-1 (1.5e-3)	8.847e-1 (3.2e-2)	*9.303e-1 (3.e-2)	8.965e-1 (3.0e-2)	8.768e-1 (2.e-2)
	5	1.184e+0 (2.7e-3)	1.169e+0 (4.3e-3)	1.211e+0 (2.3e-3)	1.203e+0 (2.9e-2)	1.167e+0 (1.4e-2)
	8	1.804e+0 (9.9e-3)	1.782e+0 (4.4e-2)	*1.844e+0 (7.0e-2)	1.634e+0 (8.9e-2)	1.68e+0 (5.9e-2)
	10	*2.207e+0 (1.1e-2)	2.108e+0 (5.7e-2)	2.010e+0 (7.0e-2)	1.935e+0 (8.6e-2)	2.075e+0 (4.6e-2)

4.2.4 Discussion

In this section, we identified two crucial drawbacks of the LAP selection scheme: occasional selection of duplicated solutions and not always prioritizing non-dominated solutions. Therefore, we proposed combining the LAP transformation with an approximation of the hypervolume indicator using polar coordinates. Then, we proposed a novel algorithm (MOEA-LAPCO) incorporating the resulting selection scheme.

We compared MOEA-LAPCO with HDE, a variant of HDE using SBX and PM, MOEA/DD, and NSGA-III using problems with three to ten objectives. The experi-

mental results showed that the MOEA-LAPCO outperforms the two variants of HDE, showing that our proposed selection scheme improves the LAP selection mechanism. Moreover, MOEA-LAPCO outperforms the other algorithms in most problems. However, MOEA-LAPCO's performance deteriorates when more than eight objectives are used.

4.3 Summary

In this chapter, we presented two new selection methods that use the LAP transformation. The first selection scheme simultaneously uses four different scalarizing functions and two weight vector sets in the LAP selection scheme. The resulting algorithm was called ESW, and the experimental results showed that it outperforms HDE and is competitive with respect to state-of-the-art algorithms. On the other hand, the second selection scheme incorporates an approximation of the hypervolume indicator in the LAP transformation, aiming to solve two of its drawbacks. Moreover, we incorporated the new scheme into a novel algorithm called MOEA-LAPCO. The experiments showed that MOEA-LAPCO outperforms HDE and state-of-the-art algorithms.

Chapter 5

On the design of performance indicators using the LAP

The solution to a multi-objective optimization problem consists of a set of non-dominated solutions which can not be easily evaluated as in the case of single-objective problems. Therefore, the performance assessment of MOEAs is an essential research topic. Over the years, a variety of indicators have been proposed to assess different characteristics of the Pareto front approximations [27, 79, 33]. These characteristics can be convergence, coverage, uniformity, or cardinality of an approximation set.

This chapter presents two indicators that used the LAP to measure the performance of an approximation set: I_{LAP} and D_{LAP}. Both indicators use a set of reference vectors and establish a cost of assigning a solution to one of the vectors. However, the I_{LAP} is designed to estimate the convergence and diversity of an approximation set, and the D_{LAP} is focused on measuring diversity.

This chapter is organized as follows. Section 5.1 describes the I_{LAP} indicator and presents some experiments to evaluate its performance. On the other hand, Section 5.2 presents the D_{LAP} and an experimental analysis to measure its efficacy. Finally, Section 5.3 summarizes the contents of this chapter.

5.1 A performance indicator based on the LAP

One of the most popular performance indicators adopted in evolutionary multi-objective optimization has been the hypervolume [27], which measures the space covered by an approximation set given a reference point. This indicator is Pareto compliant and can assess both convergence and spread of the approximations produced by a MOEA. However, its computational cost becomes unaffordable as the number of objectives increases.

Another commonly used performance indicator is $R2$ [79]. This indicator can assess the convergence and diversity of the solutions by using a set of weight vectors and a scalarizing function. Moreover, the behavior of the $R2$ indicator is similar to that of the hypervolume (although $R2$ is weakly Pareto compliant) but has a

significantly lower computational cost [79]. Nevertheless, as we will see later on, the $R2$ indicator may obtain the same value for approximation sets with different distributions.

This section introduces a novel performance indicator based on LAP [57]: I_{LAP}. The core idea of this indicator is to use the cost obtained when adopting LAP as a performance indicator. Our experimental results show that I_{LAP} correctly ranks solution sets with different distributions and shapes. Moreover, we present an example in which I_{LAP} distinguishes two approximation sets in a better way than the $R2$ indicator.

The remainder of this section is organized as follows. Subsection 5.1.1 introduces our proposed performance indicator, called I_{LAP}. Subsection 5.1.2 presents a comparison of the I_{LAP} with the R2 indicator. Subsection 5.1.3 and 5.1.4 features an experimental analysis to verify the effectiveness of the I_{LAP}. Lastly, Subsection 5.1.5 summarizes the section.

5.1.1 Our proposed approach

As we mentioned in Chapter 3, Molinet Berenguer and Coello Coello [5] transformed the selection process of a MOEA into a LAP. In this proposal, the authors consider a set of individuals and a set of weight vectors representing different regions of the Pareto front. Moreover, the cost of assigning an individual to a weight vector is computed using a scalarizing function. Therefore, after solving the LAP, the individuals assigned to a weight vector are selected for the next generation. The authors incorporated this selection scheme into the HDE algorithm, which is very competitive with respect to modern algorithms.

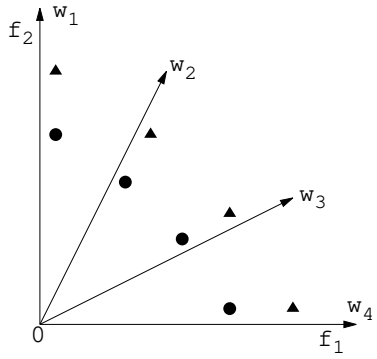
In the case of the LAP selection process, the size of the set of individuals is larger than the size of the set of weight vectors. Therefore, the Hungarian algorithm finds the subset of individuals that minimizes the overall assignment cost and discards the subsets with the worst values. Hence, we can deduce that the minimum overall assignment cost gives us an estimation of how good or bad a set is. Using this idea, we propose an indicator based on the LAP. The I_{LAP} indicator is defined in the following.

Definition 5.1. Given a set of uniformly distributed weight vectors $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$, an approximation set $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$, and a cost matrix C such that $C_{ij} = s(\boldsymbol{w}_i, \boldsymbol{a}_j)$ where s is a scalarizing function. Then, the I_{LAP} is defined as:

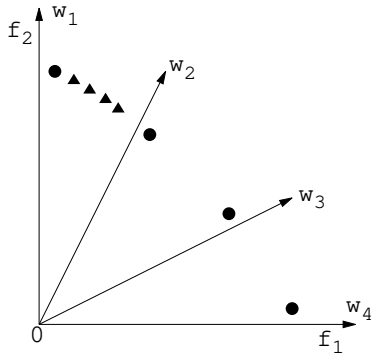
$$I_{LAP} = \frac{1}{n} \min_{x \in \mathcal{X}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij} \right\} \tag{5.1}$$

where $\mathcal{X}$ is the set of permutation matrices.

We compute the I_{LAP} by obtaining a cost matrix C using s, A, and W. Then, we solve the LAP defined by C employing the Hungarian algorithm. Finally, the indicator value is the best assignment's cost divided by n. The cost matrix computation is



(a) Measuring convergence. $I_{LAP} = 25000.3375$ for the circles' set, and $I_{LAP} = 25000.4375$ for the triangles' set.



(b) Measuring diversity. $I_{LAP} = 25000.4375$ for the circles' set, and $I_{LAP} = 197500.675$ for the triangles' set.

Figure 5.1: Examples where the I_{LAP} assesses both convergence and diversity. A lower value is preferred; therefore, the I_{LAP} ranks the sets correctly in both cases.

performed in $O(mn^2)$, where m is the number of objectives. Moreover, the LAP problem is solved in $O(n^3)$. Therefore the computational complexity of computing the I_{LAP} is $O(mn^2 + n^3)$.

In the I_{LAP}, each weight vector must be assigned to a currently unassigned solution while minimizing the cost. In the ideal case, each weight vector is assigned to a solution where it obtains its lowest cost. However, let's assume that more than one weight vector obtains its lowest cost with the same solution. In that case, the indicator will assign the solution to the vector with the lowest value and will use the second-best solutions for the remaining vectors.

This process allows the I_{LAP} to assess convergence and diversity at the same time. On the one hand, it measures convergence by always considering the best values of the scalarizing functions. On the other hand, it measures diversity because it tries to quantify how much the solutions cover the regions of the weight vectors. Examples of these two cases are shown in Fig. 5.1a and Fig. 5.1b, where the I_{LAP} successfully ranks the sets. We used the ASF scalarizing function for these examples and for the rest of this chapter.

5.1.2 Comparison between our approach and the $R2$-indicator

The I_{LAP} and $R2$ indicators have some similarities. Both use scalarizing functions and weight vectors to assess the performance of an approximation set. Moreover, the indicators will obtain the same value when the regions given by the weight vectors are equally covered (as shown in Fig. 5.2).

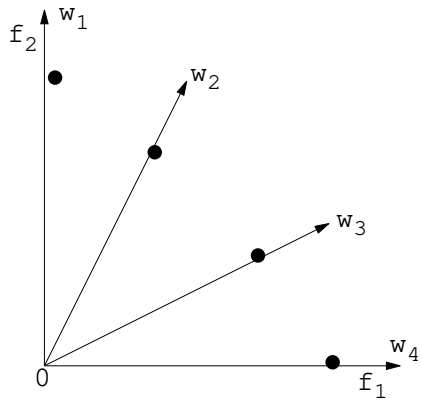


Figure 5.2: Example of a case where I_{LAP} and R2 obtain the same values: $I_{LAP} = R2 = 10000.3875$

However, the $R2$ indicator only considers the solutions with the best values of the scalarizing function, discarding the information provided by the solutions with the worst values. Therefore, the $R2$ indicator may not evaluate the performance of the whole set and may obtain the same value for two different approximations. On the other hand, the I_{LAP} indicator considers the whole set since it assigns each weight vector with a different solution and obtains the indicator's value from this assignment.

An example of the previous situation is shown in Fig. 5.3a and Fig. 5.3b. Given two different approximation sets, the $R2$ indicator obtains the same value, while the I_{LAP} obtains different values. Furthermore, I_{LAP} prefers the approximation set with a solution nearer to an uncovered vector.

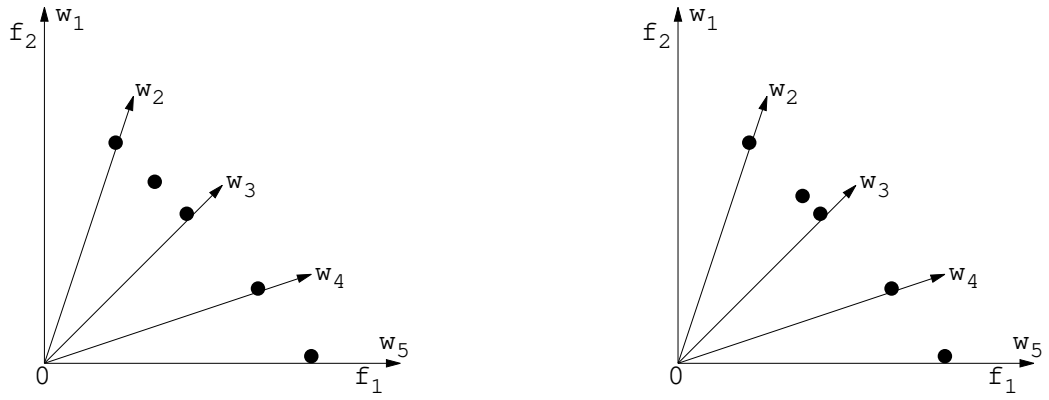
5.1.3 Evaluation in artificial many-objective Pareto fronts

In this section, we study the performance of the I_{LAP} in artificial Pareto fronts. We employed three types of solutions sets generated in a unit m-simplex:

- C1. The solutions are concentrated in one corner of the simplex.
- C2. The solutions are randomly generated.
- C3. The solutions are uniformly distributed. We employ the method proposed in [29] for this type of set.

Moreover, the set size for each dimension is shown in Table 5.1, and Figs. 5.4a to 5.5l show the parallel coordinates graphs of the sets. Regarding I_{LAP}, we use the ASF, and the UDH method [5] for generating the weight vectors.

The results are shown in Tables 5.2a and 5.2b. Moreover, we include the results of the hypervolume indicator [27] as a reference. We can observe that I_{LAP} consistently



(a) R2=44000.50133, I_{LAP}=44000.584 (b) R2=44000.50133, I_{LAP}=44000.616

Figure 5.3: The $R2$ indicator obtains the same value for two sets with distinct distributions, while the I_{LAP} indicator obtains different values.

Table 5.1: Set size for each dimension

m	3	4	5	6	7	8	9	10
Set size	100	110	120	130	140	150	160	170

ranks the C3 sets in first place, the C2 sets in second place, and the C1 sets in last place. Furthermore, the hypervolume obtains the same ranking. Therefore, I_{LAP} can correctly rank a set of solutions when dealing with problems having from 3 up to 10 dimensions.

5.1.4 Evaluation in Pareto front approximations

In this section, we use the I_{LAP}, the hypervolume, and the $R2$ indicator to evaluate the performance of two well-known MOEAs: the NSGA-II [39] and the MOEA/D [15]. For this purpose, we ran each algorithm 30 times using different problems. We adopted the DTLZ1, DTLZ2, and DTLZ7 problems from the DTLZ [64] test suite, the DTLZ1^{-1} from the Minus-DTLZ test problems [77], and the WFG1-WFG3 from the WFG [67] test suite with $m = 3$, 5, 8, and 10 objectives. Regarding the DTLZ problems, we set the number of decision variables to $n = m + k - 1$, where $k = 5$ for DTLZ1, $k = 10$ for DTLZ2, and $k = 20$ for DTLZ7. In the case of the WFG problems, we set the position-related parameters to $2 \times (m - 1)$ and the distance-related parameters to 20. Finally, we used the same configuration of DTLZ1 for DTLZ1^{-1}.

In the case of the algorithm's parameters, we set the population sizes to 100 for three objectives, 120 for five, 140 for eight, and 160 for ten. We set the crossover and mutation parameters to: $pc = 1.0$, $pm = 1/$number of variables, $n_c = 20$, and $n_m =$

Table 5.2: I_{LAP} and Hypervolume values of the sets C1, C2, and C3 for each dimension m. Darker cells imply better values.

m	C1	C2	C3
3	5.0453	1.5322	1.1263
4	5.6031	1.8407	1.3157
5	5.9506	2.3113	1.5775
6	5.8495	2.6022	1.9827
7	6.0938	2.9235	2.1356
8	5.9524	3.0488	2.4045
9	5.8106	3.2949	2.9855
10	5.5003	3.4681	3.1862

(a) I_{LAP}

m	C1	C2	C3
3	0.77462	1.076862	1.11977
4	0.906105	1.32058	1.369026
5	1.036677	1.49916	1.560266
6	1.197589	1.659354	1.737507
7	1.284942	1.862622	1.920832
8	1.400768	2.057528	2.111709
9	1.586619	2.252491	2.328822
10	1.805137	2.501373	2.563038

(b) Hypervolume

20. Regarding the MOEA/D parameters, we used a neighborhood size $T = 20$, the ASF function, and the UDH weight vectors. Finally, the I_{LAP} and the $R2$ indicators adopted the ASF function and UDH weight vectors.

Tables 5.3a, 5.3b, and 5.3c display the average and the standard deviation of each indicator. We can observe that the three indicators obtain the same results for DTLZ1, DTLZ2, DTLZ7, WFG1, WFG2, and DTLZ1^{-1}. In the case of the WFG3 problem, the $R2$ and the hypervolume get the same rank in 5 and 8 objectives. In contrast, the I_{LAP} and the hypervolume get the same rank in 3 and 5 objectives. This situation could happen because the WFG3 is a linear problem that hardly fits the shape of a simplex. Therefore, the R2 and I_{LAP} indicators may have some trouble with the performance assessment because they employ reference vectors sampled in a simplex.

5.1.5 Discussion

We proposed in this section a novel performance indicator based on the Linear Assignment Problem called I_{LAP}. The experimental results showed that our proposed I_{LAP} could successfully rank the solutions sets using different distributions and Pareto front shapes of many-objective problems. Moreover, we described an example where the $R2$ indicator (the performance indicator with the most significant similarity with the I_{LAP}) can not distinguish between two different approximation sets. In contrast, our proposed I_{LAP} can differentiate them and prefers the one with a solution nearer to an uncovered region.

Table 5.3: Average and standard deviation of the hypervolume, $R2$, and I_{LAP} indicators. The gray cells are used to show better values. Moreover, the symbol "*" represents that the algorithm is statistically better according to the Wilcoxon rank sum test.

	M	MOEA/D	NSGA-II
DTLZ1	3	1.324e+0 (8.5e-4)	*1.331e+0 (3.1e-6)
	5	*1.610e+0 (7.5e-6)	1.610e+0 (1.4e-4)
	8	*2.144e+0 (6.2e-6)	2.143e+0 (4.4e-4)
	10	*2.594e+0 (8.3e-6)	2.593e+0 (1.4e-4)
DTLZ2	3	*8.330e-1 (9.3e-4)	8.169e-1 (5.1e-3)
	5	*1.593e+0 (2.1e-3)	1.584e+0 (5.5e-3)
	8	*2.139e+0 (1.2e-3)	2.002e+0 (4.2e-2)
	10	*2.589e+0 (1.2e-3)	2.454e+0 (3.9e-2)
DTLZ7	3	6.436e-1 (5.e-2)	*6.908e-1 (2.6e-2)
	5	6.136e-1 (5.6e-2)	*8.222e-1 (1.9e-2)
	8	2.e-1 (1.2e-1)	*7.608e-1 (9.4e-2)
	10	1.102e-1 (9.4e-2)	*5.378e-1 (1.3e-1)
WFG1	3	*1.197e+0 (2.7e-2)	1.108e+0 (2.5e-2)
	5	*1.526e+0 (3.6e-2)	1.273e+0 (2.9e-2)
	8	*2.055e+0 (5.0e-2)	1.495e+0 (3.6e-2)
	10	*2.465e+0 (3.5e-2)	1.378e+0 (3.8e-2)
WFG2	3	1.072e+0 (8.e-2)	*1.155e+0 (8.2e-2)
	5	1.328e+0 (1.1e-1)	*1.58e+0 (6.9e-3)
	8	1.711e+0 (1.4e-1)	*2.129e+0 (6.3e-3)
	10	2.074e+0 (2.1e-1)	*2.576e+0 (7.8e-3)
WFG3	3	8.132e-1 (7.8e-3)	8.162e-1 (4.e-3)
	5	1.140e+0 (1.4e-2)	1.134e+0 (1.4e-2)
	8	1.472e+0 (2.9e-2)	*1.501e+0 (2.7e-2)
	10	1.518e+0 (5.1e-2)	*1.798e+0 (3.5e-2)
$DTLZ1^{-1}$	3	*2.775e-1 (2.9e-5)	2.733e-1 (2.2e-3)
	5	1.224e-2 (9.e-5)	1.215e-2 (5.3e-4)
	8	*3.472e-5 (1.3e-6)	3.167e-5 (2.3e-6)
	10	4.988e-7 (3.8e-8)	4.924e-7 (3.6e-8)

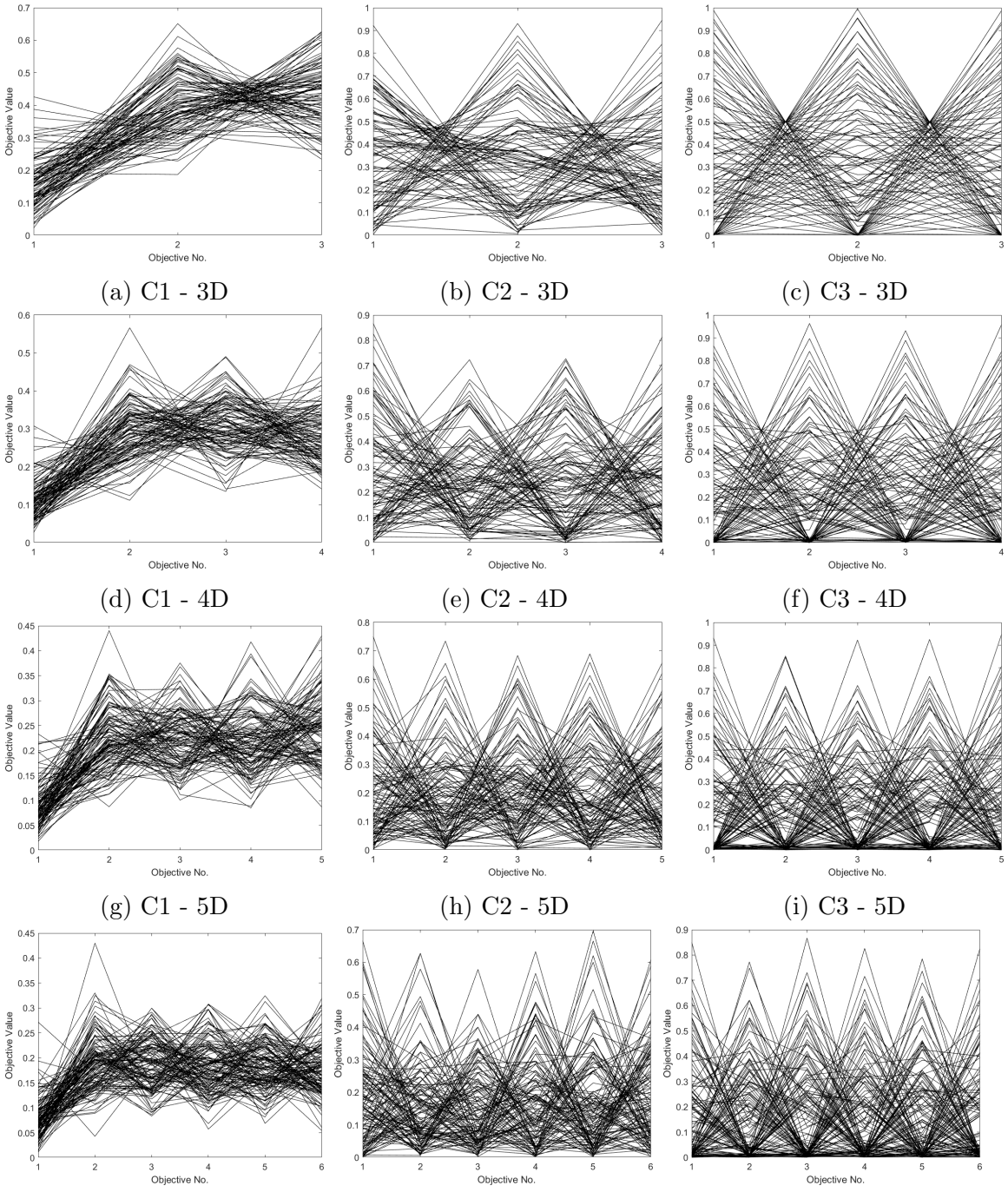
(a) HV

	M	MOEA/D	NSGA-II
DTLZ1	3	7.692e-2 (7.e-3)	*2.856e-2 (6.e-4)
	5	*1.155e-3 (3.1e-5)	2.388e-1 (1.2e-1)
	8	*1.318e-3 (5.5e-5)	8.6e-1 (3.1e-1)
	10	*1.574e-3 (9.5e-5)	1.165e+0 (2.6e-1)
DTLZ2	3	*1.360e+0 (2.5e-4)	1.446e+0 (3.8e-2)
	5	*8.176e-1 (5.2e-3)	1.196e+0 (5.9e-2)
	8	*6.764e-1 (2.4e-2)	2.832e+0 (2.4e-1)
	10	*8.836e-1 (1.0e-1)	3.216e+0 (2.0e-1)
DTLZ7	3	3.517e+0 (1.1e+0)	*3.074e+0 (5.8e-1)
	5	7.344e+0 (9.4e-1)	*6.085e+0 (1.7e-1)
	8	1.702e+1 (2.2e+0)	*1.098e+1 (4.8e-1)
	10	2.339e+1 (3.5e+0)	*1.487e+1 (9.1e-1)
WFG1	3	*1.076e+0 (1.6e-1)	1.493e+0 (2.7e-1)
	5	*1.375e+0 (2.2e-1)	2.922e+0 (2.6e-1)
	8	*1.412e+0 (2.3e-1)	4.859e+0 (2.3e-1)
	10	*1.797e+0 (2.1e-1)	8.452e+0 (2.7e-1)
WFG2	3	2.126e+0 (7.0e-1)	*1.488e+0 (7.6e-1)
	5	3.305e+0 (1.1e+0)	*1.174e+0 (5.4e-2)
	8	5.121e+0 (1.5e+0)	*1.425e+0 (5.4e-2)
	10	5.187e+0 (2.3e+0)	*1.533e+0 (6.7e-2)
WFG3	3	*2.667e+0 (2.5e-2)	2.673e+0 (2.0e-2)
	5	*3.724e+0 (6.2e-2)	3.796e+0 (6.2e-2)
	8	*5.617e+0 (8.1e-2)	5.711e+0 (7.7e-2)
	10	6.985e+0 (2.2e-1)	*6.594e+0 (1.1e-1)
$DTLZ1^{-1}$	3	*5.014e+0 (2.0e-4)	5.46e+0 (6.4e-2)
	5	*1.349e+1 (3.2e-2)	1.581e+1 (1.4e-1)
	8	*3.073e+1 (8.9e-2)	3.825e+1 (3.5e-1)
	10	*4.034e+1 (1.2e-1)	5.187e+1 (4.4e-1)

(b) R2

	M	MOEA/D	NSGA-II
DTLZ1	3	6.127e-1 (1.0e-3)	*5.061e-1 (5.9e-2)
	5	*1.155e-3 (3.0e-5)	9.361e-1 (1.0e-1)
	8	*1.318e-3 (5.6e-5)	1.800e+0 (1.1e-1)
	10	*1.577e-3 (9.8e-5)	2.105e+0 (6.0e-2)
DTLZ2	3	*1.363e+0 (3.4e-4)	1.958e+0 (7.2e-2)
	5	*8.181e-1 (5.2e-3)	1.478e+0 (6.5e-2)
	8	*6.765e-1 (2.5e-2)	4.292e+0 (2.2e-1)
	10	*8.851e-1 (1.1e-1)	5.212e+0 (1.8e-1)
DTLZ7	3	4.797e+0 (1.1e+0)	*3.298e+0 (6.3e-1)
	5	1.257e+1 (1.2e+0)	*6.904e+0 (1.7e-1)
	8	2.834e+1 (3.0e+0)	*1.333e+1 (4.4e-1)
	10	3.711e+1 (6.0e+0)	*1.825e+1 (6.9e-1)
WFG1	3	*1.359e+0 (1.7e-1)	1.721e+0 (3.8e-1)
	5	*1.62e+0 (3.1e-1)	3.338e+0 (3.3e-1)
	8	*1.727e+0 (2.9e-1)	5.834e+0 (4.5e-1)
	10	*2.213e+0 (2.8e-1)	1.021e+1 (4.6e-1)
WFG2	3	2.377e+0 (6.3e-1)	*1.75e+0 (8.5e-1)
	5	3.568e+0 (1.1e+0)	*1.954e+0 (2.3e-1)
	8	5.302e+0 (1.5e+0)	*2.953e+0 (2.5e-1)
	10	5.314e+0 (2.2e+0)	*3.271e+0 (3.0e-1)
WFG3	3	3.276e+0 (3.7e-2)	*2.929e+0 (5.6e-2)
	5	*4.172e+0 (1.2e-1)	4.804e+0 (1.3e-1)
	8	*6.833e+0 (1.6e-1)	7.874e+0 (2.3e-1)
	10	*9.165e+0 (2.5e-1)	9.488e+0 (2.6e-1)
$DTLZ1^{-1}$	3	*5.014e+0 (2.0e-4)	5.627e+0 (6.7e-2)
	5	*1.349e+1 (3.2e-2)	1.637e+1 (1.9e-1)
	8	*3.076e+1 (7.4e-2)	3.915e+1 (3.e-1)
	10	*4.037e+1 (1.2e-1)	5.325e+1 (3.7e-1)

(c) I_{LAP}



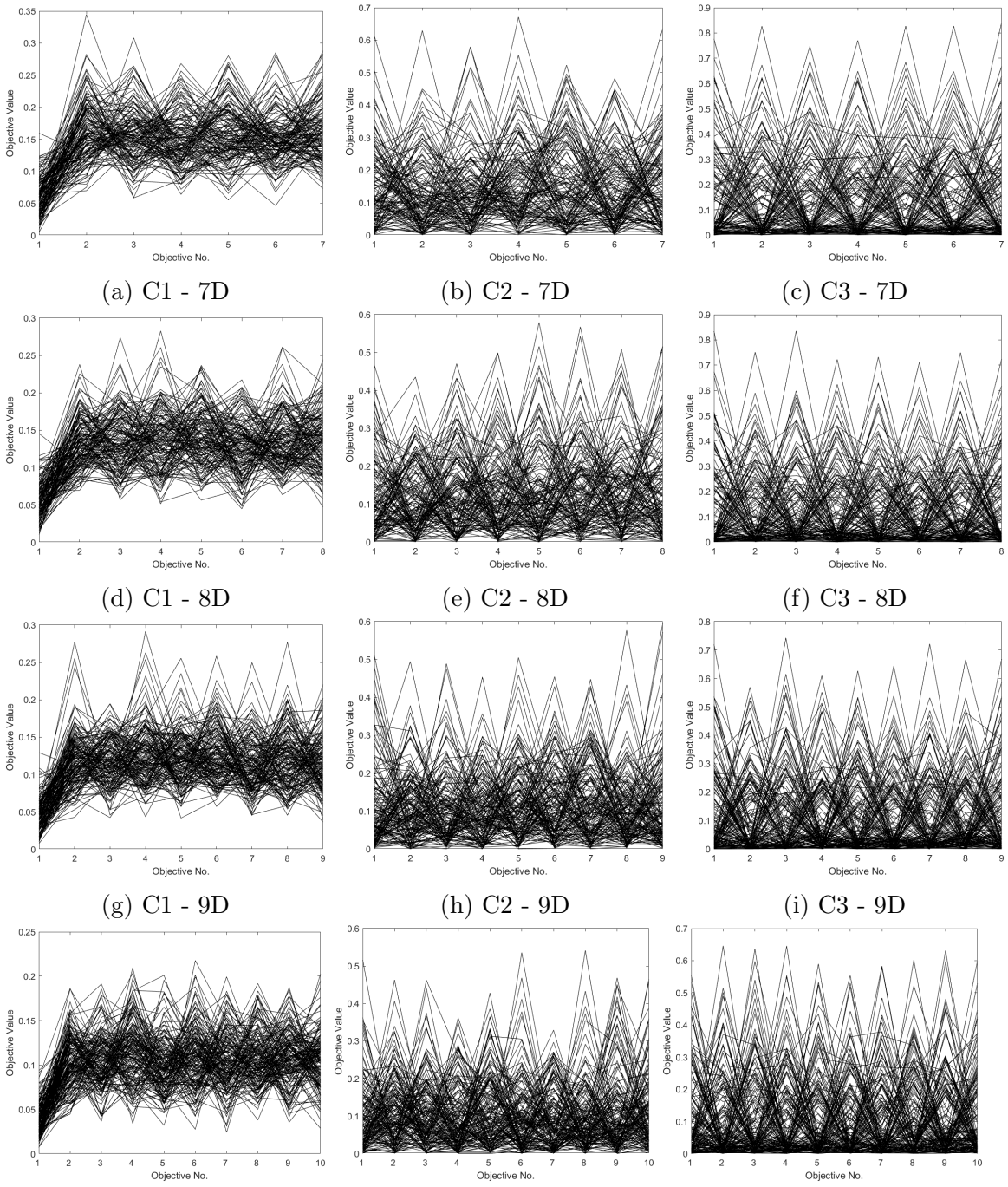
(a) C1 - 3D (b) C2 - 3D (c) C3 - 3D

(d) C1 - 4D (e) C2 - 4D (f) C3 - 4D

(g) C1 - 5D (h) C2 - 5D (i) C3 - 5D

(j) C1 - 6D (k) C2 - 6D (l) C3 - 6D

Figure 5.4: Artificial solution sets generated in a unit simplex. Solutions in C1 are concentrated in a corner, in C2 are randomly generated, and in C3 are uniformly distributed.



(a) C1 - 7D (b) C2 - 7D (c) C3 - 7D

(d) C1 - 8D (e) C2 - 8D (f) C3 - 8D

(g) C1 - 9D (h) C2 - 9D (i) C3 - 9D

(j) C1 - 10D (k) C2 - 10D (l) C3 - 10D

Figure 5.5: Artificial solution sets generated in a unit simplex (continuation)

5.2 A diversity indicator based on the LAP

This section presents a novel diversity indicator based on the LAP, the D_{LAP}. The core idea is to measure the closeness between an approximation set and a set of well-distributed reference vectors. Therefore, we compute the minimum cost of assigning the solutions in the approximation set and the reference vectors.

The remainder of this section is organized as follows. First, we describe our proposed approach in Subsection 5.2.1. Then, we evaluate the new indicator in artificial Pareto fronts (see Subsection 5.2.2) and Pareto front approximations (see Subsection 5.2.3). Finally, we discuss the results in Subsection 5.2.4.

5.2.1 Our proposed approach

We proposed a new indicator that measures the diversity of solutions of an approximation set using the LAP. We called this indicator D_{LAP}, and its formal definition is the following:

Definition 5.2. Given a set of uniformly distributed reference vectors $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$, an approximation set $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$, and a cost matrix C such that $C_{ij} = c(\boldsymbol{w}_i, \boldsymbol{a}_j)$ where c is a function that measures the closeness between $\boldsymbol{w}_i$ and $\boldsymbol{a}_j$. Then, the D_{LAP} is defined as:

$$D_{LAP} = \frac{1}{n} \min_{x \in \mathcal{X}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij} \right\} \tag{5.2}$$

where $\mathcal{X}$ is the set of permutation matrices.

We compute the D_{LAP} by measuring the closeness between each reference vector and every solution in objective space. Then, we determine the minimum cost of assigning a set of reference vectors to the approximation set. The resulting value indicates how well an approximation set covers different regions of the objective space, where a smaller value indicates a better approximation.

We proposed two different alternatives to evaluate the proximity between a solution and a reference vector. The first alternative is to use the shortest Euclidean distance between a solution and a reference vector. Given a solution $\boldsymbol{a} = [a_1, \ldots, a_m] \in A$ and a reference vector $\boldsymbol{w}$, this distance can be computed using the following expression:

$$c_{distance}(\boldsymbol{w}, \boldsymbol{a}) = \left\| \boldsymbol{a}' - \frac{\boldsymbol{a}' \cdot \boldsymbol{w}}{\boldsymbol{w} \cdot \boldsymbol{w}} \boldsymbol{w} \right\| \tag{5.3}$$

where $\boldsymbol{a'} = [a'_1, \ldots, a'_m]$ such that $a'_i = \frac{a_i - z_i^{min}}{z_i^{max} - z_i^{min}}$ for all $i = 1, \ldots, m$. Moreover, $z_i^{min} = \min_{z \in A} z_i$, and $z_i^{max} = \max_{z \in A} z_i$.

The second alternative employs the angle between a solution and a reference vector to measure proximity. Given a solution $\boldsymbol{a} = [a_1, \ldots, a_m] \in A$ and a reference vector $\boldsymbol{w}$, this angle can be computed using the following expression:

$$c_{angle}(\boldsymbol{w}, \boldsymbol{a}') = \arccos\left(\frac{\boldsymbol{w} \cdot \boldsymbol{a}'}{\|\boldsymbol{w}\| \|\boldsymbol{a}'\|} \right) \tag{5.4}$$

where $\boldsymbol{a'}$ is defined as in $c_{distance}$. Figure 5.6 illustrates how these two metrics of closeness are defined.

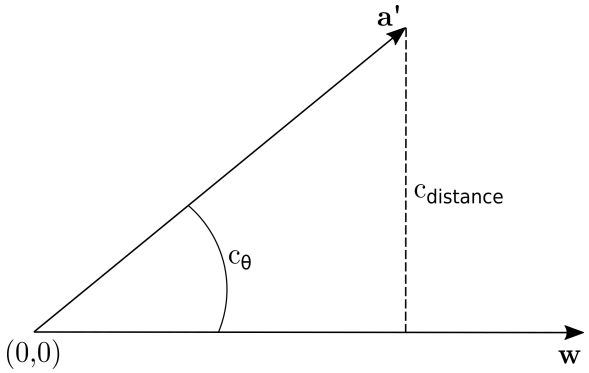


Figure 5.6: Metrics of closeness between a reference vector and a solution

5.2.2 Evaluation in artificial Pareto fronts

We conduct three experiments to validate D_{LAP} in artificially generated Pareto fronts. In each experiment, we use the UDH method to generate the reference set, setting the number of vectors equal to the size of the approximation set. Moreover, for each approximation, we compute the D_{LAP} using the $c_{distance}$ ($D_{LAP(distance)}$) and the c_{angle} ($D_{LAP(angle)}$).

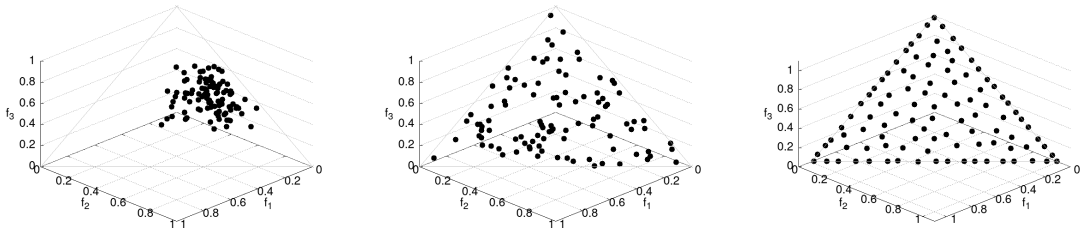
In the first experiment, we generate three approximation sets with 100 solutions on the hyperplane $f_1 + f_2 + f_3 = 1$ (see Figure 5.7). The solutions in Figure 5.7a are concentrated on a small region of the hyperplane. The solutions in Figure 5.7b are randomly distributed, and those in Figure 5.7c are uniformly distributed. We can observe that the lower the value, the better the distribution, regardless of the closeness function.

In the second experiment, we rotate the previous sets to verify if the indicator is sensible to the geometry of the Pareto front (see Figure 5.8). We can observe that $D_{LAP(angle)}$ assigns lower values to the sets with better distribution, and the same occurs for $D_{LAP(distance)}$.

Regarding the third experiment, we scale the approximation set from Figure 5.7c by 75% and 50% to test if the indicator can measure the set coverage (see Figure 5.9). The results show that both alternatives, $D_{LAP(distance)}$ and $D_{LAP(angle)}$, assign lower values to the sets with better coverage.

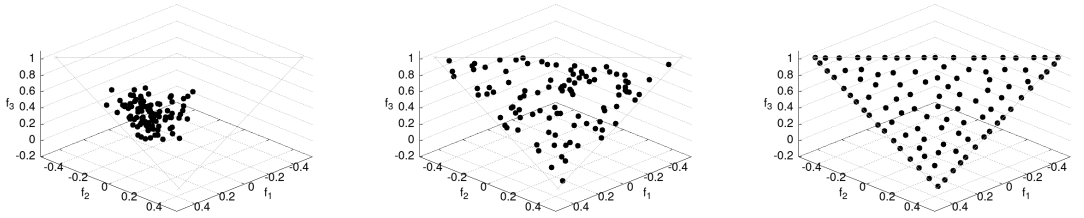
5.2.3 Evaluation in Pareto front approximations

In this section, we evaluate our proposed approach into Pareto front approximations. For this purpose, we performed 30 independent runs of NSGA-II [39] and MOEA/D [15] using three objective problems. In particular, we adopted the DTLZ1, DTLZ2, and DTLZ7 from the DTLZ test suite [64], the WFG1, and WFG3 from the WFG test suite [67], and the DTLZ1-Minus proposed in [77].



(a) $D_{LAP(angle)} = 0.4511$, $D_{LAP(distance)} = 0.2557$ (b) $D_{LAP(angle)} = 0.1053$, $D_{LAP(distance)} = 0.0699$ (c) $D_{LAP(angle)} = 0.0832$, $D_{LAP(distance)} = 0.0611$

Figure 5.7: Artificially generated Pareto fronts with different distributions located on a unit simplex.



(a) $D_{LAP(angle)} = 0.4731$, $D_{LAP(distance)} = 0.5164$ (b) $D_{LAP(angle)} = 0.3006$, $D_{LAP(distance)} = 0.3418$ (c) $D_{LAP(angle)} = 0.256$, $D_{LAP(distance)} = 0.3018$

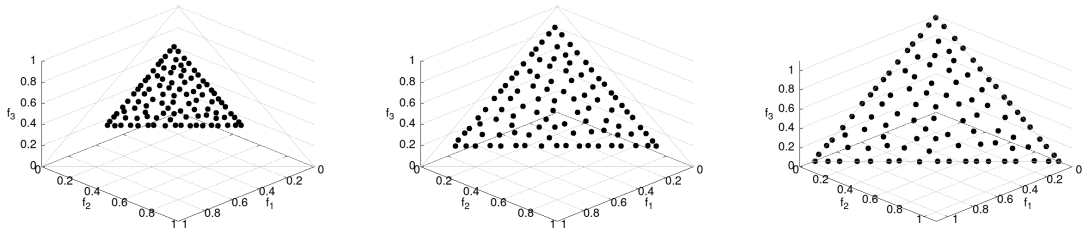
Figure 5.8: Artificially generated Pareto fronts with different distributions located on an inverted unit simplex.

In the case of the algorithm's parameters, we set the population size to 100. We set the crossover and mutation parameters to: $pc = 1.0$, $pm = 1/\text{number of variables}$, $n_c = 20$, and $n_m = 20$. Concerning the MOEA/D parameters, we used a neighborhood size $T = 20$, the ASF function, and the UDH weight vectors.

Tables 5.4 and 5.5 display the average and standard deviation of $D_{LAP(distance)}$ and $D_{LAP(angle)}$ for each algorithm and problem. Moreover, Figures 5.10 and 5.11 show the most representative Pareto front approximations that NSGA-II and MOEA/D obtained.

The results show that both indicators obtained the same ranking in DTLZ1^{-1}, DTLZ2, and DTLZ7 problems. While in problems DTLZ1, WFG1, and WFG3, the indicators obtained different rankings. Therefore, the indicators have different behavior depending on the problem.

We can observe that MOEA/D outperformed NSGA-II in generating well-distributed approximations for DTLZ1 and DTLZ2 problems. This result matches the ranking obtained by $D_{LAP(distance)}$. In contrast, $D_{LAP(angle)}$ only matches in the DTLZ2 problem. On the other hand, NSGA-II generated better approximations than MOEA/D in DTLZ7 and DTLZ1^{-1} problems. Both indicators ($D_{LAP(angle)}$ and $D_{LAP(distance)}$)



(a) $D_{LAP(angle)} = 0.2631$, $D_{LAP(distance)} = 0.1563$ (b) $D_{LAP(angle)} = 0.1138$, $D_{LAP(distance)} = 0.076$ (c) $D_{LAP(angle)} = 0.0834$, $D_{LAP(distance)} = 0.0611$

Figure 5.9: Artificially generated Pareto fronts with different coverage located on a unit simplex.

correctly rank the DTLZ7 problems, but fail in DTLZ1^{-1}. Regarding WFG3, the figures show that NSGA-II performed better than MOEA/D. This result fits the ranking obtained by $D_{LAP(distance)}$ but differs from the ranking obtained by $D_{LAP(angle)}$.

Therefore, we can conclude that $D_{LAP(distance)}$ correctly ranks more approximation sets than $D_{LAP(angle)}$. However, none of the indicators could correctly rank DTLZ1^{-1}.

Table 5.4: Average and standard deviation of $D_{LAP(distance)}$ indicator. The gray cells are used to show better values.

Problem	NSGA-II	MOEA/D
DTLZ1^{-1}	3.4424e-01 (5.42e-03)	2.6413e-01 (3.47e-05)
DTLZ1	1.0869e-01 (3.16e-02)	7.8457e-03 (4.06e-04)
DTLZ2	1.0496e-01 (6.27e-03)	7.5247e-03 (8.25e-04)
DTLZ7	2.1643e-01 (1.46e-02)	4.8157e-01 (1.24e-01)
WFG1	1.1808e-01 (1.08e-02)	1.1143e-01 (2.45e-02)
WFG3	2.2273e-01 (1.18e-02)	2.2786e-01 (6.02e-03)

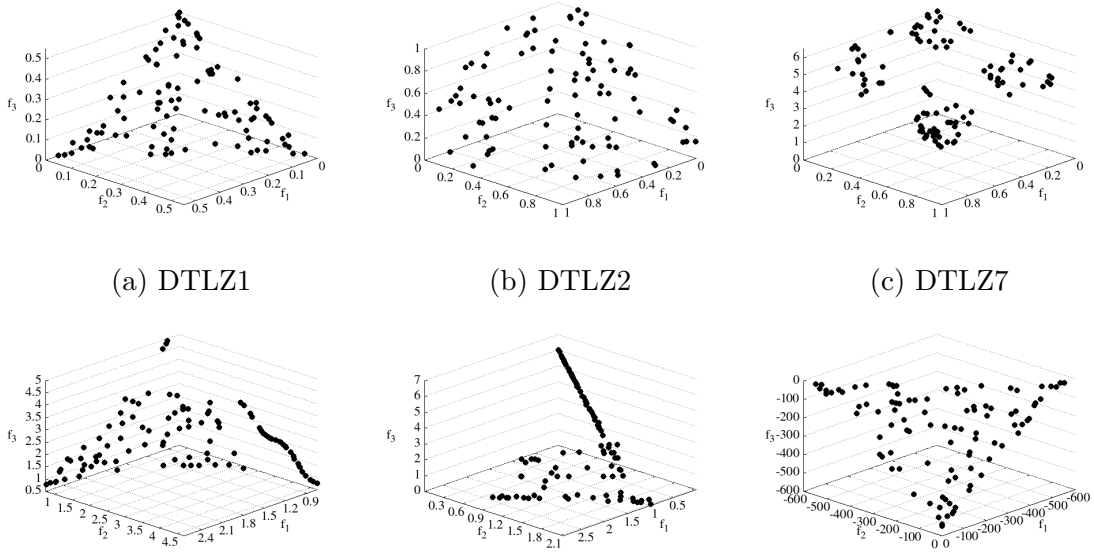
5.2.4 Discussion

We have presented a novel diversity indicator based on the LAP called D_{LAP}. The idea was to obtain the minimum cost of assigning an approximation set to a reference vector set, where the cost is computed using a closeness function between a solution and a vector. We proposed two functions to measure this proximity: the minimum distance between a point ($c_{distance}$) and a vector and the angle between a solution and a vector (c_{angle}).

The experimental results showed that this new indicator is able to measure the coverage and distribution of artificial Pareto fronts, regardless of the closeness function. Moreover, the experiments showed that the D_{LAP} with the $c_{distance}$ performed better than c_{angle} to measure the diversity of approximation sets.

Table 5.5: Average and standard deviation of the $D_{LAP(angle)}$ indicator. The gray cells are used to show better values.

Problem	NSGA2	MOEAD
DTLZ1^{-1}	2.8618e-01 (5.09e-03)	2.1678e-01 (1.96e-05)
DTLZ1	6.3980e-01 (4.02e-02)	7.1443e-01 (8.39e-04)
DTLZ2	1.0555e-01 (6.93e-03)	1.3515e-02 (1.07e-04)
DTLZ7	2.2282e-01 (7.42e-02)	4.1321e-01 (6.72e-02)
WFG1	2.2328e-01 (7.06e-02)	2.6044e-01 (2.08e-02)
WFG3	2.5687e-01 (1.09e-02)	2.3707e-01 (2.89e-03)



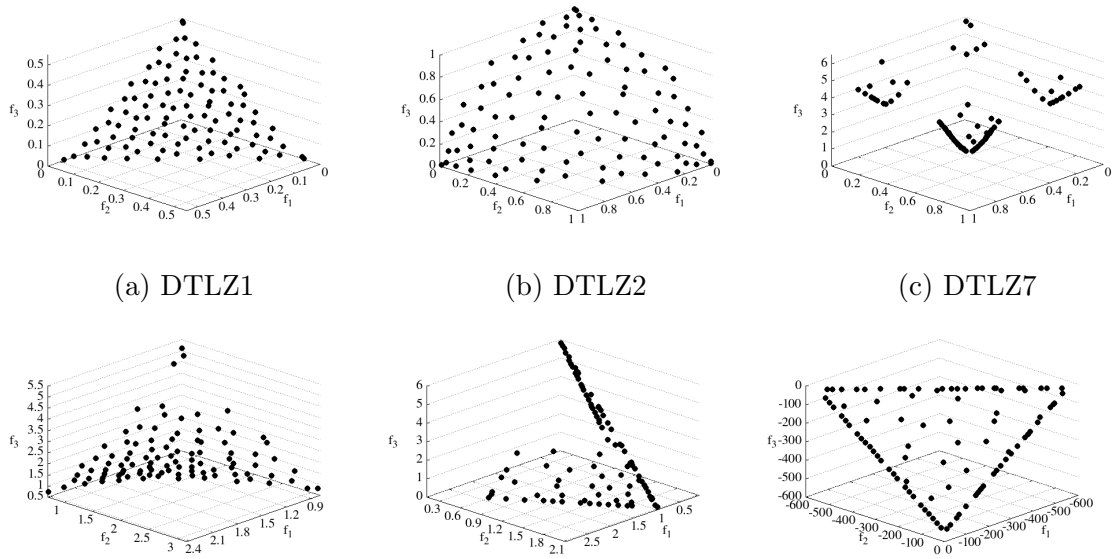
(a) DTLZ1 (b) DTLZ2 (c) DTLZ7

(d) WFG1 (e) WFG3 (f) DTLZ1^{-1}

Figure 5.10: Representative Pareto front approximations obtained by NSGA-II

5.3 Summary

In this chapter, we proposed two novel indicators that employ the LAP to assess the performance of an algorithm: I_{LAP} and D_{LAP}. The I_{LAP} was designed to measure the convergence and diversity of an approximation set. The experimental results showed that the I_{LAP} is able to evaluate the performance of approximation sets with different distributions and shapes. On the other hand, the D_{LAP} was developed to measure the diversity of approximation sets. Moreover, our experiments showed that this indicator can correctly assess the diversity of approximation sets with different distributions and coverage of the regions.



(a) DTLZ1 (b) DTLZ2 (c) DTLZ7

(d) WFG1 (e) WFG3 (f) DTLZ1^{-1}

Figure 5.11: Representative Pareto front approximations obtained by MOEAD

Chapter 6

Conclusions and future work

Most MOEAs can be categorized into three families according to their selection scheme: Pareto-based, indicator-based, and decomposition-based. Pareto-based MOEAs are widely used and effective for optimizing two-objective problems. However, their performance deteriorates when increasing the number of objectives. On the other hand, indicator-based MOEAs can deal with complicated Pareto fronts, but their computational cost usually becomes prohibitive when dealing with many-objective problems. Moreover, the solutions' distribution relies on the chosen indicator. Conversely, decomposition-based MOEAs can solve multiple single-objective problems in a single run, offering diverse solutions. However, the solutions' distribution highly depends on the adopted scalarizing function and weight vector set.

Throughout the years, the drawbacks of these three families of MOEAs have been addressed but not fully resolved. Therefore, this thesis explored alternative selection schemes that do not belong to these families. In particular, we studied the transformation of the selection process into a linear assignment problem, which is competitive with well-known MOEAs and has good performance in many-objective problems. The main contributions of this thesis are the following:

In Chapter 3, we introduced the linear assignment problem transformation and examined the algorithms containing this selection scheme. We noted that none of the algorithms require additional mechanisms to achieve accurate approximations beyond the LAP transformation. However, there have been limited studies on the behavior of this selection scheme.

In Chapter 4, we performed an experimental analysis of the influence of scalarizing functions and weight vectors in the LAP selection scheme. For this purpose, we evaluated six scalarizing functions (TCH, ATCH, ASF, AASF, PBI, and AGSF2) and two weight vector sets (SLD and UDH) in the HDE. We found that the SLD set performed best in conventional problems (DTLZ and WFG), while the UDH set was most effective in Minus problems (DTLZ^{-}1 and WFG^{-}1). Moreover, the two best scalarizing functions were the AASF and the AGSF2. We conclude that the choice of scalarizing function and weight vector set highly influences the behavior of the LAP selection scheme. Moreover, the best choice depends on the problem.

For this reason, we designed a novel MOEA that integrates different scalarizing

functions and weight vector sets using the LAP transformation. In particular, we employed an ensemble technique to simultaneously use the best pair of scalarizing functions (AASF and AGSF2) and the two previous vector sets (SLD and UDH). The resulting algorithm was called an Ensemble of Scalarizing functions and Weight vectors (ESW). We evaluated the performance of ESW using problems with three to ten objectives. The experimental results showed that ESW outperforms HDE using the individual pairs. Moreover, ESW is competitive with state-of-the-art algorithms. Therefore, we concluded that the simultaneous use of different scalarizing functions and weight vectors enhances the performance of the LAP selection scheme.

Furthermore, within this chapter, we identified two critical drawbacks of the LAP transformation: it may select duplicated solutions and it does not always prefer non-dominated solutions. In order to address these disadvantages, we proposed a novel MOEA that combines the LAP transformation with the hypervolume indicator. However, we employed a polar-coordinate approximation of the hypervolume because of its high computational cost. The resulting algorithm was called MOEA-LAPCO. The MOEA-LAPCO strategy involves dividing the selection process into two phases. During the first phase, the LAP transformation is employed to eliminate a fraction of the population, while the second phase employs the hypervolume indicator. The goal was to enhance the preference for non-dominated solutions by utilizing the hypervolume indicator. The experimental results showed that our proposed approach is competitive with modern algorithms in problems with three to ten objectives.

In Chapter 5, we presented a novel performance indicator that employs the LAP transformation as its core mechanism: the I_{LAP}. Our experimental results showed that our proposed indicator can effectively rank solution sets with distinct distributions and Pareto front shapes. In addition, we provided an example where the $R2$ indicator, which is the most similar performance indicator to the I_{LAP}, fails to differentiate between two distinct approximation sets. In contrast, our proposed I_{LAP} is able to distinguish between them and favors the solution that is closer to an uncovered area. Additionally, in this chapter, we introduced a novel diversity indicator based on the LAP: D_{LAP}. Its core idea was to measure the coverage and distribution of the solution by employing a reference set. The experimental results showed that the D_{LAP} can accurately rank artificial Pareto fronts and approximation sets with diverse characteristics.

As a part of our future work, we are interested in exploring the mathematical properties of the I_{LAP} indicator (such as Pareto compliance) and determining which scalarizing function is the best to evaluate approximation sets. Moreover, we would like to incorporate a decision-maker's preferences into the LAP transformation. This task can be done easily because each solution must be assigned to a point or vector, which can be a reference point of a preference direction. In addition, we are interested in exploring different ways to compute the assignment cost to avoid the occasional preference of dominated solutions.

Appendix A

Test functions

In this appendix, we summarize the problems that we adopted in this thesis. Mainly, we present the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite, the Walking Fish Group (WFG) test suite, and the Minus test problems.

A.1 Deb-Thiele-Laumanns-Zitzler test suite

The Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite was presented in a technical report by Deb et al. [80] (later published in [64]). It comprises nine problems (DTLZ1-DTLZ9) that can be scaled to any number of objectives. In this thesis, we employed seven of the nine problems, whose main characteristics are summarized in Table A.1. Moreover, their mathematical definitions are presented below.

Table A.1: Main characteristics of the DTLZ1-DTLZ7 problems

Problem	**Separability**	**Frontality**	**Geometry**
DTLZ1	separable	multifrontal	linear
DTLZ2	separable	unifrontal	concave
DTLZ3	separable	multifrontal	concave
DTLZ4	separable	unifrontal	concave
DTLZ5	unknown	unifrontal	degenerate ($M = 3$)/ unknown ($M > 3$)
DTLZ6	unknown	unifrontal	degenerate ($M = 3$)/ unknown ($M > 3$)
DTLZ7	separable	unifrontal	disconnected with mixed components

DTLZ1

It is a multifrontal problem with a linear Pareto Front. Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\boldsymbol{x}) = \frac{1}{2}(1+g(\boldsymbol{y}))\prod_{i=1}^{M-1} x_i, \\
& f_{j=2:M-1}(\boldsymbol{x}) = \frac{1}{2}(1+g(\boldsymbol{y}))\Big(\prod_{i=1}^{M-j} x_i\Big)(1-x_{M-j+1}), \\
& f_M(\boldsymbol{x}) = \frac{1}{2}(1+g(\boldsymbol{y}))(1-x_1), \\
\text{where} \quad & \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = 100\Big[k + \sum_{i=1}^{k}(y_i-0.5)^2 - \cos\left(20\pi(y_i-0.5)\right)\Big], \\
\text{subject to} \quad & 0 \le x_i \le 1, \text{ for } i = 1, 2, ..., n.
\end{aligned} \tag{A.1}$$

The Pareto Optimal Set corresponds to $x_i = 0.5$ for all $x_i \in \boldsymbol{y}$. Moreover, its Pareto Front lies on the hyperplane $\sum_{m=1}^{M} f_m = 0.5$. The number of variables is defined by $n = M + k - 1$, where a value of $k = 5$ is suggested.

DTLZ2

It is a multi-objective problem with a concave Pareto Front. Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\prod_{i=1}^{M-1}\cos\Big(\frac{\pi x_i}{2}\Big) \\
& f_{j=2:M-1}(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\Big(\prod_{i=1}^{M-j}\cos\Big(\frac{\pi x_i}{2}\Big)\Big)\sin\Big(\frac{\pi x_{M-j+1}}{2}\Big) \\
& f_M(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\sin\big(\frac{\pi x_1}{2}\big) \\
\text{where} \quad & \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = \sum_{i=1}^{k}(y_i-0.5)^2, \\
\text{subject to} \quad & 0 \le x_i \le 1, \text{ for } i = 1, 2, ..., n.
\end{aligned} \tag{A.2}$$

The Pareto optimal solutions correspond to $x_i = 0.5$ for all $x_i \in \boldsymbol{y}$, and the objective functions must satisfy $\sum_{m=1}^{M} f_m^2 = 1$. The number of variables is defined by $n = M + k - 1$, where a value of $k = 10$ is suggested.

DTLZ3

It is a multifrontal problem with a concave Pareto Front. Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\boldsymbol{x}) = (1 + g(\boldsymbol{y})) \prod_{i=1}^{M-1} \cos\Big(\frac{\pi x_i}{2}\Big) \\
& f_{j=2:M-1}(\boldsymbol{x}) = (1 + g(\boldsymbol{y}))\Big(\prod_{i=1}^{M-j} \cos\Big(\frac{\pi x_i}{2}\Big)\Big) \sin\Big(\frac{\pi x_{M-j+1}}{2}\Big) \\
& f_M(\boldsymbol{x}) = (1 + g(\boldsymbol{y})) \sin(\frac{\pi x_1}{2}) \\
\text{where} \quad & \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = 100\Big[k + \sum_{i=1}^{k}((y_i - 0.5)^2 - \cos(20\pi(y_i - 0.5)))\Big], \\
\text{subject to} \quad & 0 \leq x_i \leq 1, \text{ for } i = 1, 2, ..., n
\end{aligned} \tag{A.3}$$

The Pareto optimal solutions correspond to $x_i = 0.5$ for all $x_i \in \boldsymbol{y}$. The number of variables is defined by $n = M + k - 1$, where a value of $k = 10$ is suggested.

DTLZ4

It is a multi-objective problem with a concave Pareto Front. Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\boldsymbol{x}) = (1 + g(\boldsymbol{y})) \prod_{i=1}^{M-1} \cos\Big(\frac{\pi x_i^\alpha}{2}\Big) \\
& f_{j=2:M-1}(\boldsymbol{x}) = (1 + g(\boldsymbol{y}))\Big(\prod_{i=1}^{M-j} \cos\Big(\frac{\pi x_i^\alpha}{2}\Big)\Big) \sin\Big(\frac{\pi x_{M-j+1}^\alpha}{2}\Big) \\
& f_M(\boldsymbol{x}) = (1 + g(\boldsymbol{y})) \sin\Big(\frac{\pi x_1^\alpha}{2}\Big) \\
\text{where} \quad & \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = \sum_{i=1}^{k}(y_i - 0.5)^2, \\
\text{subject to} \quad & 0 \leq x_i \leq 1, \text{ for } i = 1, 2, ..., n
\end{aligned} \tag{A.4}$$

The number of variables is defined by $n = M + k - 1$, where a value of $k = 10$ is suggested. The parameter $\alpha = 100$ is also recommended by its authors.

DTLZ5

Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\prod_{i=1}^{M-1}\cos\Big(\frac{\pi\theta_i}{2}\Big) \\
& f_{j=2:M-1}(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\Big(\prod_{i=1}^{M-j}\cos\Big(\frac{\pi\theta_i}{2}\Big)\Big)\sin\Big(\frac{\pi\theta_{M-j+1}}{2}\Big) \\
& f_M(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\sin(\frac{\pi\theta_1}{2}) \\
\text{where} \quad & \theta_i = \begin{cases} x_1 & \text{if } i=1 \\ \frac{1+2g(\boldsymbol{y})x_i}{4(1+g(\boldsymbol{y}))} & \text{for } i=2,3,\ldots,(M-1), \end{cases} \\
& \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = \sum_{i=1}^{k}(y_i-0.5)^2, \\
\text{subject to} \quad & 0 \le x_i \le 1, \text{ for } i=1,2,...,n.
\end{aligned} \tag{A.5}$$

The Pareto optimal solutions correspond to $x_i = 0.5$ for all $x_i \in \boldsymbol{y}$, where the objective function values must satisfy $\sum_{m=1}^{M} f_m^2 = 1$. The number of variables is defined by $n = M + k - 1$, where a value of $k = 10$ is suggested.

DTLZ6

Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_1(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\prod_{i=1}^{M-1}\cos\Big(\frac{\pi\theta_i}{2}\Big) \\
& f_{j=2:M-1}(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\Big(\prod_{i=1}^{M-j}\cos\Big(\frac{\pi\theta_i}{2}\Big)\Big)\sin\Big(\frac{\pi\theta_{M-j+1}}{2}\Big) \\
& f_M(\boldsymbol{x}) = (1+g(\boldsymbol{y}))\sin(\frac{\pi\theta_1}{2}) \\
\text{where} \quad & \theta_i = \begin{cases} x_1 & \text{if } i=1 \\ \frac{1+2g(\boldsymbol{y})x_i}{4(1+g(\boldsymbol{y}))} & \text{for } i=2,3,\ldots,(M-1), \end{cases}, \\
& \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = \sum_{i=1}^{k} y_i^{0.1}, \\
\text{subject to} \quad & 0 \le x_i \le 1, \text{ for } i=1,2,...,n
\end{aligned} \tag{A.6}$$

The Pareto optimal solutions correspond to $x_i = 0$ for all $x_i \in \boldsymbol{y}$. The number of variables is defined by $n = M + k - 1$, where a value of $k = 10$ is suggested.

DTLZ7

It is a problem with a disconnected Pareto Front. Let M be the number of objectives, then the mathematical formulation of this problem is the following:

$$\begin{aligned}
\text{Minimize} \quad & f_{j=1:M-1}(\boldsymbol{x}) = x_j, \\
& f_M(\boldsymbol{x}) = (1 + g(\boldsymbol{y}))\Big(M - \sum_{i=1}^{M-1}\Big[\frac{f_i(\boldsymbol{x})}{1 + g(\boldsymbol{y})}(1 + \sin(3\pi f_i(\boldsymbol{x})))\Big]\Big) \\
\text{where} \quad & \boldsymbol{y} = \{x_M, ..., x_{M+k-1}\}, \\
& g(\boldsymbol{y}) = 1 + 9\sum_{i=1}^{k}\frac{y_i}{k} \\
\text{subject to} \quad & 0 \leq x_i \leq 1, \text{ for } i = 1, 2, ..., n
\end{aligned} \tag{A.7}$$

The Pareto optimal solutions correspond to $x_i = 0$ for all $x_i \in \boldsymbol{y}$. The number of variables is defined by $n = M + k - 1$, where a value of $k = 20$ is suggested.

A.2 Walking Fish Group test suite

The Walking Fish Group (WFG) test suite consists of nine scalable problems (WFG1-WFG9) proposed by Huband et al. in 2005 [67]. Each problem is defined in terms of an underlying vector of parameters $\boldsymbol{x}$, obtained through a series of transition vectors from a vector $\boldsymbol{z}$. The vector $\boldsymbol{z}$ consists of $k + l = n$ working parameters, where the first k are position parameters and the last l are distance-related parameters. It is worth noticing that n must be larger or equal to M (the number of objectives), and k must be divisible by $(M - 1)$.

Additionally, every problem of the WFG test suite is associated with shape functions that determine the nature of the Pareto optimal front, which can be linear, convex, concave, mixed, or disconnected. Table A.2 displays the shape functions employed in the problems.

On the other hand, the fitness landscape of each problem is defined through transformation functions. There are three types: bias, shift, and reduction. The bias functions impact the search process as they bias the fitness landscape. The shift functions move the location of the optima, which helps to define its position. Furthermore, reduction functions impact the separability of the problem. Table A.3 presents the transformation functions employed in these problems.

Table A.4 summarizes the main characteristics of the nine WFG test problems. Moreover, their mathematical definitions are presented below.

Table A.2: Shape functions. In every function it holds that $x_1, ..., x_{M-1} \in [0, 1]$. A and B are constants.

Name	**Function**
Linear	$\text{linear}_1(x_1, ..., x_{M-1}) = \prod_{i=1}^{M-1} x_i$ $\text{linear}_{m=2:M-1}(x_1, ..., x_{M-1}) = \Big(\prod_{i=1}^{M-m} x_i \Big)(1 - x_{M-m+1})$ $\text{linear}_M(x_1, ..., x_{M-1}) = 1 - x_1$
Convex	$\text{convex}_1(x_1, ..., x_{M-1}) = \prod_{i=1}^{M-1} \Big(1 - \cos\Big(\frac{\pi x_i}{2}\Big)\Big)$ $\text{convex}_{m=2:M-1}(x_1, ..., x_{M-1}) = \Big(\prod_{i=1}^{M-m} \Big(1 - \cos\Big(\frac{\pi x_i}{2}\Big)\Big)\Big)\Big(1 - \sin\Big(\frac{\pi x_{M-m+1}}{2}\Big)\Big)$ $\text{convex}_M(x_1, ..., x_{M-1}) = 1 - \sin\Big(\frac{\pi x_1}{2}\Big)$
Concave	$\text{concave}_1(x_1, ..., x_{M-1}) = \prod_{i=1}^{M-1} \sin\Big(\frac{\pi x_i}{2}\Big)$ $\text{concave}_{m=2:M-1}(x_1, ..., x_{M-1}) = \Big(\prod_{i=1}^{M-m} \sin\Big(\frac{\pi x_i}{2}\Big)\Big) \cos\Big(\frac{\pi x_{M-m+1}}{2}\Big)$ $\text{concave}_M(x_1, ..., x_{M-1}) = \cos\Big(\frac{\pi x_1}{2}\Big)$
Mixed convex/ concave	$\text{mixed}_M(x_1, ..., x_{M-1}) = \Big(1 - x_1 - \frac{\cos(2A\pi x_1 + \pi/2)}{2A\pi}\Big)^{\alpha}$ where $\alpha > 0, A \in \{1, 2, ...\}$
Disconnected	$disc_M(x_1, ..., x_{M-1}) = 1 - (x_1)^{\alpha} \cos^2(A(x_1)^{\beta}\pi)$ where $\alpha, \beta > 0, A \in \{1, 2, ...\}$

Table A.3: Transformation functions. The parameters y and $y_1, ..., y_{|\boldsymbol{y}|}$ always have the domain $[0, 1]$. A, B and C are constants.

Name	**Function**
Bias: Polynomial	$$b\_\text{poly}(y) = y^{\alpha}$$ where $\alpha > 0, \alpha \neq 1$.
Bias: Flat Region	$$b\_\text{flat}(y, A, B, C) = A + min(0, \lfloor y - B \rfloor)\frac{A(B-y)}{B} - min(0, \lfloor C - y \rfloor)\frac{(1-A)(y-C)}{1-C}$$ where $A, B, C \in [0, 1], B < C, B = 0 \implies A = 0 \wedge C \neq 1, C = 1 \implies A = 1 \wedge B \neq 0$.
Bias: Parameter Dependent	$$b\_\text{param}(y, \boldsymbol{y'}, A, B, C) = y^{B+(C-B)v(u(\boldsymbol{y'}))}$$ $$v(u(\boldsymbol{y'})) = A - (1 - 2u(\boldsymbol{y'}))\Big\lfloor 0.5 - u(\boldsymbol{y'}) \rfloor + A\Big\rfloor$$ where $A \in (0, 1), 0 < B < C$.
Shift: Linear	$$s\_\text{linear}(y, A) = \frac{\lvert y - A \rvert}{\lvert \lfloor A - y \rfloor + A \rvert}$$ where $A \in (0, 1)$.
Shift: Deceptive	$$s\_\text{decept}(y, A, B, C) = 1 + (\lvert y - A \rvert - B)\Big(\frac{\lfloor y - A + B \rfloor (1 - C + \frac{A-B}{B})}{A - B} + \frac{\lfloor A + B - y \rfloor (1 - C + \frac{1-A-B}{B})}{1 - A - B} + \frac{1}{B}\Big)$$ where $A \in (0, 1), 0 < B << 1, 0 < C << 1, A - B > 0, A + B < 1$.
Shift: Multimodal	$$s\_\text{multi}(y, A, B, C) = \frac{1 + \cos\Big[(4A+2)\pi\Big(0.5 - \frac{\lvert y-C \rvert}{2(\lfloor C-y \rfloor + C)}\Big)\Big] + 4B\Big(\frac{\lvert y-C \rvert}{2(\lfloor C-y \rfloor + C)}\Big)^2}{B+2}$$ where $A \in \{1, 2, ...\}, B \geq 0, (4A+2)\pi \geq 4B, C \in (0, 1)$
Reduction: Weighted Sum	$$r\_\text{sum}(\boldsymbol{y}, \boldsymbol{w}) = \Big(\sum_{i=1}^{\boldsymbol{y}} w_i y_i\Big) / \sum_{i=1}^{|\boldsymbol{y}|} w_i$$ where $\lvert \boldsymbol{w} \rvert = \lvert \boldsymbol{y} \rvert, w_1, ..., w_{|y|} > 0)$.
Reduction: Non-separable	$$r\_\text{nonsep}(\boldsymbol{y}, A) = \frac{\sum_{j=1}^{|\boldsymbol{y}|}\Big(y_j + \sum_{k=0}^{A-2} \lvert y_j - y_{1+(j+k) \mod \lvert \boldsymbol{y} \rvert} \rvert\Big)}{\frac{\lvert y \rvert}{A}\lceil A/2 \rceil (1 + 2A - 2\lceil A/2 \rceil)}$$ where $A \in \{1, ..., \lvert \boldsymbol{y} \rvert\}, \lvert y \rvert \mod A = 0$

Table A.4: Main characteristics of WFG problems

Problem	**Separability**	**Frontality**	**Geometry**
WFG1	separable	unifrontal	convex, mixed
WFG2	non-separable	multifrontal	convex, disconnected
WFG3	non-separable	unifrontal	linear, degenerate
WFG4	separable	multifrontal	concave
WFG5	separable	deceptive	concave
WFG6	non-separable	unifrontal	concave
WFG7	separable	unifrontal	concave
WFG8	non-separable	unifrontal	concave
WFG9	non-separable	multifrontal	concave

WFG1

It is a unifrontal separable problem with a mixed Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$
\begin{aligned}
\text{Minimize} \quad f_1(\boldsymbol{x}) &= x_M + 2\text{convex}_1(x_1, ..., x_{M-1}) \\
f_{m=2:M-1}(\boldsymbol{x}) &= x_M + (2m)\text{convex}_m(x_1, ..., x_{M-1}) \\
f_M(\boldsymbol{x}) &= x_M + (2M)\text{mixed}_M(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:M-1} &= r\_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \\
&\qquad \{2(i-1)k/(M-1)+1, ..., 2ik/(M-1))\}) \\
x_M &= r\_\text{sum}(\{y_{k+1}, ..., y_n\}, \{2(k+1), ..., 2n\}) \\
y_{i=1:n} &= b\_\text{poly}(y'_i, 0.02) \\
y'_{i=1:k} &= y''_i \\
y'_{i=k+1:n} &= b\_\text{flat}(y''_i, 0.8, 0.75, 0.85) \\
y''_{i=1:k} &= \frac{z_i}{2i} \\
y''_{i=k+1:n} &= s\_\text{linear}\Big(\frac{z_i}{2i}, 0.35\Big)
\end{aligned}
\tag{A.8}
$$

where M is the number of objective functions and the mixed_M function has the parameters $\alpha = 1$ and $A = 5$.

WFG2

It is a multifrontal non-separable problem with a disconnected Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given

as follows:

$$\begin{aligned}
\text{Minimize} \quad f_1(\boldsymbol{x}) &= x_M + 2\text{convex}_1(x_1, ..., x_{M-1}) \\
f_{m=2:M-1}(\boldsymbol{x}) &= x_M + (2m)\text{convex}_m(x_1, ..., x_{M-1}) \\
f_M(\boldsymbol{x}) &= x_M + (2M)\text{disc}_M(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:M-1} &= r_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \{1, ..., 1\}) \\
x_M &= r_\text{sum}(\{y_{k+1}, ..., y_{k+l/2}\}, \{1, ..., 1\}) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:k+l/2} &= r_\text{nonsep}(\{y'_{k+2(i-k)-2}, y'_{k+2(i-k)}\}, 2) \\
y'_{i=1:k} &= \frac{z_i}{2i} \\
y'_{i=k+1:n} &= s_\text{linear}\left(\frac{z_i}{2i}, 0.35\right)
\end{aligned} \tag{A.9}$$

where M is the number of objective functions, l must be a multiple of two, and the disc_M function has the parameters $\alpha = 1$ and $A = 5$.

WFG3

It is a unifrontal non-separable problem with a degenerate Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$\begin{aligned}
\text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{linear}_m(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:M-1} &= r_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \{1, ..., 1\}) \\
x_M &= r_\text{sum}(\{y_{k+1}, ..., y_{k+l/2}\}, \{1, ..., 1\}) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:k+l/2} &= r_\text{nonsep}(\{y'_{k+2(i-k)-2}, y'_{k+2(i-k)}\}, 2) \\
y'_{i=1:k} &= \frac{z_i}{2i} \\
y'_{i=k+1:n} &= s_\text{linear}\left(\frac{z_i}{2i}, 0.35\right)
\end{aligned} \tag{A.10}$$

where M is the number of objective functions, and l must be a multiple of two.

WFG4

It is a multifrontal separable problem with a concave Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$\begin{aligned}
\text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{concave}_m(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:m-1} &= r_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \{1, ..., 1\}) \\
x_M &= r_\text{sum}(\{y_{k+1}, ..., y_n\}, \{1, ..., 1\}) \\
y_{i=1:M-1} &= s_\text{multi}(z_i/(2i), 30, 10, 0.35)
\end{aligned} \tag{A.11}$$

where M is the number of objective functions.

WFG5

It is a deceptive separable problem with a concave Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$\begin{aligned} \text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{concave}_m(x_1, ..., x_{M-1}) \\ \text{where} \quad x_{i=1:m-1} &= r\_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \{1, ..., 1\}) \\ x_M &= r\_\text{sum}(\{y_{k+1}, ..., y_n\}, \{1, ..., 1\}) \\ y_{i=1:M-1} &= s\_\text{decept}(z_i/(2i), 0.35, 0.001, 0.05) \end{aligned} \tag{A.12}$$

where M is the number of objective functions.

WFG6

It is a unifrontal non-separable problem with a concave Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$\begin{aligned} \text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{concave}_m(x_1, ..., x_{M-1}) \\ \text{where} \quad x_{i=1:M-1} &= r\_\text{nonsep}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \\ &\qquad k/(M-1)) \\ x_M &= r\_\text{nonsep}(\{y_{k+1}, ..., y_n\}, l) \\ y_{i=1:k} &= \frac{z_i}{2i} \\ y_{i=k+1:n} &= s\_\text{linear}\Big(\frac{z_i}{2i}, 0.35\Big) \end{aligned} \tag{A.13}$$

where M is the number of objective functions.

WFG7

It is a unifrontal separable problem with a concave Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as

follows:

$$\begin{aligned}
\text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{concave}_m(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:M-1} &= r_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \{1, ..., 1\}) \\
x_M &= r_\text{sum}(\{y_{k+1}, ..., y_n\}, \{1, ..., 1\}) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:n} &= s_\text{linear}(y'_i, 0.35) \\
y'_{i=1:k} &= b_\text{param}\Big(\frac{z_i}{2i}, r_\text{sum}(\{\frac{z_{i+1}}{2(i+1)}, ..., \frac{z_n}{2n}\}, \{1, ..., 1\}), \frac{0.98}{49.98}, 0.02, 50\Big) \\
y'_{i=k+1:n} &= \frac{z_i}{2i}
\end{aligned} \tag{A.14}$$

where M is the number of objective functions.

WFG8

It is a unifrontal non-separable problem with a concave Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$\begin{aligned}
\text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{concave}_m(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:M-1} &= r_\text{sum}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, \{1, ..., 1\}) \\
x_M &= r_\text{sum}(\{y_{k+1}, ..., y_n\}, \{1, ..., 1\}) \\
y_{i=1:k} &= y'_i \\
y_{i=k+1:n} &= s_\text{linear}(y'_i, 0.35) \\
y'_{i=1:k} &= \frac{z_i}{2i} \\
y'_{i=k+1:n} &= b_\text{param}(\frac{z_i}{2i}, r_\text{sum}(\{\frac{z_1}{2}, ..., \frac{z_{i-1}}{2(i-1)}.\{1, ..., 1\}), \frac{0.98}{49.98}, 0.02, 50)
\end{aligned} \tag{A.15}$$

where M is the number of objective functions.

WFG9

It is a multifrontal non-separable problem with a concave Pareto Front. Given a vector $\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_n\}$, the mathematical formulation of this problem is given as follows:

$$
\begin{aligned}
\text{Minimize} \quad f_{m=1:M}(\boldsymbol{x}) &= x_M + (2m)\text{concave}_m(x_1, ..., x_{M-1}) \\
\text{where} \quad x_{i=1:M-1} &= r\_\text{nonsep}(\{y_{(i-1)k/(M-1)+1}, ..., y_{ik/(M-1)}\}, k/(M-1)) \\
x_M &= r\_\text{nonsep}(\{y_{k+1}, ..., y_n\}, l) \\
y_{i=1:k} &= s\_\text{decept}(y'_i, 0.35, 0.001, 0.05) \\
y_{i=k+1:n} &= s\_\text{multi}(y'_i, 30, 95, 0.35) \\
y'_{i=1:n-1} &= b\_\text{param}(\frac{z_i}{2i}, r\_\text{sum}(\{\frac{z_{i+1}}{i+1}, ..., \frac{z_n}{2n}\}, \{1, ..., 1\}), \frac{0.98}{49.98}, 0.02, 50) \\
y'_n &= \frac{z_n}{2n}
\end{aligned}
\tag{A.16}
$$

where M is the number of objective functions.

A.3 Minus test problems

The Minus problems were proposed by Ishibuchi et al. [77] to show that most state-of-the-art algorithms are overspecialized in the DTLZ and WFG test problems. Given a multi-objective problem of the form:

$$\text{Minimize } f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in X. \tag{A.17}$$

The core idea is multiplying all objective functions by (-1). Therefore, the problem is transformed as follows:

$$\text{Minimize } -f_1(\boldsymbol{x}), \ldots, -f_M(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in X. \tag{A.18}$$

Appendix B

Experimental results

B.1 Study of ESW performance

Tables B.1 and B.2 show the average and standard deviation of the hypervolume indicator of HDE in conventional and minus problems. These results are related to the experiments shown in Chapter 4.

Tables B.3 and B.4 show the average and standard deviation of hypervolume values of the comparison between HDE using individual pairs (scalarizing function and weight vectors) and our proposed ESW. Moreover, Tables B.5 and B.6 shows the average and standard deviation of hypervolume and s-energy values of the comparison of ESW with state-of-the-art algorithms. These results are related to the experiments shown in Chapter 4.

Table B.1: Average and standard deviation of hypervolume values of HDE using different scalarizing functions and weight vectors in conventional problems. The two best values are highlighted in gray (dark gray is the best, and light gray is the second best). The "*" indicates that the result is statistically significant.

	SLD						UDH					
	AASF	AGSF2	ASF	ATCH	PBI	TCH	AASF	AGSF2	ASF	ATCH	PBI	TCH
dtlz1	1.331e+00 (4.44e-16)	1.326e+00 (1.27e-03)	1.331e+00 (4.44e-16)	1.331e+00 (4.44e-16)	1.331e+00 (4.44e-16)	1.331e+00 (3.59e-07)	1.331e+00 (1.58e-06)	1.327e+00 (1.31e-03)	1.331e+00 (1.98e-06)	1.331e+00 (5.47e-07)	1.331e+00 (3.24e-06)	1.331e+00 (7.48e-07)
dtlz2	9.647e-01 (1.54e-04)	9.659e-01 (1.59e-04)*	9.648e-01 (1.52e-04)	9.513e-01 (5.14e-04)	9.182e-01 (7.18e-03)	9.440e-01 (1.46e-03)	9.620e-01 (4.47e-04)	9.629e-01 (4.51e-04)	9.619e-01 (3.70e-04)	9.573e-01 (6.29e-04)	9.188e-01 (4.93e-03)	9.546e-01 (9.39e-04)
dtlz3	1.331e+00 (2.40e-06)	1.326e+00 (1.10e-02)	1.331e+00 (2.64e-05)	1.330e+00 (3.57e-03)	1.331e+00 (2.01e-03)	1.331e+00 (6.06e-06)	1.331e+00 (7.11e-05)	1.331e+00 (1.33e-03)	1.330e+00 (3.47e-03)	1.330e+00 (3.50e-03)	1.331e+00 (1.63e-05)	1.331e+00 (3.04e-06)
dtlz4	9.174e-01 (1.47e-03)	9.171e-01 (7.86e-03)*	9.170e-01 (3.65e-03)	9.025e-01 (1.33e-03)	8.691e-01 (5.69e-03)	8.956e-01 (1.92e-03)	9.117e-01 (8.20e-03)	9.111e-01 (1.26e-02)	9.130e-01 (4.94e-03)	9.059e-01 (1.22e-02)	8.651e-01 (6.66e-03)	9.015e-01 (1.75e-02)
dtlz5	1.097e+00 (4.94e-06)	1.097e+00 (8.55e-06)	1.097e+00 (6.08e-06)	1.098e+00 (4.01e-06)	1.097e+00 (4.74e-04)	1.098e+00 (3.35e-06)	1.094e+00 (3.46e-03)	1.090e+00 (6.88e-04)	1.093e+00 (1.55e-03)	1.099e+00 (5.66e-06)	1.096e+00 (5.43e-04)	1.100e+00 (9.91e-06)*
dtlz6	1.094e+00 (3.22e-06)	1.095e+00 (2.10e-05)	1.094e+00 (1.32e-05)	1.096e+00 (1.97e-06)	1.095e+00 (3.82e-04)	1.096e+00 (1.65e-05)	1.089e+00 (1.17e-04)	1.087e+00 (5.21e-04)	1.089e+00 (1.45e-04)	1.097e+00 (2.21e-06)*	1.096e+00 (4.07e-04)	1.097e+00 (9.37e-05)
dtlz7	6.342e-01 (1.37e-04)	6.346e-01 (2.16e-04)*	6.339e-01 (1.74e-04)	6.098e-01 (2.88e-03)	6.236e-01 (9.32e-04)	6.153e-01 (2.62e-04)	6.339e-01 (1.09e-03)	6.328e-01 (1.11e-03)	6.340e-01 (1.05e-03)	6.250e-01 (1.35e-03)	6.068e-01 (1.49e-03)	6.269e-01 (9.68e-04)
wfg1	1.217e+00 (9.38e-03)	1.218e+00 (1.13e-02)	1.218e+00 (1.24e-02)	1.192e+00 (9.09e-03)	1.148e+00 (7.83e-03)	1.187e+00 (1.08e-02)	1.183e+00 (9.21e-03)	1.185e+00 (8.15e-03)	1.182e+00 (8.58e-03)	1.186e+00 (8.92e-03)	1.123e+00 (7.81e-03)	1.189e+00 (1.27e-02)
wfg2	1.215e+00 (1.23e-02)	1.219e+00 (1.29e-02)	1.217e+00 (1.30e-02)	1.211e+00 (1.25e-02)	1.110e+00 (1.71e-02)	1.209e+00 (1.50e-02)	1.211e+00 (1.37e-02)	1.216e+00 (1.24e-02)	1.214e+00 (1.59e-02)	1.211e+00 (1.08e-02)	1.059e+00 (2.43e-02)	1.208e+00 (1.08e-02)
wfg3	9.486e-01 (6.71e-03)*	9.441e-01 (6.77e-03)	9.444e-01 (7.25e-03)	9.288e-01 (8.92e-03)	8.754e-01 (9.90e-03)	9.244e-01 (7.82e-03)	9.388e-01 (7.37e-03)	9.366e-01 (7.44e-03)	9.374e-01 (6.30e-03)	9.337e-01 (8.37e-03)	8.568e-01 (1.13e-02)	9.283e-01 (9.85e-03)
wfg4	9.172e-01 (5.80e-03)	9.191e-01 (7.12e-03)	9.178e-01 (6.07e-03)	9.098e-01 (4.56e-03)	8.095e-01 (9.18e-03)	9.030e-01 (4.55e-03)	9.054e-01 (4.87e-03)	9.086e-01 (4.19e-03)	9.071e-01 (4.80e-03)	9.109e-01 (3.91e-03)	7.507e-01 (1.11e-02)	9.045e-01 (5.59e-03)
wfg5	7.583e-01 (4.74e-03)	7.569e-01 (4.04e-03)	7.577e-01 (4.51e-03)	7.564e-01 (1.92e-03)	7.711e-01 (3.24e-03)*	7.409e-01 (2.35e-03)	7.627e-01 (3.58e-03)	7.613e-01 (2.99e-03)	7.611e-01 (3.30e-03)	7.593e-01 (3.00e-03)	7.448e-01 (3.23e-03)	7.509e-01 (3.83e-03)
wfg6	8.058e-01 (9.24e-05)	8.077e-01 (5.57e-05)*	8.058e-01 (8.36e-05)	7.855e-01 (3.75e-04)	8.037e-01 (7.75e-04)	7.725e-01 (1.08e-03)	7.996e-01 (9.84e-04)	8.003e-01 (1.09e-03)	7.990e-01 (9.38e-04)	7.934e-01 (9.73e-04)	7.851e-01 (1.76e-03)	7.859e-01 (1.43e-03)
wfg7	9.557e-01 (4.04e-03)	9.562e-01 (4.20e-03)	9.555e-01 (4.62e-03)	9.488e-01 (4.64e-03)	7.693e-01 (8.52e-03)	9.406e-01 (4.61e-03)	9.353e-01 (5.35e-03)	9.350e-01 (4.80e-03)	9.349e-01 (5.29e-03)	9.460e-01 (4.62e-03)	7.039e-01 (1.64e-02)	9.357e-01 (5.19e-03)
wfg8	8.824e-01 (5.58e-03)	8.824e-01 (8.23e-03)	8.808e-01 (5.45e-03)	8.744e-01 (7.55e-03)	7.580e-01 (1.07e-02)	8.681e-01 (6.55e-03)	8.793e-01 (6.20e-03)	8.802e-01 (6.34e-03)	8.809e-01 (5.64e-03)	8.805e-01 (5.86e-03)	7.123e-01 (1.28e-02)	8.743e-01 (6.50e-03)
wfg9	8.734e-01 (2.90e-03)	8.735e-01 (2.20e-03)	8.731e-01 (2.94e-03)	8.670e-01 (1.88e-03)	8.747e-01 (2.26e-03)	8.672e-01 (2.01e-03)	8.765e-01 (1.41e-03)	8.765e-01 (1.59e-03)	8.765e-01 (2.22e-03)	8.730e-01 (1.73e-03)	8.589e-01 (3.81e-03)	8.717e-01 (2.31e-03)

Table B.2: Average and standard deviation of hypervolume values of HDE using different scalarizing functions and weight vectors in minus problems. The two best values are highlighted in gray (dark gray is the best, and light gray is the second best). The "*" indicates that the result is statistically significant.

	SLD						UDH					
	AASF	AGSF2	ASF	ATCH	PBI	TCH	AASF	AGSF2	ASF	ATCH	PBI	TCH
dtlz1^{-1}	2.742e-01 (6.22e-05)	2.798e-01 (5.60e-04)	2.741e-01 (8.29e-05)	2.842e-01 (4.18e-04)	2.763e-01 (8.08e-04)	2.843e-01 (3.75e-04)	2.857e-01 (4.07e-04)	2.891e-01 (3.91e-04)*	2.855e-01 (4.36e-04)	2.812e-01 (4.07e-04)	2.778e-01 (1.49e-03)	2.812e-01 (4.90e-04)
dtlz2^{-1}	9.046e-01 (5.70e-04)	9.015e-01 (5.31e-04)	9.043e-01 (6.22e-04)	9.246e-01 (1.64e-04)	9.261e-01 (3.49e-04)	9.234e-01 (1.70e-04)	9.300e-01 (5.53e-04)	9.290e-01 (2.32e-04)	9.299e-01 (5.20e-04)	9.279e-01 (2.24e-04)	9.169e-01 (1.24e-03)	9.267e-01 (1.44e-04)
dtlz3^{-1}	6.360e-01 (1.26e-02)	6.331e-01 (1.03e-02)	6.397e-01 (9.25e-03)	6.382e-01 (9.12e-03)	6.114e-01 (8.08e-03)	6.350e-01 (1.07e-02)	6.449e-01 (9.66e-03)	6.400e-01 (9.03e-03)	6.412e-01 (9.54e-03)	6.337e-01 (8.98e-03)	5.924e-01 (1.03e-02)	6.310e-01 (1.01e-02)
dtlz4^{-1}	9.066e-01 (9.68e-04)	9.020e-01 (1.17e-03)	9.047e-01 (1.10e-03)	9.265e-01 (1.79e-04)	9.252e-01 (1.22e-03)	9.233e-01 (5.19e-04)	9.313e-01 (7.71e-04)*	9.292e-01 (6.59e-04)	9.296e-01 (6.78e-04)	9.264e-01 (3.13e-04)	9.173e-01 (1.40e-03)	9.263e-01 (7.53e-04)
dtlz5^{-1}	8.033e-01 (6.44e-04)	8.010e-01 (5.27e-04)	8.032e-01 (5.91e-04)	8.221e-01 (1.53e-04)	8.210e-01 (8.40e-04)	8.215e-01 (1.34e-04)	8.263e-01 (3.18e-04)	8.268e-01 (3.05e-04)*	8.263e-01 (2.37e-04)	8.215e-01 (3.20e-04)	8.112e-01 (1.93e-03)	8.206e-01 (3.72e-04)
dtlz6^{-1}	7.234e-01 (5.20e-04)	7.211e-01 (6.82e-04)	7.235e-01 (6.16e-04)	7.396e-01 (2.30e-04)	7.428e-01 (5.11e-04)	7.388e-01 (1.64e-04)	7.480e-01 (2.41e-04)	7.500e-01 (2.19e-04)*	7.479e-01 (2.23e-04)	7.385e-01 (3.33e-04)	7.415e-01 (9.53e-04)	7.369e-01 (3.14e-04)
dtlz7^{-1}	1.200e+00 (4.82e-04)	1.201e+00 (2.37e-04)	1.201e+00 (2.80e-04)	1.201e+00 (2.23e-04)	1.200e+00 (3.73e-04)	1.201e+00 (3.49e-03)	1.200e+00 (1.60e-03)	1.201e+00 (3.91e-04)	1.200e+00 (8.59e-04)	1.201e+00 (2.82e-04)	1.197e+00 (9.36e-04)	1.202e+00 (1.84e-04)*
wfg1^{-1}	1.355e-01 (4.72e-03)	1.368e-01 (5.50e-03)	1.379e-01 (4.73e-03)	1.378e-01 (3.88e-03)	1.401e-01 (3.49e-03)	1.402e-01 (3.75e-03)	1.414e-01 (4.67e-03)	1.422e-01 (4.25e-03)	1.412e-01 (4.24e-03)	1.375e-01 (4.25e-03)	1.390e-01 (3.79e-03)	1.397e-01 (3.82e-03)
wfg2^{-1}	5.435e-01 (1.25e-02)	5.430e-01 (1.15e-02)	5.426e-01 (1.06e-02)	5.416e-01 (1.11e-02)	4.977e-01 (9.91e-03)	5.377e-01 (1.26e-02)	5.454e-01 (1.19e-02)	5.481e-01 (8.62e-03)	5.479e-01 (1.01e-02)	5.414e-01 (1.10e-02)	4.892e-01 (9.58e-03)	5.387e-01 (1.03e-02)
wfg3^{-1}	3.240e-01 (1.10e-02)	3.232e-01 (1.02e-02)	3.229e-01 (1.04e-02)	3.194e-01 (1.04e-02)	3.144e-01 (1.26e-02)	3.186e-01 (9.95e-03)	3.206e-01 (1.05e-02)	3.209e-01 (9.08e-03)	3.203e-01 (9.55e-03)	3.137e-01 (8.62e-03)	3.017e-01 (1.53e-02)	3.141e-01 (9.91e-03)
wfg4^{-1}	6.901e-01 (5.80e-04)	6.878e-01 (5.94e-04)	6.899e-01 (4.97e-04)	7.046e-01 (3.79e-04)	7.105e-01 (3.83e-04)	7.028e-01 (3.36e-04)	7.151e-01 (9.12e-04)	7.169e-01 (7.93e-04)*	7.148e-01 (9.56e-04)	7.032e-01 (5.77e-04)	7.091e-01 (1.49e-03)	7.013e-01 (6.14e-04)
wfg5^{-1}	7.037e-01 (5.71e-03)	7.036e-01 (6.24e-03)	7.038e-01 (6.38e-03)	7.216e-01 (5.99e-03)	6.895e-01 (7.45e-03)	7.176e-01 (4.64e-03)	7.203e-01 (5.84e-03)	7.206e-01 (4.00e-03)	7.211e-01 (4.69e-03)	7.221e-01 (4.38e-03)	6.453e-01 (8.60e-03)	7.161e-01 (5.12e-03)
wfg6^{-1}	7.243e-01 (1.71e-03)	7.205e-01 (2.51e-03)	7.240e-01 (2.38e-03)	7.368e-01 (2.03e-03)	7.312e-01 (3.92e-03)	7.347e-01 (1.85e-03)	7.401e-01 (2.48e-03)	7.410e-01 (1.90e-03)	7.398e-01 (2.59e-03)	7.348e-01 (2.40e-03)	7.213e-01 (3.63e-03)	7.323e-01 (1.72e-03)
wfg7^{-1}	6.919e-01 (4.72e-04)	6.897e-01 (4.85e-04)	6.919e-01 (4.27e-04)	7.059e-01 (4.82e-04)	7.128e-01 (4.57e-04)	7.042e-01 (2.88e-04)	7.157e-01 (4.54e-04)	7.182e-01 (5.39e-04)*	7.158e-01 (4.16e-04)	7.044e-01 (5.95e-04)	7.126e-01 (8.50e-04)	7.027e-01 (4.09e-04)
wfg8^{-1}	7.021e-01 (6.55e-04)	6.997e-01 (5.55e-04)	7.018e-01 (5.16e-04)	7.173e-01 (6.02e-04)	7.219e-01 (9.67e-04)	7.148e-01 (3.23e-04)	7.268e-01 (6.44e-04)	7.286e-01 (7.37e-04)*	7.264e-01 (6.22e-04)	7.159e-01 (7.60e-04)	7.196e-01 (2.29e-03)	7.134e-01 (5.39e-04)
wfg9^{-1}	6.965e-01 (3.25e-03)	6.940e-01 (2.27e-03)	6.978e-01 (2.69e-03)	7.119e-01 (3.51e-03)	7.032e-01 (4.17e-03)	7.092e-01 (2.27e-03)	7.152e-01 (2.82e-03)	7.167e-01 (2.35e-03)*	7.153e-01 (2.37e-03)	7.112e-01 (2.56e-03)	6.935e-01 (3.86e-03)	7.098e-01 (2.74e-03)

Table B.3: Average and standard deviation of hypervolume values of the comparison of standalone pairs. The two best values are highlighted in gray (dark gray is the best, and light gray is the second best). The "*" indicates that the result is statistically significant.

		SLD		UDH	
	ESW	AASF	AGSF2	AASF	AGSF2
dtlz1	1.331e+00 (5.62e-07)	1.331e+00 (3.59e-07)	1.322e+00 (3.23e-03)	1.331e+00 (6.00e-06)	1.324e+00 (1.74e-03)
dtlz2	7.562e-01 (2.87e-04)	7.544e-01 (5.43e-05)	7.565e-01 (3.28e-05)*	7.465e-01 (7.80e-04)	7.468e-01 (1.10e-03)
dtlz3	1.331e+00 (1.85e-04)	1.331e+00 (2.04e-06)	1.330e+00 (1.89e-03)	1.331e+00 (6.22e-06)	1.330e+00 (4.31e-03)
dtlz4	7.515e-01 (2.10e-02)	7.538e-01 (9.13e-05)	7.531e-01 (1.48e-02)*	7.434e-01 (9.28e-03)	7.432e-01 (6.00e-03)
dtlz5	2.671e-01 (2.63e-05)*	2.571e-01 (7.82e-06)	2.591e-01 (3.63e-06)	2.649e-01 (3.88e-06)	2.649e-01 (3.02e-05)
dtlz6	2.671e-01 (2.40e-05)*	2.571e-01 (9.78e-06)	2.591e-01 (8.01e-07)	2.649e-01 (1.32e-05)	2.648e-01 (4.79e-05)
dtlz7	6.186e-01 (6.30e-04)*	6.102e-01 (1.01e-04)	6.108e-01 (8.84e-05)	6.090e-01 (9.65e-04)	6.078e-01 (7.19e-04)
wfg1	1.168e+00 (1.98e-02)	1.190e+00 (1.46e-02)	1.194e+00 (1.70e-02)	1.120e+00 (1.86e-02)	1.122e+00 (1.55e-02)
wfg2	1.248e+00 (1.54e-03)*	1.235e+00 (3.70e-03)	1.238e+00 (5.33e-03)	1.235e+00 (2.24e-03)	1.235e+00 (2.71e-03)
wfg3	8.828e-01 (2.06e-03)*	8.700e-01 (2.37e-03)	8.703e-01 (2.26e-03)	8.626e-01 (2.52e-03)	8.566e-01 (2.59e-03)
wfg4	7.589e-01 (5.84e-03)*	7.397e-01 (6.32e-03)	7.403e-01 (8.02e-03)	7.240e-01 (2.76e-03)	7.252e-01 (3.50e-03)
wfg5	7.444e-01 (3.53e-03)*	7.250e-01 (3.24e-03)	7.248e-01 (2.34e-03)	7.283e-01 (2.94e-03)	7.257e-01 (2.98e-03)
wfg6	7.557e-01 (2.40e-04)*	7.532e-01 (4.64e-05)	7.553e-01 (2.31e-05)	7.441e-01 (8.33e-04)	7.431e-01 (1.04e-03)
wfg7	7.662e-01 (9.00e-04)*	7.507e-01 (3.75e-03)	7.503e-01 (3.22e-03)	7.385e-01 (1.88e-03)	7.381e-01 (1.74e-03)
wfg8	7.520e-01 (3.55e-03)*	7.271e-01 (5.53e-03)	7.269e-01 (4.52e-03)	7.179e-01 (4.55e-03)	7.202e-01 (4.79e-03)
wfg9	8.380e-01 (1.07e-03)*	8.218e-01 (2.44e-03)	8.227e-01 (2.76e-03)	8.280e-01 (1.94e-03)	8.285e-01 (1.30e-03)
$dtlz1^{-1}$	3.050e-01 (1.69e-04)*	2.741e-01 (5.31e-05)	2.807e-01 (2.29e-04)	2.865e-01 (1.79e-04)	2.901e-01 (1.21e-04)
$dtlz2^{-1}$	9.388e-01 (2.62e-04)*	9.041e-01 (2.74e-04)	9.014e-01 (3.21e-04)	9.298e-01 (2.39e-04)	9.289e-01 (2.60e-04)
$dtlz3^{-1}$	7.194e-01 (9.72e-04)*	6.903e-01 (7.52e-04)	6.881e-01 (5.39e-04)	7.105e-01 (1.13e-03)	7.119e-01 (1.31e-03)
$dtlz4^{-1}$	9.387e-01 (2.81e-04)*	9.058e-01 (3.82e-04)	9.017e-01 (6.27e-04)	9.316e-01 (3.21e-04)	9.290e-01 (4.70e-04)
$dtlz5^{-1}$	9.647e-01 (2.14e-04)*	9.320e-01 (2.20e-04)	9.289e-01 (2.23e-04)	9.562e-01 (4.65e-04)	9.557e-01 (6.91e-04)
$dtlz6^{-1}$	8.122e-01 (2.87e-04)*	7.776e-01 (1.75e-04)	7.745e-01 (2.64e-04)	8.042e-01 (1.76e-04)	8.054e-01 (2.04e-04)
$dtlz7^{-1}$	1.156e+00 (1.02e-04)*	1.152e+00 (6.53e-04)	1.154e+00 (3.21e-04)	1.151e+00 (2.43e-03)	1.154e+00 (6.05e-04)
$wfg1^{-1}$	1.504e-01 (3.89e-03)*	1.334e-01 (4.69e-03)	1.380e-01 (3.83e-03)	1.400e-01 (3.86e-03)	1.431e-01 (2.57e-03)
$wfg2^{-1}$	4.082e-01 (9.21e-04)*	4.002e-01 (2.61e-04)	4.004e-01 (3.12e-04)	4.004e-01 (6.79e-04)	4.001e-01 (7.57e-04)
$wfg3^{-1}$	3.116e-01 (7.05e-04)*	2.814e-01 (5.12e-04)	2.874e-01 (3.84e-04)	2.914e-01 (2.17e-03)	2.940e-01 (1.53e-03)
$wfg4^{-1}$	7.214e-01 (4.59e-04)*	6.894e-01 (2.12e-04)	6.874e-01 (3.09e-04)	7.151e-01 (7.41e-04)	7.172e-01 (8.55e-04)
$wfg5^{-1}$	7.317e-01 (1.79e-03)*	6.970e-01 (1.60e-03)	6.951e-01 (1.28e-03)	7.017e-01 (2.12e-03)	7.006e-01 (2.14e-03)
$wfg6^{-1}$	7.485e-01 (2.07e-03)*	7.167e-01 (1.31e-03)	7.139e-01 (1.38e-03)	7.284e-01 (2.16e-03)	7.297e-01 (2.50e-03)
$wfg7^{-1}$	7.227e-01 (3.95e-04)*	6.904e-01 (2.22e-04)	6.886e-01 (2.34e-04)	7.151e-01 (4.71e-04)	7.174e-01 (3.63e-04)
$wfg8^{-1}$	7.228e-01 (4.00e-04)*	6.908e-01 (2.91e-04)	6.887e-01 (3.01e-04)	7.155e-01 (7.06e-04)	7.177e-01 (7.35e-04)
$wfg9^{-1}$	7.222e-01 (4.69e-03)*	6.952e-01 (3.18e-03)	6.933e-01 (3.64e-03)	7.122e-01 (3.11e-03)	7.126e-01 (3.11e-03)

Table B.4: Average and standard deviation of s-energy values of the comparison of standalone pairs. The two best values are highlighted in gray (dark gray is the best, and light gray is the second best). The "*" represents that the result is statistically significant.

		SLD		UDH	
	ESW	AASF	AGSF2	AASF	AGSF2
dtlz1	9.978e+08 (1.59e+08)	1.093e+09 (5.54e+04)	8.506e+07 (3.98e+08)	1.337e+09 (3.16e+06)	1.151e+07 (2.13e+07)
dtlz2	8.475e+04 (7.18e+01)*	8.948e+04 (8.53e+01)	8.628e+04 (4.68e+01)	1.002e+05 (7.06e+02)	9.865e+04 (4.27e+02)
dtlz3	5.650e+08 (9.79e+08)	3.940e+11 (1.05e+12)	1.627e+11 (8.30e+11)*	1.047e+12 (2.29e+12)	6.824e+08 (1.82e+08)
dtlz4	9.058e+04 (2.85e+04)*	9.006e+04 (3.27e+02)	4.427e+08 (2.38e+09)	1.321e+05 (1.21e+05)	1.792e+05 (4.31e+05)
dtlz5	1.494e+06 (4.36e+03)	7.951e+12 (8.84e+12)	1.590e+11 (1.47e+11)	6.285e+08 (2.00e+09)	1.265e+08 (9.90e+07)
dtlz6	1.493e+06 (3.50e+03)*	8.742e+12 (7.93e+12)	1.725e+11 (1.99e+11)	3.374e+08 (3.38e+08)	5.618e+08 (9.96e+08)
dtlz7	1.527e+05 (2.54e+03)*	2.208e+12 (4.68e+12)	5.290e+09 (4.89e+09)	1.398e+12 (3.44e+12)	2.897e+11 (1.56e+12)
wfg1	2.254e+05 (1.40e+04)*	4.779e+11 (1.28e+12)	3.468e+11 (6.17e+11)	3.394e+10 (1.81e+11)	5.183e+08 (2.23e+09)
wfg2	1.453e+05 (4.82e+03)*	2.990e+11 (8.05e+11)	3.350e+11 (7.79e+11)	8.551e+08 (2.30e+09)	1.432e+10 (6.11e+10)
wfg3	3.618e+05 (1.00e+04)*	1.081e+12 (1.01e+12)	1.066e+12 (1.22e+12)	5.437e+08 (1.63e+09)	4.583e+08 (1.99e+09)
wfg4	8.874e+04 (4.32e+02)*	1.198e+11 (3.98e+11)	4.513e+10 (1.83e+11)	1.222e+09 (3.50e+09)	3.446e+09 (1.71e+10)
wfg5	8.810e+04 (9.77e+02)*	3.449e+07 (3.68e+07)	1.490e+08 (4.21e+08)	4.975e+05 (1.66e+06)	4.962e+06 (1.87e+07)
wfg6	8.463e+04 (6.55e+01)*	8.928e+04 (7.44e+01)	8.610e+04 (3.57e+01)	1.004e+05 (4.60e+02)	9.936e+04 (3.04e+02)
wfg7	8.608e+04 (1.17e+02)*	7.882e+11 (1.01e+12)	2.369e+11 (3.11e+11)	4.671e+08 (1.63e+09)	9.104e+08 (3.16e+09)
wfg8	9.903e+04 (9.84e+02)*	8.889e+11 (1.20e+12)	6.502e+11 (1.06e+12)	2.166e+10 (5.75e+10)	2.228e+10 (6.45e+10)
wfg9	1.043e+05 (5.23e+02)*	2.343e+11 (5.49e+11)	5.213e+11 (8.09e+11)	2.619e+06 (6.14e+06)	3.630e+07 (1.63e+08)
dtlz1^{-1}	1.412e+05 (1.92e+02)*	4.218e+12 (2.81e+12)	1.338e+10 (2.49e+10)	6.819e+10 (3.59e+11)	5.942e+05 (1.35e+05)
dtlz2^{-1}	1.401e+05 (1.51e+02)*	1.810e+09 (4.84e+09)	1.186e+07 (4.12e+06)	9.128e+07 (4.69e+08)	6.670e+10 (3.59e+11)
dtlz3^{-1}	8.550e+04 (2.92e+02)*	5.656e+12 (3.58e+12)	4.574e+12 (2.72e+12)	1.188e+10 (2.54e+10)	1.154e+10 (3.18e+10)
dtlz4^{-1}	1.402e+05 (1.49e+02)*	1.748e+11 (3.48e+11)	3.238e+07 (5.52e+07)	1.219e+09 (4.85e+09)	1.028e+06 (5.13e+05)
dtlz5^{-1}	1.853e+05 (2.09e+02)*	2.140e+10 (8.13e+10)	3.591e+08 (1.34e+08)	1.451e+06 (1.86e+06)	1.939e+08 (5.99e+08)
dtlz6^{-1}	1.097e+05 (1.02e+02)*	5.409e+09 (2.03e+10)	5.080e+07 (6.82e+07)	8.816e+05 (1.17e+06)	8.136e+06 (2.43e+07)
dtlz7^{-1}	5.292e+05 (1.52e+04)*	9.253e+12 (2.21e+13)	8.558e+10 (4.48e+11)	9.083e+12 (2.62e+13)	8.051e+11 (2.49e+12)
wfg1^{-1}	2.357e+05 (3.69e+04)*	2.053e+10 (5.92e+10)	1.534e+12 (1.43e+12)	5.005e+10 (2.18e+11)	3.581e+11 (1.18e+12)
wfg2^{-1}	8.977e+04 (1.93e+02)*	2.771e+13 (3.02e+13)	2.315e+13 (1.93e+13)	2.184e+10 (6.10e+10)	8.469e+10 (2.52e+11)
wfg3^{-1}	1.427e+05 (2.62e+02)*	2.050e+13 (2.48e+13)	3.406e+13 (2.68e+13)	2.404e+10 (6.81e+10)	4.770e+09 (1.73e+10)
wfg4^{-1}	8.478e+04 (1.00e+02)*	2.148e+11 (8.31e+11)	2.131e+08 (7.62e+08)	9.581e+06 (4.67e+07)	2.261e+07 (1.15e+08)
wfg5^{-1}	9.645e+04 (1.34e+03)*	1.548e+11 (3.90e+11)	1.174e+11 (2.58e+11)	1.383e+08 (4.71e+08)	5.313e+07 (1.24e+08)
wfg6^{-1}	9.037e+04 (4.82e+02)*	9.099e+11 (1.04e+12)	7.835e+11 (1.05e+12)	1.728e+08 (3.95e+08)	1.506e+08 (3.38e+08)
wfg7^{-1}	8.485e+04 (9.93e+01)*	6.028e+11 (1.24e+12)	1.636e+11 (7.27e+11)	1.059e+11 (3.73e+11)	1.731e+10 (8.97e+10)
wfg8^{-1}	8.490e+04 (8.99e+01)*	1.371e+11 (4.99e+11)	4.776e+09 (2.56e+10)	4.763e+09 (2.56e+10)	8.681e+06 (4.07e+07)
wfg9^{-1}	9.902e+04 (8.49e+02)*	1.042e+10 (2.23e+10)	1.781e+10 (6.09e+10)	1.129e+06 (3.23e+06)	3.112e+05 (3.49e+05)

Table B.5: Average and standard deviation of hypervolume values of the comparison with state-of-the-art algorithms. The two best values are highlighted in gray (dark gray is the best, and light gray is the second best). The "*" indicates that the result is statistically significant.

	m	ESW	NSGA-III	MOEA/DD	SMS-EMOA(m = 3) / SMS-EMOAHype (m = 5,7,10)
dtlz1	3	1.2974e+0 (3.5e-2)	1.3062e+0 (4.3e-5)	1.3066e+0 (2.1e-5)	*1.3067e+0 (9.7e-6)
	5	1.6103e+0 (8.2e-7)	1.6103e+0 (4.5e-5)	1.6103e+0 (1.1e-6)	1.6099e+0 (1.6e-4)
	7	1.9478e+0 (2.2e-5)	1.9474e+0 (1.8e-3)	*1.9479e+0 (3.7e-6)	1.9472e+0 (2.6e-4)
	10	2.5937e+0 (8.9e-16)	2.5937e+0 (3.6e-7)	2.5937e+0 (3.6e-7)	2.5937e+0 (8.9e-16)
dtlz2	3	7.6057e-1 (2.8e-4)	7.5514e-1 (1.5e-4)	7.5890e-1 (1.1e-6)	*7.6811e-1 (4.7e-5)
	5	*1.3508e+0 (7.0e-4)	1.3468e+0 (5.3e-4)	1.3478e+0 (4.8e-6)	1.3474e+0 (1.5e-3)
	7	1.8399e+0 (1.1e-3)	1.8434e+0 (3.5e-4)	*1.8446e+0 (4.7e-6)	1.8378e+0 (1.2e-3)
	10	2.5936e+0 (2.5e-4)	2.5842e+0 (1.7e-2)	*2.5937e+0 (4.4e-16)	2.5936e+0 (5.9e-5)
dtlz3	3	1.3226e+0 (3.3e-2)	1.3308e+0 (1.4e-5)	1.3308e+0 (3.7e-7)	*1.3308e+0 (3.7e-7)
	5	1.6105e+0 (4.6e-5)	1.6105e+0 (2.2e-16)	1.6105e+0 (2.2e-16)	1.6105e+0 (2.2e-16)
	7	1.9487e+0 (4.4e-16)	1.9487e+0 (4.4e-16)	1.9487e+0 (4.4e-16)	1.9487e+0 (4.4e-16)
	10	2.5937e+0 (8.9e-16)	2.5937e+0 (6.0e-7)	2.5937e+0 (8.9e-16)	2.5937e+0 (8.9e-16)
dtlz4	3	7.849e-1 (2.e-2)	7.6273e-1 (1.1e-1)	7.8725e-1 (1.4e-6)	*7.1642e-1 (1.3e-1)
	5	1.3351e+0 (1.3e-3)	1.3311e+0 (2.5e-4)	1.3314e+0 (4.2e-6)	*1.3365e+0 (1.3e-3)
	7	1.8510e+0 (1.4e-3)	1.8190e+0 (7.5e-2)	*1.8533e+0 (1.8e-6)	1.8529e+0 (9.4e-4)
	10	2.5937e+0 (7.e-5)	2.5937e+0 (6.0e-5)	*2.5937e+0 (8.9e-16)	2.5937e+0 (3.1e-6)
dtlz7	3	1.2285e+0 (9.2e-5)	1.2222e+0 (3.7e-3)	1.2233e+0 (1.3e-4)	*1.228e+0 (5.6e-3)
	5	*1.4522e+0 (5.9e-4)	1.439e+0 (2.0e-3)	1.3898e+0 (4.e-2)	1.2622e+0 (1.1e-1)
	7	*1.6759e+0 (1.5e-2)	1.5796e+0 (5.2e-2)	4.5235e-1 (7.6e-2)	8.2143e-1 (5.1e-1)
	10	1.9117e+0 (2.2e-1)	1.9058e+0 (8.7e-2)	2.4389e-1 (3.9e-2)	1.9654e+0 (3.8e-1)
dtlz1^{-1}	3	*3.0505e-1 (1.7e-4)	2.8573e-1 (1.3e-3)	2.6546e-1 (6.1e-4)	1.9195e-1 (1.2e-2)
	5	1.9193e-2 (1.5e-4)	1.2290e-2 (1.1e-3)	1.0192e-2 (1.6e-4)	1.8941e-2 (1.1e-3)
	7	4.4404e-4 (6.3e-6)	3.3421e-4 (2.8e-5)	2.0216e-4 (6.9e-6)	*4.8428e-4 (3.7e-5)
	10	7.8126e-7 (3.7e-8)	1.0556e-6 (1.4e-7)	1.3846e-7 (1.6e-8)	*1.2840e-6 (1.2e-7)
dtlz2^{-1}	3	9.3876e-1 (2.6e-4)	9.2134e-1 (1.8e-3)	9.1874e-1 (6.6e-4)	*9.4007e-1 (1.8e-4)
	5	*4.7853e-1 (1.8e-3)	4.2668e-1 (4.4e-3)	3.5354e-1 (1.5e-3)	4.3612e-1 (7.4e-3)
	7	1.3525e-1 (8.7e-4)	1.1411e-1 (4.4e-3)	8.4628e-2 (1.6e-3)	*1.4347e-1 (3.7e-3)
	10	1.4428e-2 (2.4e-4)	1.3372e-2 (8.5e-4)	6.8312e-3 (3.1e-4)	*2.0770e-2 (5.4e-4)
dtlz3^{-1}	3	*7.1942e-1 (9.7e-4)	7.0608e-1 (2.7e-3)	7.0371e-1 (6.5e-4)	5.0008e-1 (1.9e-2)
	5	1.6358e-1 (2.1e-3)	1.2602e-1 (6.3e-3)	8.2782e-2 (2.7e-3)	*1.9704e-1 (5.8e-3)
	7	9.7422e-3 (1.1e-3)	9.0438e-3 (1.4e-3)	8.5155e-3 (4.5e-4)	*3.5046e-2 (1.3e-3)
	10	7.6167e-5 (1.5e-5)	4.1322e-4 (1.0e-4)	5.7394e-4 (8.1e-5)	*3.8553e-3 (3.4e-4)
dtlz4^{-1}	3	9.3872e-1 (2.8e-4)	9.2217e-1 (1.9e-3)	9.1861e-1 (4.1e-4)	*9.4017e-1 (1.8e-4)
	5	*4.7745e-1 (8.e-3)	4.264e-1 (4.7e-3)	3.4831e-1 (2.2e-3)	4.4191e-1 (6.4e-3)
	7	1.3462e-1 (8.2e-4)	1.0158e-1 (5.5e-3)	8.2317e-2 (7.e-4)	*1.4507e-1 (2.8e-3)
	10	8.429e-3 (1.6e-3)	1.0796e-2 (1.e-3)	6.9314e-3 (4.6e-4)	*2.1365e-2 (5.8e-4)
dtlz7^{-1}	3	1.3116e+0 (1.1e-5)	1.3105e+0 (5.9e-4)	1.3111e+0 (4.4e-5)	*1.3117e+0 (1.4e-6)
	5	1.5785e+0 (3.5e-4)	1.5649e+0 (3.0e-3)	1.5462e+0 (6.3e-2)	1.5728e+0 (1.1e-2)
	7	*1.8984e+0 (6.7e-4)	1.8419e+0 (8.2e-3)	1.1554e+0 (3.1e-2)	1.8642e+0 (1.4e-2)
	10	2.3558e+0 (3.5e-2)	2.3513e+0 (1.3e-2)	1.2544e+0 (3.5e-2)	*2.4844e+0 (4.1e-2)

Table B.6: Average and standard deviation of s-energy values of the comparison with state-of-the-art algorithms. The two best values are highlighted in gray (dark gray is the best, and light gray is the second best). The "*" indicates that the result is statistically significant.

	m	ESW	NSGA-III	MOEA/DD	SMS-EMOA(m = 3) / SMS-EMOAHype (m = 5,7,10)
dtlz1	3	*5.4852e+5 (8.8e+4)	6.7586e+5 (1.9e+3)	6.0073e+5 (3.5e+2)	6.1437e+5 (1.5e+3)
	5	*5.8482e+8 (9.9e+7)	4.8462e+15 (2.6e+16)	1.7714e+9 (1.1e+6)	1.9274e+11 (5.4e+11)
	7	*2.7427e+7 (2.6e+6)	2.0833e+33 (1.1e+34)	2.8939e+7 (7.1e+4)	1.6979e+30 (9.1e+30)
	10	*1.0902e+30 (4.e+30)	1.5708e+54 (1.9e+54)	3.1455e+29 (1.1e+28)	1.2687e+54 (2.2e+54)
dtlz2	3	*8.5177e+4 (7.2e+1)	1.1143e+5 (3.6e+3)	8.9408e+4 (1.3e+0)	1.1827e+5 (1.8e+3)
	5	*2.9926e+5 (9.6e+2)	3.5488e+5 (4.6e+2)	3.5543e+5 (2.1e+1)	2.6909e+10 (1.3e+11)
	7	*2.8434e+5 (6.e+3)	4.6017e+5 (1.5e+3)	4.6397e+5 (8.8e+1)	3.333e+20 (1.8e+21)
	10	*6.8291e+8 (2.9e+9)	3.4296e+48 (1.8e+49)	6.9118e+9 (3.2e+8)	1.0096e+29 (4.5e+29)
dtlz3	3	*1.5886e+7 (3.6e+7)	2.782e+7 (6.6e+6)	2.0173e+7 (3.2e+4)	2.3244e+7 (3.7e+5)
	5	*1.7255e+13 (2.1e+13)	4.0516e+13 (1.3e+14)	1.6377e+13 (6.3e+10)	1.5793e+25 (2.1e+25)
	7	*1.0611e+20 (5.1e+20)	2.6420e+34 (1.4e+35)	9.6665e+18 (7.9e+16)	2.0949e+36 (2.2e+36)
	10	*3.3870e+48 (1.8e+49)	1.6999e+55 (1.3e+55)	5.5979e+34 (1.3e+34)	4.3020e+55 (7.7e+55)
dtlz4	3	*9.4209e+4 (3.e+4)	1.1207e+5 (2.3e+4)	9.2487e+4 (1.3e+0)	8.8115e+5 (1.3e+6)
	5	*2.8891e+5 (4.8e+3)	3.3831e+5 (4.9e+2)	3.3869e+5 (1.4e+1)	7.9963e+11 (4.2e+12)
	7	*3.2145e+5 (5.9e+3)	5.6666e+29 (3.1e+30)	4.8228e+5 (6.3e+1)	3.9256e+16 (2.1e+17)
	10	*2.6341e+16 (1.4e+17)	9.3176e+46 (5.0e+47)	2.5528e+11 (3.e+10)	3.6538e+33 (2.e+34)
dtlz7	3	3.4745e+5 (8.5e+3)	6.7388e+7 (1.3e+8)	*5.1791e+5 (7.e+5)	7.6398e+5 (2.6e+5)
	5	*1.0179e+13 (5.5e+13)	1.3371e+15 (6.8e+15)	4.2344e+10 (1.3e+11)	3.9672e+19 (2.1e+20)
	7	*9.4473e+4 (3.1e+3)	1.913e+19 (6.7e+19)	9.2988e+10 (1.6e+10)	7.9183e+23 (4.2e+24)
	10	*2.3439e+3 (1.3e+2)	1.5425e+34 (8.3e+34)	7.8423e+12 (3.4e+13)	1.9316e+34 (7.1e+34)
dtlz1^{-1}	3	*1.4117e+5 (1.9e+2)	2.4779e+11 (9.2e+11)	7.8507e+5 (2.2e+6)	2.0554e+6 (6.6e+5)
	5	*1.3196e+6 (1.2e+4)	6.6751e+22 (3.6e+23)	2.7882e+9 (3.5e+9)	6.4933e+11 (3.4e+12)
	7	*2.5558e+6 (1.9e+4)	1.5617e+25 (8.2e+25)	1.2222e+17 (6.5e+17)	2.5159e+16 (1.0e+17)
	10	*1.6882e+8 (3.4e+7)	1.9274e+32 (8.4e+32)	9.5644e+23 (4.7e+24)	4.3405e+23 (2.3e+24)
dtlz2^{-1}	3	*1.4005e+5 (1.5e+2)	2.3182e+9 (1.2e+10)	1.8862e+5 (5.2e+4)	2.0916e+5 (1.7e+3)
	5	*7.5610e+5 (1.3e+4)	1.6784e+20 (9.e+20)	7.4462e+9 (3.5e+10)	1.9575e+10 (9.2e+10)
	7	*8.7556e+5 (6.8e+3)	5.2344e+19 (2.7e+20)	3.5629e+13 (7.7e+13)	4.1913e+14 (1.5e+15)
	10	*2.9202e+6 (6.4e+4)	1.7751e+29 (6.9e+29)	2.3295e+21 (1.0e+22)	1.4417e+20 (6.3e+20)
dtlz3^{-1}	3	*8.5503e+4 (2.9e+2)	2.2985e+7 (7.6e+7)	1.1212e+5 (4.1e+4)	5.6999e+6 (1.6e+6)
	5	*2.8263e+5 (2.8e+3)	1.0907e+13 (3.3e+13)	2.9823e+9 (1.5e+10)	1.4955e+11 (4.8e+11)
	7	*2.5871e+5 (9.2e+3)	5.6391e+18 (3.0e+19)	2.7063e+14 (1.5e+15)	2.2508e+15 (1.2e+16)
	10	*1.9501e+6 (1.3e+5)	6.8219e+31 (3.7e+32)	7.5909e+20 (3.8e+21)	3.7049e+17 (1.8e+18)
dtlz4^{-1}	3	*1.4023e+5 (1.5e+2)	1.4933e+9 (4.2e+9)	1.3408e+10 (3.4e+10)	2.0823e+5 (1.6e+3)
	5	*7.6196e+5 (5.2e+4)	6.9684e+22 (3.6e+23)	2.088e+23 (7.9e+23)	2.5420e+11 (9.3e+11)
	7	*8.8492e+5 (7.2e+3)	8.3333e+33 (4.5e+34)	1.0620e+33 (5.6e+33)	1.3106e+15 (6.8e+15)
	10	*5.6494e+6 (1.7e+6)	9.7702e+37 (5.3e+38)	6.4856e+47 (1.9e+48)	1.0617e+23 (5.2e+23)
dtlz7^{-1}	3	4.6499e+6 (1.8e+5)	7.4787e+10 (2.9e+11)	*7.2603e+5 (1.1e+5)	1.3095e+7 (2.2e+5)
	5	*7.3814e+6 (1.1e+6)	4.6987e+17 (2.1e+18)	2.1525e+7 (2.3e+7)	1.2511e+19 (6.7e+19)
	7	*1.8044e+6 (3.8e+4)	8.3523e+20 (4.0e+21)	4.7305e+11 (1.4e+12)	7.5992e+25 (3.8e+26)
	10	9.0018e+23 (4.3e+24)	6.3153e+29 (3.4e+30)	9.8209e+14 (3.1e+15)	3.1702e+38 (1.7e+39)

Bibliography

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

[2] S. C. Cerda-Flores, A. A. Rojas-Punzo, and F. Nápoles-Rivera. Applications of multi-objective optimization to industrial processes: A literature review. *Processes*, 10(1), 2022.

[3] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[4] Michael T. M. Emmerich and Andre H. Deutz. A Tutorial on Multiobjective Optimization: Fundamentals and Evolutionary Methods. *Natural Computing*, 17(3):585–609, September 2018.

[5] José A. Molinet Berenguer and Carlos A. Coello Coello. Evolutionary Many-Objective Optimization Based on Kuhn-Munkres' Algorithm. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 3–17. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.

[6] D. C. Valencia-Rodríguez and C. A. Coello Coello. A novel performance indicator based on the linear assignment problem. In Michael Emmerich, André Deutz, Hao Wang, Anna V. Kononova, Boris Naujoks, Ke Li, Kaisa Miettinen, and Iryna Yevseyeva, editors, *Evolutionary Multi-Criterion Optimization: 12th International Conference, EMO 2023*, pages 348–360. Springer Nature Switzerland, 2023.

[7] D. C. Valencia-Rodríguez and C. A. Coello Coello. Multi-Objective Evolutionary Algorithm Based on the Linear Assignment Problem and the Hypervolume Approximation Using Polar Coordinates (MOEA-LAPCO). In G. Rudolph, A. V. Kononova, H. Aguirre, P. Kerschke, G. Ochoa, and T. Tušar, editors, *Parallel Problem Solving from Nature – PPSN XVII*, pages 221–233, Cham, 2022. Springer International Publishing.

[8] D. C. Valencia-Rodríguez and C. A. Coello Coello. An Ensemble of Scalarizing Functions and Weight Vectors for Evolutionary Multi-Objective Optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC'2021)*, pages 2459–2467. IEEE Press, 2021.

[9] Diana Cristina Valencia-Rodríguez and Carlos A. Coello Coello. Influence of the Number of Connections Between Particles in the Performance of a Multi-Objective Particle Swarm Optimizer. *Swarm and Evolutionary Computation*, 77(101231), March 2023.

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

[11] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, 1999.

[12] 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 Heike Trautmann, Günter Rudolph, Kathrin Klamroth, Oliver Schütze, Margaret Wiecek, Yaochu Jin, and Christian Grimme, editors, *Evolutionary Multi-Criterion Optimization, 9th International Conference, EMO 2017*, pages 499–513. Springer. Lecture Notes in Computer Science Vol. 10173, Münster, Germany, March 19-22 2017. ISBN 978-3-319-54156-3.

[13] V. Joseph Bowman. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In Hervé Thiriez and Stanley Zionts, editors, *Multiple Criteria Decision Making*, pages 76–86, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg.

[14] Ralph Steuer and Eng Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26:326–344, 10 1983.

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

[16] A. V. Bernabé Rodríguez and C. A. Coello Coello. Generation of New Scalarizing Functions Using Genetic Programming. In Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann, editors, *Parallel Problem Solving from Nature – PPSN XVI*, pages 3–17. Springer. Lecture Notes in Computer Science, Leiden, The Netherlands, 2020.

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

[18] Henry Scheffé. Experiments with mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):344–360, 1958.

[19] Hisao Ishibuchi, Ryo Imada, Naoki Musayama, and Yusuke Nojima. Two-Layered Weight Vector Specification in Decomposition-Based Multi-Objective Algorithms for Many-Objective Optimization Problems. In *2019 IEEE Congress on Evolutionary Computation (CEC'2019)*, pages 2434–2441, Wellington, New Zealand, June 10 - June 13 2019. IEEE Press. ISBN 978-1-7281-2153-6.

[20] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014.

[21] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, 1997.

[22] Thomas Bäck and Hans-Paul Schwefel. Evolutionary computation: an overview. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 20–29, Nagoya, Japan, May 20 – 22 1996. IEEE.

[23] Kalyanmoy Deb and Ram Bhushan Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[24] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

[25] Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys*, 52(2), mar 2019.

[26] J. G. Falcón-Cardona, M. T. M. Emmerich, and C. A. Coello Coello. On the construction of pareto-compliant combined indicators. *Evolutionary Computation*, 30(3):381–408, 09 2022.

[27] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[28] D. Hardin and E. Saff. Discretizing Manifolds via Minimum Energy Points. *Notices of the American Mathematical Society*, 51(10):1186–1194, 2004.

[29] Jesús Guillermo Falcón-Cardona, Hisao Ishibuchi, and Carlos A. Coello Coello. Riesz s-energy-based reference sets for multi-objective optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.

[30] Michael Pilegaard Hansen and Andrzej Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Technical University of Denmark, March 1998.

[31] Eckart Zitzler, Joshua Knowles, and Lothar Thiele. Quality Assessment of Pareto Set Approximations. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors, *Multiobjective Optimization. Interactive and Evolutionary Approaches*, pages 373–404. Springer. Lecture Notes in Computer Science Vol. 5252, Berlin, Germany, 2008.

[32] Carlos A. Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In Raúl Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, and Humberto Sossa, editors, *MICAI 2004: Advances in Artificial Intelligence*, pages 688–697, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[33] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, pages 110–125. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal, March 29 - April 1 2015.

[34] C.M. Fonseca and P.J. Fleming. Multiobjective Genetic Algorithms. In *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, pages 6/1–6/5. IEE, 1993.

[35] N. Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[36] E.F. Khor, K.C. Tan, T.H. Lee, and C.K. Goh. A study on distribution preservation mechanism in evolutionary multi-objective optimization. *Artificial Intelligence Review*, 23(1):31–56, May 2005.

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

[38] Carlos A. Coello Coello and Gregorio Toscano Pulido. A Micro-Genetic Algorithm for Multiobjective Optimization. In Eckart Zitzler, Kalyanmoy Deb,

Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 126–140. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

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

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

[41] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001.

[42] Raquel Hernández Gómez, Carlos A. Coello Coello, and Enrique Alba Torres. A Multi-Objective Evolutionary Algorithm based on Parallel Coordinates. In *2016 Genetic and Evolutionary Computation Conference (GECCO'2016)*, pages 565–572, Denver, Colorado, USA, 20-24 July 2016. ACM Press. ISBN 978-1-4503-4206-3.

[43] Linlin Wang and Yunfang Chen. Diversity based on entropy: A novel evaluation criterion in multi-objective optimization algorithm. *International Journal of Intelligent Systems and Applications*, 4:113–124, 2012.

[44] A. Farhang-Mehr and S. Azarm. Diversity assessment of pareto optimal solution sets: an entropy approach. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 1, pages 723–728, 2002.

[45] John David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, Tennessee, USA, 1984.

[46] Javier Del Ser, Eneko Osaba, Daniel Molina, Xin-She Yang, Sancho Salcedo-Sanz, David Camacho, Swagatam Das, Ponnuthurai N. Suganthan, Carlos A. Coello Coello, and Francisco Herrera. Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, 48:220–250, 2019.

[47] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 2424–2431, Hong Kong, June 2008. IEEE Service Center.

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

[49] Jesús Guillermo Falcón-Cardona and Carlos A. Coello Coello. Indicator-based Multi-Objective Evolutionary Algorithms: A Comprehensive Survey. *ACM Computing Surveys*, 53(2), March 2020. Article No. 29.

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

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

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

[53] Anupam Trivedi, Dipti Srinivasan, Krishnendu Sanyal, and Abhiroop Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462, 2017.

[54] Andrzej Jaszkiewicz. On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem-A Comparative Experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, August 2002.

[55] Jingda Deng and Qingfu Zhang. Approximating hypervolume and hypervolume contributions using polar coordinate. *IEEE Transactions on Evolutionary Computation*, 23(5):913–918, 2019.

[56] Johannes Bader, Kalyanmoy Deb, and Eckart Zitzler. Faster hypervolume-based search using monte carlo sampling. In Matthias Ehrgott, Boris Naujoks, Theodor J. Stewart, and Jyrki Wallenius, editors, *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, pages 313–326, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[57] Rainer E. Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment Problems, Revised Reprint.* Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2012.

[58] R. Balakrishnan and K. Ranganathan. *A Textbook of Graph Theory.* Universitext. Springer New York, NY, 2012.

[59] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

[60] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[61] Edgar Manoatl Lopez and Carlos A. Coello Coello. IGD$^+$-EMOA: A Multi-Objective Evolutionary Algorithm based on IGD$^+$. In *2016 IEEE Congress on Evolutionary Computation (CEC'2016)*, pages 999–1006, Vancouver, Canada, 24-29 July 2016. IEEE Press. ISBN 978-1-5090-0623-9.

[62] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.

[63] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[64] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[65] Luis Miguel Antonio and Carlos A. Coello Coello. Particle Swarm Optimization Based on Linear Assignment Problem Transformations. In *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 57–64, Madrid, Spain, July 11-15 2015. ACM Press. ISBN 978-1-4503-3472-3.

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

[67] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A Scalable Multi-objective Test Problem Toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 280–295, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[68] Yanan Sun, Gary G. Yen, and Zhang Yi. Global view-based selection mechanism for many-objective evolutionary algorithms. In *2017 IEEE Congress on Evolutionary Computation (CEC'2017)*, pages 427–434, 2017.

[69] Shengxiang Yang, Miqing Li, Xiaohui Liu, and Jinhua Zheng. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013.

[70] Luis Miguel Antonio, Jose A. Molinet Berenguer, and Carlos A. Coello Coello. Evolutionary Many-Objective Optimization Based on Linear Assignment Problem Transformations. *Soft Computing*, 22(16):5491–5512, August 2018.

[71] Yuan Yuan, Hua Xu, Bo Wang, and Xin Yao. A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37, February 2016.

[72] Shahin Rostami and Ferrante Neri. Covariance Matrix Adaptation Pareto Archived Evolution Strategy with Hypervolume-Sorted Adaptive Grid Algorithm. *Integrated Computer-Aided Engineering*, 23(4):313–329, 2016.

[73] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[74] R. Mallipeddi and P. N. Suganthan. Ensemble of constraint handling techniques. *IEEE Transactions on Evolutionary Computation*, 14(4):561–579, 2010.

[75] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima. Simultaneous Use of Different Scalarizing Functions in MOEA/D. In *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation (GECCO'2010)*, pages 519–526, Portland, Oregon, USA, July 7-11 2010. ACM Press. ISBN 978-1-4503-0072-8.

[76] R. Hernández Gómez and C. A. Coello Coello. A Hyper-Heuristic of Scalarizing Functions. In *2017 Genetic and Evolutionary Computation Conference (GECCO'2017)*, pages 577–584, Berlin, Germany, July 15-19 2017. ACM Press. ISBN 978-1-4503-4920-8.

[77] Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190, April 2017.

[78] Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong. An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, October 2015.

[79] D. Brockhoff, T. Wagner, and H. Trautmann. On the Properties of the R2 Indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, pages 465–472, New York, NY, USA, 2012. Association for Computing Machinery.

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