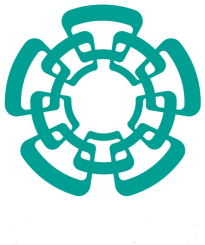


Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Cinvestav Unidad Tamaulipas

Uso de formulaciones para resolver problemas de optimización con muchos objetivos

Tesis que presenta:

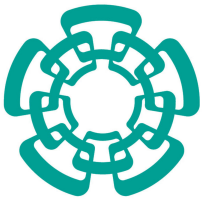
Auraham Sinhué Camacho García

Para obtener el grado de:

**Doctor en Ciencias
en Ingeniería y Tecnologías
Computacionales**

Dr. Ricardo Landa Becerra, Co-Director
Dr. Gregorio Toscano-Pulido, Co-Director

Cd. Victoria, Tamaulipas, México Febrero, 2020



Center for Research and Advanced Studies
of the National Polytechnic Institute

Cinvestav Tamaulipas

On the Use of Scalarizing Functions to Solve Many-Objective Optimization Problems

Thesis by:

Auraham Sinhué Camacho García

as the fulfillment of the
requirement for the degree of:

**Doctor of Science
in Engineering and Computing
Technologies**

Dr. Ricardo Landa Becerra, Co-Director
Dr. Gregorio Toscano-Pulido, Co-Director

Cd. Victoria, Tamaulipas, México February, 2020

The thesis of Auraham Sinhué Camacho García is approved by:

_____

_____
Dr. César Torres Huitzil

_____
Dr. Carlos Alberto Lara Álvarez

_____
Dr. Iván López Arévalo

_____
Dr. Edwyn Javier Aldana Bobadilla

_____
Dr. Ricardo Landa Becerra, Committte Co-chair

_____
Dr. Gregorio Toscano-Pulido, Committte Co-chair

Cd. Victoria, Tamaulipas, México, February 28 2020

To my family.

Contents

List of Figures

List of Tables

List of Algorithms

Resumen

Uso de formulaciones para resolver problemas de optimización con muchos objetivos

por

Auraham Sinhué Camacho García
Cinvestav Unidad Tamaulipas
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2020
Dr. Gregorio Toscano-Pulido, Co-Director
Dr. Ricardo Landa Becerra, Co-Director

Los problemas de optimización multiobjetivo (*multiobjective optimization problems*, MOPs) se encuentran en muchas disciplinas y áreas de diseño. Un MOP involucra múltiples funciones objetivo que deben ser mejoradas de manera simultánea. Generalmente, estos objetivos están en conflicto, lo que significa que una sola solución no puede mejorar todos los objetivos a la vez. Los problemas con más de tres objetivos son llamados problemas de optimización con muchos objetivos (*many-objective optimization problems*, MaOPs).

El término optimización evolutiva multiobjetivo (*evolutionary multiobjective optimization*, EMO) se refiere al uso de algoritmos evolutivos (*evolutionary algorithms*, EAs) para abordar problemas con múltiples objetivos. Muchos algoritmos evolutivos multiobjetivo (*multiobjective evolutionary algorithms*, MOEAs) han sido propuestos en la literatura. Entre ellos, los MOEAs basados en descomposición han mostrado ser computacionalmente eficientes para resolver MOPs. Este tipo de algoritmos transforma un MOP en múltiples subproblemas escalares que son resueltos de manera colaborativa. Esta transformación requiere una función de escalarización y un conjunto de vectores de referencia uniformemente distribuidos. Estos parámetros juegan un papel importante al descomponer un problema. Si no son sintonizados adecuadamente, los MOEAs basados en descomposición pueden presentar dos limitaciones, es decir, una pobre capacidad de búsqueda y deterioro en la diversidad. Para abordar estas limitaciones, requerimos de un método para determinar una configuración adecuada durante el proceso de búsqueda.

En este documento, proponemos dos enfoques para mejorar la descomposición en problemas con muchos objetivos. Primero, presentamos un método para mejorar la capacidad de búsqueda de algoritmos basados en descomposición, llamado múltiple selección de supervivientes (*Multiple Survival Selection, MSS*). Este método evalúa múltiples funciones de escalarización y selecciona la mejor de ellas con respecto a un indicador de desempeño. El desempeño de una función de escalarización depende de su habilidad para encontrar buenas soluciones en la selección de supervivientes. Dado que múltiples funciones de escalarización están involucradas, el método propuesto puede explorar múltiples conjuntos de soluciones en la selección de supervivientes. En el método propuesto, la evaluación de desempeño es llevado a cabo frecuentemente, esto significa que función adecuadas de escalarización son promovidas durante el proceso de búsqueda. Segundo, presentamos un indicador llamado mediana de la dispersión de la población (*median of dispersion of the population*, MDP) para asistir a los métodos de adaptación de vectores. En la literatura, estos métodos cambian las ubicaciones de los vectores de referencia para mejorar la diversidad de la población al descomponer un problema. Tal adaptación es realizada frecuentemente durante la búsqueda. Sin embargo, realizar adaptaciones tempranas puede ser poco efectivo, lo que conlleva a un esfuerzo computacional innecesario. El indicador propuesto es empleado para determinar el momento en que la población se estanca al medir su progreso hacia el frente de Pareto. En el método propuesto, la adaptación de vectores de referencia se realiza sólo después de alcanzar el punto de estancamiento, lo que significa que se previene la adaptación temprana. De este modo, podemos llevar a cabo una adaptación más efectiva. Finalmente, integramos ambos métodos en un único algoritmo.

Ambos métodos propuestos están diseñados para abordar las dos limitaciones mencionadas anteriormente de los algoritmos basados en descomposición. Nuestros resultados muestran que MSS es un método prometedor para mejorar la convergencia de la población, mientras que MDP es una forma adecuada para mejorar la adaptación de vectores.

Abstract

On the Use of Scalarizing Functions to Solve Many-Objective Optimization Problems

by

Auraham Sinhué Camacho García
Cinvestav Tamaulipas
Research Center for Advanced Study from the National Polytechnic Institute, 2020
Dr. Gregorio Toscano-Pulido, Co-advisor
Dr. Ricardo Landa Becerra, Co-advisor

Multiobjective optimization problems (MOPs) arise in many disciplines and design areas. An MOP involves several objective functions that need to be optimized simultaneously. Generally, these objectives are in conflict with each other, meaning that a single solution cannot improve all the objectives simultaneously. Problems with more than three objective functions are known as many-objective optimization problems (MaOPs).

The term evolutionary multiobjective optimization (EMO) refers to the use of evolutionary algorithms (EAs) for addressing problems with multiple optimization criteria. Several multiobjective evolutionary algorithms (MOEAs) have been proposed in the literature. Among them, decomposition-based MOEAs have shown to be computationally efficient for solving MaOPs. A decomposition-based algorithm transforms an MOP into several scalar subproblems that are then solved in a collaborative way. This transformation requires a scalarizing function and a set of evenly distributed reference vectors. These parameters play a crucial role when decomposing a problem. If they are not properly tuned, decomposition-based MOEAs may face two limitations, that is, poor search ability and diversity deterioration. In order to address these issues, we require a method for determining a suitable parameter setting throughout the search process.

In this document, we propose two approaches to enhance decomposition in many-objective optimization. First, we introduce a method for enhancing the search ability of decomposition-based algorithms, called Multiple Survival Selection (MSS). This method evaluates several scalarizing

functions and selects the best of them regarding a performance indicator. The performance of a scalarizing function depends on its ability to find a good solution set at survival selection. Since we are dealing with multiple scalarizing functions, the proposed method explores several solution sets for survival selection. In the proposed method, the performance assessment is conducted frequently, meaning that suitable scalarizing functions are promoted throughout the search process. Second, we introduce an indicator called Median of Dispersion of the Population (MDP) to assist weight adaptation methods. In the literature, these methods change the locations of the reference vectors to improve the diversity of the population when decomposing a problem. Such adaptation is performed frequently throughout the search. Although, performing early adaptations may be ineffective, leading to unnecessary computational burden. The proposed indicator is employed to determine whether the population is stagnant by measuring its progress toward the Pareto front. In the proposed method, the adaptation of reference vectors is performed only after reaching the stagnation point, meaning that early adaptation is avoided. This way, we can conduct a more effective adaptation. Finally, we integrate both methods into a single framework.

Both proposed methods are aimed to address the two previously mentioned limitations of decomposition-based algorithms. Our experimental results show that MSS is a promising direction for improving the convergence of the population, whereas MDP is a suitable way for enhancing the adaptation of reference vectors.

Nomenclature

DTLZ	Deb-Thiele-Leumman-Zitzler test suite
EA	Evolutionary Algorithm
EC	Evolutionary Computation
ES	Evolutionary Strategy
EP	Evolutionary Programming
DE	Differential Evolution
DIM	Dynamic Island Model
GA	Genetic Algorithm
MOEA	Multi-objective Evolutionary Algorithm
MOP	Multi-objective Optimization Problem
NSGA-II	Non-dominated Sorting Genetic Algorithm
OP	Operation Research
RL	Reinforcement Learning
ZDT	Zitzler-Deb-Thiele test suite

1 Introduction

1.1 Motivation

Multiobjective optimization problems (MOPs) involve m objective functions to be optimized simultaneously. This type of problems arises naturally in many real-world applications. Vehicle refueling planning [53], design of distributed local area networks [42], and job shop scheduling [80] are some examples. The objectives are often conflicting, meaning that a single solution cannot improve all the objectives simultaneously. Instead, a set of solutions that represents the best trade-offs among the objectives is needed. Such a set is called the Pareto set in the decision space and the Pareto front in the objective space [9]. A multiobjective optimizer is employed with the aim of finding an approximation to the Pareto front. After finishing the search process, a decision maker can select one solution according to high-level information or preferences. When the number of objectives is greater than three, these problems are referred to as many-objective optimization problems (MaOPs) [34].

Evolutionary algorithms are search methods based in natural evolution to address optimization problems [19]. Given their population-based nature, they can provide a set of trade-off solutions in a single run. Several multiobjective evolutionary algorithms (MOEAs) have been proposed in the literature [15, 83, 81, 24]. They are typically classified as follows, according to the way solutions (individuals) are chosen for survival selection: Pareto dominance, indicators, relaxed dominance, and decomposition. Among these categories, decomposition has attracted special attention in the evolutionary multiobjective optimization (EMO) community, given its efficiency to handle MaOPs. In this category, MOEA/D [83] is the most representative decomposition-based approach in the literature.

Decomposition is a paradigm for transforming an MOP into multiple scalar subproblems. In order to decompose a problem, we require a scalarizing function and a set of reference vectors. The reference vectors can be considered as search directions outlined in the objective space, whereas the scalarizing function give us way for comparing solutions. Each reference vector represent a subproblem. Promising solutions are found by optimizing the subproblems in a collaborative way, that is, by sharing good solutions among them, under the assumption that a good solution for subproblem i could be also useful for solving a neighboring subproblem j. A decomposition-based method drives solutions toward the Pareto front along the directions defined by the reference vectors and the scalarizing function. Figure 1.1 illustrates the decomposition of a two-objective problem. The reference vectors are shown in Fig. 1.1 (a). Observe that the reference vectors are uniformly distributed in the objective space with the aim of improving the coverage of the Pareto front. Figures (b) and (c) show the contour maps of two scalarizing functions, weighted sum (WS) and penalty-based boundary intersection (PBI), with respect a single reference vector. These contours determine how to compare solutions regarding a specific reference vector.

Both parameters, the scalarizing function and the set of reference vectors, play a crucial role when decomposing a problem. According to previous works [33, 52, 9, 70], decomposition-based algorithms may face two limitations if these parameters are not tuned properly:

- Poor search ability. If the scalarizing function does not help the algorithm to drive the population toward the Pareto front, then the convergence will be deteriorated.

- Diversity deterioration. If the reference vectors does not reflect the shape of the Pareto front, then the population will be unable to cover the Pareto front uniformly.

These issues have been studied in the literature. However, as far as we know, little effort has been made to address them in a single framework. On the one hand, some authors have proposed to use a pool of candidate scalarizing functions and use them in a collaborative way [30, 32, 24]. However, some of these approaches are not scalable with respect to the number of scalarizing functions. In addition, the performance of these methods may be deteriorated if most of the candidate functions are unsuitable for solving the problem. On the other hand, some authors have developed weight adaptation methods[1] for improving the diversity of the population [52, 37, 9, 70]. These methods change the locations of the reference vectors in order to get a better coverage of the Pareto front. To this end, the population is employed as a reference of the Pareto front for adapting the reference vectors. The adaptation is often more effective when the population is close to the Pareto front. However, it is unlikely that the population may provide a good approximation of the Pareto front shape at early generations. Thus, early weigt adaptation could be ineffective.

Tuning a decomposition-based algorithm is not straightforward since its parameters depend on the properties of the problem, often unknown beforehand. For instance, a scalarizing function may be useful for addressing a specific type of problems, but unhelpful in other cases. In the same way, a given set of reference vectors may be appropriate for solving problems with a regular Pareto front shape. However, if it is employed for decomposing an irregular problem, then the coverage of the Pareto front will be deteriorated. In order to improve the performance of decomposition-based algorithms, both parameters might be analyzed to determine an appropriate configuration.

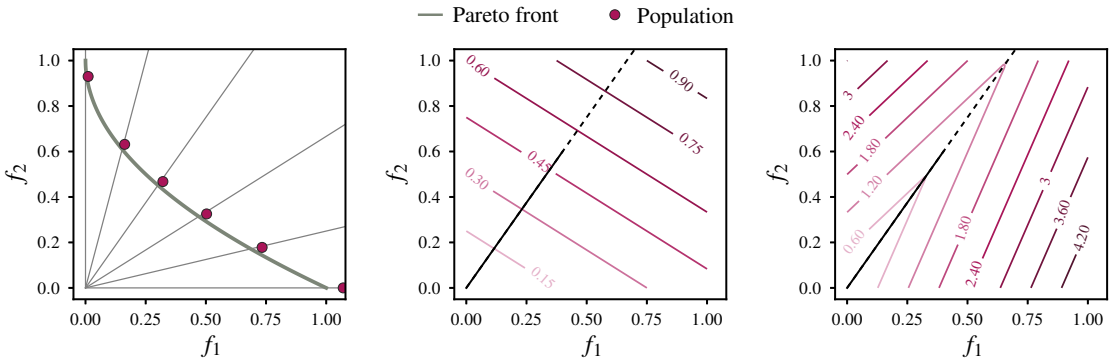
In this document, we propose two approaches to enhance decomposition in many-objective

[1] The terms *reference vector* and *weight vector* are used interchangeably in this document.

optimization. First, we introduce a method for improving the search ability of decomposition-based algorithms, called multiple survival selection (MSS). The proposed method evaluates the effectiveness of several candidate scalarizing functions and promotes the best candidates throughout the search process. To this end, we evaluate the behavior of each candidate regarding its ability to find promising solutions at the survival selection phase. After exploring several performance indicators, we adopted the diversity comparison indicator (DCI) [50] for measuring the effectiveness of the scalarizing functions. As a result, we can explore several decomposition schemes, one for each candidate function, and select the best one with respect to the performance indicator. Second, we introduce an indicator called Median of Dispersion of the Population (MDP) to assist weight adaptation methods. The proposed indicator provides a general snapshot of the overall progress of the population toward the Pareto front by analyzing the local progress of each subproblem. When the subproblems cannot be significantly improved regarding the proposed indicator, the population becomes stagnant. At this point, it may provide a better approximation of the Pareto front shape. After reaching the stagnation point, the adaptation of reference vectors is performed. This way, early adaptation can be avoided and the effectiveness of the adaptation can be improved. Finally, we integrate both methods, MSS and MDP, into a single framework to improve the performance of decomposition-based algorithms. The resulting approach is able to determine a suitable scalarizing function and adapt the reference vectors throughout the search process.

1.2 Problem Statement and Proposal

In the context of multiobjective optimization, we want to determine whether the use of $l \geq 2$ scalarizing functions, $\{g^1, g^2, \ldots, g^l\}$, improves the behavior of an decomposition-based MOEA according to a performance indicator from the literature. This problem involves two aspects: how to measure the effectiveness of g^k to guide the approach toward the Pareto front, and how to handle l functions without causing a significant overhead to the MOEA. The proposed framework comprises



(a) Reference vectors (b) WS (c) PBI

Figure 1.1: Intuition of decomposition. (a) Set of reference vectors employed to decompose the problem. Contour maps of two scalarizing functions: (b) weighted sum and (c) penalty-based boundary intersection. A decomposition-based algorithm drives solutions toward the Pareto front along the directions defined by the reference vectors and the scalarizing function.

the following elements:

- Multiple survival selection (MSS). The proposed solution handles l auxiliary populations, one for each scalarizing approach, where P^k keeps the progress made by g^k, for $k = 1, 2, \ldots, l$.
- Contribution indicator. In order to identify the most suitable approach from G for a particular problem, we require an indicator to measure the performance of g^k during the search process.
- Adaptation of reference vectors. We have also developed a weight adaptation mechanism to enhance the behavior of the proposed method in problems with irregular Pareto front shapes.

1.3 Hypothesis

Given $l \geq 2$ scalarizing functions, we assume that one of them would be appropriate for addressing a specific type of MOP. If so, an MOEA coupled with l scalarizing functions would be able to deal with a wider range of MOPs by identifying the most suitable of them for the problem at hand. Selecting a suitable scalarizing function may improve the convergence of a decomposition-based

method. However, in order to enhance its diversity, an efficient weight adaptation mechanism should be considered as well. Given this, we state the following hypothesis:

The collaborative use of l scalarizing functions and the efficient adaptation of reference vectors can improve the performance of decomposition-based MOEAs for addresing MOPs with regular and irregular Pareto front shapes with respect to a performance indicator.

1.4 Objectives

1.4.1 General Objective

To develop a MOEA able to use a set of scalarizing functions in order to effectively address MaOPs, and to be competitive with respect to state-of-the-art proposals.

1.4.2 Specific Objectives

1. To conduct a preliminary study to evaluate several scalarizing functions and determine which of them are suitable to address a particular type of MOP.

2. To evaluate the use of distributed evolutionary algorithms for the use of a set of scalarizing functions.

3. To develop a procedure to measure the effectiveness of a scalarizing function when solving an MOP.

1.5 State of the Art

A lot of effort has been made to enhance decomposition-based MOEAs. In this section, we review some approaches developed in recent years according to four areas (detailed in Table 1.1):

1. Use of $l \geq 2$ scalarizing functions.

2. Development of new scalarizing functions.

3. Adaptation of reference vectors.

4. Incorporation of Pareto dominance.

One of the early approaches to use more than a single scalarizing approach was introduced by Ishibuchi et al. [32]. The authors considered to employ both weighted sum (WS) and Tchebycheff (TE) along with MOEA/D to solve combinatorial multiobjective problems. They proposed two approaches, called Single-Grid and Multi-Grid schemes. The Single-Grid approach divides a set of N reference vectors among WS and TE by assigning a different scalarizing function alternately to each reference vector. On the other hand, in the Multi-Grid scheme, each scalarizing function has its own set of N reference vectors. According to their results, both schemes outperformed the single use of WS and TE. This observation shows the effectiveness of combining several scalarizing functions in a single MOEA. From both schemes, the Multi-Grid approach obtained the best performance. Nonetheless, the main limitation of this approach is that the population size depends on the number of scalarizing functions and reference vectors. Thus, the population size will grow with respect to the considered scalarizing functions.

Hernández and Coello [24] introduced a method to handle l scalarizing functions simultaneously, called MOMBI-III. Their approach assigns two attributes to each solution in the population: a rank and a contribution value. The rank is defined according to the number of subproblems where a solution $\boldsymbol{x}$ is the best. The lower the value of the rank, the largest the number of subproblems where $\boldsymbol{x}$ is the best solution. Lower ranks are preferred. Parent and offspring populations are arranged into fronts according to their ranks. Then, the best N solutions are preserved. To handle rank ties, the contribution value is employed. This attributed is based in a metric called s-energy [21], and is aimed to measure the contribution of a solution $\boldsymbol{x}$ to the diversity of the population. Lower

Scope	Reference	Proposed method	Description
Use of $l \geq 2$ scalarizing functions	[32] (2010)	Multi-Grid decomposition	This method combines WS and TE in a single algorithm. A set of N reference vectors is defined for each scalarizing function. Although, the population size will grow with respect to the number of considered scalarizing functions.
	[24] (2017)	MOMBI-III	A pool of l scalarizing functions is defined. The rank is determined according to the number of subproblems where a specific solution is the best. Multiple scalarizing functions are involved in the ranking procedure. The individuals with the best (lowest) ranks are preserved. To handle ties, the approach measures the contribution of a solution to the diversity of the population.
Development of scalarizing functions	[57] (2012)	Modification of ASF	This function involves two reference vectors, one for achievable reference points, and the other for unachievable reference points. This approach is designed so that an appropriate reference vector is used in each case.
	[62] (2012)	Modification of ASF	A parameterized version of ASF is proposed. An integer parameter is introduced to control the degree of metric flexibility.
	[73] (2016)	Constrained Decomposition	The improvement region of subproblem i is controlled by adding a constraint defined by an angle θ_i. This way, the balance between diversity and converge can be adjusted.
	[39] (2018)	MSF and PSF	The Multiplicative Scalarizing Function (MSF), and Penalty-based Scalarizing Function (PSF) are introduced. The improvement region is controlled by a new parameter. A strategy to adjust such a parameter is also discussed.
	[76] (2018)	Localized WS	The improvement region of WS is controlled by an hypercone to enhance its performance in non-convex problems.
Adaptation of reference vectors	[37] (2014)	A-NSGA-III	A density estimation procedure is performed to identify unhelpful and overcrowded reference vectors. Additional vectors are generated around overcrowded vectors. Then, the density estimation procedure is performed one more time to remove unhelpful vectors.
	[9] (2016)	RVEA	The reference vectors are adapted according to the scales of the objective functions.
	[52] (2017)	AdaW	An archive of non-dominated solutions is employed to reflect the shape of the Pareto front and create new reference vectors in unexplored regions.
	[70] (2017)	AR-MOEA	The reference vectors are adapted from an archive of solutions. Then, a performance indicator, called IGD-NS, is employed for survival selection along with the adapted vectors.
Incorporation of Pareto dominance	[41] (2014)	Decomposition + Pareto local search	After decomposition, a local search method, called Pareto local search (PLS), is applied to explore the neighborhood of each solution. An additional population is maintained to perform local search.
	[51] (2016)	BCE	A Bi-Criterion Evolution (BCE) framework is proposed. This method combines Pareto dominance and decomposition to select individuals.
	[67] (2017)	Parallel Pareto local search	This method divides the objective space into L subregions and alternates between decomposition and Pareto dominance to maintain an archive of solutions.

Table 1.1: Recent work to enhance decomposition-based MOEAs.

values for contribution are preferred. The authors evaluated their approach with respect to $l = 8$ scalarizing functions. From their results, they showed that MOMBI-III outperformed NSGA-III and MOEA/D. However, they did not consider the case where multiple scalarizing functions are not useful

for addressing the problem. This could impact on the ranking of solutions.

As previously mentioned, other research direction to enhance decomposition is the development of new scalarizing approaches. Previous work have been focused in modifying existing approaches of WS [76], achievement scalarizing function (ASF) [62, 57], and TE [58]. In [73], Wang et al. proposed a methodology to control the improvement region of a scalarizing function. The improvement region of $\boldsymbol{x}$ for subproblem i in the objective space can be defined as the set:

$$A = \{\boldsymbol{f}(\boldsymbol{y}) : g(\boldsymbol{y}, \boldsymbol{w}_i, \boldsymbol{z}^*) < g(\boldsymbol{x}, \boldsymbol{w}_i, \boldsymbol{z}^*)\} \tag{1.1}$$

According to the authors, the improvement regions of scalarizing approaches, such as PBI and WS, may be too large for some problems. Consequently, a single solution can replace several inferior solutions, reducing the diversity of the population [73]. The authors proposed to control the extension of the region to be improved by adding a constraint to the scalarizing function. This way, the balance between the population diversity and the convergence can be adjusted. Given a subproblem g_i, the constrained subproblem is defined as follows:

$$\begin{aligned} &\text{Minimize} \quad && g_i(\boldsymbol{x}, \boldsymbol{w}_i, \boldsymbol{z}^*) \\ &\text{Subject to} \quad && \langle \boldsymbol{w}_i, \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^* \rangle \leq 0.5\theta_i \end{aligned} \tag{1.2}$$

where $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ denotes the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$, $\boldsymbol{w} = [w_1, \ldots, w_m]^T$ is a reference vector, and $\boldsymbol{z}^* = [z_1^*, \ldots, z_m^*]^T$ is a reference point, where z_i^* is set to be a value slightly smaller than the smallest f_i value found so far, for $i = 1, \ldots, m$. The extension of the improvement region of g_i is adjusted by the angle θ_i. Finally, the authors also proposed a method for handling constraints and an adaptive strategy to control the value of θ_i during the search process.

A recent trend to improve decomposition is the adaptation of reference vectors, as illustrated in Fig. 1.2. Some vectors are not useful because they do not reach the Pareto front. A weight adaptation procedure could identify such vectors and translate them to achieve a better approximation to the

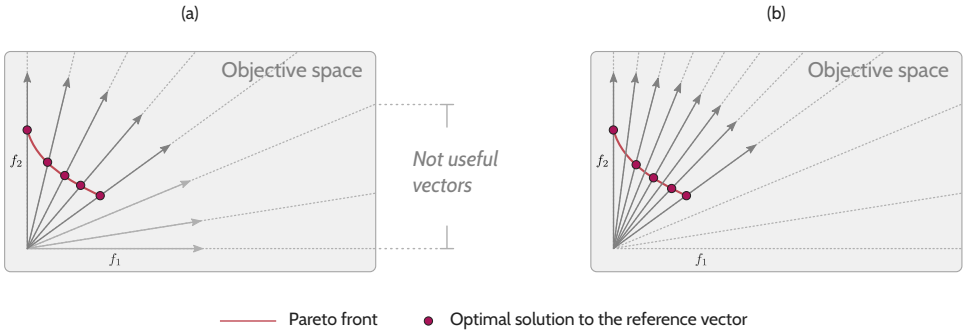


Figure 1.2: Adaptation of reference vectors. (a) Some vectors does not help to approximate the shape of the Pareto front. (b) They can be translated to obtain a better coverage of the Pareto front.

Pareto front. Li and Yao [52] proposed a method called AdaW composed by five operations: weight generation, weight addition, weight deletion, archive maintenance, and weight update frequency. Their approach maintains an archive of non-dominated solutions employed to find unexplored regions in the objective space. Additional vectors are created to cover such regions. Then, overcrowded vectors are removed to keep a fixed number of vectors. This procedure is performed regularly during the evolutionary process according to a given interval. From their experiments, AdaW outperformed some weight adaptation approaches: A-NSGA-III [37], RVEA [9], and MOEA/D-AWA [63]. Although, the authors also concluded that AdaW have some limitations, such as the overhead related to weight adaptation and the number of additional parameters.

Some researches have proposed to improve decomposition by incorporating other techniques, such as Pareto dominance. Shi and Zhang [67] introduced the Parallel Pareto Local Search based on Decomposition (PPLS/D). Their approach tackles a problem by means of decomposition along the first stages of the search process, and Pareto dominance at the end. First, the objective space is divided into L subregions according to a set of reference vectors, $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_L$. Then, a scalar optimization is conducted within each subregion to approach the Pareto front quickly. After reaching a stopping criteria, the first stage ends and the proposed process switches to Pareto dominance to find non-dominated solutions on each subregion. This method would be enhanced by introducing a performance indicator to determine whether the scalar optimization cannot be further improved,

finish the first phase, and switch to find non-dominated solutions in the second phase.

1.6 Structure of the document

The rest of the document is structured as follows:

- Chapter 2 introduces the necessary concepts to have a self-contained document.
- Chapter 3 provides a detailed description of the difficulties of decomposition-based MOEAs derived from the literature and our own experience.
- Chapter 4 introduces our proposal for the collaborative use of a set of scalarizing functions.
- Chapter 5 describes our weight adaptation proposal to address problems with irregular Pareto fronts.
- Chapter 6 describes the integration of MSS and MDP in a single framework.
- Chapter 7 presents the conclusions and future directions.

2

Background

2.1 Multiobjective optimization

Optimization refers to finding one or more feasible solutions which correspond to extreme values of one or more objectives. When an optimization problem that models a physical system involves only one objective function, the task of finding the optimal solution is called single-objective optimization [14]. When an optimization problem involves several objective functions to be improved simultaneously, it is called multiobjective optimization problem (MOP). Without loss of generality, an MOP can be stated as a minimization problem as follows [59, 56, 86]:

$$\begin{aligned} &\text{Minimize } \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T \\ &\text{subject to } \boldsymbol{x} \in X \end{aligned} \tag{2.1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the i-th objective function, for $i = 1, 2, \ldots, m$. Here, m is the number of objectives and T denotes transposition. The decision vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ comprises n

decision variables and belongs to the decision space $X \subseteq \mathbb{R}^n$. The objective vector $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$ is the image of $\boldsymbol{x}$ in the objective space $Z \subseteq \mathbb{R}^m$. Problems with more than three objectives are called many-objective optimization problems (MaOPs) [34, 48]. Often, the objectives are conflicting. That is, when one objective is improved, one or more distinct objectives are deteriorated. This conflict implies that a single solution cannot improve all the objectives simultaneously. Instead, a multiobjective optimizer is used with the aim of finding a set of trade-off solutions. Each solution in this set represents a different balance among the objectives. This set of solutions is called the Pareto set (PS) in the decision space and the Pareto front (PF) in the objective space [9]. After finding a diverse set of solutions, a decision maker selects one or a few of them according to high-level information or a particular preference. MOPs with degenerated, disconnected, or inverted Pareto front shapes are considered irregular. Some optimizers could face difficulties when addressing these characteristics.

In single-objective optimization, the relation $\leq$ (minimization) or $\geq$ (maximization) is employed to compare a pair of solutions. In multiobjective optimization, the Pareto dominance relation is usually employed to this end. Below, we introduce some concepts considering a minimization problem [56, 48, 87].

Definition 1 (Pareto dominance relation)**.** Given two solutions $\boldsymbol{x}$, $\boldsymbol{y} \in X$, $\boldsymbol{x}$ is said to (Pareto) dominate $\boldsymbol{y}$, denoted as $\boldsymbol{x} \prec \boldsymbol{y}$, if and only if:

$$\begin{aligned} f_i(\boldsymbol{x}) &\leq f_i(\boldsymbol{y}), \quad \text{for } \forall i \in \{1, 2, \ldots, m\} \\ f_j(\boldsymbol{x}) &< f_j(\boldsymbol{y}), \quad \text{for } \exists j \in \{1, 2, \ldots, m\} \end{aligned} \tag{2.2}$$

When Eq. (2.2) holds, we say that $\boldsymbol{x}$ is better than $\boldsymbol{y}$, or equivalently, $\boldsymbol{y}$ is dominated by $\boldsymbol{x}$ [31]. A solution $\boldsymbol{x}$ is said to be *non-dominated* with respect to a set of solutions S if there is no solution in S that dominates $\boldsymbol{x}$ [45].

Definition 2 (Weak Pareto dominance relation)**.** Given two solutions $\boldsymbol{x}$, $\boldsymbol{y} \in X$, $\boldsymbol{x}$ is said to weakly

(Pareto) dominate $\boldsymbol{y}$, denoted as $\boldsymbol{x} \preceq \boldsymbol{y}$, if and only if:

$$f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y}) \quad \text{for} \quad \forall i \in \{1, 2, \ldots, m\} \tag{2.3}$$

Definition 3 (Pareto optimality)**.** A solution $\boldsymbol{x} \in X$ is Pareto optimal if there does not exist another solution $\boldsymbol{y} \in X$ such that $\boldsymbol{y} \prec \boldsymbol{x}$.

Definition 4 (Pareto set)**.** The Pareto set is the union of the Pareto optimal solutions:

$$PS = \{\boldsymbol{x} \in X : \nexists \boldsymbol{y} \in X \text{ such that } \boldsymbol{y} \prec \boldsymbol{x}\} \tag{2.4}$$

Definition 5 (Pareto front)**.** Given the Pareto set, the Pareto front is defined as:

$$PF = \{\boldsymbol{f}(\boldsymbol{x}) : \boldsymbol{x} \in PS\} \tag{2.5}$$

Definition 6 (Ideal and Nadir points)**.** The ideal and Nadir points comprise the lower and upper bounds of the objective function values of the Pareto set, respectively. The ideal point $\boldsymbol{z}^* = [z_1^*, z_2^*, \ldots, z_m^*]^T$ is defined as:

$$z_i^* = \min_{\boldsymbol{x} \in PS} \{f_i(\boldsymbol{x})\} \tag{2.6}$$

whereas the Nadir point $\boldsymbol{z}^{nad} = [z_1^{nad}, z_2^{nad}, \ldots, z_m^{nad}]^T$ is defined as:

$$z_i^{nad} = \max_{\boldsymbol{x} \in PS} \{f_i(\boldsymbol{x})\} \tag{2.7}$$

It may be computationally expensive to obtain these points. In practice, they are estimated dynamically as the population advances toward the Pareto front.

The goal of an optimizer is to find a good approximation of the entire Pareto front in terms of convergence, diversity, and spread. The Pareto dominance relation is employed to compare a pair of solutions. However, it can also be extended to compare a pair of approximations A, $B \in X$ as

follows [44, 12]:

- A dominates B, denoted as $A \prec B$, if every $\boldsymbol{y} \in B$ is dominated by at least one $\boldsymbol{x} \in A$.
- A weakly dominates B, denoted as $A \preceq B$, if every $\boldsymbol{y} \in B$ is weakly dominated by at least one $\boldsymbol{x} \in A$.

In order to assess the quality of an approximation A with respect to another approximation B, a performance indicator is needed. Several performance indicators have been proposed in the literature for multi-objective optimization [12, 89, 69]. Knowles [44] proposed the use of *Pareto compliant* indicators.

Definition 7 (Pareto compliant)**.** An indicator $I : X \rightarrow \mathbb{R}$ is Pareto compliant if for all $A, B \in X : A \preceq B \Rightarrow I(A) \geq I(B)$, assuming that greater indicator values correspond to higher quality. Otherwise, $A \preceq B \Rightarrow I(A) \leq I(B)$.

The total order of X imposed by an indicator I should not contradict the partial order of X that is imposed by the weak Pareto dominance relation [12]. In other words, given two approximations A, $B \in X$, if A is preferable to B with respect to weak Pareto dominance, the indicator value for A should be at least as good as the indicator value for B. Such indicators are Pareto compliant [12].

2.2 Multiobjective Evolutionary Algorithms

Evolutionary algorithms (EAs) are search methods based on natural evolution designed for solving optimization problems [19]. Given its population-based nature, EAs are suitable to address MOPs, because they can produce an approximation to the entire Pareto front in a single run [12, 74], often simply called Pareto front approximation. A number of multiobjective evolutionary algorithms (MOEAs) for solving MaOPs have been proposed in recent years [15, 81, 49, 9, 75, 2, 78]. These approaches can be classified according to their selection mechanisms into the following categories [34, 48, 81]:

1. *Scalarization*. When an MOP is converted into a single objective optimization problem, it is known as scalarization [60]. Under some conditions, the optimal solutions of multiobjective optimization problems can be characterized as solutions of certain single objective optimization problems [59]. I-MOGLS [29] and J-MOGLS [38] were among the first MOEAs based on scalarization.

2. *Decomposition*. These methods decompose an MOP into a set of subproblems by means of a scalarizing function and a set of reference vectors. Unlike scalarization, decomposition-based MOEAs maintain several search directions simultaneously. Different decomposition-based MOEAs have been proposed [61, 84, 55, 85, 82], being MOEA/D [83] the most representative of them. Generally, a scalarizing function is employed to define each subproblem [83], although some approaches do not require it [55]. There are many active research areas in this category including the development of novel scalarizing approaches [62, 64, 9, 75], efficient allocation of resources among subproblems [85, 84], and hybridization with other methods [49, 15, 2, 82]. A comprehensive survey on decomposition-based MOEAs is given in [1].

3. *Pareto dominance*. These methods are based on Pareto dominance to guide the search process toward the Pareto front. Pareto-based MOEAs are suitable for solving problems with two and three objectives. However, they often face difficulties when the number of objectives increases as it is more likely to have a whole non-dominated population. Given this, the selection pressure is diluted when Pareto dominance is employed, deteriorating the convergence property of Pareto-based MOEAs [34].

4. *Relaxed forms of dominance*. In order to alleviate the difficulties of Pareto-dominance-based methods, some authors have proposed alternative forms of dominance. Thus, the relationship of dominance becomes flexible such that the selection pressure can be enhanced. Different methods have been proposed, including controlling dominance area of solutions (CDAS) [66], grid dominance [79], ϵ-dominance [46], α-dominance [26], and θ-dominance [81].

5. *Indicators*. These methods employ a performance indicator to measure the improvement of the population during the search process. Such an indicator helps the MOEA to guide the population to the Pareto front. Different indicator-based MOEAs have been proposed, such as SMS-EMOA [4], IBEA [88], and HypE [3]. These three methods are based on the hypervolume indicator [89]. However, the main drawback of these methods is the associated cost for computing the indicator, which usually depends on the number of objectives. Other algorithms, such as MOMBI-II [23], $R2$-EMOA [8], $R2$-MODE [18], rely on the $R2$ indicator [7] for driving the search. Compared to the hypervolume indicator, $R2$ is computationally more efficient. Like decomposition-based methods, this indicator requires a set of reference vectors. Hence, search methods based on the $R2$ indicator may inherit the issues related to decomposition, such as performance deterioration on irregular problems [33, 52, 9, 70].

From these categories, the decomposition-based approaches have gained popularity in the evolutionary multiobjective optimization (EMO) community. Despite their benefits, decomposition-based MOEAs face their own challenges [78, 48]:

1. *Number of reference vectors required for MaOPs*. Decomposition-based MOEAs suffer from the curse of dimensionality, that is, to search simultaneously in an exponentially increasing number of directions [48]. This issue is because the size of the set of reference vectors depends on the number of objectives.

2. *Selection of a scalarizing function*. The search ability of decomposition-based MOEAs can be affected by the scalarizing function at hand and its parametrization [27]. Previous works have addressed this issue by using a collection of scalarizing functions [32, 24] during the search process.

3. *Allocation of resources among subproblems*. Usually, after decomposing an MOP, the resulting set of subproblems is optimized in a collaborative manner. Commonly, each subproblem receives

the same amount of computational resources (e.g., function evaluations). However, some parts of the Pareto front can be more difficult to approximate than others [85, 43, 11, 54]. As a result, some subproblems could require a larger amount of resources than others. Given this issue, some researchers have developed techniques for efficient allocation of resources among subproblems [85, 84].

4. *Diversity deterioration*. A uniform set of reference vectors is required to specify multiple search directions toward different regions of the Pareto front [78]. Given the uniform distribution of the reference vectors, the obtained solutions are expected to cover evenly the Pareto front [78]. However, decomposition-based MOEAs may face difficulties in maintaining the diversity of the population, despite the use of a uniform set of reference vectors. Specifically, researchers have found that decomposition-based MOEAs, such as MOEA/D, present difficulties when solving MOPs in problems with irregular shapes of the Pareto front [33].

2.3 Decomposition and Scalarizing Functions

Definition 8 (Scalarizing function)**.** A scalarizing function is a parameterized function $g : \mathbb{R}^m \to \mathbb{R}$. Thus, an MOP is transformed into the following scalar problem:

$$\begin{aligned} &\text{Optimize } g(\boldsymbol{f}(\boldsymbol{x})) \\ &\text{subject to } \boldsymbol{x} \in X \end{aligned} \tag{2.8}$$

In order to decompose an MOP into N scalar subproblems, we require a scalarizing function and a set of N weight vectors[1], $W = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N\}$. The subproblem j is denoted by $g(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}_j)$, or g_j for simplicity. Each weight vector $\boldsymbol{w} \in W$ satisfies the following conditions [33]:

$$\boldsymbol{w} = [w_1, w_2, \ldots, w_m]^T \tag{2.9}$$

[1]The terms *reference vector* and *weight vector* are used interchangeably in this document.

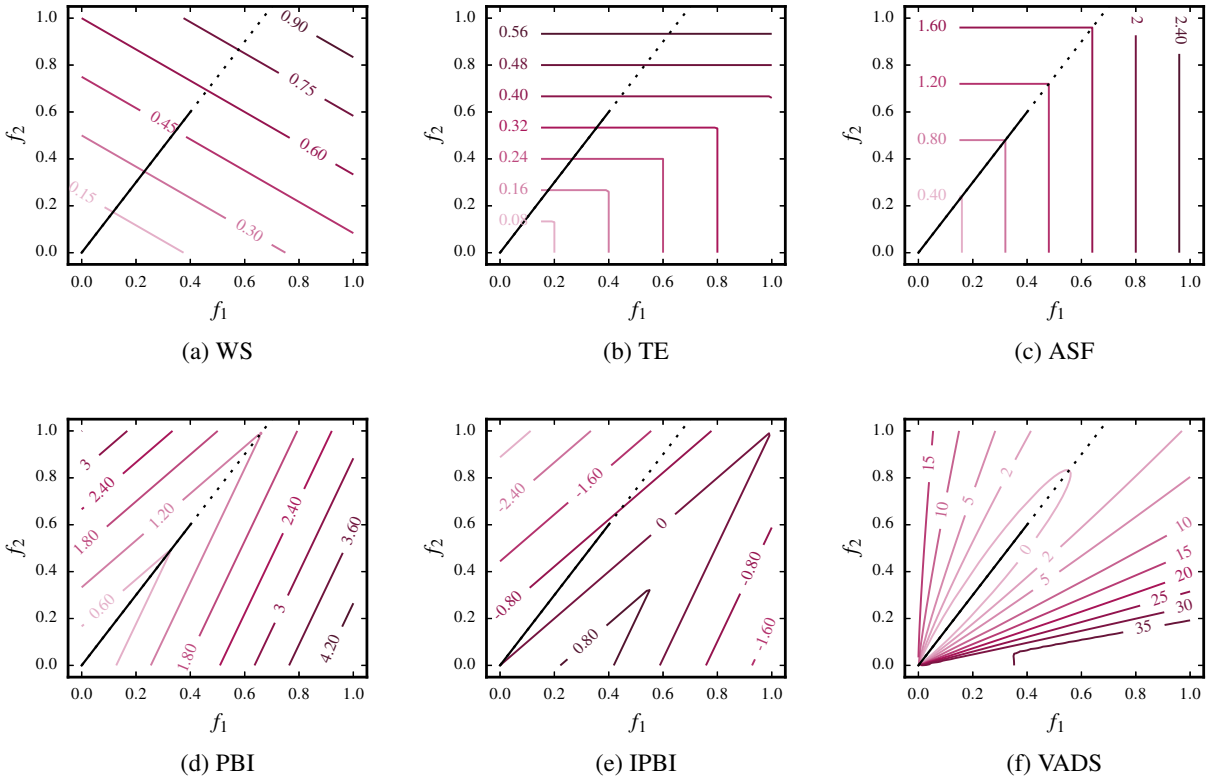


Figure 2.1: Contour plots of scalarizing functions for the reference vector $\boldsymbol{w} = [0.4, 0.6]^T$.

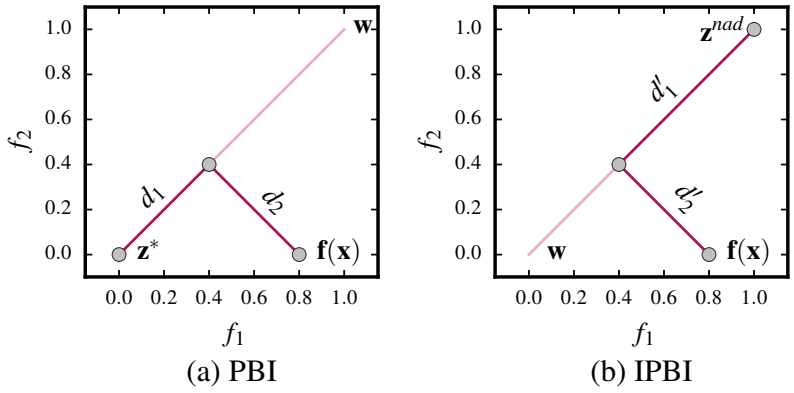


Figure 2.2: Interpretation of PBI and IPBI for $\boldsymbol{w} = [0.5, 0.5]^T$.

Name	Definition		
WS	$\min g_{ws}(\boldsymbol{x}; \boldsymbol{w})$	$=$	$\sum_{i=1}^{m} w_i f_i(\boldsymbol{x})$
TE	$\min g_{te}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{z}^*)$	$=$	$\max_{i=1,\ldots,m} \{w_i \lvert f_i(\boldsymbol{x}) - z_i^* \rvert\}$
ASF	$\min g_{asf}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{r})$	$=$	$\max_{i=1,\ldots,m} \left\{ \frac{1}{w_i} \lvert f_i(\boldsymbol{x}) - r_i \rvert \right\}$
PBI	$\min g_{pbi}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{z}^*)$	$=$	$d_1 + \theta d_2$
IPBI	$\max g_{ipbi}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{z}^{nad})$	$=$	$d_1' - \theta d_2'$
VADS	$\min g_{vads}(\boldsymbol{x}; \boldsymbol{v})$	$=$	$\frac{\lVert \boldsymbol{f}(\boldsymbol{x}) \rVert}{\left(\boldsymbol{v} \cdot \frac{\boldsymbol{f}(\boldsymbol{x})}{\lVert \boldsymbol{f}(\boldsymbol{x}) \rVert} \right)^q}$

Table 2.1: Scalarizing functions analyzed in this document.

$$\sum_{i=1}^{m} w_i = 1, \text{ and } w_i \geq 0, \text{ for } i = 1, 2, \ldots, m \tag{2.10}$$

$$w_i \in \left\{ 0, \frac{1}{H}, \frac{2}{H}, \ldots, \frac{H}{H} \right\}, \text{ for } i = 1, 2, \ldots, m \tag{2.11}$$

This common approach for generating an evenly distributed set of vectors was proposed by Das and Dennis [13]. The total number of reference vectors is given by:

$$N = \binom{H + m - 1}{m - 1} \tag{2.12}$$

where H is a positive integer. This parameter manages the number of divisions along each objective [15]. The set given in Eq. 2.11 shows the possible values that can be assigned to w_i. As the value of H grows, the number of possible values for w_i increases, leading to a better coverage of the objective space. In [33], the authors suggested to use a small number, such as 10^{-6}, when $w_i = 0$. We followed this recommendation in this document.

We have selected six scalarizing functions to compare their abilities to deal with distinct types of MOPs. These approaches are summarized in Table 2.1. Figure 2.1 depicts the contour plots

of the scalarizing functions. Most of these functions require the ideal or Nadir vectors. The ideal vector $\boldsymbol{z}^* = [z_1^*, \ldots, z_m^*]^T$ contains the minimum value for each objective function from a given set of solutions P [15]:

$$z_i^* = \min_{i=1,\ldots,m} \{f_i(x) : \boldsymbol{x} \in P\} \tag{2.13}$$

In contrast, the Nadir vector $\boldsymbol{z}^{nad} = [z_1^{nad}, \ldots, z_m^{nad}]^T$ comprises the worst value for each objective:

$$z_i^{nad} = \max_{i=1,\ldots,m} \{f_i(x) : \boldsymbol{x} \in P\} \tag{2.14}$$

The Weighted Sum (WS) associates each objective function with a weighting factor and minimizes the weighted sum of the objectives [59]. Although this approach is one of the most commonly employed scalarizing functions, it is unable to find the whole Pareto front for non-convex MOPs [59].

The weighted Tchebycheff (TE) function measures the distance from the ideal vector $\boldsymbol{z}^*$ to a solution $\boldsymbol{x}$ in the objective space [30]. According to Miettinen [59], the solutions found by this approach are weakly Pareto optimal solutions. A similar approach to TE is the Achievement Scalarizing Function (ASF) [77, 23]. Notice that the contour of ASF follows the direction defined by the reference vector $\boldsymbol{w}$, contrary to TE, as shown in Fig. 2.1. This approach requires a reference point $\boldsymbol{r} = [r_1, r_2, \ldots, r_m]^T$. In this document, we follow the recommendation by Hernández and Coello [23] to set this point as $r_i = 0$, for $i = 1, 2, \ldots, m$.

Zhang and Li [83] introduced the Penalty-based Boundary Intersection (PBI). Given an objective vector $\boldsymbol{f}(\boldsymbol{x})$ and a reference vector $\boldsymbol{w}$, this approach considers the distance from the objective vector to the ideal point $\boldsymbol{z}^*$ along the reference vector (d_1) and the perpendicular distance between both vectors (d_2):

$$d_1 = \frac{\|(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{z}^*)^T \cdot \boldsymbol{w}\|}{\|\boldsymbol{w}\|} \tag{2.15}$$

$$d_2 = \left\| \boldsymbol{f}(\boldsymbol{x}) - \left(\boldsymbol{z}^* + d_1 \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \right) \right\| \tag{2.16}$$

where $\boldsymbol{u} \cdot \boldsymbol{v}$ denotes the dot product between vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Both d_1 and d_2 are depicted in Fig. 2.2 (a).

Similar to PBI, the Inverted PBI (IPBI) [65] scalarizing approach involves two distances, d_1' and d_2', which are defined as follows:

$$d_1' = \frac{\|(\boldsymbol{z}^{nad} - \boldsymbol{f}(\boldsymbol{x}))^T \cdot \boldsymbol{w}\|}{\|\boldsymbol{w}\|} \tag{2.17}$$

$$d_2' = \left\|(\boldsymbol{z}^{nad} - \boldsymbol{f}(\boldsymbol{x})) - \left(d_1' \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right)\right\| \tag{2.18}$$

Given an objective vector $\boldsymbol{f}(\boldsymbol{x})$ and a reference vector $\boldsymbol{w}$, d_1' is the distance from the objective vector to $\boldsymbol{z}^{nad}$ along the direction of $\boldsymbol{w}$, and d_2' is the perpendicular distance between the objective vector and $\boldsymbol{w}$. Both d_1' and d_2' are illustrated in Fig. 2.2 (b). The conventional PBI moves a solution toward the obtained ideal point $\boldsymbol{z}^*$ by minimizing two distances, d_1 and d_2. In contrast, IPBI aims to find good solutions by maximizing d_1' and minimizing d_2', controlled by θ. The intuition behind this approach is that solutions with a large d_1' (distance to the Nadir point) may be close to the Pareto front. That is, solutions with a large d_1' and a small d_2' are preferred by IPBI [65]. The penalty parameter of both PBI and IPBI is defined using the recommended value $\theta = 5$.

Hughes [25] introduced the Vector Angle Distance Scaling (VADS) score. This method is designed to identify the objective front rather than solely the Pareto front. The objective front is the entire leading edge of the feasible objective space [25]. Two parameters are required for this approach: A target vector $\boldsymbol{v}$, and q. Given a weight vector $\boldsymbol{w}$, the target vector $\boldsymbol{v}$ is a unit-length vector defined as $\boldsymbol{v} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$, whereas q controls the size of the tear-drop-shaped contour defined by VADS, as shown in Fig. 2.1 (f). A higher value of q leads to a thinner shape. According to Hughes [25], a high value of q is recommended for dealing with sharp concavities in the Pareto front. In this work, we employ the recommended value for this parameter, $q = 100$.

2.4 Performance Indicators

To assess the performance of an algorithm, the inverted generational distance (IGD) [12, 40], the hypervolume [89], and the Solow-Polasky (SP) [69] indicator have been adopted in this document. Let A be the Pareto front approximation obtained by an optimizer. These indicators express the quality of approximation A in terms of either convergence or diversity.

In order to use the IGD, we need to provide a reference point set P sampled from the Pareto front. In this document, the reference point set P for each problem is analytically created as recommended in [10]. Given an approximation A and a reference point set P, this indicator expresses the closeness between A and P as follows [12, 40]:

$$IGD(A, P) = \frac{1}{|P|} \sum_{p \in P} \|p - a\|_2 \tag{2.19}$$

where a is the nearest solution to p among all solutions in A, and $\|\cdot\|_2$ denotes Euclidean distance. A low IGD value is preferred.

The hypervolume indicator measures the extension of the non-dominated region enveloped by A, and limited by a reference point $\boldsymbol{r} = [r_1, r_2, \ldots, r_m]^T$. This indicator is known to be Pareto compliant [20], and captures in a single scalar value both the spread and closeness of the solutions in A with respect to the Pareto front [6]. A larger hypervolume value is preferred. In this document, the reference vector is defined as $\boldsymbol{r} = [1.1, 1.1, \ldots, 1.1]^T$. A recent method for computing this indicator has been employed [6]. Before computing its hypervolume, an approximation A is preprocessed as follows. First, we filter the approximation using the reference vector $\boldsymbol{r}$. Only those solutions in A inferior to the reference point $\boldsymbol{r}$ are kept. Then, we use Pareto dominance to preserve non-dominated solutions. To this end, we have employed a parallel approach called very fast non-dominated sorting [68]. After applying both filters to A, the hypervolume is computed. All hypervolume values presented in this document were normalized to the range $[0, 1]$ by dividing $\Pi_{i=1}^{m} r_i$ as recommended

in [9].

To assess the diversity of a population, the SP measure is employed. This measure has been used to determine biological diversity [69] and as an indicator by other MOEAs [71, 35]. We have selected the SP measure for its properties [72]:

- *Monotonicity in variates*. The diversity of a set of solutions P should increase when adding an individual b not yet in P.
- *Twinning*. Diversity should stay constant when adding an individual p already in P.
- *Monotonicity in distance*. If all pairs of solutions in a population P are at least as dissimilar (according to a distance metric d) as those in another population P', the diversity of P should not be smaller than the diversity of P'.

Given a set P with n objective vectors. The SP measure of P is defined as [72]:

$$D(P) = \boldsymbol{e} M^{-1} \boldsymbol{e}^T \tag{2.20}$$

where $\boldsymbol{e} = [1, 1, \ldots, 1]^T$ and $M = (m_{ij})$ is an (n, n)-matrix denoted as follows:

$$m_{ij} = \exp(-\theta d(\boldsymbol{p}_i, \boldsymbol{p}_j)) \text{ for all } 1 \leq i, j \leq n \tag{2.21}$$

where $d(\boldsymbol{p}_i, \boldsymbol{p}_j)$ denote the distance between two objective vectors, and θ is an scaling factor for d. In this document, θ is specified as $\theta = 1$ and the Euclidean distance is used to compute d. A larger value of SP is preferred.

2.5 Test Problems

In order to evaluate the behavior of an algorithm, we have selected two benchmarks from the specialized literature: DTLZ [17, 37] and MaF [10]. These problems are scalable with respect to

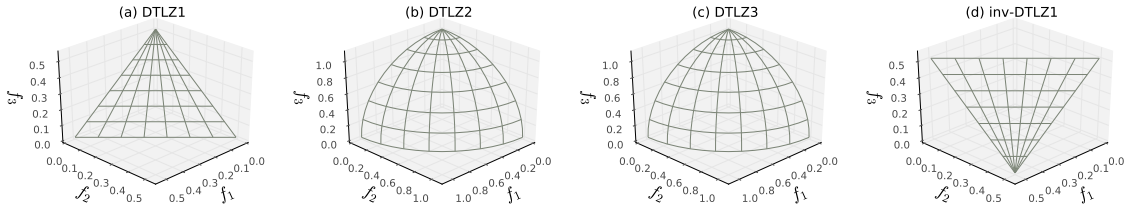


Figure 2.3: Pareto fronts of selected test problems for $m = 3$ objectives.

both the number of objectives (m) and decision variables (n). In this document, we have considered problem instances with $m = 3, 5, 8, 10$ objectives. The number of decision variables for DTLZ and MaF is determined by $n = k + m - 1$, where the recommended value of k depends on the test problem: $k = 5$ for DTLZ1 and inv-DTLZ1; $k = 10$ for DTLZ2 and DTLZ3; and $k = 10$ for all problem in MaF. Figure 2.3 depicts the Pareto front of some of these problems for $m = 3$. In the following, we provide a brief description of each problem.

DTLZ1 is a multimodal problem with $(11^k - 1)$ local Pareto optimal fronts [17]. These local fronts act as attractors that create difficulties for the convergence of a MOEA to the real Pareto front. On the other hand, inv-DTLZ1 is a recent modification of DTLZ1 which transforms the shape of the Pareto front. This transformation inverts the original Pareto front so that many of the reference vectors employed for a decomposition-based method may become unhelpful [36]. As a result, many decomposition-based MOEAs have difficulties covering the irregular shape of its Pareto front [33]. DTLZ2 is a unimodal problem with a concave Pareto front. Like DTLZ1, DTLZ3 is a multimodal problem. This problem contains $(3^k - 1)$ local Pareto optimal fronts. All local fronts are parallel and a MOEA can get stuck at any of them, before converging to the global Pareto front. These four test problems have been selected to assess the ability of a MOEA to deal with multimodality and both regular and irregular shapes of the Pareto front.

Most of the MaF problems are based on problems from the literature. MaF1 is similar to DTLZ1. This problem has a linear Pareto front. Although, its Pareto front is inverted and has a relatively simple Pareto set. This problem is specially challenging for decomposition-based problems. MaF2 is

based on DTLZ2. The shape of its Pareto front is concave. In addition, this problem is harder that DTLZ2, because MaF2 increases the difficulty of convergence. MaF3 is a multimodal problem with a convex Pareto front. Finally, both MaF4 and MaF5 are badly-scaled problems[2]. Unlike all previous problems, which the objective values are in the range [0, 1], the objectives in these two problems are scaled. MaF4 has an inverted Pareto front, whereas MaF5 has an convex Pareto front. These problems are employed to determine whether an optimizer is able to deal with scaled Pareto front.

[2]The term *badly-scaled* is often employed in the literature to refer to problems with an increasing scaling factor for each objective. For instance, a three-objective badly-scaled problem can have these functions: $f_1 \in [0, 1]$, $f_2 \in [0, 10]$, and $f_3 \in [0, 100]$, where the scaling factor for the i-th objective function is 10^{i-1} [15].

3

Difficulties of Decomposition-based MOEAs

Some researchers have found that the performance of decomposition-based methods, such as MOEA/D, depends on (1) the selection of the scalarizing function and (2) the shape of the Pareto front [33, 52, 9, 70]. In this chapter, we provide a detailed description of both issues derived from literature and our own experimentation. First, we conducted a preliminary study to evaluate the performance of MOEA/D with respect to several scalarizing functions and test problems. The goal of this study is to determine which scalarizing functions are more suitable to address a given test problem. Our experimental results show that PBI, VADS, and ASF are the best scalarizing approaches to solve the test problems. Also, we have observed that the diversity of MOEA/D is deteriorated when the reference vectors does not reflect the shape of the Pareto front. The proposed method is based on the findings described in this chapter.

3.1 Selection of Scalarizing Function

In this section, we are interested in evaluating six scalarization approaches for many-objective optimization. Four test problems from the specialized literature have been adopted: DTLZ1, DTLZ2, DTLZ3, and inv-DTLZ1 [17, 36, 15]. We have selected MOEA/D for this study given its computational efficiency to handle MaOPs. The population size of this optimizer depends on the number of objectives (m) and the granularity level (H) adopted to generate the reference vectors. To analyze its performance, four factors are considered in our experimental setup: number of objectives (8), scalarizing approach (6), granularity level (6), and test problem (4). The cardinality for each parameter is shown in parentheses. These four factors are detailed in Table 3.1.

Previous works have studied the influence of these four factors separately. In [27, 30], the authors analyzed the behavior of MOEA/D with respect to distinct scalarizing approaches in combinatorial test problems, whereas in [31], the authors were interested on how the population size affected the metaheuristic. Unlike these works, our goal is to study the overall influence of these four factors in the performance of MOEA/D.

In this experiment, $1,152$ $(= 8 \times 6 \times 6 \times 4)$ configurations of MOEA/D have been evaluated, one for each parameter setting of the factors given in Table 3.1. Also, each configuration has been evaluated 31 times to gather statistical information. A distinct seed was employed for the pseudo-random number generator each time. Simulated binary crossover and polynomial mutation are employed as variation operators [16]. The configuration of these operators is as follows. The distribution indices for crossover and mutation are $\eta_c = 20$ and $\eta_m = 15$, respectively. The crossover rate is $p_c = 1.0$, whereas the mutation rate is $p_m = 1/n$, where n is the number of decision variables. We use the recommended neighborhood size for MOEA/D, that is, $T = 20$ [83]. The search process is finished after reaching a predefined number of generations. The number of generations assigned to each test problem was defined as $m \times C$, where m is the number of objectives and $C = 30,000$.

Table 3.2 shows the overall performance of MOEA/D. This table has been filled as follows.

Factor	Values
Number of objectives (m)	$3, 4, \ldots, 10$
Scalarizing function	WS, TE, ASF, PBI, IPBI, VADS
Granularity level (H)	$2, 3, \ldots, 7$
Test problem	DTLZ1, DTLZ2, DTLZ3, inv-DTLZ1

Table 3.1: Factors considered for the experimental setup.

Consider WS and DTLZ1 for $m = 3$ objectives. First, we evaluate the scalarizing function using six granularity levels, $H = 2, 3, 4, 5, 6, 7$ (6×31 runs in total). We then measure the performance of each configuration with respect to the hypervolume indicator. Finally, we compute the mean, median, and standard deviation of the hypervolume values. From this table, we can see that ASF and PBI are preferable for addressing DTLZ1. On the other hand, PBI showed good performance in DTLZ2. ASF and VADS are well suited for DTLZ3 for $m = 3$ objectives. Although, PBI have shown better performance in this problem when the number of objectives increases. Finally, TE seems to be the best function to address inv-DTLZ1.

Figure 3.1 shows the general performance of MOEA/D regarding the four test problems. A heatmap for each pair of test problem and scalarizing approach is given. The performance of the corresponding scalarizing function according to the number of objectives (m), granularity level (H), and test problem is shown as shades of blue. The darker the shade, the better (higher) the performance of the corresponding configuration with respect to the hypervolume indicator.

Consider the behavior of MOEA/D with WS, as shown in Fig. 3.1 (a)-(d). First, we can observe that its performance is almost the same in DTLZ1 regardless of the number of objectives or granularity level. In other words, the parameters m and H do not influence significantly on the performance of WS in DTLZ1. Now, consider Fig. 3.1 (b) and (c) for DTLZ2 and DTLZ3, respectively. From these plots, it can be observed a gradient-like pattern, where higher (better) hypervolume values are located in the upper right corner. This suggest that a larger population size is recommended to address both problems with WS. The performance of WS is inferior to most of the remaining scalarizing functions

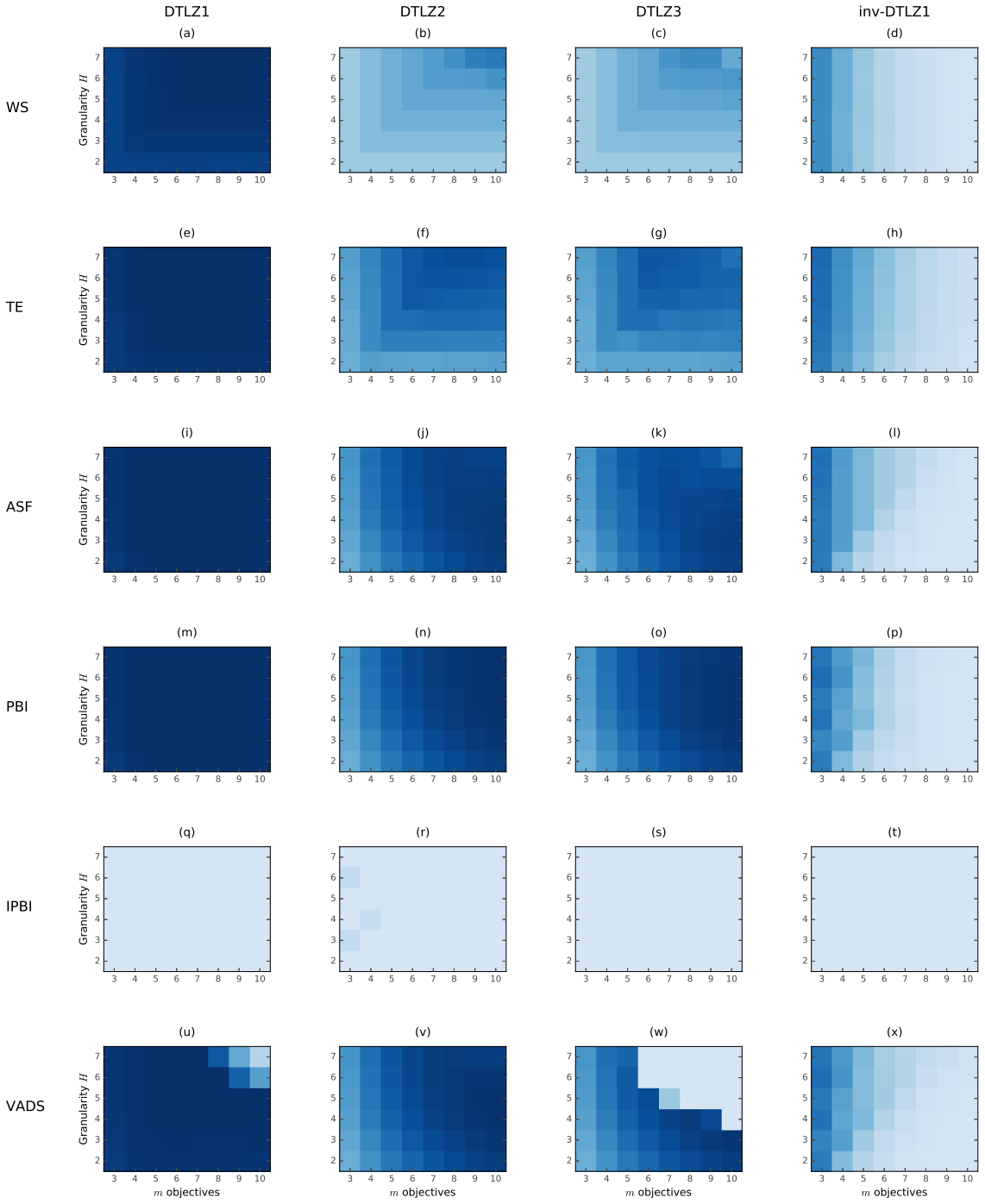


Figure 3.1: Overall performance of scalarizing functions. Each heatmap considers four parameters: scalarizing function, number of objectives (m), granularity level (H), and test problem. The performance of a given scalarizing function according to these parameters is shown as shades of blue. The darker the shade, the better the performance of the scalarizing function.

for DTLZ2 and DTLZ3. This is attributed to the limited performance of WS in non-convex Pareto front shapes. In this case, only the extremes of the Pareto front can be achieved by this approach. It is worth noting that both plots, Fig. 3.1 (b) and (c), show the same gradient-like pattern since the shape of the Pareto front is the same for DTLZ2 and DTLZ3. This also explains the difference between the heatmaps of DTLZ1 and inv-DTLZ1: They are different because the shape of their corresponding Pareto front is also different. We recommend to compare the scalarizing functions through the columns of Fig. 3.1. That is, to compare the performance of the scalarizing functions regarding the same problem.

Figure 3.1 (e)-(h) shows the behavior of MOEA/D with TE. From these plots, we can see that TE behaves similarly to WS. However, TE achieved a higher hypervolume in DTLZ2, DTLZ3, and inv-DTLZ1. The parameter H alters the performance of TE; a higher population size leads to a higher hypervolume value for DTLZ2 and DTLZ3. From our results, TE is the best scalarizing function for addressing inv-DTLZ1. Figure 3.1 (m)-(p) shows the overall performance of PBI. Our experimental results show that this approach has an excellent behavior regardless of the test instance. Similar results are also observed when using ASF. On the other hand, IPBI did not perform well in any of the test problems. Finally, Fig. 3.1 (u)-(x) shows the performance of MOEA/D with VADS. In this case, the influence of H is higher than in the other scalarizing functions, specially in DTLZ1 and DTLZ3. Given a number of objectives, increasing the population size does not necessarily lead to a better performance when using VADS in these multimodal problems. Notice how the heatmap shown in Fig. 3.1 (w) is divided into two regions. Consider the case for $m = 8$ and $H = 4, 5$ for DTLZ3. A high hypervolume value is achieved by using $H = 4$. However, when the granularity level is increased to five, the hypervolume value is reduced considerably. We attribute this issue to the coverage of the Pareto front of DTLZ3. As H increases, it seems that the coverage of the Pareto front is deteriorated. Figure 3.2 shows a parallel coordinates plot (PCP) for the approximations obtained by MOEA/D-VADS with $m = 8$, and $H = 3, 4, 5$. This plot shows the coverage of each approximation regarding each objective function. The range of each objective function of DTLZ3 is $[0, 1]$. The

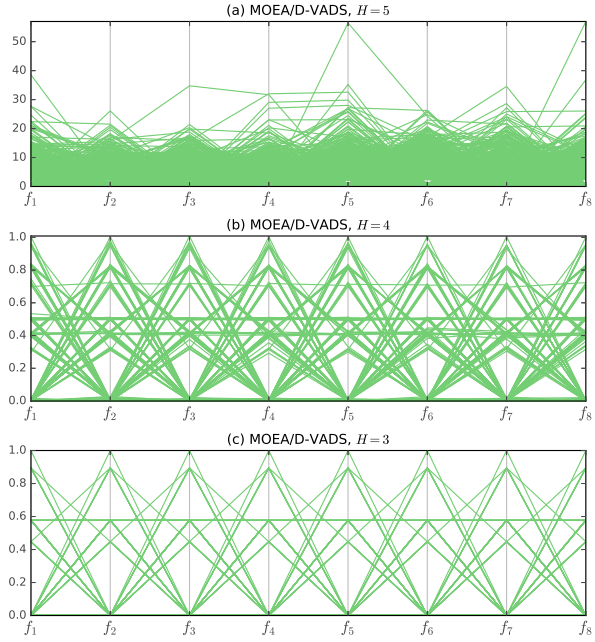


Figure 3.2: Population of MOEA/D-VADS using three granularity levels in DTLZ3: (a) $H = 5$, (b) $H = 4$, and (c) $H = 3$.

broader the coverage of this range, the better the performance of the given approximation. From this figure, we can see how the population that corresponds to $H = 4$ has a better coverage. On the other hand, many solutions of the population corresponding to $H = 5$ are not in the range [0, 1]. Instead, most of them are in the range $[0, 20]$. When the hypervolume is computed, the approximation is filtered with respect to a reference point. After filtering the approximation corresponding to $H = 5$, we found that any of the solutions in such an approximation are inferior to the reference point. That is, this approximation is far away from the Pareto front and its hypervolume value tends to zero.

3.2 Diversity deterioration in irregular problems

Ishibuchi et al. [33] showed that decomposition-based approaches have difficulty to address MOPs when the shape of its Pareto front is irregular. To get a better understanding of such a deterioration, we analyze the performance of MOEA/D in inv-DTLZ1, a test problem similar to the one employed

in their study [33]. After experimentation, we identify that the following issues are related to the diversity deterioration of MOEA/D in irregular problems:

1. If multiple reference vectors conduct to the same solution, then these vectors are considered redundant. As a result, some vectors will be over-crowded and the computational resources will be inefficiently spent on improving redundant subproblems.

2. If the reference vectors does not reflect the shape of the Pareto front, then some regions in the Pareto front will be poorly-crowded. That is, the locations of the reference vectors regarding the Pareto front is important to achieve a broader coverage.

The first issue is illustrated in Fig. 3.3. This figure shows the solution set obtained by MOEA/D with PBI. We are interested in analyzing the relationship between the reference vectors and the solutions. Fig. 3.3 (left) shows the set of reference vectors, denoted as W. Each plot highlights a different reference vector: $\boldsymbol{w}_1$, $\boldsymbol{w}_{15}$, $\boldsymbol{w}_{28}$, and $\boldsymbol{w}_{40}$. The plots in Fig. 3.3 (center) and (right) show the solution set found in DTLZ1 and inv-DTLZ1, respectively. Notice that the solutions are shaded. The lighter the shade, the better (lower) the value of the scalarizing function of the corresponding solution with respect to the reference vector $\boldsymbol{w}_i$. The best solutions are shown as white points: $\boldsymbol{x}_1$, $\boldsymbol{x}_{15}$, $\boldsymbol{x}_{28}$, and $\boldsymbol{x}_{40}$. Observe the relationship between reference vectors and solutions in Fig. 3.3 (left) and (center). There is a one-to-one mapping between them, as mentioned in [33], since the best solution for subproblem i (white point) is close to the location of the reference vector $\boldsymbol{w}_i$. However, this relationship does not hold for inv-DTLZ1. Note that, regardless of the location of the four reference vectors, all of them conduct to the same solution (white point). In other words, the same solution can be shared by multiple reference vectors in inv-DTLZ1 [33]. In this example, these four reference vectors are redundant because all of them help MOEA/D to find the same solution. As a result, computational resources are inefficiently employed by optimizing the subproblems corresponding to redundant vectors. A better approach would have been to remove

redudant vectors to save computational resources, or adapt the locations of the redudant vectors to improve the diversity of the population.

The second issue is illustrated in Fig. 3.4. This figure shows three different sets of reference vectors, denoted as W, A, and B. A set like W is often employed when decomposing a problem. On the other hand, A and B are created from W by adding some vectors in two different regions. Both A and B have 154 vectors, where 63 of them are distributed in distinct regions of the objective space to increase the density of solutions in such regions. We use these sets to show that the locations of the reference vectors can either improve or deteriorate the diversity of the solution set obtained by MOEA/D.

Figure 3.5 shows the approximation achieved by MOEA/D in DTLZ1 using the three sets. From this figure, we can see that the solution set obtained with W has a uniform coverage of the Pareto front, whereas the solutions in the other two approximations resemble the vectors in A and B. On the other hand, Fig. 3.6 shows the approximation obtained in inv-DTLZ1. Notice that the diversity of the solutions reached when using W is not as uniform as in the previous problem. In this case, there are some poorly-crowded niches, whereas many of the solutions are gathered in the boundaries of the Pareto front. That is, the approximation is not completely uniform. Now, compare the solutions obtained when using W and A, as shown in Fig. 3.6 (a) and (b). Observe that when the reference vectors are properly added, the previous niches are now more crowded. Weight adaptation methods perform a similar task. These methods reallocate existing reference vectors with the aim of improving the coverage of the Pareto front. Finally, observe the solutions obtained by using B. Notice that many solutions are garethed in one of the boundaries. This example show that adding reference vectors can improve the coverage of the Pareto front if the vectors are added in the appropriate regions.

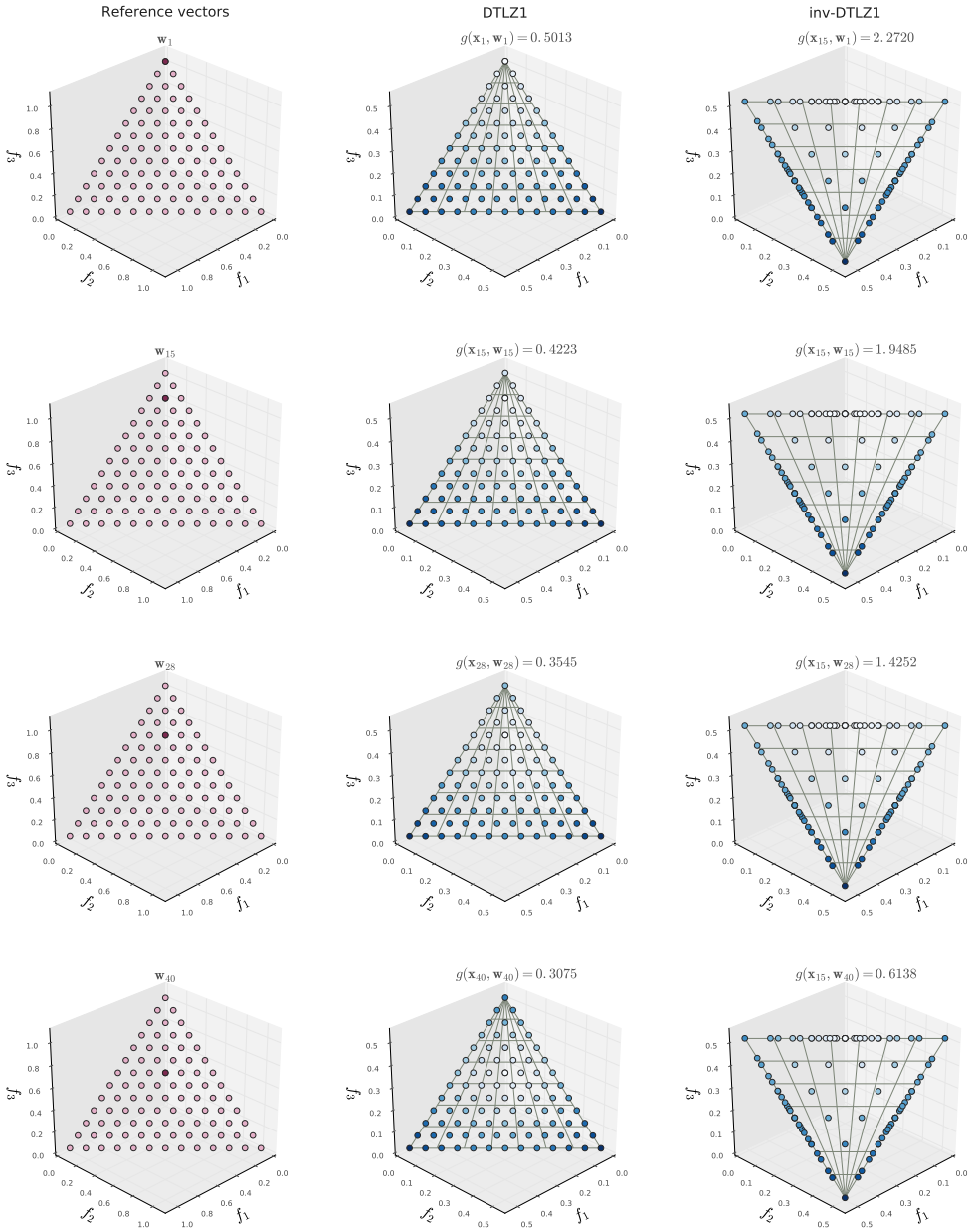


Figure 3.3: Relationship between reference vectors and solutions found by MOEA/D. The value of $g(\boldsymbol{x}, \boldsymbol{w})$ is shown as shades of blue. The lighter the shade, the better the value of $g(\boldsymbol{x}, \boldsymbol{w})$. (left) Location of four reference vectors from W: $\boldsymbol{w}_1$, $\boldsymbol{w}_{15}$, $\boldsymbol{w}_{28}$, and $\boldsymbol{w}_{40}$. The best solution of these vectors is highlighted in (center) for DTLZ1 and (right) for inv-DTLZ1.

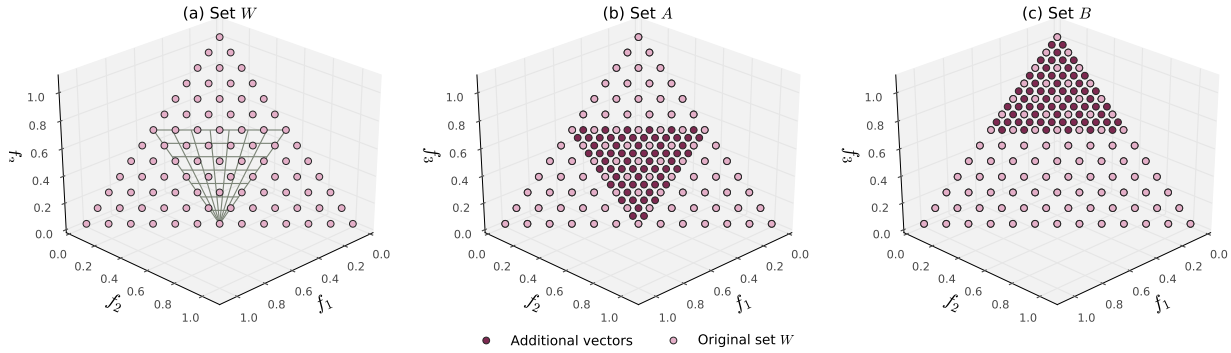


Figure 3.4: Sets of reference vectors. (a) Relation between the set W and the Pareto front of inv-DTLZ1. Dense sets (b) A and (c) B with 154 reference vectors, where 63 of them are distributed in distinct regions of the objective space to increase the density of solutions in such regions.

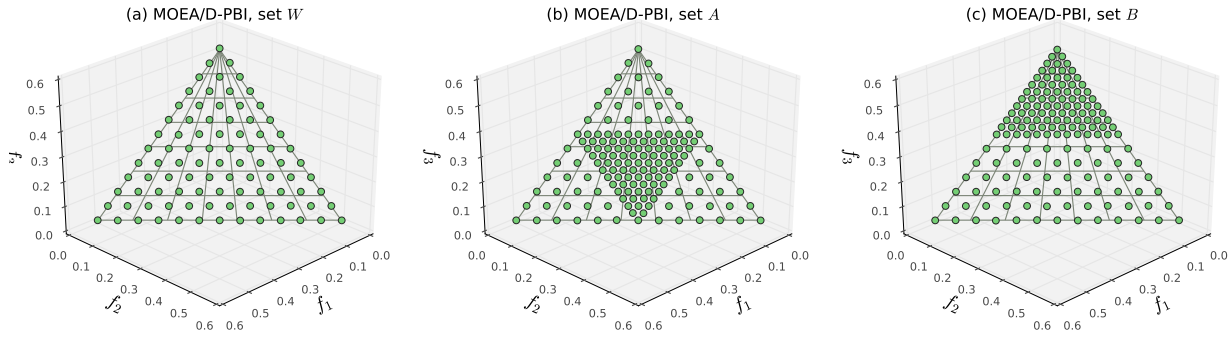


Figure 3.5: Approximation obtained by MOEA/D in DTLZ1 using sets (a) W, (b) A, and (c) B.

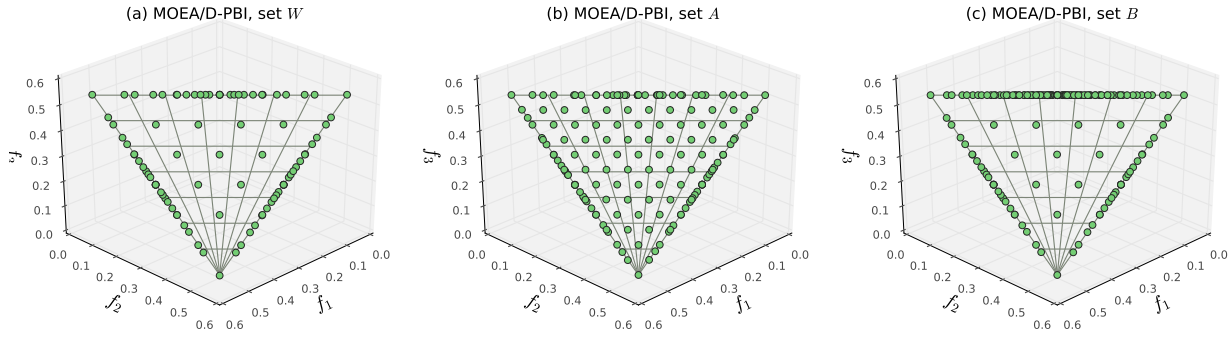


Figure 3.6: Approximation obtained by MOEA/D in inv-DTLZ1 using sets (a) W, (b) A, and (c) B.

3.3 Summary

The aim of this chapter is to gain a better insight about the limitations of decomposition-based methods. To this end, we use a representative algorithm from the literature: MOEA/D. Here, we study the behavior of this algorithm with respect to several scalarizing approaches: Weighted Sum (WS), Tchebycheff (TE), Achievement Scalarizing Function (ASF), Penalty-based Boundary Intersection (PBI), Inverted PBI (IPBI), and Vector Angle Distance Scaling (VADS) score. We want to determine which of these scalarizing approaches are more suitable to address a particular problem. Unlike previous studies [27, 30, 31], we are interested in analyzing four factors: scalarizing function, test problem, number of objectives (m), and granularity level (H), where the last two parameters controls the population size of MOEA/D. In the following, we summarize the conclusions regarding the overall performance of each scalarizing function:

- Five of the six scalarizing functions perform well in DTLZ1, regardless of the number of objectives or granularity level.
- The behavior of both PBI and ASF is almost the same in the four tested problems. Both scalarizing functions perform well in DTLZ1. According to our results, these functions are recommended to address DTLZ2 and DTLZ3 for many-objective optimization.
- The performance of WS and TE is similar. However, these functions are not recommended to address DTLZ2 and DTLZ3. Instead, PBI and ASF should be employed for solving these test problems.
- The performance of VADS seems to be dependent on the number of objectives, granularity level, and test problem. Our results suggest that there is a relationship between these factors, specially H. We observe that a higher granularity level can lead to a deterioration on the performance in some cases. This issue is attributed to the coverage of the Pareto front.

- The six scalarizing functions are not suitable for inv-DTLZ1. Other methods, such as adaptation of reference vectors [36, 37, 70, 52], can be employed for tackling this kind of problems.

We also analyze how the performance of MOEA/D is deteriorated in inv-DTLZ1. We find two underlying issues. First, if multiple reference vectors lead to the same solution, then they are redundant. As a result, computational resources are inefficiently spent in improving redundant subproblems. Second, if the references vectors cannot reflect the shape of the Pareto front, then the diversity of the population will be deteriorated. In our experiments, we show that the diversity of the population can be improved when more vectors are added in specific regions. As a result, selecting a scalarizing function and a set reference vectors is vital for achieving a close and well-distributed approximation of the Pareto front.

Problem	m	WS	TE	ASF	PBI	IPBI	VADS
DTLZ1		0.91301343	0.96199359	**0.96915218**	0.96913659	0.00009173	0.96295474
	3	0.91200850	0.96251850	0.97196600	**0.97199800**	0.00000000	0.96576550
		±0.00585176	±0.00567034	±0.00819284	±0.00818117	±0.00124761	±0.00792521
		0.97265478	0.99334184	0.99838010	**0.99859651**	0.00000000	0.99217229
	5	0.98463650	0.99798500	0.99872400	**0.99906150**	0.00000000	0.99253400
		±0.02358850	±0.00896156	±0.00098020	±0.00107675	±0.00000000	±0.00345707
		0.97745739	0.99614076	**0.99995411**	0.99987537	0.00000000	0.94190910
	8	0.99354550	0.99936150	**0.99997400**	0.99992750	0.00000000	0.99004700
		±0.02919819	±0.00635252	±0.00005530	±0.00013847	±0.00000000	±0.15831913
		0.97862737	0.99631397	**0.99998081**	0.99951932	0.00000000	0.78409624
	10	0.99272700	0.99932600	**0.99999000**	0.99975500	0.00000000	0.98042350
		±0.02794588	±0.00595063	±0.00003317	±0.00052347	±0.00000000	±0.33473203
DTLZ2		0.24868500	0.44005134	0.47791997	0.47791907	0.03487035	**0.47792000**
	3	0.24868500	0.43658600	0.48903450	0.48903200	0.00000000	**0.48903450**
		±0.00000000	±0.02702079	±0.04407808	±0.04407879	±0.04940156	±0.04407897
		0.34699800	0.65685181	0.75155927	0.76355194	0.00267512	**0.76361330**
	5	0.37907900	0.71037650	0.76826350	0.78176050	0.00000000	**0.78177850**
		±0.04946862	±0.09180497	±0.04998117	±0.05688671	±0.00792118	±0.05694512
		0.40765095	0.71523978	0.91214085	**0.93722712**	0.00000000	0.93416734
	8	0.40004650	0.74999450	0.91811800	**0.95151650**	0.00000000	0.94636400
		±0.10716271	±0.13683920	±0.02430886	±0.03208154	±0.00000000	±0.03000677
		0.42918351	0.70281454	0.93525853	**0.97348270**	0.00000000	0.96186445
	10	0.38949050	0.74344050	0.93801450	**0.98204650**	0.00000000	0.96787750
		±0.13733405	±0.13327099	±0.02005683	±0.01696217	±0.00000000	±0.02078744
DTLZ3		0.24284839	0.43355590	0.47099927	0.46973971	0.00000000	**0.47291813**
	3	0.24385200	0.42692750	**0.48002500**	0.47456100	0.00000000	0.47809050
		±0.00435664	±0.02800872	±0.04466939	±0.04407677	±0.00000000	±0.04474942
		0.34490141	0.64207752	0.73297141	**0.75607637**	0.00000000	0.74178063
	5	0.37585050	0.70780050	0.73246250	**0.76818950**	0.00000000	0.76673200
		±0.04917925	±0.10329643	±0.04823467	±0.05489116	±0.00000000	±0.07699039
		0.41072554	0.67757565	0.86964718	**0.92808227**	0.00000000	0.45302311
	8	0.38500450	0.71597800	0.87386300	**0.94107700**	0.00000000	0.35585750
		±0.11335353	±0.12278610	±0.03484023	±0.02739239	±0.00000000	±0.45372167
		0.39123034	0.64698315	0.88045067	**0.96351090**	0.00000000	0.33856477
	10	0.37953800	0.66042900	0.90840300	**0.97172900**	0.00000000	0.00000000
		±0.10275298	±0.10060717	±0.07179428	±0.05662490	±0.00000000	±0.44956973
inv-DTLZ1		0.57544831	**0.69796438**	0.67098342	0.66448404	0.00027757	0.66839931
	3	0.57256850	**0.70429000**	0.66294100	0.66483550	0.00000000	0.67234400
		±0.00818616	±0.02415931	±0.01980219	±0.03550512	±0.00377536	±0.03137438
		0.26205674	**0.38662516**	0.29698290	0.28877831	0.00000000	0.28950847
	5	0.26121300	**0.39166050**	0.34383450	0.31506950	0.00000000	0.32058450
		±0.00853930	±0.02825750	±0.07030009	±0.06081845	±0.00000000	±0.06614932
		0.06498948	**0.12813544**	0.05375448	0.04025093	0.00000000	0.05580406
	8	0.06337550	**0.13275450**	0.04428050	0.03952800	0.00000000	0.05105800
		±0.00419684	±0.01552830	±0.02995543	±0.00753026	±0.00000000	±0.03266603
		0.02281222	**0.05496235**	0.01193435	0.01167460	0.00000000	0.01756915
	10	0.02269050	**0.05847850**	0.01157450	0.01126900	0.00000000	0.01590750
		±0.00083633	±0.00735339	±0.00416892	±0.00247945	±0.00000000	±0.01170026

Table 3.2: Overall performance of scalarizing functions. Mean, median, and standard deviation of hypervolume values are shown for each test instance. Best performance is shown in bold.

4

Multiple Survival Selection

In this chapter, we propose a framework based on MOEA/D for the collaborative use of scalarizing functions, called multiple survival selection (MSS). Our approach involves two steps: evaluation of scalarizing functions and performance assessment. First, the proposed method evaluates $l = 6$ candidate scalarizing functions for survival selection. In this step, each candidate identifies a solution set with the best individuals from both the parents and the offspring. Second, we evaluate the effectiveness of each scalarizing function regarding a performance indicator. The effectiveness of a scalarizing function depends on its ability to find a good solution set. Both steps are performed frequently throughout the search process, meaning that good scalarizing functions are promoted. In this chapter, we provide the detailed description of the proposed method and evaluate its performance on several test problems.

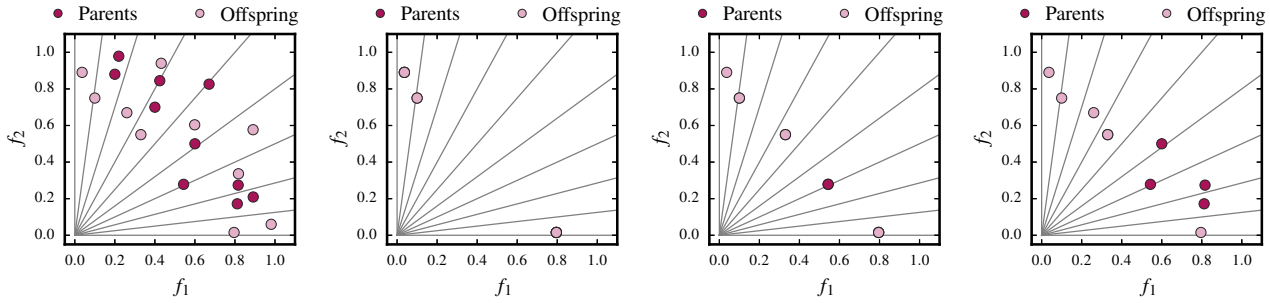
4.1 Description

Figure 4.1 illustrates how to use three scalarizing functions for survival selection: WS, TE, and PBI. We employ them to identify a solution set of N individuals from the parent and offspring solutions ($2N$), in a similar way to MOEA/D. Notice that the same solution can be selected multiple times. In this example, WS selects solutions that are close to the extremes of both objectives, whereas the solutions selected by TE and PBI are better distributed. We determine the effectiveness of each scalarizing function by evaluating each solution set with respect to a performance indicator called diversity comparison indicator (DCI) [50], introduced in Section 4.1.2. In this example, the solution set obtained by PBI is the best regarding such an indicator. A high DCI value implies that the selected solutions are better distributed in the objective space.

From Fig. 4.1, we can see that the solution set obtained in each case is different, and depends on both the scalarizing function and the properties of the problem. This observation remarks how important is selecting a scalarizing function for decomposing a problem.

We could test several candidate scalarizing functions exhaustively to identify the best one for a given problem, but this will be time consuming and computationally expensive. Instead, a better approach could be to evaluate them collaboratively, and identify the best one according to its effectiveness during a stage of the search process. We hypothesize that a decomposition-based method can address a broader range of problems by using multiple scalarizing functions collaboratively, promoting the best performing candidates throughout the search process.

In the proposed method, a different scalarizing function is employed at each stage of the search process depending on its performance. The duration of each stage is 25 generations in this document. At the end of each stage, the effectiveness of the scalarizing functions is assessed again to select the most suitable for the next stage. The framework of the proposed approach is given in Algorithm 1. Below, we provide the details of the performance indicator, the process of maintaining an auxiliary population for each scalarizing function, and the proposed procedure for performance assessment.



(a) Merged population (b) WS, DCI = 0.50 (c) TE, DCI = 0.83 (d) PBI, DCI = 1.00

Figure 4.1: Multiple survival selection. Each plot shows the individuals identified for each scalarizing function during survival selection. The performance of each scalarizing function (DCI) depends on its ability to identify promising solutions. A higher value is preferred.

4.1.1 Multiple Survival Selection

In this document, we consider six scalarizing functions, denoted as $\{g^1, g^2, \ldots, g^l\}$, where $l = 6$. For each scalarizing function, an auxiliary population P^k is defined. The goal of this auxiliary population is to maintain the solutions selected by a given scalarizing function during the search process, as illustrated in Fig. 4.1.

The proposed MSS is shown in Algorithm 1. First, the main population P is created randomly. The auxiliary populations $\{P^1, P^2, \ldots, P^l\}$ are also initialized with solutions taken from the main population. As the evolutionary process progresses, the content of each auxiliary population will depend on the solution set selected by the corresponding scalarizing function. The neighborhood structure B is then created. For each weight vector $\boldsymbol{w}_i$, the T nearest reference vectors to $\boldsymbol{w}_i$ are defined as its neighbors in the same manner as MOEA/D.

The main loop comprises three tasks: creation of offspring solutions, survival selection using auxiliary populations, and performance assessment. The offspring solutions are created from the current population in the same way as MOEA/D. We adopt simulated binary crossover (SBX) and polynomial mutation [16] as variation operators.

The next step is survival selection. Given the union of offspring and current solutions ($2N$), the best N solutions are selected for the next generation. In MOEA/D, only a single scalarizing function

Algoritmo 1 Multiple survival selection

Create initial population P.
Create auxiliary population $P^k = P$, for $k = 1, 2, \ldots, l$.
Evaluate each individual $\boldsymbol{x} \in P$ on m objective functions.
Create neighborhood structure $B(i)$, for $i = 1, 2, \ldots, N$.
Set $c_{k,j} = 0$, for $k = 1, 2, \ldots, l$, and $j = 1, 2, \ldots, b$.
Set $r = 0$
while termination condition is not met **do**
    $t = 2$
    for $i = 1, 2, \ldots, N$ **do**
        Create offspring $\boldsymbol{y}_i$ from $B(i)$
        Evaluate $\boldsymbol{y}_i$ on m objective functions
        for $k = 1, 2, \ldots, l$ **do**
            Find subproblem j according to Eq. 4.1.
            if $g^k(\boldsymbol{y}_i, \boldsymbol{w}_j) < g^k(\boldsymbol{x}_j, \boldsymbol{w}_j)$ **then**
                $P^k[j] = \boldsymbol{y}_i$ ▷ Update auxiliary population
            end if
        end for
    end for
    if t is 2 or multiple of 25 **then**
        Performance assessment:
            $c_k = \text{DCI}(P^k)$ for $k = 1, 2, \ldots, l$
        Update performance track:
            $j = t \bmod b$
            $c_{k,j} = c_k$, for $k = 1, 2, \ldots, l$
        Compute mean performance:
            $s_k = \frac{1}{b} \sum_{j=1}^{b} c_{k,j}$, for $k = 1, 2, \ldots, l$
        Select the best auxiliary population P^r:
            $r = \underset{k \in \{1,2,\ldots,l\}}{\arg\max} \{s_k\}$
    end if
    Update main population P:
        $P = P^r$
    $t = t + 1$
end while

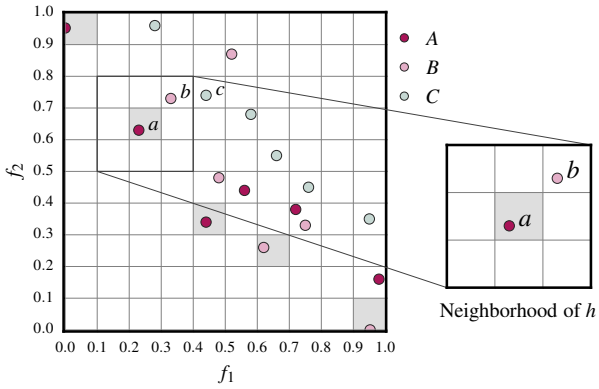
is employed in this step. Nonetheless, the proposed approach considers l scalarizing functions for this purpose. In other words, we have l potential solution sets for survival selection, where one of them may be better than the rest for the current stage of the search process.

In the following, we provide the details for comparison of solutions and maintenance of auxiliary populations. In this paper, we adopt the comparison strategy proposed by Zhou and Zhang [85] for MOEA/D-GRA. Unlike the conventional comparison strategy of MOEA/D, where a single offspring can replace multiple solutions, this approach is more restrictive. In other words, an offspring can replace at most a single solution in its neighborhood. The aim of this strategy is to preserve the diversity of the population. The comparison and replacement of solutions is performed in lines 12-17 of Algorithm 1. Given an offspring solution $\boldsymbol{y}_i$, we first identify the best-matched subproblem j for this solution. To this end, we find the j-th subproblem that can be improved most by trial solution $\boldsymbol{y}_i$:

$$j = \underset{p \in \{1,\ldots,N\}}{\arg\max} \left\{ \frac{g^k(\boldsymbol{x}_p, \boldsymbol{w}_p) - g^k(\boldsymbol{y}_i, \boldsymbol{w}_p)}{g^k(\boldsymbol{x}_p, \boldsymbol{w}_p)} \right\} \tag{4.1}$$

The maintenance of auxiliary populations is performed in line 15 of Algorithm 1. After finding j, we can compare $\boldsymbol{y}_i$ and $\boldsymbol{x}_j$ with respect to g^k. If $g^k(\boldsymbol{y}_i, \boldsymbol{w}_j) < g^k(\boldsymbol{x}_j, \boldsymbol{w}_j)$, then $\boldsymbol{y}_i$ replaces $\boldsymbol{x}_j$, denoted as $P^k[j] = \boldsymbol{y}_i$. Otherwise, the solution $\boldsymbol{x}_j$ is preserved. This procedure is performed for each scalarizing function.

The proposed framework can explore l different decomposition schemes for survival selection. Although the computational burden increases by a factor of l, ranking solutions using a scalarizing function is computationally more efficient than other approaches, such as using the hypervolume indicator [34]. A second advantage of this approach is that the number of function evaluations does not depend on the number of scalarizing functions.



P	$h^{(0,9)}$	$h^{(2,6)}$	$h^{(4,3)}$	$h^{(6,2)}$	$h^{(9,0)}$	DCI(P)
A	1	1	1	$1/3$	$2/3$	0.8
B	0	$1/3$	$2/3$	1	1	0.6
C	0	0	0	0	0	0.0

Figure 4.2: DCI calculation. The approximations A, B, and C are placed into the grid. Hyperboxes with non-dominated solutions are highlighted in gray. For a given approximation, the distances from its solutions to these hyperboxes are employed to determine its diversity. If a solution is located within the neighborhood of a hyperbox h, the contribution degree will be greater than zero.

4.1.2 Performance Indicator

In this document, we employ the diversity comparison indicator (DCI) [50] to evaluate the performance of the scalarizing functions. Generally, in order to determine the diversity of a set of non-dominated solutions, a performance indicator only considers the solutions in such a set. On the other hand, DCI assesses the relative diversity among several sets of non-dominated solutions rather than providing an absolute measure of distribution for a single solution set [50]. This indicator relies on an m-dimensional grid to measure the relative diversity among a set of Pareto front approximations. All the considered approximations are placed into the grid. Hyperboxes containing at least one non-dominated solution are employed as a reference to determine the diversity of an approximation. To this end, we consider the distance from the solutions to each reference hyperbox within a given threshold. Such a distance is called contribution degree. The DCI value is a number in the range [0, 1] and is defined as the average contribution degree between an approximation and the reference

hyperboxes. Higher values are preferred. A solution set with the best DCI value implies that its coverage of the reference hyperboxes is better with respect to the other solutions sets involved in the diversity assessment.

The outcome of this indicator depends on both the extension and the location of the grid. It does not need to envelop the whole objective space but cover a region of interest determined by the user. If this region is unknown beforehand, it can be delimited using both the ideal and Nadir points estimated from all the solution sets involved in the procedure. These points are determined as follows:

$$ub_i = z_i^{nad} + \frac{z_i^{nad} - z_i^*}{2 \times div} \tag{4.2}$$

$$lb_i = z_i^* \tag{4.3}$$

where ub_i and lb_i are the upper and lower bounds of the grid in the i-th dimension, respectively, and div is the number of divisions in each dimension. Below, we introduce some concepts required in the DCI procedure.

Definition 9 (Hyperbox size)**.** The hyperbox size in the i-th objective is defined as:

$$d_i = \frac{ub_i - lb_i}{div} \tag{4.4}$$

Definition 10 (Grid location)**.** Given a solution p, its grid location $G(p)$ is defined as:

$$G_i(p) = \lfloor (f_i(p) - lb_i)/d_i \rfloor \tag{4.5}$$

where $f_i(p)$ is the value of the i-th objective function and $G_i(p)$ is the grid coordinate of p in the i-th objective, for $i = 1, 2, \ldots, m$. The range of $G_i(p)$ is $[0, div - 1]$.

Definition 11 (Grid distance between hyperboxes)**.** The grid distance between two hyperboxes h^1

and h^2 is defined as:

$$GD(h^1, h^2) = \sqrt{\sum_{i=1}^{m} (h_i^1 - h_i^2)^2} \tag{4.6}$$

where h_i^1 and h_i^2 are the grid coordinates of h^1 and h^2 in the i-th objective, respectively.

Definition 12 (Distance from an approximation to an hyperbox)**.** Given an approximation P and an hyperbox h, the distance between them is defined as:

$$D(P, h) = \min_{p \in P} \{GD(h, G(p))\} \tag{4.7}$$

That is, the distance between P and h is the shortest grid distance to h among all the hyperboxes occupied by the solutions in P.

Definition 13 (Contribution degree)**.** The contribution degree of an approximation P to a hyperbox h is defined as:

$$CD(P, h) = \begin{cases} 1 - D(P, h)^2/(m+1), & D(P, h) < t \\ 0, & D(P, h) \geq t \end{cases} \tag{4.8}$$

Given an approximation P and a hyperbox h, the contribution degree $CD(P, h)$ denotes the closeness between P and h. A higher value is preferred. It is determined by the distance between P and h with respect to a threshold t. Such a threshold is defined as $t = \sqrt{m+1}$ in [50]. This threshold creates a neighborhood around the hyperbox h. If a solution of P is located in the hyperbox h, then the contribution degree will achieve the highest value. On the other hand, if the distance between P and h is higher than the threshold t (i.e., there is no solution of P within the neighborhood of h), the contribution degree will be zero.

The DCI procedure is given in Algorithm 2. Consider the following example. Figure 4.2 shows the DCI calculation for three approximations A, B, and C. The lower part of this figure shows the individual contribution to each reference hyperbox. The diversity value of an approximation is the mean contribution degree to all the reference hyperboxes. In this case, A has the best (highest)

Algoritmo 2 Diversity Comparison Indicator

1: Construct the grid using Eqs. (4.2)-(4.5).
2: Put approximations $P_1, P_2, \ldots, P_l$ into the grid.
3: Find non-dominated solutions from $P_1 \cup P_2 \ldots \cup P_l$.
4: Define the hyperboxes $h_1, h_2, \ldots, h_s$ occupied by non-dominated solutions.
5: **for** $h \in \{h_1, h_2, \ldots, h_s\}$ **do**
6: **for** $P \in \{P_1, P_2, \ldots, P_l\}$ **do**
7: Compute contribution degree $CD(P, h)$.
8: **end for**
9: **end for**
10: **for** $P \in \{P_1, P_2, \ldots, P_l\}$ **do**
11: DCI(P) $= \frac{1}{s} \sum_{i=1}^{s} CD(P, h_i)$
12: **end for**
13: **return** DCI(P_1), DCI(P_2), ..., DCI(P_l)

diversity value, while the diversity value of C is the lowest. Intuitively, a high DCI value means that a given approximation covers a large number of hyperboxes. From the solutions a, b, and c in Fig. 4.2, we can see that the contribution degree depends on the grid location of these solutions with respect to the reference hyperbox h. The best contribution degree is achieved when a solution is located in the reference hyperbox (see solution a). If the neighborhood of a reference hyperbox is not occupied by any solution of an approximation, then its contribution degree will be zero (see solution c of approximation C in Fig. 4.2).

4.1.3 Performance Assessment

The performance of each scalarizing function is determined by the individuals identified during survival selection. Such selection is stored in auxiliary populations $\{P^1, P^2, \ldots, P^l\}$, where P^k contains the solution set defined by the k-th scalarizing function. The performance of the scalarizing function g^k is given as follows:

$$c_k = \text{DCI}(P^k) \tag{4.9}$$

After analyzing several performance indicators in our preliminary experiments, we decided to adopt DCI to determine the efficiency of each scalarizing function (Algorithm 1, line 20). It is worth noting

that we can employ a different performance indicator by modifying this step in Algorithm 1. It should also be noted that Eq. (4.9) gives us a way of selecting an appropriate scalarizing function at a specific time step of the search process. Nevertheless, the outcome of the adopted performance indicator may be misleading in some generations due to external factors (such as the influence of dominance resistant solutions, detailed in the next section). Instead of relying on the performance assessment at a single time step, we examine a larger time frame. In this document, we track the progress of a scalarizing function by considering the mean of the last b times that its performance has been evaluated. Given this, the mean performance s_k of the scalarizing function g^k is defined as follows:

$$s_k = \frac{1}{b} \sum_{j=1}^{b} c_{k,j} \tag{4.10}$$

The length of the time frame is defined as $b = 10$ in this document. When the time frame is full, the algorithm removes older entries (Algorithm 1, line 21). Higher DCI values are preferred. Thus, the scalarizing function with the highest mean performance is considered the best alternative in our framework:

$$r = \underset{k \in \{1,\dots,l\}}{\arg\max} \{s_k\} \tag{4.11}$$

After this procedure, the main population P is replaced (Algorithm 1, line 25). In this way, we preserve the solution set selected by the best scalarizing function for the next generation. The proposed procedure for performance assessment can establish which scalarizing function must be employed in the current stage. As shown in Algorithm 1, this procedure is conducted at $t = 2$ for the first stage, and when t is multiple of 25 for subsequent stages.

The following example illustrates the proposed method to compute the mean performance

considering three scalarizing functions:

$$(c_{k,j}) = \begin{array}{c} \begin{array}{ccc} \mathsf{DCI}(P^1) & \mathsf{DCI}(P^2) & \mathsf{DCI}(P^3) \end{array} \\ \left(\begin{array}{ccc} 1.0 & 0.8 & 0.5 \\ 0.8 & 0.6 & 0.5 \\ 0.0 & 0.0 & 0.0 \\ \vdots & \vdots & \vdots \\ 0.0 & 0.0 & 0.0 \end{array} \right) \end{array} \begin{array}{l} t = 2 \\ t = 25 \\ t = 50 \\ \vdots \\ t = 225 \end{array}$$

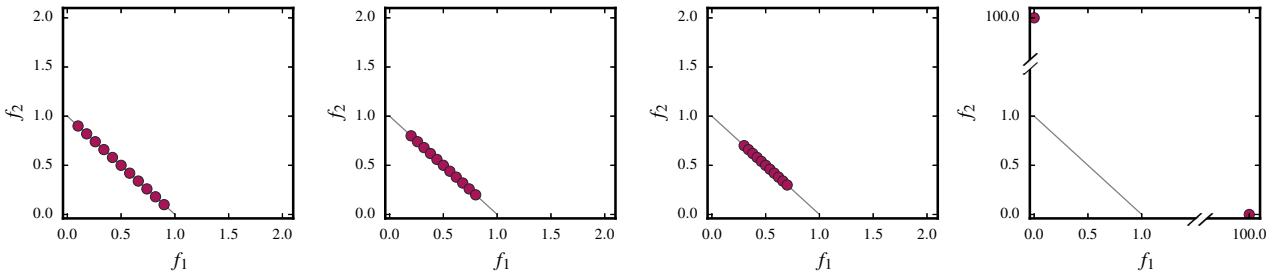
s_1	s_2	s_3
0.18	0.14	0.10

In this example, $(c_{k,j})$ contains the DCI values of each scalarizing function until generation $t = 25$. The remaining values are not obtained yet. In order to determine the scalarizing function for the next stage (from $t = 25$ to $t = 50$), we compute the mean of the DCI values in the matrix. The mean performance is shown below each column. By considering the mean of these values, the negative influence of outliers can be reduced. In this case, the scalarizing function corresponding to P^1 has the best mean performance and it is selected for the next stage.

4.1.4 Dealing with Dominance Resistant Solutions

An important issue in the use of DCI as a performance indicator is to define the boundaries of the grid properly. If the grid does not reflect the region of interest of the user, then the outcome of the indicator may be misleading.

Consider Fig. 4.3. In this case, we compare four approximations using different specifications of the grid. The table below shows the DCI values for each approximation using three settings for the



(a) *A* [0.1, 0.9] (b) *B* [0.2, 0.8] (c) *C* [0.3, 0.7] (d) *D* [0.0, 100.0]

Figure 4.3: Influence of DRS. Solutions in approximations A, B, and C are close to the region of interest. Although solutions in approximation D are non-dominated, they are far away from the region of interest. If they are not handled properly, these solutions can negatively impact the estimation of the Nadir point.

Nadir point ($div = 19$ in this example):

DCI(P)	$z_i^{nad} = 1$	$z_i^{nad} = 5$	$z_i^{nad} = 10$	$z_i^{nad} = 100$
DCI(A)	**0.84**	**1.00**	**1.00**	0.33
DCI(B)	0.78	0.90	**1.00**	0.33
DCI(C)	0.62	0.71	**1.00**	0.33
DCI(D)	0.00	0.00	0.00	**0.67**

Notice that as the boundaries of the grid grow relative to the Nadir point, the similarity in the DCI values increases. In other words, the performance indicator has reduced ability to discriminate between approximations if the grid is not defined correctly. In the last case, approximation D is preferable when $\boldsymbol{z}^{nad} = [100, 100]^T$. However, this approximation is far away from the Pareto front. From Fig. 4.3, we can see that this approximation contains only two solutions. Although they are non-dominated, these solutions can negatively impact the estimation of the Nadir point. In Section 4.1.2, we describe how to specify the boundaries of the grid using the Nadir point $\boldsymbol{z}^{nad}$. A reliable estimation of this point is therefore required to use DCI effectively.

From our preliminary analysis, we noticed that the proposed method coupled with DCI was unable

to identify a suitable scalarizing function due to the presence of dominance resistant solutions (DRS). Solutions of this type can be considered outliers. Although they may have near optimal values in most of the objectives, one or more objectives have a significantly poor value [56]. From early experiments, we observed that the Nadir point is estimated incorrectly when the population contains DRS, as for approximation D in Fig. 4.3. Thus, the specification of the grid required for the DCI procedure may be larger than the real region of interest. As a result, it will be harder to determine which scalarizing function in the set is suitable for the problem at hand according to DCI.

To address this limitation, we propose a mechanism to reduce the negative influence of DRS. The proposed method estimates the Nadir point using the auxiliary populations. We obtain a candidate Nadir point from each auxiliary population. Then, we analyze the candidate points to determine a midpoint. This midpoint is employed to truncate the objective values of potential DRS in the population. The proposed method is detailed in Algorithm 3, and is used as a preprocessing step to handle DRS in the population. Following this, we use the filtered non-dominated solutions, stored in R, to estimate the Nadir point using the procedure proposed by Blank and Deb [5]. Given a set of non-dominated solutions, this procedure constructs a hyperplane using the extremes of the non-dominated solutions. The intercepts of this hyperplane with the objective axes are then employed to define the Nadir point.

Consider Fig. 4.3. In this case, there are four approximations, A, B, C, and D, each with a particular range. We first obtain a candidate Nadir point from each approximation. Then, we compute the mean of the minimum and maximum values of each dimension of these candidate points as follows:

Algoritmo 3 Nadir Point Estimation

Initialize set $R = \{\}$
Sort auxiliary populations $P^1, \ldots, P^l$ into s fronts using non-dominated sorting:
    $F_1, F_2, \ldots, F_s = \text{NDS}(P^1 \cup P^2 \cup \ldots \cup P^l)$
Obtain candidate points $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^l$:
for $k = 1, 2, \ldots, l$ **do**
    $Q = \{\boldsymbol{x} \,|\, \boldsymbol{x} \in P^k \text{ and } x \in F_1\}$
    $z_i^k = \max_{\boldsymbol{x} \in Q} \{f_i(\boldsymbol{x})\}$, for $i = 1, 2, \ldots, m$
end for
Obtain minimum and maximum values from candidate points, for $k = 1, 2, \ldots, l$ and $i = 1, 2, \ldots, m$:
    $a_i = \min\{z_i^k\}$
    $b_i = \max\{z_i^k\}$
Define midpoint:
    $p_i = \frac{1}{2}(a_i + b_i)$, for $i = 1, 2, \ldots, m$
Filter potential DRS:
for $k = 1, 2, \ldots, l$ **do**
    $Q = \{\boldsymbol{f}(\boldsymbol{x}) \,|\, \boldsymbol{x} \in P^k \text{ and } \boldsymbol{x} \in F_1\}$
    for $\boldsymbol{u} \in Q$ **do**
        if $u_i > p_i$ **then**
            $u_i = p_i$ ▷ Truncate DRS
        end if
    end for
    $R = R \cup Q$
end for
Estimate Nadir point using a hyperplane from R.

	f_1	f_2
$\boldsymbol{z}^A$	0.90	0.90
$\boldsymbol{z}^B$	0.80	0.80
$\boldsymbol{z}^C$	**0.70**	**0.70**
$\boldsymbol{z}^D$	**100.00**	**100.00**
Midpoint	50.35	50.35

The midpoint is employed to truncate the objective values of potential DRS in the population. In

this case, only the solutions of approximation D are modified. The resulting non-dominated solutions in R are then employed to estimate the Nadir point. According to our previous example, the DCI value of approximation D goes from 0.67 to 0 when the Nadir point is defined as $\boldsymbol{z}^{nad} = [50.35, 50.35]^T$. In other words, the influence of the DRS in D is reduced, meaning that the other three approximations, A, B, and C, are preferable to D in terms of this indicator.

4.2 Experimental Design

This section describes the experimental design employed to evaluate the performance of the proposed solution. To facilitate further discussion, we organize the experimental setup as follows:

- *Analysis of sensitivity*. The proposed method employs DCI as a mean to estimate the performance of each scalarizing function. The behavior of DCI depends on the parameter div. According to the authors [50], this parameter must be configured according to the number of objectives. In this experiment, we analyze the influence of the parameter div on the performance of MSS using the following values: $div \in \{5, 6, 8, 10, 12, 14, 16, 18, 19, 20\}$. This range comprises the parameter settings analyzed by Li et al. [50] for $m = 3, 5, 8, 10$. The aim of this experiment is to determine whether there is a relationship between div and m. If so, this could ease the configuration of MSS.

- *Complementary search*. This experiment highlights the importance of the auxiliary populations in the proposed method. Here, we compare the single use of a given scalarizing function with the behavior of the same function in a collaborative way.

- *Handling DRS*. Suitable treatment handling of this type of solution is crucial to assess the performance of scalarizing functions. In this experiment, we study three approaches to deal with DRS. As discussed below, the behavior of MSS depends on the way the DRS are tackled.

Table 4.1: Parameter settings.

Objectives	H	Population Size	Generations
3	12	91	300
5	6	210	500
8	3	120	800
10	3	220	1000

- *Influence of the selection mechanism*. In this experiment, we replace the adaptive selection mechanism based on DCI with a fixed selection mechanism. To ease discussion, these approaches are referred to as MSS-A and MSS-F, respectively. In MSS-F, the performance of each scalarizing function in the set is not required. Instead, a predefined scalarizing function is chosen. Notice that the scalarizing functions in the set are still used collaboratively. Although, instead of identifying the best of them, we use a predefined alternative to update the main population.

As testbed problems, we adopted two benchmarks from the literature: DTLZ [17] and MaF [10] for $m = 3, 5, 8, 10$ objectives. Table 4.1 shows the computational budget and population size defined for each problem according to the number of objectives. The number of decision variables is $n = k+m-1$, where $k = 5$ for DTLZ1 and inv-DTLZ1, and $k = 10$ for DTLZ2 and DTLZ3 [15]. In the case of MaF, the number of decision variables is $n = k + m - 1$, where $k = 10$. Each algorithm was evaluated using 50 independent runs for every test instance. The number of reference vectors (N) depends on both the number of divisions (H) and the number of objectives (m). Simulated binary crossover (SBX) and polynomial mutation [16] are employed as variation operators. The distribution indices for crossover and mutation are $\eta_c = 20$ and $\eta_m = 15$, respectively. The crossover rate is $p_c = 1.0$ and the mutation rate is $p_m = 1/n$. The size of the neighborhood for MOEA/D is defined as $T = 20$ [83]. Finally, we use the IGD as performance indicator.

Table 4.2: Best values for parameter div.

Problem	Number of objectives			
	3	5	8	10
DTLZ1	20	5	6	6
DTLZ2	20	14	6	19
DTLZ3	10	20	19	5
inv-DTLZ1	19	10	12	12
MaF3	18	16	6	8
MaF4	19	6	5	20

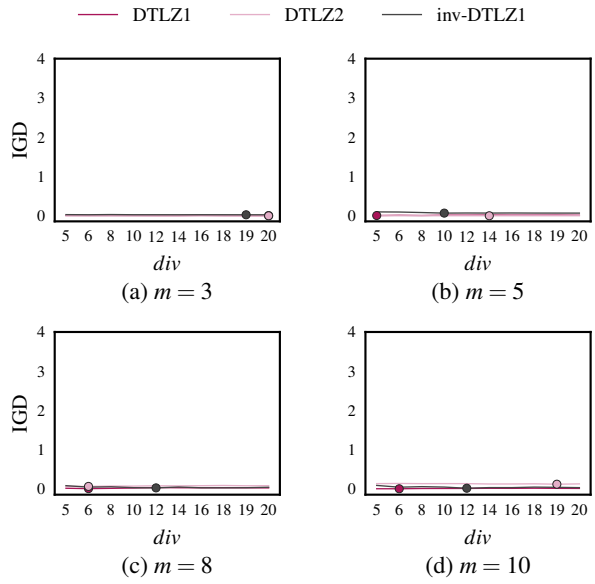


Figure 4.4: Stable behavior. The performance of the proposed method is almost the same regardless of the configuration of div.

4.3 Experimental Results

4.3.1 Analysis of Sensitivity

In this section, we evaluate the behavior of MSS with respect to the parameter div. We also consider the effect of normalizing the objective vectors. In this experiment, we want to determine: (i) the influence of the parameter div, (ii) whether the normalization impacts on the performance of MSS, and (iii) whether there is a relationship between div and m.

Table 4.2 shows the best values of the parameter div for each problem according to the number of objectives. From this table, we can see that the relationship between the parameters div and m is not evident. To aid our analysis, we show the performance of the proposed approach with regard to IGD. Figures 4.4-4.6 show the mean IGD obtained by MSS with normalization with respect to a range of values for div. The dot on each plot represents the best configuration of this parameter for each test instance. From these figures, we notice that the behavior of MSS depends on the specification of div. Here, we identify three types of behavior:

- Stable: From Fig. 4.4, it can be seen that the behavior of the proposed method does not change significantly. That is, we can expect the same performance regardless of the setting of div in DTLZ1, DTLZ2, and inv-DTLZ1.

- Semi-stable: This behavior is seen only in DTLZ3, as shown in Fig. 4.5. Although we observe that there is not evident pattern for $m = 3$ objectives, a higher value of this parameter may be more suitable for $m = 5$ and 8. Finally, it seems that the influence of div is not significant for $m = 10$ objectives.

- Unclear: The relationship between m and div is not noticeable in MaF3 and MaF4. From Fig. 4.6 (a) and (b), it can be seen that the performance of MSS improves dramatically for $div \geq 8$ in MaF3 for $m = 3, 5$ objectives. Although, a higher value for div is not recommended for $m = 8$. In the case of MaF4, the behavior of MSS does not vary significantly for $m = 3, 5, 8$. However, the configuration of this parameter is important in this problem for $m = 10$.

From our results, we conclude that the performance of the proposed method is (to some extent) determined by the configuration of div, specifically in DTLZ3, MaF3, and MaF4. As discussed below, these problems are challenging due to their characteristics, such as multimodality, badly-scaled objectives, and the shape of their Pareto fronts. In the rest of the paper, we compare the performance of the proposed method using $div = 8$.

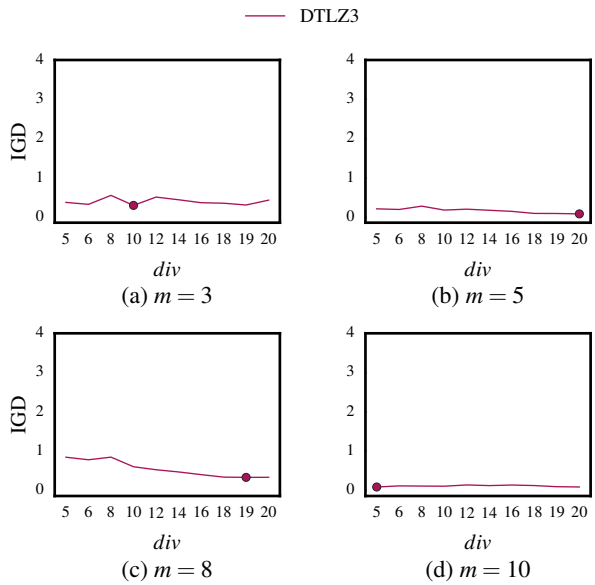


Figure 4.5: Semi-stable behavior. The performance of the proposed method slightly varies with regard to the parameter div.

In the following, we examine the importance of using normalization. In this paper, we normalize the objective vectors using this procedure [83]:

$$\hat{f}_i(\boldsymbol{x}) = \frac{f_i(\boldsymbol{x}) - z_i^*}{z_i^{nad} - z_i^*}, \text{ for } i = 1, 2, \ldots, m \tag{4.12}$$

Figure 4.7 shows the performance of the proposed method and MOEA/D for badly-scaled problems. The aim of this figure is to highlight the importance of the normalization procedure. In Fig. 4.7 (a), the performance of MOEA/D and MSS without normalization is shown on the left of each plot, and with normalization on the right. As can be seen, both approaches perform better when normalization is used. Our results show that TE is a suitable scalarizing function for this problem. We also notice that the behavior of MSS is similar to MOEA/D with TE. That is, the proposed method is able to use the set of scalarizing functions effectively in MaF4. Similar results were observed for MaF5.

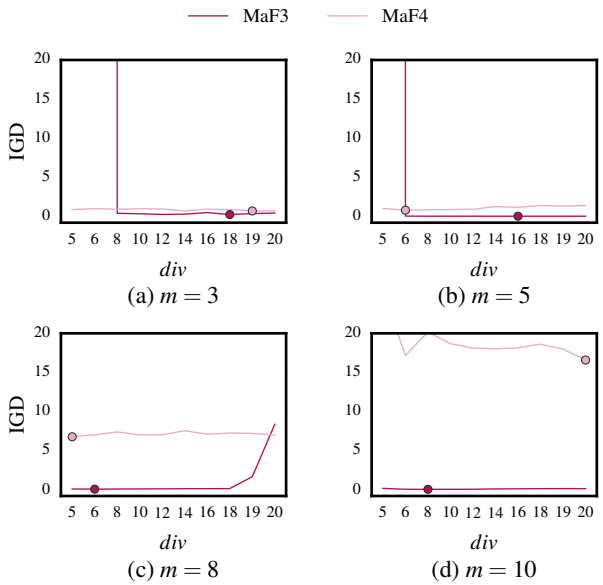


Figure 4.6: Unclear behavior. The relationship between m and div is not evident in MaF3 and MaF4. A proper configuration of div is therefore required to achieve good performance in these problems.

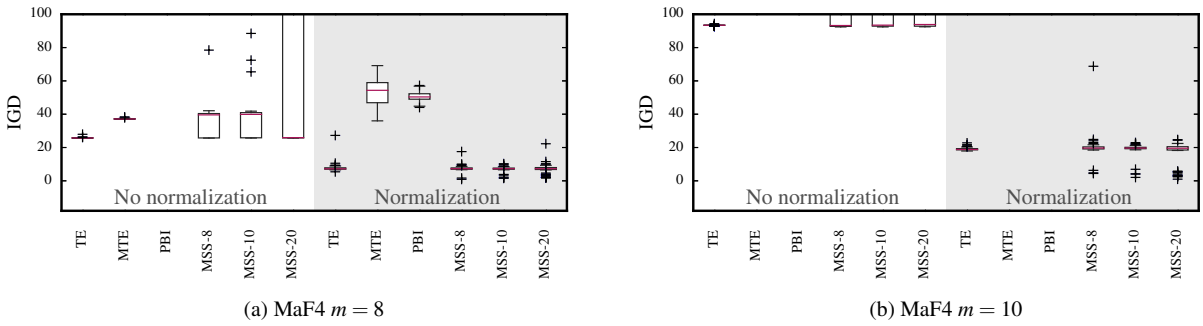


Figure 4.7: Use of normalization. The performance of MOEA/D and MSS without normalization is shown on the left of each plot, and with normalization on the right.

4.3.2 Complementary Search

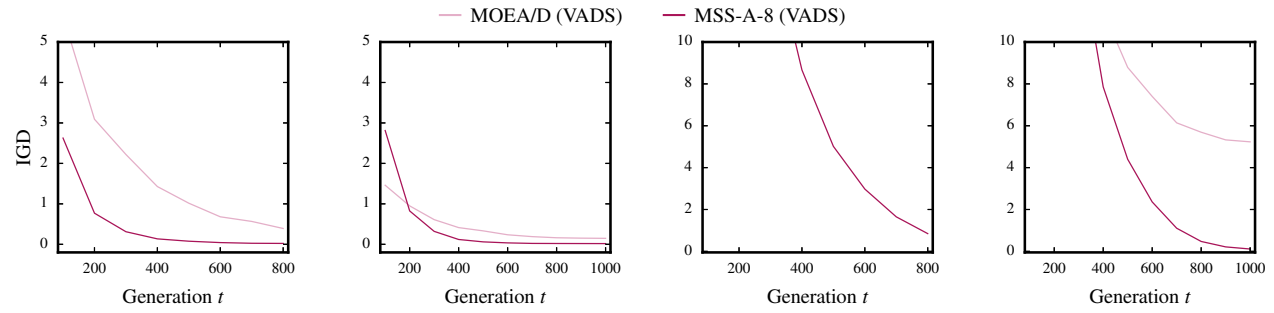
In this section, we examine the collaborative use of scalarizing functions. Specifically, we would like to know how this approach can help to improve the search behavior of a given scalarizing function independently employed.

Figure 4.8 illustrates the complementary search of the scalarizing functions employed in MSS. We adopt MOEA/D with VADS as the baseline. We also show the search process of the corresponding

Table 4.3: Mean and standard deviation of IGD values over 50 runs for each test instance. The best mean value for each test problem is shown in bold. The best three results for each test problem are shaded.

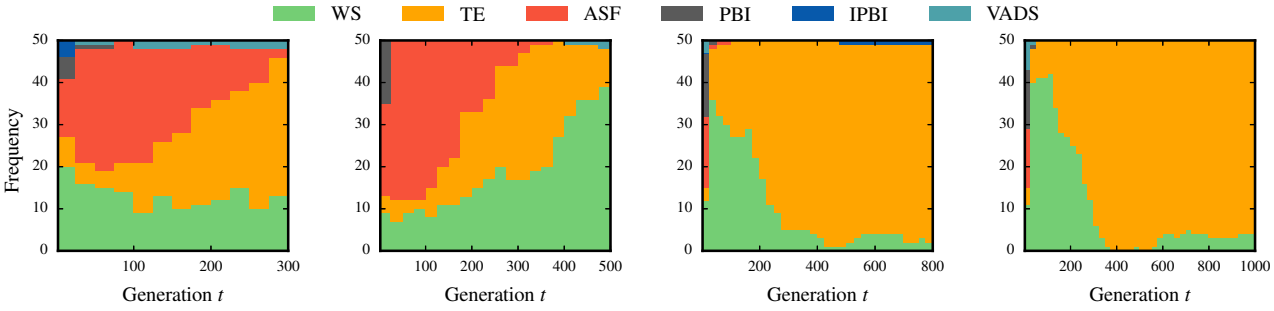
Problem	m	WS	TE	ASF	PBI	VADS	MSS-DCI
DTLZ1	3	0.251 (0.041)	0.049 (0.079)	0.036 (0.063)	0.206 (0.194)	**0.006** (0.001)	0.009 (0.006)
	5	0.320 (0.070)	0.156 (0.105)	0.106 (0.096)	0.462 (0.087)	**0.007** (0.002)	0.009 (0.006)
	8	0.439 (0.043)	0.344 (0.049)	0.538 (0.021)	0.535 (0.029)	0.391 (0.389)	**0.021** (0.026)
	10	0.476 (0.039)	0.371 (0.033)	0.532 (0.046)	0.546 (0.020)	0.147 (0.075)	**0.017** (0.017)
DTLZ2	3	0.456 (0.030)	0.076 (0.001)	0.010 (0.008)	**0.001** (0.000)	0.001 (0.000)	0.002 (0.002)
	5	0.595 (0.090)	0.295 (0.000)	0.107 (0.042)	**0.001** (0.000)	0.001 (0.000)	0.003 (0.004)
	8	0.931 (0.065)	0.795 (0.072)	1.124 (0.289)	1.214 (0.173)	**0.001** (0.000)	0.079 (0.037)
	10	1.006 (0.050)	0.910 (0.048)	1.273 (0.000)	1.273 (0.000)	**0.000** (0.000)	0.141 (0.027)
DTLZ3	3	0.595 (0.220)	**0.148** (0.205)	0.223 (0.265)	0.748 (0.381)	0.903 (0.932)	0.555 (0.673)
	5	0.748 (0.140)	0.428 (0.221)	0.378 (0.268)	1.119 (0.035)	1.587 (1.538)	**0.281** (0.524)
	8	1.032 (0.074)	0.945 (0.118)	1.240 (0.001)	1.240 (0.001)	13.665 (6.101)	**0.849** (0.813)
	10	1.097 (0.059)	1.061 (0.078)	1.273 (0.000)	1.273 (0.000)	5.232 (3.640)	**0.112** (0.112)
inv-DTLZ1	3	0.225 (0.046)	**0.029** (0.004)	0.034 (0.001)	0.063 (0.110)	0.033 (0.008)	0.033 (0.006)
	5	0.254 (0.014)	**0.069** (0.005)	0.112 (0.000)	0.131 (0.146)	0.096 (0.022)	0.084 (0.019)
	8	0.278 (0.011)	**0.004** (0.004)	0.246 (0.008)	0.316 (0.133)	0.237 (0.001)	0.061 (0.080)
	10	0.283 (0.021)	**0.002** (0.002)	0.280 (0.005)	0.312 (0.111)	0.257 (0.001)	0.064 (0.088)
MaF1	3	0.436 (0.027)	**0.051** (0.001)	0.066 (0.000)	0.060 (0.000)	0.062 (0.001)	0.052 (0.003)
	5	0.522 (0.029)	0.134 (0.001)	0.226 (0.001)	0.142 (0.002)	0.216 (0.030)	**0.127** (0.005)
	8	0.571 (0.019)	**0.004** (0.003)	0.504 (0.022)	0.485 (0.085)	0.470 (0.003)	0.079 (0.149)
	10	0.599 (0.026)	**0.003** (0.002)	0.571 (0.017)	0.521 (0.001)	0.516 (0.003)	0.028 (0.066)
MaF2	3	0.253 (0.000)	0.041 (0.001)	**0.038** (0.000)	0.039 (0.001)	0.038 (0.000)	0.039 (0.000)
	5	0.300 (0.000)	0.146 (0.002)	0.163 (0.004)	**0.103** (0.001)	0.110 (0.001)	0.108 (0.002)
	8	0.525 (0.000)	0.474 (0.030)	0.837 (0.062)	0.848 (0.000)	**0.170** (0.001)	0.285 (0.016)
	10	0.585 (0.000)	0.542 (0.021)	0.870 (0.001)	0.870 (0.000)	**0.171** (0.001)	0.231 (0.079)
MaF3	3	0.287 (0.200)	**0.218** (0.252)	0.463 (0.870)	0.921 (0.511)	83.120 (264.735)	1.117 (2.209)
	5	0.316 (0.154)	0.334 (0.131)	0.318 (0.130)	0.909 (0.178)	0.154 (0.157)	**0.075** (0.003)
	8	0.537 (0.127)	0.476 (0.119)	1.018 (0.003)	1.028 (0.005)	0.305 (0.214)	**0.131** (0.044)
	10	0.566 (0.098)	0.512 (0.096)	1.019 (0.001)	1.026 (0.003)	0.589 (0.253)	**0.097** (0.040)
MaF4	3	**0.252** (0.307)	0.512 (0.386)	0.631 (0.504)	0.485 (0.114)	1.490 (1.713)	1.220 (1.551)
	5	**0.329** (0.404)	1.376 (0.193)	3.827 (0.027)	3.013 (0.183)	2.451 (0.036)	0.886 (0.587)
	8	**1.240** (0.681)	7.859 (2.897)	53.893 (8.238)	50.730 (2.838)	32.193 (1.584)	7.427 (2.012)
	10	**1.459** (0.888)	19.136 (0.930)	227.282 (19.780)	198.425 (6.609)	123.725 (3.098)	20.176 (7.902)
MaF5	3	4.274 (1.089)	3.587 (1.730)	3.380 (1.855)	4.394 (1.383)	0.853 (1.823)	**0.641** (1.080)
	5	12.828 (5.776)	8.578 (5.714)	6.863 (4.668)	19.230 (8.662)	**0.018** (0.002)	0.394 (0.382)
	8	106.230 (68.313)	82.731 (74.824)	98.050 (72.218)	211.030 (24.849)	21.599 (64.326)	**7.039** (3.271)
	10	239.321 (141.713)	173.832 (127.675)	227.300 (121.720)	886.722 (0.002)	35.547 (173.745)	**17.816** (6.718)

auxiliary population in the proposed method. This figure shows the mean IGD value of both approaches over 50 independent runs. From Fig. 4.8 (a) and (b), it can be seen that the convergence of the auxiliary population is slightly better than the baseline in DTLZ1. This improvement is more noticeable in DTLZ3, as shown in Fig. 4.8 (c) and (d). Since the other auxiliary populations address the same problem collaboratively, they can share good solutions among them. As a result, the auxiliary population using VADS can benefit from the solutions found by other scalarizing functions.



(a) DTLZ1 $m = 8$ (b) DTLZ1 $m = 10$ (c) DTLZ3 $m = 8$ (d) DTLZ3 $m = 10$

Figure 4.8: Complementary search. The convergence of the auxiliary population using VADS is better than MOEA/D with VADS. That is, VADS can benefit from solutions found by other auxiliary populations in MSS.



(a) MaF4 $m = 3$ (b) MaF4 $m = 5$ (c) MaF4 $m = 8$ (d) MaF4 $m = 10$

Figure 4.9: Overview of the search process. Each plot shows the relevance of the scalarizing functions in the proposed method.

Below, we examine the performance of the proposed approach with respect to MOEA/D. To this end, we have employed six scalarizing functions in MSS. The results are given in Table 4.3. This table shows the mean and standard deviation of IGD after 50 independent runs. From this table, we can see that the performance of MOEA/D varies according to the scalarizing function used. For instance, TE is a suitable option for inv-DTLZ1, while PBI and VADS are preferred for MaF1. In this table, the best three results for each test problem are shaded. As can be seen, the proposed approach is one of the three best methods for most of the test instances. That is, MSS was able to use the set of scalarizing functions effectively in this experiment. From Table 4.3, we can see that the proposed method achieved better results than any of the other scalarizing functions in some test problems, such as DTLZ1, DTLZ3, MaF3, and MaF5.

Observe the performance of the proposed method in MaF4, as given in Table 4.3. We can see that MOEA/D coupled with WS and TE achieved the best performance in this problem. Unlike other test instances, WS showed better performance in MaF4 given the shape of its Pareto front. In this case, the proposed method also showed competitive performance in this problem. Figure 4.9 provides an overview of the search process of the proposed method. Each plot shows the relevance of the scalarizing functions throughout the search. Notice that WS and TE are the most relevant candidate functions at the end of the search. In other words, MSS promotes two of the best scalarizing functions for MaF4. Observe that TE is preferred over WS in some cases at the end of the search. This issue is due to the estimation of the Nadir point and it is futher explained in Section 4.3.5. We observed that the approximation of the Nadir point is slightly inferior when the test problem is badly-scaled, like MaF4.

The results given in this section suggests that the proposed method not only selects a suitable scalarizing function, but also is able to used them collaboratively.

4.3.3 Handling DRS

One of the main challenges in measuring the performance of scalarizing functions is controlling the negative influence of DRS. If this kind of solution is not properly handled during performance assessment, the outcome may be inaccurate. This is because DRS can affect the estimation of the Nadir point used in the normalization and the construction of the grid for the DCI procedure. As a consequence, an unsuitable scalarizing function could be chosen, reducing the progress of the population toward the Pareto front.

In our approach, we need to estimate both $\boldsymbol{z}^*$ and $\boldsymbol{z}^{nad}$ to normalize the objective vectors before computing the performance indicator. To do this, we use the normalization procedure given in Eq. (4.12).

From our experiments, we realized that some of the scalarizing functions were prone to find DRS in some problems, and particularly in MaF3. In this test problem, we faced difficulties in assessing

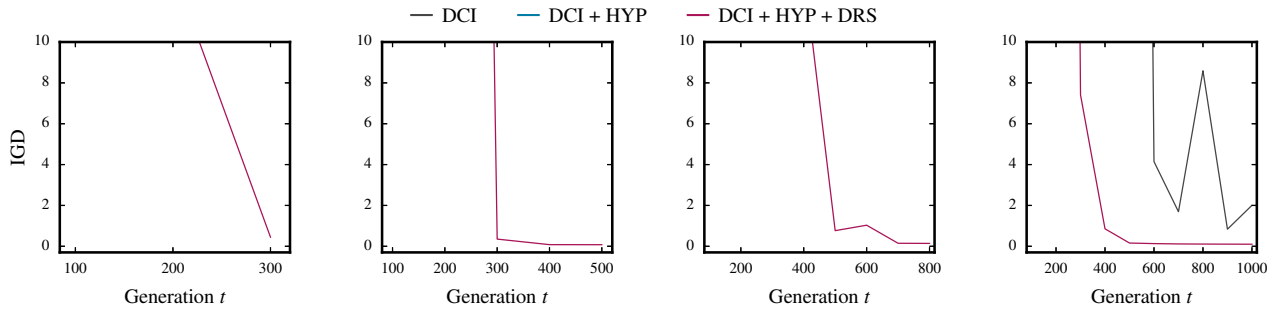
the behavior of each scalarizing function in the set. In order to improve the effectiveness of the proposed method, we systematically evaluate three approaches to estimate the Nadir point from a set of non-dominated solutions. These approaches are described below:

- DCI: In this approach, we estimate both $\boldsymbol{z}^*$ and $\boldsymbol{z}^{nad}$ from a set of non-dominated solutions, as suggested in [28]. We first combine all auxiliary populations into a single merged population. We then identify non-dominated solutions using non-dominated sorting. This procedure can be computationally expensive for many-objective optimization. Thus, we use a parallel method called very fast non-dominated sorting (VFNS) [68] for this purpose. Finally, the ideal and Nadir points are estimated from the obtained non-dominated solutions considering the minimum and maximum values of each objective.

- DCI + HYP: In this case, we use the normalization procedure of NSGA-III proposed by Deb et al. [15, 5] as follows. First, we identify the non-dominated solutions from all the auxiliary populations, as mentioned above. We then construct an m-dimensional hyperplane using m extreme non-dominated solutions. Finally, the intercepts of this hyperplane with respect to the m objective axes are employed to define the Nadir point.

- DCI + HYP + DRS: Unlike the previous approach, we identify and truncate potential DRS before constructing the hyperplane as follows. Given a set of non-dominated solutions, we define a midpoint to filter DRS. The objective vectors outside the region delimited by the midpoint are considered to be DRS. The objective values of these solutions are then truncated using the corresponding objective values of the midpoint. Following this, we use the hyperplane-based normalization procedure of Deb et al. [15, 5].

We evaluate these three approaches using MaF3, and the results are given in Table 4.4 and Fig. 4.10. Table 4.4 shows the mean and standard deviation of the IGD after 50 independent runs. On the other hand, Fig. 4.10 shows the mean IGD value of each approach throughout the search



(a) MaF3 $m = 3$ (b) MaF3 $m = 5$ (c) MaF3 $m = 8$ (d) MaF3 $m = 10$

Figure 4.10: Dealing with DRS. The performance of MSS depends on the way in which DRS are handled.

Table 4.4: Dealing with DRS. Mean and standard deviation of IGD values over 50 runs are shown for each test instance. The best mean value for each test problem is shown in bold.

Problem	m	MSS DCI	MSS DCI HYP	MSS DCI HYP DRS
MaF3	3	55609.911 (226710.843)	243336.828 (1065294.542)	**0.445** (0.920)
	5	2303.078 (6293.951)	22320298105.627 (154475041342.057)	**0.077** (0.012)
	8	116.055 (809.076)	25597106.398 (179179735.447)	**0.140** (0.046)
	10	2.004 (4.241)	905.505 (6337.302)	**0.101** (0.042)

process for $m = 3, 5, 8, 10$ objectives. From this figure, we can see that the proposed approach for handling DRS helps MSS to achieve better convergence.

From the preliminary results, we notice that although the first approach, MSS DCI in Table 4.4, achieved good performance in most of the test problems, it was unable to address MaF3. This is a challenging problem given the shape of its Pareto front, its multimodality, and the existence of DRS. In view of this, we opted to try the normalization procedure of NSGA-III as an alternative method for Nadir point estimation.

Contrary to our intuition, the second approach, MSS DCI+HYP, did not perform well in this experiment. We attribute this issue to the construction of the hyperplane. Under the influence of DRS, the extension of the hyperplane can be enlarged significantly far away from the Pareto front. As a result, the estimation of the Nadir point is affected.

From Table 4.4, it can be seen that neither DCI or DCI+HYP were able to cope with DRS in

MaF3. The high IGD values are caused by the incorrect selection of scalarizing function after the performance assessment. In consequence, the main population of MSS is incapable to attain the Pareto front.

4.3.4 Influence of the Selection Mechanism

We also studied the performance of the proposed method without the adaptive selection mechanism. In this case, there is no need to perform the following tasks in Algorithm 1:

- Performance evaluation of the scalarizing functions in the set (line 18).
- Tracking the progress of each scalarizing function (lines 19 and 20).
- Selection of the best alternative in the set (line 21).

Notice that we still rely on the collaborative use of the scalarizing functions, although the selection is fixed. That is, we select the same predefined scalarizing function. In this way, we can reduce not only the computational load but also the number of parameters required for MSS (i.e., the time frame size b and div). This simplification can reduce the effectiveness of the proposed method. However, we hypothesize that a fixed selection can reduce the computational burden without severely affecting the effectiveness of MSS. As discussed below, the behavior of MSS is not significantly impacted if the predefined scalarizing function is chosen properly.

We evaluated this new approach using TE and ASF. The results are shown in Figs. 4.11 and 4.12. These figures show the mean IGD value of 50 independent runs. From these figures, we can compare:

- The independent use of the predefined scalarizing function in MOEA/D.
- MSS with adaptive selection (MSS-A).
- MSS with fixed selection (MSS-F).

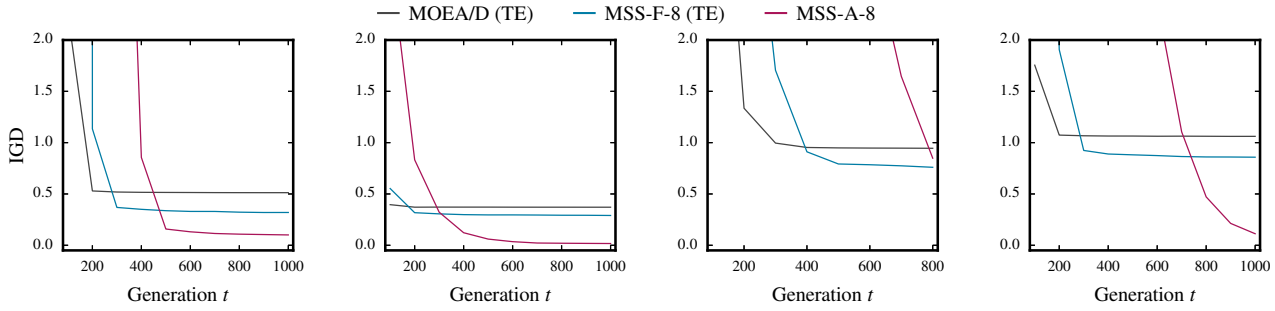
We adopted a $div = 8$ for both configurations of the proposed approach. From Fig. 4.11, we can see that the performance of MSS with fixed selection is comparable to that of MSS with adaptive selection. It can also be noted that the behavior of both settings of the proposed approach outperformed MOEA/D with either TE or ASF.

From Fig. 4.12, we observe that both baselines, MOEA/D and MSS-F, converged faster than MSS-A. Also, notice that MSS-F was faster than MOEA/D. Unlike MOEA/D, MSS-F employs ASF together with the other scalarizing functions in the set. Consequently, it improves the search process compared to that of MOEA/D with ASF alone. It is interesting that both MOEA/D and MSS-F converge quickly and their populations then stagnate, especially when TE is employed (see Fig. 4.11). Although the convergence of the proposed approach MSS-A is not as fast as the baselines, it is less prone to stagnation. This can be explained by the way the main population is updated in MSS-F and MSS-A. In the first case, only the predefined scalarizing function is used to update the main population. On the other hand, in MSS-A, a different scalarizing function can be employed for this purpose.

We observe that the fixed selection mechanism can be useful in some test instances, such as DTLZ3 with $m = 8$ objectives, particularly when ASF is employed. From Fig. 4.12, it seems that MSS-A requires additional effort to reach MSS-F. In this case, updating the main population using a predefined scalarizing function may be better than using a set.

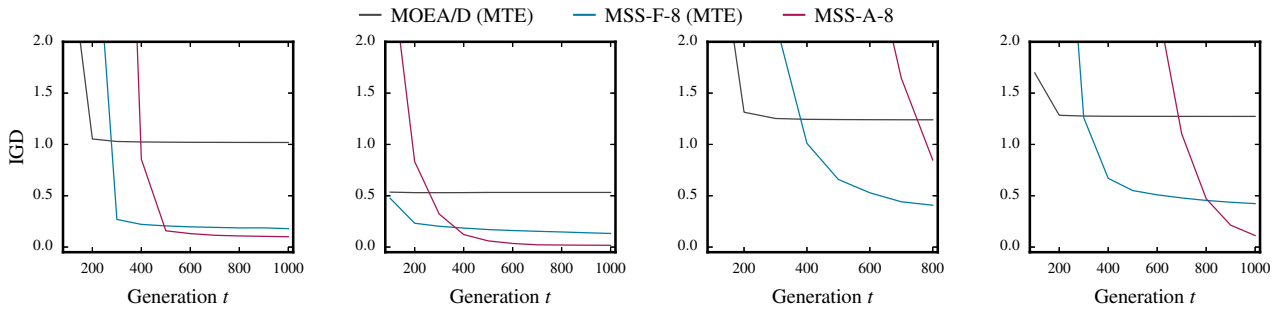
From our experimentation, we note that the behavior of the fixed selection mechanism depends on the predefined scalarizing function. A comparison of Fig.4.11 (d) and Fig. 4.12 (d) shows that the population of MSS-F with ASF was less prone to stagnation than MSS-F with TE. In addition, the performance of ASF was better than that of TE.

In general, MSS with adaptive selection achieves the best results in most of the evaluated test instances. Although, if the computational resources are limited, we can employed MSS with fixed selection and ASF to give comparable results.



(a) MaF3 $m = 10$ (b) DTLZ1 $m = 10$ (c) DTLZ3 $m = 8$ (d) DTLZ3 $m = 10$

Figure 4.11: MSS with adaptive and fixed selection (TE). Both MSS-F and MOEA/D converge faster than MSS-A. Although, MSS-A achieves better performance in most of the evaluated test problems.



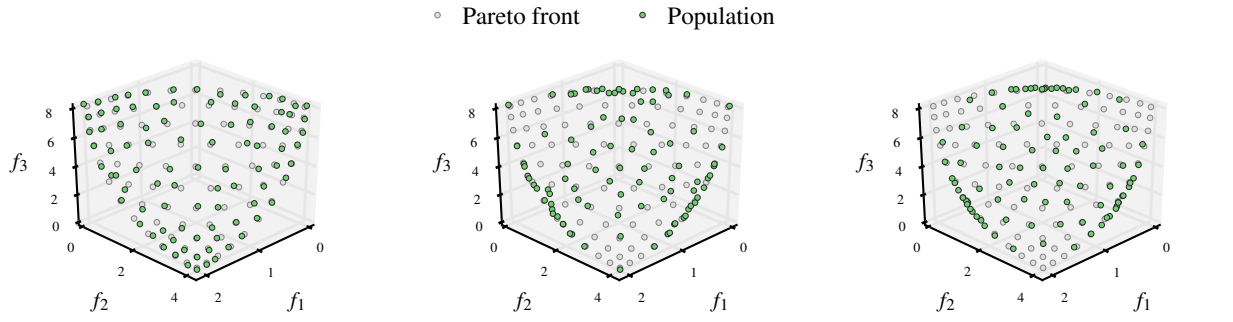
(a) MaF3 $m = 10$ (b) DTLZ1 $m = 10$ (c) DTLZ3 $m = 8$ (d) DTLZ3 $m = 10$

Figure 4.12: MSS with adaptive and fixed selection (ASF). Both MSS-F and MOEA/D converged faster than MSS-A. Although, MSS-A achieved a better performance in most of the evaluated test problems.

4.3.5 About the Nadir Point Estimation

Figure 4.13 shows three of the six solution sets maintained in MSS, one for each scalarizing function. The DCI value of each solution set is also shown below each plot. We noticed that there are some cases when the proposed approach promotes a slightly inferior scalarizing function. In this example, TE has a better DCI value, meaning that it is considered the best candidate function. Although, WS is better distributed than TE. This may seem counterintuitive. However, the reason of this issue is the outcome of the performance indicator and the estimation of the Nadir point.

As mentioned in Section 4.3.3, it is important for the proposed method to obtain a reliable



(a) MaF4, $t = 500$
WS, DCI $= 0.9500$, DCI$^* = 0.9324$

(b) MaF4, $t = 500$
TE, DCI $= 0.9750$, DCI$^* = 0.8851$

(c) MaF4, $t = 500$
VADS, DCI $= 0.9700$, DCI$^* = 0.8829$

Figure 4.13: Solution sets corresponding to WS, TE, and VADS in MaF4. We evaluate each solution set regarding a estimation of the Nadir point, $\boldsymbol{z}'$ (DCI), and the real Nadir point, $\boldsymbol{z}^{nad}$ (DCI *). Notice how the specification of this point influences on the performance assessment.

estimation of the Nadir point since it is needed for the performance assessment (DCI). As a result, we introduced a method for improving the Nadir point estimation when DRS are present. Nevertheless, we noticed that the adopted method for estimating the Nadir point [5] is sensible to the range of the objectives. The solution sets shown in Fig. 4.13 were obtained when solving MaF4 using the proposed MSS. This test problem is badly-scaled, meaning that the objective values are not in the range [0, 1]. In this case, the range of the i-th objective function of MaF4 is $[0, 2^i]$, for $i = 1, 2, \ldots, m$.

In the proposed method, the solutions of the auxiliary populations are processed before the performance assessment. First, we merge all the solutions and keep the non-dominated solution only. Then, we truncate the DRS, if any, and use the adopted method for approximating the Nadir point [5]. We then use the Nadir point estimation for computing the DCI procedure and evaluating the quality of the solution sets. Finally, we select the scalarizing function corresponding to the solution set with the best DCI value.

In this example, the resulting approximation of the Nadir point is:

$$\boldsymbol{z}' = [3.2669, 6.6037, 12.7691]^T \tag{4.13}$$

Another common way to approximate the Nadir point from a set P of non-dominated solutions

is considering the maximum value for each objective vector:

$$z_i = \max \{f_i(\boldsymbol{x})\}, \text{ for } \boldsymbol{x} \in P \tag{4.14}$$

Considering this approach, the resulting approximation of the Nadir point is:

$$\boldsymbol{z}'' = [2.0578.4.0958, 8.2199]^T \tag{4.15}$$

In this case, the real Nadir point of MaF4 is $\boldsymbol{z}^{nad} = [2, 4, 8]^T$. As it can be seen, the second approach provides a better approximation, meaning that the outcome of the DCI procedure may be more reliable when using $\boldsymbol{z}''$. Table 4.5 shows the DCI values of the three solution sets illustrated in Fig. 4.13, considering the two specifications of the Nadir point given in Eqs.(4.13) and (4.15), and the real point. Although its simplicity, the second estimation approach leads to a better performance assessment. As a result, WS has a better performance regarding DCI, as empirically shown in Fig. 4.13. However, we also observed that this simple approximation approach can lead to misleading results in our experiments.

In some cases, the proposed method is unable to select a well-performing scalarizing function, as the case shown in Fig.4.13. However, this limitation is due to the estimation of the Nadir point. This issue does not severely impacts on the behavior of the proposed method throughout the search, since the solution sets of each candidate function are stored in separated populations. If a slightly inferior scalarizing function is selected (as TE in this case), the solutions of other well-performing candidate functions can still be shared among other populations during the search. On the other hand, if this issue happens in the last generation, it might impact the performance of the proposed method. In this document, we measure the performance of MSS regarding only one of the six auxiliary populations, that is, the population with the higher DCI value in the last generation. If a inferior population is selected at the end of the search process, the behavior of MSS may be affected

Table 4.5: DCI values of three solution sets (WS, TE, and VADS) considering three specifications of the Nadir point. Best values are shown in bold.

Point	DCI(WS)	DCI(TE)	DCI(VADS)
z'	0.9500	**0.9750**	0.9700
z''	**0.9324**	0.8851	0.8829
z^{nad}	**0.9237**	0.8856	0.8644

regarding a performance indicator, such as IGD or Hypervolume.

The current configuration of the proposed method relies on a specific Nadir point estimation method [5] and a performance indicator [50]. This setup has shown to be promising in most test problems in our experiments. Although, there are still some corner cases, like the one described in this section, that may affect the behavior of the proposed method.

4.4 Summary

In this chapter, we introduced a framework for the collaborative use of scalarizing functions, called multiple survival selection (MSS). The proposed approach aims to improve the performance of decomposition in many-objective optimization.

Our method consists of two steps: evaluation of scalarizing functions and performance assessment. We first define a set of potential scalarizing functions for solving a given problem. Although, it is not straightforward to determine which candidates are suitable for the problem at hand. We therefore employ a performance indicator to assess the effectiveness of each candidate.

In the proposed method, the performance of each scalarizing function depends on its ability to identify promising solutions, from both the parent and the offspring populations, at the survival selection. The effectiveness of the scalarizing functions is evaluated every 25 generations, meaning that suitable candidate functions are promoted throughout the search process.

The collaborative use of a set of scalarizing functions offers significant advantages. Firstly, the search capability of a particular scalarizing function can be enhanced by other candidates in the

set by sharing good solutions among them. We can also explore several decomposition schemes simultaneously, one for each candidate in the set. Finally, given the design of MSS, the number of function evaluations does not increase with the number of scalarizing functions in the set.

We evaluated the performance of the proposed approach against MOEA/D. That is, we compared the independent use of a scalarizing function using MOEA/D with respect to the collaborative use of multiple scalarizing functions via MSS. We adopted six scalarizing functions from the literature: WS, TE, ASF, PBI, IPBI, and VADS. In addition, we employed two widely-used benchmarks, DTLZ and MaF, considering up to ten objectives. Our experimental results show that:

- As we hypothesized, the independent use of a single scalarizing function can be improved if additional approaches are used in combination. We noticed this specifically using VADS. As outlined in Section 4.3.2, the convergence of MOEA/D-VADS is improved when the other scalarizing functions in the set are employed in MSS.

- The behavior of the proposed approach depends on the indicator adopted for performance assessment. In this paper, we employed DCI to estimate the effectiveness of each scalarizing function in the set. This indicator measures the relative diversity among a set of approximations using an m-dimensional grid. The extension of this grid is controlled by the Nadir point, and the number of hyperboxes is determined by an integer parameter div. We noticed that the specification of this parameter affects the behavior of MSS in some test problems, particularly DTLZ3, MaF3, and MaF4. These problems are challenging given their properties, such as multimodality and the shapes of their Pareto fronts. From our results, we recommend setting this parameter as $div = 8$.

- We highlight the importance of normalization in badly-scaled problems, such as MaF4 and MaF5. We compared the performance of both MOEA/D and MSS with and without normalization. In this experiment, we showed that MOEA/D-TE with normalization achieved good results with respect to IGD in MaF4 and MaF5, whereas the performance of other

scalarizing functions was limited. In addition, we noticed that MSS with normalization behaved similarly to MOEA/D-TE. Hence, the proposed method was able to use the set of scalarizing functions effectively.

- We confirmed the importance of handling dominance resistant solutions (DRS) properly in order to achieve good performance. From preliminary experiments, we discovered that MSS was unable to identify a suitable scalarizing function when DRS are present. This is because these solutions negatively impact the estimation of the Nadir point. In consequence, the construction of the grid for the DCI procedure is affected, altering its outcome. As a result, we developed a mechanism to reduce the influence of DRS in MSS. This mechanism analyses the auxiliary populations and defines a midpoint from them. This midpoint serves as a filter to identify potential DRS. Following this, the objective values of these solutions are replaced by the corresponding objective values of the midpoint. As described in Section 4.3.3, this procedure enhanced the performance of MSS in MaF3.

5

Indicator-based Weight Adaptation

Weight adaptation methods can enhance the diversity of solutions obtained by decomposition-based approaches when addressing irregular Pareto front shapes. Generally, these methods adapt the location of each weight vector during the search process. However, early adaptation could be unnecessary and ineffective because the population does not provide a good Pareto front approximation at early generations. In order to improve its performance, a better approach would be to trigger such adaptation only when the population has reached the Pareto front.

In this chapter, we introduce a performance indicator to assist weight adaptation methods, called the median of dispersion of the population (MDP). The proposed indicator provides a general snapshot of the overall progress of the population toward the Pareto front by analyzing the local progress of each subproblem. When the population becomes stagnant according to the proposed indicator, the adaptation of weight vectors starts. We evaluate the performance of the proposed approach in both regular and irregular test problems. Our experimental results show that the proposed approach triggers the weight adaptation when it is needed. In addition, we also show

that the proposed indicator can be used for detecting a perturbation in the population when the location of the Pareto front is modified. This attribute could be employed to address dynamic problems.

5.1 Description

As shown in previous work, the performance of decomposition-based methods strongly depends on the shape of the Pareto front. When the shape of the Pareto front is regular, many decomposition-based methods are able to find a well-distributed Pareto front approximation. However, when the shape is irregular (e.g., an inverted triangular surface), the diversity of the population is severely affected. This deterioration is due to the reference vectors does not reflect the shape of the Pareto front. As a result, the diversity of the population is reduced.

To address this issue, many authors have proposed techniques to adapt the locations of the reference vectors throughout the search process. An early approach was introduced by Deb and Jain for improving the performance of the NSGA-III in irregular problems [36, 37]. Since then, many other methods have been proposed [52, 37, 9, 70]. Most of those weight adaptation methods share two aspects in common:

- The population is employed for weight adaptation.
- The adaptation of weight vectors is performed continuously throughout the search process.

The population is employed as reference of the Pareto front shape before weight adaptation. However, it is unlikely that the population provides a good approximation of such a shape at early generations. Thus, early weight adaptation could be ineffective. A better approach would be to perform the weight adaptation only when the population has reached the Pareto front, in order to start the redistribution of weight vectors and, therefore, provide a better approximation to the Pareto

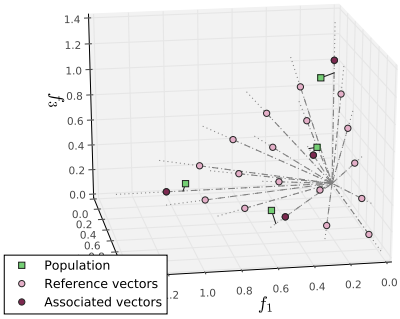


Figure 5.1: Association between members of the population and reference vectors. Each individual is associated with its nearest reference vector according to their perpendicular distance.

front. Such an approach would reduce the computational overhead associated to weight adaptation as it is not required before finding the Pareto front.

In this document, we propose an indicator to assist weight adaptation methods, called the median of dispersion of the population (MDP). This indicator relies on the local progress made for each subproblem after a given interval, also known as relative improvement [85]. By analyzing the local progress of every subproblem, the proposed indicator provides a general snapshot of the global progress of the population toward the Pareto front.

We use the proposed indicator to determine when the population becomes stagnant. We hypothesize that the population will provide a good approximation of the shape of the Pareto front after reaching the stagnation point. This approximation can therefore be used for weight adaptation. As a result, early adaptation can be avoided and the effectiveness of the adaptation can be improved. The proposed method is detailed in Algortihm 4. In the following, we prodive the details of the adaptation strategy.

5.1.1 Density Estimation

Deb and Jain [15] proposed a procedure to associate each individual of the population with its nearest reference vector. This procedure aims to estimate the density of the reference vectors, as illustrated

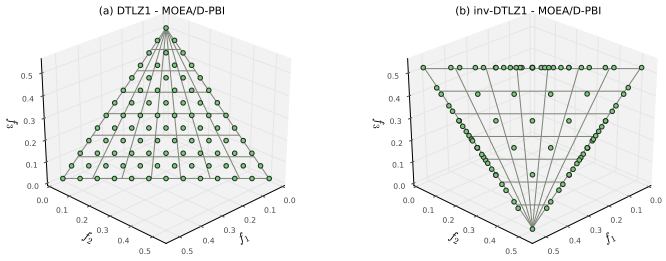


Figure 5.2: Approximations obtained by MOEA/D. (a) In DTLZ1, the surface of the Pareto front is uniformly covered. (b) In inv-DTLZ1, the surface of the Pareto front is not uniformly covered. The difference between DTLZ1 and inv-DTLZ1 is the shape of the Pareto front.

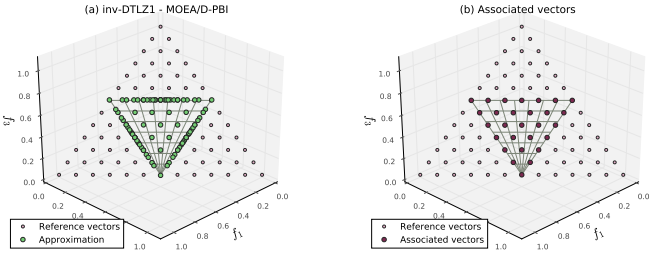


Figure 5.3: Density estimation. (a) Approximation obtained by MOEA/D in inv-DTLZ1. Each solution from this approximation will be associated with its nearest reference vector. (b) The association procedure allows us to identify the reference vectors with more density, that is, those vectors associated with at least one solution. These vectors are candidates to be expanded.

in Fig. 5.1. The larger the number of invididuals associated with a given reference vector, the larger the density of this reference vector is. The individual $\boldsymbol{x}_i$ is associated with $\boldsymbol{w}_j$, denoted as $\pi(i) = j$, as follows:

$$\pi(i) = \arg\min_j \ \left(d^{\perp}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j)\right) \tag{5.1}$$

where $d^{\perp}(\boldsymbol{u}, \boldsymbol{v})$ denotes the perpendicular distance between $\boldsymbol{u}$ and $\boldsymbol{v}$. Thus, $\boldsymbol{x}_i$ is associated with $\boldsymbol{w}_j$ if the distance between its image $\boldsymbol{f}(\boldsymbol{x}_i)$ and $\boldsymbol{w}_j$ is the smallest among the set of reference vectors. We employ this method to estimate the density of each reference vector. To this end, we use a counter ρ_j that represents the number of individuals associated with $\boldsymbol{w}_j$:

$$\rho_j = |\{i : \pi(i) = j, i = 1, 2, \ldots, N\}|, \text{ for } j = 1, 2, \ldots, N \tag{5.2}$$

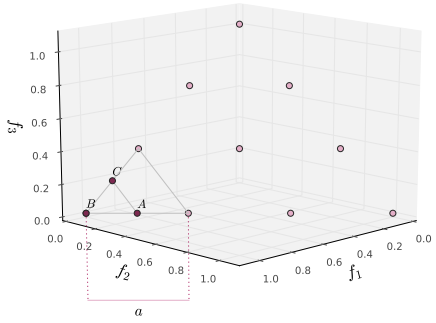


Figure 5.4: Simplex ABC. The size of each side depends on the length of a and scaling factor s.

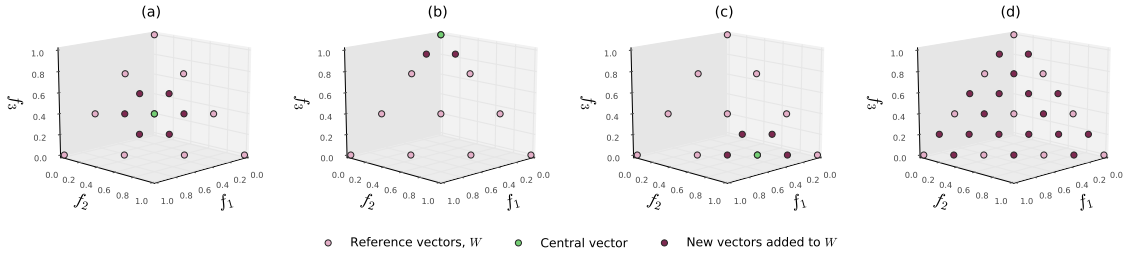


Figure 5.5: Addition of reference vectors. The locations of new vectors depends on the central vector. When every reference vector is expanded, a new set of uniformly distributed vectors is obtained (d).

This way, we can estimate the density of the niche (defined by $\boldsymbol{w}_j$) by counting the individuals in their vicinity. We can also determine whether $\boldsymbol{w}_j$ is needed to cover the Pareto front. Figure 5.3 (a) shows the approximation achieved by MOEA/D on an irregular problem called inv-DTLZ1 [36, 37]. After density estimation, we can identify the weight vectors that better approximate the Pareto front, as illustrated in Fig. 5.3 (b). Every vector $\boldsymbol{w}_j$ highlighted as an associated vector in this figure is associated with at least one individual, that is, $\rho_j > 0$.

5.1.2 Addition of Reference Vectors

Deb and Jain [36, 15] proposed a method to uniformly increase the number of weight vectors to enhance the coverage of NSGA-III in irregular Pareto front shapes. This procedure involves translating an m-dimensional simplex around a central vector. By doing this, additional reference vectors are created uniformly.

Consider the simplex labeled as ABC shown in Fig. 5.4. The length of each side of such a simplex is defined by the distance between two consecutive reference vectors, denoted by a. We can control the length of each side by using a scaling factor s. In this document, we use $s = 0.5$ as the scaling factor. Additional vectors are created by translating the simplex around the central vector. Notice that the locations of the new vectors depend on the central vector, as shown in Fig. 5.5 (a)-(c). When every vector $\boldsymbol{w} \in W$ is used as central vector, a new set of evenly-distributed vectors is obtained, as shown in Fig. 5.5 (d).

5.1.3 Proposed Indicator

In this section, we describe how to measure the relative improvement of each subproblem in a given interval. We then explain how to use these measurements to get an overview of the global progress of the population toward the Pareto front, that is, the MDP value of the population. When the subproblems cannot be significantly improved, then the population becomes stagnant. In the proposed method, the stagnation point corresponds to the generation where the value of the proposed indicator is below a threshold k. After this stagnation point, we can trigger a weight adaptation method.

Relative improvement Let $\boldsymbol{x}_t^j$ and $\boldsymbol{x}_{t-\Delta T}^j$ denote the solutions of subproblem g_j in the current generation t and the $(t - \Delta T)$-th generation, respectively. The relative improvement of subproblem g_j in the last ΔT generations is defined as follows [85]:

$$u_j = \frac{g_j(\boldsymbol{x}_{t-\Delta T}^j; \boldsymbol{w}^j) - g_j(\boldsymbol{x}_t^j; \boldsymbol{w}^j)}{g_j(\boldsymbol{x}_{t-\Delta T}^j; \boldsymbol{w}^j)} \tag{5.3}$$

This indicator is employed to infer the local progress made for each subproblem in the last ΔT generations. If u_j is close to zero, then the solution $\boldsymbol{x}_j$ of subproblem g_j may be stagnant. The relative improvement u_j of subproblem g_j has been computed every $\Delta T = 10$ generations in this

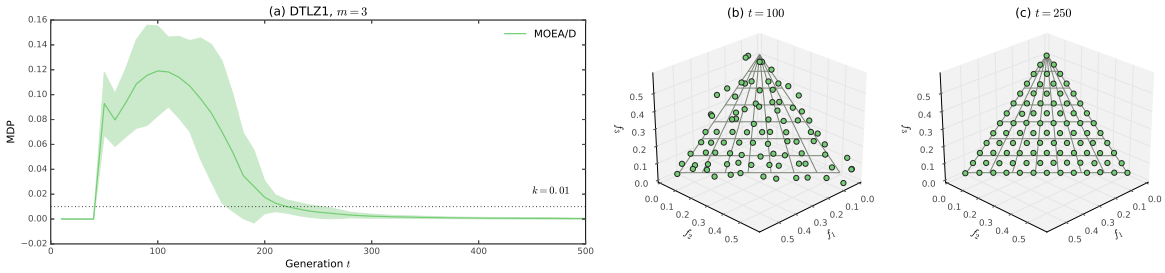


Figure 5.6: Indicator MDP. (a) Mean and standard deviation of the proposed indicator. As the indicator approaches zero, the population becomes more stable. Population (b) before and (c) after reaching the threshold $k = 0.01$.

document.

Time frame of relative improvement In order to gain a better insight into the progress of the population, we collect the last l relative improvements for each subproblem to create a time frame of relative improvements $\boldsymbol{u}_j = [u_{j,1}, \ldots, u_{j,l}]^T$. In this document, the size of the time frame is defined as $l = 5$.

Description of time frames of relative improvement The time frame $\boldsymbol{u}_j$ acts as a buffer to track the progress of subproblem g_j. This frame is reduced to a single scalar value using a measure of spread. If the spread of $\boldsymbol{u}_j$ is low, then the progress of subproblem g_j could be stagnant. On the other hand, if the spread of $\boldsymbol{u}_j$ is high, then the subproblem g_j is still moving toward the Pareto front. We proposed a spread measure called coefficient of dispersion (CD), based on the coefficient of variation (CV). This measure is defined as follows [22]:

$$CV = \frac{s}{\bar{x}} \tag{5.4}$$

where s and $\bar{x}$ denote the standard deviation and mean of the sample ($\boldsymbol{u}_j$), respectively. The CV expresses the variability of a sample in the units of its mean. This is a useful descriptive statistic to compare the dispersion among data sets (when the means are different across the data sets) [47, 22].

Given the definition of the CV, we proposed the following spread measure:

$$CD = \frac{v}{\bar{x}} \tag{5.5}$$

where v represents the variance of the sample ($\boldsymbol{u}_j$). In our preliminary experimentation, we noticed better results by using CD instead of CV. In other words, it is easier to infer the stability of the population by using the variance instead of the standard deviation of the time frame $\boldsymbol{u}_j$. After computing the CD of $\boldsymbol{u}_j$, a vector of dispersion $\boldsymbol{v}$ is obtained. This vector represents the dispersion at generation t of the relative improvements on the subproblems:

$$\boldsymbol{v} = [CD_1, \ldots, CD_N]^T \tag{5.6}$$

Median of dispersion of the population (MDP) The median of the dispersion vector $\boldsymbol{v}$ is employed to estimate the overall improvement of the population. A value close to zero means that most of the subproblems have not made significant progress toward the ideal vector, which suggests stagnation.

Figure 5.6 illustrates the proposed indicator. This figure shows the mean and standard deviation of MDP for MOEA/D. The shaded area in this figure refers to the standard deviation. MOEA/D was evaluated 30 independent times using DTLZ1 as test problem. This figure shows that the MDP is able to capture the dynamics of the population during the search process. After 250 generations, the population becomes steady, as suggested by the low value of the indicator. This implies that the subproblems cannot be improved considerably in subsequent generations. As we can see in this figure, this occurs when the MDP value is lower than $k = 0.01$.

5.1.4 Weight Adaptation Strategy

The proposed strategy is designed to be embedded into a decomposition-based method. There are two sets of reference vectors: W, which contains the initial set of reference vectors created uniformly,

Algoritmo 4 Median of Dispersion of the Population

Entrada:

Population P.

Set of reference vectors W.

Salida:

MDP value of P.

1: **for** each subproblem $j = 1, \ldots, N$ **do**

2: Create the time frame $\boldsymbol{u}$ with the last l relative improvements using Eq. (5.3):

$\boldsymbol{u}_j = [u_1^j, \ldots, u_l^j]^T$

3: Compute the CD of $\boldsymbol{u}_j$ to obtain its dispersion:

$CD_j = \frac{Var(\boldsymbol{u}_j)}{Mean(\boldsymbol{u}_j)}$

4: **end for**

5: Create the vector of dispersion $\boldsymbol{v}$ from the dispersion of each subproblem:

$\boldsymbol{v} = [CD_1, \ldots, CD_N]^T$

6: Compute the median of dispersion of the population:

$MDP = Median(\boldsymbol{v})$

and W', which contains the additional vectors created by an adaptation method. Algorithm 5 provides the details of the proposed method. In summary, the proposed adaptation is as follows:

1. Track the progress of the population toward the Pareto front using the MDP.
2. If the MDP value is below a threshold k, then:
   (a) Identify the vectors from W with higher density as explained in Section 5.1.1.
   (b) Create additional reference vectors as explained in Section 5.1.2. The set of new vectors is defined as W'.
   (c) Optimize the subproblems related the new vectors in W' in the remainder of the search process. The subproblems related to vectors in W are not longer optimized.
   (d) Select N reference vectors from W and W' when the search process is finished. This is required to keep a fixed population size.

When decomposing a problem, there is a subproblem g for each reference vector $\boldsymbol{w}$ in W. We can consider each reference vector as a direction search outlined in the objective space. In this strategy,

the subproblems related to the vectors in W are optimized until the stagnation point is reached, that is, when the MDP value reaches the threshold k. After this, an adaptation method is employed to create additional vectors. The rest of the search process is focused on optimizing the subproblems related to these new direction vectors. The intuition behind this is that, since the population is stagnant, the subproblems defined by W cannot be significantly improved anymore. Instead of using the remiander of the computational budget on them, we may optimize the subproblems defined by the new reference vectors.

After creating the new vectors, the number of solutions also increases because there is a solution associated with each subproblem. In order to keep a fixed population size, we need to filter the solutions returned by the decomposition-based method. To do this, we rank the vectors in W and W' according to their densities and scope. Lower ranks are prefered. Algorithm 6 describes the filtering method adopted in the proposed strategy. The filtering process is as follows. Vectors from W with higher density, that is, $\rho_j > 0$, have a rank $r_j = 1$. Similarly, vectors from W' with higher density have a rank $r_j = 2$. Then, the vectors with no associated solutions are ranked. Vectors from W with no density, that is, $\rho_j = 0$, have a rank $r = 3$. Finally, vectors from W' with no density have a rank $r_j = 4$. Once all vectors have been ranked, we preserve the solutions associated with vectors with better (lower) ranks, as detailed in Algorithm6 (lines 8-14). Figure 5.7 illustrates the filtering process. Observe that vectors with a lower rank provide a better approximation of the Pareto front shape than the vectors with a higher rank.

5.1.5 MDP for Dynamic Problems

In the previous section, the proposed indicator is employed to trigger the addition of reference vectors. Given that this indicator tracks the historical progress of each subproblem, it provides a snapshot of the population at a given generation. Although, it could also be employed for detecting changes in the population caused by a perturbation in the environment. This information could be helpful in other contexts, such as dynamic multiobjective optimization problems.

Algoritmo 5 Adaptive Strategy

Entrada:
Population P.
Set of reference vectors W, where $N = |W|$.
Salida:
Set of additional vectors W'.
Compute the density estimation procedure to obtain $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_N]^T$:
$\boldsymbol{\rho} = \text{DensityEstimation}(P, W)$
if $\rho_j = 1$, for $j = 1, \ldots, N$ **then**
Expansion is not necessary
else
for $j = 1, \ldots, N$ **do**
if $\rho_j > 0$ **then**
Create new reference vectors around $\boldsymbol{w}_j$
Add new vectors to W'
end if
end for
end if
return W'

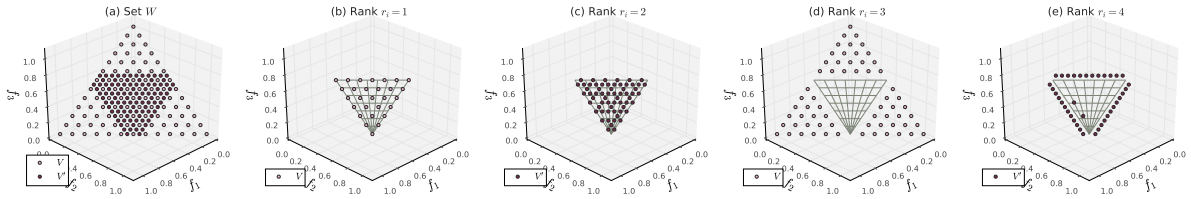


Figure 5.7: Ranking vectors. (a) Union of W and W'. Crowded vectors from (b) W and (c) W'. Vectors with no associated solutions from (d) W and (e) W'. Vectors with a lower rank provide a better approximation of the Pareto front shape. The individuals associated to them are preferable for the filtering process.

In this section, we evaluate another property of the proposed indicator. Here, we are interested in tracking an alteration in the population resulting from changing the shape of the Pareto front. We proposed a new test problem called Dyn-inv-DTLZ. This problem is based on DTLZ1 and inv-DTLZ1. Every T generations, the objective functions are switched, from inv-DTLZ1 to DTLZ1, so that the location of the Pareto front is altered.

Before explaining the proposed mechanism for change detection, consider Fig. 5.8. This figure

Algoritmo 6 Filter Population

Entrada:

Sets of vectors, W and W'.

Enlarged population P, where $|P| = |W| + |W'|$.

Salida:

Population Q, where $|Q| = N$.

Compute $\boldsymbol{z}^*$ and $\boldsymbol{z}^{nad}$, for $i = 1, \ldots, m$:

$z_i^* = \min\{f_i(\boldsymbol{x}) : \boldsymbol{x} \in P\}$

$z_i^{nad} = \max\{f_i(\boldsymbol{x}) : \boldsymbol{x} \in P\}$

Normalize $\boldsymbol{f}(\boldsymbol{x})$, for $\boldsymbol{x} \in P$, and $i = 1, \ldots, m$:

$f_i'(\boldsymbol{x}) = \frac{f_i(\boldsymbol{x}) - z_i^*}{z_i^{nad} - z_i^*}$

Associate solutions, $\boldsymbol{w}_j \in W \cup W'$:

for $i = 1, 2, \ldots, |W| + |W'|$ **do**

$k = \arg\min_j \left(d^{\perp}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{w}_j)\right)$

$\rho_k = \rho_k + 1$ ▷ Density of $\boldsymbol{w}_k$

end for

Rank reference vectors, $\boldsymbol{w}_j \in W \cup W'$:

$$r_j = \begin{cases} 1 & \text{if } \boldsymbol{w}_j \in W \text{ and } \rho_j > 0 \\ 2 & \text{if } \boldsymbol{w}_j \in W' \text{ and } \rho_j > 0 \\ 3 & \text{if } \boldsymbol{w}_j \in W \text{ and } \rho_j = 0 \\ 4 & \text{if } \boldsymbol{w}_j \in W' \text{ and } \rho_j = 0 \end{cases}$$

Select solutions according their ranked reference vectors:

for $k = 1, 2, 3, 4$ **do**

  for $j = 1, 2, \ldots, |W| + |W'|$ **do**

    if rank r_j equals k and $|Q| < N$ **then**

      $Q = Q \cup \{\boldsymbol{x}_j\}$

    end if

  end for

end for

return Q

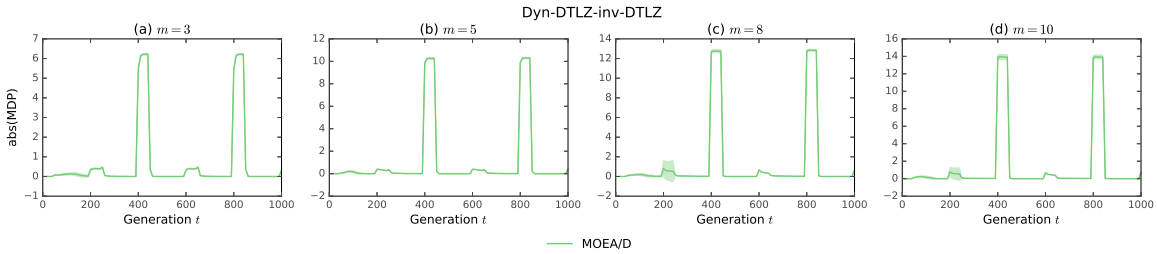


Figure 5.8: Mean of absolute value of MDP indicator in Dyn-inv-DTLZ for up to $m = 10$ objectives. The shaded region represents the standard deviation for 30 independent runs. The proposed indicator is able to detect changes in the population at generations $t = 200, 400, 600, 800, 1000$. Therefore, it might be used as a change detector in a dynamic multiobjective optimization problem.

Algoritmo 7 Peak Detection Mechanism

Entrada:

MDP time series $\{m_1, m_2, \ldots, m_k\}$.
MDP time step m.

Salida:

Boolean flag that denotes whether a peak is found, $flag$.

1: $flag =$ False
2: Update mean and standard deviation
$\bar{m} = Mean(\{m_1, m_2, \ldots, m_k\})$
$\sigma = SD(\{m_1, m_2, \ldots, m_k\})$
3: **if** $m \geq \bar{m} + 2\sigma$ **then**
4: $flag =$ True
5: **else**
6: Add m into the time series.
7: **end if**
8: **return** $flag$

shows the behavior of MOEA/D in the proposed dynamic problem regarding the absolute value of the MDP. Each plot depicts the mean of the absolute MDP value over 30 independent runs, whereas the shaded areas represent the standard deviation. From this figure, we can see how the indicator shows peaks near generations $t = 200, 400, 600, 800, 1000$. That is, the proposed indicator is able to detect perturbations in the population resulting from changing the Pareto front. Similar results were observed for $m = 5, 8, 10$ objectives.

In the following, we introduce a method for detecting a perturbation in the population through the use of the MDP. When a change is detected, an additional method can be performed as a

callback. The details of the proposed method are given in Algorithm 7. We track the MDP as a time series $\{m_1, m_2, \ldots, m_k\}$ throughout the search, where each sample represents the MDP value at a given generation. Before adding a new time step (m) into the time series, we compute the mean and standard deviation of all the available samples, denoted as $\bar{m}$ and σ respectively. We then determine whether the new time step is an outlier. If m is greater than $\bar{m} + 2\sigma$, then the time step is considered an outlier, meaning that a perturbation is detected. If so, the time step m is not added into the time series to prevent modifying its mean and standard deviation abruptly. This help us identify incoming samples as outliers incorrectly.

5.2 Experimental Design

This section describes the experimental design employed to evaluate the performance of the proposed approach. We have embedded the proposed indicator along with the addition operation of weight vectors into MOEA/D. The addition of weight vectors is performed if (a) the MDP value is below a given threshold k, and (b) there are overcrowded weight vectors. The first condition is needed to get a better approximation of the Pareto front shape. If it is achieved, the second condition is evaluated. This condition is needed to determine whether the addition of vectors is required: If every weight vector is associated with one solution, then the addition is not required; otherwise, there are overcrowded vectors and the addition of weight vectors can be performed. The resulting approach is called AMOEA/D. Four experiments have been conducted:

- Experiment 1. The goal of this experiment is to determine whether the proposed method for adaptation of reference vectors improves the diversity of MOEA/D in inv-DTLZ1, without deteriorating its performance in DTLZ1. The value of the threshold was defined as $k = 0.01$ for $m = 3, 5, 8$, and $k = 0.005$ for $m = 10$.

- Experiment 2. The proposed method is based on a density estimation procedure that identifies promising niches for expansion. Such a procedure associates a pair of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$

according to the perpendicular distance between them. However, other association criteria can be employed. In this experiment, we explored two additional approaches for vector association: ASF [77], and the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$. The goal of this experiment is to determine whether a specific association criterion is more suitable than other for the proposed method.

- Experiment 3. The proposed method requires the definition of the expansion threshold k. In order to get a better insight of the influence of this parameter, we evaluated five configurations of AMOEA/D by using $k \in \{0.02, 0.01, 0.005, 0.0025, 0.00125\}$ in inv-DTLZ1.

- Experiment 4. The goal of this experiment is to determine whether the MDP indicator can detect a change in the population when the location of the Pareto front is modified. We have employed Dyn-inv-DTLZ1 as a proof-of-concept. The location of the Pareto front changes every 200 generations.

Each algorithm has been evaluated 30 independent times on every test instance. The computational budget assigned for each problem instance is given in Table 5.1. Table 5.2 shows the population size defined for each problem according to the number of objectives. Also, the number of divisions (H) required for the Simplex-Lattice procedure is shown. The number of reference vectors (N) depends on both the number of divisions and objectives. Simulated binary crossover (SBX) and polynomial mutation [16] have been employed as variation operators with the following configuration: $\eta_c = 20$, and $p_c = 1.0$, for SBX, and $\eta_m = 15$, and $p_m = 1/n$ for polynomial mutation.

5.3 Experimental Results

Experiment 1 Table 5.3 summarizes the mean, median, and standard deviation of the hypervolume indicator for both compared approaches. As it can be seen, the performance of both algorithms is the same in DTLZ1 in every test instance. This is because AMOEA/D is able to determine whether an expansion of reference vectors is needed. In this case, such an expansion is not required because

Objectives (m)	Problem DTLZ1	inv-DTLZ1
3	300	600
4	500	1000
5	600	1200
6	650	1300
7	700	1400
8	800	1600
10	1000	2000

Table 5.1: Maximum number of generations specified for each test instance.

Objectives (m)	Divisions (H)	Population Size (N)
3	12	91
4	7	120
5	6	210
6	4	126
7	4	210
8	3	120
10	3	220

Table 5.2: Population size (N) according to the number of objectives (m) and divisions (H).

the surface of the Pareto front is uniformly covered and every reference vector is occupied by one solution, as shown in Fig. 5.9. This figure shows the approximation obtained by each algorithm in DTLZ1 for $m = 3$. As it can be seen, the approximation of both algorithms is the same. On the other hand, AMOEA/D was superior to MOEA/D in inv-DTLZ1 in most cases. Figure 5.10 shows the approximations obtained by both approaches. Figure 5.11 shows hypervolume boxplots for Experiment 1. From this figure, we can see how the proposed strategy improves the performance of MOEA/D as the number of objectives increases.

Table 5.4 summarizes the mean, median, and standard deviation of SP values for both algorithms. In this case, the performance of both approaches is the same in the first test problem. On the other hand, AMOEA/D reached better results than MOEA/D in terms of diversity in the irregular problem.

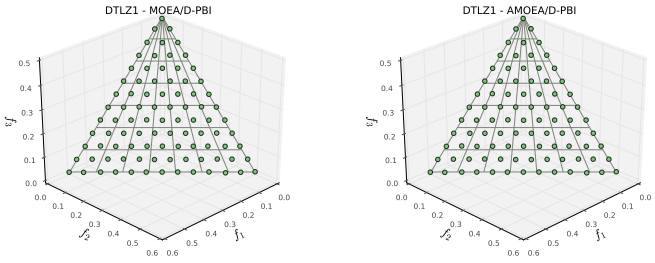


Figure 5.9: Approximations obtained by MOEA/D (a) and AMOEA/D (b) in DTLZ1.

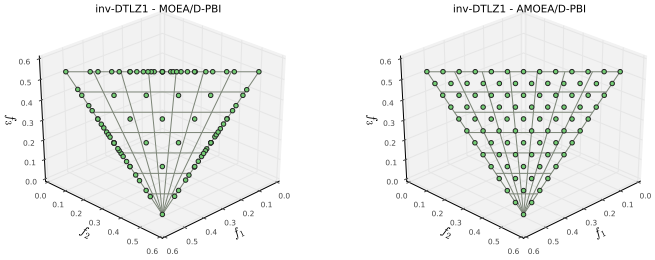


Figure 5.10: Approximations obtained by MOEA/D (a) and AMOEA/D (b) in inv-DTLZ1.

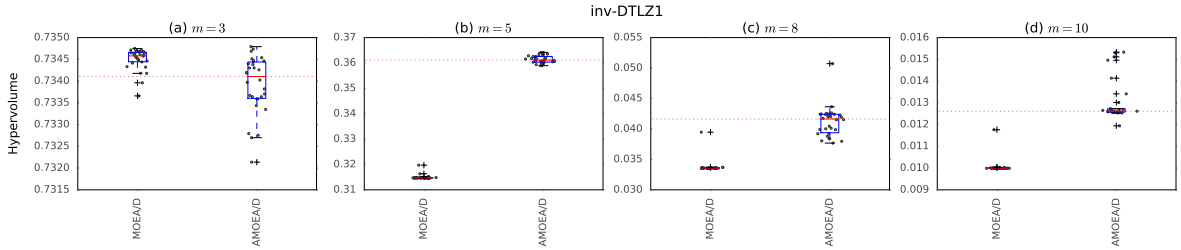


Figure 5.11: Hypervolume boxplots for each algorithm evaluated in Experiment 1. The horizontal dotted line represents the median of the hypervolume reached by the proposed method.

Figure 5.10 depicts the approximations obtained by both algorithms for $m = 3$ objectives. As it can be seen, AMOEA/D obtained a better coverage of the Pareto front than that achieved by MOEA/D. Also, the approximation reached by AMOEA/D is uniformly distributed over the Pareto front. This uniformity is achieved by the uniform addition of reference vectors during the search process. In this case, the proposed method was able to determine that an expansion of reference vectors was required. Thus, poorly-crowded niches were populated successfully.

Problem	m	MOEA/D	AMOEA/D
DTLZ1		0.9794	0.9794
	3	0.9795	0.9795
		±0.0005	±0.0005
		0.9994	0.9994
	5	0.9994	0.9994
		±0.0000	±0.0000
		1.0000	1.0000
	8	1.0000	1.0000
		±0.0000	±0.0000
		1.0000	1.0000
	10	1.0000	1.0000
		±0.0000	±0.0000
inv-DTLZ1		**0.7345**	0.7339
	3	**0.7346**	0.7341
		±0.0002	±0.0007
		0.3150	**0.3614**
	5	0.3147	**0.3611**
		±0.0009	±0.0015
		0.0338	**0.0412**
	8	0.0336	**0.0416**
		±0.0011	±0.0025
		0.0101	**0.0130**
	10	0.0100	**0.0126**
		±0.0003	±0.0009

Table 5.3: Mean, median, and standard deviation for hypervolume indicator in test instances. Best performance is shown in bold.

Experiment 2 Figure 5.12 shows hypervolume boxplots for each evaluated configuration of AMOEA/D. From this figure, it can be seen that the three approaches behave similarly for the tested instances of inv-DTLZ1. Hence, they can be interchangeable in the proposed method. Additional experiments are needed to determine whether a given approach is more appropriate for our method.

Problem	m	MOEA/D	AMOEA/D
DTLZ1		1.5337	1.5337
	3	1.5330	1.5330
		±0.0043	±0.0043
		1.7333	1.7333
	5	1.7334	1.7334
		±0.0009	±0.0009
		1.8806	1.8806
	8	1.8830	1.8830
		±0.0067	±0.0067
		1.9596	1.9596
	10	1.9631	1.9631
		±0.0078	±0.0078
inv-DTLZ1		1.5252	**1.5300**
	3	1.5252	**1.5298**
		±0.0003	±0.0008
		1.5607	**1.6355**
	5	1.5607	**1.6355**
		±0.0002	±0.0011
		1.3950	**1.4143**
	8	1.3918	**1.4291**
		±0.0158	±0.0519
		1.3945	**1.4760**
	10	1.3918	**1.4625**
		±0.0162	±0.0272

Table 5.4: Mean, median, and standard deviation for Solow-Polasky indicator in test instances. Best performance is shown in bold.

Experiment 3 Figure 5.13 shows hypervolume boxplots for each evaluated configuration of AMOEA/D. Like previous experiment, it is not evident which value for k is better for a particular test instance. From Fig.5.13 (d), the four configurations have a similar performance according to the median of the hypervolume value. Nevertheless, the difference among the maximum and minimum hypervolume values is the lowest for $k = 0.0025$.

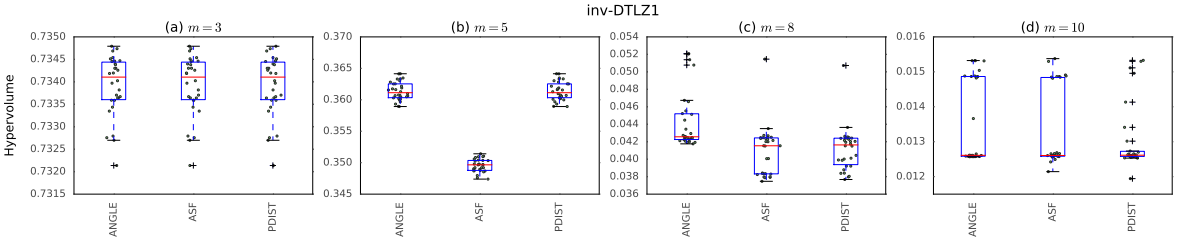


Figure 5.12: Hypervolume boxplots of AMOEA/D with three association criteria from Experiment 2.

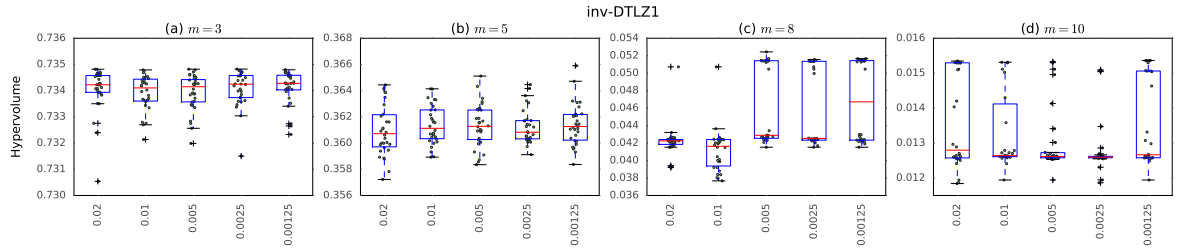


Figure 5.13: Hypervolume boxplots of AMOEA/D for $k \in \{0.02, 0.01, 0.005, 0.0025, 0.00125\}$ from Experiment 3.

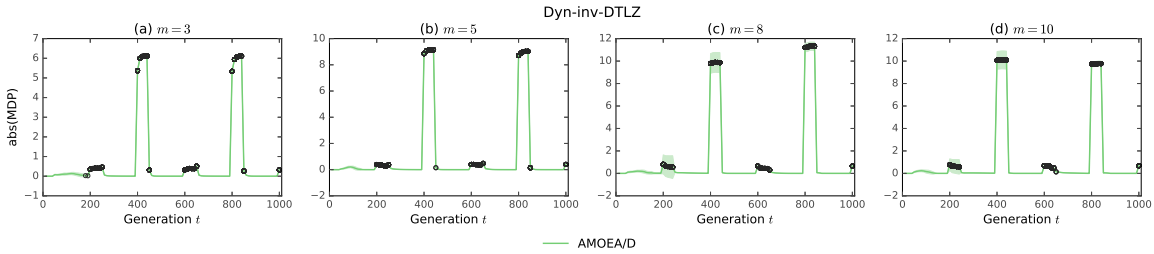


Figure 5.14: Mean of absolute value of MDP indicator in Dyn-inv-DTLZ for up to $m = 10$ objectives. The shaded region represents the standard deviation for 30 independent runs. The black circles represent the generations where a change in the population was detected by the proposed mechanism.

Experiment 4 The proposed indicator has been employed to determine when the addition of reference vectors will take place. In this case, we are interested on determining whether the MDP indicator can detect a change in the population when the location of the Pareto front is modified. To evaluate the performance of the mechanism for change detection, the proposed dynamic problem was employed. Fig. 5.14 depicts the mean and standard deviation of the absolute value of the MDP indicator. The proposed AMOEA/D coupled with the change detection mechanism was evaluated 30 independent times. The black circles represent those generations where a change was detected by the

proposed mechanism. The extent of each peak in the MDP indicator is about 50 generations. For this reason, the mechanism reports about five circles for each peak. When a change is detected for the first time (i.e., the first of the five circles in each peak), the adaptive strategy is invoked. However, other strategy can be used as well. From this figure, we can see how the proposed AMOEA/D along with the change detection mechanism is able to identify the variations in the dynamic test problem. This identification is possible by tracking the MDP during the search process. Although this test is a proof-of-concept, it shows the potential application of the proposed indicator for dynamic multiobjective optimization problems.

5.4 Summary

A common approach to enhance decomposition in irregular problems is to adapt the reference vectors throughout the search process. However, premature adaptation could be ineffective because the population does not provide a good approximation of the Pareto front shape in early generations.

In this chapter, we follow a different approach. Here, we adapt the reference vectors only when the population becomes stagnant. We assume that the population will provide a better approximation of the Pareto front after stagnation. To this end, we introduce an indicator called the median of dispersion of the population (MDP). This indicator expresses the overall progress of the population toward the Pareto front. When the MDP value is below a threshold, the population is considered stagnant, meaning that the adaptation can be performed. We also employ this indicator to detect a perturbation in the population caused by changing the Pareto front. As a proof of concept, we define a new test problem that induces a change in the population regarding a predefined interval. Our experimental results show that:

- The proposed indicator can be used for enhancing a weight adaptation mechanism. That is, we can prevent premature adaptation and reduce the computational burden related to the adaptation.

- Also, we use the MDP for detecting a change in the population. The proposed method for change detection was able to identify the perturbation introduced by the proposed problem every 200 generations. This shows the potential application of the proposed indicator for dynamic problems.

6

Final improvements

We have described the two main contributions of this work: an efficient method for the collaborative use of scalarizing functions (MSS), and a method for adapting reference vectors based on the relative progress of subproblems (MDP). In this chapter, we introduce an additional improvements that combines both methods. The aim of this framework, called as MSS+MDP hereinafter, is to improve the effectiveness of MSS in problems with inverted Pareto front shapes. We hypothesize that, if MSS is coupled with MDP, then the diversity of the population can be enhanced in irregular problems. Our results show that MSS+MDP is better that MSS in irregular problems regarding the Hypervolume indicator.

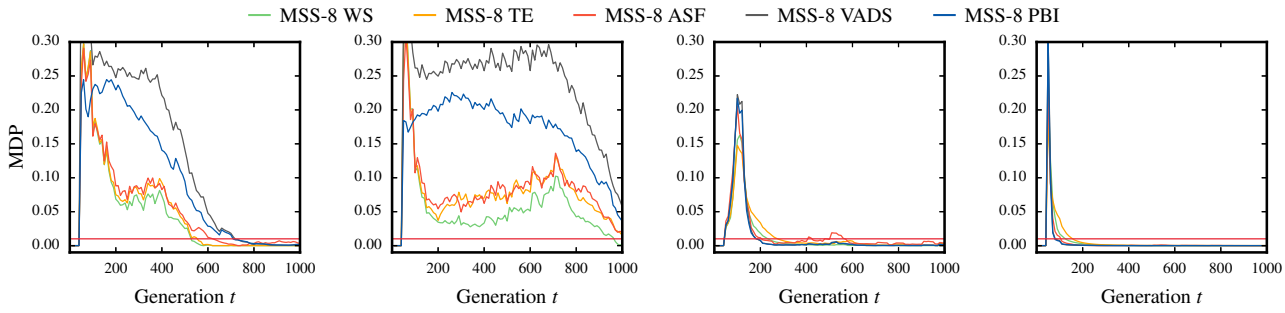
6.1 Integration of techniques

Figure 6.1 shows the convergence of the auxiliary populations of MSS with regards to MDP. The red horizontal line represents the MDP threshold (0.01). From preliminary experimentation, we noticed that the proposed method converged faster in DTLZ2, inv-DTLZ1, and MaF1. DTLZ2 is not difficult

for many evolutionary algorithms since it is a unimodal problem. For this reason, it is relatively easy to converge toward the Pareto front. MaF1 shares some aspects of DTLZ2. As a result, we can expect that an evolutionary algorithm can approach its Pareto front quickly, as shown in Fig. 6.1. Although inv-DTLZ1 is not a unimodal problem, the proposed method exhibited a similar behavior. In this case, most of the auxiliary populations did not make significant progress toward the Pareto front after near 300 generations. This is attributed to the shape of the Pareto front. In practice, a large proportion of the population of a decomposition-based method lies on the boundaries of the Pareto front of inv-DTLZ1. Once the population reaches the boundaries, it becomes stagnant. Now, compare this behavior with DTLZ1 and DTLZ3. In DTLZ1, the proposed method took a larger amount of time to converge, since the populations did not make a significant progress after 800 generations. From Fig. 6.1 (b), we can see that only the population corresponding to WS reached the MDP value. Although, only the extreme solutions of the Pareto front are likely to be found with this function. Observe that other populations, like PBI and VADS, did not achieved the given threshold. This means that those populations can still be improved. In general, we noticed that the proposed approach reached the Pareto front of inverted problems quickly. As shown in 6.1, we can use the proposed MDP to determine whether the populations can be further improved or are stagnant.

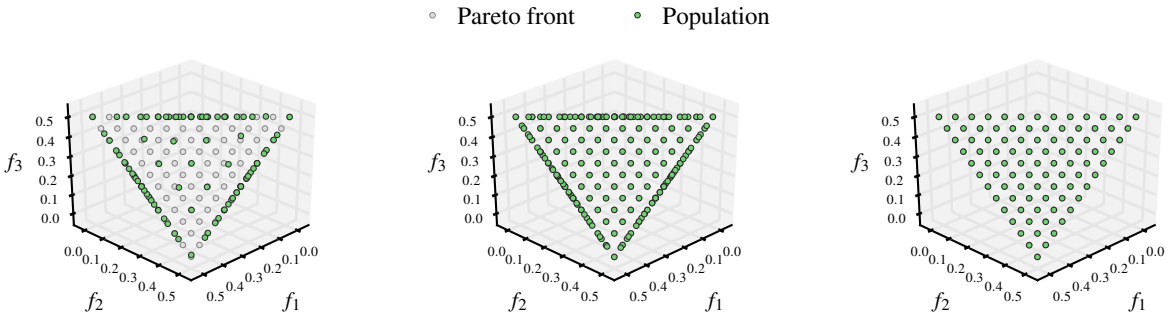
In this section, our idea is to use MDP to track the progress of each auxiliary population as explained above. Once all the populations reaches the stagnation point, we can adapt the reference vectors.

Figures 6.2 and 6.3 illustrates the adaptation of reference vectors in inv-DTLZ1 and MaF1, respectively. We can see the population of MSS at three different phases of the search process. First, we show the population after reaching the stagnation point. Figure 6.2 (a) shows the auxiliary population corresponding to PBI in inv-DTLZ1 at generation $t = 150$. As expected, most of the solutions lies on the boundaries of the Pareto front. Although, it provides a good approximation of the Pareto front shape. Next, we show the population at the end of the search process. In the



(a) DTLZ1 $m = 10$ (b) DTLZ3 $m = 10$ (c) inv-DTLZ1 $m = 10$ (d) MaF1 $m = 10$

Figure 6.1: Convergence of MSS. Each plot shows the behavior of the auxiliary populations of MSS regarding MDP. The red horizontal line represents the MDP threshold. The proposed method converges (stagnates) faster in inv-DTLZ1 and MaF1.



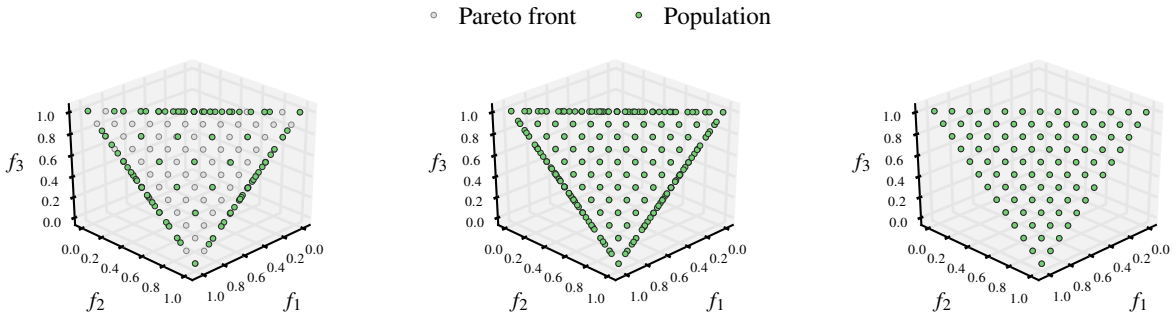
(a) inv-DTLZ1, $t = 150$
Population before adaptation

(b) inv-DTLZ1, $t = 500$
Population before filtering

(c) inv-DTLZ1, $t = 500$
Population after filtering

Figure 6.2: Illustration of the adaptation strategy in inv-DTLZ1. We show the population of MSS at three different phases of the search process: (a) when the population reaches the stagnation point and the adaptation of reference vectors can be performed, (b) at the end of the search process, and (c) after filtering solutions to keep a fixed population size.

proposed strategy, additional reference vectors are created in empty niches to enhance the diversity of the population. The resulting solution set is shown in Fig. 6.2 (b). As we can see, the coverage of the Pareto front is improved. We then filter solutions in order to maintain a fixed population size, we filter solutions, as shown in Fig. 6.2 (c).



(a) MaF1, $t = 70$
Population before adaptation

(b) MaF1, $t = 500$
Population before filtering

(c) MaF1, $t = 500$
Population after filtering

Figure 6.3: Illustration of the adaptation strategy in MaF1. We show the population of MSS at three different phases of the search process: (a) when the population reaches the stagnation point and the adaptation of reference vectors can be performed, (b) at the end of the search process, and (c) after filtering solutions to keep a fixed population size.

6.2 Experimentation

In order to evaluate the proposed integration, we use inv-DTLZ1 and MaF1 test problems with $m = 3, 5, 8, 10$. The goal of this experiment is to determine whether using MSS and MDP provides a solution set with a higher diversity instead of using MSS alone. To this end, we adopt the Hypervolume as performance indicator. Finally, we evaluate the proposed method 50 independent times on each problem.

Table 6.1 shows the performance of the proposed method. We included a method, termed as MSS+MDP*. Unlike MSS+MDP, this method does not filter the final population. As a result, it returns l times more solutions. From this table, we can see than both MSS+MDP and MSS+MDP* outperformed MSS with respect to the Hypervolume indicator. In addition, we can achieve a slightly better performance by returning all the solutions.

Table 6.1: Mean and standard deviation of Hypervolume values over 50 runs for each test instance. The best mean value for each test problem is shown in bold.

Problem	m	MSS	MSS+MDP	MSS+MDP*
inv-DTLZ1	3	0.68925 (0.06263)	0.86360 (0.08094)	$\mathbf{0.87864}$ (0.07810)
	5	0.25133 (0.13571)	0.32537 (0.17787)	$\mathbf{0.56042}$ (0.19556)
	8	0.05936 (0.11369)	0.09006 (0.11495)	$\mathbf{0.23504}$ (0.17667)
	10	0.04042 (0.08909)	0.12156 (0.11048)	$\mathbf{0.21215}$ (0.21242)
MaF1	3	0.81682 (0.03032)	0.93204 (0.01016)	$\mathbf{0.94779}$ (0.01186)
	5	0.24638 (0.09441)	0.20991 (0.06906)	$\mathbf{0.31807}$ (0.08455)
	8	0.08162 (0.07756)	0.11584 (0.07367)	$\mathbf{0.14418}$ (0.11676)
	10	0.03094 (0.03125)	$\mathbf{0.20633}$ (0.01719)	0.20606 (0.11429)

6.3 Summary

In this chapter, we introduced an additional improvements to the proposed method. We combined both MSS and MDP for tackling irregular problems. To this end, we track the relative improvement of the auxiliary populations using the proposed MDP. When the populations becomes stagnant, we then trigger the adaptation of reference vectors. The resulting approach, MSS+MDP, was evaluated on two irregular problems, inv-DTLZ1 and MaF1. The aim of this approach is to enhance the diversity of the population of MSS by adapting the reference vectors. Our results showed that the resulting approach is better than MSS regarding the Hypervolume indicator.

7

Final Remarks

Decomposition-based MOEAs have been extensively employed in the literature given their efficiency for addressing MaOPs. In this document, we studied how to improve the performance of these methods. Specifically, we tackled the following challenges:

Selection of scalarizing function In order to decompose a problem, a scalarizing function g is required. This scalarizing function is employed for comparing solutions in the objective space. Consequently, it plays a crucial role for driving the search. However, selecting a good scalarizing approach is not straightforward since it often depends on the properties of the problem, such as the shape of the Pareto front. Moreover, such properties are generally unknown beforehand the search. Therefore, the performance of decomposition-based algorithms significantly depends on the choice of an appropriate scalarizing function.

Diversity deterioration in irregular problems In addition to g, a set of evenly distributed reference vectors is needed for decomposing a problem. These reference vectors can be considered as search

directions spread out in the objective space. Similarly to the choice of a scalarizing function, the reference vectors must be specified regarding the shape of the Pareto front. When the shape of the Pareto front is regular, decomposition-based methods generally can find a well-distributed set of solutions on the Pareto front. Nevertheless, when the Pareto front shape is irregular, it is difficult for these methods to obtain a good Pareto front approximation. This issue is related to the locations of the reference vectors in the objective space. When such vectors does not reflect the shape of the Pareto front, the diversity of decomposition-based MOEAs is deteriorated. In consequence, some authors have proposed techniques to adapt these vectors throughout the search process, commonly called weight adaptation methods, to obtain a better approximation of the Pareto front.

In this document, we proposed two approaches to face these challenges. In summary, these were the steps that we followed to develop our methods:

- First, we conducted a preliminary study to gain a better insight of the importance of the scalarizing approach when a problem is transformed via decomposition. To this end, we analyzed six scalarizing functions from the literature on multiple test problems. The aim of this study was to determine whether a given scalarizing function g is suitable for addressing a particular type of problem. Our experimental results showed that PBI, VADS, and ASF were good choices for decomposing an MOP. They showed promising results in most of the test instances with respect to the Hypervolume indicator. On the other hand, we observed that MOEA/D coupled with IPBI was unable to reach a good approximation in most of the test instances.

- We then developed a second study to analyze the diversity deterioration of MOEA/D in irregular problems. We hypothesized that the diversity of MOEA/D could be improved by creating additional reference vectors in the appropriate region in the objective space. To validate this, we defined three sets of reference vectors to solve a problem with a regular (DTLZ1) and an irregular (inv-DTLZ1) Pareto front shape. Our results showed that the diversity of

MOEA/D was improved only when the additional reference vectors were created near empty niches in the objective space. On the other hand, when the reference vectors were added in overcrowded regions, such as the boundaries of the Pareto front, the diversity does not improved considerably. This remarks the importance of reallocating the reference vectors to enhance the coverage of the Pareto front.

- In order to address our first challenge, we propose a framework to use scalarizing functions collaboratively, called Multiple Survival Selection (MSS). Our approach considers multiple candidate scalarizing functions and selects one of them based on their ability to identify promising individuals in the survival selection phase. The effectiveness of each candidate function is measured periodically, meaning that well-performing candidates are promoted throughout the search.

  One of the main concerns we faced when developing the proposed method was finding a way for measuring the effectiveness of the scalarizing functions throughout the search. After analyzing several approaches, we adopted the diversity comparison indicator (DCI) [50] as performance indicator. This indicator measures the relative diversity among several sets of non-dominated solutions rather than providing an absolute measure of distribution for a single solution set. A solution set with the best (highest) DCI value implies that its coverage is better with respect to the other solutions sets involved in the diversity assessment. In our context, we select the scalarizing function g^k corresponding to the solution set P^k with the best DCI value. This means that the solutions selected by the k-th scalarizing function are better than those selected by the other candidate functions in terms of relative diversity, suggesting that it is a suitable scalarizing function for the problem at hand.

  Our approach assumes that, given multiple candidate scalarizing functions, at least one of them is appropriate for the problem to be solved. If so, suitable candidate functions can be promoted throughout the search. Another advantage of this collaborative approach is that

good solutions of well-performing candidate functions can be shared among other scalarizing functions, enhancing the convergence of the population. Our experimental results show that the proposed approach can use multiple scalarizing functions efficiently on several test problems, improving the performance of decomposition in many-objective optimization regarding IGD and Hypervolume.

- In order to tackle the second challenge, we proposed an indicator-based weight adaptation method to improve the diversity of MOEA/D in irregular problems. Previous works [52, 36, 37] have shown that adapting the reference vectors regularly throughout the evolutionary process is a promising alternative for improving the diversity of the population in irregular problems. Most of these adaptation method relies on the population to obtain an approximation of the Pareto front shape for weight adaptation. This way, it could be possible to identify empty niches in the objective space from such an approximation. However, such a recurrent adaptation could cause a significant overhead. For example, early adaptation could be unnecessary and ineffective since the population does not provide a good Pareto front approximation at early generations. Instead, we proposed to adapt the reference vectors when the population provides a good approximation of the Pareto front. The proposed indicator, called median of dispersion of the population (MDP), is employed to estimate the progress of the population toward the Pareto front and trigger the weight adaptation when needed.

  The proposed method is based on a utility function called relative improvement [85]. This function was originally developed for measuring the hardness of a subproblem in an online manner [85]. If the relative improvement of a subproblem is high, then it can be regarded as a easy-to-improve subproblem; otherwise, it can be considered as a hard subproblem. In our context, we use this function to determine the progress made by a subproblem within a period of time. A high relative improvement means that the corresponding subproblem has progressed significantly toward the Pareto front. On the other hand, if the relative improvement is small

(i.e., close to zero), then we considered that the subproblem cannot be further significantly improved, suggesting stagnation. The proposed MDP considers the relative improvements of all the subproblems at a given phase of the search. It provide us a way to determine whether the population keeps moving toward the Pareto front, or it is mostly stagnant. If so, we can use the population as an approximation of the Pareto front shape and trigger the adaptation of reference vectors. The aim of the proposed indicator is to conduct a more effective weight adaptation by suggesting when to adapt the reference vectors.

- We then integrated both MSS and MDP into a single framework to face the two challenges described in this section. The ultimate goal of this integration is to improve the perfomance of decomposition-based methods by selecting a suitable scalarizing function automatically and adapting the reference vectors.

  Our experiments show that the performance of MSS+MDP is better than MSS alone in inv-DTLZ1 and MaF1 in terms of Hypervolume. Although the performance was enhanced, it was not significantly improved as we initially imagine. We attribute this to the method employed for weight adaptation. For this reason, we hypothesize that we could achieve a better performance by modifying this component of the proposed method. We need to verify this assumption in future work.

We obtained the following conclusions from the development process:

- MSS is a promising alternative for selecting suitable scalarizing functions. However, its behavior depends on the performance indicator (DCI). We observed that the proposed method was unable to select a suitable scalarizing function under given circumstances. For instance, we noted this issue on MaF4. After analyzing the behavior of the proposed method, we noticed that this issue is due to the procedure adopted for normalizing the objective vectors [5]. We found that this method is sensible to the range of the objectives. When the problem is badly-scaled, as the case of MaF4, this normalization procedure has difficulty to estimate the Nadir

point. In consequence, this will affect the outcome of the DCI procedure, deteriorating the effectiveness of the proposed MSS. As future work, we need to use a normalization procedure reliable on badly-scaled problems.

- MSS is a promising alternative for the selection of scalarizing functions. This method showed good performance in multimodal problems. Although, additional work is needed to improve its behavior when most of the scalarizing functions are not helpful for addressing a particular problem.

- The proposed MDP can be incorporated into a weight adaptation method to enhance the diversity of MOEA/D. As a proof of concept, we adopted an early weight adaptation method from the literature due to its simplicity, proposed by Deb and Jain [36, 37]. Although, we can also employed more sophisticated weight adaptation techniques.

The effectiveness of the proposed methods has been evaluated. As future work, we consider the following directions:

- To extend our preliminary study to include additional scalarizing functions and test problems. The goal of such an extension is to get a broader sense of the scope of applicability of a given scalarizing function.

- To analyze alternative indicators to measure the effectiveness of the scalarizing function. The current configuration of the proposed method achieved good performance on most of the test problems of our experiments. However, we could improve the performance of the proposed method regarding specific issues, like MaF4, by considering another indicator.

- Coarse and fine grained selection. The current approach performs a coarse grained selection because all the solutions selected by the best scalarizing function are preserved for the next generation. We hypothesize that we could make better use of other scalarizing functions if a

fine grained selection is employed. In this sense, we could select the best solutions from each auxiliary population and keep them during survival selection. To achieve this, we must explore alternative methods for performance assessment.

- Use of additional scalarizing functions. As a proof of concept, we selected a representative set of scalarizing functions from the literature. However, it would be interesting to evaluate recent methods, such as constrained PBI [73], localized weighted sum [76], Pareto adaptive scalarizing methods [74], among others [57, 62, 39].

Bibliography

[1] A. Trivedi and D. Srinivasan and K. Sanyal and A. Ghosh (2017). A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462.

[2] Asafuddoula, M., Ray, T., and Sarker, R. (2015). A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460.

[3] Bader, J. and Zitzler, E. (2011). Hype: An Algorithm for Fast Hypervolume-based Many-objective Optimization. *Evol. Comput.*, 19(1):45–76.

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

[5] Blank, J., Deb, K., and Roy, P. (2019). Investigating the Normalization Procedure of NSGA-III. In *Evolutionary Multi-Criterion Optimization*, pages 229–240. Springer International Publishing.

[6] Bradstreet, L., While, L., and Barone, L. (2010). A Fast Many-objective Hypervolume Algorithm using Iterated Incremental Calculations. In *IEEE Congress on Evolutionary Computation*, pages 1–8.

[7] Brockhoff, D., Wagner, T., and Trautmann, H. (2012). On the Properties of the R2 Indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 465–472, New York, NY, USA. ACM.

[8] Brockhoff, D., Wagner, T., and Trautmann, H. (2015). R2 Indicator-based Multiobjective Search. *Evol. Comput.*, 23(3):369–395.

[9] Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. (2016). A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791.

[10] Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., and Yao, X. (2017). A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81.

[11] Chiang, T. C. and Lai, Y. P. (2011). MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1473–1480.

[12] Coello, C., Lamont, G., and Veldhuizen, D. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2 edition.

[13] Das, I. and Dennis, J. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. on Optimization*, 8(3):631–657.

[14] Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd.

[15] Deb, K. and Jain, H. (2014). 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.

[16] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

[17] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. In Abraham, A., Jain, L., and Goldberg, R., editors,

Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, pages 105–145. Springer London, London.

[18] Díaz, A., Toscano, G., Coello, C., and Landa, R. (2013). A ranking method based on the R2 indicator for many-objective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1523–1530.

[19] Eiben, A. and Smith, J. (2015). *Introduction to Evolutionary Computing*. Springer.

[20] Fleischer, M. (2003). The Measure of Pareto Optima Applications to Multi-objective Metaheuristics. In *Evolutionary Multi-Criterion Optimization*, pages 519–533, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

[22] Havbro, M. (2012). *Statistics and Probability Theory. In Pursuit of Engineering Decision Support*. Springer.

[23] Hernández, R. and Coello, C. (2015). Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 679–686, New York, NY, USA. ACM.

[24] Hernández, R. and Coello, C. (2017). A Hyper-heuristic of Scalarizing Functions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 577–584, New York, NY, USA. ACM.

[25] Hughes, E. (2008). Fitness Assignment Methods for Many-Objective Problems. In Knowles, J., Corne, D., Deb, K., and Chair, D., editors, *Multiobjective Problem Solving from Nature: From Concepts to Applications*, pages 307–329. Springer Berlin Heidelberg, Berlin, Heidelberg.

[26] Ikeda, K., Kita, H., and Kobayashi, S. (2001). Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal? In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 2, pages 957–962 vol. 2.

[27] Ishibuchi, H., Akedo, N., and Nojima, Y. (2013). A Study on the Specification of a Scalarizing Function in MOEA/D for Many-Objective Knapsack Problems. In Nicosia, G. and Pardalos, P., editors, *Learning and Intelligent Optimization: 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers*, pages 231–246. Springer Berlin Heidelberg, Berlin, Heidelberg.

[28] Ishibuchi, H., Doi, K., and Nojima, Y. (2017). On the effect of normalization in moea/d for multi-objective and many-objective optimization. *Complex & Intelligent Systems*, 3(4):279–294.

[29] Ishibuchi, H. and Murata, T. (1998). A Multi-objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. *Trans. Sys. Man Cyber Part C*, 28(3):392–403.

[30] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009a). Adaptation of Scalarizing Functions in MOEA/D: An Adaptive Scalarizing Function-Based Multiobjective Evolutionary Algorithm. In *Evolutionary Multi-Criterion Optimization*, pages 438–452. Springer Berlin Heidelberg.

[31] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009b). Evolutionary Many-Objective Optimization by NSGA-II and MOEA/D with Large Populations. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 1758–1763.

[32] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2010). Simultaneous Use of Different Scalarizing Functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 519–526, New York, NY, USA. ACM.

[33] Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. (2016). Performance of

Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

[34] Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426.

[35] Ishibuchi, H., Yamane, M., Akedo, N., and Nojima, Y. (2012). Two-objective solution set optimization to maximize hypervolume and decision space diversity in multiobjective optimization. In *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, pages 1871–1876.

[36] Jain, H. and Deb, K. (2013). An Improved Adaptive Approach for Elitist Nondominated Sorting Genetic Algorithm for Many-Objective Optimization. Technical report, Indian Institute of Technology, Kanpur, India. Department of Mechanical Engineering.

[37] Jain, H. and Deb, K. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622.

[38] Jaszkiewicz, A. (2002). Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137(1):50 – 71.

[39] Jiang, S., Yang, S., Wang, Y., and Liu, X. (2018). Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 22(2):296–313.

[40] Jin, Y., Zhou, A., Zhang, Q., Sendhoff, B., and Tsang, E. (2008). Modeling Regularity to Improve Scalability of Model-Based Multiobjective Optimization Algorithms. In Knowles, J., Corne,

D., Deb, K., and Chair, D., editors, *Multiobjective Problem Solving from Nature: From Concepts to Applications*, pages 331–355. Springer Berlin Heidelberg, Berlin, Heidelberg.

[41] Ke, L., Zhang, Q., and Battiti, R. (2014). Hybridization of Decomposition and Local Search for Multiobjective Optimization. *IEEE Transactions on Cybernetics*, 44(10):1808–1820.

[42] Khan, S. and Engelbrecht, A. (2012). A fuzzy particle swarm optimization algorithm for computer communication network topology design. *Applied Intelligence*, 36(1):161–177.

[43] Khan, W. and Zhang, Q. (2010). MOEA/D-DRA with two crossover operators. In *2010 UK Workshop on Computational Intelligence (UKCI)*, pages 1–6.

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

[45] Landa, J. and Burke, E. (2004). Using Diversity to Guide the Search in Multi-objective Optimization. In C.Coello and Lamont, G., editors, *Applications of Multi-Objective Evolutionary Algorithms*, pages 727–751. World Scientific.

[46] Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.*, 10(3):263–282.

[47] Lee, H. (2014). *Foundations of Applied Statistical Methods*. Springer.

[48] Li, B., Li, J., Tang, K., and Yao, X. (2015a). Many-Objective Evolutionary Algorithms: A Survey. *ACM Comput. Surv.*, 48(1):13:1–13:35.

[49] Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015b). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.

[50] Li, M., Yang, S., and Liu, X. (2014). Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization. *IEEE Transactions on Cybernetics*, 44(12):2568–2584.

[51] Li, M., Yang, S., and Liu, X. (2016). Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):645–665.

[52] Li, M. and Yao, X. (2017). What Weights Work for You? Adapting Weights for Any Pareto Front Shape in Decomposition-based Evolutionary Multi-Objective Optimisation. *CoRR*, abs/1709.02679.

[53] Lin, S. (2014). Multi-objective vehicle refueling planning using mixed integer programming. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 677–681.

[54] Liu, H., Chen, L., Deb, K., and Goodman, E. (2017). Investigating the Effect of Imbalance Between Convergence and Diversity in Evolutionary Multiobjective Algorithms. *IEEE Transactions on Evolutionary Computation*, 21(3):408–425.

[55] Liu, H., Gu, F., and Zhang, Q. (2014). Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455.

[56] López, J. and Coello, C. (2015). Many-Objective Problems: Challenges and Methods. In Kacprzyk, J. and Pedrycz, W., editors, *Springer Handbook of Computational Intelligence*, pages 1033–1046. Springer Berlin Heidelberg, Berlin, Heidelberg.

[57] Luque, M., Miettinen, K., Ruiz, A., and Ruiz, F. (2012). A two-slope achievement scalarizing function for interactive multiobjective optimization. *Computers & Operations Research*, 39(7):1673 – 1681.

[58] Ma, X., Zhang, Q., Tian, G., Yang, J., and Zhu, Z. (2018). On Tchebycheff Decomposition Approaches for Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 22(2):226–244.

[59] Miettinen, K. (1994). *On the Methodology of Multiobjective Optimization with Applications*. PhD thesis, University of Jyväskylä. Department of Mathematics.

[60] Miettinen, K. and Mäkelä, M. (2002). On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24(2):193–213.

[61] Murata, T. and Gen, H. I. M. (2001). Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, pages 82–95, London, UK, UK. Springer-Verlag.

[62] Nikulin, Y., Miettinen, K., and Mäkelä, M. (2012). A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34(1):69–87.

[63] Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). MOEA/D with Adaptive Weight Adjustment. *Evol. Comput.*, 22(2):231–264.

[64] Sato, H. (2014). Inverted PBI in MOEA/D and Its Impact on the Search Performance on Multi and Many-objective Optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 645–652, New York, NY, USA. ACM.

[65] Sato, H. (2015). Analysis of Inverted PBI and Comparison with Other Scalarizing Functions in Decomposition Based MOEAs. *Journal of Heuristics*, 21(6):819–849.

[66] Sato, H., Aguirre, H., and Tanaka, K. (2007). Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., and Murata, T., editors, *Evolutionary Multi-Criterion Optimization: 4th International Conference,*

EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings, pages 5–20. Springer Berlin Heidelberg, Berlin, Heidelberg.

[67] Shi, J. and Zhang, Q. (2017). PPLS/D: Parallel Pareto Local Search based on Decomposition. *The Computing Research Repository (CoRR)*, abs/1709.09785.

[68] Smutnicki, C., Rudy, J., and Zelazny, D. (2014). Very Fast Non-Dominated Sorting. *Decision Making in Manufacturing and Services*, 8(1-2):13–23.

[69] Solow, A. and Polasky, S. (1994). Measuring biological diversity. *Environmental and Ecological Statistics*, 1(2):95–103.

[70] Tian, Y., Cheng, R., Zhang, X., Cheng, F., and Jin, Y. (2017). An Indicator Based Multi-Objective Evolutionary Algorithm with Reference Point Adaptation for Better Versatility. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

[71] Ulrich, T., Bader, J., and Thiele, L. (2010). Defining and Optimizing Indicator-Based Diversity Measures in Multiobjective Search. In Schaefer, R., Cotta, C., Kołodziej, J., and Rudolph, G., editors, *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, pages 707–717. Springer Berlin Heidelberg, Berlin, Heidelberg.

[72] Ulrich, T. and Thiele, L. (2011). Maximizing Population Diversity in Single-objective Optimization. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 641–648, New York, NY, USA. ACM.

[73] Wang, L., Zhang, Q., Zhou, A., Gong, M., and Jiao, L. (2016a). Constrained Subproblems in a Decomposition-Based Multiobjective Evolutionary Algorithm. *IEEE Transactions on Evolutionary Computation*, 20(3):475–480.

[74] Wang, R., Zhang, Q., and Zhang, T. (2016b). Decomposition-based algorithms using pareto adaptive scalarizing methods. *IEEE Transactions on Evolutionary Computation*, 20(6):821–837.

[75] Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., and Zhang, T. (2017). Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

[76] Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., and Zhang, T. (2018). Localized Weighted Sum Method for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):3–18.

[77] Wierzbicki, A. (1980). The Use of Reference Objectives in Multiobjective Optimization. In *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter, West Germany, August 20–24, 1979*, pages 468–486, Berlin, Heidelberg. Springer Berlin Heidelberg.

[78] Xiang, Y., Zhou, Y., Li, M., and Chen, Z. (2017). A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 21(1):131–152.

[79] Yang, S., Li, M., Liu, X., and Zheng, J. (2013). A Grid-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736.

[80] Yuan, Y. and Xu, H. (2015). Multiobjective Flexible Job Shop Scheduling Using Memetic Algorithms. *IEEE Transactions on Automation Science and Engineering*, 12(1):336–353.

[81] Yuan, Y., Xu, H., Wang, B., and Yao, X. (2016). A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37.

[82] Zapotecas, S., Derbel, B., Liefooghe, A., Brockhoff, D., Aguirre, H., and Tanaka, K. (2015). Injecting CMA-ES into MOEA/D. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 783–790, New York, NY, USA. ACM.

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

[84] Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *2009 IEEE Congress on Evolutionary Computation*, pages 203–208.

[85] Zhou, A. and Zhang, Q. (2016). Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 20(1):52–64.

[86] Zhu, C., Xu, L., and Goodman, E. D. (2016). Generalization of pareto-optimality for many-objective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 20(2):299–315.

[87] Zitzler, E. (2012). Evolutionary Multiobjective Optimization. In Rozenberg, G., Bäck, T., and Kok, J., editors, *Handbook of Natural Computing*, pages 871–904. Springer Berlin Heidelberg, Berlin, Heidelberg.

[88] Zitzler, E. and Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings*, pages 832–842, Berlin, Heidelberg. Springer Berlin Heidelberg.

[89] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative

case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Bibliography

[1] A. Trivedi and D. Srinivasan and K. Sanyal and A. Ghosh (2017). A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462.

[2] Asafuddoula, M., Ray, T., and Sarker, R. (2015). A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460.

[3] Bader, J. and Zitzler, E. (2011). Hype: An Algorithm for Fast Hypervolume-based Many-objective Optimization. *Evol. Comput.*, 19(1):45–76.

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

[5] Blank, J., Deb, K., and Roy, P. (2019). Investigating the Normalization Procedure of NSGA-III. In *Evolutionary Multi-Criterion Optimization*, pages 229–240. Springer International Publishing.

[6] Bradstreet, L., While, L., and Barone, L. (2010). A Fast Many-objective Hypervolume Algorithm using Iterated Incremental Calculations. In *IEEE Congress on Evolutionary Computation*, pages 1–8.

[7] Brockhoff, D., Wagner, T., and Trautmann, H. (2012). On the Properties of the R2 Indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 465–472, New York, NY, USA. ACM.

[8] Brockhoff, D., Wagner, T., and Trautmann, H. (2015). R2 Indicator-based Multiobjective Search. *Evol. Comput.*, 23(3):369–395.

[9] Cheng, R., Jin, Y., Olhofer, M., and Sendhoff, B. (2016). A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):773–791.

[10] Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., and Yao, X. (2017). A benchmark test suite for evolutionary many-objective optimization. *Complex & Intelligent Systems*, 3(1):67–81.

[11] Chiang, T. C. and Lai, Y. P. (2011). MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1473–1480.

[12] Coello, C., Lamont, G., and Veldhuizen, D. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2 edition.

[13] Das, I. and Dennis, J. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. on Optimization*, 8(3):631–657.

[14] Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd.

[15] Deb, K. and Jain, H. (2014). 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.

[16] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

[17] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. In Abraham, A., Jain, L., and Goldberg, R., editors,

Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, pages 105–145. Springer London, London.

[18] Díaz, A., Toscano, G., Coello, C., and Landa, R. (2013). A ranking method based on the R2 indicator for many-objective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1523–1530.

[19] Eiben, A. and Smith, J. (2015). *Introduction to Evolutionary Computing*. Springer.

[20] Fleischer, M. (2003). The Measure of Pareto Optima Applications to Multi-objective Metaheuristics. In *Evolutionary Multi-Criterion Optimization*, pages 519–533, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

[22] Havbro, M. (2012). *Statistics and Probability Theory. In Pursuit of Engineering Decision Support*. Springer.

[23] Hernández, R. and Coello, C. (2015). Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 679–686, New York, NY, USA. ACM.

[24] Hernández, R. and Coello, C. (2017). A Hyper-heuristic of Scalarizing Functions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 577–584, New York, NY, USA. ACM.

[25] Hughes, E. (2008). Fitness Assignment Methods for Many-Objective Problems. In Knowles, J., Corne, D., Deb, K., and Chair, D., editors, *Multiobjective Problem Solving from Nature: From Concepts to Applications*, pages 307–329. Springer Berlin Heidelberg, Berlin, Heidelberg.

[26] Ikeda, K., Kita, H., and Kobayashi, S. (2001). Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal? In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 2, pages 957–962 vol. 2.

[27] Ishibuchi, H., Akedo, N., and Nojima, Y. (2013). A Study on the Specification of a Scalarizing Function in MOEA/D for Many-Objective Knapsack Problems. In Nicosia, G. and Pardalos, P., editors, *Learning and Intelligent Optimization: 7th International Conference, LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers*, pages 231–246. Springer Berlin Heidelberg, Berlin, Heidelberg.

[28] Ishibuchi, H., Doi, K., and Nojima, Y. (2017). On the effect of normalization in moea/d for multi-objective and many-objective optimization. *Complex & Intelligent Systems*, 3(4):279–294.

[29] Ishibuchi, H. and Murata, T. (1998). A Multi-objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. *Trans. Sys. Man Cyber Part C*, 28(3):392–403.

[30] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009a). Adaptation of Scalarizing Functions in MOEA/D: An Adaptive Scalarizing Function-Based Multiobjective Evolutionary Algorithm. In *Evolutionary Multi-Criterion Optimization*, pages 438–452. Springer Berlin Heidelberg.

[31] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009b). Evolutionary Many-Objective Optimization by NSGA-II and MOEA/D with Large Populations. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 1758–1763.

[32] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2010). Simultaneous Use of Different Scalarizing Functions in MOEA/D. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, GECCO '10, pages 519–526, New York, NY, USA. ACM.

[33] Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. (2016). Performance of

Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

[34] Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426.

[35] Ishibuchi, H., Yamane, M., Akedo, N., and Nojima, Y. (2012). Two-objective solution set optimization to maximize hypervolume and decision space diversity in multiobjective optimization. In *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, pages 1871–1876.

[36] Jain, H. and Deb, K. (2013). An Improved Adaptive Approach for Elitist Nondominated Sorting Genetic Algorithm for Many-Objective Optimization. Technical report, Indian Institute of Technology, Kanpur, India. Department of Mechanical Engineering.

[37] Jain, H. and Deb, K. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622.

[38] Jaszkiewicz, A. (2002). Genetic local search for multi-objective combinatorial optimization. *European Journal of Operational Research*, 137(1):50 – 71.

[39] Jiang, S., Yang, S., Wang, Y., and Liu, X. (2018). Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 22(2):296–313.

[40] Jin, Y., Zhou, A., Zhang, Q., Sendhoff, B., and Tsang, E. (2008). Modeling Regularity to Improve Scalability of Model-Based Multiobjective Optimization Algorithms. In Knowles, J., Corne,

D., Deb, K., and Chair, D., editors, *Multiobjective Problem Solving from Nature: From Concepts to Applications*, pages 331–355. Springer Berlin Heidelberg, Berlin, Heidelberg.

[41] Ke, L., Zhang, Q., and Battiti, R. (2014). Hybridization of Decomposition and Local Search for Multiobjective Optimization. *IEEE Transactions on Cybernetics*, 44(10):1808–1820.

[42] Khan, S. and Engelbrecht, A. (2012). A fuzzy particle swarm optimization algorithm for computer communication network topology design. *Applied Intelligence*, 36(1):161–177.

[43] Khan, W. and Zhang, Q. (2010). MOEA/D-DRA with two crossover operators. In *2010 UK Workshop on Computational Intelligence (UKCI)*, pages 1–6.

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

[45] Landa, J. and Burke, E. (2004). Using Diversity to Guide the Search in Multi-objective Optimization. In C.Coello and Lamont, G., editors, *Applications of Multi-Objective Evolutionary Algorithms*, pages 727–751. World Scientific.

[46] Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.*, 10(3):263–282.

[47] Lee, H. (2014). *Foundations of Applied Statistical Methods*. Springer.

[48] Li, B., Li, J., Tang, K., and Yao, X. (2015a). Many-Objective Evolutionary Algorithms: A Survey. *ACM Comput. Surv.*, 48(1):13:1–13:35.

[49] Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015b). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.

[50] Li, M., Yang, S., and Liu, X. (2014). Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization. *IEEE Transactions on Cybernetics*, 44(12):2568–2584.

[51] Li, M., Yang, S., and Liu, X. (2016). Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):645–665.

[52] Li, M. and Yao, X. (2017). What Weights Work for You? Adapting Weights for Any Pareto Front Shape in Decomposition-based Evolutionary Multi-Objective Optimisation. *CoRR*, abs/1709.02679.

[53] Lin, S. (2014). Multi-objective vehicle refueling planning using mixed integer programming. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 677–681.

[54] Liu, H., Chen, L., Deb, K., and Goodman, E. (2017). Investigating the Effect of Imbalance Between Convergence and Diversity in Evolutionary Multiobjective Algorithms. *IEEE Transactions on Evolutionary Computation*, 21(3):408–425.

[55] Liu, H., Gu, F., and Zhang, Q. (2014). Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455.

[56] López, J. and Coello, C. (2015). Many-Objective Problems: Challenges and Methods. In Kacprzyk, J. and Pedrycz, W., editors, *Springer Handbook of Computational Intelligence*, pages 1033–1046. Springer Berlin Heidelberg, Berlin, Heidelberg.

[57] Luque, M., Miettinen, K., Ruiz, A., and Ruiz, F. (2012). A two-slope achievement scalarizing function for interactive multiobjective optimization. *Computers & Operations Research*, 39(7):1673 – 1681.

[58] Ma, X., Zhang, Q., Tian, G., Yang, J., and Zhu, Z. (2018). On Tchebycheff Decomposition Approaches for Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 22(2):226–244.

[59] Miettinen, K. (1994). *On the Methodology of Multiobjective Optimization with Applications*. PhD thesis, University of Jyväskylä. Department of Mathematics.

[60] Miettinen, K. and Mäkelä, M. (2002). On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24(2):193–213.

[61] Murata, T. and Gen, H. I. M. (2001). Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, pages 82–95, London, UK, UK. Springer-Verlag.

[62] Nikulin, Y., Miettinen, K., and Mäkelä, M. (2012). A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34(1):69–87.

[63] Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). MOEA/D with Adaptive Weight Adjustment. *Evol. Comput.*, 22(2):231–264.

[64] Sato, H. (2014). Inverted PBI in MOEA/D and Its Impact on the Search Performance on Multi and Many-objective Optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 645–652, New York, NY, USA. ACM.

[65] Sato, H. (2015). Analysis of Inverted PBI and Comparison with Other Scalarizing Functions in Decomposition Based MOEAs. *Journal of Heuristics*, 21(6):819–849.

[66] Sato, H., Aguirre, H., and Tanaka, K. (2007). Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., and Murata, T., editors, *Evolutionary Multi-Criterion Optimization: 4th International Conference,*

EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings, pages 5–20. Springer Berlin Heidelberg, Berlin, Heidelberg.

[67] Shi, J. and Zhang, Q. (2017). PPLS/D: Parallel Pareto Local Search based on Decomposition. *The Computing Research Repository (CoRR)*, abs/1709.09785.

[68] Smutnicki, C., Rudy, J., and Zelazny, D. (2014). Very Fast Non-Dominated Sorting. *Decision Making in Manufacturing and Services*, 8(1-2):13–23.

[69] Solow, A. and Polasky, S. (1994). Measuring biological diversity. *Environmental and Ecological Statistics*, 1(2):95–103.

[70] Tian, Y., Cheng, R., Zhang, X., Cheng, F., and Jin, Y. (2017). An Indicator Based Multi-Objective Evolutionary Algorithm with Reference Point Adaptation for Better Versatility. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

[71] Ulrich, T., Bader, J., and Thiele, L. (2010). Defining and Optimizing Indicator-Based Diversity Measures in Multiobjective Search. In Schaefer, R., Cotta, C., Kołodziej, J., and Rudolph, G., editors, *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I*, pages 707–717. Springer Berlin Heidelberg, Berlin, Heidelberg.

[72] Ulrich, T. and Thiele, L. (2011). Maximizing Population Diversity in Single-objective Optimization. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, pages 641–648, New York, NY, USA. ACM.

[73] Wang, L., Zhang, Q., Zhou, A., Gong, M., and Jiao, L. (2016a). Constrained Subproblems in a Decomposition-Based Multiobjective Evolutionary Algorithm. *IEEE Transactions on Evolutionary Computation*, 20(3):475–480.

[74] Wang, R., Zhang, Q., and Zhang, T. (2016b). Decomposition-based algorithms using pareto adaptive scalarizing methods. *IEEE Transactions on Evolutionary Computation*, 20(6):821–837.

[75] Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., and Zhang, T. (2017). Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

[76] Wang, R., Zhou, Z., Ishibuchi, H., Liao, T., and Zhang, T. (2018). Localized Weighted Sum Method for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):3–18.

[77] Wierzbicki, A. (1980). The Use of Reference Objectives in Multiobjective Optimization. In *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter, West Germany, August 20–24, 1979*, pages 468–486, Berlin, Heidelberg. Springer Berlin Heidelberg.

[78] Xiang, Y., Zhou, Y., Li, M., and Chen, Z. (2017). A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 21(1):131–152.

[79] Yang, S., Li, M., Liu, X., and Zheng, J. (2013). A Grid-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736.

[80] Yuan, Y. and Xu, H. (2015). Multiobjective Flexible Job Shop Scheduling Using Memetic Algorithms. *IEEE Transactions on Automation Science and Engineering*, 12(1):336–353.

[81] Yuan, Y., Xu, H., Wang, B., and Yao, X. (2016). A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(1):16–37.

[82] Zapotecas, S., Derbel, B., Liefooghe, A., Brockhoff, D., Aguirre, H., and Tanaka, K. (2015). Injecting CMA-ES into MOEA/D. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 783–790, New York, NY, USA. ACM.

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

[84] Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *2009 IEEE Congress on Evolutionary Computation*, pages 203–208.

[85] Zhou, A. and Zhang, Q. (2016). Are All the Subproblems Equally Important? Resource Allocation in Decomposition-Based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 20(1):52–64.

[86] Zhu, C., Xu, L., and Goodman, E. D. (2016). Generalization of pareto-optimality for many-objective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 20(2):299–315.

[87] Zitzler, E. (2012). Evolutionary Multiobjective Optimization. In Rozenberg, G., Bäck, T., and Kok, J., editors, *Handbook of Natural Computing*, pages 871–904. Springer Berlin Heidelberg, Berlin, Heidelberg.

[88] Zitzler, E. and Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings*, pages 832–842, Berlin, Heidelberg. Springer Berlin Heidelberg.

[89] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative

case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.