Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco
Departamento de Computación

Técnicas de preservación de diversidad en problemas de optimización multiobjetivo

T E S I S

Que presenta

Amín Vanya Bernabé Rodríguez

Para obtener el grado de

Doctor en Ciencias

En la especialidad de

Computación

Director de la Tesis:

Dr. Carlos Artemio Coello Coello

Ciudad de México Abril, 2024

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Zacatenco Campus
Computer Science Department

Diversity maintenance techniques for multi-objective optimization problems

THESIS

Submitted by

Amín Vanya Bernabé Rodríguez

as the fulfillment of the requirement for the degree of

Ph.D. in Computer Science

Advisor

Dr. Carlos Artemio Coello Coello

Mexico City April, 2024

Resumen

Una de las técnicas más comúnmente usadas para resolver problemas de optimización multi-objetivo (POMs) son los llamados algoritmos evolutivos multi-objetivo (AEMOs), los cuales son metaheurísticas poblacionales que emulan el proceso evolutivo biológico. Los AEMOs usan el principio de “supervivencia del más apto”, de la teoría de Darwin para resolver POMs de alta complejidad. La población utilizada por los AEMOs presenta la ventaja de permitir, con una manipulación adecuada, procesar un conjunto de soluciones óptimas a cada iteración, en vez de operar con una sola a la vez, como ocurre con las técnicas de programación matemática.

Sin embargo, el uso de una población hace también que los AEMOs requieran un mecanismo que les permita mantener un cierto grado de diversidad durante el proceso evolutivo. De no mantener adecuadamente la diversidad, los resultados producidos por un AEMO serán de muy baja calidad o se podría producir convergencia prematura. Esto ha motivado el desarrollo de una amplia variedad de técnicas de preservación de diversidad. Estas técnicas pueden ser clasificadas de forma general en tres categorías: (1) estimadores de densidad, (2) restricciones a la cruza y (3) poblaciones secundarias.

Los POMs con cuatro o más objetivos, también conocidos como problemas de optimización con muchos objetivos, añaden nuevos desafíos a los AEMOs, principalmente debido a que el incremento en el número de objetivos conlleva un incremento exponencial en el tamaño del espacio de búsqueda. Consecuentemente, esto genera una disminución en la presión de selección en los AEMOs basados en dominancia de Pareto, y también produce un costo computacional que puede volverse prohibitivo en algunos casos (p-ej., cuando se usa un mecanismo de selección basado en el hipervolumen). Adicionalmente, existen aún diferentes áreas sin explorar en torno a la preservación de diversidad, específicamente para problemas de alta dimensionalidad.

El trabajo presentado en esta tesis tiene el objetivo de contribuir al estado del arte de técnicas de preservación de diversidad en problemas de optimización con muchos objetivos. El trabajo presentado en esta tesis se concentra en dos áreas principales de investigación. La primera, concerniente al uso de restricciones a la cruza basadas en energía-s, propone una nueva y útil técnica de preservación de diversidad basada en un conjunto de estas restricciones. Nuestros resultados experimentales muestran que el uso

de un indicador de diversidad (energía-s) en una técnica de preservación de diversidad (restricciones a la cruza) produce una mejora en problemas de optimización con muchos objetivos. Por otro lado, investigamos el uso de evolución gramatical para producir nuevos componentes de un AEMO. Específicamente, generamos nuevas funciones de escalarización (usadas en AEMOs basados en descomposición) y nuevas aproximaciones del hipervolumen (usadas en AEMOs basados en indicadores). Todos los componentes que generamos con este método producen mejores resultados al ser comparados contra alternativas del estado del arte. Esto indica que el uso de evolución gramatical para generar componentes de un AEMO es una área de investigación prometedora, y que es una alternativa viable para generar nuevas funciones de escalarización, aproximaciones del hipervolumen, y potencialmente, nuevas técnicas de preservación de diversidad (p.ej., nuevos estimadores de densidad).

Abstract

One of the most commonly used techniques for solve multi-objective optimization problems (MOPs) are the so-called multi-objective evolutionary algorithms (MOEAs), which are population-based metaheuristics that emulate the evolutionary process that occurs in Nature. MOEAs use the "survival of the fittest" principle from Darwin's evolutionary theory to solve high complexity MOPs. The population adopted by MOEAs presents the advantage of allowing, with a proper manipulation, the processing of a set of optimal solutions at each iteration, instead of operating with one solution at a time, as occurs with mathematical programming techniques.

However, the use of a population, also makes MOEAs to require a mechanism that allows them to keep a certain level of diversity during the evolutionary process. If diversity is not properly maintained, the results produced by a MOEA will be of very low quality or premature convergence may occur. This has motivated the development of a wide variety of techniques to maintain diversity. Such techniques can classified, in general, in three categories: (1) density estimators, (2) mating restrictions and (3) secondary populations.

MOPs having four or more objectives, which are also known as many-objective optimization problems (MaOPs), add new challenges to MOEAs, mainly because the increase in the number of objectives involves an exponential increase in the size of the search space. Consequently, this generates a decrease in the selection pressure of MOEAs based on Pareto dominance and also produces a computational cost that may become prohibitive in some cases (e.g., when using a selection mechanism based on the hypervolume). Additionally, there are still several unexplored research areas around diversity maintenance, specifically for MaOPs.

The work presented in this thesis has as its main goal to contribute to the state of the art in diversity maintenance techniques in MaOPs. The work presented here focuses on two main research areas. The first is related to the use of mating restrictions adopting s-energy, in which we propose a new and useful diversity maintenance technique based on the use of a set of mating restrictions. Our experimental results show that the use of a diversity indicator (s-energy) in a diversity maintenance technique (mating restrictions) produces an improvement in MaOPs. On the other hand, we also studied the use of grammatical evolution to produce new components of a MOEA. Specifically, we generated new scalarizing functions (used in

MOEAs based on decomposition) and new hypervolume approximations (used in MOEAs based on performance indicators). All the components that we generated using this approach are able to produce better results when compared with alternative approaches from the state of the art. This indicates that the use of grammatical evolution to generate components of a MOEA is a promising research area, and that this is indeed a viable choice to generate new scalarizing functions, hypervolume approximation and, potentially, new diversity maintenance techniques (e.g., new density estimators).

Agradecimientos

Primeramente quiero agradecer a mi familia: a mi madre, Débora, quien en medio de las circunstancias más cambiantes ha estado ahí para mí; y a mis hermanas, Bet y Deni, por ayudarme y aconsejarme siempre que lo necesité.

Un enorme agradecimiento a mi asesor, el Dr. Carlos Coello, por guiarme y aconsejarme en cada etapa de este doctorado. Nada de esto habría sido posible sin gran trabajo, disposición y toda la orientación que me brindó.

Agradezco a los revisores de esta tesis, Dr. Luis Gerardo de la Fraga, Dr. Amilcar Meneses, Dra. Xiaoou Li, Dr. Carlos Hernández y Dr. Victor Sosa por el tiempo invertido y sus enriquecedoras observaciones.

Gracias al personal del departamento por todo el apoyo que me dieron durante mi permanencia en el mismo: Katia Ocampo, Erika Ríos, Jose Luis Flores, Santiago Domínguez y Arcadio Morales.

Y muy especiales agradecimientos a las personas que estuvieron más cerca de mí estos años, y sin quienes no sé cómo habría concluido esta etapa. Fernanda, gracias por estar en todo momento. Y Santiago, gracias por acompañarme, y por ayudarme a crecer en tantas áreas.

Finalmente, agradezco al CINVESTAV y a CONAHCYT por los apoyos económicos otorgados durante la realización de mi doctorado.

Contents

List of Figures

List of Tables

1

Introduction

Multi-objective optimization problems (MOPs) can model a variety of real-world problems found in many different fields of knowledge. This has generated a great interest in developing techniques to solve such problems. In contrast to single-objective optimization, MOPs have no single optimal solution. Instead, in this case a set of solutions representing the best possible trade-offs among the objective functions is sought.

Over the years, a variety of mathematical programming techniques have been developed to solve MOPs [84]. However, these techniques have several limitations in practice. For example, they can't be applied when the Pareto front is disconnected or when there are multiple false Pareto fronts (also known as multifrontality). Additionally, they often return a single Pareto optimal solution per run, meaning that several executions are required to obtain an approximation of the Pareto front. The aforementioned limitations of mathematical programming techniques have motivated the use of evolutionary algorithms to solve MOPs [24].

Evolutionary algorithms present several advantages in the process of solving MOPs. One of the most important advantages is that they are population-based stochastic search techniques. This means that they operate on a set of solutions which, through the iterative evolutionary process, can produce a set of multiple Pareto optimal solutions in a single execution.

In addition to this, evolutionary algorithms are less dependant on the shape or continuity of the Pareto front. Also, they do not require the objective functions to be differentiable, which is the case for some mathematical programming techniques. In fact, evolutionary algorithms do not even require the objective functions to be an analytical expression (e.g., they can be the outcome of a simulation). These algorithms are known as multi-objective evolutionary algorithms (MOEAs) and they have become more and more important in the past ten years [31, 26, 24]. The first MOEA was presented in 1984 [104, 103], and ever since, multiple different algorithms have been developed.

1.1 Motivation

MOEAs can be classified into different categories, depending on the baseline mechanism they use to determine which are the best solutions. The most well known categories are: Pareto dominance-based, decomposition-based and indicator-based MOEAs. However, regardless of the category they belong to, one of the most important aspects of a MOEA is diversity, since the lack of diversity can lead to a poor approximation or to premature convergence. Hence, modern MOEAs incorporate at least one diversity maintenance technique. These techniques can be classified in three broad categories: (1) density estimators, (2) mating restrictions and (3) secondary populations.

The term diversity is usually used to measure two features of a given population: (1) solutions should be well spread across all objectives, usually covering the extreme points and (2) solutions should have a uniform distribution. Diversity has been studied since the early days of evolutionary computation, but there are still few studies centering on diversity in the context of multi-objective optimization, and even less in the context of many-objective optimization (which refers to MOPs with four or more objective functions).

The increase of objective functions present in many-objective optimization introduces some additional challenges to MOEAs, particularly in dominance-based MOEAs. This occurs because each additional objective increases the size of the search space, allowing more solutions to become non-dominated, which is the main selection criterion in these algorithms. Thus, the selection pressure of dominance-based MOEAs becomes significantly reduced, producing poor results. Additionally, some diversity maintenance techniques may be infeasible to use in this type of problems due to lack of scalability, or because they become too computationally expensive.

1.2 Problem statement

Even though there exists a large number of diversity maintenance techniques, many of them have been originally designed considering two and three objective optimization problems. And, even though there are some proposals in the context of diversity maintenance specifically for many-objective optimization, there is a significant number of areas that remain to be explored. For instance, there are performance indicators such as s-energy which can be used to measure diversity in a population, and their use in diversity maintenance techniques has not been explored. On the other hand, there are evolutionary computation algorithms that could be used to generate new elements that could improve MOEAs' performance, such as the generation of new density estimators.

1.3 Objectives

1.3.1 Research aim

To contribute to the advance of the state of the art in diversity maintenance techniques used in multi-objective evolutionary optimization, particularly in many-objective optimization problems.

1.3.2 Research objectives

- To study already developed diversity maintenance techniques.
- To propose a new diversity maintenance technique that can be used in many-objective optimization problems.
- To develop a new algorithm based on an approach focused on diversity improvement.
- To analyze the potential of grammatical evolution to develop a new density estimator for many-objective problems.

1.4 Contributions

The following are the most important contributions that we achieved during our research:

- **S-energy based mating restrictions.** We proposed three different mating restrictions based on the s-energy performance indicator and tested their performance in problems with 2, 3 and 5 objectives. We presented this work at the *2021 IEEE Congress on Evolutionary Computation* [97].

- **Ensemble of s-energy based mating restrictions.** As a follow up of the previous work done on mating restrictions, we proposed an ensemble of such restrictions and explored its performance in problems with up to 7 objectives. We presented the obtained results at the 2021 IEEE Symposium Series on Computational Intelligence (SSCI) [98].

- **Generation of new scalarizing functions using genetic programming.** We implemented a hybrid implementation of genetic programming and a MOEA to generate new scalarizing functions. Using this implementation we were able to generate new scalarizing functions with some favorable results which were presented at the sixteenth international Conference on Parallel Problem Solving from Nature (PPSN XVI) [96]. A follow up work generating new scalarizing functions for MOPs with up to 7 objectives both in standard and inverse benchmark problems was presented in the Grammatical Evolution Workshop (GEWS2023) held during the Conference on Genetic and Evolutionary Computation (GECCO 2023) [99].

- **Generation of hypervolume approximations using genetic programming.** We modified the genetic programming implementation used in the previous work to generate different hypervolume approximations for 2-5 objectives and compared them against some state-of-the-art hypervolume approximations. We published this work at the Swarm and Evolutionary Computation journal [95].

1.5 List of publications

1.5.1 Conference papers

- Bernabé Rodríguez, A. V. and Coello Coello, C. A., Generation of New Scalarizing Functions Using Genetic Programming. In *Parallel Problem Solving from Nature – PPSN XVI*, Leiden, The Netherlands.

- Bernabé Rodríguez, A. V. and Coello Coello, C. A., An Empirical Study on the Use of the S-energy Performance Indicator in Mating Restriction Schemes for Multi-Objective Optimizers. In *2021 IEEE Congress on Evolutionary Computation (CEC'2021)*, Krákow, Poland. ISBN 978-1-7281-8393-0.

- Bernabé Rodríguez, A. V. and Coello Coello, C. A., An Ensemble of S-energy Based Mating Restricti ons for Multi-Objective Evolutionary Algorithms. In *IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2021)*, Orlando, USA.

- Bernabé Rodríguez, A. V. and Coello Coello, C. A., Designing Scalarizing Functions Using Grammatical Evolution In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation (GECCO 2023 Companion)*, Lisbon, Portugal.

1.5.2 Journal publication

- Bernabé Rodríguez, A. V., Alejo-Cerezo, B. I., and Coello Coello, C. A., Improving Multi-Objective Evolutionary Algorithms using Grammatical Evolution. In *Swarm and Evolutionary Computation* 84:101434, 2024.

1.6 Structure of this document

This thesis consists of 8 chapters, including this first one. Next, we describe briefly the contents of each of the following chapters.

In Chapter 2 we present the essential background needed to discuss our work. We provide the definition of multi-objective optimization problems as well as an overview of evolutionary algorithms, with a particular emphasis on multi-objective evolutionary algorithms. At the end of this chapter we also present some indicators used to assess the performance of such algorithms.

In Chapter 3 we define diversity in the context of evolutionary optimization and discuss its relevance to MOEAs. We also review some of the diversity maintenance techniques used in the area.

In Chapter 4 we present some new mating restrictions which are based in a diversity indicator and we evaluate their performance experimentally.

In Chapter 5 we propose an ensemble of the mating restrictions proposed in the previous chapter. This ensemble is able to adapt the number of individuals paired with each of the mating restrictions at each generation.

In Chapter 6 we explain how we used grammatical evolution (an evolutionary computation technique) to generate new scalarizing functions (commonly used in decomposition-based MOEAs).

In Chapter 7 we employ a modification of the grammatical evolution implementation presented in the previous chapter, but with the purpose of generating hypervolume approximations.

Finally, in Chapter 8 we present our conclusions and some future research paths.

2

Background

This chapter provides some essential concepts needed to discuss the work presented in this document. In Section 2.1 we present some basic notions of evolutionary algorithms. Then, in Section 2.2 we introduce the definition of multi-objective optimization. Next, we discuss the use of evolutionary algorithms to solve multi-objective optimization problems in Section 2.3. Finally, in Section 2.4 we define some of the most popular performance indicators used to assess the quality of the results obtained in this type of algorithms.

2.1 Evolutionary Algorithms

Evolutionary Algorithms take inspiration from the evolutionary modern synthesis, which is the result of merging Charles Darwin's theory of evolution with Gregor Mendel's notions of heredity. Broadly, this theory states that evolution occurs due to changes in the genetic material of individuals through generations, which are guided by natural selection. The two main mechanisms that generate these changes are mutation and recombination (also called crossover).

Biologically, the genetic material of each organism is contained in the deoxyribonucleic acid (DNA), where all the information needed to build cell components is stored. DNA is structured in **chromosomes**, which are long sub-segments of itself. In turn, chromosomes are divided in multiple **genes**, which are small segments of DNA that encode a defined biochemical function. Additionally, in a pair of homologous chromosomes, the value that a certain genetic position can acquire is called **allele**.

DNA contains all the genetic information of an organism. However, this does not imply that all the information contained in the DNA will be used at a given time during the development of such organism. We call **genotype** to the ensemble of all genetic material of an individual. Whereas the information that is actually used (or expressed) and turns into observable features of the individual is known as **phenotype**.

Evolutionary algorithms emulate these biological structures and the processes involving their manipulation with the goal of solving certain types of problems. There are different ways of performing this emulation, as well as different levels of abstraction, which yields different evolutionary techniques. However, in spite of their differences, all evolutionary algorithms make an analogy between an evolutionary process and the solution of a problem. Also, there are some common elements in most of these algorithms, such as:

- **Individual:** It is the representation of a solution. It is usually a data structure with multiple parameters that model a simplified version of a biological organism.
- **Population:** Set of individuals that can interact with each other. It is usually generated at random at the beginning of the algorithm, and obtained through genetic operators in the subsequent iterations.
- **Generation:** It is a single iteration of the main algorithm.
- **Fitness:** It is a value that quantifies the quality of an individual as a potential solution of the problem being solved. It is one of the core elements of evolutionary algorithms, since it allows to determine if an individual is desirable or not.
- **Genetic operators:** They are the mechanisms used to modify the genetic material of the individuals.
  - **Crossover:** It allows to form new individuals from the information of other parent individuals.
  - **Mutation:** It alters the genetic material of an individual. These alteration are usually small.
- **Selection mechanism:** It is the process that determines which individuals of the population will be recombined using the crossover operator. It usually contains a stochastic component but is still guided by the fitness values of the individuals.

There are three main paradigms in evolutionary computation. The main differences between them are the genetic operators used, the way in which they are used at each generation, and the parameters required. In the following subsections we discuss briefly each of these paradigms.

2.1.1 EVOLUTION STRATEGIES

Evolution Strategies (ESs) are an evolutionary technique developed in 1964 at the Technical University of Berlin [4]. It its original version, the main genetic operator is mutation, while crossover is a secondary operator. An ES can be denoted with the following notation:

$$(\mu/\rho \overset{+}{,} \lambda) - ES,$$

where μ, ρ and λ are positive integers. At each generation, there is a population of μ parent individuals, from which ρ individuals are chosen to apply a recombination operator, generating a new population of λ offspring individuals. Then, fitness values are obtained for each of the new individuals to determine which will be preserved in the next generation. This process is performed using some selection operator. The two most common selection operators are:

- **"+" selection** The new parents population will be created selecting the best μ individuals from the set formed by the union of parents and offspring in the current generation. This allows that the parents that are not improved by the offspring are preserved in the next generation, instead of some of the offspring individuals. In this case, if an individual is not improved by the offspring, it can exist in the population for several generations.

- **","** selection The new parents population will be created selecting only the best μ individuals from the λ offspring individuals. In this case, $\lambda \geq \mu$, and all individuals will exist in the population for only one generation.

Usually, the ρ value is not specified in the notation of the ES, simplifying it to:

$$(\mu \overset{+}{,} \lambda) - ES.$$

The simplest ES is denoted by $(1 + 1) - ES$, meaning that there is one parent individual and one offspring individual. At each generation, the offspring is generated by applying a mutation operator to the parent. Then, the offspring fitness is evaluated, if it is greater than the parent's fitness, the parent is replaced by its offspring. Otherwise, the offspring is discarded.

A basic mutation operator works as follows. Given a parent $\bar{x}^t = \left(x_1, x_2, \cdots, x_n\right)^T$, the offspring is generated using:

$$\bar{x}^{t+1} = \bar{x}^t + N(0, \bar{\sigma}),$$

where t is the number of the current generation, and $N(0, \bar{\sigma})$ is a normally distributed random variable with mean 0 and standard deviation 1.

2.1.2 Evolutionary Programming

Evolutionary Programming (EP) was proposed by Fogel, Owens and Walsh in the 60s [42]. It was originally proposed to solve prediction problems. To achieve this, the algorithm is provided a training set, formed by input and output values. Then, the EP looks for a program that for each input value generates the corresponding output value defined in the training set.

Similarly to ESs, the main genetic operator is the mutation. However, EP's most particular feature is that the individuals are represented as finite-state machines. A finite-state machine is a model where, given an input signal, generates an output signal by transitioning through different internal states. These models are defined by a finite alphabet, a state-transition function, and a finite set of states, which must include an initial state and a set of final states.

The mutation performed in EP usually allows to change almost all the components of the individuals, such as the number of states, the state-transition function, the initial states or the final states. Additionally, the best half of the population is usually preserved, at each generation, whereas the remaining half is formed with the best offspring individuals generated through mutation.

2.1.3 Genetic Algorithms

Genetic Algorithms (GAs) were first proposed by John H. Holland in 1962 [56]. They are mainly used to solve optimization problems, where a mathematical function (called objective function) needs to be either maximized or minimized.

Taking into consideration that individuals in evolutionary algorithms emulate living organisms, the individuals in GAs usually emulate the following biological concepts:

- **Phenotype:** The values that are evaluated in the objective function create the phenotype. For instance, given a function with 2 input variables $x_1, x_2 \in \mathbb{R}$ in the interval $[0, 1]$, the phenotype of one individual could be $(0.25, 0.75)$.
- **Genotype:** It is formed by the phenotype values but encoded in a different way. Traditionally, binary values are used. For instance, the previous values could be encoded with the strings $(00100101, 01110101)$.
- **Chromosome:** It is the string that encodes all the individual's variables. Following the same example, the chromosome would be the concatenation of both variables' strings, resulting in the string 0010010101110101.
- **Gene:** It is the string that encodes each of the individual's variables. The example chromosome is formed by two genes: 00100101 and 01110101.
- **Allele:** It is the possible values that can be assigned to a certain position in a gene. Using binary encoding, each position can only have the values 0 or 1.

A generic GA usually performs the following steps:

1. Generate the initial population at random.
2. Select the individuals which will generate the offspring population. The selection mechanism is performed stochastically, but still considering the quality of each individual in terms of optimality (using its fitness value).
3. Generate a new population applying the crossover operator to the pairs of individuals selected in the previous step. Mutation is also used as a secondary genetic operator in the resulting individuals.

Steps 2 and 3 are repeated iteratively until termination the criterion is reached. The most common termination criteria are: a maximum number of iterations reached or a fitness threshold reached.

2.2 MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

In a great variety of real-world problems, it is necessary to simultaneously optimize two or more objective functions [112, 52]. These are known as

Multi-objective Optimization Problems (MOPs). Even though the problem may involve both maximization and minimization of certain objective functions, without loss of generality, we will assume only minimization in all the definitions used in this work.

2.2.1 MOP definition

Formally, a MOP is defined as follows:

$$\text{minimize } \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})] \tag{2.1}$$

subject to:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \ldots, p \tag{2.2}$$

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, q \tag{2.3}$$

where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, ..., k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, ..., p$, $j = 1, ..., q$ are the constraint functions of the problem.

2.2.2 Pareto Optimality

In order for a MOP to be non-trivial, the objective functions f_i must be in conflict with each other, causing that there is no single solution to a MOP. Instead of that, we attempt to generate a set of solutions that represent the best possible trade-offs among the objectives. Pareto dominance is commonly used to characterize such solutions.

Definition 1. Given two vectors $\vec{x}, \vec{y} \in \mathbb{R}^k$, we say that $\vec{x} \leq \vec{y}$ if $x_i \leq y_i$ for $i = 1, ..., k$, and that $\vec{x}$ **dominates** $\vec{y}$ (denoted by $\vec{x} \prec \vec{y}$) if $\vec{x} \leq \vec{y}$ and $\vec{x} \neq \vec{y}$.

Definition 2. We say that a vector of decision variables $\vec{x} \in \mathcal{X} \subset \mathbb{R}^n$ is **non-dominated** with respect to $\mathcal{X}$, if there does not exist another $\vec{x}' \in \mathcal{X}$ such that $\vec{f}(\vec{x}') \prec \vec{f}(\vec{x})$.

Definition 3. We say that a vector of decision variables $\vec{x}^* \in \mathcal{F} \subset \mathbb{R}^n$ ($\mathcal{F}$ is the feasible region) is **Pareto-optimal** if it is non-dominated with respect to $\mathcal{F}$.

Definition 4. The **Pareto Optimal Set** $\mathcal{P}^*$ is defined by:

$$\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ is Pareto-optimal}\}$$

Definition 5. The **Pareto Front** $\mathcal{PF}^*$ is defined by:

$$\mathcal{PF}^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k | \vec{x} \in \mathcal{P}^*\}$$

Then, when solving a given MOP, our goal is to find the Pareto optimal set ($\mathcal{P}^*$) from the feasible region ($\mathcal{F}$) of all the decision variable vectors that satisfy (2.2) and (2.3).

Note however that in practice, not all the Pareto optimal set is normally desirable or achievable, and decision makers tend to prefer certain types of solutions or regions of the Pareto front [12].

Additionally, the Pareto fronts may have different geometries such as linear, concave, convex or combinations of them. Also, there are some degenerate fronts, which are of a lower dimension than the objective space in which they are embedded [61].

2.3 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

The use of evolutionary algorithms to solve multi-objective optimization problems has become increasingly popular in the last decades. The so-called Multi-Objective Evolutionary Algorithms (MOEAs) present some important advantages compared to classical mathematical programming techniques, from which, perhaps the most remarkable is that MOEAs operate on a set of solutions (called population). This allows MOEAs (if properly manipulated) to generate several Pareto optimal solutions in a single execution, which contrasts with mathematical programming techniques, which normally generate a single Pareto optimal solution per execution. Additionally, MOEAs require little domain-specific information and do not impose requirements neither on the objective functions nor on the constraints of a MOP, contrasting with mathematical programming techniques which usually do (e.g., the gradient of the functions or the need for them to be expressed in algebraic form) [21].

There is a wide variety of MOEAs in the specialized literature, but they can be broadly classified into the 3 following categories: [116]:

1. **Pareto-based MOEAs**: These algorithms were developed during the 1990s, and use a ranking procedure (called non-dominated sorting) based on Pareto optimality to classify solutions. They also adopt a mechanism responsible for maintaining diversity (which is called density estimator). These MOEAs were very popular for several years, being the Non-dominated Sorting Genetic Algorithm for Multi-objective

Optimization (NSGA-II) [30] one of the most popular MOEAs, but their use is not effective in MOPs having more than three objectives (the so-called many-objective optimization problems). This is because the number of non-dominated solutions grows exponentially with the number of objectives, and this quickly dilutes the selection pressure [124].

2. **Indicator-based MOEAs:** In these algorithms, the idea is to use a performance indicator to select solutions instead of using Pareto optimality [140]. The most representative of these algorithms is the S-metric Selection Evolutionary Multi-objective Algorithm (SMS-EMOA) [7], which aims to maximize the hypervolume value of its population. The main advantage of using hypervolume is that it is the only performance indicator currently known to be fully Pareto compliant (i.e., strictly monotonic with respect to Pareto optimality). However, its main drawback is its high computational cost, which has led to explore alternatives such as the use of hypervolume approximations instead of the exact hypervolume [67, 6, 82]. Although indicator-based MOEAs based on $R2$ (which is weakly Pareto compliant) are computationally efficient and have a good performance, their use is not very popular today. Approaches based on $R2$ use a scalarizing function and their performance is sensitive to the specific scalarizing function adopted [55].

3. **Decomposition-based MOEAs:** The idea of using decomposition (or scalarization) methods was originally proposed in mathematical programming more than 25 years ago [28] and it consists in transforming the given MOP into several single-objective optimization problems (SOPs) which are then solved to generate the non-dominated solutions of the original MOP. Scalarizing functions are used to perform this transformation by aggregating the multiple objective functions into a single one using weight vectors. This allows to easily incorporate preference information if needed. Linear scalarizing functions, such as the Weighted Sum Function (which is the most commonly used, due to its simplicity), cannot generate solutions in the concave portion of the Pareto Front [22]. However, non-linear scalarizing functions allow these MOEAs the generation of non-convex portions of the Pareto front and work even in disconnected Pareto fronts. Furthermore, there's evidence that shows that the performance of these algorithms depends strongly on the scalarizing function used [54]. Another advantage is that, unlike with domination-based MOEAs, de-

composition techniques are not easily affected by selection pressure issues [124], which leads to a better performance in many-objective problems. The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), introduced in 2007 [130] presents an important advantage with respect to methods proposed in the mathematical programming literature (such as Normal Boundary Intersection (NBI) [28]): it uses neighborhood search to solve simultaneously all the SOPs generated from the transformation.

2.4 Performance Assessment

In order to assess the quality of the solutions obtained using a MOEA (or any other MOP solving technique) there are different criteria that can be measured [139]:

1. Convergence: The distance between the approximation set and the true Pareto front should be minimized.

2. Distribution: A uniform distribution of the solutions is desirable.

3. Spread: The extent of the approximation set should be maximized, i.e., for each objective, the solutions should cover a wide range of values.

Ideally, a good approximation set to a given MOP has good convergence, distribution and spread. Performance indicators are used to quantitatively measure these properties allowing the comparison of different algorithms. In the following subsections, we define some of these indicators, as they will be used later in this document to compare our results against other state-of-the-art algorithms.

In the following definitions we mention the use of a set of points $\mathcal{A}$, which represents the approximation set of which we want to obtain its indicator value. Additionaly, some indicators require the use of a reference set $\mathcal{B}$. This reference set is usually a discretization of the true Pareto front of the MOP being used. This means that this information must be available in order to use these indicators. Although this information is usually not known *a priori* in most real-world problems, it is available for most benchmark MOPs, which are commonly used to assess the performance of MOEAs.

2.4.1 Hypervolume Indicator

Let $\mathcal{A} = \{\vec{a}_1, \ldots, \vec{a}_n\} \in \mathbb{R}^m$ be a set of points in an $m-$dimensional space, and let $\vec{r} \in \mathbb{R}^m$ be a reference point which is dominated by every point in $\mathcal{A}$. Then, the set $H(\mathcal{A}, \vec{r})$ is formed by all the points that are dominated by at least one element in $\mathcal{A}$ and which also dominate $\vec{r}$:

$$H(\mathcal{A}, \vec{r}) = \{\vec{z} \in \mathbb{R}^m \mid \exists\, \vec{a} \in \mathcal{A} : \vec{a} \prec \vec{z} \prec \vec{r}\}. \tag{2.4}$$

The hypervolume indicator $I_H(\mathcal{A}, \vec{r})$ is defined as follows:

$$I_H(\mathcal{A}, \vec{r}) = \lambda(H(\mathcal{A}, \vec{r})), \tag{2.5}$$

where λ represents the Lebesgue measure [6]. In order to compare the hypervolume values of two different sets, the calculation must be made using the same reference point $\vec{r}$. The greater the hypervolume, the better the approximation set is. This indicator measures the convergence and the spread of the set.

2.4.2 Generational Distance Indicator

Let $\mathcal{A} = \{\vec{a}_1, \ldots, \vec{a}_n\} \in \mathbb{R}^m$ be a set of points and $B = \{\vec{b}_1, \ldots, \vec{b}_n\} \in \mathbb{R}^m$ be a reference set. Then, the Generational Distance Indicator is defined as follows:

$$I_{GD}(\mathcal{A}, \mathcal{B}) = \frac{1}{n}\left(\sum_{i=1}^{n} d(\vec{a}_i, \mathcal{B})^p\right)^{\frac{1}{p}} \tag{2.6}$$

where $p > 0$ is a user-defined parameter (usually $p = 2$ is used), and $d(\vec{a}, \mathcal{B})$ is the Euclidean distance from vector $\vec{a}$ to its nearest element from $\mathcal{B}$. This indicator only measures the convergence of the set. The smaller the generational distance, the better the convergence of the approximation set is.

2.4.3 Inverted Generational Distance Indicator

Let $\mathcal{A} = \{\vec{a}_1, \ldots, \vec{a}_n\} \in \mathbb{R}^m$ be a set of points and $B = \{\vec{b}_1, \ldots, \vec{b}_n\} \in \mathbb{R}^m$ be a reference set. Then, the Inverted Generational Distance Indicator is defined as follows:

$$I_{IGD}(\mathcal{A}, \mathcal{B}) = \frac{1}{n}\left(\sum_{i=1}^{n} d(\vec{b}_i, \mathcal{A})^p\right)^{\frac{1}{p}} \tag{2.7}$$

where $p > 0$ is a user-defined parameter (usually $p = 2$ is used), and $d(\vec{b}, \mathcal{A})$ is the Euclidean distance from vector $\vec{b}$ to its nearest element from $\mathcal{A}$. The only difference with respect to the previous indicator, is that the distances are measured from each element in the reference set $\mathcal{B}$ to each of the elements in the approximation set $\mathcal{A}$. This change improves the indicator significantly, since this allows the indicator to measure not only convergence, but also distribution and spread, given that the reference set has a good distribution and is well spread. Once again, the smaller the inverted generational distance, the better the approximation set is.

2.4.4 S-ENERGY INDICATOR

Let $\mathcal{A} = \{\vec{a}_1, \ldots, \vec{a}_n\} \in \mathbb{R}^m$ be a set of points, its s-energy is defined as follows:

$$I_{Se}(\mathcal{A}) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \frac{1}{\mid \vec{a}_i - \vec{a}_j \mid^s} \tag{2.8}$$

where $\mid \cdot \mid$ represents the Euclidean distance and $s > 0$ is a user-defined parameter (usually $s = m - 1$ is used). This indicator exclusively measures the uniformity of the solutions. The smaller the $s-$energy, the better the uniformity of the set is.

2.4.5 R2 INDICATOR

Let $\mathcal{A} = \{\vec{a}_1, \ldots, \vec{a}_n\} \in \mathbb{R}^m$ be a set of points, and $\mathcal{W}$ a set of $m-$dimensional weight vectors, the R2 indicator is defined as follows:

$$I_{R2}(\mathcal{A}, \mathcal{W}) = \frac{1}{|\mathcal{W}|} \sum_{\vec{w} \in \mathcal{W}} \min_{\vec{a} \in \mathcal{A}} \{u_{\vec{w}}(a)\} \tag{2.9}$$

where $u_{\vec{w}}(a)$ is a utility function, parameterized by weight vectors $\vec{w} \in \mathcal{W}$, that assigns a real value to each solution vector. This indicator assesses convergence, uniformity and spread of the set.

3

DIVERSITY

Diversity is a desirable characteristic in MOEAs populations due to two main reasons: (1) it helps providing different solutions to the decision maker and (2) it is necessary for a good performance of the algorithm. In Section 3.1 we present the definition of diversity used in this work, along with a deeper discussion on its importance. In Section 3.2 we enlist some of the existing techniques used to maintain diversity in MOEAs.

3.1 DIVERSITY IN MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZATION

In the previous chapter we mentioned three desirable characteristics for an approximation set $\mathcal{A}$: convergence, distribution and spread. Although there is no universal definition for diversity in the multi-objective evolutionary optimization field, most authors agree that diversity measures the latter two of these three criteria: distribution and spread [125, 1, 45, 17, 114]. Then, an approximation set with good diversity would have its solutions uniformly distributed along the entirety of the Pareto Front. In Fig 3.1 we show an example to illustrate these concepts in two dimensions. In Fig 3.1a the feasible region of the MOP is shown in gray, while the corresponding Pareto Front with its two extreme points is shown with a thicker black line. Then in Fig 3.1b we show the image of an approximation set A_1 with a poor distribution but well spread, since it contains both of the extreme points of the Pareto Front. Next, in Fig 3.1c, the image of approximation set A_2 has a much better distribution, but lacks solutions close to the extreme points, causing it to be badly spread. Finally, in Fig 3.1d, the image of approxima-

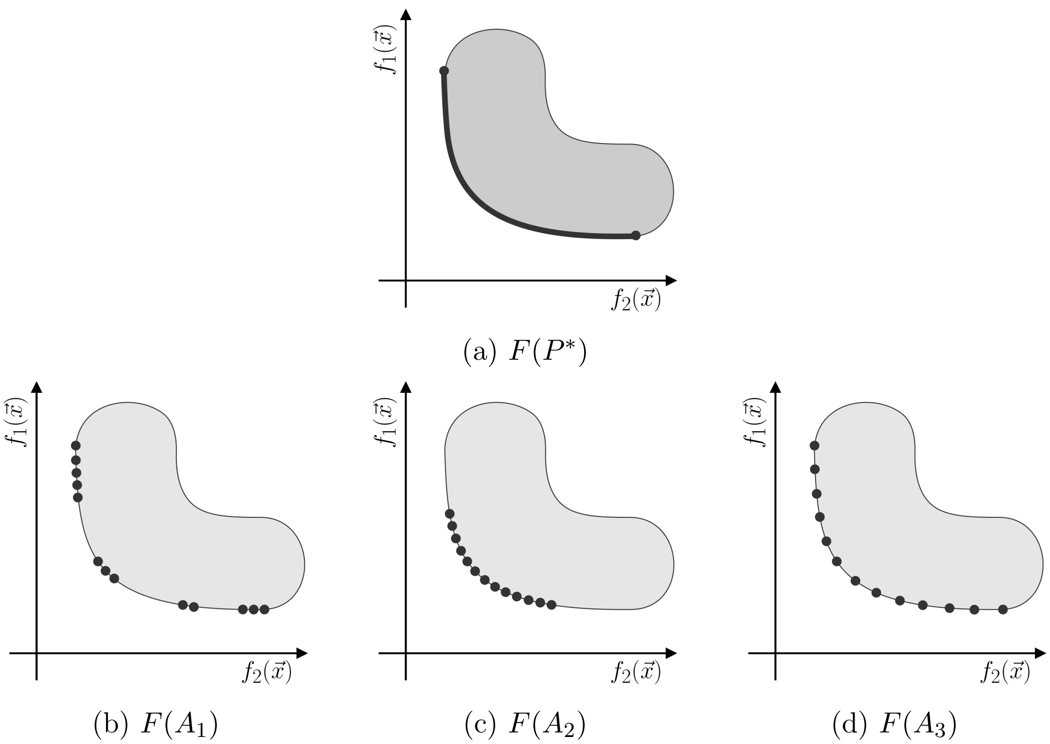


Figure 3.1: Example of a Pareto Front for a two-objective MOP (a) and three different approximation sets for the same problem. One with a poor distribution but well spread (b), another one with a good distribution but badly spread (c), and finally one with good distribution and well spread (d).

tion set A_3 exhibits a good distribution and is well spread.

Although this is the most used definition for diversity, there are some authors who consider that an approximation set with good uniformity does not necessarily mean that it also has good diversity, and vice versa. This is because they consider that a set with a uniform distribution provides redundant information for some objective values, whereas a good diversity should provide decision makers the maximum amount of (different) information [119].

3.1.1 Importance of diversity in the final approximation set

The goal of solving a MOP is to find solutions in the Pareto optimal set ($\mathcal{P}^*$). However, for many MOPs, this set has an infinite size, causing it to be impossible to enumerate every single optimal solution. In this case, the best we can do is to aim to obtain a representative subset of $\mathcal{P}^*$, which we call the approximation set $\mathcal{A}$. Ideally, this set should include solutions as diverse as possible, because the decision-maker will have to select one or some of these solutions. Hence, it is desirable to offer a wide variety of Pareto optimal solutions, allowing him to select the one that adapts the best to his specific needs [10].

3.1.2 Importance of diversity during the evolutionary process

The solution process of any MOEA involves an intrinsic trade-off between the goals of convergence (also known as proximity) and diversity of the solutions. And, even though diversity tends to be considered the secondary goal out of these two, it is also of great relevance in the process of solving most MOPs [9].

Biologically speaking, the loss of diversity also exists in nature, in a phenomenon called genetic drift, in which, due to random occurrences, gene variants can disappear entirely, thus reducing genetic variation. In the context of MOEAs, there are two well-known causes for loss of diversity:

1. Selection pressure: is the natural result of the expected value of the selection process. Lower performing solutions (in terms of fitness values) are expected to dissappear from a finite population, which will lead to preserving only high performing solutions.

2. Selection noise: in a finite population, random choices among identically fit solutions can occur, adding noise to the expected count for each individual.

These two phenomena can cause premature convergence of the algorithm to local optima, or even lead the population to a point where it mostly consists of copies or slight variants of the same high performing individual [80]. Hence, diversity maintenance techniques have become a standard component of MOEAs and are a fundamental research topic in evolutionary computation.

3.2 Diversity maintenance techniques

The first explicit attempt to maintain diversity in an evolutionary algorithm was made by Holland [57] who proposed the *crowding* operator to identify situations in which many individuals coexist in a given niche. De Jong later experimented with this operator [29] creating the first experimental study on a diversity maintenance mechanism.

Ever since, this problem has been studied thoroughly by many researchers [101], and different mechanisms have been proposed to preserve diversity which may be classified in three main categories: density estimators, mating restrictions and secondary populations. Each of them is discussed in the following subsections.

3.2.1 Density estimators

The role of diversity in MOEAs has been studied by a number of researchers (see for example [50, 126, 118, 102, 119]) over the years and the main outcome of such studies is the fact that the density estimator has become a standard mechanism in modern MOEAs.

Density estimators are responsible for blocking the selection mechanism of a MOEA with the aim of allowing it to generate different solutions in a single run. Due to this very broad definition, there are many different density estimators, which can be grouped in the following subcategories:

3.2.1.1 Niches and Fitness sharing

Goldberg proposed the use of fitness sharing to favor unexplored regions of the search space by penalizing the individuals in crowded regions (or niches). Hence, individuals that are isolated in their own niches have a better chance of being selected, since their fitness is not penalized [47]. The shared fitness of an individual $\vec{x}_i \in \mathbb{R}^m$ is given by the following expression:

$$F_{sh}(\vec{x}_i) = \frac{F_0(\vec{x}_i)}{\sum_{j \in \mathcal{P}} s(d(i,j))} \tag{3.1}$$

where $F_0(\vec{x}_i)$ is the original fitness of the individual, $\vec{x}_i$, $\mathcal{P}$ is the population of individuals, and $s(d(\vec{x}_i, \vec{x}_j))$ is zero if the Euclidean distance of individual $\vec{x}_i$ to individual $\vec{x}_j$ is bigger than the niche radius σ_{share}, and a value in range $[0, 1]$ otherwise. The main drawback of this method is the setting of the parameter σ_{share}, which defines the size of the niche considered, and it is problem dependent [50]. In this regard, Fonseca and Fleming proposed a method to automatically set this parameter based on the distribution of the population at each generation [44].

In the context of many-objective optimization, one related technique that uses distances among individuals is the Pure Diversity (PD) metric [119], which measures the dissimilarity of solutions to the rest of the population using a greedy ordering and L_p-norm-based distances. One of its main advantages is that it requires no parameters. Additionally, a diversity maintenance derived from PD has shown a good performance in MOPs with up to ten objectives.

3.2.1.2 Crowding

This technique was proposed by Deb et al. as a diversity maintenance mechanism for the Non-dominated Sorting Genetic Algorithm (NSGA-II) [34]. Essentially, it estimates the density of each individual by cumulatively summing up the absolute distances from the individual to its two closest neighbors across all objectives.

This process is repeated for all individuals, being the only exceptions the extreme solutions since they are automatically preserved, and the individuals with the lower crowding value are the ones that are deleted at each generation.

However, there is a pathological problem with this technique, where it fails to detect a crowded region. It occurs when individuals are really close from one of its neighbors, but considerably far away from the second closest neighbor. In this case, the distance from the second neighbor might help preserving such individual, even though there are two individuals in an already crowded region (the individual in question and its closest neighbor).

Adra and Fleming have proposed an improvement to traditional crowding by using two novel density estimators DM1 and DM2 in addition to crowding to solve many-objective optimization problems [1]. DM1 is an adaptive strategy to regulate population diversity according to the solutions dispersion (measured by the maximum spread metric) and DM2 is an adaptive mutation operator that defines the interval in which decision variables can be mutated and is governed by both the spread metric and Deb's crowding measure. This proposal was evaluated using MOPs with up to 20 objectives, however, only DM1 presents favorable evidence of improving results.

3.2.1.3 Adaptive mesh

Knowles and Corne proposed the use of a position-based external archive with the intention of obtaining a uniform distribution [71, 70]. This approach divides the objective space into a grid and defines diversity in terms of how many solutions are located in each sub-region.

Solutions are stored in an archive using this space division, and once the archive has reached its maximum size, it will eliminate solutions selected at random from the most crowded sub-region in order to store the new solution. This is a useful approach with the advantage of having a simple measurement of crowding [78, 25].

However, it has some problems related to the amount of divisions used in the grid, since different values can generate different sub-regions to be the most crowded. Additionally, when individuals are located in the corners of adjacent sub-regions, it can also lead to a wrong detection of the most crowded sub-regions [50].

3.2.1.4 Clustering

Zitzler et al. proposed a clustering method to maintain the size of external archives [144, 142] where a desirable number of "clusters" is defined and a clustering algorithm is used to obtain them.

In this proposal each non-dominated individual initially represents a distinct cluster. Then, if the number of clusters/individuals is greater than the maximum archive size, the two closest individuals, considering Euclidean distances, are merged into a single cluster. This process is repeated until the number of clusters is smaller than or equal to the maximum size of the archive. Then, the archive is trimmed by selecting only one "representative" individual per cluster. To this end, the individual with minimal average distance to all other points in the cluster is selected. One of the advantages of this approach is that using a high number of clusters induces uniformity in the population [115].

3.2.1.5 Use of entropy

Farhang-Mehr and Azarm [39] proposed the use of Shannon's entropy [111], which measures how evenly a set of numbers is spread, to assess the diversity of solutions. Shannon's entropy is large given a set of numbers which are approximately the same, and low when the numbers are very different. Consequently, the higher the entropy of a solution set, the more evenly spread throughout the feasible region is and the better coverage of the space it provides.

This concept has been used by dividing the feasible domain into a grid and applying an entropy-based density function to each sub-region [69, 27, 39].

More recently, in the context of many-objective optimization, Zhou et al. [136] proposed an entropy based evolutionary algorithm with adaptive reference points (EARPEA) which exhibits a good performance in problems with up to 10 objectives. However, its main drawback is that it is parameter dependent.

3.2.2 Mating restrictions

Mating restrictions are discussed in Goldberg's seminal book on genetic algorithms [46] as a mechanism to prevent or minimize the propagation of the so-called "lethals" (offspring with low fitness values). In other words, mating restrictions were originally proposed as a mechanism to bias the way in which individuals mate during recombination aiming to increase the effectiveness and efficiency of a genetic algorithm. Goldberg provided a simple example of mating restrictions based on genotypic similarities and pointed out that, biologically, mating restrictions are equivalent to geographical isolation or to establishing a barrier that constrains the flow of genes. Thus, mating restrictions are closely related to biological speciation, which gives rise to new species.

Deb and Goldberg [32] proposed mating restrictions in single-objective genetic algorithms based on the phenotypic distance between the individuals. In their proposal, the mating companion of an individual was selected from individuals lying within a user-defined distance (defined with a parameter called σ_{mate}). By pairing relatively similar parents in objective space, their goal was to prevent, or decrease, the generation of *lethals*, hence improving the performance of the genetic algorithm.

Ever since these two proposals were introduced, different mating restriction schemes have been proposed, exploring the effect of measuring the distance between individuals in both objective and decision space, as well as pairing similar or dissimilar individuals according to a user-defined metric [43, 44, 103, 59].

Since mating restrictions determine which individuals are to be paired in the recombination step of an evolutionary algorithm, they have a direct effect in both the exploration and the exploitation capacity of the algorithm. Hence, if the mating restriction biases the population towards the generation of new different individuals, it can be considered as a diversity maintenance technique.

In the context of MOEAs, there are some studies focused on the role of mating restrictions. For instance, Zitzler and Thiele [143] as well as Van Veldhuizen and Lamont [117] found that there was not enough empirical evidence to argue that the use of mating restrictions would either improve or worsen the performance of a MOEA. On the other hand, Ishibuchi studied the use of mating restrictions using either Euclidean or Hamming distances (as well as mating of either similar or dissimilar individuals) in MOPs with two and three objectives, finding that mating restrictions can indeed improve the performance of MOEAs but they are problem-dependent as well as algorithm-dependent [65, 64].

A variety of mating restriction mechanisms have been proposed in the literature with the aim of improving the overall performance of MOEAs, either by improving the population's diversity or the convergence speed. However, few of such proposals have been tested with many-objective optimization problems (MOPs with 4 or more objective functions.

Multi-objective evolutionary algorithms enhancement has been performed in [128], where two decomposition-based MOEAs, namely MOEA/D [135] and EFR [127], were improved using a mating restriction. The resulting algorithms (MOEA/D-DU and EFR-RR) are able to outperform their original versions in most tested problems with up to 13 objective functions. The mechanism used consists in determining neighborhoods for each individual based on their perpendicular distance to weight vectors in objective space. Using this information, the mating restriction allows to balance diversity and convergence by mating individuals within the same neighborhood in MOEA/D or by selecting individuals from the same neighborhood in the ranking performed in EFR.

Another modification to MOEA/D in order to improve the results obtained in many-objective optimization problems is MOEA/D-LWS [120] which implements a localized weighted sum method. This algorithm implements a weighted sum scalarizing function paired with a mating restriction scheme in order to use the function locally (within a hypercone around each weight vector). This modification allows to obtain solutions in non-convex portions of the Pareto fronts, which is the most well-known downside of using a weighted sum. Experimental results comparing against two other MOEA/D variants as well as three other MOEAs show really good results in problems with up to seven objectives.

The Enhanced-Mating-Selection-Many-Objective-NSGA-II (EMS-MO -NSGA-II) [19] enhances MO-NSGA-II's mating selection mechanism by utilizing two mating mechanisms. The first one is a reference-point based selection procedure, while the second one is a neighborhood-based selection scheme. These two strategies were experimentally evaluated, both individually and combined, by solving the DTLZ 1-4 test problems [35] with up to 10 objectives. The results obtained showed a significant improvement when using both strategies at the same time.

The many-objective evolutionary algorithm based on directional diversity and favorable convergence (MaOEA-DDFC) [20] uses a scalarizing function to obtain convergence degrees of individuals in the population. Then, using a binary tournament selection scheme, individuals are selected and compared using both Pareto dominance and convergence degrees to create a mating pool in which only the best individuals are chosen. This algorithm was compared with respect to seven other MOEAs, obtaining good results and improving their performance in the majority of the test problems used, which comprise problems with up to ten objectives from the DTLZ and WFG test suites.

The constrained MOEA/D with Directed Mating and Archives of infeasible solutions (CMOEA/D-DMA) [85] relies on useful infeasible solutions which are generated during the search process. Up to eight infeasible solutions per weight vector are stored in an archive, and they are randomly selected to be mated with feasible solutions. This mechanism was coupled to cMOEA/D and used to solve the mCDTLZ problems [87] as well as m objectives k knapsack problems [68] with up to eight objective functions. The results obtained indicate that this algorithm outperforms the original cMOEA/D as well as NSGA-III and TNSDM [86] (Two-stage Nondominated Sorting and Directed Mating) in most of the test instances adopted.

The spectral clustering based multi-source mating selection strategy (SMMS) is designed to detect regularity properties and to balance population diversity and convergence. It was coupled to SMS-EMOA [7] giving rise to the so-called SMMEA [121]. Given an individual $\vec{x}$, this algorithm adopts three different sources for selecting a mate: (1) a sub-population from the same cluster of $\vec{x}$, (2) a sub-population from a cluster adjacent to the cluster containing $\vec{x}$, or (3) the whole population. The selection of one mating source is performed using adaptive probabilities for the first two sources, obtained from each source efficiency. This proposal was compared with respect to six MOEAs in the solution of the MOPF [75], UF [131] and GLT [49] test problems, with two and three objective functions. The results showed that SMMEA had a significantly better performance than the other MOEAs adopted.

3.2.3 SECONDARY POPULATIONS

In the context of evolutionary algorithms, an archive stores a population of individuals, and the process of updating this archive consists in comparing new solutions with those already stored and deciding which ones are kept and which ones are discarded.

In theory, the population of any traditional MOEA can be seen as an archive that is updated at each generation, since new offspring individuals are compared with the parent individuals already in the population. However, in practice, archives commonly refer to the storage of individuals in addition to the main population of the MOEA.

According to Horn [58], every MOEA should use a secondary population since the goal is to obtain a discrete image of a possibly continuous Pareto front. Therefore, it is desirable to store as many solutions as possible in order to obtain a Pareto front approximation with a good distribution. However, even though researchers agree that secondary populations are useful, the computational cost of maintaining an unbounded archive makes this infeasible. Hence, there are some important questions that remain to be answered such as the type of interaction between the main population and the secondary one, as well as the filter used to keep the external archive at a reasonable size.

Although the first MOEAs which implemented secondary populations used arrays to store the solutions, several alternative data structures have been used to this end. For instance, Mostaghim et al. [88] proposed the use of quadtrees in the strength Pareto Evolutionary Algorithm (SPEA) [144]. Bringmann et al. [15] proposed the *Approximation Guided Evolutionary algorithm* (AGE), which employs non-bounded files. Fieldsend et al. [41] provide some remarks on the negative consequences of bounding the secondary population in the produced Pareto fronts. They also propose the use of two new data structures (non-dominated trees and PQRS trees) to store and efficiently retrieve solutions in a non-bounded file.

Laumanns et al. [74] proposed a relaxed form of Pareto dominance called ε-dominance, which is used to filter solutions in an external file. The idea is to define a series of ε sized boxes and allow only one non-dominated solution in each of them. However, in a later study, it has been shown that ε-dominance can be detrimental for problems where the number of feasible objective vectors is relatively small, since the optimization process can be slow down drastically [60]. Additionally, other archiving strategies have been proposed using ε-dominance and Hausdorff metrics [106, 105].

Zapotecas and Coello [81] proposed the use of *Convex Hull of Individual Minima (CHIM)* to maintain well distributed solutions in the secondary population of a MOEA.

Another proposal is to locally maximize the hypervolume dominated by the archive [72]. This has led to more recent studies on the convergence of archiving algorithms with respect to the hypervolume [14, 13].

3.3 Summary

Diversity is a desirable property in approximation sets provided by MOEAs to the decision maker and an elemental component that aids the algorithm to avoid premature convergence. In this chapter we have described some of the diversity maintenance techniques used in traditional MOEAs as well as some of the proposals made specifically for many-objective optimization problems. However, there is still room for improvement in several of these diversity maintenance techniques. Particularly, the use of different of these techniques remains to be explored with more detail in many-objective optimization problems. In the following two chapters we explore the use of mating restrictions based on a performance indicator used to assess diversity (s-energy) and their effect in many-objective optimization problems.

4

S-energy based mating restrictions

In MOEAs, mating restrictions determine the criterion used to pair the individuals from the population in the recombination step of the algorithm. In doing so, they can significantly impact the exploration and exploitation capabilities of the search process. In terms of diversity maintenance, mating restrictions are useful because they can bias the population towards the generation of offspring individuals which are relatively different from the parents, increasing the diversity in the population. This can be achieved by combining parents with different characteristics. In order to determine how different two individuals are there are several metrics which can be used. In this chapter, we explore the use of the s-energy performance indicator to measure the contribution of each individual to the whole population's diversity, and we present some new mating restriction schemes based on these contributions along with their experimental validation.

4.1 State of the art mating restrictions

Initially, mating restrictions were based on distance between individuals in objective space (although distance in decision space has also been used). However, there have been different proposals using clustering in addition to these distances, as well as other restrictions based on additional measures.

One of these proposals using clustering is the mating restriction strategy based on survival length [76] (MRSL), which is a self-adaptive mechanism that employs clustering to obtain the structure of the population and then assigns different probabilities to individuals in each cluster based on their corresponding survival length.

The underlying idea of this proposal is that individuals with a high survival length are high-quality individuals and the surrounding area should be exploited, while individuals with a low survival length are newly generated, and therefore, exploration is needed to assess their quality. Experimental results implementing MRSL in five MOEAs of the state-of-the-art show that its use improves results when solving MOPs having two and three objectives.

Another similar proposal is the decomposition based multiobjective evolutionary algorithm with self-adaptive mating restriction strategy [77] (MOEA/D-MRS). This algorithm implements another mating restriction based on survival length. However, this approach is specifically designed for a decomposition-based MOEA, which is MOEA/D. It was compared to other MOEA/D variants, obtaining better results in most of the test problems adopted.

The fuzzy c-means clustering-based mating restriction [132] (FMR) is another restriction in which clustering is used to discover the structure of the population. However, solutions have different degrees of membership to each cluster, resulting in the fact that one solution can belong to more than one cluster. This is used to generate a *mating pool* for each individual, which contains the individuals with which they can mate. FCMMO is a MOEA designed around FMR and it utilizes differential evolution to recombine individuals from a given *mating pool* and the hypervolume-based environmental selection mechanism of the SMS-EMOA [7]. FCMMO obtained good experimental results with respect to five other MOEAs in MOPs with two and three objectives.

Finally, the Manifold Learning-Based Mating Restriction Strategy (MRML) is another mechanism which aims to improve the performance of a MOEA by calculating the *manifold distances* between individuals, which considers both objective and decision space distances. MRML employs a niche radius R to obtain the neighborhood of a solution based on the previously calculated *manifold distances*. Once the neighborhood of a solution has been obtained, it is paired to the closest solution as its mating companion, and the remaining solutions in the neighborhood are discarded. This causes that some solutions cannot be paired due to missing individuals. However, this is solved by recombining such individuals with mutated versions of themselves. MRML was coupled to three MOEAs and was used to solve MOPs with complicated Pareto sets, having two and three objectives, obtaining good results [91].

All of the proposals mentioned above have been validated adopting test problems with complex Pareto sets or considerably difficult features from the GLT [49], UF [131] and WFG [63] test suites. Concerning the MOEAs adopted to compare results, the most commonly used are well-known MOEAs such as NSGA-II, SMS-EMOA, SPEA/R, SPEA2 and MOEA/D-DE. However, in all cases, three is a maximum number of objectives considered, which may be due to the considerable computational cost involved in the use of clustering techniques or to the additional cost of using hypervolume-based selection in approaches such as FCMMO.

4.2 S-ENERGY BASED MATING RESTRICTIONS

In this section, we propose the use of the s-energy indicator to design 4 different mating restrictions. We also provide experimental validation of these restrictions to assess their performance.

4.2.1 S-ENERGY

Riesz s-energy (E_s) was proposed by Hardin and Saff [51], and has been used as a performance indicator to measure the uniformity of the distribution of a set of points [37]. Given a set $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$ of m-dimensional points, its s-energy is defined as follows:

$$E_s(X) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \frac{1}{\mid \vec{x}_i - \vec{x}_j \mid^s} \tag{4.1}$$

where $| \cdot |$ represents the Euclidean distance and $s > 0$ is a fixed parameter. In this work, we use $s = m - 1$ in all cases. This indicator should be minimized in order to obtain a population with a good diversity. Moreover, the individual s-energy contribution (C_{si}) of a given point $\vec{x}_i$ may be computed as:

$$C_{si} = E_s(X) - E_s(X \setminus \{\vec{x}_i\}) \tag{4.2}$$

where $\vec{x}_i \in X$. Since $E_s(X)$ is to be minimized, a high value of C_{si} means that the individual $\vec{x}_i$ is in a "crowded" region, since at least one other individual in the population is relatively close to it. On the other hand, a low C_{si} value means that the individual $\vec{x}_i$ has a better contribution to the global distribution, since it is in a "non-crowded" region. Using this information, we can rank the population based on their individual contribution measured in objective space, considering the individual with the highest contribution as the worst individual, and the individual with the lowest contribution as the best one. Given such ranking, we can establish different mating restriction schemes which may favor the reproduction of the individuals with the best contributions. This is the core underlying idea of our proposed s-energy based mating restriction (SMR) schemes, and our goal is to use population diversity information to improve the performance of the evolutionary search. In the following two subsections we describe the different restrictions proposed.

4.2.2 SMR1: Similar vs. Dissimilar

The first two schemes we propose are the most intuitive in terms of considering s-energy contributions: (1) pairing individuals with similar contributions and (2) pairing individuals with dissimilar contributions.

Both restrictions require the computation of the contribution C_{si} for each individual in the population, at every generation. Once these contributions have been obtained, we rank the population according to these values and we proceed to pair individuals as follows.

In the first strategy, called SMR1_SIM, we choose the best individual from the population and we pair it with the second best individual. Then, we repeat this process with the next two best individuals from the population until all individuals are paired. This is illustrated in Figure 4.1.

The second strategy SMR1_DIS pairs the best individual from the population with the worst individual. Once again, this process is repeated with the second best and the second worst individuals, and so on, until all pairs of parents are obtained, as depicted in Figure 4.2.

These two schemes are evidently the simplest possible mating restrictions using $s-$energy individual contributions. Intuitively, we expect SMR1_SIM to pair the best individuals (meaning the ones with low s-energy individual contributions) with each other, which would combine individuals in non-clustered regions, and would potentially generate offspring in the regions between them. However, it will also pair the worst individuals (meaning the ones with high s-energy invidivual contributions) with each other, which would pair individuals in clustered regions with other individuals that are also in clustered regions, potentially creating offspring in already crowded regions. On the other hand, we expect SMR1_DIS to pair the best individuals (located in non-clustered regions) with the worst ones (located in clustered regions). This seems to be more useful than the previous restriction, as it could potentially prevent the generation of solutions in

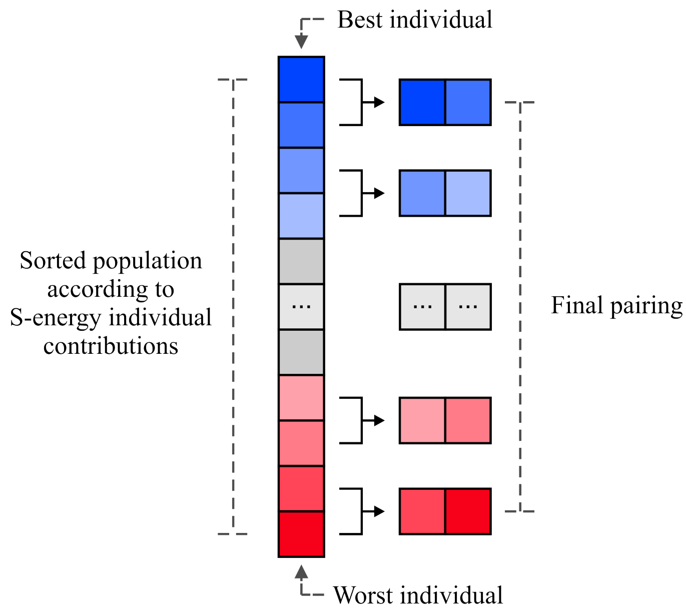


Figure 4.1: Example of our proposed SMR1_SIM mechanism.

already crowded regions.

4.2.3 SMR2 and SMR3: Mating Pool and Replacement

In addition to the previous restrictions, we proposed another variants of SMR1_DIS, in which instead of directly pairing the best individual and the worst individual, a mating pool containing a fixed number of worst solutions is created. The size of the mating pool is set by $\sigma_{pool} > 0$, which is a user-defined parameter. Once the best individual is paired with an individual from the mating pool, they are both no longer considered in the current iteration, and the worst individual gets replaced by the next worst individual available. Next, the second best individual gets paired with an individual from the updated mating pool, and so on until finding all pairs. This mechanism is illustrated in Figure 4.3.

We adopted two criteria to select individuals from the mating pool: (1) select the solution in the mating pool with the largest Euclidean distance to the individual considered as the best one, and (2) select the solution in the mating pool with the smallest Euclidean distance to the best individual. This yields two different mating restrictions, called SMR2 and SMR3, respectively.

In both cases the Euclidean distance is measured in objective space, and

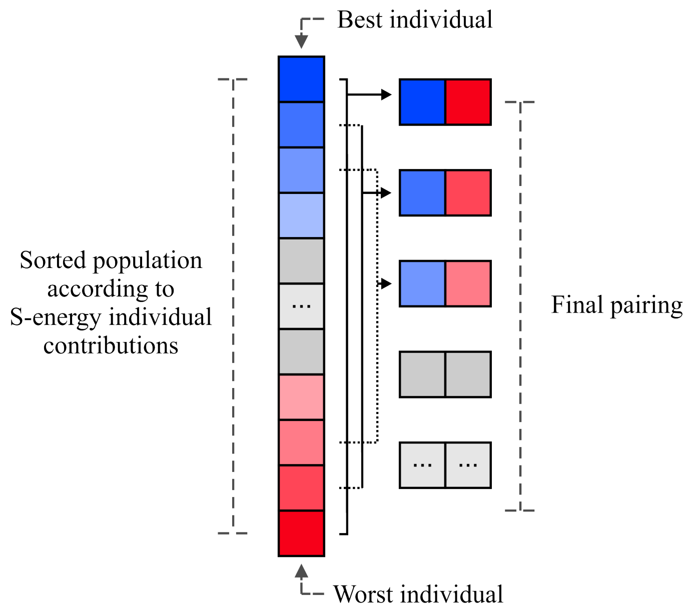


Figure 4.2: Example of our proposed SMR1_DIS mechanism.

these distances do not need to be additionally computed, since they were already obtained when computing s-energy individual contributions.

Since both SMR2 and SMR3 are variants of SMR1_DIS, we still expect them to improve population diversity by pairing an individual from a clustered region with an individual from a non-clustered region. However, we add the mating pool to either increase, or decrease the Euclidean distance between the paired individuals. This is done to emphasize either behavior (individuals that may be close or far from each other), since the comparison of s-energy individual contributions only provides information of how good or bad individuals are in terms of how many other individuals are close to them, but it doesn't provide information about the distance between the individuals compared.

Moreover, it is important to notice that in both SMR2 and SMR3 it will occur that individuals with similar contributions will be paired at some point of the mating restriction strategy, which could once again, potentially generate offspring in already clustered regions. Hence, we propose another feature to avoid this, which we call replacement.

Since the pairing of individuals with similar contributions will always occur in the last pairs formed, we propose to replace a percentage of the last pairs of parents with some other, hopefully more useful, pairs. This mechanism is shown in Figure 4.4. In order to create these new pairs, we

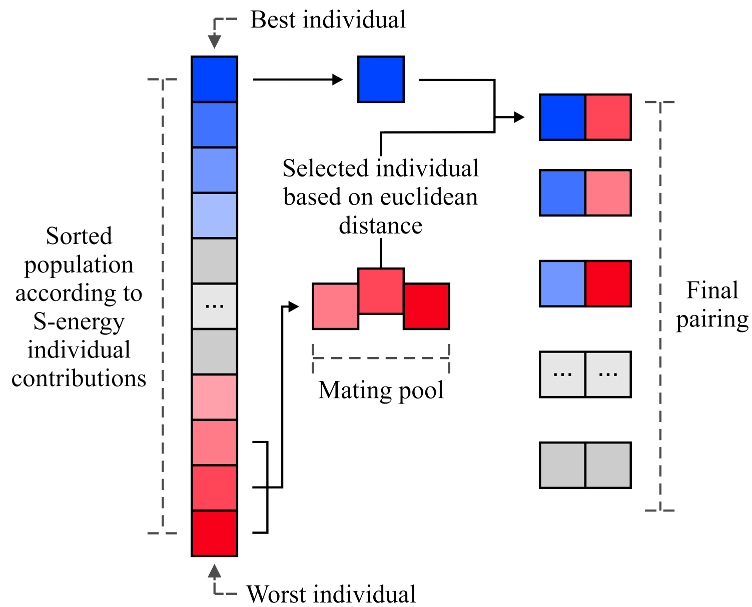


Figure 4.3: Example of replacement technique after applying the mating restriction scheme.

select the best individual from the population and we pair it with a mutated version of this same individual. Then, we select the second best individual and repeat the process with a mutated version of it, and so on.

Using this replacement feature, we aim to improve the exploitation capability of the algorithm in the vicinity of the individuals with the best contributions. The implementation of this feature yields mating restrictions SMR2_R and SMR3_R. In this work, we considered $\frac{1}{6}$ of the population as the percentage to be replaced.

4.3 Experimental validation

In order to evaluate the impact of our proposed mating restrictions we implemented them in NSGA-III [33] and compared the results obtained with and without our s-energy based mating restrictions. We adopted the Deb-Thiele-Laumanns-Zitzler (DTLZ) [35] and Walking-Fish-Group (WFG) [63] test suites, since they contain scalable MOPs with solutions that include Pareto sets with different geometrical features.

In order to assess the quality of the approximation sets obtained, we adopted the hypervolume (HV) performance indicator [138] as well as the inverted generational distance (IGD) [23]. Additionally, we also used s-

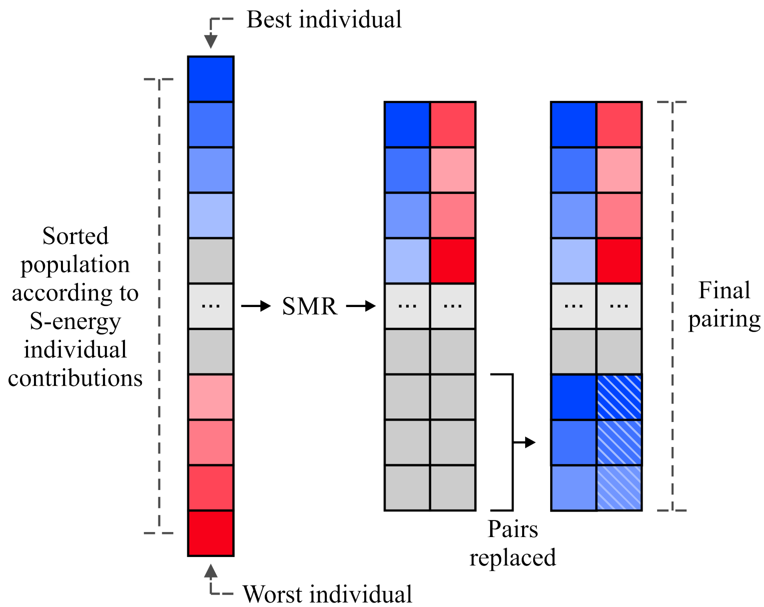


Figure 4.4: Example of the SMR2/SMR3 mechanism with replacement feature.

energy [51] in order to directly measure the distribution of approximation sets, since both the hypervolume and IGD measure it indirectly.

Even though s-energy is the core of all of our mating restrictions, we only employed it as a quality criterion to rank individuals, and not as a performance indicator to be specifically improved. Hence, s-energy values are only used for comparison purposes, since its direct minimization is not part of our proposal.

We performed 30 independent executions using each algorithm, and we averaged the values of each performance indicator. We present these results in the following tables, where the best value for each indicator is written in **boldface**, and the cases where the use of a mating restriction improved the results of the original NSGA-III are shown in gray.

Using this experimental setup we first compared SMR1_SIM and SMR1_DIS against NSGA-III, and we obtained the results shown in Table 4.1. SMR1_SIM outperforms the original NSGA-III in 27 out of 48 test problems when comparing hypervolume values, in 26 problems when using IGD and in 18 test problems when using s-energy. It can be seen that SMR1_SIM shows a clear advantage in the WFG test problems with 2 and 5 objectives. On the other hand, SMR1_DIS only outperforms 19 of the HV values, 22 of the IGD values, and 26 of the s-energy values, being less consistent than SMR_SIM.

In Table 4.2 we present the results of comparing SMR2 with and without

Table 4.1: Comparison of the average HV, IGD and *S*-energy values obtained when using SMR1_SIM and SMR1_DIS.

Problem	Objectives	Hypervolume			IGD			S-energy		
		NSGA-III	SMR1_SIM	SMR1_DIS	NSGA-III	SMR1_SIM	SMR1_DIS	NSGA-III	SMR1_SIM	SMR1_DIS
DTLZ1	2	**2.1237E+00**	2.1234E+00	2.1237E+00	**1.7840E-03**	1.8225E-03	1.7841E-03	1.1725E+05	1.1797E+05	**1.1725E+05**
DTLZ2		3.2101E+00	**3.2101E+00**	3.2101E+00	3.9625E-03	**3.9625E-03**	3.9626E-03	5.3391E+04	**5.3391E+04**	5.3392E+04
DTLZ3		3.2098E+00	3.2096E+00	**3.2098E+00**	**3.9867E-03**	4.0062E-03	3.9906E-03	**5.3386E+04**	5.3404E+04	5.3425E+04
DTLZ4		**2.8874E+00**	2.8068E+00	2.8874E+00	**2.0079E-01**	2.4999E-01	2.0079E-01	3.9153E+04	**3.5594E+04**	3.9153E+04
DTLZ5		3.2101E+00	**3.2101E+00**	3.2101E+00	3.9625E-03	**3.9625E-03**	3.9626E-03	5.3391E+04	**5.3391E+04**	5.3392E+04
DTLZ6		3.0542E+00	**3.0700E+00**	3.0676E+00	9.2585E-02	**8.4144E-02**	8.5291E-02	5.4260E+04	**4.9077E+04**	4.9814E+04
DTLZ7		**4.4173E+00**	4.4172E+00	4.4173E+00	**5.1851E-03**	5.2685E-03	5.1885E-03	**7.3309E+04**	3.1605E+05	7.2876E+05
WFG1		3.9940E+00	**4.2139E+00**	4.0734E+00	1.4331E+00	**1.3816E+00**	1.4195E+00	**1.2784E+05**	1.5188E+05	1.6825E+05
WFG2		9.2570E+00	**9.3155E+00**	9.2557E+00	6.5750E-01	**6.4416E-01**	6.5751E-01	5.9184E+04	2.1309E+05	**5.8157E+04**
WFG3		1.0437E+01	1.0442E+01	**1.0463E+01**	4.3466E-02	3.9505E-02	**3.9135E-02**	2.9325E+04	**2.6734E+04**	2.6829E+04
WFG4		8.2336E+00	**8.4349E+00**	8.1320E+00	2.8252E-02	**1.7609E-02**	3.5015E-02	**2.2522E+04**	2.3392E+04	2.2684E+04
WFG5		7.9344E+00	**7.9741E+00**	7.8600E+00	7.7568E-02	**7.6630E-02**	8.2881E-02	2.2769E+04	2.4107E+04	**2.2023E+04**
WFG6		7.9733E+00	**8.1083E+00**	7.9903E+00	6.8255E-02	**5.9293E-02**	6.6291E-02	2.4668E+04	2.4496E+04	**2.1931E+04**
WFG7		6.6706E+00	**7.1797E+00**	6.6927E+00	2.9508E-01	**1.7524E-01**	2.8986E-01	4.2732E+04	3.9234E+04	**3.6990E+04**
WFG8		7.3725E+00	**7.7891E+00**	7.2931E+00	1.4150E-01	**1.0197E-01**	1.5346E-01	**4.2315E+04**	1.0935E+06	4.6764E+04
WFG9		**8.1696E+00**	8.1685E+00	8.1564E+00	**3.5258E-02**	3.5311E-02	3.6042E-02	2.2445E+04	**2.1248E+04**	2.1764E+04
DTLZ1	3	3.3477E+00	**3.3489E+00**	3.3481E+00	2.2906E-02	**1.9120E-02**	2.2479E-02	1.0500E+10	**2.9185E+06**	2.1935E+09
DTLZ2		**7.4184E+00**	7.4184E+00	7.4184E+00	**4.9314E-02**	4.9330E-02	4.9325E-02	**9.1884E+04**	8.3415E+05	2.8241E+05
DTLZ3		7.4116E+00	7.4167E+00	**7.4177E+00**	5.6253E-02	5.0024E-02	**4.9583E-02**	1.6166E+09	**2.8146E+05**	1.2600E+06
DTLZ4		**7.0466E+00**	6.7855E+00	7.0335E+00	2.2902E-01	3.0514E-01	**2.0993E-01**	3.0716E+11	2.7960E+11	**9.9681E+10**
DTLZ5		**4.0043E+00**	3.8615E+00	3.9751E+00	6.5059E-02	9.3704E-02	**5.4936E-02**	8.1297E+11	1.8147E+12	**2.4368E+11**
DTLZ6		3.9910E+00	4.0051E+00	**4.0196E+00**	1.0158E-01	9.2757E-02	**8.8402E-02**	**1.1581E+11**	8.2169E+11	1.5469E+11
DTLZ7		1.3202E+01	1.3291E+01	**1.3304E+01**	7.8863E-02	7.0082E-02	**6.8713E-02**	8.0305E+07	2.8752E+09	**3.8662E+06**
WFG1		5.4149E+01	**5.5575E+01**	5.5332E+01	1.1700E+00	**1.1456E+00**	1.1514E+00	4.1152E+08	1.4598E+09	**8.0898E+06**
WFG2		**9.6392E+01**	9.5886E+01	9.5399E+01	**2.4560E-01**	2.6005E-01	2.6954E-01	**1.0067E+05**	1.9316E+06	2.3302E+05
WFG3		**2.4037E+01**	2.3975E+01	2.4018E+01	**9.3331E-02**	9.8401E-02	9.3440E-02	**7.1353E+08**	7.4582E+10	9.6926E+09
WFG4		7.6820E+01	**7.6857E+01**	7.6822E+01	2.0001E-01	1.9994E-01	**1.9993E-01**	7.2705E+03	5.8622E+03	**5.1335E+03**
WFG5		7.3458E+01	7.3220E+01	**7.3486E+01**	2.1384E-01	2.1422E-01	**2.1292E-01**	5.6060E+03	4.8130E+05	**5.2243E+03**
WFG6		**7.4081E+01**	7.3964E+01	7.4000E+01	**2.0938E-01**	2.1008E-01	2.0995E-01	6.9892E+04	**5.3703E+03**	3.8665E+04
WFG7		**7.7045E+01**	7.7027E+01	7.7035E+01	**2.0004E-01**	2.0012E-01	2.0004E-01	2.8209E+04	**1.0277E+04**	1.0823E+05
WFG8		7.0872E+01	**7.1071E+01**	7.0825E+01	2.5035E-01	**2.4753E-01**	2.5020E-01	**1.2580E+04**	3.1617E+05	5.2681E+04
WFG9		7.0274E+01	6.9398E+01	**7.0321E+01**	**2.2365E-01**	2.3352E-01	2.2455E-01	1.9415E+08	**4.4917E+06**	2.3654E+07
DTLZ1	5	**7.5927E+00**	7.5927E+00	7.5927E+00	**5.0423E-02**	5.2351E-02	5.0776E-02	**2.2272E+10**	2.5017E+11	1.1585E+11
DTLZ2		3.1698E+01	3.1698E+01	**3.1698E+01**	1.5539E-01	1.5540E-01	**1.5537E-01**	**7.7099E+09**	1.7439E+10	8.7488E+09
DTLZ3		3.1694E+01	3.1688E+01	**3.1694E+01**	1.5825E-01	1.6315E-01	**1.5722E-01**	1.1507E+11	**4.4104E+10**	9.6380E+10
DTLZ4		3.1679E+01	3.1659E+01	**3.1698E+01**	1.6342E-01	1.7193E-01	**1.5534E-01**	1.4004E+10	2.6691E+11	**1.3712E+10**
DTLZ5		**5.9019E+00**	5.8276E+00	5.8168E+00	2.3194E-01	2.3313E-01	**2.1629E-01**	**2.3861E+11**	2.5894E+12	4.2134E+11
DTLZ6		1.8920E+00	**2.6810E+00**	6.0006E-01	9.3461E-01	**7.7449E-01**	1.2100E+00	4.5468E+11	7.9833E+11	**2.8968E+11**
DTLZ7		7.7036E+01	7.6447E+01	**7.7458E+01**	**2.7009E-01**	2.8704E-01	2.7255E-01	5.3521E+10	1.8419E+11	**1.4913E+10**
WFG1		4.0787E+03	**4.2274E+03**	4.0783E+03	1.8441E+00	**1.7953E+00**	1.8308E+00	**7.1824E+10**	9.4357E+10	1.6422E+11
WFG2		9.8846E+03	**1.0155E+04**	1.0129E+04	5.6849E-01	**4.5785E-01**	4.6164E-01	**6.8679E+08**	1.9635E+10	8.0177E+09
WFG3		8.0982E+01	**8.5504E+01**	8.0353E+01	5.2549E-01	**4.7552E-01**	5.2642E-01	**6.9978E+10**	6.5755E+11	7.7536E+10
WFG4		8.8925E+03	**8.9640E+03**	8.8805E+03	9.1408E-01	**9.1257E-01**	9.1469E-01	**2.7649E+03**	1.0109E+04	7.4131E+06
WFG5		8.6746E+03	**8.6909E+03**	8.6628E+03	9.2626E-01	**9.2389E-01**	9.2761E-01	3.0457E+08	1.1009E+06	**4.7232E+05**
WFG6		8.8660E+03	**8.8872E+03**	8.8594E+03	**9.1416E-01**	9.1438E-01	9.1481E-01	1.4504E+07	**1.0850E+05**	1.2308E+07
WFG7		9.1345E+03	**9.1359E+03**	9.1303E+03	9.0612E-01	**9.0594E-01**	9.0643E-01	**1.1485E+07**	8.1259E+07	4.6796E+07
WFG8		**8.4412E+03**	8.3933E+03	8.4393E+03	9.5154E-01	9.5415E-01	**9.5117E-01**	1.2729E+07	3.2473E+07	**1.1793E+07**
WFG9		**8.0920E+03**	8.0616E+03	8.0749E+03	9.8101E-01	9.8128E-01	**9.8011E-01**	1.5351E+03	1.7351E+05	**1.4390E+03**

the replacement feature against NSGA-III. SMR2 without replacement produces improvements in 16 ouf of 48 problems regarding the hypervolume, and it produces improvements in 20 problems regarding both the IGD and s-energy indicators. SMR2 with replacement improves the results obtained in 21 out of the 48 problems using hypervolume and IGD, while it improves 27 of the results obtained with s-energy. The latter mating restriction obtains particularly good results in the DTLZ problems with 3 objectives as well as in the WFG test problems with 2 objectives. However, both of these fail to obtain good results in problems with 5 objectives, which could be an indicator of a bad scaling capability.

The results obtained when comparing SMR3 with and without replacement against NSGA-III are shown in Table 4.3. In this case, SMR3 without replacement produces improvements in 26 out of 48 test instances with respect to the hypervolume, and it produces improvements in 22 problems with respect to both the IGD and s-energy indicators. This approach was able to produce good results in the WFG test problems with 2 and 5 objectives. In contrast, its version with replacement achieves an improvement in 27 problems with respect to the three indicators used.

In addition to the above results, we decided to evaluate the performance of the two best performing strategies (SMR2_R and SMR3_R) with a different mating pool size ($\sigma_{pool} = 5$). These results are shown in Table 4.4. Concerning SMR2_R, it outperformed the original NSGA-III in 20 out of the 48 test problems with respect to the hypervolume, in 21 problems with respect to the IGD indicator, and in 27 problems with respect to the s-energy. This approach obtained good results in the DTLZ test problems with 3 objectives and in the WFG test problems with 2 objectives. However, it performed poorly in both the DTLZ and the WFG test problems with 5 objectives. On the other hand, SMR3_R improved the results obtained in 29 problems with respect to the hypervolume, in 27 problems with respect to the IGD indicator, and in 21 problems with respect to the s-energy. This approach obtained particularly good results in the DTLZ test problems with 3 objectives, and in the WFG test problems with 2 and 5 objectives.

4.4 SUMMARY

In this chapter, we presented three different mating restriction strategies as well as an experimental evaluation of their implementation in the well-known NSGA-III. From the results obtained we can state that the mating restrictions tested in our study do have an impact on the algorithms' final

Table 4.2: Comparison of the average HV, IGD and S-energy values obtained when using SMR2 with and without replacement, with $\sigma_{pool} = 3$.

Problem	Objectives	Hypervolume			IGD			S-energy		
		NSGA-III	SMR2	SMR2_R	NSGA-III	SMR2	SMR2_R	NSGA-III	SMR2	SMR2_R
DTLZ1	2	**2.1237E+00**	2.1237E+00	2.1237E+00	**1.7840E-03**	1.7841E-03	1.7841E-03	1.1725E+05	**1.1725E+05**	1.1725E+05
DTLZ2		3.2101E+00	**3.2101E+00**	3.2101E+00	**3.9625E-03**	3.9626E-03	3.9627E-03	5.3391E+04	**5.3391E+04**	5.3392E+04
DTLZ3		3.2098E+00	**3.2098E+00**	3.2097E+00	3.9867E-03	**3.9862E-03**	3.9996E-03	5.3386E+04	**5.3382E+04**	5.3385E+04
DTLZ4		2.8874E+00	2.8471E+00	**3.0488E+00**	2.0079E-01	2.2539E-01	**1.0238E-01**	3.9153E+04	**3.7374E+04**	4.6272E+04
DTLZ5		3.2101E+00	**3.2101E+00**	3.2101E+00	**3.9625E-03**	3.9626E-03	3.9627E-03	5.3391E+04	**5.3391E+04**	5.3392E+04
DTLZ6		3.0542E+00	3.0731E+00	**3.0814E+00**	9.2585E-02	8.2071E-02	**7.7864E-02**	5.4260E+04	**4.9079E+04**	4.9343E+04
DTLZ7		**4.4173E+00**	4.4173E+00	4.3635E+00	5.1851E-03	**5.1793E-03**	1.9733E-02	7.3309E+04	7.5555E+04	**6.8418E+04**
WFG1		**3.9940E+00**	3.6979E+00	3.8801E+00	**1.4331E+00**	1.4998E+00	1.4624E+00	1.2784E+05	1.5947E+05	**1.0995E+05**
WFG2		9.2570E+00	9.2427E+00	**9.2880E+00**	6.5750E-01	6.5785E-01	**6.5703E-01**	**5.9184E+04**	7.3249E+04	6.9966E+04
WFG3		1.0437E+01	1.0111E+01	**1.0849E+01**	4.3466E-02	8.0586E-02	**1.2645E-02**	2.9325E+04	3.1228E+04	**2.0534E+04**
WFG4		8.2336E+00	7.8051E+00	**8.2952E+00**	2.8252E-02	6.9665E-02	**2.7971E-02**	**2.2522E+04**	2.4655E+04	2.3629E+04
WFG5		**7.9344E+00**	7.6572E+00	7.8556E+00	**7.7568E-02**	1.0236E-01	8.9524E-02	2.2769E+04	3.0152E+04	**2.2608E+04**
WFG6		7.9733E+00	7.9159E+00	**8.3114E+00**	6.8255E-02	6.7294E-02	**3.1370E-02**	2.4668E+04	2.2878E+04	**2.1583E+04**
WFG7		6.6706E+00	6.4966E+00	**7.6059E+00**	2.9508E-01	3.4195E-01	**9.7001E-02**	4.2732E+04	3.9256E+04	**3.2833E+04**
WFG8		7.3725E+00	6.4317E+00	**7.9924E+00**	1.4150E-01	3.3014E-01	**7.3920E-02**	**4.2315E+04**	9.5149E+04	1.1297E+05
WFG9		8.1696E+00	8.1725E+00	**8.1892E+00**	3.5258E-02	3.4784E-02	**3.3108E-02**	2.2445E+04	2.4766E+04	**2.0915E+04**
DTLZ1	3	3.3477E+00	3.3489E+00	**3.3491E+00**	2.2906E-02	1.9796E-02	**1.8696E-02**	1.0500E+10	2.5160E+10	**9.4944E+05**
DTLZ2		**7.4184E+00**	7.4183E+00	7.4180E+00	**4.9314E-02**	4.9337E-02	4.9358E-02	**9.1884E+04**	1.9561E+06	2.1869E+05
DTLZ3		7.4116E+00	**7.4179E+00**	7.4178E+00	5.6253E-02	4.9430E-02	**4.9350E-02**	1.6166E+09	9.4484E+04	**7.2603E+04**
DTLZ4		7.0466E+00	6.9675E+00	**7.3173E+00**	2.2902E-01	2.4253E-01	**9.8324E-02**	3.0716E+11	4.1412E+11	**2.2249E+11**
DTLZ5		4.0043E+00	3.9703E+00	**4.1862E+00**	6.5059E-02	4.7812E-02	**1.4199E-02**	8.1297E+11	4.0529E+11	**3.4682E+11**
DTLZ6		3.9910E+00	3.9865E+00	**4.0079E+00**	1.0158E-01	1.0413E-01	**9.2818E-02**	**1.1581E+11**	2.7441E+11	1.8917E+11
DTLZ7		1.3202E+01	1.2929E+01	**1.3301E+01**	7.8863E-02	1.0705E-01	**7.0273E-02**	8.0305E+07	7.0115E+06	**1.4823E+06**
WFG1		5.4149E+01	5.4087E+01	**5.5611E+01**	1.1700E+00	1.1769E+00	**1.1434E+00**	4.1152E+08	2.6679E+09	**2.4534E+07**
WFG2		9.6392E+01	9.6983E+01	**9.7458E+01**	2.4560E-01	2.3164E-01	**2.2091E-01**	1.0067E+05	**5.0129E+04**	6.4537E+06
WFG3		2.4037E+01	**2.4047E+01**	2.3562E+01	9.3331E-02	**8.7034E-02**	1.2045E-01	**7.1353E+08**	3.7569E+10	4.6382E+10
WFG4		**7.6820E+01**	7.6790E+01	7.6730E+01	2.0001E-01	**1.9990E-01**	2.0005E-01	7.2705E+03	**5.3568E+03**	6.4978E+03
WFG5		7.3458E+01	7.3513E+01	**7.3529E+01**	2.1384E-01	**2.1271E-01**	2.1302E-01	**5.6060E+03**	2.8725E+05	1.4801E+05
WFG6		7.4081E+01	7.4086E+01	**7.4172E+01**	2.0938E-01	2.0944E-01	**2.0916E-01**	6.9892E+04	**6.1958E+03**	2.2903E+05
WFG7		**7.7045E+01**	7.7022E+01	7.7005E+01	**2.0004E-01**	2.0008E-01	2.0012E-01	2.8209E+04	2.4284E+04	**8.4907E+03**
WFG8		**7.0872E+01**	7.0336E+01	7.0272E+01	**2.5035E-01**	2.5935E-01	2.6112E-01	**1.2580E+04**	2.5467E+05	2.9949E+08
WFG9		**7.0274E+01**	6.9864E+01	6.9135E+01	**2.2365E-01**	2.2751E-01	2.3545E-01	1.9415E+08	**1.1912E+04**	3.0530E+04
DTLZ1	5	**7.5927E+00**	7.5927E+00	7.5927E+00	**5.0423E-02**	5.0651E-02	5.2614E-02	**2.2272E+10**	6.8670E+10	2.3357E+11
DTLZ2		**3.1698E+01**	3.1697E+01	3.1694E+01	1.5539E-01	**1.5531E-01**	1.5562E-01	7.7099E+09	2.0486E+10	**1.6918E+08**
DTLZ3		3.1694E+01	**3.1694E+01**	3.1691E+01	1.5825E-01	**1.5545E-01**	1.5625E-01	1.1507E+11	**5.0298E+10**	8.9069E+10
DTLZ4		3.1679E+01	**3.1698E+01**	3.1698E+01	1.6342E-01	**1.5534E-01**	1.5536E-01	**1.4004E+10**	3.3714E+10	4.1948E+10
DTLZ5		**5.9019E+00**	5.8751E+00	5.6463E+00	2.3194E-01	**2.2842E-01**	2.3629E-01	**2.3861E+11**	5.1000E+11	3.4009E+11
DTLZ6		**1.8920E+00**	2.5272E-01	0.0000E+00	**9.3461E-01**	1.3792E+00	2.4709E+00	4.5468E+11	1.4314E+11	**2.9434E+09**
DTLZ7		7.7036E+01	**7.7616E+01**	6.9739E+01	**2.7009E-01**	2.7200E-01	3.4618E-01	5.3521E+10	5.7123E+10	**3.9399E+10**
WFG1		**4.0787E+03**	3.9844E+03	4.0727E+03	**1.8441E+00**	1.8623E+00	1.8552E+00	7.1824E+10	**3.9802E+09**	5.8272E+10
WFG2		9.8846E+03	9.9906E+03	**1.0264E+04**	5.6849E-01	5.1831E-01	**4.0606E-01**	6.8679E+08	2.0875E+09	**3.1197E+08**
WFG3		8.0982E+01	**8.2007E+01**	7.5577E+01	5.2549E-01	**4.9372E-01**	5.7865E-01	6.9978E+10	8.1739E+10	**1.1726E+10**
WFG4		**8.8925E+03**	8.7860E+03	8.7304E+03	**9.1408E-01**	9.1886E-01	9.2261E-01	**2.7649E+03**	4.1867E+04	3.5352E+05
WFG5		**8.6746E+03**	8.6401E+03	8.6047E+03	9.2626E-01	**9.2503E-01**	9.2816E-01	3.0457E+08	**6.2629E+04**	1.1084E+08
WFG6		**8.8660E+03**	8.8277E+03	8.8222E+03	**9.1416E-01**	9.1648E-01	9.1647E-01	**1.4504E+07**	6.0085E+07	3.3354E+07
WFG7		**9.1345E+03**	9.0887E+03	9.0657E+03	**9.0612E-01**	9.0672E-01	9.0784E-01	**1.1485E+07**	4.2095E+08	9.5148E+07
WFG8		**8.4412E+03**	8.3391E+03	8.3235E+03	**9.5154E-01**	9.5899E-01	9.6068E-01	1.2729E+07	4.2775E+07	**2.9814E+03**
WFG9		**8.0920E+03**	8.0770E+03	8.0551E+03	9.8101E-01	**9.8052E-01**	9.8610E-01	**1.5351E+03**	6.9474E+03	3.1118E+03

Table 4.3: Comparison of the average HV, IGD and S-energy values obtained when using SMR3 with and without replacement, with $\sigma_{pool} = 3$.

Problem	Objectives	Hypervolume			IGD			S-energy		
		NSGA-III	SMR3	SMR3_R	NSGA-III	SMR3	SMR3_R	NSGA-III	SMR3	SMR3_R
DTLZ1	2	2.1237E+00	2.1237E+00	**2.1237E+00**	**1.7840E-03**	1.7842E-03	1.7842E-03	1.1725E+05	**1.1725E+05**	1.1725E+05
DTLZ2		**3.2101E+00**	3.2101E+00	3.2101E+00	**3.9625E-03**	3.9627E-03	3.9626E-03	5.3391E+04	5.3392E+04	**5.3391E+04**
DTLZ3		**3.2098E+00**	3.2097E+00	3.2096E+00	**3.9867E-03**	3.9960E-03	4.0205E-03	**5.3386E+04**	5.3587E+04	5.3423E+04
DTLZ4		2.8874E+00	2.8874E+00	**3.0084E+00**	2.0079E-01	2.0079E-01	**1.2698E-01**	3.9153E+04	**3.9153E+04**	4.4493E+04
DTLZ5		**3.2101E+00**	3.2101E+00	3.2101E+00	**3.9625E-03**	3.9627E-03	3.9626E-03	5.3391E+04	5.3392E+04	**5.3391E+04**
DTLZ6		3.0542E+00	3.0491E+00	**3.0747E+00**	9.2585E-02	9.5855E-02	**8.1569E-02**	5.4260E+04	1.6700E+05	**4.9265E+04**
DTLZ7		**4.4173E+00**	4.4172E+00	4.4173E+00	**5.1851E-03**	5.2280E-03	5.1947E-03	**7.3309E+04**	1.6963E+05	7.4962E+04
WFG1		3.9940E+00	4.4503E+00	**4.5165E+00**	1.4331E+00	1.3207E+00	**1.3035E+00**	1.2784E+05	**8.7193E+04**	1.1013E+05
WFG2		9.2570E+00	9.2665E+00	**9.2876E+00**	6.5750E-01	6.5745E-01	**6.5702E-01**	**5.9184E+04**	6.8224E+04	6.9475E+04
WFG3		1.0437E+01	1.0594E+01	**1.0860E+01**	4.3466E-02	2.7563E-02	**1.2276E-02**	2.9325E+04	2.5815E+04	**2.1805E+04**
WFG4		8.2336E+00	8.2341E+00	**8.3776E+00**	2.8252E-02	2.7391E-02	**2.2568E-02**	2.2522E+04	2.1329E+04	**2.0878E+04**
WFG5		**7.9344E+00**	7.8674E+00	7.8262E+00	**7.7568E-02**	8.3617E-02	9.6234E-02	2.2769E+04	2.3775E+04	**2.1972E+04**
WFG6		7.9733E+00	7.9980E+00	**8.2086E+00**	6.8255E-02	6.4347E-02	**4.5157E-02**	2.4668E+04	2.4087E+04	**2.1324E+04**
WFG7		6.6706E+00	6.8319E+00	**7.6937E+00**	2.9508E-01	2.5382E-01	**8.5066E-02**	4.2732E+04	4.8046E+04	**2.7849E+04**
WFG8		7.3725E+00	7.4536E+00	**8.0740E+00**	1.4150E-01	1.2743E-01	**6.7437E-02**	**4.2315E+04**	1.2405E+05	7.1553E+04
WFG9		**8.1696E+00**	8.1420E+00	8.1663E+00	3.5258E-02	3.6425E-02	**3.3411E-02**	2.2445E+04	**2.0999E+04**	2.2023E+04
DTLZ1	3	3.3477E+00	3.3491E+00	**3.3491E+00**	2.2906E-02	1.8829E-02	**1.8702E-02**	1.0500E+10	1.1762E+08	**1.6654E+07**
DTLZ2		**7.4184E+00**	7.4184E+00	7.4176E+00	**4.9314E-02**	4.9337E-02	4.9411E-02	**9.1884E+04**	8.6253E+05	1.0465E+05
DTLZ3		7.4116E+00	**7.4180E+00**	7.4179E+00	5.6253E-02	**4.9325E-02**	4.9333E-02	1.6166E+09	**8.6616E+04**	3.2032E+06
DTLZ4		7.0466E+00	7.1819E+00	**7.3172E+00**	2.2902E-01	1.6368E-01	**9.8335E-02**	3.0716E+11	9.9760E+10	**5.0508E+10**
DTLZ5		4.0043E+00	3.8906E+00	**4.1683E+00**	6.5059E-02	8.2023E-02	**1.6986E-02**	8.1297E+11	**5.2806E+11**	5.3297E+11
DTLZ6		**3.9910E+00**	3.9880E+00	3.9852E+00	**1.0158E-01**	1.0245E-01	1.0305E-01	**1.1581E+11**	2.9867E+11	1.8305E+11
DTLZ7		1.3202E+01	1.3196E+01	**1.3277E+01**	7.8863E-02	7.9276E-02	**7.1865E-02**	8.0305E+07	1.8347E+07	**1.4284E+07**
WFG1		5.4149E+01	5.4513E+01	**5.5103E+01**	1.1700E+00	**1.1577E+00**	1.1593E+00	4.1152E+08	**5.0280E+07**	6.8727E+07
WFG2		9.6392E+01	9.5372E+01	**9.9121E+01**	2.4560E-01	2.6905E-01	**1.8427E-01**	**1.0067E+05**	4.2105E+05	1.7319E+05
WFG3		**2.4037E+01**	2.4025E+01	2.3848E+01	**9.3331E-02**	9.7524E-02	1.0711E-01	7.1353E+08	7.0553E+09	**4.8357E+07**
WFG4		7.6820E+01	**7.6893E+01**	7.6863E+01	2.0001E-01	**1.9998E-01**	2.0010E-01	7.2705E+03	7.5974E+03	**5.1578E+03**
WFG5		**7.3458E+01**	7.3416E+01	7.3407E+01	2.1384E-01	**2.1377E-01**	2.1436E-01	**5.6060E+03**	5.6888E+07	6.7134E+03
WFG6		7.4081E+01	**7.4084E+01**	7.4078E+01	**2.0938E-01**	2.0965E-01	2.0966E-01	6.9892E+04	**1.0785E+04**	2.0773E+04
WFG7		7.7045E+01	**7.7047E+01**	7.7034E+01	**2.0004E-01**	2.0014E-01	2.0018E-01	**2.8209E+04**	3.4087E+04	1.8370E+08
WFG8		7.0872E+01	**7.0943E+01**	7.0904E+01	2.5035E-01	**2.4855E-01**	2.4920E-01	**1.2580E+04**	6.7374E+05	4.0960E+04
WFG9		**7.0274E+01**	7.0263E+01	6.8993E+01	**2.2365E-01**	2.2384E-01	2.3688E-01	1.9415E+08	2.4694E+06	**1.9541E+05**
DTLZ1	5	**7.5927E+00**	7.5927E+00	7.5927E+00	**5.0423E-02**	5.1071E-02	5.2628E-02	**2.2272E+10**	1.2492E+11	1.8359E+11
DTLZ2		3.1698E+01	**3.1698E+01**	3.1696E+01	**1.5539E-01**	1.5539E-01	1.5556E-01	7.7099E+09	**2.2677E+09**	9.7765E+09
DTLZ3		3.1694E+01	**3.1695E+01**	3.1693E+01	1.5825E-01	**1.5678E-01**	1.5798E-01	1.1507E+11	**2.8331E+10**	1.1363E+11
DTLZ4		3.1679E+01	3.1660E+01	**3.1698E+01**	1.6342E-01	1.7146E-01	**1.5540E-01**	1.4004E+10	1.0865E+11	**7.4339E+09**
DTLZ5		5.9019E+00	5.9270E+00	**5.9463E+00**	2.3194E-01	**2.1696E-01**	2.1813E-01	**2.3861E+11**	2.6928E+11	4.8276E+11
DTLZ6		**1.8920E+00**	1.6630E+00	2.3333E-06	**9.3461E-01**	9.7971E-01	2.0990E+00	4.5468E+11	4.3651E+11	**5.1400E+10**
DTLZ7		7.7036E+01	**7.8003E+01**	7.1112E+01	**2.7009E-01**	2.7161E-01	3.4077E-01	5.3521E+10	9.0555E+10	**3.4682E+10**
WFG1		4.0787E+03	4.2356E+03	**4.2463E+03**	1.8441E+00	**1.7866E+00**	1.7995E+00	7.1824E+10	**4.7644E+10**	1.0796E+11
WFG2		9.8846E+03	1.0025E+04	**1.0218E+04**	5.6849E-01	5.1666E-01	**4.3216E-01**	**6.8679E+08**	3.3787E+09	5.4226E+09
WFG3		8.0982E+01	8.1019E+01	**8.2890E+01**	5.2549E-01	5.2947E-01	**4.9865E-01**	6.9978E+10	2.7869E+10	**5.6526E+09**
WFG4		8.8925E+03	**8.9669E+03**	8.9299E+03	9.1408E-01	**9.1185E-01**	9.1407E-01	**2.7649E+03**	1.6509E+04	5.3241E+06
WFG5		8.6746E+03	**8.6966E+03**	8.6649E+03	**9.2626E-01**	9.2716E-01	9.2983E-01	3.0457E+08	5.2736E+08	**8.4276E+06**
WFG6		8.8660E+03	**8.8836E+03**	8.8627E+03	9.1416E-01	**9.1398E-01**	9.1480E-01	1.4504E+07	**3.1101E+06**	2.8868E+07
WFG7		9.1345E+03	**9.1625E+03**	9.1449E+03	9.0612E-01	**9.0544E-01**	9.0612E-01	1.1485E+07	1.4527E+07	**1.0765E+07**
WFG8		8.4412E+03	**8.4984E+03**	8.4860E+03	9.5154E-01	**9.4711E-01**	9.4829E-01	1.2729E+07	**5.3446E+06**	1.1486E+10
WFG9		**8.0920E+03**	8.0819E+03	8.0532E+03	**9.8101E-01**	9.8359E-01	9.8227E-01	**1.5351E+03**	2.1161E+03	1.7019E+03

Table 4.4: Comparison of the average HV, IGD and S-energy values obtained when using SMR2 and SMR3 with replacement with $\sigma_{pool} = 5$.

Problem	Objectives	Hypervolume			IGD			S-energy		
		NSGA-III	SMR2 R5	SMR3 R5	NSGA-III	SMR2 R5	SMR3 R5	NSGA-III	SMR2 R5	SMR3 R5
DTLZ1	2	**2.1237E+00**	2.1237E+00	2.1237E+00	**1.7840E-03**	1.7841E-03	1.7843E-03	**1.1725E+05**	1.1725E+05	1.1725E+05
DTLZ2		**3.2101E+00**	3.2101E+00	3.2101E+00	**3.9625E-03**	3.9625E-03	3.9626E-03	5.3391E+04	5.3391E+04	**5.3391E+04**
DTLZ3		**3.2098E+00**	3.2095E+00	3.2096E+00	**3.9867E-03**	4.0309E-03	4.0092E-03	**5.3386E+04**	5.3388E+04	5.3425E+04
DTLZ4		2.8874E+00	**3.0891E+00**	2.9681E+00	2.0079E-01	**7.7772E-02**	1.5158E-01	**3.9153E+04**	4.8052E+04	4.2713E+04
DTLZ5		**3.2101E+00**	3.2101E+00	3.2101E+00	**3.9625E-03**	3.9625E-03	3.9626E-03	5.3391E+04	5.3391E+04	**5.3391E+04**
DTLZ6		3.0542E+00	3.0779E+00	**3.0832E+00**	9.2585E-02	7.9947E-02	**7.6878E-02**	5.4260E+04	**4.9193E+04**	4.9453E+04
DTLZ7		**4.4173E+00**	4.2559E+00	4.4172E+00	**5.1851E-03**	4.8976E-02	5.1932E-03	**7.3309E+04**	1.2243E+05	7.9670E+04
WFG1		3.9940E+00	3.8339E+00	**4.6962E+00**	1.4331E+00	1.4698E+00	**1.2558E+00**	1.2784E+05	2.9172E+05	**9.6583E+04**
WFG2		9.2570E+00	9.2874E+00	**9.2883E+00**	6.5750E-01	6.5703E-01	**6.5700E-01**	**5.9184E+04**	1.3126E+05	6.3061E+04
WFG3		1.0437E+01	1.0842E+01	**1.0872E+01**	4.3466E-02	1.2716E-02	**1.1960E-02**	2.9325E+04	2.9934E+04	**2.0208E+04**
WFG4		8.2336E+00	8.2821E+00	**8.3389E+00**	2.8252E-02	3.1142E-02	**2.6048E-02**	2.2522E+04	**2.2348E+04**	2.8639E+04
WFG5		**7.9344E+00**	7.8474E+00	7.7655E+00	**7.7568E-02**	9.0569E-02	1.0578E-01	2.2769E+04	2.4668E+04	**2.2315E+04**
WFG6		7.9733E+00	**8.3237E+00**	8.1427E+00	6.8255E-02	**2.9592E-02**	5.5345E-02	2.4668E+04	**2.1575E+04**	2.2302E+04
WFG7		6.6706E+00	7.5509E+00	**7.7180E+00**	2.9508E-01	1.0718E-01	**8.1722E-02**	4.2732E+04	**3.0842E+04**	3.4039E+04
WFG8		7.3725E+00	7.8800E+00	**8.0608E+00**	1.4150E-01	8.5660E-02	**6.9887E-02**	4.2315E+04	4.6592E+04	**3.8096E+04**
WFG9		8.1696E+00	**8.1784E+00**	8.1444E+00	3.5258E-02	**3.3998E-02**	3.5096E-02	**2.2445E+04**	2.4591E+05	2.2531E+04
DTLZ1	3	3.3477E+00	3.3491E+00	**3.3491E+00**	2.2906E-02	**1.8692E-02**	1.8701E-02	1.0500E+10	1.0653E+06	**7.1158E+05**
DTLZ2		**7.4184E+00**	7.4180E+00	7.4176E+00	**4.9314E-02**	4.9369E-02	4.9411E-02	**9.1884E+04**	1.7820E+05	5.8420E+05
DTLZ3		7.4116E+00	7.4177E+00	**7.4177E+00**	5.6253E-02	**4.9352E-02**	4.9401E-02	1.6166E+09	**9.9917E+04**	4.0336E+06
DTLZ4		7.0466E+00	**7.3174E+00**	7.2495E+00	2.2902E-01	**9.8312E-02**	1.3101E-01	3.0716E+11	**1.0708E+10**	1.5578E+11
DTLZ5		4.0043E+00	4.1776E+00	**4.1783E+00**	6.5059E-02	**1.5283E-02**	1.6123E-02	8.1297E+11	8.5467E+11	**5.4758E+11**
DTLZ6		3.9910E+00	**3.9965E+00**	3.9911E+00	1.0158E-01	**9.8436E-02**	9.9011E-02	**1.1581E+11**	1.8009E+11	2.8578E+11
DTLZ7		1.3202E+01	**1.3303E+01**	1.3258E+01	7.8863E-02	**6.9520E-02**	7.1768E-02	8.0305E+07	**1.8462E+06**	2.6558E+06
WFG1		5.4149E+01	**5.4946E+01**	5.4935E+01	1.1700E+00	1.1689E+00	**1.1639E+00**	4.1152E+08	**3.8875E+06**	5.9378E+07
WFG2		9.6392E+01	9.4797E+01	**9.7478E+01**	2.4560E-01	2.8402E-01	**2.2252E-01**	**1.0067E+05**	2.5029E+06	3.1865E+05
WFG3		**2.4037E+01**	2.3528E+01	2.3832E+01	**9.3331E-02**	1.2405E-01	1.1183E-01	7.1353E+08	**1.1163E+08**	2.5357E+09
WFG4		7.6820E+01	7.6714E+01	**7.6866E+01**	2.0001E-01	**1.9992E-01**	2.0019E-01	7.2705E+03	**5.0849E+03**	5.8875E+03
WFG5		7.3458E+01	**7.3525E+01**	7.3322E+01	2.1384E-01	**2.1240E-01**	2.1457E-01	**5.6060E+03**	6.6398E+03	3.0764E+04
WFG6		7.4081E+01	7.4043E+01	**7.4101E+01**	**2.0938E-01**	2.1002E-01	2.0965E-01	6.9892E+04	**5.3176E+03**	8.4170E+04
WFG7		**7.7045E+01**	7.6997E+01	7.7031E+01	**2.0004E-01**	2.0005E-01	2.0021E-01	2.8209E+04	**8.5091E+03**	1.7101E+04
WFG8		**7.0872E+01**	7.0276E+01	7.0800E+01	2.5035E-01	2.6113E-01	**2.4855E-01**	**1.2580E+04**	2.0820E+04	6.3507E+09
WFG9		**7.0274E+01**	6.8797E+01	6.9357E+01	**2.2365E-01**	2.3891E-01	2.3173E-01	1.9415E+08	2.5239E+07	**2.7641E+04**
DTLZ1	5	**7.5927E+00**	7.5927E+00	7.5927E+00	**5.0423E-02**	5.0631E-02	5.1905E-02	**2.2272E+10**	5.6993E+10	3.1609E+11
DTLZ2		**3.1698E+01**	3.1694E+01	3.1696E+01	**1.5539E-01**	1.5564E-01	1.5562E-01	7.7099E+09	**1.9388E+08**	3.5682E+10
DTLZ3		3.1694E+01	3.1691E+01	**3.1694E+01**	1.5825E-01	**1.5565E-01**	1.5760E-01	1.1507E+11	**6.5824E+09**	1.5785E+11
DTLZ4		3.1679E+01	**3.1698E+01**	3.1679E+01	1.6342E-01	**1.5536E-01**	1.6346E-01	1.4004E+10	**3.4033E+09**	1.0034E+11
DTLZ5		5.9019E+00	5.6280E+00	**6.0829E+00**	2.3194E-01	2.5228E-01	**2.0718E-01**	**2.3861E+11**	4.2277E+11	3.8949E+11
DTLZ6		**1.8920E+00**	0.0000E+00	3.6870E-04	**9.3461E-01**	2.4951E+00	1.9476E+00	4.5468E+11	**1.8908E+10**	3.8261E+10
DTLZ7		**7.7036E+01**	6.9028E+01	7.2048E+01	**2.7009E-01**	3.6074E-01	3.2807E-01	5.3521E+10	**7.1014E+09**	1.5851E+10
WFG1		4.0787E+03	4.1560E+03	**4.2788E+03**	1.8441E+00	1.8413E+00	**1.8074E+00**	7.1824E+10	**6.2041E+10**	1.0491E+11
WFG2		9.8846E+03	**1.0260E+04**	1.0222E+04	5.6849E-01	**4.0650E-01**	4.3222E-01	6.8679E+08	**3.3836E+05**	2.1037E+10
WFG3		8.0982E+01	7.5389E+01	**8.3216E+01**	5.2549E-01	5.8067E-01	**5.0038E-01**	6.9978E+10	**6.7758E+09**	1.6107E+10
WFG4		8.8925E+03	8.6764E+03	**8.9734E+03**	9.1408E-01	9.2611E-01	**9.1228E-01**	2.7649E+03	**1.6412E+03**	7.5004E+03
WFG5		8.6746E+03	8.5996E+03	**8.6801E+03**	**9.2626E-01**	9.2728E-01	9.3455E-01	3.0457E+08	**7.9671E+07**	2.3247E+10
WFG6		8.8660E+03	8.8006E+03	**8.8703E+03**	**9.1416E-01**	9.1731E-01	9.1502E-01	1.4504E+07	**4.4048E+05**	2.4931E+08
WFG7		9.1345E+03	9.0514E+03	**9.1579E+03**	**9.0612E-01**	9.0822E-01	9.0628E-01	**1.1485E+07**	4.6832E+07	1.5277E+09
WFG8		8.4412E+03	8.3023E+03	**8.5127E+03**	9.5154E-01	9.6270E-01	**9.4652E-01**	**1.2729E+07**	5.1636E+07	8.7401E+07
WFG9		**8.0920E+03**	8.0536E+03	8.0699E+03	9.8101E-01	9.8377E-01	**9.8053E-01**	**1.5351E+03**	6.5673E+03	4.2882E+06

convergence. However, no single mating restriction was able to improve all the test problem instances adopted. The best overall performing strategy was SMR3_R, which outperformed the original algorithm in 60.4% of the test problems adopted considering the hypervolume indicator.

Regarding similar vs. dissimilar contributions pairing, our experimental results indicated that the combination of individuals with a similar contribution was slightly better than the alternative. Additionally, the replacement strategy proved to be very useful in both SMR2 and SMR3, since it not only increased the number of problems improved, but it also produced higher hypervolume values in most of the test instances. This may be a direct consequence of exploiting the area surrounding the individuals with the best s-energy contribution (i.e., the individuals in the least crowded regions). Moreover, the change of mating pool size caused a good performance of SMR3_R in MOPs with 5 objectives. However, it caused a slight worsening in MOPs with 2 objectives. This is more evidence that the mating restriction effect seems to be problem-dependent and scale-dependent.

Because of these reasons, we considered that the best way of continuing this work was to propose a mating restriction meta-strategy, which is able to employ different strategies depending on information obtained from the particular problem being solved. This proposal is presented in the next chapter.

5

Ensemble of S-energy based mating restrictions

5.1 Introduction

The s-energy based mating restrictions (SMR) presented in the previous chapter exhibit a certain amount of improvement in some test problems. However, no single SMR was able to produce a significant amount of improvements in the results since they seem to be both problem and scale dependent. Hence, we decided to combine them in an ensemble of mating restrictions which is able to decide which SMR to use at different points during the execution of the MOEA.

5.1.1 S-energy based mating restrictions used

We selected the following four SMRs to create our ensemble of mating restrictions:

- SMR1_SIM: This restriction mates individuals with similar s-energy contribution values. The first pair is formed with the two best individuals from the population. The next pair will select the next two best individuals from the remaining individuals, and so on, until all pairs are formed.
- SMR1_DIS: This restriction pairs individuals with dissimilar s-energy contributions. The first pair selects the best individual and the worst individual. The next pair mates the second best individual with the second worst. This is repeated with the remaining individuals.

- SMR3: This restriction employs a mating pool with size $\sigma_{pool} > 0$, which is a user defined parameter. In this pool are contained the worst individuals from the population which have not been selected yet. SMR3 pairs the best individual from the population with one of the individuals in the mating pool. In order to select one individual from the mating pool, Euclidean distances in objective space are measured and the individual with the smallest distance is selected. In our ensemble we adopted this restriction twice: the first one with $\sigma_{pool} = 5$, and the second one with $\sigma_{pool} = 9$.

We combined these SMRs to create an ensemble of mating restrictions which is described in detail in the following section.

5.2 ENSEMBLE OF sENERGY BASED MATING RESTRICTIONS

The ensemble we propose combines the simultaneous use of four SMRs at each generation. However, since we already have experimental evidence that no single SMR is able to improve the quality of the results obtained across different benchmark problems, we propose the use of two metrics to adapt the number of pairs that will be generated at each generation using each of the restrictions. In doing so, we allow different SMRs to be more predominantly used at certain points of the search process according to their performance in the past generations. This is because one SMR may be better than the rest during the first stages of the algorithm, but not in the latter, or there could be one SMR that works really well with one type of problem, but that performs badly in other problems. In order to measure the performance of each SMR, we propose the use of the following two metrics:

- Mating restriction's efficiency: It is the percentage of children generated by each SMR that were selected to survive in the next generation. It is obtained by computing the quotient of the number of selected individuals which are offspring of a certain mating restriction divided by the total number of children generated by this restriction so far.

- Mating restriction's dominance: It is the sum of individuals generated by each mating restriction which dominate either (or both) of their parents. In contrast with efficiency, we propose this value to be obtained using only the information from the last few generations, determined by a user-defined parameter $t_r > 0$.

The idea behind using two metrics is to obtain a combination of "global" and "local" information, in the context of the search process performed by the algorithm. In this sense, we use mating restriction's dominance to obtain "local" information since it only considers the last few generations. On the other hand, we use mating restriction's efficiency as a "global" information indicator, since it stores how efficient each restriction has been from the beginning of the algorithm. We alternate between these two metrics in the following way.

During the first t_r generations, we use mating restriction's dominance to determine the number of pairs to be obtained with each restriction, allowing the ones which have generated better children in the last generations to be the ones with more offspring in the next generation. Here, we call better children to the ones that Pareto-dominate their parents. However, every t_r generations the dominance metric will be reset to zero (in order for it to reflect local behavior information), and at this restarting point mating restriction's efficiency will be used to assign the number of pairs to be selected with each restriction instead of dominance. This is to ensure that restrictions which have proven to be the most useful in the solution of the current problem keep being used throughout the execution, even if they may not be the ones with better offspring in a given particular generation. In this work, we propose the use of $t_r = 5$, which we determined to be a good value after experimental validation.

We show the pseudocode of a generic MOEA implementing our ensemble in Algorithm 1. Since this ensemble can be implemented in a wide variety of MOEAs, we do not get into the details of some generic steps such as the population initialization (line 1), the individuals' crossover and mutation (lines 17 and 18) and selection of individuals which will survive to the next generation (line 23).

In lines 2-10 we show the ensemble's variables initialization. For the very first generation, each mating restriction will be assigned an equal number of pairs (mating restriction size) to be selected with it (line 5). Then, during the main loop of the algorithm (lines 11-38), the following will occur. In line 12 we will obtain the pairing from each restriction in the ensemble, according to the mating restriction size, previously set. Next, in lines 13-22 we will generate the offspring population using the pairing generated using the SMRs from the ensemble. In this part, we use a procedure to store the dominance information of each pair of children generated for each SMR (line 19). In line 23, the final population of this generation is selected (either directly from the offspring population or from a combination of offspring and parents population). Here, we require to count how many individu-

Algorithm 1: Main algorithm of the mating restrictions ensemble.

Input: MOP, t_r

MOEA_InitializePopulation$(\vec{P}_p)$
$n_r \leftarrow 4$
$t \leftarrow 0$
for $i \leftarrow 1$ **to** n_r **do**
    $mr_{size}[i] \leftarrow \vec{P}_p.size/n_r$
    $mr_{dom}[i] \leftarrow 0$
    $mr_{effic}[i] \leftarrow 0$
    $mr_{offspring}[i] \leftarrow 0$
    $mr_{selected}[i] \leftarrow 0$
end
while *stopping criterion not fulfilled* **do**
    $mr_{pairing} \leftarrow$ **MR_select**$(\vec{P}_p, mr_{size})$
    $\vec{P}_0 \leftarrow \emptyset$
    for $i \leftarrow 1$ **to** $\vec{P}.size$ **do**
        $p_1 \leftarrow \vec{P}[mr_{pairing}[i]]$
        $p_2 \leftarrow \vec{P}[mr_{pairing}[i+1]]$
        $c_1, c_2 \leftarrow$ **MOEA_crossover**(p_1, p_2)
        $c_1, c_2 \leftarrow$ **MOEA_mutate**(p_1, p_2)
        MR_dominates$(mr_{dom}, p_1, p_2, c_1, c_2)$
        $\vec{P}_0 \leftarrow \vec{P}_0 \cup \{c_1, c_2\}$
        $i \leftarrow i + 2$
    end
    $\vec{P}_p, mr_{selected} \leftarrow$ **MOEA_select** $(\vec{P}_p, \vec{P}_0)$
    for $i \leftarrow 1$ **to** n_r **do**
        $mr_{offspring}[i] \leftarrow mr_{offspring}[i] + mr_{size}[i]$
        $mr_{effic}[i] \leftarrow mr_{selected}[i]/mr_{offspring}[i]$
    end
    if $t == t_r$ **then**
        MR_adjust($\vec{P}_p, mr_{size}, mr_{effic}$)
        for $i \leftarrow 1$ **to** n_r **do**
            $MR_{dom}[i] \leftarrow 0$
        end
        $t \leftarrow 0$
    else
        MR_adjust$(\vec{P}_p, mr_{size}, mr_{dom})$
    end
    $t \leftarrow t + 1$
end

als generated with each mating restriction made it into the final population to update the efficiency of each restriction. Next, we update this efficiency (lines 24-27). Finally, if t_r generations have already passed, mating restrictions' dominance will be set to zero and mating restrictions' efficiency will be used to assign each restriction size (lines 29-33). If it is not the case, mating restrictions' dominance will be used instead (line 35). The auxiliary procedures of this algorithm are shown in Algorithms 2, 3 and 4.

First, the mating restrictions ensemble selection mechanism is shown in Algorithm 2, and it simply consists of obtaining the regular pairing from each individual mating restriction (lines 4-6), and then the selection of the first pairs from each pairing, according to the predefined mating restriction size (lines 7-13).

Algorithm 2: MR_SELECT

Input : $\vec{P}, mr_{size}$
Output: $mr_{pairing}$

```
1  n_r ← 4
2  counter ← 0
3  mr_pairing ← {0, 0, ..., 0}
4  for i ← 1 to n_r do
5  |   Pair_i = SMR_i(P⃗)
6  end
7  for i ← 1 to n_r do
8  |   for j ← 1 to mr_size[i] do
9  |   |   mr_pairing[counter] = Pair_i[j]
10 |   |   mr_pairing[counter + 1] = Pair_i[j + 1]
11 |   |   counter ← counter + 2
12 |   end
13 end
14 return mr_pairing
```

Next, the update of the dominance metric is shown in Algorithm 3. Given a pair of parents and their corresponding children, we add the number of parents that each child dominates (lines 2-8). Then, we store this value adding it to the previous dominance value of the corresponding SMR (lines 9-10).

Finally, in Algorithm 4, we show the procedure used to adjust each mating restriction size according to a given metric (either efficiency or dominance, since the steps are the same for both of them). Given the metric,

Algorithm 3: MR_DOMINATES

Input: $mr_{dom}, p_1, p_2, c_1, c_2$

$counter \leftarrow 0$
for $c_i \in \{c_1, c_2\}$ **do**
    for $p_i \in \{p_1, p_2\}$ **do**
        if c_i *Pareto dominates* p_i **then**
            $counter \leftarrow counter + 1$
        end
    end
end
$mr \leftarrow$ mating restriction used to generate p_1 and p_2
$mr_{dom}[mr] \leftarrow mr_{dom}[mr] + counter$

we compute its total sum (line 2) and then we use direct proportions to assign each mating restriction size (lines 3-5), which means that the number of pairs that will be generated with each SMR in the next generation is directly proportional to each restriction's contribution to the metric sum.

Algorithm 4: MR_ADJUST

Input: $\vec{P}, mr_{size}, mr_{metric}$

1 $n_r \leftarrow 4$
2 $metric\_sum \leftarrow \sum_{i=1}^{n_r} mr_{metric}[i]$
3 **for** $i \leftarrow 1$ **to** n_r **do**
4     $mr_{size}[i] = mr_{metric}[i]/metric\_sum * \vec{P}.size$
5 **end**

The computational cost of implementing our mating restrictions ensemble is directly related to the cost of computing s-energy contributions of each individual in the population. This is because these contributions are required for all the mating restrictions adopted in the ensemble, and because it is the most computational time consuming part of the algorithm.

Using a naive approach, the calculation of an individual's s-energy contribution is $O(n^2)$, being n the size of the population. However, since we need to obtain the contribution of each individual, the total computational cost raises up to $O(n^3)$ per generation. However, this can be done more efficiently, using the memoization structure proposed in [37], producing a total computational cost of $O(n^2)$ to compute all the individuals' s-energy contributions. Once these contributions are obtained, this information can be used by all four mating restrictions, so it will only be performed once per generation.

On the other hand, the mating restriction selection procedure has a computational cost of $O(n)$, while the dominance metric update has a computational cost of $O(nm^2)$, being m the number of objectives. Both the efficiency metric update and the mating restriction size adjustment procedure have a computational cost of $O(n_r)$, being $n_r = 4$ the number of mating restrictions used in the ensemble.

Finally, the total cost of implementing our ensemble is $O(n^2 + nm^2)$ per generation, which reduces to $O(n^2)$ for populations with over 100 individuals and problems with up to 10 objectives.

5.3 EXPERIMENTAL VALIDATION

In order to validate the functionality of our proposed ensemble, we implemented it in NSGA-III, so that we could solve a series of test problems with and without the ensemble to compare the results obtained. Once again we adopted the DTLZ and the WFG test suites. From these, we used DTLZ1-DTLZ7 and WFG1-WFG9 with 2-7 objectives, to assess how well the ensemble performs in problems with many objectives (more than three). This gives a total of 96 test problems, each of which we solved 30 times using both the original NSGA-III implementation and the one with the ensemble of mating restrictions incorporated within it. We adopted three performance indicators to compare the Pareto approximations obtained in each of these test instances. The indicators used were the hypervolume (HV), the inverted generational distance (IGD), and the s-energy. The performance indicator values obtained were compared using the Wilcoxon rank-sum test at a confidence interval of 95%.

Our experimental results are shown in Tables 5.1- 5.3. In each pair of columns, the best value is shown in **boldface**, while the cells in gray represent the values that are statistically better according to the Wilcoxon test used.

From Table 5.1, we can observe that the use of our ensemble of mating restrictions produced a better performance in 57 out of the 96 test instances when comparing with respect to the hypervolume, while it only had a worst performance in one test problem (DTLZ3 with 3 objectives). In the remaining 38 problems, results are statistically similar at a confidence interval of 95%. From Table 5.2, regarding s-energy values, the ensemble outperformed the original algorithm in 20 test problems, whereas the orig-

Table 5.1: Comparison of the average hypervolume values obtained using the proposed mating restrictions ensemble in NSGA-III.

Problem	Number of objectives	Hypervolume		Number of objectives	Hypervolume	
		NSGA-III	NSGA-III+Ensemble		NSGA-III	NSGA-III+Ensemble
DTLZ1	2	5.3357E-01	**5.3368E-01**	5	**2.9055E-01**	2.9054E-01
DTLZ2		4.1999E-01	**4.2013E-01**		1.3054E+00	**1.3064E+00**
DTLZ3		**4.1811E-01**	4.1800E-01		3.9891E+05	**3.9891E+05**
DTLZ4		3.3742E-01	**3.6844E-01**		1.3021E+00	**1.3021E+00**
DTLZ5		4.1999E-01	**4.2013E-01**		1.2508E+02	**1.2619E+02**
DTLZ6		4.6875E+00	**4.7444E+00**		6.2840E+03	**6.3378E+03**
DTLZ7		7.3494E-01	**7.3499E-01**		3.0123E+00	**3.0622E+00**
WFG1		**6.1667E-01**	5.9821E-01		1.0342E+02	**1.0699E+02**
WFG2		1.3095E+00	**1.3119E+00**		**3.4511E+03**	3.3359E+03
WFG3		3.8506E+00	**3.9094E+00**		**2.5916E+03**	2.5810E+03
WFG4		2.2182E+00	**2.2363E+00**		3.0867E+03	**3.2460E+03**
WFG5		1.9551E+00	**1.9620E+00**		3.0391E+03	**3.1292E+03**
WFG6		**2.6415E+00**	2.6216E+00		2.7510E+03	**2.8592E+03**
WFG7		2.0603E+00	**2.1096E+00**		2.9288E+03	**3.0308E+03**
WFG8		3.2720E+00	**3.3934E+00**		3.2599E+03	**3.3744E+03**
WFG9		**2.2130E+00**	2.2053E+00		2.8869E+03	**2.9086E+03**
DTLZ1	3	**8.7400E-01**	8.7383E-01	6	6.5860E+00	**6.5860E+00**
DTLZ2		7.4898E-01	**7.4905E-01**		1.7054E+00	**1.7078E+00**
DTLZ3		**7.4327E-01**	7.3947E-01		**1.9893E+06**	1.9893E+06
DTLZ4		6.4110E-01	**6.8052E-01**		1.5375E+00	**1.5453E+00**
DTLZ5		1.3322E-01	**1.3872E-01**		4.1175E+02	**4.1222E+02**
DTLZ6		4.2607E+01	**4.2730E+01**		5.6260E+04	**5.8987E+04**
DTLZ7		1.5000E+00	**1.5079E+00**		3.5956E+00	**3.6867E+00**
WFG1		3.0108E+01	**3.1520E+01**		1.8159E+01	**1.8701E+01**
WFG2		3.9822E+01	**4.0654E+01**		**3.4097E+04**	3.3910E+04
WFG3		2.6393E+01	**2.6680E+01**		2.3652E+04	**2.3992E+04**
WFG4		2.3866E+01	**2.4057E+01**		3.9876E+04	**4.2229E+04**
WFG5		2.2018E+01	**2.2069E+01**		4.0046E+04	**4.1580E+04**
WFG6		2.2226E+01	**2.2362E+01**		4.1404E+04	**4.3121E+04**
WFG7		2.4337E+01	**2.4397E+01**		3.8606E+04	**4.0161E+04**
WFG8		2.4152E+01	**2.4370E+01**		4.2133E+04	**4.3719E+04**
WFG9		**2.3192E+01**	2.2571E+01		4.0286E+04	**4.0794E+04**
DTLZ1	4	**3.1888E+00**	3.1888E+00	7	**4.0388E+00**	4.0387E+00
DTLZ2		**1.1648E+00**	1.1644E+00		1.7665E+00	**1.7699E+00**
DTLZ3		7.7152E+01	**7.7161E+01**		1.3269E+14	1.3269E+14
DTLZ4		1.0208E+00	**1.1058E+00**		2.5112E+00	**2.5227E+00**
DTLZ5		3.8983E+00	**3.9205E+00**		**7.3010E+02**	7.2946E+02
DTLZ6		8.4201E+02	**8.4454E+02**		3.7784E+05	**3.9944E+05**
DTLZ7		2.2224E+00	**2.2505E+00**		3.7340E+00	**3.8467E+00**
WFG1		2.0317E+02	**2.1846E+02**		3.7775E+00	**4.2824E+00**
WFG2		**3.5401E+02**	3.5210E+02		**4.0164E+05**	3.9833E+05
WFG3		3.2675E+02	**3.2734E+02**		5.5539E+05	**5.6725E+05**
WFG4		2.4600E+02	**2.4978E+02**		5.9929E+05	**6.1767E+05**
WFG5		2.3152E+02	**2.3346E+02**		6.0194E+05	**6.2018E+05**
WFG6		2.3278E+02	**2.3480E+02**		5.4668E+05	**5.6211E+05**
WFG7		2.5949E+02	**2.6122E+02**		5.7509E+05	**5.9665E+05**
WFG8		2.8328E+02	**2.8470E+02**		7.3226E+05	**7.3653E+05**
WFG9		**2.3721E+02**	2.3435E+02		**5.6265E+05**	5.6151E+05

Table 5.2: Comparison of the average s-energy values obtained using the proposed mating restrictions ensemble in NSGA-III.

Problem	Number of objectives	S-energy		Number of objectives	S-energy	
		NSGA-III	NSGA-III+Ensemble		NSGA-III	NSGA-III+Ensemble
DTLZ1		1.1956E+05	**1.1723E+05**		1.6853E+11	**6.9391E+10**
DTLZ2		5.3542E+04	**5.3392E+04**		**2.0765E+10**	1.3373E+11
DTLZ3		**5.3496E+04**	5.4529E+04		**1.1463E+11**	1.3830E+11
DTLZ4		**3.9162E+04**	4.4493E+04		**8.0186E+10**	1.3632E+11
DTLZ5		5.3542E+04	**5.3392E+04**		**3.2513E+11**	3.1227E+12
DTLZ6		**5.1342E+04**	7.4879E+04		**2.7431E+10**	1.5967E+11
DTLZ7		**8.6183E+04**	8.8229E+05		**1.8919E+10**	1.0755E+11
WFG1	2	1.3790E+05	**1.0715E+05**	5	2.4901E+10	**1.5357E+10**
WFG2		1.3326E+05	**6.6367E+04**		**3.6792E+07**	6.7396E+09
WFG3		2.8483E+04	**2.5233E+04**		**3.5651E+10**	1.6769E+11
WFG4		**2.5083E+04**	2.9992E+04		**1.3402E+03**	5.7594E+05
WFG5		2.3735E+04	**2.1515E+04**		**2.1655E+06**	7.1581E+07
WFG6		**2.3202E+04**	2.4043E+04		**3.3400E+06**	9.9717E+07
WFG7		4.0793E+04	**3.5452E+04**		**1.1611E+06**	2.0022E+10
WFG8		**5.6673E+04**	8.1960E+04		**9.5577E+05**	3.7906E+07
WFG9		**2.3704E+04**	1.9445E+05		8.6278E+04	**3.7789E+03**
DTLZ1		**3.6472E+06**	5.0248E+09		2.2901E+05	**2.0608E+05**
DTLZ2		**1.8627E+05**	1.5963E+06		**6.6797E+04**	6.6926E+04
DTLZ3		**1.1968E+06**	1.5548E+07		6.6059E+04	**6.1406E+04**
DTLZ4		5.2214E+11	**2.8705E+11**		**7.7555E+04**	7.8952E+04
DTLZ5		7.4448E+11	**6.9681E+11**		**6.8224E+04**	9.4598E+04
DTLZ6		**1.4683E+11**	5.2693E+11		**1.5554E+04**	1.7993E+05
DTLZ7		**1.7976E+06**	1.9217E+10		**4.1904E+04**	4.3668E+04
WFG1	3	**1.4561E+06**	1.2302E+07	6	**8.1161E+04**	1.0444E+05
WFG2		**1.5196E+05**	1.4073E+06		**2.8538E+04**	2.8846E+04
WFG3		**1.9949E+07**	1.1890E+10		**2.3608E+04**	9.0770E+04
WFG4		**5.1545E+03**	5.4314E+03		9.5086E+03	**9.4351E+03**
WFG5		1.1436E+05	**1.3130E+04**		9.5051E+03	**9.4559E+03**
WFG6		1.7143E+04	**1.1931E+04**		9.4639E+03	**9.4293E+03**
WFG7		3.1662E+05	**2.6767E+04**		9.4405E+03	**9.4211E+03**
WFG8		1.1761E+08	**4.1370E+05**		9.6780E+03	**9.6254E+03**
WFG9		1.4636E+04	**8.1013E+03**		1.0119E+04	**1.0014E+04**
DTLZ1		2.8918E+09	**4.5400E+06**		**2.8691E+11**	1.8059E+12
DTLZ2		**8.4928E+04**	8.4942E+04		**2.7608E+05**	2.8053E+05
DTLZ3		3.9478E+07	**2.1937E+07**		**7.1103E+11**	1.8082E+12
DTLZ4		2.1598E+11	**6.8190E+10**		**9.3437E+11**	3.9384E+12
DTLZ5		**5.2442E+11**	2.5628E+12		**1.4659E+12**	3.2859E+12
DTLZ6		**1.9505E+11**	4.2948E+11		**1.4011E+10**	7.1655E+10
DTLZ7		**1.7251E+08**	5.6368E+09		**1.1474E+11**	4.4794E+11
WFG1	4	**1.2278E+09**	2.6709E+09	7	**7.5017E+11**	4.0344E+12
WFG2		**5.0981E+07**	6.2388E+09		**2.0529E+11**	2.3815E+11
WFG3		**3.8606E+09**	1.1193E+11		**3.4851E+11**	1.3329E+12
WFG4		1.2469E+03	**1.2170E+03**		**1.3301E+02**	4.5438E+09
WFG5		1.4598E+03	**1.4079E+03**		2.0522E+02	**1.8461E+02**
WFG6		1.2446E+03	**1.2099E+03**		9.7861E+01	**8.4634E+01**
WFG7		1.1993E+03	**1.1790E+03**		9.8455E+01	**8.4844E+01**
WFG8		**1.6037E+06**	5.1743E+06		**4.8377E+09**	2.3037E+11
WFG9		1.5574E+03	**1.5563E+03**		**1.7632E+02**	5.3218E+04

Table 5.3: Comparison of the average IGD values obtained using the proposed mating restrictions ensemble in NSGA-III.

Problem	Number of objectives	IGD		Number of objectives	IGD	
		NSGA-III	NSGA-III+Ensemble		NSGA-III	NSGA-III+Ensemble
DTLZ1	2	1.8466E-03	**1.7915E-03**	5	**5.0368E-02**	5.0654E-02
DTLZ2		4.0073E-03	**3.9632E-03**		1.5560E-01	**1.5548E-01**
DTLZ3		4.4526E-03	**4.4405E-03**		2.0915E-01	**1.9739E-01**
DTLZ4		2.0079E-01	**1.2698E-01**		1.6338E-01	**1.6333E-01**
DTLZ5		4.0073E-03	**3.9632E-03**		2.3114E-01	**1.9705E-01**
DTLZ6		1.0889E-01	**8.1561E-02**		1.9394E+00	**1.7942E+00**
DTLZ7		5.1686E-03	**5.1674E-03**		2.7940E-01	**2.6670E-01**
WFG1		1.5035E+00	**1.4348E+00**		1.9929E+00	**1.9588E+00**
WFG2		6.5824E-01	**6.5764E-01**		**5.0293E-01**	6.4402E-01
WFG3		6.9185E-02	**3.5092E-02**		**5.9354E-01**	6.2907E-01
WFG4		4.8849E-02	**3.3118E-02**		9.4493E-01	**9.2570E-01**
WFG5		9.2752E-02	**8.4103E-02**		9.5438E-01	**9.3867E-01**
WFG6		**6.3791E-02**	6.7504E-02		9.3486E-01	**9.2226E-01**
WFG7		3.6238E-01	**2.7336E-01**		9.2918E-01	**9.1124E-01**
WFG8		2.1185E-01	**1.7505E-01**		9.9004E-01	**9.6865E-01**
WFG9		**3.9788E-02**	4.0386E-02		**1.0043E+00**	1.0102E+00
DTLZ1	3	**1.8958E-02**	1.9614E-02	6	6.4333E-02	**6.2869E-02**
DTLZ2		4.9361E-02	**4.9359E-02**		2.1009E-01	**2.0974E-01**
DTLZ3		**4.9667E-02**	5.0218E-02		**2.6801E-01**	3.7702E-01
DTLZ4		2.2900E-01	**1.6367E-01**		2.2195E-01	**2.1492E-01**
DTLZ5		5.9339E-02	**2.3632E-02**		3.5843E-01	**3.3632E-01**
DTLZ6		1.2659E-01	**1.1320E-01**		3.6113E+00	**3.1331E+00**
DTLZ7		**6.8762E-02**	7.0361E-02		4.1830E-01	**3.8385E-01**
WFG1		1.2766E+00	**1.2311E+00**		2.2494E+00	**2.2314E+00**
WFG2		3.3398E-01	**2.9433E-01**		**7.5766E-01**	7.9188E-01
WFG3		1.1101E-01	**1.0206E-01**		**8.5291E-01**	8.7398E-01
WFG4		2.0165E-01	**2.0121E-01**		1.4087E+00	**1.3843E+00**
WFG5		2.1580E-01	**2.1530E-01**		1.4132E+00	**1.3932E+00**
WFG6		2.1430E-01	**2.1276E-01**		1.3901E+00	**1.3730E+00**
WFG7		**2.0034E-01**	2.0050E-01		1.3870E+00	**1.3647E+00**
WFG8		2.6456E-01	**2.5984E-01**		1.4386E+00	**1.4086E+00**
WFG9		**2.2803E-01**	2.3811E-01		1.5206E+00	**1.5023E+00**
DTLZ1	4	**4.0922E-02**	4.1081E-02	7	**8.8486E-02**	9.3266E-02
DTLZ2		**1.1382E-01**	1.1385E-01		2.8168E-01	**2.8133E-01**
DTLZ3		1.2657E-01	**1.1775E-01**		**7.6479E-01**	1.0279E+00
DTLZ4		2.9346E-01	**1.9139E-01**		3.1834E-01	**3.0798E-01**
DTLZ5		1.1110E-01	**8.8104E-02**		5.1439E-01	**5.0450E-01**
DTLZ6		5.7990E-01	**4.7092E-01**		3.8287E+00	**3.1421E+00**
DTLZ7		2.0624E-01	**2.0128E-01**		6.0591E-01	**5.9054E-01**
WFG1		1.6261E+00	**1.5393E+00**		2.6143E+00	**2.5898E+00**
WFG2		**4.5992E-01**	4.7892E-01		1.1664E+00	**1.1423E+00**
WFG3		**2.9053E-01**	3.0261E-01		9.0588E-01	**8.3514E-01**
WFG4		**5.6932E-01**	5.7066E-01		**2.0410E+00**	2.0628E+00
WFG5		**5.6728E-01**	5.6884E-01		**2.0456E+00**	2.0481E+00
WFG6		5.6924E-01	**5.6907E-01**		2.0245E+00	**2.0205E+00**
WFG7		**5.7119E-01**	5.7204E-01		2.0327E+00	**2.0233E+00**
WFG8		6.2484E-01	**6.2229E-01**		**2.1442E+00**	2.2305E+00
WFG9		5.7438E-01	**5.7430E-01**		2.1393E+00	**2.1357E+00**

inal algorithm obtained better values in 24 test problems. Finally, from Table 5.3, the ensemble improved the results obtained in 45 problems when comparing IGD values, while the original algorithm only had better results in 7 test problems.

From these results, we can observe that the use of our ensemble does improve the performance of NSGA-III in more than half of the test problems when comparing HV values. In particular, good results were obtained in problems with 5 and 6 objectives (there were improvements in 68% of the test problems adopted) and in problems with 2 and 3 objectives (improvement of 62% of the test problems adopted). On the other hand, the worst performing scenarios were the problems with 4 and 7 objectives (43% and 50% of the test problems improved, respectively). Regarding the test problems adopted, the WFG test suite was the one with the largest number of problems improved (around 64%), while the DTLZ test problems had a smaller improvement (52%).

Regarding the IGD values, a similar behavior was obtained, being the problems with 6 and 2 objectives the ones with a larger number of problems improved (68% and 62%, respectively). However, problems with 3 and 4 objectives were the least improved (only 31%).

Concerning the s-energy results, this was the only performance indicator for which the original algorithm obtained a better overall performance than the ensemble. The only exceptions are problems with 2, 4 and 6 objectives, where the ensemble improved more problems than the original algorithm.

In Fig. 5.1 we show the average execution time required to solve all 16 test problems used with the different numbers of objectives. From these values, we can observe that there is a slight increase in execution time due to the use of the mating restriction ensemble. Nonetheless, such time increase is, in all cases, smaller than one second.

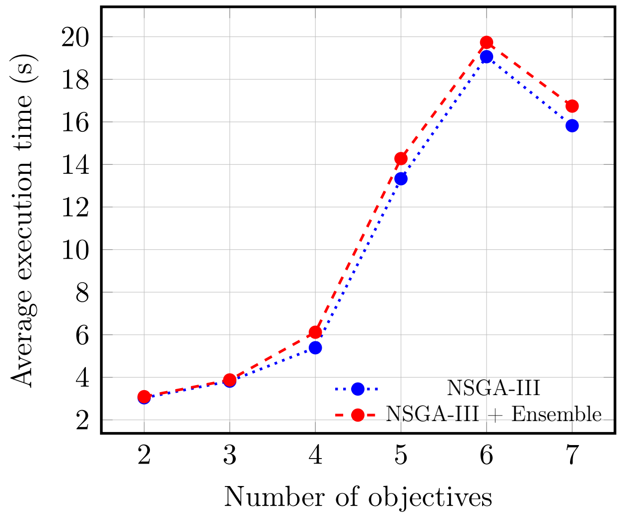


Figure 5.1: Average execution time of our proposal compared against the original NSGA-III.

5.4 Summary

In this chapter, we presented an ensemble of four s-energy based mating restrictions along with its experimental validation, where we implemented it in NSGA-III and compare the obtained results against the standard version of this same MOEA. We proposed the use of two performance measures to balance the number of pairs generated with each of the SMRs according to their performance in the current execution. This feature provides flexibility to our proposal, allowing different SMRs to be more predominantly used for different problems and even at different moments of the search process.

The experimental results show that the implementation of our ensemble improves more than half of the test instances used (57 out of 96), which comprised MOPs with 2-7 objectives. It is important to notice that in addition to the improvement of these test instances, the remanining instances obtained a similar performance to the original NSGA-III (with the exception of only one test instance). Hence, our ensemble allows to obtain similar or better results in most cases without a degradation of the performance in a considerable group of problems. This is a major improvement compared to the performance of individual SMRs, which were able to improve results in certain test instances but at the expense of worsening several others. We also believe that the experimental results obtained with our ensemble are clearly indicate that the use of mating restrictions is a viable way of improving MOEA results even in many-objective problems.

6

Use of Grammatical Evolution to design new scalarizing functions

6.1 Introduction

MOEAs can be broadly classified into 3 categories: Pareto dominance-based, decomposition-based and indicator-based [116]. In this chapter, we present an alternative to generate new elements that can improve the performance of decomposition-based MOEAs, namely the generation of new scalarizing functions using grammatical evolution.

Decomposition-based MOEAs work by transforming a MOP into two or more single-objective optimization problems, which are solved simultaneously using neighborhood search [134]. One of the advantages of this sort of MOEAs is that they are not easily affected by selection pressure issues [124], which makes them particularly useful in MOPs with many objectives (4 or more). These MOEAs use scalarizing functions to aggregate the multiple objective functions into a single function. There are multiple scalarizing functions with different properties, such as the optimality of the solutions found (weak/strong Pareto optimality), as well as with different requirements (reference points, utopian/Nadir objective vector, aspiration levels or additional classifications) [83].

The performance of decomposition-based MOEAs is closely related to both, the scalarizing function adopted, which can determine the type of solutions found [92], as well as the weight vectors used, since they strongly determine the distribution of such solutions [137]. In this work, we propose a methodology to produce new scalarizing functions to improve the performance of these MOEAs.[1] It is important to mention that here, we extend the methodology originally presented in [96], by using a different (improved) implementation, as well as by including a comprehensive series of experiments and a new application which is described in the next chapter.

Genetic Programming (GP) is a well-known evolutionary computation technique that allows the automatic generation of computer programs to solve a given problem. Similar to other evolutionary techniques, this is achieved by genetically breeding a population of individuals and applying genetic operators iteratively until a certain termination criteria is met. However, the most distinctive feature of GP is that the individuals encode computer programs, usually using tree data structures [73]. On the other hand, Grammatical Evolution (GE) is a well-known variant of GP. They both share many similarities, but GE is a grammar-based form of GP. Hence, whereas GP's individuals usually consist of trees, GE genotypes are usually integer or binary lists [90]. Consequently, one of the main advantages of GE is its versatility and that it can be easily applied to different problem domains. The two main elements needed to implement GE are: (1) a grammar to define the syntax of potential solutions, usually given in a Backus-Naur form, and (2) a fitness function to evaluate such solutions[89].

We chose a Python-coded GE implementation that is able to generate new scalarizing functions which can improve the performance of decomposition-based MOEAs. However, our implementation has the potential to be modified to also generate new elements that could potentially improve different types of MOEAs.

6.2 PREVIOUS RELATED WORK

There are several works involving the use of GP to automatically generate components that improve different Artificial Intelligence algorithms. For instance, a GP based system (EvoCK) [2] has been combined with a Machine Learning (ML) algorithm specialized in planning (Hamlet) [8], giving rise to Hamlet-EvoCK [3], which alleviates some of the handicaps of the original approach. One of these handicaps is that the ML technique can refine its behavior when being presented an appropriate set of examples. However, when the example-space is large, it deteriorates its performance due to the

[1] It is worth mentioning that indicator-based MOEAs based on $R2$ also use scalarizing functions [16].

computational time required. Then, in Hamlet-EvoCK, the authors propose to use the GP system to generate examples with certain characteristics, such as simplicity, while also evolving the population of examples based on a fitness function that measures the efficiency of such examples. Another example is the use of GP to automatically develop Artificial Neural Networks (ANNs), which can be done by evolution of the weights, by evolution of the architectures or by evolution of the learning rules. In [94] and [93] two proposals are made to achieve this automatic generation of ANNs requiring minimal to no human intervention and obtaining either average or better results than other automatic ANN generation techniques. More recently, a GP hyperheuristic was proposed in [129] to evolve scheduling heuristics for dynamic flexible job-scheduling problems. The generation of these heuristics is performed only with the selected features determined by the GP, since feature selection is a key part of this process.

In the context of automatic generation of scalarizing functions, to the best of our knowledge, the only related work is the one presented in [96]. In that work, a hybrid implementation is presented combining EllenGP [18], which is a GP implementation with local search used to generate scalarizing functions, and MOMBI-II [54], which is a MOEA adopted to assess the performance of the scalarizing functions generated. The search process of these new scalarizing functions involves coding the new functions into MOMBI-II and use it to solve a given MOP. In this work, we use a very similar methodology, but we extend the number of training MOPs solved in the search process to 2 instead of only 1, along with some other modifications such as the addition of a threshold to reduce the number of function evaluations invested in scalarizing functions which obtain bad results in a sample MOP. The increase in the number of training MOPs is proposed to favor the generation of scalarizing functions with a good performance in MOPs different from the one used in the training process.

6.3 Proposed implementation

We decided to use `PonyGE2`, which is a grammatical evolution implementation coded in Python [40]. This implementation handles the creation and reproduction of individuals until a certain stopping criterion is met. In all our experiments, this stopping criterion is a given maximum number of generations reached.

Therefore, our main contribution is the methodology used to create new scalarizing functions using `PonyGE2`. To do this, we adopted two components: a MOEA that uses scalarizing functions (MOMBI-II) and a performance indicator (hypervolume) to assess the quality of the functions generated. An overview of this process is shown in Fig 6.1.

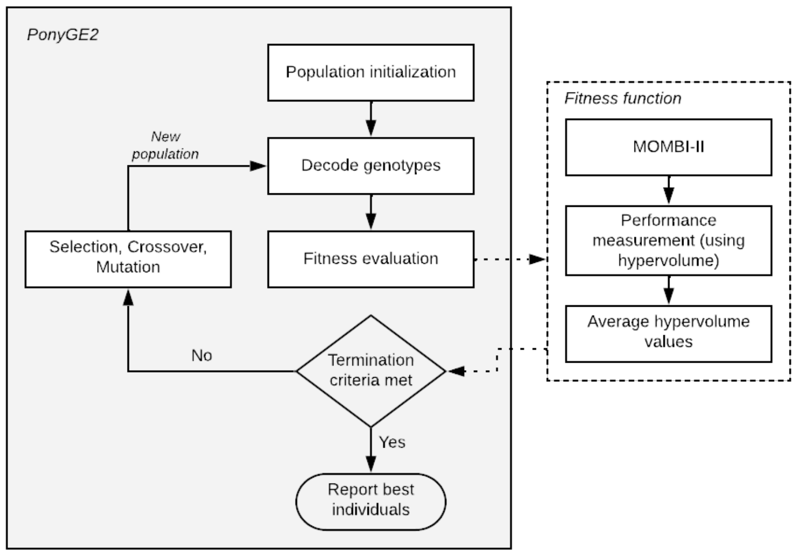


Figure 6.1: Diagram of grammatical evolution used to generate scalarizing functions.

MOMBI-II is a metaheuristic developed to solve many-objective optimization problems [54]. We adopted MOMBI-II in our implementation because it is a competitive algorithm which has shown a good performance when compared against state-of-the-art MOEAs in benchmark problems with up to 10 objectives. Additionally, it uses scalarizing functions, since it is based on the $R2$ indicator. By default, it uses the Achievement Scalarizing Function (ASF) [123] which is defined as follows:

$$ASF(\vec{f}(\vec{x}), \vec{w}) := \max_{i \in 1,\ldots,m} \left(\frac{f_i(\vec{x})}{w_i} \right) \tag{6.1}$$

where $\vec{f}(\vec{x})$ is the image of $\vec{x}$ in objective space and $\vec{w} \in \mathbb{R}^m$ is a weight vector. However, other scalarizing functions can be used, such as the Tchebycheff (TCH) scalarizing function [11], defined as follows:

$$TCH(\vec{f}(\vec{x}), \vec{w}) := \max_{i \in 1,\ldots,m} \left(w_i |f_i(\vec{x})| \right). \tag{6.2}$$

Another commonly used scalarizing function in decomposition-based MOEAs is the Penalty Boundary Intersection (PBI) [133], defined as follows:

$$PBI(\vec{f}(\vec{x}), \vec{w}) := \theta d_2 - d_1 \tag{6.3}$$

where $d_1 := \left| \vec{f}(\vec{x}) \cdot \vec{w}/\|\vec{w}\| \right|$, $d_2 := \left\| \vec{f}(\vec{x}) - d_1 \vec{w}/\|\vec{w}\| \right\|$ and θ is a penalty parameter, usually set by default as $\theta = 5$.

We implemented each of the new scalarizing functions generated by `PonyGE2`, replacing ASF. Then, we used this modified version of MOMBI-II to solve multiple MOPs. Finally, we used the hypervolume to assess the quality of the Pareto fronts obtained. The average hypervolume values obtained were used to set the fitness of the new function.

6.3.1 Fitness function

The fitness function that we propose to guide the search process of `PonyGE2` requires the following information:

- The parameters needed to define the training MOP. This includes the MOP, the number of objectives, the number of executions and the maximum number of function evaluations. For example, we could define that the training MOP is DTLZ4, with 2 objectives, considering 30 executions with a maximum number of 50,000 function evaluations each.
- The maximum hypervolume value for each of the MOPs defined, along with the reference point used to obtain such hypervolume value. This value is used to normalize hypervolume results. Following the previous example, the maximum hypervolume for DTLZ4 with 2 objectives can be set to 0.210 considering the reference point [1,1].

These parameters are enough for GE to generate new scalarizing functions. However, we included a step where we solve the training MOP with a smaller number of function evaluations and a lower number of executions to improve the search speed, because the use of the hypervolume can make the process computationally expensive. Thus, we need the following additional parameters:

- The parameters needed to define the sample MOP, which is the same as the training MOP, but with a lower number of function evaluations and a lower number of executions as well. This sample MOP is used to determine if the current function is worth performing more evaluations. Once again, following the previous example, we define the sample MOP to be DTLZ4, with 2 objectives, but considering 15 executions with a maximum number of 10,000 function evaluations.

- The maximum hypervolume value corresponding to the sample MOP. We need this since a lower number of function evaluations results in a different maximum hypervolume value.
- A hypervolume threshold that defines the percentage of the real maximum hypervolume that should be attained by the executions performed using the sample MOP in order to perform the full executions of the main training MOP. In case this threshold is not attained, no more function evaluations are spent in the current individual.

In Algorithm 5, we show the fitness function in detail, given the decoded phenotype of the individual (i.e., the scalarizing function) as well as the aforementioned parameters. We start by initializing variables (lines 1-2). Then, we use MOMBI-II with the new scalarizing function to solve the sample MOP. Then, we obtain the hypervolume of the corresponding Pareto front and we store this value. This process is repeated (lines 3-7) to obtain an average hypervolume (line 8). If the average hypervolume is below some percentage of the real hypervolume (defined by the threshold parameter) we normalize the average hypervolume (line 10) and then we penalize this value by dividing it by the number of MOPs (line 11). This penalty is simply intended to significantly decrease the fitness of individuals that do not reach the threshold in order to avoid the propagation of such individuals in the population. We chose to divide the current fitness by the number of MOPs as it is a simple way of decreasing the fitness while still keeping a remainder of the original fitness in case that most of the population contains bad individuals, which may occur in the first generations of the execution. In the event that the average hypervolume is greater than the desired threshold, we proceed to evaluate each of the desired MOPs using the new scalarizing function (lines 15-19). For each MOP, after averaging its hypervolume (line 20) and normalizing it (line 21), we cumulatively store this value to be the individual's final fitness (line 22).

When incorporating the new scalarizing functions in MOMBI-II we used the max operator in the following way. Given the decoded phenotype to be $SF(f_i(\vec{x}), w_i)$, where $\vec{f}(\vec{x})$ is the objective functions vector and $\vec{w}$ is the weight vector, we obtain the final Grammatical Evolution Scalarizing Function (GE_SF) as follows:

Algorithm 5: Fitness function used to generate scalarizing functions

Input : phenotype, $\vec{MOP} = \{mop_1, \ldots, mop_j\}$, $\vec{HV} = \{hv_1, \ldots, hv_j\}$, $\vec{N} = \{n_1, \ldots, n_j\}, sample\_mop, sample\_hv, sample\_n, threshold$;

Output: fitness;

$fitness \leftarrow 0$;
$avg\_hv \leftarrow 0$;
for $i \leftarrow 0$ **to** $sample\_n$ **do**
    PF_i ←MOMBI2($sample\_mop$, phenotype);
    $aux\_hv \leftarrow$ compute hypervolume value of PF_i;
    $avg\_hv \leftarrow avg\_hv + aux\_hv$;
end
$avg\_hv \leftarrow avg\_hv / sample\_n$;
if $avg\_hv < threshold * sample\_hv$ **then**
    $fitness \leftarrow avg\_hv / sample\_hv$;
    $fitness \leftarrow fitness / size(\vec{MOP})$;
else
    foreach $mop \in \vec{MOP}$ **do**
        $avg\_hv \leftarrow 0$;
        for $i \leftarrow 0$ **to** n_j **do**
            PF_i ←MOMBI2(mop_i, phenotype);
            $aux\_hv \leftarrow$ compute hypervolume value of PF_i;
            $avg\_hv \leftarrow avg\_hv + aux\_hv$;
        end
        $avg\_hv \leftarrow avg\_hv / n_i$;
        $avg\_hv \leftarrow avg\_hv / hv_i$;
        $fitness \leftarrow fitness + avg\_hv$;
    end
end
return $fitness$;

$$GE\_SF(\vec{f}(\vec{x}), \vec{w}) := \max_{i \in 1,\ldots,m} (SF(f_i(\vec{x}), w_i)). \tag{6.4}$$

6.3.2 GRAMMAR

The grammar used to generate the phenotypes consists of basic arithmetic operations and the square root, as shown below.

$$\begin{aligned} <e> \quad ::= \quad & <e> + <e> \mid <e> - <e> \mid <e> * <e> \mid <e> \,/\, <e> \\ & \mid \text{sqrt}(<e>) \mid <c><c> . <c><c> \mid f_i(\vec{x}) \mid w_i \end{aligned}$$

$$<c> \quad ::= \quad 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

We deliberately chose not to include more complex functions in the grammar, such as trigonometric functions, because some previous experiments showed that they generate poor performing scalarizing functions. In Table 6.1 we show 4 scalarizing functions generated using our implementation and the same grammar previously described, but with the addition of trigonometric functions ($\sin, \cos, \tan, \sin^{-1}, \cos^{-1}$ and $\tan^{-1}$). We performed a hypervolume comparison against ASF in 96 test instances, and we show the amount of problems improved (+), worsened (-) or with a similar performance ($\sim$).

Table 6.1: Scalarizing functions generated using trigonometric functions.

Training problem	$SF(f_i(\vec{x}), w_i)$	Hypervolume comparison		
		+	-	$\sim$
DTLZ1	$max\{\sin(\sum f_i(\vec{x}) + f_i(\vec{x})/w_i/0.21)\} - 0.27 * 34.44$	0	96	0
DTLZ2	$max\{\tan^{-1}(\tan^{-1}(\sqrt{0.41 * 14.77 + f_i(\vec{x})/w_i}))\} + 1.77 * \sum f_i(\vec{x})$	0	96	0
WFG1	$max\{\cos^{-1}(f_i(\vec{x})) + w_i/f_i(\vec{x})\} + 0.38 * \sum f_i(\vec{x})$	0	96	0
WFG3	$max\{\cos^{-1}(f_i(\vec{x})/w_i)\} * \sin(0.98 + \sin(\max f(\vec{x})) + 1/m \sum f_i(\vec{x})$	2	92	2

Out of the 96 test instances, three of the four generated functions obtained worse results than ASF in every single problem, while the remaining function was only able to improve two of the test instances. Thus, our hypothesis is that the use of trigonometric functions leads to an overspecification of the scalarizing functions in the training process, resulting in functions that have a really bad generalization when used to solve different problems. Hence, we did not include trigonometric functions in the following experiments.

6.4 EXPERIMENTAL WORK

Using the implementation previously described, we performed a series of executions adopting the benchmark problems from the Deb-Thiele-Laumanns-Zitzler (DTLZ) [35] and Walking-Fish-Group (WFG) [62] test suites.

We chose one test problem per execution. However, we solved the problem with 2, 3 and 5 objectives, in an attempt to create scalarizing functions which could have a good performance in different dimensions. We used DTLZ1-DTLZ7 and WFG1-WFG9. In tables 6.2 to 6.17 we show the parameters that we adopted for each execution. As can be seen, the sample MOP is exactly the same as the one adopted for training. However, in our experiments, we only performed 3 MOMBI-II executions. Then, if the function is worth more evaluations we proceed to execute MOMBI-II for each of the training MOPs for a total of 30 times.

Table 6.2: Parameters adopted to generate scalarizing functions using DTLZ1.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ1	2	2.434	[1.6,1.6]	30
		3	3.322	[1.5,1.5,1.5]	30
		5	7.565	[1.5,1.5,1.5,1.5,1.5]	30
Sample MOP		2	2.433	[1.6,1.6]	3

Table 6.3: Parameters adopted to generate scalarizing functions using DTLZ2.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ2	2	3.21	[2,2]	30
		3	7.353	[2,2,2]	30
		5	31.598	[2,2,2,2,2]	30
Sample MOP		2	3.21	[2,2]	3

Additionally, for each execution, we set the maximum number of generations to 40, and the threshold to 0.15. To validate the performance of the best scalarizing functions obtained in each execution, we used each of them to solve all 16 test problems (DTLZ1-DTLZ7 and WFG1-WFG9) considering 2, 3, 4, 5, 6 and 7 objectives. This generates a total of 96 test problems. We

Table 6.4: Parameters adopted to generate scalarizing functions using DTLZ3.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ3	2	3.619	[2.1,2.1]	30
		3	8.637	[2.1,2.1,2.1]	30
		5	40.484	[2.1,2.1,2.1,2.1,2.1]	30
Sample MOP		2	2.111	[2.1,2.1]	3

Table 6.5: Parameters adopted to generate scalarizing functions using DTLZ4.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ4	2	3.21	[2,2]	30
		3	7.37	[2,2,2]	30
		5	31.686	[2,2,2,2,2]	30
Sample MOP		2	3.21	[2,2]	3

Table 6.6: Parameters adopted to generate scalarizing functions using DTLZ5.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ5	2	3.21	[2,2]	30
		3	4.731	[1.8,1.8,2]	30
		5	104.98	[1.8,1.8,4.4,4.5,2]	30
Sample MOP		2	3.21	[2,2]	3

Table 6.7: Parameters adopted to generate scalarizing functions using DTLZ6.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ6	2	5.26	[2.5,2.5]	30
		3	6.522	[2,2,2.2]	30
		5	2997.455	[4,3.5,8.5,9.7,2.8]	30
Sample MOP		2	4.662	[2.5,2.5]	3

Table 6.8: Parameters adopted to generate scalarizing functions using DTLZ7.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ7	2	4.148	[1.9,5]	30
		3	11.878	[1.9,1.9,7]	30
		5	86.182	[1.9,1.9,2,2,11.8]	30
Sample MOP		2	4.146	[1.9,5]	3

Table 6.9: Parameters adopted to generate scalarizing functions using WFG1.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG1	2	5.308	[3.7,4.1]	30
		3	71.692	[3.5,5.6,6.5]	30
		5	81.055	[3.6,2.2,2.7,2.9,5.6]	30
Sample MOP		2	4.724	[3.7,4.1]	3

Table 6.10: Parameters adopted to generate scalarizing functions using WFG2.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG2	2	7.444	[2.4,5.1]	30
		3	80.645	[2.8,4.6,6.9]	30
		5	3492.889	[2.5,4,5.5,7.2,9.3]	30
Sample MOP		2	7.381	[2.4,5.1]	3

Table 6.11: Parameters adopted to generate scalarizing functions using WFG3.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG3	2	11.311	[3.1,5.1]	30
		3	59.045	[4,3.1,7.1]	30
		5	10353.854	[3.9,5.3,7.6,8.6,11.1]	30
Sample MOP		2	10.743	[3.1,5.1]	3

Table 6.12: Parameters adopted to generate scalarizing functions using WFG4.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG4	2	8.897	[3.1,5.1]	30
		3	83.189	[3.1,5.1,7.1]	30
		5	9901.827	[3.1,5.1,7.1,9.1,11.1]	30
Sample MOP		2	8.031	[3.1,5.1]	3

Table 6.13: Parameters adopted to generate scalarizing functions using WFG5.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG5	2	8.668	[3.1,5.1]	30
		3	79.515	[3.1,5.1,7.1]	30
		5	9437.534	[3.1,5.1,7.1,9.1,11.1]	30
Sample MOP		2	8.044	[3.1,5.1]	3

Table 6.14: Parameters adopted to generate scalarizing functions using WFG6.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG6	2	8.827	[3.1,5.1]	30
		3	80.543	[3.1,5.1,7.1]	30
		5	9506.954	[3.1,5.1,7.1,9.1,11.1]	30
Sample MOP		2	8.297	[3.1,5.1]	3

Table 6.15: Parameters adopted to generate scalarizing functions using WFG7.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG7	2	7.836	[3.1,5.3]	30
		3	83.5	[3.1,5.1,7.1]	30
		5	9989.461	[3.1,5.1,7.1,9.1,11.1]	30
Sample MOP		2	7.303	[3.1,5.3]	3

Table 6.16: Parameters adopted to generate scalarizing functions using WFG8.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG8	2	8.89	[3.3,5.2]	30
		3	83.425	[3.3,5.1,7.1]	30
		5	11786.065	[4,5.5,7.1,9.1,11.1]	30
Sample MOP		2	7.702	[3.3,5.2]	3

Table 6.17: Parameters adopted to generate scalarizing functions using WFG9.

	Problem	m	Maximum Hypervolume	Reference point	n
Training MOPs	WFG9	2	8.945	[3.1,5.1]	30
		3	79.929	[3.2,5.1,7.1]	30
		5	9195.075	[3.2,5.2,7.2,9.2,11.2]	30
Sample MOP		2	8.608	[3.1,5.1]	3

performed 30 independent runs and computed the hypervolume and the s-energy values of each Pareto front. S-energy is a performance indicator used to measure the uniformity of the distribution of a set of points. The lower the s-energy value the more uniform the distribution is [37]. Then, we used the Wilcoxon rank-sum test under the null hypothesis that the indicator results generated with each new scalarizing function come from the same distribution as the indicator results generated with ASF, considering a confidence level of 5%. We counted the number of problems where the null hypothesis is rejected and the new scalarizing function indicator average is greater than that of ASF under the column "+" of each comparison. Conversely, if the null hypothesis is rejected but the ASF indicator average is greater, we count these under the column "-". Finally, all problems in which the null hypothesis could not be rejected are counted in the column "$\sim$". We show these results in Table 6.18, including the execution time required to complete the 40 generations. These executions were performed on an Intel Core i7-8700 CPU, with 16 GB of RAM.

From these initial results, we can observe that many scalarizing functions (especifically those generated using DTLZ3, DTLZ6, WFG4-WFG9) include the $f_i(\vec{x})/w_i$ term from the original ASF. In fact, the scalarizing function generated with DTLZ3 was indeed, ASF. The best performing function

Table 6.18: Behavior of scalarizing functions generated with Grammatical Evolution. The comparison of the hypervolume and S-energy values show the number of test problems (a total of 96) in which the results improved(+), worsened(-) or were similar($\sim$) with respect to ASF.

Training problem	$SF(f_i(\vec{x}), w_i)$	Execution time (s)	Hypervolume comparison			S-energy comparison		
			+	-	$\sim$	+	-	$\sim$
DTLZ1	$f_i(\vec{x}) * w_i$	4455.67	17	59	20	21	48	27
DTLZ2	$w_i/w_i/w_i + w_i - \sqrt{w_i} + \sqrt{f_i(\vec{x})}$	3997.07	10	82	4	3	83	10
DTLZ3	$f_i(\vec{x})/w_i$	3949.98	0	0	96	0	0	96
DTLZ4	$\sqrt{w_i} * f_i(\vec{x})$	3403.82	9	81	6	34	49	13
DTLZ5	$\sqrt{f_i(\vec{x})} + w_i$	5831.65	15	72	9	31	53	12
DTLZ6	$f_i(\vec{x})/w_i + f_i(\vec{x})$	4667.40	44	23	29	37	25	34
DTLZ7	$f_i(\vec{x}) * w_i$	4357.98	17	59	20	21	48	27
WFG1	$\sqrt{w_i} * 84.75 + f_i(\vec{x}) * f_i(\vec{x})$	8338.57	3	92	1	1	91	4
WFG2	$\sqrt{f_i(\vec{x})} * w_i * w_i$	11567.13	19	72	5	10	81	5
WFG3	$w_i + 01.52 + \sqrt{f_i(\vec{x})}$	11135.62	15	72	9	30	50	13
WFG4	$f_i(\vec{x})/w_i - w_i - w_i$	12056.09	19	71	6	4	74	18
WFG5	$f_i(\vec{x})/w_i - w_i - w_i$	9838.63	19	71	6	4	74	18
WFG6	$f_i(\vec{x})/w_i - w_i * w_i$	11726.39	13	58	25	14	48	34
WFG7	$f_i(\vec{x})/w_i - f_i(\vec{x})$	9361.49	15	73	8	8	63	25
WFG8	$f_i(\vec{x})/w_i - w_i - w_i$	11093.16	19	71	6	4	74	18
WFG9	$f_i(\vec{x})/w_i - w_i/f_i(\vec{x})$	10509.27	11	7	78	11	13	72

is the one generated with DTLZ6, as it outperforms ASF in more problems (44) and it worsens them in 23 problems with respect to the hypervolume. A similar behavior is observed with respect to the s-energy values. However, we considered that it should be possible to obtain a better scalarizing function using more test problems and more generations.

Hence, we performed another round of experiments. This time we used two different MOPs per execution. In Table 6.19 we present the parameters of the problems used to generate another scalarizing function. The threshold parameter was set to 0.15 and we set the maximum number of generations to 80.

Table 6.19: Parameters adopted to generate a scalarizing function using DTLZ4 and WFG4.

	Problem	Objectives	Maximum Hypervolume	Reference point	n
Training MOPs	DTLZ4	2	0.210	[1,1]	30
	DTLZ4	3	0.420	[1,1,1]	30
	DTLZ4	5	0.7	[1,1,1,1,1]	30
	WFG4	2	2.100	[2.1,4.1]	30
	WFG4	3	21.500	[2.1,4.1,6.1]	30
	WFG4	5	2035	[2.1,4.1,6.1,8.1,10.1]	30
Sample MOP	DTLZ4	2	0.210	[1,1]	3

We performed three independent executions using these parameters, and here we present the best performing individual, which we call GE_SF1, and it is defined as follows:

$$GE\_SF1(\vec{f}(\vec{x}), \vec{w}) := \max_{i \in 1,\dots,m} \left(\frac{f_i(\vec{x})}{w_i} * f_i(\vec{x}) + \frac{f_i(\vec{x})}{w_i} - f_i(\vec{x}) \right). \quad (6.5)$$

Additionally, we performed three independent executions using I-DTLZ4 as the training problem, which is a variant belonging to the inverse DTLZ test problems [66]. The number of objectives used was 2, with a maximum hypervolume of 18.0, a reference point [1,1] and n set to 30. The resulting function, GE_SF2, is defined as follows.

$$GE\_SF2(\vec{f}(\vec{x}), \vec{w}) := \max_{i \in 1,\dots,m} \left(f_i(\vec{x}) + w_i + 1 - \frac{f_i(\vec{x})}{22.42} \right). \quad (6.6)$$

6.5 RESULTS

To assess the performance of GE_SF1 and GE_SF2 we coupled them with MOMBI-II and solved the benchmark problems DTLZ1-DTLZ7, WFG1-WFG9 and I-DTLZ1-I-DTLZ7 using 2, 3, 4, 5, 6 and 7 objectives. We performed 30 independent executions and measured the hypervolume values of the Pareto fronts generated. We repeated the same process with the original ASF, as well as with TCH and PBI.

In the following 3 tables, we show the best average hypervolume for each problem in **boldface**. Aditionally, we performed the Wilcoxon rank-sum test under the null hypothesis that the hypervolume results generated with one scalarizing function come from the same distribution as the hypervolume results generated with another scalarizing function, considering a confidence level of 5%. This test was performed with each pair of scalarizing functions, and we show the cases in which the null hypothesis was rejected and the current function's average hypervolume is greater under the "+" column of each function. Additionally, we indicate with gray cells all of the scalarizing function results in which the null hypothesis could not be rejected when comparing against the best result (marked in **boldface**).

In Tables 6.20 and 6.21 we show the average results of the hypervolume indicator in standard benchmark problems (DTLZ and WFG).

On the other hand, in Table 6.22 we show the average results of the hypervolume indicator in the inverse DTLZ problems.

From Tables 6.20 and 6.21 we counted the number of gray cells of each scalarizing function (i.e., the best performing functions for each problem)

Table 6.20: Hypervolume comparison using different scalarizing functions in problems with 2-4 objectives.

Problem	m	ASF(1)		PBI(2)		TCH(3)		GE_SF1(4)		GE_SF2(5)	
		HV	+	HV	+	HV	+	HV	+	HV	+
DTLZ1	2	2.327E-01	–	**2.337E-01**	1,5	2.337E-01	1,5	2.337E-01	1,5	2.336E-01	1
DTLZ2		4.182E-01	–	4.201E-01	1,3	4.201E-01	1	4.203E-01	1,2,3	**4.205E-01**	1,3,2,4
DTLZ3		4.172E-01	–	4.174E-01	–	4.188E-01	1,2	4.188E-01	1,2	**4.191E-01**	1,2
DTLZ4		2.988E-01	5	3.101E-01	1,3,5	3.101E-01	1,5	**3.103E-01**	1,3,2,5	2.692E-01	–
DTLZ5		4.182E-01	–	4.201E-01	1,3	4.201E-01	1	4.203E-01	1,2,3	**4.205E-01**	1,3,2,4
DTLZ6		1.274E+00	–	1.268E+00	–	1.261E+00	–	**1.291E+00**	3	1.265E+00	–
DTLZ7		6.433E-01	–	6.445E-01	1	6.447E-01	1,2,5	**6.448E-01**	1,3,2,5	6.446E-01	1,2
WFG1		2.252E+00	2,3,5	1.519E+00	–	1.589E+00	2	**2.363E+00**	1,3,2,5	1.546E+00	–
WFG2		4.906E+00	2,3,5	4.300E+00	–	4.315E+00	2	**4.929E+00**	1,3,2,5	4.323E+00	2,3
WFG3		4.529E+00	2,3,5	4.470E+00	–	4.491E+00	2	**4.543E+00**	1,3,2,5	4.503E+00	2,3
WFG4		2.272E+00	2,3,5	2.223E+00	–	2.235E+00	2,5	**2.281E+00**	1,3,2,5	2.223E+00	–
WFG5		1.973E+00	2,3,5	1.963E+00	–	1.966E+00	2	**1.980E+00**	1,3,2,5	1.965E+00	2
WFG6		2.024E+00	–	2.047E+00	–	2.062E+00	–	2.035E+00	–	**2.071E+00**	1,4
WFG7		2.686E+00	2,3,5	2.350E+00	–	2.371E+00	2,5	**2.707E+00**	1,3,2,5	2.343E+00	–
WFG8		3.795E+00	2,5	3.424E+00	–	3.743E+00	2	**3.848E+00**	1,2,5	3.747E+00	2
WFG9		1.997E+00	–	2.210E+00	1,4	**2.221E+00**	1,2,4	1.917E+00	–	2.216E+00	1,4
DTLZ1	3	3.805E+01	5	**3.845E+01**	1,4,3,5	3.844E+01	1,4,5	3.805E+01	5	3.614E+01	–
DTLZ2		7.394E-01	3,5	**7.494E-01**	1,4,3,5	7.099E-01	5	7.410E-01	1,3,5	6.533E-01	–
DTLZ3		2.387E+02	–	**2.408E+02**	1,4,3,5	2.408E+02	1,4,5	2.387E+02	–	2.075E+02	–
DTLZ4		8.609E-01	3,5	**8.704E-01**	1,4,3,5	8.111E-01	5	8.628E-01	1,3,5	7.519E-01	–
DTLZ5		2.188E+01	4	2.207E+01	1,4	**2.212E+01**	1,4,2,5	2.182E+01	–	2.211E+01	1,2,4
DTLZ6		2.079E+02	4	2.087E+02	1,4	2.094E+02	1,2,4	2.074E+02	–	**2.097E+02**	1,3,2,4
DTLZ7		1.449E+00	2,3,5	1.409E+00	3	1.389E+00	–	**1.450E+00**	2,3,5	1.344E+00	–
WFG1		3.462E+01	2,5	3.265E+01	5	3.504E+01	2,5	**3.504E+01**	2,5	2.692E+01	–
WFG2		4.455E+01	2,3,5	4.440E+01	3,5	4.333E+01	5	**4.486E+01**	1,3,2,5	4.088E+01	–
WFG3		4.824E+01	2,5	4.785E+01	–	**4.850E+01**	1,4,2,5	4.819E+01	2,5	4.788E+01	–
WFG4		2.401E+01	2,3,5	2.312E+01	3,5	2.190E+01	5	**2.409E+01**	1,3,2,5	2.063E+01	–
WFG5		2.192E+01	2,3,5	2.172E+01	3,5	1.987E+01	5	**2.203E+01**	1,3,2,5	1.917E+01	–
WFG6		2.444E+01	2,3,5	2.352E+01	3,5	2.218E+01	5	**2.450E+01**	2,3,5	2.088E+01	–
WFG7		2.426E+01	2,3,5	2.359E+01	3,5	2.206E+01	5	**2.434E+01**	1,3,2,5	2.075E+01	–
WFG8		2.566E+01	2,3,5	2.547E+01	3,5	2.461E+01	5	**2.591E+01**	1,3,2,5	2.335E+01	–
WFG9		2.519E+01	2,3,5	2.391E+01	3,5	2.313E+01	–	**2.555E+01**	1,3,2,5	2.237E+01	–
DTLZ1	4	1.015E+01	5	**1.034E+01**	1,4,3,5	1.033E+01	1,4,5	1.016E+01	5	8.635E+00	–
DTLZ2		1.014E+00	3,5	**1.032E+00**	1,4,3,5	8.857E-01	5	1.016E+00	1,3,5	7.018E-01	–
DTLZ3		1.017E+00	3,5	**1.018E+00**	1,3,5	8.739E-01	5	1.018E+00	3,5	7.006E-01	–
DTLZ4		1.021E+00	3,4,5	**1.032E+00**	1,4,3,5	8.642E-01	5	1.018E+00	3,5	7.113E-01	–
DTLZ5		4.380E+01	5	**4.420E+01**	1,4,3,5	4.419E+01	1,4,5	4.382E+01	5	4.350E+01	–
DTLZ6		6.929E+03	2,5	6.890E+03	–	6.925E+03	2,5	**6.934E+03**	1,3,2,5	6.909E+03	2
DTLZ7		1.872E+00	2,3,5	1.820E+00	3,5	1.607E+00	–	**1.875E+00**	1,3,2,5	1.701E+00	3
WFG1		2.510E+02	2,5	2.419E+02	5	**2.753E+02**	1,4,2,5	2.504E+02	2,5	2.038E+02	–
WFG2		3.868E+02	2,3,5	3.695E+02	3,5	3.623E+02	–	**3.873E+02**	2,3,5	3.616E+02	–
WFG3		3.774E+02	5	**3.886E+02**	1,4,3,5	3.836E+02	1,4,5	3.788E+02	5	3.729E+02	–
WFG4		**2.511E+02**	2,4,3,5	2.421E+02	3,5	1.623E+02	–	2.499E+02	2,3,5	1.612E+02	–
WFG5		2.331E+02	2,3,5	2.301E+02	3,5	1.473E+02	–	**2.334E+02**	2,3,5	1.530E+02	3
WFG6		**2.341E+02**	2,3,5	2.289E+02	3,5	1.367E+02	–	2.337E+02	2,3,5	1.517E+02	3
WFG7		2.538E+02	2,3,5	2.516E+02	3,5	1.706E+02	5	**2.547E+02**	1,3,2,5	1.636E+02	–
WFG8		2.944E+02	3,5	**3.130E+02**	1,4,3,5	2.142E+02	–	3.106E+02	1,3,5	2.205E+02	3
WFG9		2.228E+02	3,5	2.273E+02	1,3,5	1.461E+02	–	**2.294E+02**	1,3,5	1.580E+02	3

Table 6.21: Hypervolume comparison using different scalarizing functions in problems with 5-7 objectives.

Problem	m	ASF(1)		PBI(2)		TCH(3)		GE_SF1(4)		GE_SF2(5)	
		HV	+	HV	+	HV	+	HV	+	HV	+
DTLZ1	5	7.580E-02	3,4,5	**7.673E-02**	1,3,4,5	7.467E-02	5	7.575E-02	3,5	6.270E-02	–
DTLZ2		1.291E+00	3,5	**1.308E+00**	1,3,4,5	1.146E+00	5	1.294E+00	1,3,5	7.531E-01	–
DTLZ3		1.129E+04	4,5	1.129E+04	1,4,5	**1.130E+04**	1,2,4,5	1.128E+04	5	8.178E+03	–
DTLZ4		1.304E+00	3,5	1.308E+00	1,3,5	1.146E+00	5	**1.310E+00**	1,2,3,5	7.864E-01	–
DTLZ5		1.198E+02	–	**1.285E+02**	1,3,4,5	1.238E+02	1,4	1.222E+02	1	1.229E+02	1,4
DTLZ6		9.618E+04	2,3,5	9.590E+04	3	9.510E+04	–	**9.626E+04**	1,2,3,5	9.593E+04	3
DTLZ7		2.985E+00	3,5	2.994E+00	1,3,5	2.588E+00	5	**3.006E+00**	1,2,3,5	1.886E+00	–
WFG1		9.512E+02	2,4,5	9.023E+02	5	**1.022E+03**	1,2,4,5	9.355E+02	2,5	8.513E+02	–
WFG2		**3.952E+03**	2,3,5	3.842E+03	5	3.899E+03	2,4,5	3.878E+03	2,5	3.801E+03	–
WFG3		5.134E+03	3	5.141E+03	3,5	4.812E+03	–	**5.196E+03**	1,2,3,5	5.094E+03	3
WFG4		2.980E+03	2,3,5	2.864E+03	3,5	1.955E+03	5	**2.989E+03**	1,2,3,5	1.593E+03	–
WFG5		2.767E+03	2,3,5	2.749E+03	3,5	1.534E+03	–	**2.793E+03**	1,2,3,5	1.485E+03	–
WFG6		2.890E+03	2,3,5	2.850E+03	3,5	1.069E+03	–	**2.908E+03**	2,3,5	1.532E+03	3
WFG7		3.085E+03	2,3,5	3.059E+03	3,5	2.076E+03	5	**3.094E+03**	1,2,3,5	1.631E+03	–
WFG8		4.742E+03	3,5	**5.218E+03**	1,3,4,5	3.059E+03	–	4.861E+03	1,3,5	3.092E+03	–
WFG9		2.690E+03	3,5	**2.815E+03**	1,3,5	1.192E+03	–	2.751E+03	1,3,5	1.667E+03	3
DTLZ1	6	4.607E-02	3,5	**4.643E-02**	1,3,4,5	4.597E-02	5	4.604E-02	5	3.836E-02	–
DTLZ2		1.536E+00	3,5	**1.549E+00**	1,3,4,5	1.432E+00	5	1.539E+00	1,3,5	8.118E-01	–
DTLZ3		7.854E+03	4,5	**7.857E+03**	1,3,4,5	7.857E+03	1,4,5	7.854E+03	5	7.301E+03	–
DTLZ4		1.548E+00	3,5	1.551E+00	1,3,5	1.438E+00	5	**1.555E+00**	1,2,3,5	8.471E-01	–
DTLZ5		4.951E+01	2,3	4.444E+01	–	4.806E+01	2	5.031E+01	1,2,3	**5.037E+01**	1,2,3
DTLZ6		3.316E+05	2,3,5	3.297E+05	3	3.150E+05	–	**3.318E+05**	2,3,5	3.295E+05	3
DTLZ7		3.173E+00	3,4,5	**3.206E+00**	1,3,4,5	2.736E+00	5	3.145E+00	3,5	1.915E+00	–
WFG1		2.555E+02	2,5	2.414E+02	–	**2.683E+02**	1,2,4,5	2.561E+02	2,5	2.403E+02	–
WFG2		4.426E+04	2	4.334E+04	–	**4.580E+04**	1,2,5	4.519E+04	1,2,5	4.458E+04	1,2
WFG3		7.153E+04	2,3,5	6.801E+04	3	6.512E+04	–	**7.230E+04**	1,2,3,5	6.924E+04	3
WFG4		3.845E+04	2,3,5	3.681E+04	3,5	2.644E+04	5	**3.929E+04**	1,2,3,5	1.886E+04	–
WFG5		3.849E+04	2,3,5	3.236E+04	3,5	2.207E+04	5	**3.910E+04**	1,2,3,5	1.893E+04	–
WFG6		4.297E+04	2,3,5	4.218E+04	3,5	1.429E+04	–	**4.324E+04**	1,2,3,5	2.127E+04	3
WFG7		4.006E+04	2,3,5	3.265E+04	5	2.835E+04	5	**4.036E+04**	1,2,3,5	1.903E+04	–
WFG8		6.954E+04	3,5	**7.983E+04**	1,3,4,5	3.340E+04	–	7.044E+04	1,3,5	4.440E+04	3
WFG9		3.328E+04	3,5	**3.763E+04**	1,3,4,5	1.251E+04	–	3.454E+04	1,3,5	2.141E+04	3
DTLZ1	7	3.240E-02	3,4,5	**3.258E-02**	1,3,4,5	3.176E-02	5	3.227E-02	3,5	2.622E-02	–
DTLZ2		1.761E+00	3,4,5	**1.773E+00**	1,3,4,5	1.366E+00	5	1.751E+00	3,5	8.475E-01	–
DTLZ3		1.744E+00	3,5	1.754E+00	1,3,5	1.339E+00	5	**1.755E+00**	1,3,5	9.017E-01	–
DTLZ4		1.773E+00	3,4,5	**1.774E+00**	1,3,4,5	1.469E+00	5	1.771E+00	3,5	9.113E-01	–
DTLZ5		8.148E+00	2,3	6.169E+00	–	7.842E+00	2	8.228E+00	2,3	**8.532E+00**	1,2,3,4
DTLZ6		6.208E+05	3,5	**6.302E+05**	1,3,5	4.979E+05	–	6.281E+05	1,3,5	6.166E+05	3
DTLZ7		**3.360E+00**	3,4,5	3.197E+00	3,5	2.468E+00	5	3.307E+00	2,3,5	2.446E+00	–
WFG1		1.392E+02	2	1.237E+02	–	**1.675E+02**	1,2,4,5	1.429E+02	1,2,5	1.391E+02	2
WFG2		6.303E+05	2,5	6.070E+05	–	**6.350E+05**	2,5	6.223E+05	2,5	6.176E+05	2
WFG3		1.105E+06	2,3	1.075E+06	3	9.999E+05	–	1.094E+06	2,3	**1.109E+06**	2,3
WFG4		5.364E+05	3,5	5.426E+05	1,3,5	3.237E+05	5	**5.506E+05**	1,3,5	2.694E+05	–
WFG5		5.268E+05	2,3,5	3.514E+05	3,5	2.125E+05	–	**5.402E+05**	1,2,3,5	2.420E+05	3
WFG6		**6.158E+05**	2,3,5	5.490E+05	3,5	3.520E+05	5	6.150E+05	3,5	2.901E+05	–
WFG7		5.822E+05	2,3,5	5.345E+05	3,5	3.862E+05	5	**5.904E+05**	1,3,5	2.800E+05	–
WFG8		9.453E+05	3,5	**1.196E+06**	1,3,4,5	2.015E+05	–	9.908E+05	1,3,5	5.819E+05	3
WFG9		4.745E+05	3,5	**6.323E+05**	1,3,4,5	1.526E+05	–	5.271E+05	1,3,5	3.747E+05	3

Table 6.22: Hypervolume comparison using different scalarizing functions in inverted problems.

Problem	m	ASF(1)		PBI(2)		TCH(3)		GE_SF1(4)		GE_SF2(5)	
		HV	+	HV	+	HV	+	HV	+	HV	+
DTLZ1_MINUS	2	**1.503E+05**	2,5	1.50E+05	5	**1.503E+05**	2,5	1.503E+05	2,5	1.50E+05	–
DTLZ2_MINUS		9.56E+00	2	9.53E+00	–	9.56E+00	2	9.56E+00	1,2,3	**9.572E+00**	1,2,3,4
DTLZ3_MINUS		3.79E+06	2	3.78E+06	–	3.79E+06	2	3.79E+06	1,2,3	**3.794E+06**	1,2,3,4
DTLZ4_MINUS		9.56E+00	2	9.53E+00	–	9.56E+00	2	9.56E+00	1,2,3	**9.572E+00**	1,2,3,4
DTLZ5_MINUS		9.56E+00	2	9.53E+00	–	9.56E+00	2	9.56E+00	1,2,3	**9.572E+00**	1,2,3,4
DTLZ6_MINUS		9.44E+01	2	9.42E+01	–	9.44E+01	2	9.45E+01	1,2,3	**9.455E+01**	1,2,3,4
DTLZ7_MINUS		6.55E-01	2	4.28E-01	–	6.55E-01	2	6.55E-01	2	**6.548E-01**	1,2,3,4
DTLZ1_MINUS	3	1.76E+07	2,3	1.74E+07	–	1.76E+07	2	1.82E+07	1,2,3	**2.176E+07**	1,2,3,4
DTLZ2_MINUS		1.88E+01	–	1.91E+01	1,3,4	1.89E+01	1	1.91E+01	1,3	**2.025E+01**	1,2,3,4
DTLZ3_MINUS		4.70E+09	–	4.76E+09	1,3	4.71E+09	1	4.76E+09	1,2,3	**5.053E+09**	1,2,3,4
DTLZ4_MINUS		1.88E+01	–	1.91E+01	1,3	1.88E+01	1	1.91E+01	1,3	**2.025E+01**	1,2,3,4
DTLZ5_MINUS		1.95E+01	2,3	1.95E+01	3	1.93E+01	–	1.96E+01	1,2,3	**2.024E+01**	1,2,3,4
DTLZ6_MINUS		5.90E+02	–	5.96E+02	1,3	5.92E+02	1	5.98E+02	1,2,3	**6.286E+02**	1,2,3,4
DTLZ7_MINUS		4.44E-01	2,3	1.80E-01	–	4.01E-01	2	4.45E-01	1,2,3	**4.453E-01**	1,2,3,4
DTLZ1_MINUS	4	5.00E+08	2,3	4.17E+08	3	1.25E+08	–	6.25E+08	1,2,3	**1.345E+09**	1,2,3,4
DTLZ2_MINUS		1.69E+01	–	2.20E+01	1,3,4	2.14E+01	1,4	2.05E+01	1	**3.223E+01**	1,2,3,4
DTLZ3_MINUS		2.67E+12	–	3.42E+12	1,3,4	3.39E+12	1,4	3.25E+12	1	**5.069E+12**	1,2,3,4
DTLZ4_MINUS		1.67E+01	–	2.18E+01	1,3,4	2.13E+01	1,4	2.04E+01	1	**3.224E+01**	1,2,3,4
DTLZ5_MINUS		2.31E+01	–	2.75E+01	1,3,4	2.60E+01	1,4	2.42E+01	1	**3.256E+01**	1,2,3,4
DTLZ6_MINUS		1.88E+03	–	2.32E+03	1,3,4	2.21E+03	1,4	2.13E+03	1	**3.145E+03**	1,2,3,4
DTLZ7_MINUS		**1.905E+00**	2,3,4,5	2.00E-01	–	1.40E+00	2	1.86E+00	2,3,5	1.85E+00	2,3
DTLZ1_MINUS	5	7.49E+08	2	6.10E+07	–	1.74E+10	1,2,4	9.61E+08	2	**4.107E+10**	1,2,3,4
DTLZ2_MINUS		1.58E+00	–	2.22E+01	1,3,4	2.00E+01	1,4	1.28E+01	1	**4.467E+01**	1,2,3,4
DTLZ3_MINUS		3.40E+14	–	2.24E+15	1,3,4	2.04E+15	1,4	1.32E+15	1	**4.424E+15**	1,2,3,4
DTLZ4_MINUS		1.66E+00	–	2.20E+01	1,3,4	1.88E+01	1,4	1.26E+01	1	**4.460E+01**	1,2,3,4
DTLZ5_MINUS		1.81E+01	–	3.55E+01	1,3,4	3.34E+01	1,4	2.06E+01	1	**4.563E+01**	1,2,3,4
DTLZ6_MINUS		2.26E+03	–	8.08E+03	1,3,4	7.36E+03	1,4	4.45E+03	1	**1.374E+04**	1,2,3,4
DTLZ7_MINUS		2.04E+00	2,3	1.00E-01	–	1.58E+00	2	**2.188E+00**	1,2,3	1.911E+00	2,3
DTLZ1_MINUS	6	2.73E+10	2	8.05E+08	–	**1.874E+11**	1,2,4,5	3.39E+10	2	1.13E+11	1,2,4
DTLZ2_MINUS		2.70E+00	–	2.58E+01	1,3,4	4.09E+00	1,4	3.01E+00	1	**4.872E+01**	1,2,3,4
DTLZ3_MINUS		2.08E+17	–	1.67E+18	1,3,4	4.62E+17	1,4	2.62E+17	1	**3.048E+18**	1,2,3,4
DTLZ4_MINUS		1.04E+00	–	2.46E+01	1,3,4	3.10E+00	1	1.68E+00	1	**4.834E+01**	1,2,3,4
DTLZ5_MINUS		2.06E+01	–	4.02E+01	1,3,4	2.96E+01	1,4	2.09E+01	1	**5.094E+01**	1,2,3,4
DTLZ6_MINUS		8.08E+03	–	2.93E+04	1,3,4	9.66E+03	1,4	8.47E+03	1	**4.756E+04**	1,2,3,4
DTLZ7_MINUS		**3.968E+00**	2,3,5	1.60E+00	–	3.48E+00	2	3.785E+00	2,3,5	3.19E+00	2
DTLZ1_MINUS	7	**3.333E+12**	2,3,5	1.24E+11	3,5	4.88E+09	5	3.321E+12	2,3,5	1.99E+09	–
DTLZ2_MINUS		3.24E+00	3	2.29E+01	1,3,4	2.30E-02	–	2.96E+00	3	**3.890E+01**	1,2,3,4
DTLZ3_MINUS		1.50E+20	3	9.72E+20	1,3,4	2.97E+18	–	1.60E+20	3	**1.553E+21**	1,2,3,4
DTLZ4_MINUS		1.17E+00	3	2.15E+01	1,3,4	3.53E-03	–	1.27E+00	3	**3.810E+01**	1,2,3,4
DTLZ5_MINUS		2.24E+01	3	2.98E+01	1,3,4	1.33E+01	–	2.24E+01	3	**4.337E+01**	1,2,3,4
DTLZ6_MINUS		2.64E+04	3	7.29E+04	1,3,4	2.52E+03	–	2.69E+04	3	**1.187E+05**	1,2,3,4
DTLZ7_MINUS		3.745E+00	2,5	1.60E+00	–	**3.919E+00**	2,5	3.457E+00	2,5	2.38E+00	2

and grouped them by number of objectives. We show these results in Table 6.23. Here, we can observe that GE_SF1 is the best performing scalarizing function, having the best results in 56 out of 96 test instances, followed by PBI, which obtained the best results in 36 cases. This is the behavior we were looking for by combining DTLZ4 and WFG4, which is to improve the performance obtained with ASF. However, not only GE_SF1 outperformed ASF, but it also outperformed the other scalarizing functions adopted in this comparison. Conversely, GE_SF2 is the scalarizing function with the worst performance, as it only obtained the best results in 10 test instances, which was also an expected behavior since it was trained using I-DTLZ4. Nonetheless it is interesting to see that it was able to obtain top results in DTLZ5 with 2, 6 and 7 objectives as well as DTLZ6 with 2 and 3 objectives, even though it was not designed to successfully solve standard MOPs.

Even though GE_SF1 attained the largest number of problems improved, it is evident that not even this function is able to outperform the others in every test problem, since it only obtained top results in 58.33% of the test instances. This is also an expected behavior since we are dealing with many different types of problems/number of objectives/geometries, which makes it difficult for a single scalarizing function to improve results in every possible setup.

Table 6.23: Number of DTLZ and WFG problems in which each scalarizing function obtained the best performance (or a statistically similar performance to the best one) using the hypervolume indicator.

Test problems	m	Total problems	ASF	PBI	TCH	GE_SF1	GE_SF2
DTLZ1-7, WFG1-9	2	16	1	3	5	**12**	6
	3	16	3	4	3	**9**	1
	4	16	4	**8**	1	**8**	0
	5	16	2	5	2	**10**	0
	6	16	1	6	2	**9**	1
	7	16	4	**10**	2	8	2
		Total	15	36	15	**56**	10

From Table 6.22 we counted the number of gray cells of each scalarizing function and we show these results in Table 6.24. These results correspond to the validation using inverse DTLZ problems. Here, we can observe that GE_SF2, which was generated specifically using an inverse test MOP in the search process of `PonyGE2`, obtains the largest number of problems improved. Overall, GE_SF2 is able to obtain top results in 36 out of 42 test instances.

Finally, we performed two additional comparisons against MOEA/D [133]

Table 6.24: Number of I-DTLZ problems in which each scalarizing function obtained the best performance (or a statistically similar performance to the best one) using the hypervolume indicator.

Test problems	m	Total problems	ASF	PBI	TCH	GE_SF1	GE_SF2
I-DTLZ1-7	2	7	1	0	1	1	**6**
	3	7	0	0	0	0	**7**
	4	7	1	0	0	0	**6**
	5	7	0	0	0	1	**7**
	6	7	1	0	1	1	**5**
	7	7	2	0	1	2	**5**
		Total	5	0	3	5	**36**

and NSGA-III, as well as the default version of MOMBI-II with ASF. In the first comparison, shown in Table 6.25 we compare these three MOEAs against MOMBI-II with GE_SF1, since it was the best performing scalarizing function for standard MOPs. In this comparison, MOMBI-II with GE_SF1 obtained the largest number of top results, with 54 out of 96, followed by NSGA-III with 40 top results.

Table 6.25: Comparison of the number of problems improved, using the hypervolume indicator, with different MOEAs in the DTLZ and the WFG test problems.

Test problems	m	Total problems	MOEA/D	NSGA-III	MOMBI-II$_{ASF}$	MOMBI-II$_{GESF1}$
DTLZ1-7, WFG1-9	2	16	4	5	0	**14**
	3	16	3	8	3	**10**
	4	16	2	**7**	6	**7**
	5	16	3	5	2	**9**
	6	16	1	5	2	**9**
	7	16	3	**10**	4	5
		Total	16	40	17	**54**

In the second comparison against other MOEAs, we compared the performance of GE_SF2, since it was the best performing scalarizing function for inverted MOPs. These results are shown in Table 6.26. Similar to the scalarizing functions comparison from Table 6.22, MOMBI-II with GE_SF2 obtained the best results across all dimensions, and significantly improved the results obtained by all the other MOEAs.

In Figure 6.2 we show the contour lines of the 5 scalarizing functions used in our comparisons, with 5 different weight vectors. All these plots are shown in the interval [0,1] for two objectives, for an easier comparison. We can observe that the contour lines from ASF and GE_SF1 are really similar.

Table 6.26: Comparison of the number of problems improved, using the hypervolume indicator, with different MOEAs in the I-DTLZ problems.

Test problems	m	Total problems	MOEA/D	NSGA-III	MOMBI-II$_{ASF}$	MOMBI-II$_{GE_SF2}$
I-DTLZ1-7	2	7	1	2	1	**5**
	3	7	0	0	1	**6**
	4	7	0	0	1	**6**
	5	7	1	0	0	**7**
	6	7	0	1	1	**5**
	7	7	0	1	2	**5**
		Total	2	4	6	**34**

In an analogous way, the contour lines of TCH and GE_SF2 also share some similarities. However, GE_SF1 exhibits a sharper steepness with respect to ASF, since for all 5 weight vectors it reached a smaller value in the lower values of the plot. Conversely, GE_SF2 exhibits a lower steepness when compared to TCH, since this time TCH reaches the smallest values with all the 5 weight vectors adopted.

Finally, we show a comparison of the Pareto fronts obtained in each inverted test problem with 2 objectives in Fig. 6.3 and in inverted problems with 3 objectives in Fig. 6.4. The fronts shown in these figures correspond to the results at the median of the hypervolume values obtained from 30 independent executions. In all cases, we can observe that the Pareto fronts generated with GE_SF2 have a better distribution than those generated with ASF.

It is particularly interesting to notice that there was no weight vector adaptation mechanism used in the inverse problems, meaning that the same weight vectors used for the standard DTLZ and WFG test problems are used to solve the inverse DTLZ problems. However, from the plots presented in Figures 6.3 and 6.4 it is noticeable that GE_SF2 is replacing the role of a weight vector adaptation, which would typically be an easier way to improve the obtained results in inverse problems.

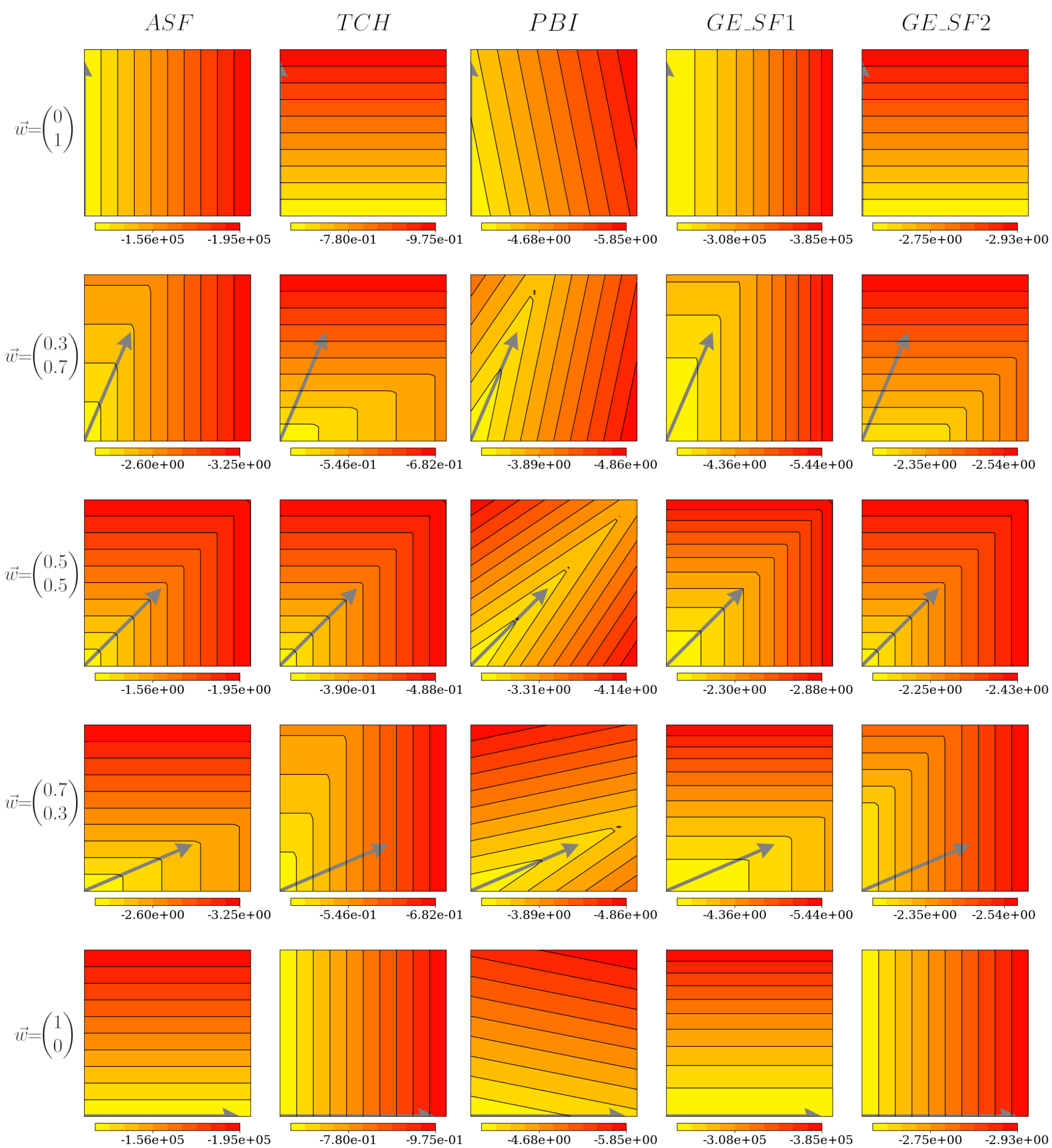
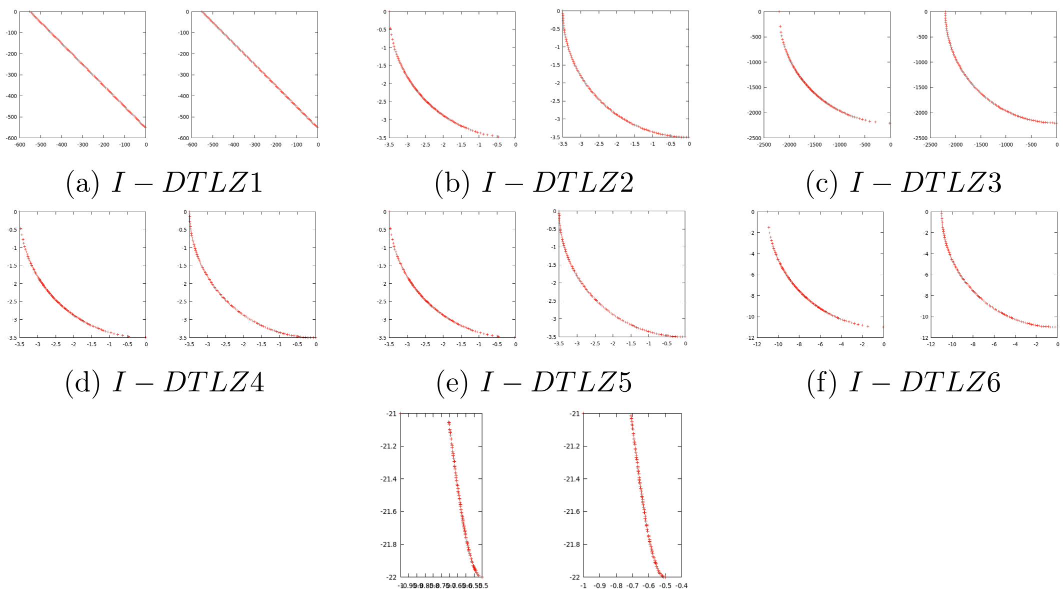


Figure 6.2: Contour lines for ASF, TCH, PBI, GE_SF1 and GE_SF2 with different weight vectors $\vec{w}$.



(a) $I - DTLZ1$ (b) $I - DTLZ2$ (c) $I - DTLZ3$

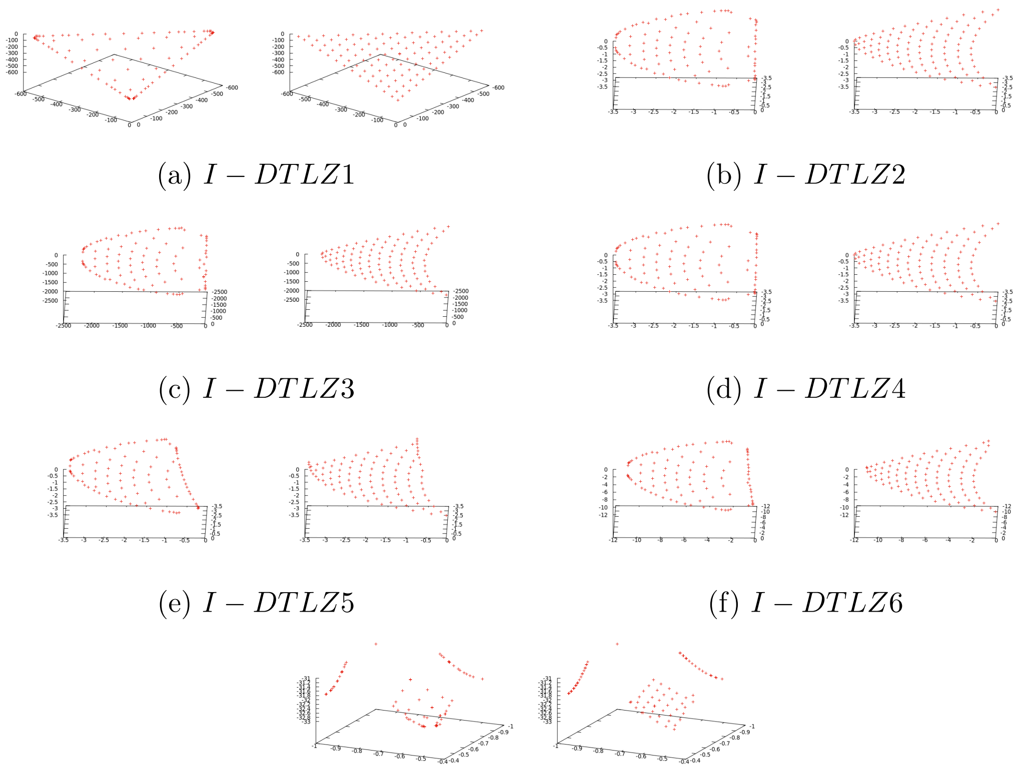
(d) $I - DTLZ4$ (e) $I - DTLZ5$ (f) $I - DTLZ6$

(g) $I - DTLZ7$

Figure 6.3: Pareto fronts obtained using ASF (left) and GE_SF2 (right) in DTLZ inverse problems with 2 objectives.

6.6 Summary

In this chapter we have presented a methodology that allows the automatic generation of scalarizing functions using grammatical evolution. Using this methodology we have generated two different scalarizing functions (GE_SF1 and GE_SF2). The former was obtained using DTLZ4 and WFG4 and the latter was obtained using I-DTLZ4. We have provided experimental evidence that shows that these two functions outperform ASF, TCH and PBI in the test problems considered, as well as other MOEAs such as MOEA/D and NSGA-III. GE_SF1 obtained the largest number of wins in the comparisons using the standard DTLZ and WFG problems, whereas GE_SF2 obtained the largest number of wins in the comparisons using the inverted DTLZ problems. Since our methodology employs hypervolume calculations, it can become computationally expensive. In the worst case, which occurred when using WFG4 as the training problem, it took 12,056 seconds to complete 40 generations, averaging close to 300 seconds per generation. However, it is important to notice that this is the cost of generating the scalarizing function. Once we have obtained it, using such a scalarizing function has a similar computational cost to that of ASF or TCH.



(a) $I - DTLZ1$

(b) $I - DTLZ2$

(c) $I - DTLZ3$

(d) $I - DTLZ4$

(e) $I - DTLZ5$

(f) $I - DTLZ6$

(g) $I - DTLZ7$

Figure 6.4: Pareto fronts obtained using ASF (left) and GE_SF2 (right) in the DTLZ inverse problems with 3 objectives.

Also, in the case of standard benchmark problems, the percentage of problems with top results obtained with GE_SF1 was under 60%, which evidences that even the best performing scalarizing function from our comparisons is not able to generalize the improvements in all benchmark problems. Thus, we can conjecture that in order to achieve improvements in more test instances, an ensemble of multiple complementary scalarizing functions could be used, and possibly couple it to a weight vector adaptation mechanism.

Finally, the functions found using GE were trained using a specific problem or specific data, but there is evidence that they can generalize their good performance to problems/data that were not considered in the training. Based on this evidence, we believe that this implementation can be used to obtain better results in a wide variety of problems.

7

Use of Grammatical Evolution to obtain hypervolume approximations

7.1 Introduction

Indicator-based MOEAs adopt performance indicators to measure the quality of a given individual with respect to the rest of the population and use this information to assign its fitness value. The individual with the worst contribution is usually deleted from the population and replaced in the next generation [141]. One of the most widely used indicators in these algorithms is the hypervolume. This is due to the fact that the hypervolume is Pareto compliant, which means that it preserves the order imposed by the Pareto dominance relation [38]. Additionally, it provides both convergence and (to some extent) diversity information, since a set with a better coverage of the Pareto front will have a better hypervolume value. Nonetheless, the main drawback of the hypervolume is that its computational cost increases exponentially with the number of objectives. This has motivated a lot of research aiming to find more efficient ways of calculating it [53, 122] or ways of approximating it [5, 36, 110] to reduce its computational cost.

In this chapter, we present another application of the Grammatical Evolution (GE) implementation presented in the previous chapter. This additional application allows the generation of new hypervolume approximations, which decrease the computational time required while still having a good performance.

7.2 Hypervolume approximations

7.2.1 Monte Carlo approximation

One of the most popular methods to approximate the hypervolume value of a non-dominated set is the Monte Carlo approach [5]. Given a data set $X = \{x_1, \ldots, x_n\}$, $x_i \in \mathbb{R}^m$ we can approximate its hypervolume by sampling a given number of points in the region delimited by X. Then, for each sample point we determine if it is dominated by any x_i, counting the dominated sample points in variable $hits$. Then, the Monte Carlo approximation is obtained using the following expression:

$$MC(X) = \lambda(X) * \frac{hits}{\text{sample size}}. \tag{7.1}$$

The quality of the approximation depends on the sample size: the larger the sample is, the better the approximation results. However, the computational cost increases proportionally to this value.

7.2.2 R2 hypervolume approximation

Another recent proposal to approximate both the hypervolume value of a non-dominated set and the individual hypervolume contribution is based on the use of the R2 indicator [108]. This is a linear-based approach, defined as follows:

$$R_2^{HV}(X, \Lambda, \vec{r}, m) = \frac{1}{|\Lambda|} \sum_{\vec{\lambda} \in \Lambda} \max_{\vec{x} \in X} \left(g^{mtch}(\vec{x}|\vec{\lambda}, \vec{r}) \right)^m \tag{7.2}$$

where X is a set of non-dominated solutions $\vec{x} \in \mathbb{R}^m$, Λ is a set of direction vectors, each direction vector $\vec{\lambda} = \{\lambda_1, \ldots, \lambda_m\} \in \Lambda$ satisfies $\|\lambda\|_2 = 1$ and $\lambda_i \geq 0$, $i = 1, \ldots, m$, $\vec{r} = \{r, \ldots, r\}$ is the reference point and m is the dimensionality of the space. The function g^{mtch} is defined as follows.

$$g^{mtch}(\vec{x}|\vec{\lambda}, \vec{r}) = \min_{j \in 1, \ldots, m} \left(\frac{|r_j - a_j|}{\lambda_j} \right). \tag{7.3}$$

7.2.3 GP generated approximations

A GP-based methodology was recently proposed to approximate, in an efficient way, the hypervolume value and the individual hypervolume contribution in 3, 4 and 5 dimensional spaces [100]. For each data set $X \in \mathbb{R}^{n \times m}$, there are some statistical features that are obtained in order to use these approximations. These features are shown in Table 7.1. All of them are obtained for each of the m objectives in the data.

Table 7.1: Statistical features extracted from data to obtain its hypervolume approximation.

Notation	Statistical feature
$\vec{\mu}$	mean
$\vec{\sigma}$	standard deviation
$\vec{Q1}$	1st quartile
$\vec{Q2}$	2nd quartile
$\vec{Q3}$	3rd quartile
$\vec{\kappa}$	kurtosis
$\vec{\lambda}$	skewness

The hypervolume approximation functions for 3, 4 and 5 dimensions are defined as shown in eqns (7.4), (7.5) and (7.6) respectively. These equations were extracted from the source code provided in [100].

$$\begin{aligned} M^3_{4,6}(X) = & \frac{6236736683876353 * \vec{\mu}_1 * \vec{\mu}_2 * \vec{Q1}_1}{4503599627370496} \\ & - \frac{3173082762593237 * \vec{\mu}_2}{2251799813685248} - \frac{1771343023238655 * \vec{\mu}_3}{2251799813685248} \\ & - \frac{5290754323525305 * \vec{\sigma}_1}{4503599627370496} - \frac{8680371570911475 * \vec{Q1}_1}{36028797018963968} \\ & - \frac{2893457190303825 * |\vec{\gamma}_3|}{36028797018963968} - \frac{5906898887207671 * log(\vec{\kappa}_1)}{36028797018963968} \\ & - \frac{6829965597182259 * log(\vec{\sigma}_3)}{9007199254740992 * log(10)} \\ & - \frac{2294397315973779 * log(\vec{\kappa}_2 + \vec{\gamma}_2 + \vec{Q2}_2 * \vec{\gamma}_1)}{18014398509481984 * log(10)} \\ & - \frac{2893457190303825 * \vec{Q2}_3 * \vec{\gamma}_1}{36028797018963968} - \frac{6826873663546647 * \vec{Q2}_3 * \vec{\gamma}_2}{36028797018963968} \\ & - \frac{6066879653575323 * \vec{\mu}_1}{4503599627370496} + \frac{1215041356731309 * \vec{Q2}_3 * \vec{Q1}_1 * \vec{\gamma}_2}{4503599627370496} \\ & + \frac{5647080252797291}{2251799813685248} \end{aligned} \quad (7.4)$$

$$
\begin{aligned}
M^4_{5,6}(X) = & \frac{3818342324243663 * (\vec{\sigma}_3^9)^{1/2}}{17592186044416} - \frac{8367714107802115 * \vec{Q1}_4}{9007199254740992} \\
& - \frac{4684844483679697 * \sin(\sin(\vec{\mu}_2 * \vec{\sigma}_4))}{1125899906842624} \\
& - \frac{4831284340333437 * \sin((\vec{\sigma}_2^2 * \vec{\gamma}_2)/\cos(\vec{Q1}_3))}{4503599627370496} \\
& - \frac{3387877962052739 * exp(exp(\vec{\sigma}_3^3))}{70368744177664} - \frac{8649112683782109 * \vec{Q1}_3}{18014398509481984} \\
& + \frac{2421873864284579}{35184372088832 * \cos(\vec{\sigma}_3)} + \frac{188607530293811 * 1/\tan(\vec{\sigma}_4)^{1/4}}{562949953421312} \\
& - \frac{12795601803451 * \vec{\mu}_1 * \vec{\sigma}_3}{8796093022208} \\
& - \frac{3714273495887619 * \vec{Q1}_2 * \cos(\vec{Q1}_2) * \sin(\vec{\gamma}_4)}{18014398509481984} \\
& + \frac{8842052222550475}{140737488355328}
\end{aligned}
\tag{7.5}
$$

$$
\begin{aligned}
M_{1,5}^{5}(X) = {} & \frac{226757085449091 * \vec{\mu}_4}{562949953421312} - \frac{49212418854775 * \vec{\mu}_1}{140737488355328} \\
& + \frac{316327101333597 * \vec{Q2}_2}{562949953421312} + \frac{4208801459950455 * \vec{Q2}_3}{18014398509481984} \\
& + \frac{4208801459950455 * \vec{\sigma}_2}{9007199254740992} - \frac{4420306567464985 * \vec{\sigma}_3}{281474976710656} \\
& + \frac{1059206653244855 * \vec{Q1}_5}{18014398509481984} + \frac{50672178223625 * \vec{Q3}_1}{70368744177664} \\
& + \frac{8274937251716701 * \sin^{-1}(\sin^{-1}(\vec{Q2}_2 * \vec{\sigma}_3))}{2251799813685248} \\
& - \frac{8229011520467723 * \sin^{-1}(\vec{\sigma}_3^2)}{562949953421312} \\
& + \frac{2896651095433129 * \sin^{-1}(\tan(\vec{\sigma}_3))}{281474976710656} \\
& + \frac{501121421306637 * \cos(\vec{Q2}_2 + \vec{Q2}_3 - \vec{Q1}_5 + \vec{Q3}_4 + \vec{Q3}_5 + log(\vec{\kappa}_3))}{4503599627370496} \\
& + \frac{5140569679074477 * \cos(\vec{\mu}_3 + \vec{\mu}_4 + \vec{Q1}_5 + \vec{Q3}_1 + |\vec{\sigma}_2|)}{4503599627370496} \\
& - \frac{49212418854775 * \tan(\vec{Q2}_2)}{281474976710656} + \frac{2896651095433129 * \tan(\vec{\sigma}_3)}{281474976710656} \\
& - \frac{49212418854775}{281474976710656 * \cos(\vec{Q3}_5)} + \frac{4208801459950455 * \vec{Q3}_1 * \vec{Q3}_4}{18014398509481984} \\
& - \frac{49212418854775 * \vec{Q1}_2 * \vec{\gamma}_5}{281474976710656} \\
& - 5.2299056961053288716811948688701 * \vec{Q2}_2 * \sin(\vec{\sigma}_5) \\
& + 0.27075243546777693026683664356824.
\end{aligned}
\tag{7.6}
$$

These approximations were obtained using a similar methodology to the one we propose in our work. However, instead of obtaining information from the population as a whole, and building approximation functions with such information, we use point-wise information, meaning that we evaluate the approximation functions iteratively using each of the points in the data. This increases the computational time $\mathcal{O}(n)$, where n is the number of points in the data, but also improves the quality of the approximation.

7.3 Proposed implementation

We propose to generate hypervolume approximations by averaging a cumulative-sum of function values. We used the same implementation described in the previous chapter (`PonyGE2` [40]). However, we made some modifications described in this section in order to generate hypervolume approximations instead of scalarizing functions.

The overall process is depicted in Fig 7.1. First, we obtain the approximation of the hypervolume of some training data using the new approximation function. Then, we obtain the mean squared error using this hypervolume approximation and the real hypervolume and assign the individual's fitness accordingly.

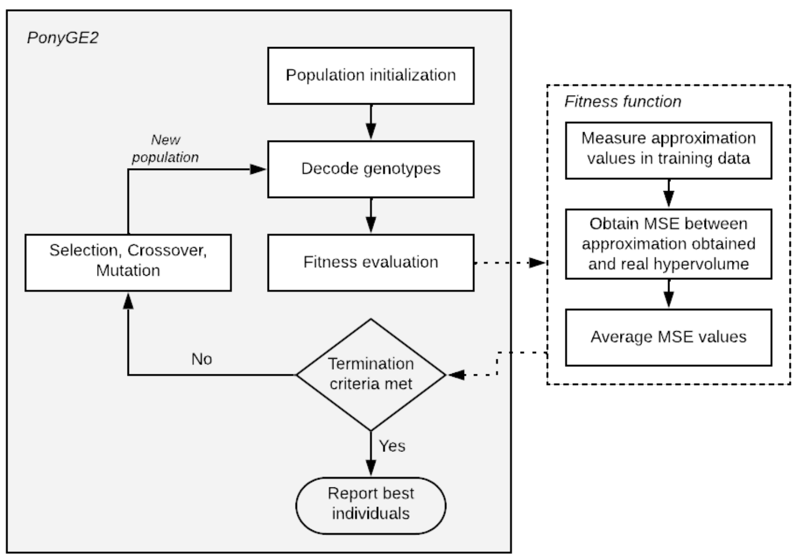


Figure 7.1: Diagram of grammatical evolution used to generate hypervolume approximations.

7.3.1 Fitness function

In Algorithm 6 we present the outline of the corresponding fitness function. In order to evaluate the fitness of each individual we require its decoded phenotype, a training set $\vec{X} = \{X_1, \ldots, X_n\}$, where each X_i contains a set of non-dominated points, and the corresponding set of real hypervolume values $\vec{I_H} = \{I_1, \ldots, I_n\}$, where I_i is the real hypervolume of the file X_i. Then, we iterate through each file X_i to evaluate the phenotype using vector-wise information. This means that for a given file of size n we will evaluate the phenotype n times, storing the result in a cumulative-sum variable. Then, we average this value with respect to n. This process is repeated with every training set, and we use the average mean squared error (MSE) of each approximation to assign the final fitness.

Algorithm 6: Fitness function used to generate hypervolume approximations using our proposal

Input : phenotype, $\vec{X} = \{X_1, \ldots, X_n\}$, $\vec{I_H} = \{I_1, \ldots, I_n\}$;
Output: fitness;

$mse \leftarrow 0$;
for $i \leftarrow 0$ **to** n **do**
    $approximation \leftarrow 0$;
    foreach $\vec{x} \in X_i$ **do**
        $approximation \leftarrow approximation + \texttt{evaluate}(\text{phenotype}, \vec{x})$;
    end
    $approximation \leftarrow approximation / \texttt{size}(X_i)$;
    $mse \leftarrow mse + (I_i - approximation)^2$;
end
$fitness \leftarrow mse / \texttt{size}(\vec{X})$;
return $fitness$;

7.3.2 GRAMMAR

We adopted two different grammars to generate hypervolume approximations. The first one is shown below, where $\vec{x} \in \mathbb{R}^m$ corresponds to the m-dimensional non-dominated points in a given training set X of size n, and $\vec{sum} \in \mathbb{R}^m$ stores the sum of all objectives of $\vec{x}$ considering $sum_j = \sum_{i=1}^{n} x_{i,j}$. In contrast with the grammar adopted to generate scalarizing functions, we decided to include $\sin$, $\cos$, $\log$ and $\exp$ in addition of the basic arithmetic operators.

$$
\begin{array}{lll}
< e > & ::= & < e > + < e > \mid < e > - < e > \mid < e > * < e > \mid < e > / < e > \\
 & & sin(< e >) \mid cos(< e >) \mid log(< e >) \mid exp(< e >) \\
 & & \vec{x} \mid s\vec{um} \mid n \\
 & & < c >< c > . < c >< c > \\
 & & \\
< c > & ::= & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{array}
$$

The second grammar adopted is similar to the previous one, but with the addition of the statistical features used in [100]. In a similar way to $s\vec{um}$, all statistical features are calculated component-wise.

$$
\begin{array}{lll}
< e > & ::= & < e > + < e > \mid < e > - < e > \mid < e > * < e > \mid < e > / < e > \\
 & & sin(< e >) \mid cos(< e >) \mid log(< e >) \mid exp(< e >) \\
 & & \vec{x} \mid s\vec{um} \mid n \\
 & & \vec{\mu} \text{ (mean)} \mid \vec{\sigma} \text{ (standard deviation)} \\
 & & \vec{Q}1 \text{ (1st quartile)} \mid \vec{Q}2 \text{ (2nd quartile)} \mid \vec{Q}3 \text{ (3rd quartile)} \\
 & & \vec{\kappa} \text{ (kurtosis)} \mid \vec{\lambda} \text{ (skewness)} \\
 & & < c >< c > . < c >< c > \\
 & & \\
< c > & ::= & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \, .
\end{array}
$$

7.3.3 Training and validation data

In order to generate our training/validation data, we used some of the different geometries provided by problems from the DTLZ and from the WFG test suites. In Table 7.2, we show the problems adopted to generate sets with different geometries. There is a difference in the number of problems since some problems (such as DTLZ5 and WFG3) are degenerate with 3 or more objectives, changing the shape that they present in their 2-objectives versions.

We used NSGA-III [33] to solve each of the selected problems and stored the population every 100 generations. For problems with 2 objectives we set the population size to be 100 individuals, whereas the problems with 3 and 4 objectives were set to 120 individuals and the problems with 5 objectives were set to 140 individuals. Then, we filtered each of the resulting files to delete all dominated solutions present in the data. This changed the size of elements in each file, since not all solutions are non-dominated in each generation. Next, we normalized the remaining solutions to the interval

Table 7.2: Test problems used in the generation of training/validation sets grouped by their geometry.

(a) Problems used to generate 2-dimensional training sets

Geometry	Test problems
Linear	DTLZ1
Concave	DTLZ2
Mixed	DTLZ7, WFG1, WFG2

(b) Problems used to generate 3, 4 and 5 dimensional training sets

Geometry	Test problems
Linear	DTLZ1, WFG3
Convex	WFG2
Concave	DTLZ2, DTLZ5
Mixed	DTLZ7, WFG1

[0,1]. This is done to avoid the definition of a reference point in the hypervolume approximation function's grammar. Once these files were obtained, we randomly separated them into the training and the validation set. The total number of data files generated with each problem, as well as the size of the corresponding training/validation sets are shown in Table 7.3. The final training/validation sets were created by combining all resulting files from each problem in each dimension, creating 4 different pairs of training/validation sets (one pair per dimension).

In Fig. 7.2 we show the computational time used in each step of the process to create the training/validation data. In the first step we used the PlatEMO [113] implementation of NSGA-III to store the populations. Then, we used Python scripts to Pareto filter such files and to normalize them. Finally, we employed a C implementation [48] to obtain the real hypervolume values of each file. All these steps were performed and measured on an Intel Core i7-8700 CPU, with 16 GB of RAM.

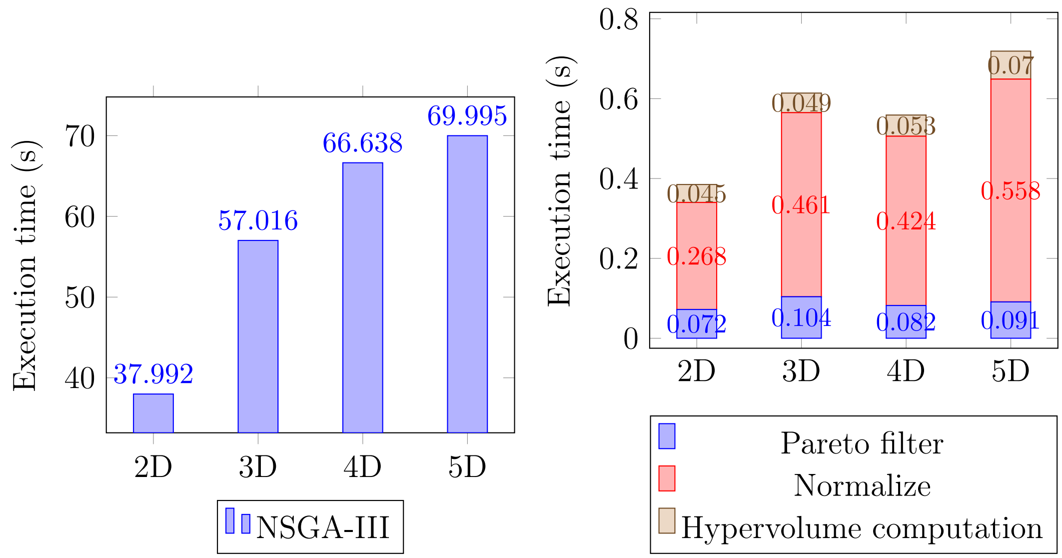
7.4 EXPERIMENTAL WORK

We performed 8 different executions, considering the two grammars previously described for each of the four training sets generated. The maximum number of generations was set to 300, whereas the population size was set to 20. Here, we present the best result obtained from each of these executions.

In eqns (7.7), (7.8), (7.9) and (7.10) we present the best hypervolume approximation functions found using the first grammar for data with 2, 3, 4 and 5 objectives, respectively.

Table 7.3: Number of files generated using each of the selected test problems.

Objectives	Problem	Total files generated	Training set size	Validation set size
2	DTLZ1	996	491	505
	DTLZ2	891	432	459
	DTLZ7	894	433	461
	WFG1	750	383	367
	WFG2	747	363	384
3	DTLZ1	832	438	394
	DTLZ2	868	401	467
	DTLZ5	832	416	416
	DTLZ7	832	431	401
	WFG1	750	388	362
	WFG2	832	400	432
	WFG3	832	398	434
4	DTLZ1	873	408	465
	DTLZ2	868	435	433
	DTLZ5	832	404	428
	DTLZ7	832	411	421
	WFG1	750	382	368
	WFG2	832	433	399
	WFG3	832	421	411
5	DTLZ1	831	406	425
	DTLZ2	833	406	427
	DTLZ5	792	401	391
	DTLZ7	792	392	400
	WFG1	752	362	390
	WFG2	792	406	386
	WFG3	792	375	417



(a) Total computational time used to generate the population files using NSGA-III.

(b) Total computational time used to filter, to normalize and to obtain the real hypervolume of the population files.

Figure 7.2: Computational times used to generate the training/validation sets used for hypervolume approximations.

$$
\begin{aligned}
GEHV1_{2D}(X) =& \frac{1}{n}\sum_{i=1}^{n}\Big(\cos(\vec{x}_{i,1}/\cos(\vec{x}_{i,2}+\sin(34.56))\\
&+\sin(\sin(\vec{x}_{i,1}/\cos(\sin(\vec{x}_{i,2}))\\
&+\vec{x}_{i,2}*\cos(\sin(\sin(\vec{x}_{i,1}*\sin(n-\vec{x}_{i,2}*\cos(\sin(\sqrt{\log(n)}))\\
&*\sqrt{\vec{x}_{i,2}}/\cos(\sin(\sin(\sin(\sin(\vec{x}_{i,1}*\sin(\vec{x}_{i,2})))\\
&*\cos(\sin(\cos(\sin(\sin(\sin(\sin(\cos(\cos(\vec{x}_{i,1}*\sin(\vec{x}_{i,2}))))\\
&*\vec{x}_{i,2}/\cos(\vec{x}_{i,2})))*\cos(\sin(\vec{x}_{i,1}*\sin(\vec{x}_{i,2})*\vec{x}_{i,2}/\sin(\exp(\vec{x}_{i,2}))\\
&*\vec{x}_{i,2}*\vec{x}_{i,2})*\vec{x}_{i,2}))))))*\vec{x}_{i,2}*\cos(\sin(\sin(\sin(\vec{x}_{i,1}*\sin(\vec{x}_{i,2})\\
&*\cos(\sin(\sin(\sin(\cos(07.07+\vec{x}_{i,1})*s\vec{u}m_1*\sin(\vec{x}_{i,1}\\
&*\sin(\vec{x}_{i,2})*\vec{x}_{i,2}/\cos(\vec{x}_{i,2})))))))*\cos(\exp(\vec{x}_{i,1})*n))*\vec{x}_{i,2})\\
&*\cos(n)))))*\sin(\sin(n*\sin(18.34))*\cos(\sin(\vec{x}_{i,2}))\\
&+\vec{x}_{i,2}))*\vec{x}_{i,2})*\sin(\sin(\vec{x}_{i,1}*\sin(18.71))\\
&*\log(\sin(\exp(\sin(34.51)))))))*\sin(\sqrt{\vec{x}_{i,2}})))\Big)
\end{aligned}
\tag{7.7}
$$

$$
\begin{aligned}
GEHV1_{3D}(X) =& \frac{1}{n}\sum_{i=1}^{n}\Big(\cos(\vec{x}_{i,1}+\sin(\cos(\vec{x}_{i,3}))*\exp(\vec{x}_{i,3}/\sqrt{\cos(\vec{x}_{i,3})}))\\
&+\cos(\sin(\vec{x}_{i,2}*\sin(\exp(\vec{x}_{i,2}))+\cos(\sin(\cos(\cos(\vec{x}_{i,3})\\
&*\cos(\sin(\cos(\log(\cos(\cos(n+\vec{x}_{i,3}))))*\vec{x}_{i,3}\\
&+\vec{x}_{i,3}/\cos(\sin(\cos(s\vec{u}m_1))))/\cos(\sin(n\\
&*\sin(\cos(\cos(\vec{x}_{i,3}+\cos(\sin(s\vec{u}m_2)-\vec{x}_{i,2}\\
&-\cos(\vec{x}_{i,1}+\sin(\cos(\vec{x}_{i,3}))+s\vec{u}m_2)))))\\
&+\cos(n+\vec{x}_{i,3})-\vec{x}_{i,3}))))))))\Big)
\end{aligned}
\tag{7.8}
$$

$$
GEHV1_{4D}(X) = \frac{1}{n}\sum_{i=1}^{n}\cos(\vec{x}_{i,1}+\vec{x}_{i,2}*\vec{x}_{i,3}+\vec{x}_{i,4}) \tag{7.9}
$$

$$
\begin{aligned}
GEHV1_{5D}(X) = & \frac{1}{n}\sum_{i=1}^{n}\Bigg(\cos(\vec{x}_{i,1} + \sin(\vec{x}_{i,5}) + \vec{x}_{i,3} * \sin(\vec{x}_{i,1} + \sin(\vec{x}_{i,1}) \\
& - \sin(\sin(s\vec{u}m_5/98.27 * \vec{x}_{i,4} * \vec{x}_{i,3} \\
& + \sin(\sin(\vec{x}_{i,5} - \vec{x}_{i,3} + \sin(\sin(98.22)) \\
& - \sin(s\vec{u}m_2 - s\vec{u}m_3 + \vec{x}_{i,1}) - \sin(\vec{x}_{i,5} * \sin(\vec{x}_{i,5}) + \vec{x}_{i,3} + n) \\
& - \vec{x}_{i,4} + \vec{x}_{i,1} - \sin(\vec{x}_{i,5}) + \vec{x}_{i,3} * \sin(s\vec{u}m_1 + 68.66)) - \sin(\vec{x}_{i,2}) \\
& + \vec{x}_{i,5} * \vec{x}_{i,5} - \vec{x}_{i,3} - \vec{x}_{i,1} + \sin(\exp(\sin(s\vec{u}m_1 \\
& * \sin(s\vec{u}m_1 + \sin(\vec{x}_{i,2}) - \vec{x}_{i,5} * \sin(\vec{x}_{i,3} + \vec{x}_{i,2} * \vec{x}_{i,5} + s\vec{u}m_1 \\
& + \sqrt{\vec{x}_{i,5}} * \vec{x}_{i,3} * \sin(\sin(\vec{x}_{i,5}))))))) + \vec{x}_{i,3} * \sin(\vec{x}_{i,1}/n * \sin(\vec{x}_{i,2}) \\
& * \vec{x}_{i,5} + \sin(\sqrt{s\vec{u}m_4} - 98.06 + \vec{x}_{i,3} * \sqrt{\vec{x}_{i,1} + s\vec{u}m_3 - \vec{x}_{i,2} * \vec{x}_{i,5}} \\
& * \sin(\cos(98.67) + \vec{x}_{i,5} + 96.65 * 97.82 * \sin(\exp(\vec{x}_{i,1}) * \vec{x}_{i,3} \\
& - 98.62 * \sin(\vec{x}_{i,1} * \vec{x}_{i,1}) + \exp(\sin(s\vec{u}m_1 + \vec{x}_{i,2} * \vec{x}_{i,5} + s\vec{u}m_1 \\
& + 29.20 * \sqrt{\vec{x}_{i,5}})) - \vec{x}_{i,3}) + \exp(n)) * \sin(\vec{x}_{i,3} * \exp(s\vec{u}m_3) \\
& * \vec{x}_{i,4} + \vec{x}_{i,1} + s\vec{u}m_2 * \sin(\vec{x}_{i,3}) * \vec{x}_{i,3}) * \sin(s\vec{u}m_1 * \vec{x}_{i,1})))))) \\
& * \sin(\vec{x}_{i,1}) - \sin(\sin(\sqrt{s\vec{u}m_5 + s\vec{u}m_3} + \sqrt{\sqrt{\vec{x}_{i,3}}})))\Bigg) .
\end{aligned}
\tag{7.10}
$$

In eqns (7.11), (7.12), (7.13) and (7.14) we present the best hypervolume approximation functions found using the second grammar for data with 2, 3, 4 and 5 objectives, respectively.

$$
GEHV2_{2D}(X) = \frac{1}{n}\sum_{i=1}^{n}\sin(\cos(\vec{Q2}_2 * \cos(\vec{\sigma}_1) + \vec{Q2}_1)) \tag{7.11}
$$

$$
\begin{aligned}
GEHV2_{3D}(X) =& \frac{1}{n}\sum_{i=1}^{n}\Bigg(\cos(\vec{\mu}_1 + \vec{\mu}_3 + \sin(\sin(\vec{Q1}_2 * \sin(\sin(\vec{Q2}_1 * \vec{\lambda}_1 * \vec{x}_{i,3} \\
& + \sin(\sqrt{\sin(\vec{x}_{i,2})}) * \vec{\mu}_3)) + \exp(\vec{\sigma}_2) * \vec{Q1}_2 * \vec{\sigma}_2 \\
& * (\vec{x}_{i,2} + \vec{x}_{i,1} + \sin(\vec{\mu}_1 + \vec{\mu}_3 + \sin(\sin(\vec{Q1}_2 \\
& * \sin(\sin(\vec{x}_{i,3} * \vec{\lambda}_1 * \sin(\sqrt{\sqrt{\vec{x}_{i,2}}}) * \vec{Q3}_3 + \vec{Q1}_2 \\
& * \sin(\cos(\sin(\vec{x}_{i,2} * \vec{x}_{i,1} * \sin(\sin(\sin(\vec{Q3}_3 + \vec{Q1}_2)) \\
& * \sin(\cos(\vec{\mu}_1 + \vec{\mu}_3 * \vec{\mu}_1 * \vec{\mu}_3 - \sin(\sin(\vec{Q1}_2 \\
& * \sin(\sin(s\vec{u}m_3 * \vec{\sigma}_1)) + s\vec{u}m_2 * \vec{\lambda}_1 * \vec{x}_{i,3} \\
& * \sin(\sqrt{\sqrt{\vec{x}_{i,2}}}))))))) + \vec{\mu}_3 * \exp(\vec{\sigma}_2) - \vec{Q1}_2 * \vec{\sigma}_2)))) \\
& + \sqrt{\vec{x}_{i,2} + \sqrt{\vec{x}_{i,2}} + \vec{x}_{i,1}} * \sin(\vec{\mu}_1 + \vec{\mu}_1 + \vec{\mu}_3 \\
& + \sin(\sin(\vec{\mu}_1 * \vec{\mu}_3 * \vec{Q1}_3 * \vec{Q1}_3)) + \vec{Q1}_2 \\
& * \sin(\sin(\vec{Q2}_1 * \vec{\lambda}_1 * \cos(\sin(\sqrt{\sin(\vec{x}_{i,2})})) + \vec{\mu}_3 \\
& * \exp(\vec{\sigma}_2))) + \vec{Q1}_2 * \vec{\sigma}_2 \\
& * (\vec{x}_{i,2} + \vec{x}_{i,1} + \sin(\sin(\sin(\vec{Q3}_3 + \vec{\sigma}_1)) \\
& * \sin(\cos(\vec{x}_{i,3} * \sin(\vec{\kappa}_2) - \vec{\sigma}_2^2))) + \vec{x}_{i,2} + \sqrt{\vec{x}_{i,2}})^{1/2}) \\
& * \log(\vec{\mu}_1 + \vec{\mu}_3))) + \vec{Q2}_1))^{1/2})))\Bigg)
\end{aligned}
\tag{7.12}
$$

$$
\begin{aligned}
GEHV2_{4D}(X) =& \frac{1}{n}\sum_{i=1}^{n}\Bigg(\cos(\vec{x}_{i,1} + \vec{Q}1_3 + \vec{Q}2_4 + \exp(\vec{Q}3_4) * \vec{\sigma}_2 * \vec{\sigma}_4 \\
& * \cos(\vec{x}_{i,1} * \vec{Q}3_4 * \vec{\sigma}_2 * \vec{\sigma}_4 * \cos(\vec{Q}3_4 * \vec{\sigma}_4) \\
& * \exp(\vec{\kappa}_3) * \vec{x}_{i,3} * \vec{\sigma}_4 * \cos(s\vec{u}m_4) * \vec{\sigma}_4 * \cos(\vec{Q}1_4) \\
& * \vec{x}_{i,1} + \vec{Q}1_3 + \vec{Q}2_4 - \vec{\lambda}_2 * \vec{\sigma}_2 * \vec{\lambda}_4 + \vec{Q}1_3 \\
& + \vec{\sigma}_3^2 / \exp(\vec{Q}3_4) * \vec{Q}1_1 * \vec{\sigma}_4 * \vec{Q}3_1 - \vec{\lambda}_2 * \vec{\sigma}_2 \\
& * \vec{\lambda}_4 + \vec{Q}1_3 + \vec{Q}2_1 * \exp(\vec{Q}3_4) * \vec{x}_{i,1} * \vec{\sigma}_4 * \vec{\sigma}_2 * \vec{x}_{i,1} \\
& * \vec{Q}3_1 - \vec{\lambda}_2 * \vec{\sigma}_2 * \vec{\kappa}_2 + \vec{Q}1_3 + \vec{\sigma}_3^2 * \vec{x}_{i,4} \\
& * \exp(\vec{Q}3_1) * \vec{\lambda}_2 * \vec{Q}3_4 * \vec{\kappa}_2 + \vec{Q}1_3 + \vec{\sigma}_2 * \exp(\vec{Q}3_1) \\
& * \vec{\sigma}_2 * \vec{\sigma}_4 * \vec{\sigma}_4 * \cos(\vec{\mu}_4 * \vec{\sigma}_2 * \vec{\sigma}_2 * \vec{\sigma}_4 \\
& * \cos(\vec{x}_{i,1} * \vec{\sigma}_4^2 * \vec{\sigma}_4^2 * \vec{x}_{i,3} * \vec{x}_{i,2} * \cos(\vec{Q}1_4) \\
& * s\vec{u}m_3 + \exp(\vec{x}_{i,2}) - \vec{Q}1_3 * \vec{Q}2_4 + \vec{\lambda}_2 * \vec{\sigma}_2) \\
& * \vec{Q}1_2 * \vec{\sigma}_2^2 + \sin(\vec{\kappa}_2 + \vec{Q}1_3 * \vec{\sigma}_3^2 * \vec{Q}1_3 * \vec{\sigma}_3^2 * \vec{x}_{i,4} \\
& - \exp(\vec{Q}3_1) * \vec{\lambda}_2 * \vec{\sigma}_2 * \vec{\sigma}_1 + \vec{\mu}_2 * \vec{\sigma}_2 * \vec{Q}3_3 * \vec{\sigma}_2 \\
& * \vec{\sigma}_4 * \cos(\vec{Q}3_1) * \vec{\lambda}_4 + \vec{Q}1_3 + \vec{\sigma}_3^2) * \exp(\vec{\sigma}_2) \\
& * \vec{Q}2_4 * \vec{\sigma}_2 * \vec{\sigma}_1 + \vec{\mu}_2 * \vec{\sigma}_2 * \vec{Q}3_3 * \vec{\sigma}_2 * \vec{\sigma}_4 \\
& * \cos(\vec{Q}3_1) * \vec{\lambda}_4 + \vec{Q}1_3 + \vec{\sigma}_3^2 * \exp(\vec{\sigma}_2) * \vec{Q}2_4 * \vec{\lambda}_2 \\
& * \vec{\lambda}_2 * \vec{\sigma}_2 * \vec{\kappa}_2 + \vec{Q}1_3 + \vec{\sigma}_2 * \vec{\sigma}_2 * \vec{\sigma}_4 * \cos(\vec{Q}2_4) \\
& + \vec{\lambda}_2 * \vec{\sigma}_2 * \vec{Q}1_2 * \vec{\sigma}_2^2 + \sin(s\vec{u}m_3 * \exp(\vec{Q}3_1)) \\
& * \vec{\sigma}_2 * \vec{\sigma}_3^2 + \cos(\vec{Q}3_1)) * \vec{\lambda}_4 + \exp(\vec{Q}1_2) * \vec{\sigma}_3^2 * \vec{\sigma}_2 \\
& * \vec{\sigma}_2 * \vec{\sigma}_4 * \vec{\lambda}_4 * s\vec{u}m_4))\Bigg)
\end{aligned} \tag{7.13}
$$

$$
GEHV2_{5D}(X) = \frac{1}{n}\sum_{i=1}^{n}\cos(\vec{Q}3_5 * \vec{x}_{i,5} + \vec{\mu}_4 + \vec{\mu}_1). \tag{7.14}
$$

In Fig 7.3 we show the computational time required to generate each of these 8 hypervolume approximations. It is important to notice that even though the execution times are considerably high, this process does not need to be repeated each time that we want to obtain the hypervolume ap-

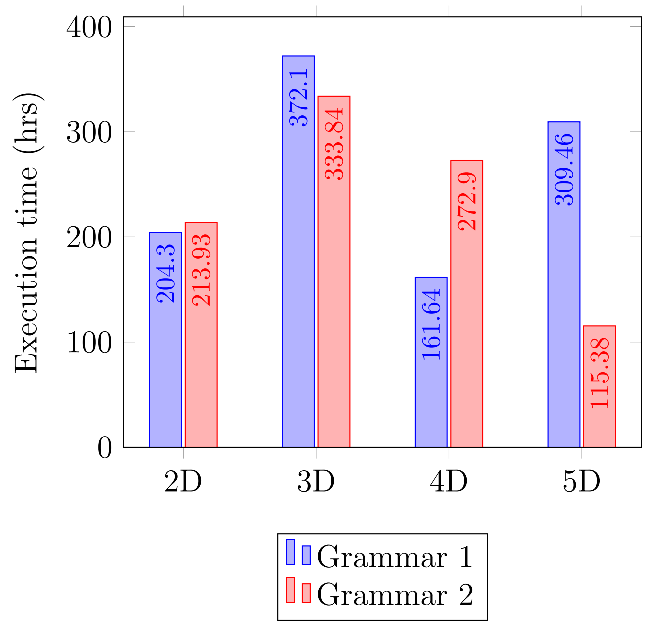


Figure 7.3: Computational time used to generate each of the hypervolume approximation functions using our proposal.

proximation of a given data set. The actual time needed to obtain hypervolume approximations, once the hypervolume approximation functions have been obtained, are shown and compared in the next section.

7.5 Results

In order to validate the performance of our hypervolume approximation functions, we compared them against the Monte Carlo method, considering 10,000 sample points, against the $R_2^H V$ approximation and against the GP-generated approximations [100] using two measures: the average MSE and the average execution time. All these approximations were obtained using Python in the same hardware previously described.

For the $R_2^H V$ approximation, we used the simplex lattice design method [133] to generate an initial set of weight vectors. Then, for each weight vector $\vec{w}$, we generated the corresponding direction vector $\vec{\lambda} = \frac{\vec{w}}{\|\vec{w}\|_2}$, as stated in [109]. In total, we used 9870 direction vectors for 2 and 3-dimensions, 9880 vectors for 4-dimensions and 8855 vectors for 5 dimensions. On the other hand, the reference point $\vec{r} = \{r, \ldots, r\}$ was defined as follows for all the experiments.

$$r = 1 + \frac{1}{H} \tag{7.15}$$

where H is an integer satisfying $C_{m-1}^{H+m-1} \leq N < C_{m-1}^{H+m}$, and C_m^n is the total number of combinations for choosing m elements from a set of n elements, as used in [107].

In Table 7.4, we present the comparison of different hypervolume approximations using MSE on the validation data previously described. In this table we used a gradient to better illustrate the results. The darker the cell, the better the value, which in this case is the smallest value. In all 4 cases, Monte Carlo obtained the smallest error with a two orders of magnitude improvement compared to the second best performing approximation, which are the approximations generated using our proposal. Although all 8 of our approximations share the same order of magnitude, which in turn is two or three orders of magnitude better than the GP-generated approximations, the use of the second grammar produced slightly better results in 3, 4 and 5-dimensional data. This hints that the use of statistical features provides useful information to generate better hypervolume approximation functions. The next best performing approximations are the GP-generated approximations. Lastly, R_2^{HV}, obtained the worst result in all cases.

Table 7.4: Average MSE comparison of hypervolume approximations in validation data.

m	Validation files	Average MSE				
		M^m	$GEHV1_{mD}$	$GEHV2_{mD}$	Monte Carlo	R_2^{HV}
2	2176	-	2.211E-03	3.721E-03	2.209E-05	4.709E-02
3	2906	1.431E-01	2.242E-03	2.018E-03	1.542E-05	8.770E-01
4	2925	2.129E-01	5.879E-03	2.436E-03	1.459E-05	9.202E+00
5	2836	2.765E-01	5.272E-03	5.238E-03	1.395E-05	7.357E+01

On the other hand, we present the comparison of the average computational time, per file, in Table 7.5. Once again, the darker the cell, the better the value, meaning the smallest value. Here, we can observe that the GP-generated approximations obtained the best results in all three available cases, whereas Monte Carlo obtained the worst results in 2 and 3-dimensional data, and R_2^{HV} obtained the worst results in 4 and 5-dimensional data. Regarding the approximations generated with our proposal, they ranged from one order up to two orders of magnitude worsening compared to the best results. This is an expected behavior, since our averaging variant involves an additional iteration of the data against the GP-generated approximations. Additionaly, most of the functions generated with our proposal are more complex. However, three of our eight functions obtained relatively better execution times due to them being the simplest equations generated with our proposal, namely $GEHV1_{4D}$, $GEHV2_{2D}$ and $GEHV2_{5D}$.

Table 7.5: Average computational time comparison of hypervolume approximations in validation data.

m	Validation files	Average Computation Time (s)				
		M^m	$GEHV1_{mD}$	$GEHV2_{mD}$	Monte Carlo	R_2^{HV}
2	2176	-	2.050E-02	1.983E-03	2.224E+00	1.518E+00
3	2906	9.935E-04	1.755E-02	3.288E-02	3.032E+00	2.910E+00
4	2925	1.038E-03	2.617E-03	3.988E-02	2.405E+00	3.873E+00
5	2836	1.113E-03	3.311E-02	2.670E-03	2.340E+00	4.518E+00

From these results we can notice that although Monte Carlo obtained the best approximations in terms of quality (measured by MSE), it is also the most computationally costly. In contrast, our functions obtained better values in terms of quality against the GP-generated approximations at the expense of increasing the computational time required.

Finally, we used the average ratio $\frac{HV_{approx}}{HV_{real}}$ to illustrate better the improvement of the approximations obtained with our functions against the GP-generated approximations. The closer this value is to 100%, the closer the approximation is to the real hypervolume value. A value smaller than 100% represents an underestimation. Consequently a value greater than 100% indicates an overestimation of the hypervolume. In Tables 7.6 to 7.8 we show the average ratio for all available approximations in the validation data classified by the problem used to generate each file.

From these results we can observe that the approximation functions generated with our proposal have a consistent behavior close to 100% in the final average for all three tables. However, this does not occur with the GP-generated approximations. In 3-dimensional data there is an average overestimation of 186.43%, whereas in 4-dimensional data there is an average overestimation of 175.97%. Finally, in 5-dimensional data, the average overestimation consists of 173.5%.

Table 7.6: Average $\frac{HV_{approx}}{HV_{real}}$ ratio comparison of hypervolume approximations in 3-dimensional validation data.

Validation set	Validation files	HV_{approx}/HV_{real}		
		$M^3_{4,6}$	$GEHV1_{3D}$	$GEHV2_{3D}$
DTLZ1_3D	394	143.67%	94.50%	93.59%
DTLZ2_3D	467	161.85%	99.23%	106.01%
DTLZ5_3D	416	333.23%	110.20%	106.61%
DTLZ7_3D	401	184.97%	94.77%	99.97%
WFG1_3D	362	164.59%	105.03%	99.73%
WFG2_3D	432	150.79%	97.81%	93.96%
WFG3_3D	434	165.92%	113.92%	102.01%
Average		186.43%	102.21%	100.27%

Table 7.7: Average $\frac{HV_{approx}}{HV_{real}}$ ratio comparison of hypervolume approximations in 4-dimensional validation data.

Validation set	Validation files	HV_{approx}/HV_{real}		
		$M^4_{5,6}$	$GEHV1_{4D}$	$GEHV2_{4D}$
DTLZ1_4D	465	151.95%	90.89%	95.04%
DTLZ2_4D	433	172.80%	106.59%	106.68%
DTLZ5_4D	428	223.63%	105.03%	98.16%
DTLZ7_4D	421	199.50%	96.65%	99.92%
WFG1_4D	368	161.74%	96.78%	100.02%
WFG2_4D	399	153.36%	93.39%	97.64%
WFG3_4D	411	168.79%	118.16%	99.67%
Average		175.97%	101.07%	99.59%

Table 7.8: Average $\frac{HV_{approx}}{HV_{real}}$ ratio comparison of hypervolume approximations in 5-dimensional validation data.

Validation set	Validation files	HV_{approx}/HV_{real}		
		$M^5_{1,5}$	$GEHV1_{5D}$	$GEHV2_{5D}$
DTLZ1_5D	425	161.56%	93.58%	94.70%
DTLZ2_5D	427	160.83%	96.46%	100.86%
DTLZ5_5D	391	171.04%	96.86%	91.84%
DTLZ7_5D	400	189.48%	107.87%	99.04%
WFG1_5D	390	168.16%	96.92%	98.99%
WFG2_5D	386	163.94%	95.71%	97.06%
WFG3_5D	417	199.49%	114.09%	120.85%
Average		173.50%	100.21%	100.48%

7.6 Summary

In this chapter we have proposed a methodology based on grammatical evolution to generate hypervolume approximations and we obtained two different approximation functions for 2, 3, 4 and 5-dimensional data. We used two different grammars, one composed by basic arithmetic operators and trigonometric functions, and the second one adding statistical features of the data. From our results we can conclude that the use of statistical features seems to generate better approximation functions using our proposal. In all cases, we found a consistent behavior in our experimental validation, both in terms of quality and computational cost: Monte Carlo obtained the best quality values while the GP-generated approximations obtained the best execution time values. However, approximations found with our proposal obtained better trade-offs between these two measures, since they consistently obtained better quality values when compared against the GP-generated approximations at a significantly lower computational time when compared against the Monte Carlo method.

Even though the hypervolume becomes very computationally expensive with 5 or more objectives, we consider that the methodology that we proposed here can be easily extended to generate approximations for higher dimensional spaces.

Also, we showed a ratio comparison, between the hypervolume approximation and the real hypervolume, which we consider to be evidence that the approximation functions generated with GP are possibly not generalizing its good performance in the type of data files that we created in this work. This is particularly noticeable in validation data generated for DTLZ5 with 3 dimensions (with an average overestimation of 333.23%) and DTLZ5 with 4 dimensions (with an average overestimation of 223.63%). These are the extreme cases, but we can find bad quality approximations across all the validation data adopted. However, this same behavior is potentially also present in the approximations we created using GE for different validation data. We believe that both GP and GE are able to generate good results for a certain type of data and even obtain a good generalization but up to a certain point. And in order to ensure a consistent capability to generalize in different data/problems it seems to be necessary to use a large number of different training data.

8

Conclusions

Evolutionary computation techniques, specifically MOEAs, are one of the most useful techniques used to solve multi-objective optimization problems. However, many-objective optimization problems raise additional challenges to MOEAs, leaving room for improvement in different aspects. Namely, diversity maintenance in many-objective problems, is one area that could benefit from more studies and proposals, due to the fact that there is still no consensus on some basic aspects such as which is a representative population size according to the number of objectives, which distribution of solutions is the most adequate, or even a formal definition of diversity.

8.1 Our contributions

In this thesis we have presented our work related to two different aspects of MOEAs in many-objective optimization, which are:

1. The use of s-energy based mating restrictions.
2. The generation of new scalarizing functions and hypervolume approximations using grammatical evolution.

The first subject is directly related to diversity maintenance, while the second one leads to the generation of new density estimators.

8.1.1 S-energy mating restrictions

This work includes the proposal of several mating restrictions based on s-energy, which has been used to assess diversity in MOEAs' populations.

Additionally, we have created an ensemble of four of these mating restrictions that adapts the number of pairs created with each of the mating restrictions throughout the MOEA's execution. Our experimental results show an improvement in the results obtained by incorporating this ensemble against the original algorithm.

From these results we can mainly conclude that, even though diversity is often considered a secondary goal in the optimization process (being convergence the main one), the use of a diversity maintenance technique (such as mating restrictions) that are based in a diversity metric (such as s-energy) can produce a considerable improvement in convergence (measured by the hypervolume) in many-objective optimization problems.

8.1.2 Use of grammatical evolution to improve MOEAs components

In this work we have produced new scalarizing functions and new hypervolume approximations using a grammatical evolution/MOEA hybrid implementation. The former are functions used in decomposition-based MOEAs, while the latter are used both in indicator-based MOEAs and to assess MOEAs' results.

Our new scalarizing functions exhibit an improvement when compared against some state-of-the-art functions. And our experiments around inverse benchmark problems show that scalarizing functions have a direct effect on the diversity of the solutions found. This is because, traditionally, weight vectors need to be modified when working with inverse problems in decomposition-based MOEAs. However, the functions found by our grammatical evolution implementation was able to obtain solutions with a significantly better distribution than the other functions used in the comparison without changing the weight vectors. This is an expected behavior, since in this case the new functions were specifically trained in this type of inverse problems, but this raises the possibility of changing the scalarizing function instead of the weight vectors when dealing with such problems.

On the other hand, the experimental validation of our new hypervolume approximations against state-of-the-art methods show that our proposal represents a good trade-off between accuracy (improving results obtained by most of the other approximations) and computational cost (reducing the execution time of the most accurate approximation).

Both of these results lead us to believe that the use of grammatical evolution to produce different MOEA's components, such as a new density estimator, is a promising research area.

8.2 FUTURE WORK

There are different paths for future research that can be derived from the work presented in this thesis.

1. New s-energy based mating restrictions can be proposed using decision-space information for multi-modal multi-objective optimization problems.

2. An automatic parameters tuning tool, such as the Irace package [79], could be used to select the best mating pool sizes in the s-energy based mating restrictions, or even to select which is the best combination of mating restrictions to generate the ensemble.

3. Our grammatical evolution implementation used to generate scalarizing functions could be modified to generate new functions, but with the goal of improving an explicit combination of diversity and convergence metrics, since the ones we have generated were created with the goal of maximizing the hypervolume obtained by the approximation sets obtained.

4. The scalarizing functions we have generated could be incorporated in an ensemble that is able to select which function to use depending on their performance in the given MOP.t

5. More hypervolume approximations could be generated for higher dimensional spaces or considering training sets with different characteristics.

6. Our grammatical evolution implementation could be modified to generate new diversity maintenance techniques, such as density estimators.

Bibliography

[1] Salem F. Adra and Peter J. Fleming. Diversity Management in Evolutionary Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, April 2011.

[2] R. Aler, D. Borrajo, and P. Isasi. Grammars for learning control knowledge with GP. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*. IEEE, 2001.

[3] Ricardo Aler, Daniel Borrajo, and Pedro Isasi. Using genetic programming to learn and improve control knowledge. *Artificial Intelligence*, 141(1-2):29–56, October 2002.

[4] Thomas Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Inc., USA, 1996.

[5] Johannes Bader, Dimo Brockhoff, Samuel Welten, and Eckart Zitzler. On Using Populations of Sets in Multiobjective Optimization. In Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, pages 140–154. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009.

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

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

[8] Daniel Borrajo and Manuela Veloso. *Artificial Intelligence Review*, 11(1/5):371–405, 1997.

[9] Peter Alexander Nicolaas Bosman. *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*. PhD thesis, Institute of Information and Computing Sciences, Universiteit Utrecht, Utrecht, The Netherlands, 2003.

[10] Peter A.N. Bosman and Dirk Thierens. The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, April 2003.

[11] V. Joseph Bowman. On the relationship of the tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Lecture Notes in Economics and Mathematical Systems*, pages 76–86. Springer Berlin Heidelberg, 1976.

[12] Jürgen Branke and Kalyanmoy Deb. Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Yaochu Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 461–477. Springer, Berlin Heidelberg, 2005. ISBN 3-540-22902-7.

[13] Karl Bringmann and Tobias Friedrich. Tight Bounds for the Approximation Ratio of the Hypervolume Indicator. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature–PPSN XI, 11th International Conference, Proceedings, Part I*, pages 607–616. Springer, Lecture Notes in Computer Science Vol. 6238, Kraków, Poland, September 2010.

[14] Karl Bringmann and Tobias Friedrich. Convergence of Hypervolume-Based Archiving Algorithms. *IEEE Transactions on Evolutionary Computation*, 18(5):643–657, October 2014.

[15] Karl Bringmann, Tobias Friedrich, Frank Neumann, and Markus Wagner. Approximation-Guided Evolutionary Multi-Objective Optimization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1198–1203, Barcelona, Spain, 16-22 July 2011. AAAI Press.

[16] Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the Properties of the $R2$ Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 465–472, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.

[17] Xinye Cai, Haoran Sun, and Zhun Fan. A Diversity Indicator Based on Reference Vectors for Many-Objective Optimization. *Information Sciences*, 430:467–486, March 2018.

[18] William La Cava, Thomas Helmuth, Lee Spector, and Kourosh Danai. Genetic programming with epigenetic local search. In *Proceedings of*

the 2015 on Genetic and Evolutionary Computation Conference - GECCO 15. ACM Press, 2015.

[19] Shao-Wen Chen and Tsung-Che Chiang. Evolutionary Many-objective Optimization by MO-NSGA-II with Enhanced Mating Selection. In *2014 IEEE Congress on Evolutionary Computation (CEC'2014)*, pages 1397–1404, Beijing, China, 6-11 July 2014. IEEE Press. ISBN 978-1-4799-1488-3.

[20] Jixiang Cheng, Gary G. Yen, and Gexiang Zhang. A Many-Objective Evolutionary Algorithm With Enhanced Mating and Environmental Selections. *IEEE Transactions on Evolutionary Computation*, 19(4):592–605, August 2015.

[21] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, August 1999.

[22] Carlos A. Coello Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys*, 32(2):109–143, June 2000.

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

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

[25] Carlos A. Coello Coello, Gregorio Toscano Pulido, and Maximino Salazar Lechuga. Handling Multiple Objectives With Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004.

[26] Yann Collette and Patrick Siarry. *Multiobjective Optimization. Principles and Case Studies*. Springer, August 2003.

[27] Xunxue Cui, Miao Li, and Tingjian Fang. Study of Population Diversity of Multiobjective Evolutionary Algorithm Based on Immune and Entropy Principles. In *Proceedings of the Congress on Evolutionary*

Computation 2001 (CEC'2001), volume 2, pages 1316–1321, Piscataway, New Jersey, May 2001. IEEE Service Center.

[28] D.B. Das and C. Patvardhan. New multi-objective stochastic search technique for economic load dispatch. *IEE Proceedings on Generation, Transmission and Distribution*, 145(6):747–752, November 1998.

[29] A. Kenneth De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, 1975.

[30] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.

[31] Kalyanmoy Deb and Tushar Goel. A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 385–399. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[32] Kalyanmoy Deb and David E. Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 42–50, San Mateo, California, June 1989. George Mason University, Morgan Kaufmann Publishers.

[33] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014.

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

[35] Kalyanmoy Deb and Santosh Tiwari. Omni-optimizer: A Procedure for Single and Multi-objective Optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference,*

EMO 2005, pages 47–61, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

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

[37] Jesus Guillermo Falcon-Cardona, Hisao Ishibuchi, and Carlos A. Coello Coello. Exploiting the Trade-off between Convergence and Diversity Indicators. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI'2020)*, pages 141–148. IEEE, 1-4 December 2020. ISBN 978-1-7281-2548-0.

[38] Jesús Guillermo Falcón-Cardona, Saúl Zapotecas-Martínez, and Abel García-Nájera. Pareto compliance from a practical point of view. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, June 2021.

[39] Ali Farhang-Mehr and Shapour Azarm. Diversity Assessment of Pareto Optimal Solution Sets: An Entropy Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 723–728, Piscataway, New Jersey, May 2002. IEEE Service Center.

[40] Michael Fenton, James McDermott, David Fagan, Stefan Forstenlechner, Michael O'Neill, and Erik Hemberg. Ponyge2: Grammatical evolution in python. 2017.

[41] Jonathan E. Fieldsend, Richard M. Everson, and Sameer Singh. Using Unconstrained Elite Archives for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, June 2003.

[42] David Fogel. Artificial intelligence through simulated evolution. *Evolutionary Computation: The Fossil Record*, pages 227–296, 11 2009.

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

[44] Carlos M. Fonseca and Peter J. Fleming. Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 42–52, Sheffield, UK, September 1995. IEE.

[45] Sen Bong Gee, Kay Chen Tan, Vui Ann Shim, and Nikhil R. Pal. Online Diversity Assessment in Evolutionary Multiobjective Optimization: A Geometrical Perspective. *IEEE Transactions on Evolutionary Computation*, 19(4):542–559, August 2015.

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

[47] David E. Goldberg and Jon Richardson. Genetic algorithm with sharing for multimodal function optimization. In John J. Grefenstette, editor, *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, New Jersey, 1987. Lawrence Erlbaum.

[48] Raquel Hernández Gómez, Jesús Guillermo Falcón-Cardona, and Carlos A. Coello Coello. Considerations in the incremental hypervolume algorithm of the WFG. In *Advances in Computational Intelligence*, pages 410–422. Springer Nature Switzerland, 2022.

[49] Fangqing Gu, Hai lin Liu, and Kay Chen Tan. A Multiobjective Evolutionary Algorithm Using Dynamic Weight Design Method. *International Journal of Innovative Computing Information and Control*, 8(5B):3677–3688, May 2012.

[50] Nasreddine Hallam, Peter Blanchfield, and Graham Kendall. Handling Diversity in Evolutionary Multiobjective Optimisation. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 3, pages 2233–2240, Edinburgh, Scotland, September 2005. IEEE Service Center.

[51] D. P. Hardin and E. B. Saff. Minimal Riesz energy point configurations for rectifiable d-dimensional manifolds. *Advances in Mathematics*, 193(1):174–204, 2005.

[52] Cheng He, Ye Tian, Handing Wang, and Yaochu Jin. A repository of real-world datasets for data-driven evolutionary multiobjective optimization. *Complex & Intelligent Systems*, 6(1):189–197, November 2019.

[53] Raquel Hernández Gómez. *Parallel Hyper-Heuristics for Multi-Objective Optimization*. PhD thesis, Department of Computer Science, CINVESTAV-IPN, Mexico City, México, March 2018.

[54] Raquel Hernández Gómez and Carlos A. Coello Coello. Improved Metaheuristic Based on the $R2$ Indicator for Many-Objective Optimization. In *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 679–686, Madrid, Spain, July 11-15 2015. ACM Press. ISBN 978-1-4503-3472-3.

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

[56] John H. Holland. Concerning efficient adaptive systems. *Self-Organizing Systems*, 230, 1962.

[57] John H. Holland. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, Ann Arbor, Michigan, USA, 1975.

[58] Jeffrey Horn. Multicriterion Decision Making. In Thomas Bäck, David Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, volume 1, pages F1.9:1 – F1.9:15. IOP Publishing Ltd. and Oxford University Press, 1997.

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

[60] Christian Horoba and Frank Neumann. Benefits and Drawbacks for the Use of ε-Dominance in Evolutionary Multi-Objective Optimization. In *2008 Genetic and Evolutionary Computation Conference (GECCO'2008)*, pages 641–648, Atlanta, USA, July 2008. ACM Press. ISBN 978-1-60558-131-6.

[61] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A scalable multi-objective test problem toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, pages 280–295, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

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

[63] Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.

[64] Hisao Ishibuchi, Kaname Narukawa, Noritaka Tsukamoto, and Tusuke Nojima. An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. *Europan Journal of Operational Research*, 188(1):57–75, 1 July 2007.

[65] Hisao Ishibuchi and Youhei Shibata. An Empirical Study on the Effect of Mating Restriction on the Search Ability of EMO Algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 433–447, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[66] Himanshu Jain and Kalyanmoy Deb. 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, August 2014.

[67] Siwei Jiang, Jie Zhang, Yew-Soon Ong, Allan N. Zhang, and Puay Siew Tan. A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm. *IEEE Transactions on Cybernetics*, 45(10):2202–2213, October 2015.

[68] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer Berlin Heidelberg, 2004.

[69] Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 504–512, Berlin, Germany, September 1996. Springer-Verlag.

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

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

[72] Joshua D. Knowles, David W. Corne, and Mark Fleischer. Bounded Archiving using the Lebesgue Measure. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2490–2497, Canberra, Australia, December 2003. IEEE Press.

[73] John R. Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *IJCAI*, volume 89, pages 768–774. Springer-Verlag, 1989.

[74] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.

[75] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, April 2009.

[76] Xin Li, Hu Zhang, and Shenmin Song. A self-adaptive mating restriction strategy based on survival length for evolutionary multiobjective optimization. *Swarm and Evolutionary Computation*, 43:31–49, December 2018.

[77] Xin Li, Hu Zhang, and Shenmin Song. A decomposition based multiobjective evolutionary algorithm with self-adaptive mating restriction strategy. *International Journal of Machine Learning and Cybernetics*, 10(11):3017–3030, February 2019.

[78] Haiming Lu. *State-of-the-art Multiobjective Evolutionary Algorithms—Pareto Ranking, Density Estimation and Dynamic Population*. PhD thesis, Oklahoma State University, Stillwater, Oklahoma, August 2002.

[79] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[80] Samir W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, USA, 1996.

[81] Saúl Zapotecas Martínez and Carlos A. Coello Coello. An Archive Strategy Based on the Convex Hull of Individual Minima for MOEAs. In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, pages 912–919, Barcelona, Spain, July 18–23 2010. IEEE Press.

[82] Adriana Menchaca-Mendez and Carlos A. Coello Coello. An alternative hypervolume-based selection mechanism for multi-objective evolutionary algorithms. *Soft Computing*, 21(4):861–884, August 2015.

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

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

[85] Minami Miyakawa, Hiroyuki Sato, and YujI Sato. Directed Mating in Decomposition-Based MOEA for Constrained Many-Objective Optimization. In *2018 Genetic and Evolutionary Computation Conference (GECCO'2018)*, pages 721–728, Kyoto, Japan, July 15–19 2018. ACM Press. ISBN: 978-1-4503-5618-3.

[86] Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato. Two-Stage Non-Dominated Sorting and Directed Mating for Solving Problems with Multi-Objectives and Constraints. In *2013 Genetic and Evolutionary Computation Conference (GECCO'2013)*, pages 647–654, New York, USA, July 2013. ACM Press. ISBN 978-1-4503-1963-8.

[87] Minami Miyakawa, Keiki Takadama, and Hiroyuki Sato. Controlling selection areas of useful infeasible solutions for directed mating in evolutionary constrained multi-objective optimization. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):25–46, May 2015.

[88] Sanaz Mostaghim, Jürgen Teich, and Ambrish Tyagi. Comparison of Data Structures for Storing Pareto-sets in MOEAs. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 843–848, Piscataway, New Jersey, May 2002. IEEE Service Center.

[89] Miguel Nicolau and Alexandros Agapitos. Understanding grammatical evolution: Grammar design. In *Handbook of Grammatical Evolution*, pages 23–53. Springer International Publishing, 2018.

[90] M. O'Neill and C. Ryan. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358, 2001.

[91] Linqiang Pan, Lianghao Li, Ran Cheng, Cheng He, and Kay Chen Tan. Manifold learning-inspired mating restriction for evolutionary multiobjective optimization with complicated pareto sets. *IEEE Transactions on Cybernetics*, 2021. (in press).

[92] Miriam Pescador-Rojas, Raquel Hernández Gómez, Elizabeth Montero, Nicolás Rojas-Morales, María-Cristina Riff, and Carlos A. Coello Coello. An overview of weighted and unconstrained scalarizing functions. In *Lecture Notes in Computer Science*, pages 499–513. Springer International Publishing, 2017.

[93] Daniel Rivero, Julian Dorado, Juan Rabuñal, and Alejandro Pazos. Generation and simplification of artificial neural networks by means of genetic programming. *Neurocomputing*, 73(16-18):3200–3223, October 2010.

[94] Daniel Rivero, Julián Dorado, Juan R. Rabuñal, and Alejandro Pazos. Modifying genetic programming for artificial neural network development for data mining. *Soft Computing*, 13(3):291–305, May 2008.

[95] Amín V. Bernabé Rodríguez, Braulio I. Alejo-Cerezo, and Carlos A. Coello Coello. Improving multi-objective evolutionary algorithms using grammatical evolution. *Swarm and Evolutionary Computation*, 84:101434, February 2024.

[96] Amín V. Bernabé Rodríguez and Carlos A. Coello Coello. Generation of new scalarizing functions using genetic programming. In *Parallel Problem Solving from Nature – PPSN XVI*, pages 3–17. Springer International Publishing, 2020.

[97] Amín V. Bernabé Rodríguez and Carlos A. Coello Coello. An empirical study on the use of the s-energy performance indicator in mating restriction schemes for multi-objective optimizers. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, June 2021.

[98] Amín V. Bernabé Rodríguez and Carlos A. Coello Coello. An ensemble of s-energy based mating restrictions for multi-objective evolutionary algorithms. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, December 2021.

[99] Amín V. Bernabé Rodríguez and Carlos A. Coello Coello. Designing scalarizing functions using grammatical evolution. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, GECCO '23 Companion. ACM, July 2023.

[100] Cristian Sandoval, Oliver Cuate, Luis C. González, Leonardo Trujillo, and Oliver Schütze. Towards fast approximations for the hypervolume indicator for multi-objective optimization problems by genetic programming. *Applied Soft Computing*, 125:109103, August 2022.

[101] Bruno Sareni and Laurent Krähenbühl. Fitness Sharing and Niching Methods Revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106, September 1998.

[102] Hiroyuki Sato, Hernan Aguirre, and Kiyoshi Tanaka. Variable Space Diversity, Crossover and Mutation in MOEA Solving Many-Objective Knapsack Problems. *Annals of Mathematics and Artificial Intelligence*, 68(4):197–224, August 2013.

[103] J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

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

[105] O. Schütze and C. Hernández. *Archiving Strategies for Evolutionary Multi-objective Optimization Algorithms*. Springer International Publishing, 2021.

[106] O. Schütze, C. Hernández, E-G. Talbi, J. Q. Sun, Y. Naranjani, and F.-R. Xiong. Archivers for the representation of the set of approximate solutions for mops. *Journal of Heuristics*, 25(1):71–105, July 2018.

[107] Ke Shang and Hisao Ishibuchi. A new hypervolume-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 24(5):839–852, October 2020.

[108] Ke Shang, Hisao Ishibuchi, and Xizi Ni. R2-based hypervolume contribution approximation. *IEEE Transactions on Evolutionary Computation*, 24(1):185–192, February 2020.

[109] Ke Shang, Hisao Ishibuchi, Min-Ling Zhang, and Yiping Liu. A new r2 indicator for better hypervolume approximation. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, July 2018.

[110] Ke Shang, Weiduo Liao, and Hisao Ishibuchi. HVC-net: Deep learning based hypervolume contribution approximation. In *Lecture Notes in Computer Science*, pages 414–426. Springer International Publishing, 2022.

[111] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, July 1948.

[112] Theodor Stewart, Oliver Bandte, Heinrich Braun, Nirupam Chakraborti, Matthias Ehrgott, Mathias Göbelt, Yaochu Jin, Hirotaka Nakayama, Silvia Poles, and Danilo Di Stefano. Real-World Applications of Multiobjective Optimization. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors, *Multiobjective Optimization. Interactive and Evolutionary Approaches*, pages 285–327. Springer. Lecture Notes in Computer Science Vol. 5252, Berlin, Germany, 2008.

[113] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87, November 2017.

[114] Ye Tian, Ran Cheng, Xingyi Zhang, Miqing Li, and Yaochu Jin. Diversity assessment of multi-objective evolutionary algorithms: Performance metric and benchmark problems [research frontier]. *IEEE Computational Intelligence Magazine*, 14(3):61–74, August 2019.

[115] Gregorio Toscano Pulido and Carlos A. Coello Coello. Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer. In Kalyanmoy Deb et al., editor, *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I*, pages 225–237, Seattle, Washington, USA, June 2004. Springer-Verlag, Lecture Notes in Computer Science Vol. 3102.

[116] Anupam Trivedi, Dipti Srinivasan, Krishnendu Sanyal, and Abhiroop Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2016.

[117] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Optimization with Messy Genetic Algorithms. In *Proceedings of the 2000 ACM Symposium on Applied Computing*, pages 470–476, Villa Olmo, Como, Italy, 2000. ACM.

[118] Mario Alberto Villalobos-Arias, Gregorio Toscano Pulido, and Carlos A. Coello Coello. A new mechanism to maintain diversity in multiobjective metaheuristics. *Optimization*, 61(7):823–854, 2012.

[119] Handing Wang, Yaochu Jin, and Xin Yao. Diversity Assessment in Many-Objective Optimization. *IEEE Transactions on Cybernetics*, 47(6):1510–1522, June 2017.

[120] Rui Wang, Zhongbao Zhou, Hisao Ishibuchi, Tianjun Liao, and Tao Zhang. Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):3–18, February 2016.

[121] Shuai Wang, Hu Zhang, Yi Zhang, Aimin Zhou, and Peng Wu. A spectral clustering-based multi-source mating selection strategy in evolutionary multi-objective optimization. *IEEE Access*, 7:131851–131864, 2019.

[122] Lyndon While, Lucas Bradstreet, and Luigi Barone. A Fast Way of Calculating Exact Hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, February 2012.

[123] Andrzej P. Wierzbicki. The use of reference objectives in multiobjective optimization. In *Lecture Notes in Economics and Mathematical Systems*, pages 468–486. Springer Berlin Heidelberg, 1980.

[124] Qian Xu, Zhanqi Xu, and Tao Ma. A survey of multiobjective evolutionary algorithms based on decomposition: Variants, challenges and future directions. *IEEE Access*, 8:41588–41614, 2020.

[125] Gaowei Yan, Gang Xie, Keming Xie, and Tsau Young Lin. Granular computing based sorting method in multi-objective optimization. In T.Y. Lin, X. Hu, J. Han, X. Shen, and Z. Li, editors, *GRC: 2007 IEEE International Conference on Granular Computing*, pages 78–82, San Jose, California, USA, November 2-4 2007. IEEE Computer Society Press. ISBN 978-0-7695-3032-1.

[126] Long Yu, Pan Wang, and Haoshen Zhu. A Novel Diversity Preservation Strategy based on Ranking Integration for Solving Some Specific Multi-Objective Problems. In Q. Guo and Y. Guo, editors, *Proceedings of the ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES 2010)*, pages 97–101, Hong Kong, China, August 10-12 2010. IEEE Computer Society Press. ISBN 978-0-7695-4110-5.

[127] Yuan Yuan, Hua Xu, and Bo Wang. Evolutionary Many-Objective Optimization Using Ensemble Fitness Ranking. In *2014 Genetic and Evolutionary Computation Conference (GECCO 2014)*, pages 669–676, Vancouver, Canada, July 12-16 2014. ACM Press. ISBN 978-1-4503-2662-9.

[128] Yuan Yuan, Hua Xu, Bo Wang, Bo Zhang, and Xin Yao. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 20(2):180–198, April 2016.

[129] Fangfang Zhang, Yi Mei, Su Nguyen, and Mengjie Zhang. Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling. *IEEE Transactions on Cybernetics*, 51(4):1797–1811, April 2021.

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

[131] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63, February 2008.

[132] Yi Zhang, Zimu Li, Hu Zhang, Zhen Yu, and Tongtong Lu. Fuzzy c-means clustering-based mating restriction for multiobjective optimization. *International Journal of Machine Learning and Cybernetics*, 9(10):1609–1621, April 2017.

[133] Zhuhong Zhang. Immune optimization algorithm for constrained nonlinear multiobjective optimization problems. *Applied Soft Computing*, 7(3):840–857, June 2007.

[134] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, March 2011.

[135] Aimin Zhou, Qingfu Zhang, and Guixu Zhang. A multiobjective evolutionary algorithm based on decomposition and probability model. In *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*, pages 3151–3158, Brisbane, Australia, June 10-15 2012. IEEE Press.

[136] Chong Zhou, Guangming Dai, Cuijun Zhang, Xiangping Li, and Ke Ma. Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems. *Information Sciences*, 465:232–247, October 2018.

[137] Xin Zhou, Xuewu Wang, and Xingsheng Gu. A decomposition-based multiobjective evolutionary algorithm with weight vector adaptation. *Swarm and Evolutionary Computation*, 61:100825, March 2021.

[138] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicator Via Weighted Integration. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO*

2007, pages 862–876, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

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

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

[141] Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In Xavier Gandibleux, Marc Sevaux, Kenneth Sörensen, and Vincent T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 3–37, Berlin, 2004. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535.

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

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

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