CIMAT

Centro de Investigación en Matemáticas, A.C.

THE IMPORTANCE OF DIVERSITY IN EVOLUTIONARY ALGORITHMS

T H E S I S

A thesis submitted in fulfillment of
the requirements for the degree of

Doctor of Science

with orientation in

Computer Science

Author:

Joel Chacón Castillo

Supervisor:

Dr. Carlos Segura González

Guanajuato, Gto., 11 de 10 de 2022

Centro de Investigación en Matemáticas, A.C.

THE IMPORTANCE OF DIVERSITY IN EVOLUTIONARY ALGORITHMS

T H E S I S

A thesis submitted in fulfillment of
the requirements for the degree of

Doctor of Science

with orientation in

Computer Science

Author:
Joel Chacón Castillo

Supervisor:
Dr. Carlos Segura González

Authorization for final version

Guanajuato, Gto., 09 de 01 de 2023

Acknowledgments

I would like to acknowledge my advisor Carlos for all his guidance and all his time spent in my academic training. Thanks to him, some subjects of computer science are an important part of my life. I am very grateful to my family for supporting me with their love, always lending me their hand in the most difficult times. Thanks, Beatriz. My mother, who has also been very important to me at all times. My two brothers, Humberto and Luis, with their smiles and jokes and their unconditional support. With no less importance, I am grateful to my friend Raul for holding me with his patient and cheering me up, giving me love and having fun in the most difficult times. I am also grateful for all the colleagues that I have met during the eight years of assisting at CIMAT, in particular with the Department of Software Development (Gerencia de Desarrollo de Software), the Department of School Services (Servicios Escolares) and last but not least the Department of Computer Science. Finally, I would like to thank the National Council of Science and Technology (CONACyT) for providing me with financial support during all my PhD degree studies with projects "Ciencia Básica" Project No. 285599, and for the support from "Laboratorio de Supercómputo del Bajio" Project No. 300832. This work was also supported by the Spanish Ministry of Economy, Industry and Competitiveness within the Program "I+D+i Orientada a los Retos de la Sociedad" with contact number TIN2016-78410-R.

Declaration

The author hereby declares that, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this or any other university. This dissertation is my own work and contains nothing that is the outcome of work done in collaboration with others, except as specified in the text and acknowledgments.

Joel Chacón Castillo
January 2023

Abstract

Population-based algorithms are flexible methods that can be effectively applied to complex optimization problems. As part of their design, several aspects have to be taken into consideration. One of them, perhaps the most important, is the early loss of diversity, which leads to premature convergence of the population. However, several works have shown that this situation can be alleviated by taking into account mechanisms that attempt to control the diversity of the population. The aim of this work is to show that the quality of current population-based algorithms can be enhanced by integrating mechanisms to explicitly manage diversity. The key is to consider the stopping criterion and elapsed period in order to dynamically alter the importance granted to the diversity. In this dissertation, enough evidence is collected to conclude that this strategy benefits population-based algorithms mainly for long runs. Even more, this strategy is empirically analyzed in both single-objective and multi-objective problems as well as for continuous and discrete domains.

In the first part of this document, single-objective optimization problems are studied. At the beginning, a Differential Evolution variant with enhanced diversity maintenance is proposed. The main novelty is the use of a dynamic balance between exploration and exploitation to adapt the optimizer to the requirements of the different optimization stages. For this section, the experimental validation is carried out with several benchmark tests proposed in competitions of the "IEEE Congress on Evolutionary Computation" and with the top-rank algorithms of each competition, as well as other diversity-based schemes. The new method avoids premature convergence and significantly improves further the results obtained by state-of-the-art algorithms. Thereafter, regarding single-objective optimization, the Menu Planning Problem is addressed. This optimization problem is a complex task that involves finding a combination of menu items taking into account several types of features, such as nutritional, economical, and level of repetition, among others. To deal with the menu planning problem, some of these features are transformed into constraints. Several variants of this problem have been defined, and memetic algorithms have been quite successful in solving them. Specifically, two formulations based on the transformation of menu planning into a single-objective constrained optimization problem are studied. Each one of these formulations seeks to minimize the costs and the level of repetition, respectively. In general, the proposed memetic algorithm integrates the use of an efficient iterated local search and a novel crossover operator. The importance of integrating an explicit control of diversity is analyzed by using several well-known strategies to control the diversity, as well as our proposal. The results show that to solve this problem in a robust way, the incorporation of explicit control of diversity and ad-hoc operators is mandatory. Note that the last proposal is part of a flexible application that supports user-specification of meals and nutrients intakes, and is currently considered by nutritionists at the *Integral Center of Care for Children and Teenagers* (CIANNA).

The second part of this work is aimed at the field of multi-objective optimization. Most state-of-the-art Multi-objective Evolutionary Algorithms promote the preservation of diversity of objective function space but neglect the diversity of decision variable space. The aim of this part is to show that explicitly managing the amount of diversity maintained in the decision variable space is useful to increase the quality of multi-objective evolutionary algorithms. Since multi-objective evolutionary algorithms can be based on Pareto

dominance, indicators, and/or decomposition, this work proposes an algorithm of each category. Firstly, a novel Dominance-Based Multi-Objective Evolutionary Algorithm, which explicitly considers the diversity of both decision variable and objective function space. In particular, at the initial stages, decisions made by the approach are more biased by information on the diversity of the variable space, whereas it gradually grants more importance to the diversity of the objective function space as the evolution progresses. The latter is achieved through a novel density estimator. The new method is compared with state-of-the-art multi-objective evolutionary algorithms using several benchmarks with two and three objectives. This novel proposal yields much better results than state-of-the-art schemes when considering metrics applied to the objective function space. In the same research line, a novel multi-objective evolutionary algorithm based on decomposition that relies on similar principles by means of replacement phase is proposed. Note that since the aim of this approach is to improve the results when considering metrics in the objective space, the importance given to the diversity in the variable space is reduced as the evolution progresses, meaning that in the later phases, it behaves more similarly to traditional algorithms. Furthermore, to ensure that this decomposition-based algorithm maintains high-quality solutions despite the penalty scheme, an external archive based on the R2-indicator is incorporated. Extensive experimentation shows the clear benefits provided by this design principle. Note that this decomposition algorithm is tested with Differential Evolution operators and more challenging biased problems. Finally, indicator-based multi-objective evolutionary algorithms are popular methods that under simple modification can integrate preference information imposed by a Decision Maker. Therefore, the last multi-objective algorithm is designed following this principle. As with previous approaches, this algorithm explicitly manages the amount of diversity maintained in the decision variable space, which shows benefits for the integration of preference information imposed by the decision maker. This indicator-based algorithm is analyzed with two indicators: the hypervolume indicator and the R2-indicator. Empirical results show a clear improvement of such methods against their original variants. Since important benefits would be obtained in all the tested fields, it can be concluded that the design principle of selecting the amount of diversity to the elapsed period and stopping criterion is beneficial, and it is a principle that should be taken into account when designing meta-heuristics. Finally, it is important to note that these advances have been published in some of the best forums dedicated to optimization, such as the "MIT Press Evolutionary Computation Journal" and the "Genetic and Evolutionary Competition Conference".

Table of contents

List of figures

List of tables

Part I

Motivation and Background

Chapter 1

Introduction

1.1 Research Background

Over the past few decades, the field of optimization has grown significantly. On the basis of recent advances, optimization methods are popularly divided into deterministic and heuristic approaches (Lin et al., 2012). The deterministic approaches take several analytical properties of the problem to generate a sequence of solutions to converge to an optimal solution where some specific features are fulfilled (e.g. linear programming, nonlinear programming, and mixed-integer nonlinear programming). On the other hand, a heuristic approach seeks a solution that better approximates the problem. This is done by taking some decisions that intuitively generate better decisions and that usually incorporate probabilistic steps, meaning that the quality of the solution obtained cannot be guaranteed.

Another common classification divides optimization algorithms into exact and approximate methods (Beheshti and Shamsuddin, 2013; Ezugwu et al., 2021). While the former methods guarantee an exact solution (e.g. dynamic programming, backtracking, and branch-and-bound), the latter methods generate approximate solutions (e.g. greedy, naturally inspired, and local search), which usually turns them into a flexible and efficient approach. These methods are usually non-deterministic strategies, and are usually applied to approximate the most difficult problems in polynomial time (Lewis, 1983), e.g. NP-hard problems with huge search spaces (Chakraborty, 2008). Among non-deterministic algorithms, population-based algorithms are popular methods that have been applied in both continuous and discrete domains (Glover and Kochenberger, 2005; Segura et al., 2015c). In fact, population-based algorithms are usually preferred for the most complete problems and in particular for highly multi-modal problems. Note that in these methods, an optimal solution is not guaranteed, and in some challenging problems finding adequate parameterizations might not be an easy task. A key aspect in population-based approaches is to induce a proper balance between exploration and exploitation of the search space (Talbi, 2009). Several studies have shown that this proper balance can be achieved with the management of diversity in the population (Črepinšek et al., 2013). Diversity can be taken into account in the design of several components. For example, in Evolutionary Algorithms (EAs), which is a popular population-based algorithm, diversity can be managed in the variation stage (Herrera and Lozano, 2003), the replacement phase (Segura et al., 2015c; Wang et al., 2015) and/or by altering the population models (Koumousis and Katsaras, 2006).

In recent years, several population-based algorithm studies have shown that explicitly managing diversity to properly induce a balance between exploration and exploitation is highly important to achieve high-quality solutions (Črepinšek et al., 2013). The explicit consideration of diversity leads to the avoidance of premature convergence, which means that integrating diversity in the design of population-based algorithms is particularly important when dealing with long-term executions. Recently, some diversity management algorithms that combine diversity information, stopping criteria, and elapsed generations have been devised. They have yielded a gradual loss of diversity that depends on the time or evaluations granted to the execution (Segura et al., 2015c). Specifically, the aim of this methodology is to promote exploration in the initial generations and gradually alter the behavior toward intensification. These schemes have provided highly promising results. In fact, in combinatorial optimization (Segura et al., 2017a) some novel replacement strategies that dynamically alter this balance in the search have appeared. This principle has allowed to find new best-known solutions for several classic and popular problems in the scientific community. Some of them are the following. A variant of the frequency assignment problem (Segura et al., 2016a), the two-dimensional packing problem (Segura et al., 2015c), the graph partitioning problem (Romero Ruiz and Segura, 2018), and the job shop scheduling problem (Constantino and Segura, 2021). Furthermore, this novel idea has allowed to solve the three Sudoku puzzles that are considered the most difficult ones (Segura et al., 2016b, 2017b), to solve adequately the Menu Planning Problem (Segura et al., 2019), and to generate Boolean functions with high nonlinearity (López-López et al., 2020).

Finally, the principle of considering the elapsed time and the stopping criterion in the optimization process of population-based algorithms has attracted the attention in recent years (Liu et al., 2019; Wei et al., 2021). In fact, it has been taken into account in the diversity promotion mechanism in several diversity-aware algorithms in both single-objective and multi-objective optimization (Bolufé-Röhler et al., 2013; Chen and Montgomery, 2013; Liu et al., 2016, 2013, 2019; Tamayo-Vera et al., 2016; Wang et al., 2015). Thus, this topic of finding better ways to manage the diversity of the population is an active research line.

1.2 Problem Statement

Population-based algorithms have been adopted to deal with a large number of problems and have been tested under various circumstances. Despite their flexibility and efficiency, a common drawback of these algorithms is premature convergence. This disadvantage is particularly susceptible for long runs, resulting in the waste of computing resources, which becomes more important as the processing power increases. A common strategy to alleviate the premature convergence is the proper promotion of diversity throughout the optimization process. Although several population-based methods have been proposed to manage diversity throughout the search process, very few algorithms are capable of completely avoiding the loss of diversity in long runs. Therefore, diversity might be maintained with the intention of achieving a proper balance between exploration and exploitation throughout the process, and this process should depend on the stopping criteria. Note that attention must be paid to attain this proper balance. A too large exploration degree prevents the proper intensification of the best-located regions, usually resulting in a too slow convergence. In contrast, an excessive degree of exploitation provokes a loss of diversity, meaning that only a limited number of regions are sampled. The problem addressed in this document is precisely to define better ways for the management of diversity that can be applied for different problems and in different subfields of optimization, such as multi-objective, single-objective, combinatorial and continuous optimization.

1.3 Hypothesis

Most of the population-based algorithms face difficulties in controlling the speed of convergence. Therefore, it is quite common that these methods suffer premature convergence, particularly for long runs. The hypothesis of this work is that the performance of standard population-based algorithms can be significantly improved with a proper balance between exploration and exploitation by integrating the elapsed time, and stopping criterion in the management of the diversity of the population. Our hypothesis is that these benefits improve as long as the stopping criterion is extended, so it is specifically useful with the increase of computational resources. Note that the previous statement is independent of the kind of optimization. Therefore, this principle might be applied to design better optimizers in single-objective and multi-objective with discrete and continuous domains.

1.4 Research Objectives

The main objective of this thesis is to demonstrate that most state-of-the-art population-based algorithms suffer premature convergence in long-term executions and that their performance can be improved significantly with an explicit maintenance of diversity that follows the guiding principles described before. This should be validated by integrating strategies that maintain diverse solutions throughout the optimization process into several well-known population-based optimizers. In particular, these strategies should be designed to integrate the elapsed time and the stopping criterion into the components that manage the diversity. In addition, a secondary objective is to clearly show the advantages/disadvantages of the proposed methods against the state-of-the-art algorithms through the analyses of the diversity and convergence in both single-objective and multi-objective problems.

The main objective of this work can be approached through the compliance of the following particular objectives:

- Analyze the way in which loss of diversity affects standard population-based algorithms for long runs.
- Design and test population-based algorithms that maintain a diverse population along the optimization process with the integration of the elapsed time and the stopping criterion in their internal decisions.
- Evaluate the effect of this principle in the most popular benchmarks and in real-world problems.

The validation of the hypothesis is carried out with the analyses of the proposed algorithms in the following ways:

- Single-objective continuous optimization with academic functions.
- Multi-objective continuous optimization with academic functions.
- Real-world combinatorial single-objective and multi-objective optimization problems.

1.5 Research Questions

This section presents the questions that will be answered in this work. In this context, the term diversity refers to the decision variable space.

- How effective is the explicit promotion of diversity in population-based algorithms against state-of-the-art algorithms?
- What kind of problems show less benefits with the explicit promotion of diversity for long runs?
- What kind of problems show more benefits with the promotion of diversity?
- How sensitive are the algorithms to the stopping criterion in diversity-aware algorithms?
- What are the advantages/disadvantages of diversity promotion methods?
- Regarding the methods that explicitly maintain diversity. How sensitive is the decay model that preserves the diversity?
- How sensitive are the algorithms to the parameters required to manage the diversity?
- What is the effect of increasing the number of variables in diversity-aware algorithms?
- Regarding multi-objective optimization. What is the effect of maintaining diversity with two objectives and three objectives problems?

1.6 Significance of the Research Line

Population-based algorithms are one of the most effective methods to deal with optimization problems. Despite their success, a few quantity of algorithms have been designed to entirely avoid premature convergence with the explicit preservation of diversity. Perhaps the main cause of this is that, depending on the problem, these types of mechanisms might delay the convergence of the search process, requiring a large amount of computational power to produce high-quality solutions. However, in recent years, the computational power has been constantly increasing, allowing the evaluation of a huge amount of generations in reasonable time. Nevertheless, most population-based algorithms still preserve the classic validation scheme where the stopping criterion is set for relatively short runs, even though real-world problems are more challenging each time. Recently, a new trend of diversity-aware has emerged where some methods alter their internal decisions depending on the stopping criterion. This principle has shown a remarkable advantage over standard state-of-the-art algorithms in long-term executions. In particular, it controls the convergence of the algorithm with the explicit preservation of diversity taking into account the elapsed time and stopping criterion. Perhaps the most important advantage of this principle is that the quality of solutions can be improved as long as the stopping criterion is augmented because premature convergence is avoided; in other words, these diversity-aware methods allow for an improvement of scalability in terms of computational power. Note that these schemes have allowed to achieve better solutions in the most complex problems. Since at each time these are more complex problems with huge search spaces, as well as more computational power, the significance of these kinds of method will increase in the future.

1.7 Scope of Study

In the literature, a huge quantity of population-based meta-heuristics can be found. Therefore, different ways to classify meta-heuristics as the one proposed in Beheshti and Shamsuddin (2013) (Figure 1.1) have been devised. Note that the hypothesis established in this research refers to all population-based meta-heuristics. However, validation is only performed with Evolutionary Algorithms (EAs). In particular, those EAs are Differential Evolution (DE) and Genetic Algorithms (GA), DE is widely used in continuous single-objective and multi-objective optimization. In contrast, GA can be applied in discrete single-objective optimization and continuous multi-objective optimization. All the proposals designed in this research show the same principle: the explicit promotion of diversity is done by taking into account the elapsed time and the stopping criterion. Therefore, this mechanism is integrated into each strategy. Note that other population-based schemes, such as PSO or EDAs, are not tested. Since these mechanisms do not use the elapsed time and stopping criterion in their internal decisions, the schemes proposed in this investigation seem quite promising in advancing these methods. However, such additional analyses are left for future work.

1.8 Limitations

Despite the fact that this work is limited to EAs, some algorithms that belong to this category are not covered. Some of them are Genetic Programming (GP), Evolutionary Programming (EP), and Evolution Strategy (ES). It is worth mentioning that these strategies can be extended with the same principle designed in this research. However, testing every variant of EA is not possible, so this research selects some of the most popular methods to carry out the validation.

1.9 Definitions of Terms

- Heuristics
  - Heuristics are methods to find good (near-) optimal solutions at a reasonable computational cost without guaranteeing feasibility or optimality (Beheshti and Shamsuddin, 2013).
  - A heuristic is a methodology of reasoning in problem solving that enables a solution to a problem driven by trial and error (Hussain et al., 2019).
- Meta-heuristic
  - An iterative generation process that guides a subordinate heuristic by combining intelligently different concepts to explore and exploit the search space, learning strategies are used to structure information to find near-optimal solutions efficiently (Osman and Laporte, 1996).
  - An iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions per iteration. Subordinate heuristics may be high-level (or low-level) procedures, a simple local search, or just a construction method (Voß et al., 2012).
- Population-based algorithms
  - In mathematical optimization, these are strategies that store and preserve a whole group of candidate solutions, where each individual solution equivalent to a distinct point within the search space of the problem being solved are regularly updated (Babalola et al., 2020)

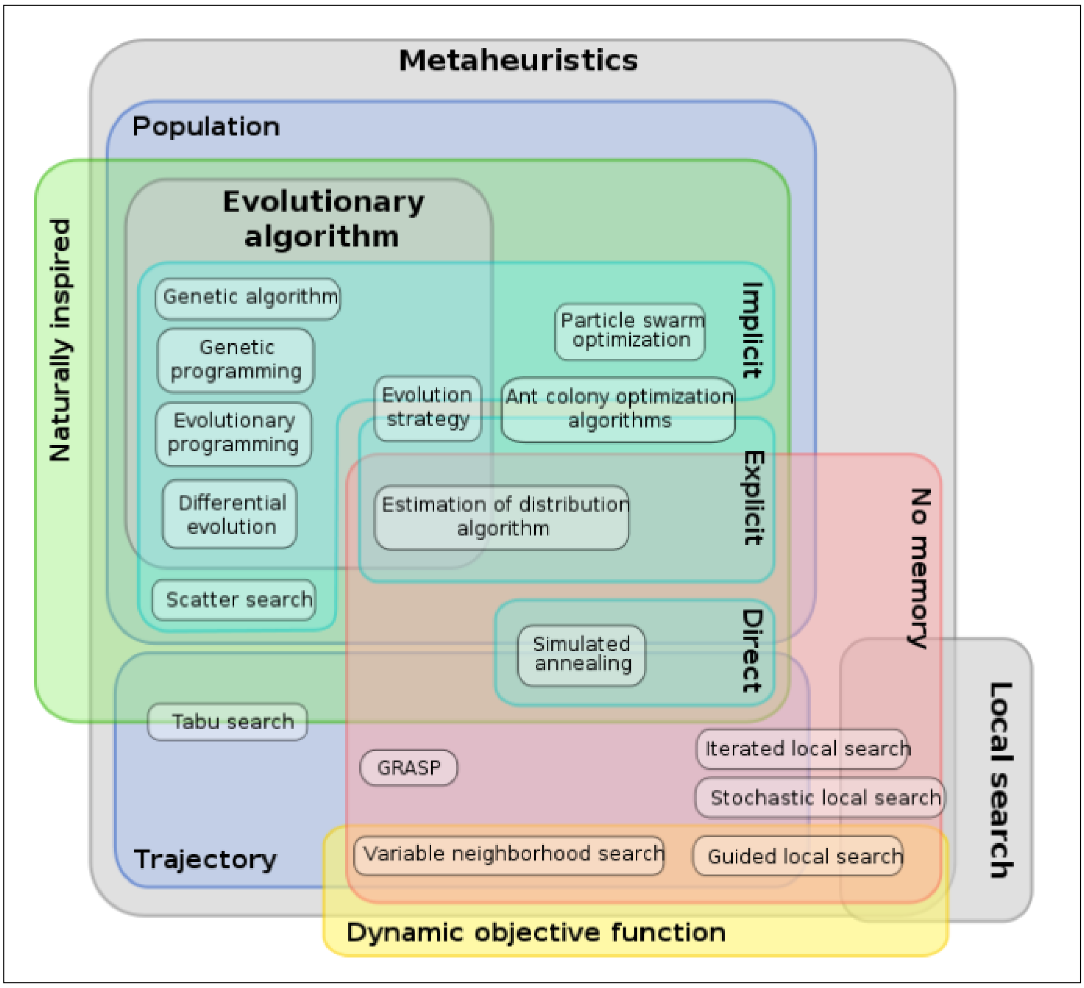

Fig. 1.1 Common classifications of meta-heuristics. Figure taken from Beheshti and Shamsuddin (2013)

- Premature convergence
  - In genetic algorithms, it is a situation where the population of chromosomes reaches a configuration such that the crossover no longer produces offspring that can outperform their parents, as must be the case in a homogeneous population (Fogel, 1994).
  - A situation in which all members of the population are located in a reduced part of the search space (different from the optimal region) and the selected components do not allow escape from such a region (Segura et al., 2015c).
- Exploration
  - The ability of a search algorithm to discover a diverse assortment of solutions, spread over different regions of the search space (Morales-Castañeda et al., 2020).
  - The process of visiting entirely new regions of a search space (Črepinšek et al., 2013).
- Exploitation

  - Emphasizes the idea of intensifying the search process over promising regions of the solution space, with the aim of finding better solutions or improving existing ones (Morales-Castañeda et al., 2020).
  - The process of visiting those regions of a search space within the neighborhood of points previously visited (Črepinšek et al., 2013).
  - Finding the local optimum within a given basin (Gonzalez-Fernandez and Chen, 2015).

- Attraction basins

  - Part of a search region with a point called an attractor to which a system tends to evolve (Jerebic et al., 2021).
  - A region of the search space that contains all the solutions that lead to a particular local optimum when (greedy) a local search is used (Gonzalez-Fernandez and Chen, 2015).

- Genetic drift

  - A state that occurs when there is not enough pressure to discriminate between two or more distinct solutions (Harik et al., 1999).

1.10 Contributions

The main contributions of this thesis are:

- **A review of works related to the management of diversity in population-based algorithms**. This review considers a vast number of works related to diversity in the decision-variable space. In particular, this review takes into account the kind of strategies that attempt to induce a balance between exploration and exploitation by properly managing diversity throughout the optimization process.

- **Analyses of the diversity of some popular optimizers**. The validation of each section includes an analysis of the diversity maintained by the state-of-the-art methods in the most popular test problems, which is information that is not usually analyzed in the corresponding papers.

- **The integration of a novel principle in several variants of evolutionary algorithms**. A new design principle is devised and integrated with several single-objective, multi-objective, continuous, and combinatorial optimization problems, providing important benefits in all these fields.

1.11 Outline of the Thesis

This thesis is divided into two parts. The first part "Motivation and Background" addresses the theoretical framework to carry out this thesis and provides a comprehensive review of the literature on diversity in population-based algorithms. The second part "Proposals and Achievements" is devoted to describe the algorithmic proposal with its validation. Note that each chapter of the second part presents its own experimental validation. This thesis is composed of eight chapters and the appendixes.

Chapter 2 provides a review of the most popular methods that promote diversity in the decision variable space for single-objective and multi-objective optimization. Note that the quantity of strategies that manage diversity has been increasing in recent years, so it is important to present them in an ordered way to better show the novelty of this research.

Chapter 3 presents an extension of Differential Evolution that incorporates a novel replacement strategy that includes a mechanism to preserve diversity explicitly. The main contribution is that it promotes a dynamic balance between exploration and exploitation to adapt the optimizer to the requirements of the different optimization stages. As mentioned before, this balance depends on the stopping criterion.

Chapter 4 addresses the menu planning problem that involves finding a combination of menu items by taking into account several kinds of features, such as nutritional and economical. In this section, two memetic algorithms are proposed. One considers a set of fixed constraints and was developed in collaboration with the "Universidad de La Laguna" and the second introduces user-defined constraints elaborated in collaboration with the "Centro Integral de Atención a Niñas, Niños y Adolescentes" (CIANNA).

Chapter 5 introduces a novel multi-objective evolutionary algorithm based in dominance which explicitly considers the diversity in both decision variable and objective function space. This information is used to properly adapt the balance between exploration and intensification during the optimization process. In this section, the new method is compared with state-of-the-art methods using several benchmarks with two and three objectives. Note that although our proposal is not designed for multi-modal multi-objective optimization, the validation takes into account a multi-modal multi-objective algorithm.

Chapter 6 focuses on the design of a multi-objective evolutionary algorithm based on decomposition. This section shows that the quality of these types of methods can be enhanced by integrating mechanisms to explicitly manage the diversity in the variable space. In particular, the validation of this proposal considers Differential Evolution operators and a recent set of test problems with bias features, which is one of the most challenging difficulties that multi-objective evolutionary algorithms might face in real-world problems.

Chapter 7 is devoted to indicator-based multi-objective evolutionary algorithms, which are especially useful to integrate preference information imposed by the decision maker. The validation shows the benefits of maintaining diversity into hypervolume and R2 indicator algorithms. It also shows the advantages of explicitly managing diversity when preference information is integrated.

The conclusions gathered along all the thesis as some possible future lines of research are presented in Chapter 8.

The Appendices provide some basic concepts and definitions of evolutionary algorithms, as well as the definition of all the test problems that are considered in this thesis.

1.12 Publications and Participation in Congresses

The following book chapters, conferences, and journals were produced during the development of this thesis:

- Joel Chacón, Carlos Segura. Analysis and Enhancement of Simulated Binary Crossover. 2018 IEEE Congress on Evolutionary Computation (CEC) (Pages 1-8).

- Carlos Segura, Joel Chacón. Importancia de la Diversidad en el Diseño de Algoritmos Evolutivos. Academia Mexicana de Computación (AMEXCOMP) (Pages 121-154).

- Carlos Segura, Gara Miranda, Eduardo Segredo, Joel Chacón. A Novel Memetic Algorithm with Explicit Control of Diversity for the Menu Planning Problem. 2019 IEEE Congress on Evolutionary Computation (CEC), (Pages 2191-2198). DOI: 10.1109/CEC.2019.8790339

- Joel Chacón, Carlos Segura. Differential Evolution with Enhanced Diversity Maintenance. Optimization Letters (2019), (Pages 1471-1490), ISSN: 1862-4480.

- Joel Chacón, Carlos Segura. A variant of Differential Evolution with Enhanced Diversity Maintenance. Numerical and Evolutionary Optimization (2020).

- Joel Chacón, Carlos Segura, Carlos Coello. VSD-MOEA: A Dominance-Based Multi-Objective Evolutionary Algorithm with Explicit Variable Space Diversity Management. Evolutionary Computation Journal (2021), ISSN: 1063-6560.

- Carlos Segura, Joel Chacón, Oliver Schütze. The Importance of Diversity in the Variable Space in the Design of Multi-objective Evolutionary Algorithms. Applied Soft Computing Journal (Under Review)

- Carlos Segura, Joel Chacón. Explicit Diversity Variable Space Promotion with Preference Information in Indicator-Based Selection Multi-Objective Evolutionary Algorithms (to be submitted).

Chapter 2

Literature Review

Abstract

This chapter is devoted to review state-of-the-art methods in single-objective and multi-objective optimization and to provide a comprehensive review of mechanisms that attempt to achieve a balance between exploration and exploitation in population-based meta-heuristics. Initially, the attendance of Differential Evolution variants along the last decade is briefly described in IEEE Congress in Evolutionary Computation (CEC) contests [1]. Note that this last review takes into account several kinds of contests, e.g. learning-based optimization, large-scale optimization, and classic real parameter optimization. Then, a description of the multi-objective evolutionary algorithms which takes part in experimental validations of several chapters is presented. Later, a detailed review of the most cited works regarding exploration and exploitation is carried out. Finally, a careful review of diversity strategies designed to promote a balance between exploration and exploitation is summarized.

2.1 Single-Objective Optimization

2.1.1 Performance of IEEE Congress in Evolutionary Computation Contests

In recent years, several contests have been organized at the IEEE CEC to facilitate comparisons among single-objective continuous optimizers. Such contests usually define a set of optimization functions with different features and complexities, so analyzing the results through the years offers insight about which principles and algorithms provide more advantages. This section is devoted to summarize the methods and ideas with a stronger contribution. In this thesis, a novel DE variant is devised, so our efforts are focused on DE variants with the aim of detecting design tendencies in the DE field. A detailed review of "Global Optimization Competitions" can be explored in Molina et al. (2018, 2017).

[1] The codes that were submitted in the contests can be obtained in the repository of Suganthan https://github.com/P-N-Suganthan

The first real-parameter competition was carried out in CEC 2005 (Suganthan et al., 2005), where a DE variant attained the second rank and the self-adaptive DE variant called SaDE (also known as L-SaDE) obtained the third rank in 10 dimensions. However, they performed poorly with more than 30 dimensions. Subsequently, in the 2008 competition on large-scale global optimization (Tang et al., 2007), a self-adaptive DE (jDEdynNP-F) reached third place, confirming the importance of parameter adaptation. In fact, in other kinds of competitions, such as in the 2006 constrained optimization one, the benefits of adaptation were also shown, where SaDE obtained the third place. In a subsequent competition in large-scale optimization (CEC 2010), DE variants did not reach the top ranks. This, together with the fact that several DE variants performed properly only in low dimensionality, is an indicator of the susceptibility regarding the parameterization in DE for large-scale problems. In fact, this situation regarding the curse of the dimensionality is shown in Segura et al. (2015b). Thus, it is known that there is room for improvement in terms of scalability. However, large-scale optimization is out of the scope of this thesis. In the CEC 2011 competition with real-world optimization problems (Das and Suganthan, 2010), hybrid algorithms including DE have performed properly. For instance, the second place was obtained by the hybrid DE called DE-Λ_{CR}. Again, a self-adaptive multi-operator DE (SAMODE) performed well and obtained the third position. Note that for this contest, most of the proposals were DE variants (approximately 70% of the proposals) and most of them were memetic algorithms. Finally, in CEC 2013 DE variants did not achieve a top place: however, for this competition, preliminary variants of the next winning algorithms such as MVMO (Rueda and Erlich, 2013) and SHADE (Tanabe and Fukunaga, 2013) were designed. In fact, several variants of these algorithms are ranked on the top for recent contests. Note that SHADE is mainly based on DE.

In recent years, adaptive variants have also stood out. However, the complexity of the best schemes has increased considerably. In the 2014 competition on real parameter optimization (Liang et al., 2013), the first place was reached by the Linear Population Size Reduction Success-History Based Adaptive DE (L-SHADE). Similarly to other adaptive variants, this proposal adapts the internal parameters of DE. Note that this method attempts to achieve a better balance between exploration and exploitation dynamically by reducing the population size. In the 2015 competition based on learning (Liang et al., 2014), a variant of the previous approach obtained the first place. Furthermore, two DE variants with parameter adaptation achieved the second and third positions.

In the case of 2016 (Liang et al., 2013), the first place was reached with the United Multi-Operator Evolutionary Algorithm (UMOEAs-II) (Elsayed et al., 2016). Note that although this approach is not a DE scheme, some of the DE operators are taken into account. The second place was reached by Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood (L-SHADE-EpSin) (Awad et al., 2016) and the third place was attained by the Improved L-SHADE (iL-SHADE). Note that these two approaches were again variants of SHADE. In fact, variants of SHADE have also excelled in learning-based competitions (Liang et al., 2014). In the CEC 2017 case (Wu et al., 2017), the first place was obtained by the Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase (EBOwithCMAR) (Kumar et al., 2017), which is not a DE variant. EBOwithCMAR is an extension of UMOEAs-II. The second place was reached by jSO (Brest et al., 2017), which is an improvement of iL-SHADE. Finally, L-SHADE-EpSin, again a variant of SHADE, attained the third place. The influence of DE on CEC global competitions has been remarkable, which is summarized in Table 2.1.

In general, these competitions are valuable because comparisons allow us to identify the most promising novel ideas. Nevertheless, as is shown in Molina et al. (2017), there is not always a clear improvement from one competition to another one; in other words, over the years, the winner algorithms do not always obtain better results than the previous ones; in fact, some previous winners can improve new ones in the most recent

Table 2.1 List of algorithms in CEC global competitions (sorted by ranking), variants of DE are in **bold**. This table can be complemented with the information presented in Molina et al. (2018).

Benchmark	Competitions and Year	Algorithm	Reference
CEC 2005 Global Optimization	CEC 2005	IPOP-CMAES	Auger and Hansen (2005)
		SaDE	Qin and Suganthan (2005)
		DMS-L-PSO	Liang and Suganthan (2005)
CEC 2006 Constraint Optimization	CEC 2006	**εDE**	Takahama and Sakai (2006)
		DMS-PSO	Liang and Suganthan (2005)
		jDE-2	Brest et al. (2006)
CEC 2008 LSGO	CEC 2008 LSGO	MTS	Tseng and Chen (2008)
		LSEDA-gl	Wang and Li (2008)
		jDEdynNP	Brest et al. (2008)
CEC 2011 Global Optimization	CEC 2011	GA-MPC	Elsayed et al. (2011b)
		DE-Λ_{CR}	Reynoso-Meza et al. (2011)
		SAMODE	Elsayed et al. (2011a)
CEC 2013 Global Optimization	CEC 2013	IPOP-CMA-ES	Liao and Stuetzle (2013)
		NBIPOPaCMA	Loshchilov (2013)
		DRMA-LSCh-CMA	Lacroix et al. (2013)
CEC 2014 Global Optimization	CEC 2014	**L-SHADE**	Tanabe and Fukunaga (2014)
		UMOEA	Elsayed et al. (2014)
		MVMO-SH	Erlich et al. (2014)
CEC 2015 Learning-Based Optimization	CEC 2015	**SPS-L-SHADE-EIG**	Guo et al. (2015)
		DEsPA	Awad et al. (2015)
		MVMO	Rueda and Erlich (2015)
		LSHADE- ND	Sallam et al. (2015)
CEC 2014 Global Optimization	CEC 2016	UMOEA-II	Elsayed et al. (2016)
		LSHADE-EpSin	Awad et al. (2016)
		DRPSO	Tanweer et al. (2015)
		iL-SHADE	Brest et al. (2016)
CEC 2015 Learning-Based Optimization	CEC 2016	MVMO-PHM	Rueda Torres and Erlich (2016)
		LSHADE44	Poláková et al. (2016)
		CCLSHADE	Omidvar et al. (2013)
CEC 2017 Global Optimization	CEC 2017	EBOwithCMAR	Kumar et al. (2017)
		jSO	Brest et al. (2017)
		LSHADE-cnEpSin	Awad et al. (2017)

competitions. This might occurs since old winners are not always considered for new competitions, but they might be still competitive on modern benchmarks.

Interestingly, in the last decade three algorithms have strongly influenced the development of modern methods in real-parameter competitions, which are CMA-ES, L-SHADE, and MVMO as explained in Molina et al. (2018). In addition, by examining the characteristics of the different approaches, the following trend is detected:

- Typically, the parameters are altered during the run to adapt the optimizer to the requirements of the different optimization stages.

- In some of the last algorithms, the adaptation of parameters or components considers the stopping criterion and elapsed generations to bias the decisions taken by the optimizer. For instance, some proposals decrease the population size and in other cases DE is modified to further intensify in the last stages.

- The overall complexity of the winners has increased significantly. Particularly, several variants include modifications to perform promising movements with a higher probability, for instance by including the principles of the Covariance Matrix Adaptation scheme.

- Several algorithmic proposals integrate strategies that consider a diversity measure. For instance UMOEAs-II and EBOwithCMAR calculate the normalized diversity as part of its improvement measure, which affects the update of the parameters.

Note that the previous conclusions are taken into account for the algorithmic proposal described in Chapter 3, which, according to our hypothesis, for long runs simpler variants with explicit control of diversity are enough to excel. In this new scheme, some of the classic modifications (e.g. success history-based adaption) might be counterproductive. In fact, it is known that parameter adaptation might cause some improper movements that affect performance for long runs (Montgomery and Chen, 2010). Note that by controlling the diversity, the degree between exploration and exploitation can be properly altered automatically. As a result, the adaptation of the parameters is not included in this thesis. Nevertheless, we consider that incorporating an adaptive parameterization might be beneficial, but it should be included carefully, thus it is left for future work. Note that methods that include adaptive parameterization are taken into account for validation.

2.1.2 State-of-the-art for Single-Objective Optimization

This section describes the four algorithms taken as state-of-the-art in Chapter 3. In particular, the results obtained with the benchmarks proposed in the competitions in real-parameter optimization CEC 2016 and CEC 2017 are taken into account[2] for selecting the state-of-the-art algorithms. Note that the top algorithms of two years were considered since the winner algorithms of 2017 might be improved by previous years or the other way around (Molina et al., 2017). In particular, four algorithms are taken into account: UMOEAs-II (Elsayed et al., 2016), EBOwithCMAR (Kumar et al., 2017), L-SHADE (Tanabe and Fukunaga, 2014), and jSO (Brest et al., 2017). Although the first two are multi-operator frameworks, the second two follow the same principles of DE.

First, UMOEAs-II adopts the concept of multiple algorithms empowered by multiple operators, in which the initial population is divided into two sub-populations, each of which evolves independently using a multi-operator DE or CMA-ES. The probability of applying each algorithm is dynamically changed based on the quality of the solutions and the diversity. In addition, it adopts a sharing scheme and, in the last stages of optimization, it applies a local search based on the interior point method (Nesterov and Nemirovskii, 1994). Similarly, EBOwithCMAR is inspired in the framework of UMOEAs-II which is mainly conformed by two algorithms: Effective Butterfly Optimizer (EBO) and Covariance Matrix Adapted Retreat Phase (CMAR). The former is a self-adaptive butterfly optimizer that incorporates two new crossover operators to increase the diversity of the population. The latter uses a covariance matrix to generate a new solution that improves the local search capability of the EBO. Note that in this framework, the parameters are adapted during the optimization process, through the Success History-Based Adaption (SHBA). Additionally, this method considers the Linear Population Size Reduction (LPSR) and to improve the exploitation potential, it integrates Sequential Quadratic Programming (SQP) at the latter phases with a dynamic probability.

L-SHADE-EpSin can be considered as an extension of L-SHADE (Tanabe and Fukunaga, 2014) and introduces a self-adaptive framework to adapt the control settings during search. In contrast, it uses an ensemble sinusoidal approach to automatically adapt the values of the scaling factor of DE. This ensemble approach consists of a mixture of two sinusoidal formulas: A non-adaptive sinusoidal decreasing adjustment and an adaptive history-based sinusoidal increasing adjustment. Similarly to L-SHADE, it also applies a linear population size reduction, and a local search is triggered as soon as the population size is decreased to 20

[2]The test problems presented in CEC 2016 are the same problems presented on CEC 2014.

individuals. In particular, this local search is based on the Gaussian walk, applied in the last generations to improve the potential for exploitation. Finally, jSO is considered to be one of the most recent variants of SHADE. If we look at the history of how an algorithm was inspired by its predecessor, a close relation might be JADE (Zhang and Sanderson, 2009), SHADE (Tanabe and Fukunaga, 2013), L-SHADE (Tanabe and Fukunaga, 2014), iL-SHADE (Brest et al., 2016) and jSO (Brest et al., 2017). In jSO the main features of iL-SHADE remain unchanged. However, jSO introduces a new weighted version for the mutation strategy. The aim of this weighted mutation strategy is to apply a smaller factor that multiplies difference vectors at the early stages of the evolutionary process, while in later stages, a higher factor is taken into consideration. To summarize, the main features of iL-SHADE are as follows:

- Higher values for the CR control parameter (crossover probability) are propagated during the optimization process.
- A memory update mechanism stores historical memory values of the current generation, and it uses weighted Lehmer means to calculate historical memory values.
- Very high values of F and low values of CR are not allowed in the early stages of the evolutionary process.
- The mutation strategy changes dynamically depending on the stopping criterion.

Note that SHADE and its variants have obtained several top positions in the last decade, for instance CEC 2015 with L-SHADE-ND (Sallam et al., 2015), and SPS-L-SHADE-EIG (Guo et al., 2015), and in CEC 2016 with LSHADE-EpSin (Awad et al., 2016), iLSHADE (Brest et al., 2016), and LSHADE44 (Poláková et al., 2016).

2.2 Multi-Objective Optimization

In recent decades, the field of MOEAs has gained popularity. As a result, there are a large number of MOEAs available. In order to better classify the different schemes, several taxonomies have been proposed (Bechikh et al., 2016). Most current techniques consider in some way at least one of the following concepts (Trivedi et al., 2016): Pareto dominance, indicators, and/or decomposition. Note that several MOEAs use more that one of these principles, but most practitioners make the effort to classify proposals as domination-based, indicator-based, or decomposition-based. Since all of them have quite competitive representatives, this thesis proposes new methods belonging to each one of the different groups.

The experimental validation carried out along this thesis includes the Non-Dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002a), the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Li and Zhang, 2008; Zhang and Li, 2007; Zhang et al., 2009), S-metric Selection Evolutionary Multi-Objective Optimization Algorithm (SMS-EMOA) (Beume et al., 2007), the R2-Indicator-Based Evolutionary Multi-Objective Optimization Algorithm (R2-EMOA) (Trautmann et al., 2013), the Non-Dominated Sorting Genetic Algorithm III (NSGA-III) (Deb and Jain, 2013) and the Convergence Penalized Diversity Evolutionary Algorithm (CPDEA) (Liu et al., 2019). Note that at least one algorithm of each group is included.

2.2.1 Domination-Based MOEAs

This kind of MOEAs are based on the application of the Pareto dominance relation. Since the dominance relation does not inherently promote the preservation of diversity in the objective space, auxiliary techniques such as niching, crowding, and/or clustering are usually integrated with the aim of obtaining a proper spread and diversity of the objective space. One popular algorithm for this group is the NSGA-II. This algorithm (Deb et al., 2002a) implements a special parent selection operator and incorporates the use of elitism in the replacement phase. The selection operator is based on two mechanisms: fast-non-dominated-sort and crowding. The first one provides a bias based on the Pareto dominance relation, whereas the second one promotes the preservation of diversity in the objective space. This method has been extended in a large number of ways (Kukkonen and Deb, 2006b). NSGA-II is now a mature algorithm that has been incorporated in many popular frameworks. Thus, it is usual to compare the results of new algorithms against this method, so NSGA-II has been one of the methods selected of this group. The most traditional MOEAs belonging to this group are quite effective for optimization problems with two and three objectives. Nevertheless, they face some difficulties for many-objective optimization. NSGA-III (Deb and Jain, 2013) extends NSGA-II by replacing the crowding selection with a strategy based on generating distributed reference points that implicitly decompose the objective space (Trivedi et al., 2016). While this method is aimed at many-objective optimization, it also provides some benefits for problems with three-objectives.

2.2.2 Decomposition-Based MOEAs

Decomposition-based MOEAs (Zhang and Li, 2007) are based on transforming the MOP into a set of single-objective optimization problems that are tackled simultaneously. This transformation can be performed in several ways, for example, with a linear weighted sum, a weighted Tchebycheff function, or with the Achievement Scalarizing Function (ASF) (Hernández Gómez and Coello Coello, 2015; Pescador-Rojas et al., 2017). Given a set of weights to establish different single-objective functions, the MOEA searches for a single high-quality solution for each of them. Weight vectors should be selected with the aim of obtaining a well-spread set of solutions. However, this is a difficulty for these kinds of approaches because the selection of proper weights might depend on the form of the Pareto front (Qi et al., 2014).

MOEA/D (Zhang and Li, 2007) is one of the most popular decomposition-based MOEAs. Its main principles include problem decomposition, weighted aggregation of objectives, and mating restrictions through the use of neighborhoods. Different ways of aggregating the objectives with MOEA/D have been tested. Among them, the use of the Tchebycheff approach and the Achievement Scalarizing Function approach are quite popular.

A special feature of MOEA/D is the definition of neighborhoods. Each subproblem is associated with a set of close subproblems in terms of the distances of their weights. These subproblems are said to belong to its neighborhood. Then, in each mating operation, a subproblem is selected, and two individuals belonging to the corresponding neighborhood are used as input of the genetic operators to create a new individual given an update criterion. Finally, the new individual is used to update — if required — the best solution found for any of the subproblems of the considered neighborhood. In some variants of MOEA/D, the maximum number of updates is limited. For instance, this is the case of MOEA/D-DE (Li and Zhang, 2008), which is winner of several optimization competitions; MOEA / D-DE provides important advances by incorporating DE operators, polynomial mutation, a dynamic computational resource allocation strategy, mating restrictions, and a modified replacement operator to prevent excessive replication of individuals. Note that the best individual for each weighted function is always preserved, so MOEA/D is an elitist algorithm. The principle that guides the definition of neighborhoods is that, usually, close subproblems are properly solved by close solutions; thus, MOEA/D promotes the mating of close solutions, resulting in further intensification.

Given that some Pareto front forms may pose challenges when applying evenly distributed weights, some methods have been developed to automatically adapt the weights and improve diversity in the objective space (Qi et al., 2014). Additionally, some authors have noted that close subproblems do not necessarily induce close solutions in the search space. Thus, in MOEA/D-AMS (Chiang and Lai, 2011), the neighborhood is defined in terms of the space of the variables. In any case, the principle is the same because this restriction favors the mating of similar individuals.

2.2.3 Indicator-Based MOEAs

In order to compare the performance of MOEAs, several quality indicators that map Pareto set approximations to real numbers have been devised. Since these indicators measure the quality of the approximations attained by MOEAs, a paradigm based on the application of these indicators was proposed. In these cases, instead of the Pareto dominance, indicators are used in the MOEAs to guide the optimization process. In particular, the parent selection and replacement phase are usually modified by incorporating the use of an indicator. Among the different indicators, hypervolume is a widely accepted Pareto-compliant quality indicator. One of the advantages of indicator-based schemes is that the indicators usually take into account both the quality of the candidate solutions and their diversity. Among the different indicator-based MOEAs, SMS-EMOA (Beume et al., 2007) has been used extensively, probably due to its simplicity and superiority over many other approaches (Gómez et al., 2016). This algorithm might be considered as a hybrid of indicator-based and domination-based because it integrates the non-dominated sorting with the use of the hypervolume metric. In particular, the replacement phase erases the individual of the worst-ranked front with the minimum contribution to the hypervolume. Thus, SMS-EMOA uses the hypervolume as a density estimator. Since calculating hypervolume is a computationally expensive task, some faster variants, such as the Simple and Fast Hypervolume Indicator-based MOEA (Jiang et al., 2014) have been proposed. Other popular alternatives belonging to this group are based on the R2 indicator, as is the case of the R2-Indicator-Based Evolutionary Multi-objective Algorithm (R2-EMOA) (Trautmann et al., 2013). R2-EMOA follows the same principle as SMS-EMOA, but instead of computing the hypervolume, it applies the R2-Indicator. Note that since it considers the R2-Indicator, it requires a set of weight vectors, which should satisfy the preferences of the Decision Maker.

2.2.4 A Failure Mode of Multi-Objective Evolutionary Algorithms

In this section, an analysis that exposes a drawback related to premature convergence in multi-objective optimization is presented. Note that the results of this section were described in (Castillo et al., 2017). This analysis confirms the negative effect of the issue addressed in this thesis in the multi-objective optimization case.

In the last decades, a large number of MOEAs have been devised. Two aims are usually considered: the solutions should be close to the Pareto front, and the diversity of the approximation — measured in the objective space — should be maximized (Zitzler et al., 2004). In spite of the large number of methods, some well-known benchmarks such as the WFG tests are not optimally solved. As mentioned previously, in the case of single-objective optimization, it has been found experimentally that integrating rules to promote diversity in the variable space can bring benefits in terms of the objective function. The main reason behind this finding is that EAs tend to quickly lose diversity, so premature convergence might appear, meaning that many resources might be wasted. Recently, a new design principle has been successfully used to obtain new best-known solutions to several well-known single-objective problems (Segura et al., 2015c). Note that in

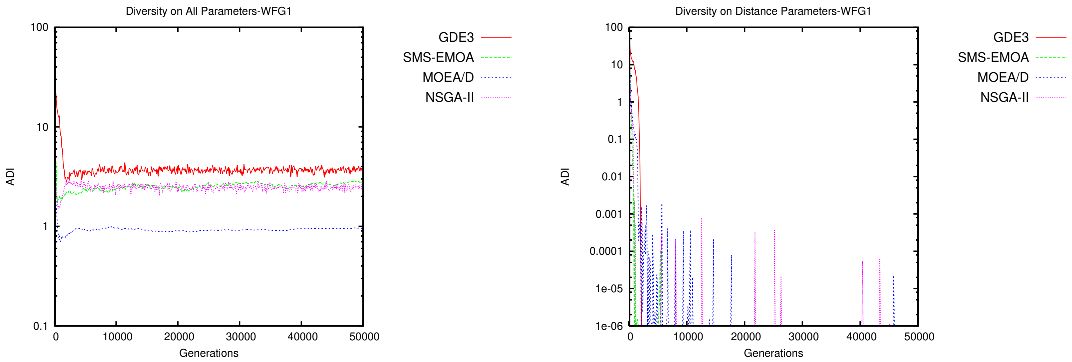

Fig. 2.1 Evolution of diversity for WFG1

MOEAs, complete convergence does not appear because some degree of diversity is explicitly maintained in the multi-objective space. However, the total amount of diversity maintained in the variable space might be too low. In such cases, a situation similar to premature convergence might appear with MOPs, i.e. the diversity might not be large enough to further improve the results.

Several benchmarks to study the performance of MOEAs have been proposed (Huband et al., 2006). Among them, the WFG benchmark (Huband et al., 2005) is one of the most widely accepted ones. As a result, it has been selected to perform our studies. The WFG problems divide the decision variables into two kinds of parameters: the distance parameters and the position parameters. A parameter x_i is a distance parameter when for all parameter of $\vec{x}$, modifying x_i in $\vec{x}$ results in a parameter vector that dominates $\vec{x}$, is equivalent to $\vec{x}$, or is dominated by $\vec{x}$. However, if x_i is a position parameter, modifying x_i in $\vec{x}$ always results in a vector that is incomparable or equivalent to $\vec{x}$ (Huband et al., 2005).

This section shows that state-of-the-art MOEAs do not always maintain a high enough diversity. In particular, a problem is used to show that premature convergence appears in the set of distance parameters. As a result, the crossover loses its exploratory strength. To illustrate the previously mentioned drawback, the WFG1 test has been selected. In this analysis, WFG1 is selected because it has a simple definition, but most current MOEAs face difficulties with it. In fact, WFG1 is an unimodal and separable problem. The distance parameter values associated with the Pareto optimal solutions for WFG1 (Huband et al., 2005) have exactly the same values in the distance parameters. This value is shown in Eqn. (2.1).

$$x_{i=k+1:n} = 2i \times 0.35 \tag{2.1}$$

The jMetalcpp (López-Camacho et al., 2014) framework was used to perform our executions. Taking into account the stochastic behavior of MOEAs, 35 independent executions were run. In all of them, the stopping criterion was established equal to $50,000$ generations and the population size equal to 250. In order to analyze the diversity, the average Euclidean distance among individuals (ADI) is calculated, i.e. the mean value of all pair-wise distances among individuals in the population is reported. Fig. 2.1 shows the evolution of the diversity for GDE3, SMS-EMOA, MOEA/D and NSGA-II. The top part of the figure shows the evolution of the ADI taking into account the whole set of parameters, whereas in the bottom part only the distance parameters are taken into account. Note that a logarithmic scale is used; in the case of the distance parameters, the regions where no information is plotted correspond to generations where the distance is zero. As we can see, all methods maintain some degree of diversity when all the parameters are considered. Otherwise, the diversity in

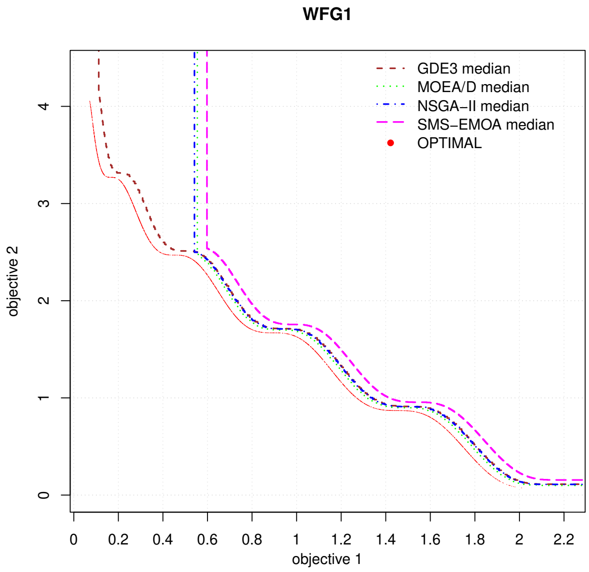

Fig. 2.2 50% Attainment surfaces archieved for WFG1

the objective space would be lost. However, when the distance parameters are the only ones considered, all the methods reach a zero distance relatively fast. In fact, after about 5,000 generations, all of them have reached a generation where the distance is equal to zero. This means that in relatively few generations, all the methods have converged in the distance parameters. We can see that in some generations, some degree of diversity is recovered; this is because of the action of the mutation. Thus, one of the reasons for the poor performance of some state-of-the-art MOEAs in this case is the apperance of premature convergence in the distance parameters. In fact, after this loss of diversity, the MOEAs are basically modifying the position parameters, so most of the time is spent in improving the diversity in the objective space instead of the quality. Figure 2.2 shows the attainment surface reached by these methods as well as the Pareto front. It is clear that there is a large room for improvement for this problem.

2.3 Exploration and Exploitation in Meta-heuristics

Population-based meta-heuristics might achieve good results with an adequate trade-off of exploration and exploitation (Črepinšek et al., 2013; Eiben and Schippers, 1998; Xu and Zhang, 2014). In the literature, there are many works that attempt to achieve a proper balance between exploration and exploitation with the consideration of indirect measures; however, there are few works that propose direct measures (Jerebic et al., 2021). Perhaps, the reason for this is that there is not a direct measure of exploration and exploitation widely accepted by the community, which is still an open question in population-based algorithms (Beyer and Deb, 2001).

One of the most popular works that defines exploration and exploitation is presented in Črepinšek et al. (2013), which is "*Exploration is the process of visiting entirely new regions of a search space, whilst exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points*". This work proposes a direct measure of exploration and exploitation; however, it was probably not adopted in the community, as it proposes a threshold value that defines the boundary of the neighborhood of the closest neighbor. In particular, it defines a fixed threshold to delimit exploration from exploitation. This definition is problem-dependent and might fail with several kinds of problems (e.g. asymmetrical landscapes). Later, the same authors (Jerebic et al., 2021) proposed a more accurate technique that no longer requires a threshold value. In particular, it computes the attraction basins by an approximation method (also known as surrogate model) for every problem. Although this new technique showed somehow better features than the previous one, it might be computationally expensive for high-dimensional problems and/or highly irregular functions (e.g. biased functions).

Similarly, Morales-Castañeda et al. (2020) defines exploration and exploitation as: "*Exploration refers to the ability of a search algorithm to discover a diverse assortment of solutions, spread within different regions of the search space. On the other hand, exploitation emphasizes the idea of intensifying the search process over promising regions of the solution space, with the aim of finding better solutions or improving existing ones*". Although the concepts of exploration and exploitation can be intuitive to most readers, in the literature there is not a popular and formal definition.

A recent and direct measure of exploration and exploitation is based on the notion of attraction basins (Chen et al., 2019; Gonzalez-Fernandez and Chen, 2015). A formal definition is that an attraction basin $A(x^*)$ is a part of a search space S with a local optimum x^* called an attractor, which can be accessed from any other point $y \in A(x^*)$ in this attraction basin simply by performing a local search (Chen et al., 2019; Jerebic et al., 2021; Locatelli and Wood, 2005). In other words, an attraction basin is a sub-region of the feasible region with a point called the "attractor" to which a process converges. An attraction basin is also defined as "*A region of the search space containing all of the solutions that lead to a particular local optimum when (greedy) local search is used*" (Gonzalez-Fernandez and Chen, 2015). Therefore, a multi-modal problem is composed by several attraction basins and its attractors are their local(global)-optimal points. Thus, exploration can be defined as the process of identifying different attraction basins, i.e., creating a new individual that belongs to a different attraction basin than its parents. Conversely, exploitation can be defined as the task of finding the local optimum within a basin, i.e., creating a new individual that belongs to the same attraction basin as its parent.

It is well known that in population-based algorithms, diversity influences exploration and exploitation in the search. In fact, some works use diversity as an indirect measure of exploration and exploitation. That is, if diversity is high, then the population is in a phase of exploration. Contrarily, if diversity is low, then the population is in a phase of exploitation (Squillero and Tonda, 2016). Several works claim that diversity is the key to adapt the search from exploration to exploitation (Gabor and Belzner, 2017; Squillero and Tonda, 2016; Ursem, 2002), Nevertheless, the way in which diversity is measured and the ability to distinguish between phases is problem-dependent, resulting in a problematic situation. For example, in Ursem (2002) the well-known distance-to-average point is applied to alternate between phases of exploration (mutation) and phases of exploitation (recombination and selection). Thus, diversity is measured based on the distance between individuals and the average individual. In a more recent work (Morales-Castañeda et al., 2020) a dimension-wise measurement is considered to assess the balance. These authors considered several meta-heuristics and in their results the balance that produced the best results was about 90% of exploitation and 10% of exploration. They also concluded that good performance is produced through the combination of competitive search mechanisms (e.g. crossover and selection) and an ideal balance response adopted to the problem.

2.3.1 Classification of Strategies for the Promotion of Diversity

The proper balance between exploration and exploitation is one of the keys to designing a successful EAs. At least in single-objective optimization, there are several methods showing that the proper management of the diversity of the decision variable space is a way to achieve this balance, and as a consequence, a large number of diversity management techniques have been devised (Pandey et al., 2014). Specifically, these methods are classified according to the component(s) of the EA that is modified to alter the way in which diversity is maintained (Črepinšek et al., 2013). A popular classification defines *uniprocess*-driven and *multiprocess*-driven (Liu et al., 2009). In uniprocess-driven the diversity is individually managed by a unique process or unique operator (e.g. selection, population resizing, and mutation); therefore, there is no coordination between its components. Moreover, a unique process is focused on balancing exploration and exploitation by using different techniques (e.g. adapting control parameters). Note that this does not mean that other processes are uninvolved; instead, a unique process manages different techniques. For instance, in some works, selection controls the level of exploration/exploitation by altering the selection pressure (Back, 1994). Differently, in multiprocess driven, proper balance is achieved by the coordination of different components. Therefore, more than one process/operator can be selected to control the two aspects individually. For instance, a high selection pressure is applied to induce exploitation, and a mutation operator with a high mutation rate is biased to exploration, promoting exploration and exploration simultaneously at different stages. For its simplicity, uniprocess-driven techniques are quite common, and most of the proposals described in this thesis belong to this category. Note that Liu et al. (2009) describes more comprehensive studies and some classifications for diversity approaches.

Furthermore, Črepinšek et al. (2013) classifies the methods in *controlling*, *learning*, *maintaining*, and *other direct approaches*. All the algorithmic proposals presented in this thesis belong to *maintenance approach*. In these sort of methods, the diversity is maintained to achieve an ideal balance along the search. The main challenge of these strategies is to determine the amount of diversity that in some way helps to find the most promising regions, which is popularly known as *useful diversity*. Among maintenance approaches, non-niching techniques are divided as follows (Eiben et al., 1999): *population-based*, *selection-based*, *crossover/mutation-based*, and *hybrid*.

Recently, a new taxonomy based on methods that controls premature convergence has been presented (Segura et al., 2017a). Note that diversity is a prerequisite for exploration in order to avoid premature convergence (Črepinšek et al., 2013). In this taxonomy, the methods are grouped on the basis of the components that are modified. Thus, the schemes are grouped into the following: *selection-based*, *population-based*, *crossover/mutation-based*, *fitness-based*, and *replacement-based*.

Selection-based schemes are one of the most common strategies. They modify the pressure of parental selection to better explore the most promising regions. This is one of the most mature schemes to control premature convergence; however, these types of schemes are not capable of preserving diversity in long runs (Blickle and Thiele, 1996).

In the population-based schemes, the typical panmictic fixed population model is not used. This group contains strategies such as population resizing, duplicate management, infusion techniques, external archives, and island-based models, among others. Island-based schemes allow individuals to recombinate with a subset of the population, delaying convergence, but not avoiding it (Alba, 2005; Martin et al., 1997). Infusion techniques are also quite useful. In this case, new individuals are randomly inserted according to some rules (Salah et al., 2016).

In the crossover/mutation-based methods, the recombination can be altered to better explore the search space. This group also includes mating restrictions, which impose rules to recombine individuals. Note that the crossover and mutation operators in evolutionary algorithms are often seen as exploration and exploitation strategies, respectively. However, it is complicated to roughly classify crossover and mutation as exploration or exploitation operators, as their intended behavior could easily be altered by adjusting their respective rates (Morales-Castañeda et al., 2020).

In fitness-based schemes, the fitness of each individual is modified taking into account the features of the population. Some of the most popular strategies that belong to this category are fitness sharing (Mahfoud, 1995), clearing (Pétrowski, 1996), crowding, modified clearing (Singh and Deb, 2006), and clustering (Yin and Germay, 1993). These methods are frequently included in the niching algorithms (Sareni and Krahenbuhl, 1998). Note that niching methods promote the information and maintenance of stable sub-populations within the neighborhood of optimal solutions (Mahfoud, 1995).

Finally, replacement-based schemes follow the principle of inducing higher levels of exploration by diversifying the survivor of the population (Segura et al., 2015c). The most mature strategies in this category include deterministic crowding (Mahfoud et al., 1992), probabilistic crowding (Mengshoel and Goldberg, 1999), adaptive generalized crowding (Mengshoel et al., 2014), and restricted tournament selection (Harik et al., 1995). Some quite distinct strategies consider multi-objectivization (Segura et al., 2013a), combined objectives (Vidal et al., 2013), and replacement restrictions (Lozano et al., 2008).

2.3.2 Diversity Management Techniques: A Classical Approach

In the literature, it is widely known that achieving a balance between exploration and exploitation in meta-heuristics is of primary importance (Črepinšek et al., 2013; Eiben and Schippers, 1998; Xu and Zhang, 2014). In particular, premature convergence is one of the most important drawbacks related to the pour management of this balance. Premature convergence is a well-known drawback that is described in Segura et al. (2015c) as "*Premature convergence arises when all the population members are located in a reduced part of the search space –different from the optimal region– and the components selected do not allow escaping from such a region*". One way to avoid premature convergence is to alter the management of diversity along the optimization process (Črepinšek et al., 2013). Note that many strategies have been designed in an effort to avoid this drawback. This section describes the most relevant works that motivated the development of the proposals presented in this thesis. For more information, the reader is invited to review Črepinšek et al. (2013); Lozano et al. (2008); Mahfoud et al. (1992); Morales-Castañeda et al. (2020); Pandey et al. (2014); Squillero and Tonda (2016).

Most of the early studies on evolutionary algorithms involved generational approaches, in which a population of parents produces a new population, which replaces the previous population regardless of their fitness (De Jong, 2016). Later, EAs were improved by maintaining some of the best individuals in the current population through elitist policies. This modification shifts the balance towards more exploitation and less exploration (De Jong, 1975a). Therefore, these strategies integrate a new component, which is the survivor selection strategy to determine which individuals survive for the next generation.

The first attempts to alter the balance between exploration and exploitation were made with selection-based schemes. Initially, several studies analyzed the loss of diversity for the most classic schemes. They showed that most initial parent selection schemes lead to fast convergence (Blickle and Thiele, 1996). Therefore, the classical selection schemes were modified to alter the balance between exploration and exploitation with the inclusion of several unusual selection rules (Chen et al., 2009). However, parent selection schemes have not been very successful in this area.

Population-based schemes have also been proposed to improve the preservation of diversity. Specifically, in island-based models, individuals have extensive periods of isolated evolution that simplifies its parallelization and promotes large levels of diversity (Martin et al., 1997). Since diversity is implicitly promoted in these methods, the maintenance of it for long runs is not guaranteed (Alba, 2005). These and more inconveniences are mitigated in Wineberg and Chen (2004), which includes a mechanism to manage dynamic population sizes along with population restarts. However, the parameterization needs to be adjusted, which might be quite complicated for some problems.

An alternative strategy to further preserve diversity along the run is to impose restrictions on the individuals to be recombined. These schemes are better known as matting restrictions and prevent the recombination of some individuals depending on their features. Most existing approaches of mating restriction are based on the similarity of the individuals (Galán et al., 2013). The principle that guides this kind of schemes is that recombining distant solutions results in a larger exploration and recombining close solutions results in further intensification (Castillo et al., 2017). The ability to explore multiple regions along the optimization process with these schemes has led to the improvement of multi-modal optimization (Deb and Goldberg, 1989). Note that mating restrictions have been used to delay or accelerate convergence, but premature convergence is not entirely avoided.

Variation operators also affect the ratio between exploration and exploitation, especially with the control of its parameters. The logic is that high crossover and, especially, mutation rates promote more exploration than low rates. Some authors argue that very low rates might mimic a local search (Črepinšek et al., 2013). The process of selecting the parameterization of the operators can be done in several ways. Usually, they are adopted considering the history, and in many cases self-adaptive is used (Qin et al., 2008). In addition, in some strategies the parameters are adaptively controlled according to the optimization states. For example, to promote fast convergence and population diversity simultaneously, a two-level parameter adaptation strategy was proposed (Yu et al., 2013).

Restarting schemes are quite popular for promoting diversity in the search. In these schemes, a rule triggers a partial or complete restart of the population, directly increasing the diversity. These methods aim to perform a more global exploration of the search space and are designed to avoid being trapped in local optima (Tsutsui et al., 1997). Note that the rule and the procedure to generate new solutions should be carefully defined. However, these schemes can be implemented in simple ways and can be easily combined with many different strategies (Koumousis and Katsaras, 2006).

2.3.3 Diversity in Replacement-Based Schemes: Classic and Modern Approaches

The introduction of elitist policies in EAs gave rise to the inclusion of replacement-based methods to alter the management of diversity. The principle of these methods is to control the level of diversity in successive generations by diversifying the survivors of the population (Segura et al., 2015c). In the first years, replacement-based methods were used mainly to deal with multi-modal optimization. In this kind of optimization, several promising regions must be maintained and explored simultaneously. Since most generational methods face difficulties with multi-modal problems, replacement-based methods have been quite successful. Note that in classic schemes, if multiple solutions in the current population have identical good fitness values, then, over generations, the population will tend to converge to one of these solutions. This behavior is termed *genetic drift*, which occurs due to cumulative sampling errors in the expected selection rates originated from the use of a finite population (Purshouse, 2003). Two classic niching methods that are used to improve this situation are *sharing* (Goldberg et al., 1987) and *crowding* (De Jong, 1975b). In the sharing method, the fitness of the solution is degraded if it is within a certain distance from other solutions. In the *crowding* method, the basic principle is to remove individuals from the population that are similar to those offspring that are selected to survive (De Jong, 1975b; Mahfoud et al., 1992)[3]. Thus, *crowding* promotes the replacement

[3] For more information of niche methods the reader is invited to review the PhD thesis of Samir W. Mahfoud Mahfoud (1995)

of individuals that are similar to those accepted. This concept can be carried out in several ways. Some well-known methods are *deterministic crowding* (Mahfoud, 1995), *probabilistic crowding* (Mengshoel and Goldberg, 1999), *elitist recombination* (Thierens, 1998) and *restricted tournament selection* (Harik et al., 1995). Several studies suggested that replacing parents arbitrarily or replacing only the worst parent (as Cavicchio's preselection does) results in too many replacement errors to maintain niching. To alleviate this shortcoming, the *deterministic crowding* was proposed. In this method, all elements of the population are randomly paired, without replacement. At each time, it processes two parents and two offspring: each child replaces the nearest parent if it has better fitness. Since this strategy might be too greedy, a probabilistic extension was proposed. In *probabilistic crowding* the winner of the parent and offspring tournament is chosen by using a probability proportional to the fitness. Note that this extension follows a principle similar to that presented in the simulated annealing method (Bertsimas and Tsitsiklis, 1993). Similarly, in *elitist recombination*, the best pair of both parents and offspring survives to the next generation.

Another classical and effective method is the *Restricted Tournament Selection* (RTS) (Harik et al., 1995). This method is based on the paradigm of local competition. The motivation behind RTS is to avoid fast convergence caused by the binary tournament selection. To remedy this in RTS after creating a new individual (C), CF individuals from the current population are randomly selected (without replacement). Then C and its most similar individuals –from those in the selected set– compete for a place in the population using a traditional binary tournament. This form of tournament aims to avoid the competition between very different elements. Note that the creation of individuals is done in a steady state manner.

Other strategies are based on considering diversity to calculate a fitness value for the replacement stage. Specifically, in the *Hybrid Genetic Search with Adaptive Diversity Control* (HGSADC) (Vidal et al., 2013), the individuals — union of parents and offspring — are sorted by their contribution to diversity and by their original cost. In order to calculate the contribution to diversity of an individual, its mean distance to the closest N_{Close} individuals is calculated. For an individual I, the ranking in terms of diversity is denoted as $RD(I)$, whereas the ranking for the original cost is denoted as $RC(I)$. Then, the rankings are combined to generate the biased fitness value $BF(I)$ using Eqn. (2.2). In each step of the replacement phase, the individual with the lowest biased fitness is erased, and the ranks are recalculated. This process is carried out until the desired population size is attained. It is important to note that this scheme requires the setting of parameters: N_{Close} and N_{Elite}. Finally, note that $N_{population}$ refers to the number of individuals that have not yet been erased by the replacement scheme.

$$BF(I) = RC(I) + \left(1 - \frac{N_{Elite}}{N_{population}}\right) RD(I) \tag{2.2}$$

Another approach, related to fitness sharing, is the *Clearing Procedure* (CLR) (Pétrowski, 1996) which is a classical niching method (Holland, 1992). CLR uses a dissimilarity measure between individuals to determine if they belong to the same subpopulation or not. The procedure of CLR works as follows. First, the population is sorted according to its fitness. Then, a dominant solution –the one with the best fitness– is identified as a niche. Thereafter, the individuals whose distance to this dominant solution is less than a *clearing radius* are then associated with this niche. This process is repeated until all individuals belong to a niche. The main difference between CLR and fitness sharing is that while the former maintains a number of winners in each niche (elitism), the latter assigns a proportional fitness to the individuals of each niche. Note that for highly multi-modal problems all the individuals might be dominant leading to stagnation. To alleviate this, in some variants dominant solutions should also be better that the mean fitness. Note that the principle imposed by CLR is also used with some modifications in more modern strategies (Yang et al., 2016). For instance, in *speciation* the population is divided into several species (Li, 2005). Each species is identified by a seed and a radius of the niche. Note that currently *crowding* and *speciation* are two of the most popular strategies for multi-modal optimization.

A quite different approach is the *Contribution-Diversity Replacement-Worst* (CD/RW) (Lozano et al., 2008). CD/RW is another steady-state algorithm in which a new individual replaces another one that is worse both in diversity contribution and fitness. If there is no other individual that is the worst in both criteria, then the individual with the worst fitness is erased.

Recently, multi-objectivization methods have gained popularity. These methods transform a single-objective optimization problem into a multi-objective optimization problem by adding an artificial objective function (Bui et al., 2005). This additional objective might be aimed to maintain diversity. A recent variant that uses this principle is the *Replacement with Multiobjective-Based Dynamic Diversity Control Strategy* (RMDDC) scheme (a.k.a. *multi-dynamic*) (Segura et al., 2015c, 2013b). RMDDC can be considered one of the most effective long-term methods. One of the main differences between RMDDC and the classic multi-objectivization techniques is that RMDDC applies the principle of adapting the degree of exploration and exploitation to the requirements of the different optimization stages by taking into account the stopping criterion set by the user. Specifically, a minimum desired contribution to diversity (D) is dynamically calculated. In particular, the initial D is calculated with an initial distance factor (D_I). Then, D is linearly reduced so that at the end of the execution it is equal to zero. The replacement phase is based on an iterative process to select all survivors. At each iteration, among the individuals with a diversity contribution larger than D, one is selected considering both its contribution to diversity and its quality. Particularly, these two values are considered as the two objectives to optimize, and the non-dominated set is calculated. Then, an individual of the non-dominated set is selected randomly. In this way, the decision taken in the replacement-phase considers the stopping criterion and elapsed execution period. Note that RMDDC has been reported to be superior to the state-of-the-art in the long term for several single-objective combinatorial optimization problems. In fact, it has allowed to find new best-known solutions in several of these problems.

Finally, the replacement *Best-Non Penalized* (BNP) was proposed (Romero Ruiz and Segura, 2018). BNP, which is also a survivor selection strategy, can be considered as a successor to RMDDC. One of the principles of BNP is to avoid the selection of too close individuals. Specifically, the approach attempts to avoid the selection of pairs of individuals that are closer than a threshold (D). Similarly, in RMDDC, a threshold (D) is linearly decreased so that it reaches zero at the end of the execution. The main difference between BNP and RMDDC is that the former method has an additional bias toward high-quality solutions and the latter method gives the same importance to quality and diversity. Since RMDDC considers a multi-objective selection scheme, it has a less aggressive performance in terms of fitness. Intuition suggests that BNP might be better than RMDDC for shorter runs, nevertheless there are no clear results that confirms it; both algorithms have allowed to reach new best-known solutions for different problems, so both are considered to be quite competitive replacement methods for long runs. Note that BNP is simpler and allows for a reduction of the computational cost.

2.3.4 Diversity in Differential Evolution

Differential Evolution (DE) is highly susceptible to the loss of diversity, partially due to the greedy strategy applied in the replacement phase. Thus, several analyses to better deal with this issue have been carried out. Since the general implications of each DE parameter on diversity are known, one of the alternatives is to theoretically estimate the appropriate values for the DE parameters (Zaharie, 2003). On the contrary, some analyses on the effects of the magnitude of the difference vectors used in the mutation have also been performed (Montgomery, 2009). Such analyses and additional empirical studies regarding the crossover allowed to conclude that some kind of movements should be disallowed to delay the convergence (Montgomery and Chen, 2012). In this last study, the kind of accepted movements varies along the run. Specifically, it discards movements with a norm below a threshold, and this threshold decreases taking into account the elapsed generations. Other ways to alter the kind of accepted movements have been proposed (Bolufé-Röhler et al., 2013). Note that these types of methods have similarities with the proposals addressed in this thesis in the sense that decisions depend on the number of elapsed generations. However, this thesis focuses on replacement operators rather than mutation phases. Moreover, these methods do not explicitly consider the differences that appear in the entire population. Instead, restrictions apply to the differences that appear in the reproduction phase.

A different alternative operates by altering the selection operator (Sá et al., 2008). In particular, the selection pressure is decreased through probabilistic selection to maintain population diversity and consequently to allow escape from the basin of attraction of local optima. Since it considers the fitness to establish the acceptance probabilities, it is very sensitive to scale transformations. In this case, the decisions are not biased by the elapsed generations. Finally, in the *Auto-Enhanced Population Diversity* (AEPD), the diversity is explicitly measured and it triggers a mechanism to diversify the population when a too low diversity is detected (Yang et al., 2014). Strategies with similar principles but with different disturbance schemes have also been devised (Zhao et al., 2016).

Note that DE variants with best performance in competitions do not apply these modifications and that most of these extensions have not been implemented in the most widely used optimization frameworks. As a result, these extensions are not so widely used in the community in spite of their important benefits for some cases. This is probably due to the fact that these schemes only provide benefits in quite long runs.

2.3.5 Diversity in the Decision Variable Space in MOEAs: A Brief Review

Despite the vast amount of MOEAs proposed, none of the state-of-the-art techniques explicitly manage diversity in the variable space, which is a notable difference with respect to single-objective optimizers. One of the main reasons behind this difference is probably that there is a relationship between the diversity maintained in the objective space and the diversity maintained in the variable space, so even if the diversity in the variable space is not managed explicitly, any negative effect that do appear are usually not as bad as in single-objective optimization. However, the relationship between the two different diversities depends on each MOP (Shir et al., 2009), meaning that including the successful principles of single-objective optimizers might result in more robust MOEAs. Most current multi-objective algorithms that take into account the diversity in the variable space are devoted to multi-modal multi-objective optimization (Cuate and Schütze, 2019; Deb and Tiwari, 2008). These algorithms are better known as Evolutionary Multi-modal Multi-objective Algorithms (EMMA) and are designed to identify high-quality solutions that are diverse both in the objective space and in the decision variable space. However, some attempts to apply these mechanisms in traditional multi-objective optimization have also been made.

The Non-Dominated Sorting Genetic Algorithm (NSGA) developed in 1994 (Srinivas and Deb, 1994) was one of the first MOEAs to employ diversity in the variable space. Specifically, it relies on fitness sharing to discriminate between solutions on the same front. However, this decision came at the cost of not considering the diversity on the objective space. Note that this method was designed in an opposite way to current methods: the diversity in the variable space is considered, but at the cost of disregarding the information on the diversity in the objective space. The performance of NSGA is not even close to that of the current MOEAs, and one of the reasons is precisely that it does not consider the diversity in the objective space.

In 2003, the GDEA (Toffolo and Benini, 2003) proposed by Toffolo and Benini integrated diversity into the search as an additional objective, which modifies the ranking of the individuals and favors maintaining distant individuals during the entire optimization process. In 2005, Chan and Ray (Chan and Ray, 2005) proposed the application of selection operators to encourage the preservation of distant solutions in both the objective and variable space. Later, Deb and Tiwari proposed the Omni-optimizer (Deb and Tiwari, 2008), which was designed as a generic multi-objective, multi-optima optimizer, which extends the original idea of NSGA. In the multi-objective case, it is an extension of NSGA-II, where the crowding distance considers both the objective and the variable space. Since it first uses the typical rank procedure considering only the objective space, more importance is given to this space, and diversity plays an inferior role. Unfortunately, there is no way to easily alter the importance given to each space. In 2009, Shir et al. showed that in many cases the diversity in the variable space can be significantly enhanced without hampering the convergence to a diverse Pareto front approximation. Following this insight, the CMA-ES niching framework was proposed (Shir et al., 2009). Estimation of Distribution Algorithms have also considered the information in the variable space to achieve better approximations of the Pareto set (Zhou et al., 2009). Furthermore, the Diversity Integrating Hypervolume-Based Search Algorithm (DIVA) (Ulrich et al., 2010) weights the contribution of the hypervolume against the diversity in the variable space. The application of mating restrictions during the selection phase has also been used to indirectly alter the amount of diversity (Castillo et al., 2017; Chiang and Lai, 2011).

Note that of the methods discussed, the only one that shows any improvement in terms of objective-space metrics is GDEA. However, the results attained by GDEA are not as competitive as those attained by modern solvers, so it is not currently considered as a state-of-the-art MOEA, but rather as an easily applicable and general approach. The remaining methods focus on showing that solutions with a higher level of diversity in the variable space can be obtained without an overly negative effect on the approximation of the Pareto front.

Finally, some methods alter several components simultaneously (Liu et al., 2019; Shi et al., 2019; Zadorojniy et al., 2012). Among them, the Convergence Penalized Density EA (CPDEA) (Liu et al., 2019) is one of the most recent and effective approaches. In this case, two different variation operators are used. The first-one is devoted to exploration and is applied in the first half of the run, whereas the second one is applied in the second half and is devoted to intensification. Moreover, the replacement considers both quality and diversity in the objective space, and the density in decision variable space.

In light of the results of the approaches described above, it is clear that considering the diversity of the decision variable space in the design phase yields benefits for decision makers, since the final solutions obtained by these methods exhibit higher diversity of the decision variable space than those obtained by traditional approaches (Deb and Tiwari, 2008; Rudolph et al., 2007). Thus, while clear improvements are obtained when metrics related to the decision variable space are taken into account, the benefits in terms of the objective function space are not as clear, and until now, they have only been achieved in certain specific MOPs (Liu et al., 2019). We claim that one of the reasons for this behavior might be that the diversity of the decision variable space is considered in the entire optimization process. However, in a similar way to the single-objective domain, reducing the importance allocated to the diversity of the decision variable space as generations progress (Segura et al., 2015c) might be truly important in obtaining better approximations of the Pareto front. Currently, no MOEA considers this idea, and this was the motivation for the design of the novel MOEAs proposed in this thesis.

Part II

Proposals and Achievements in Single-Objective Optimization

Chapter 3

Differential Evolution with Enhanced Diversity Maintenance

Abstract

Differential Evolution (DE) is a popular population-based meta-heuristic that has been successfully used in complex optimization problems. Premature convergence is one of the most important drawbacks that affects its performance. In this chapter, a novel replacement strategy that combines the use of an elite population and a mechanism to preserve diversity explicitly is devised. The proposal is integrated with DE to generate the DE *with Enhanced Diversity Maintenance* (DE-EDM). The main novelty is the use of a dynamic balance between exploration and exploitation to adapt the optimizer to the requirements of the different optimization stages. Experimental validation is carried out on several benchmark tests proposed in competitions of the well-known IEEE Congress on Evolutionary Computation. The top-rank algorithms of each competition, as well as other diversity-based schemes, are used to illustrate the usefulness of the proposal. The new method avoids premature convergence and significantly improves the results obtained by state-of-the-art algorithms.

3.1 Introduction

Evolutionary Algorithms (EAs) are one of the most widely used techniques to deal with complex optimization problems. Several variants of these strategies have been devised (Talbi, 2009) and applied in many fields, such as in science, economics, and engineering (Chakraborty, 2008). Among them, Differential Evolution (DE) (Storn and Price, 1997) is one of the most effective strategies to deal with continuous optimization. In fact, it has been the winning strategy of several optimization competitions (Das and Suganthan, 2011). Similarly to other EAs, DE is inspired by the natural evolution process and it involves the application of mutation, recombination, and selection. The main peculiarity of DE is that it considers the differences among the vectors present in the population to explore the search space. In this sense, it is similar to the Nelder-Mead (Nelder and Mead, 1965) and the Controlled Random Search (CRS) (Price, 1983) optimizers.

Despite the effectiveness of DE, there exist several weaknesses that have been detected and partially solved by extending the standard variant (Das and Suganthan, 2011; Locatelli and Vasile, 2015). Among them, the sensitivity to its parameters (Zhang and Sanderson, 2009), the appearance of stagnation due to the reduced exploration capabilities (Lampinen et al., 2000; Sá et al., 2008) and premature convergence (Zaharie, 2003) are some of the most well-known issues. This last issue is addressed in this chapter. Note that, attending to the proper design of population-based meta-heuristics (Talbi, 2009), special attention must be paid to attain a proper balance between exploration and exploitation. A too large degree of exploration prevents proper intensification of the best-located regions, usually resulting in a too slow convergence. Differently, an excessive exploitation degree provokes a loss of diversity, meaning that only a limited number of regions are sampled. In the case of DE, since its inception, some criticism appeared because of its incapability to maintain a large enough diversity due to the use of a selection with high pressure (Sá et al., 2008). Thus, several extensions of DE to deal with premature convergence have been devised. For instance, parameter adaptation (Zaharie, 2003), auto-enhanced population diversity (Yang et al., 2015) and selection strategies with a lower selection pressure (Sá et al., 2008). Some of the last studies on the design of population-based meta-heuristics (Črepinšek et al., 2013) show that explicitly controlling the diversity to properly balance the exploration and intensification degree is particularly useful. Specifically, in the field of combinatorial optimization, some novel replacement strategies that dynamically alter the balance between exploration and exploitation have appeared (Segura et al., 2015c). The main principle of such proposals is to use the stopping criterion and elapsed generations to bias the decisions taken by the optimizers with the aim of promoting exploration in the initial stages and exploitation in the last ones. Probably their main weakness is that the time required to obtain high-quality solutions increases. Our novel proposal, which is called DE *with Enhanced Diversity Maintenance* (DE-EDM), integrates a similar principle in DE.

The rest of this chapter is organized as follows. First, the proper definition of the kind of single-objective problems and basic concepts of Differential Evolution are described in Section 3.2. Our proposal is described in Section 3.3. The experimental validation, which includes comparisons with state-of-the-art approaches, and the parameter analysis of our proposal are detailed in Section 3.4. Note that this validation also includes two popular replacement strategies. Finally, a summary of the most important findings is given in Section 3.5.

3.2 Differential Evolution: Basic Concepts

This section is devoted to summarize the classic DE variant and to introduce some of the most important terms used in the DE field. The classic DE scheme is called the DE/rand/1/bin, which has been extensively used to generate more complex DE variants (Das and Suganthan, 2011). DE was originally proposed as a direct search method for single-objective continuous optimization. The proper definition of a single-objective optimization problem addressed in DE is formally presented as follows.

A continuous box-constrained minimization problem, which is the kind of problem addressed in this thesis, is formally defined in Definition 3.2.1, where n corresponds to the dimensionality of the decision variable space, $\vec{x}$ is a vector of n decision variables $\vec{x} = (x_1, ..., x_n) \in \mathbb{R}^n$, which are constrained by $\vec{x}^{(L)}$ and $\vec{x}^{(U)}$, that is, the lower bound and the upper bound. The feasible space bounded by $\vec{x}^{(L)}$ and $\vec{x}^{(U)}$ is denoted by Ω. Each solution is mapped to the objective space with the function $F : \Omega \rightarrow \mathbb{R}$.

Definition 3.2.1 (Single-Objective Optimization Problem). *Minimize a function f with respect to a vector of variables $\vec{x} = (x_1, x_2, ..., x_n)$ in a universe Ω, i.e.,*

$$\begin{aligned} \min \quad & f(\vec{x}) \\ \text{subject to} \quad & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, ..., n. \end{aligned}$$

DE is a population-based stochastic algorithm, so it iteratively evolves a multi-set of candidate solutions. In DE, such candidate solutions are usually called vectors. In the basic DE variant for each member of the population — they are called *target vectors* — a new *mutant vector* is created. Then, the mutant vector is combined with the target vector to generate a *trial vector*. Finally, a selection phase is applied to choose the survivors. In this way, several generations are evolved until a stopping criterion is reached. The i-th vector of the population at generation G is denoted $\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, ..., x_{n,i,G}]$. In the following, more details are given for each component of DE.

Initialization

DE usually starts the optimization process with a randomly initiated population of N vectors. Since there is commonly no information about the performance of different regions, uniform random generators are usually applied. Therefore, the j -th component of the i-th vector is initialized as $x_{j,i,0} = x_j^{(L)} + rand_{i,j}[0,1](x_j^{(U)} - x_j^{(L)})$, where $rand_{i,j}[0,1]$ is a uniformly distributed random number lying between 0 and 1.

Mutation

For each target vector a mutant vector is created. Several ways of performing such a process have been proposed. In the classic DE variant the rand/1 strategy is applied. In this case, the mutant vector $V_{i,G}$ is created as follows:

$$\vec{V}_{i,G} = \vec{X}_{r_1,G} + F \times (\vec{X}_{r_2,G} - \vec{X}_{r_3,G}) \quad r_1 \neq r_2 \neq r_3 \tag{3.1}$$

The indices $r_1, r_2, r_3 \in [1, N]$ are mutually different integers randomly chosen from the range $[1, N]$. In addition, they are all different from the index i. It is important to take into account that the difference between the vectors is scaled with the number F, which is usually defined in the interval $[0.4, 1]$. The scaled difference is added to a third vector, meaning that when diversity decreases and consequently differences are low, mutant vectors are similar to target vectors. As a result, maintaining some degree of diversity is especially important in DE.

Crossover

In order to combine information of different candidate solutions and with the aim of increasing diversity, the crossover operator is applied. Specifically, each target vector $\vec{X}_{i,G}$ is mixed with its corresponding mutant vector $V_{i,G}$ to generate the trial vector $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, ..., u_{n,i,G}]$. The most typical crossover is the *binomial* one, which operates as follows:

$$\vec{U}_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if}(rand[0,1] \leq CR \quad or \quad j = j_{rand}) \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{3.2}$$

where $rand[0,1]$ is a uniformly distributed random number, j_{rand} is a randomly chosen index which ensures that $\vec{U}_{i,G}$ inherits at least one component from $\vec{V}_{i,G}$ and $CR \in [0,1]$ is the crossover rate.

Selection

Finally, a greedy selection is performed to determine the survivors of the next generation. Each trial vector is compared with its corresponding target vector and the best one survives:

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G}, & \text{if} \quad f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \\ \vec{X}_{i,G}, & \text{otherwise} \end{cases} \tag{3.3}$$

Therefore, each population member either improves or remains with the same objective value in each generation. Since members never deteriorate, it is considered to be a selection with high pressure. Note that in case of a tie, the trial vector survives.

3.3 Algorithmic Proposal

Our proposal is motivated by two main works in the area of control of diversity in EAs. The first is the empirical study developed by Montgomery et al. (Montgomery and Chen, 2012), which presents several empirical analyses that confirm issues related to premature convergence in DE. The second work, by Segura et al. (Segura et al., 2015c), provides significant improvements in the combinatorial optimization field by developing a novel replacement strategy called *Replacement with Multi-objective based Dynamic Diversity Control* (RMDDC) that relates the control of diversity with the stopping criterion and elapsed generations. Important benefits were attained by methods including RMDDC, so given the conclusions of these previous works, the proposal of this chapter is a novel DE variant that includes an explicit mechanism that follows some of the principles of RMDDC. Since this is applied in the context of DE, the parent population and offspring population are referred to as target vectors and trial vectors respectively, and the term vector is used with the same meaning as individual. This novel optimizer is called *Differential Evolution with Enhanced Diversity Maintenance* (DE-EDM) and its source code is freely available [1].

The core of DE-EDM (see Algorithm 1) is quite similar to the standard DE. In fact, the way of creating new trial vectors is not modified at all (lines 5 and 6). The novelty is the incorporation of elite vectors (E) and a novel diversity-based replacement strategy. The former records the vectors that have the best objective function value in relation to each target vector; therefore, there are N elite vectors, which must be considered as a multi-set since repeated individuals might appear. In order to select the members of the elite vectors, the original greedy replacement of DE is used (line 7). In the case of the replacement strategy (line 8), which is in charge of selecting the next target vectors, it follows the same principle that guided the design of RMDDC, i.e. vectors that contribute too little to diversity should not be accepted as members of the next generation. In this way, the greedy selection strategy of DE is not used to maintain the target vectors (X). In order to establish the minimum acceptable diversity contribution to be selected, the stopping criterion and elapsed generations are taken into account. One of the main weaknesses of RMDDC is that its convergence is highly delayed. Thus, to promote faster convergence than in RMDDC, two modifications are performed. First, no concepts of the

[1] The code in C++ can be downloaded in the next link https://github.com/joelchaconcastillo/Diversity_DE_Research.git

Algorithm 1 General scheme of DE-EDM

```
Randomly initialize the N target vectors, where each one is uniformly distributed.
G = 0
while stopping criterion is not satisfied do
    for i = 1 to N do
        Mutation: Generate the mutant vector (V_{i,G}) according to Eqn. (3.1).
        Crossover: Use recombination to generate the trial vector (U_{i,G}) according to Eqn. (3.2).
        Selection: Update the elite vector (E_{i,G} instead of X_{i,G}) according to Eqn. (3.3).
    end for
    Replacement: Select the target vectors (X_{G+1}) according to Algorithm 2 .
    G = G + 1
end while
```

Algorithm 2 Replacement Phase

```
1:  Input: Target vectors, Trial vectors, Elite, and ITV
2:  Update D_t = ITV − ITV ∗ (nfes/(FMDP × max_nfes))
3:  Current = Target ∪ Trial ∪ Elite.
4:  Survivors = Penalized = ∅.
5:  while |Survivors| < N And |Current| > 0 do
6:      Selected = Select the best vector of Current.
7:      Remove Selected from Current.
8:      Copy Selected to Survivors.
9:      Find the vectors from Current with a distance to Selected lower than D_t and move them to Penalized. Normalized distance is considered (Eqn. 3.4).
10: end while
11: while |Survivors| < N do
12:     Selected = Select the vector from Penalized with the largest distance to the closest vector in Survivors.
13:     Remove Selected from Penalized.
14:     Copy Selected to Survivors.
15: end while
16: return Survivors
```

multi-objective field are applied, instead a more greedy selection is taken into account. Second, the elite vectors are also considered as an input of the replacement strategy.

The replacement strategy (see Algorithm 2) operates as follows. It receives as input the target vectors, the trial vectors, and the elite vectors. In each generation, the N trial vectors of the next generation must be selected. First, it calculates a desired minimum distance between selected vectors (D_t) given the current number of elapsed function evaluations (line 2). Then, it joins the three multi-set of vectors in a multi-set of current vectors (line 3). Then, the multi-set of survivors and penalized vectors are initialized to the empty set (line 4). In order to select the N survivors (next target vectors) an iterative process is repeated (lines 5 - 13). In each step the best vector in *Current*, i.e. the one with best objective function is selected to survive, i.e. it is moved to *Survivors* (line 6 - 8). Then, the vectors in *Current* with a distance lower than D_t to the selected vector are transferred to *Penalized* (line 9). The way to calculate the distance between two vectors is by using the normalized Euclidean distance described in Eqn. (3.4), where n is the dimension of the problem, and $x_d^{(L)}, x_d^{(U)}$ are the minimum and maximum bounds of dimension d. In cases where *Current* is empty previous to the selection of N vectors, *Survivor* is filled by selecting in each step the vector in *Penalized* with the largest distance to the closest vector in *Survivor* (lines 10 - 13).

$$distance(\vec{x},\vec{y}) = \frac{\sqrt{\sum_{d=1}^{n}\left(\frac{x_d - y_d}{x_d^{(U)} - x_d^{(L)}}\right)^2}}{\sqrt{n}} \tag{3.4}$$

In order to complete the description it is important to specify the logic behind the way of calculating D_t. The value of D_t is used to alter the degree between exploration and exploitation so it should depend on the optimization stage. Specifically, this value should be reduced as the stopping criterion is reached with the aim of promoting exploitation. In our scheme, an initial threshold value (ITV) for D_t must be established. Then, similarly than in Segura et al. (2015c), a linear reduction of D_t is performed by taking into account the elapsed function evaluations and stopping criterion. Particularly, in this work, the stopping criterion is set by function evaluations. Similarly, a final moment of diversity promotion ($FMDP$) must be set, this parameter refers to the fraction of stopping criterion for which diversity is explicitly promoted. For example, $FMDP = 0.9$ means that the reduction of diversity is calculated in such a way that by 90% of the maximum number of evaluations, the resulting value D_t is 0. Therefore, in the remaining 10% diversity is not considered at all, which means that intensification is promoted. Thus, if max_nfes is the maximum number of evaluations and $nfes$ is the elapsed number of evaluations, D_t is calculated as $D_t = ITV - ITV * (nfes/(FMDP \times max\_nfes))$.

The initial threshold value or initial distance (ITV) affects the performance of DE-EDM. If this parameter is fixed large enough, then at the first optimization stages the algorithm aims to maximize the diversity of the target vectors, so a proper exploration is performed which is very important in some kinds of problems such as highly multi-modal and deceptive ones. Thus, the effect of premature convergence might be alleviated. However, a too large ITV might induce too much exploration, resulting in an improper exploitation phase. In the opposite case, a too low ITV might result in an improper exploration phase, thus hindering the avoidance of local optima. Depending on the type of fitness landscape and the stopping criterion, the optimal ITV could vary. For instance, deceptive and highly multi-modal problems usually require larger values than uni-modal problems. However, in our proposal, ITV is not adapted to each problem; instead, an experimental study is attached in the experimental validation section to check the robustness of different values of ITV.

In the same way that in the standard DE, in DE-EDM the crossover probability (CR) and the mutation factor (F) must be set. The first one is perhaps the most important for the performance according to several studies developed by Montgomery et al. (Montgomery and Chen, 2010). These authors empirically proved that extreme CR values lead to vastly different search behaviors. They explained that low values of CR result in a search that is aligned with a small number of axes of the search space and induces small displacements. This provokes a gradual and slow convergence that, in some scenarios, might result in robust behavior. Furthermore, high values CR might generate higher quality solutions with a lower probability. However, these transformations cause large displacements that could significantly improve the solutions when they are successful. According to this, we employ both high and low CR values as is indicated in Eqn. (3.5).

$$CR = \begin{cases} Normal(0.2, 0.1), & \text{if} \quad rand[0,1] \leq 0.5 \\ Normal(0.9, 0.1), & \text{otherwise} \end{cases} \tag{3.5}$$

Following the principles of several SHADE variants (Awad et al., 2016; Brest et al., 2016), the elapsed function evaluations are considered in the random generation of the mutation factor F. In particular, each F is sampled from a Cauchy distribution (Eqn. 3.6).

$$Cauchy(0.5, 0.5 * nfes/max\_nfes) \tag{3.6}$$

Therefore, in the initial optimization stages, F values near 0.5 are generated with high probability. Then, as the execution progresses, the density function suffers a gradual transformation and the variance increases, meaning that values outside the interval $[0.0, 1.0]$ are generated with a higher probability. In the cases where values greater than 1.0 are generated, the value 1.0 is used. In the case of generating a negative value, the F is re-sampled. This means that the probability of generating large F-values increases as the execution progresses. The principle behind this decision is to help in the avoidance of fast convergence.

Type Function	CEC 2016	CEC 2017
Uni-modal	f_1 - f_3	f_1 - f_3
Simple multi-modal	f_4 - f_{16}	f_4 - f_{10}
Hybrid (multi-modal)	f_{17} - f_{22}	f_{11} - f_{20}
Composition (multi-modal)	f_{23} - f_{30}	f_{21} - f_{30}

Table 3.1 Problems grouped by properties for the CEC 2016 and CEC 2017 benchmarks

3.4 Experimental Validation

This section presents the experimental study carried out to validate the performance of DE-EDM. Specifically, we show that by explicitly controlling the diversity in DE, the results of state-of-the-art algorithms are further improved. In particular, the benchmarks of CEC 2016 and CEC 2017 are considered (see Appendix A). Each one of them is composed of thirty different problems, meaning that the validation is performed on a set of large and diverse functions. The kind of problems of each benchmark is divided in uni-modal, simple multi-modal, hybrid, and composition functions. Table 3.1 shows the problems that belong to each category for the benchmarks of CEC 2016 and CEC 2017. In the hybrid functions, the variables are randomly divided into some sub-components and each one is related to basic functions. In the same line, the composition functions merge the properties of several sub-functions and maintain continuity around each local optima. Also the local optima which has the smallest bias value is the global optimum. The search space is bounded by the range $\Omega = \prod_{j=1}^{n}[-100, 100]^n$.

The experimental validation takes into account the algorithms that attained the first places of each year competition, as well as the standard DE. Additionally, comparisons against other diversity-based schemes are included. The algorithms considered from CEC 2016 are UMOEAs-II (Elsayed et al., 2016) and L-SHADE-EpSin (Awad et al., 2016), which achieved the first and second places, respectively[2]. Similarly, the top algorithms from CEC 2017 are taken into account, i.e. EBOwithCMAR (Kumar et al., 2017) and jSO (Brest et al., 2017)[3]. Following the recommendations given in Molina et al. (2017), all of these algorithms are tested with both benchmarks.

Given that the optimizers taken into account are stochastic algorithms, each execution was repeated 51 times with different seeds. In all cases, the stopping criterion was set to 25×10^6 function evaluations. In addition, problems were configured by setting $n = 10$ (i.e. only ten variables were taken into account). The validation follows the guidelines of CEC benchmark competitions and the statistical tests proposed in Durillo et al. (2010) are also included. Note that, as it is usual in these competitions, when the gap between the values of the best solution found and the optimal solution is 10^{-8} or smaller, the error is treated as 0. The parameterization indicated by the authors was used in every algorithm and it is as follows:

- **EBOwithCMAR**: For EBO, the maximum population size of $S_1 = 18n$, minimum population size of $S_1 = 4$, maximum population size of $S_2 = 146.8n$, minimum population size of $S_2 = 10$, historical memory size H=6. For CMAR Population size $S_3 = 4 + 3log(n)$, $\sigma = 0.3$, CS = 50, probability of local search $pl = 0.1$ and $cfe_{ls} = 0.4 * FE_{max}$.

- **UMOEAs-II**: For MODE, maximum population size of $S_1 = 18n$, minimum population size of $S_1 = 4$, size memory H=6. For CMA-ES Population size $S_2 = 4 + \lfloor 3log(n) \rfloor$, $\mu = \frac{N}{2}$, $\sigma = 0.3$, CS = 50. For local search, $cfe_{ls} = 0.2 * FE_{max}$.

[2]Source codes available at https://github.com/P-N-Suganthan/CEC2016

[3]Source codes available at https://github.com/P-N-Suganthan/CEC2017-BoundContrained

- **jSO**: Maximum population size = $25log(n)\sqrt{n}$, historical memory size H= 5, initial mutation memory $M_F = 0.5$, initial probability memory $M_{CR} = 0.8$, minimum population size = 4, initial p-best = $0.25 * N$, final p-best = 2.

- **L-SHADE-EpSin**: Maximum population size = $25log(n)\sqrt{n}$, historical memory size H= 5, initial mutation memory $M_F = 0.5$, initial probability memory $M_{CR} = 0.5$, initial memory frequency $\mu_F = 0.5$, minimum population size = 4, initial p-best = $0.25 * N$, final p-best = 2, generations of local search $G_{LS} = 250$.

- **DE-EDM**: $ITV = 0.3$, $FMDP = 0.9$, population size = 250, CR $\sim Normal(\{0.2, 0.9\}, 0.1)$, F $\sim Cauchy(0.5, 0.5 * nfes/max_{nfes})$.

- **Standard-DE**: population size = 250 (operators as DE-EDM), CR $\sim Normal(\{0.2, 0.9\}, 0.1)$, F $\sim Cauchy(0.5, 0.5 * nfes/max_{nfes})$.

Our experimental analyses are based on the error, i.e. the difference between the optimal solution and the best obtained solution. In order to statistically compare the results, a similar guideline than the one proposed in Durillo et al. (2010) was used. First, a Shapiro-Wilk test was performed to check whether the values of the results followed a Gaussian distribution. If so, the Levene test was used to check for the homogeneity of the variances. If samples had equal variance, an ANOVA test was done; if not, a Welch test was performed. For non-Gaussian distributions, the non-parametric Kruskal-Wallis test was used to test whether samples are drawn from the same distribution. An algorithm X is said to win algorithm Y when the differences between them are statistically significant, and the mean and median error obtained by X are lower than the mean and median achieved by Y.

Tables 3.2 and 3.3 offer a summary of the results obtained for CEC 2016 and CEC 2017, respectively. The column tagged with "Always Solved" shows the number of functions where a zero error was obtained in the 51 runs. Additionally, the column tagged with "At Least One Time Solved" shows the number of functions that were solved at least in one run. Practically all functions (28 of them) of the CEC 2017 benchmark were solved with DE-EDM at least one time. Additionally, 21 functions of CEC 2016 were also solved. This contrasts with the results obtained by state-of-the-art algorithms. They were able to reach optimal values in significantly fewer functions. In order to confirm the superiority of DE-EDM, the pair-wise statistical tests described above were used. The column tagged with the symbol $\uparrow$ shows the number of cases where the superiority of each method could be confirmed, whereas the column tagged with the symbol $\downarrow$ counts the number of cases where the method was inferior. Finally, the number comparisons with not significant differences are shown in the column tagged with the symbol $\longleftrightarrow$. The results of the statistical tests show that DE-EDM achieved the best results in both years. The number of victories in CEC 2016 and CEC 2017 were 77 and 88, while the number of losses were 25 and 6, respectively. DE-EDM is the approach with the largest number of victories and lowest number of losses in both benchmarks, confirming the superiority of the proposal. The last column — tagged with "Score" — considers the official score of CEC's competitions. Particularly, the raking of the algorithms is attained by taking into account the two scores defined in Eqn. (3.7). Then, the final score is calculated as the sum $Score = Score_1 + Score_2$. In Eqn. (3.7), the SE of an algorithm is the sum of the mean error values obtained in the 30 benchmark functions, i.e. $SE = \sum_{i=1}^{30} error\_f_i$. Then, SE_{min} is the minimal SE from all the algorithms. In order to calculate SR and SR_{min}, algorithms are sorted in each function in base of the attained mean error. Then, a rank is assigned to each algorithm in base of such an ordering. Finally, the SR of a method is the sum of the ranks obtained for each function, and SR_{min} is the minimal SR of all algorithms.

Table 3.2 Summary of the results obtained by DE-EDM and state-of-the-art algorithms- CEC 2016

Algorithm	Always Solved	At Least One Time Solved	Statistical Tests			Score
			↑	↓	⟷	
DE-EDM	13	21	77	25	48	100.00
UMOEAs-II	9	14	51	31	68	62.45
Standard-DE	11	19	50	46	54	56.29
jSO	9	17	47	51	52	55.43
EBOwithCMAR	8	14	35	56	59	50.28
L-SHADE-Epsilon	7	13	20	71	59	50.12

Table 3.3 Summary of the results obtained by DE-EDM and state-of-the-art algorithms - CEC 2017

Algorithm	Always Solved	At Least One Time Solved	Statistical Tests			Score
			↑	↓	⟷	
DE-EDM	21	28	88	6	56	100.00
Standard-DE	12	21	56	29	65	42.91
EBOwithCMAR	9	18	34	46	70	37.14
L-SHADE-Epsilon	8	19	7	81	62	32.78
jSO	8	15	29	55	66	29.30
UMOEAs-II	11	15	43	40	67	26.89

$$
\begin{aligned}
Score_1 &= \left(1 - \frac{SE - SE_{min}}{SE}\right) \times 50, \\
Score_2 &= \left(1 - \frac{SR - SR_{min}}{SR}\right) \times 50,
\end{aligned}
\tag{3.7}
$$

Note that, on the basis of this definition, the best attainable score is 100. This happens when a given approach obtains both SR_{min} and SE_{min}. DE-EDM attained the best attainable score in both years, confirming its clear superiority compared to both state-of-the-art and standard DE. In these long-term executions, the standard DE reached third and second positions in the problems of CEC 2016 and CEC 2017, respectively. This means that the performance of state-of-the-art algorithms is not so impressive in long-term executions.

Since our proposal is based on the explicit control of diversity, Fig. 3.1 shows the evolution of the mean of the diversity calculated as the mean distance to the closest vector with the aim of better understanding its behavior. In particular, functions f_1 and f_{30} were selected for this analysis because they have quite different features (easy uni-modal vs. complex multi-modal). The left side shows the diversity of the Elite population. It is remarkable that, while there are no direct constraints in the Elite population related to diversity, the diversity is implicitly maintained. The right side shows the diversity of the target vectors. As expected, the diversity gradually decreases, and the degree of diversity is maintained until 90% of the total function evaluations is reached.

Finally, in order to provide comparable results of our proposal, Tables 3.4 and 3.5 report the best, worst, median, mean, standard deviation, and success rate for both benchmarks. These tables show that all the uni-modal problems were solved by our proposal. Additionally, several simple and even some of the most complex multi-modal functions were optimally solved. In fact, several complex functions that had never been solved by state-of-the-art could be solved by DE-EDM.

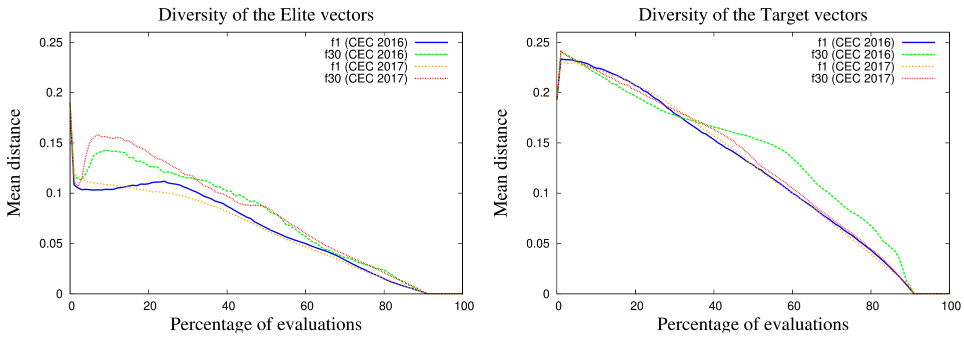

Fig. 3.1 Mean distance to the closest vector of the 51 executions with the problems f_1 and f_{30} (CEC 2016 and CEC 2017). The initial distance factor considered corresponds to $ITV = 0.3$

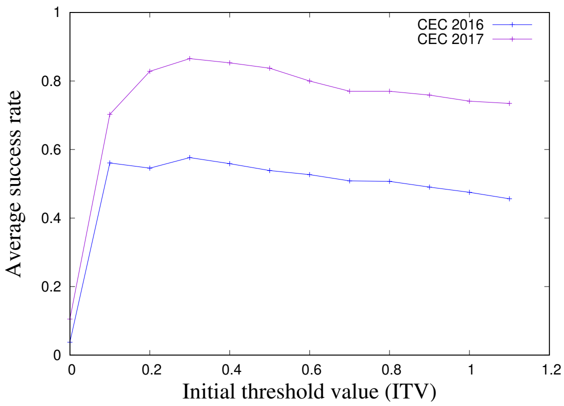

Fig. 3.2 Mean success rate for different ITV in the CEC 2016 and CEC 2017 benchmarks with a population size equal to 250 and 25×10^6 function evaluations

3.4.1 Comparison of Diversity Replacement-Based Schemes

In order to better validate the advantages provided by the replacement phase proposed in this chapter, comparisons against two additional diversity replacement-based strategies were developed. Particularly, the replacement-based methods taken into consideration are the *Restricted Tournament Selection* (RTS) and the *Hybrid Genetic Search with Adaptive Diversity Control* (HGSADC). These replacement-based strategies were incorporated to the standard DE framework. The three variants were executed with the same parameterization. Note that both RTS and HGSADC require additional parameters. Based on preliminary analyses, RTS was performed with a sample size $CF = 25$, and HGSADC was performed with $N_{Close} = 1$ and $N_{Elite} = 8$. The remaining configuration follows the same parameterization previously defined. Tables 3.6 and 3.7 summarize the results attained for both benchmarks with the same meaning that in the previous experiment. Again, DE-EDM achieved the best results in both years. It is clear that DE-EDM achieved a significantly higher *Score* than the remaining diversity-based methods. Note that the main difference of our proposal in contrast to the schemes RTS and HGSADC lies in the fact that they provide modifications with the aim of delaying convergence but in an indirect way, meaning that premature convergence might not always be avoided. Second, while these methods have been quite successful in comparison to Evolutionary Algorithms with a high selection pressure, DE already incorporates a replacement strategy that follows some of the principles of crowding. Thus, the advantages of

Table 3.4 Results for DE-EDM in the CEC 2016 problems

	Best	Worst	Median	Mean	Std.	Succ. Rate
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_6	0.00E+00	3.60E-02	4.00E-03	7.39E-03	1.15E-02	3.92E-01
f_7	2.00E-02	1.02E-01	5.90E-02	5.77E-02	4.93E-02	0.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{10}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{11}	0.00E+00	6.00E-02	0.00E+00	5.88E-03	1.90E-02	9.02E-01
f_{12}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{13}	1.00E-02	8.00E-02	5.00E-02	4.67E-02	2.60E-02	0.00E+00
f_{14}	1.00E-02	5.00E-02	3.00E-02	2.82E-02	2.13E-02	0.00E+00
f_{15}	0.00E+00	4.70E-01	2.20E-01	1.99E-01	1.55E-01	1.96E-02
f_{16}	4.00E-02	1.50E-01	8.00E-02	8.47E-02	4.96E-02	0.00E+00
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{18}	0.00E+00	2.00E-02	1.00E-02	7.65E-03	6.32E-03	3.14E-01
f_{19}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{20}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{22}	0.00E+00	3.00E-02	0.00E+00	3.73E-03	2.76E-02	7.65E-01
f_{23}	0.00E+00	1.00E+02	0.00E+00	2.55E+01	5.10E+01	7.45E-01
f_{24}	0.00E+00	6.90E-01	0.00E+00	2.61E-02	1.33E-01	9.61E-01
f_{25}	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00	0.00E+00
f_{26}	8.00E-02	1.00E+02	5.29E+01	5.20E+01	3.19E+01	0.00E+00
f_{27}	2.50E-01	9.10E-01	5.40E-01	5.60E-01	2.92E-01	0.00E+00
f_{28}	0.00E+00	3.57E+02	3.43E+02	2.76E+02	1.60E+02	1.96E-01
f_{29}	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00	0.00E+00
f_{30}	1.84E+02	1.84E+02	1.84E+02	1.84E+02	3.25E-02	0.00E+00

incorporating RTS and HGSADC into DE are diminished. Therefore, making decisions by taking into account the stopping criterion and elapsed generations seems to be the key to properly avoid premature convergence in long-term executions.

3.4.2 Empirical Analyses of the Initial Threshold Value

In our proposal, diversity is explicitly promoted and the total amount of diversity maintained in the population depends on the Initial Threshold Value (ITV). Therefore, the effect of this parameter on quality is analyzed in this section. Particularly, the same scheme previously mentioned was taken into account. However, several initial threshold values were considered ($ITV = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1\}$).

Fig. 3.2 shows the mean success rate achieved for both benchmarks when considering different values of ITV. The most relevant conclusions are:

- If diversity is not promoted ($ITV = 0.0$) there is an important degradation in the performance.
- The performance is quite robust in the sense that a large range of ITV values provide good enough results. For instance, values in the range $[0.2, 0.6]$ provide high-quality solutions.
- To some extent, if the initial threshold value (ITV) is very large (values larger than 0.6), the quality of the solutions is affected.

To summarize, DE-EDM incorporates a novel parameter which is important in its performance. However, results are quite robust in the sense that even if it is not tuned for each problem, high-quality results can be obtained and that a large range of values provide competitive results.

Table 3.5 Results for DE-EDM in the CEC 2017 problems

	Best	Worst	Median	Mean	Std.	Succ. Ratio
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{10}	0.00E+00	1.20E-01	0.00E+00	1.65E-02	3.39E-02	7.45E-01
f_{11}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{12}	0.00E+00	2.20E-01	0.00E+00	6.37E-02	1.76E-01	6.67E-01
f_{13}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{14}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{15}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{16}	0.00E+00	2.10E-01	0.00E+00	2.47E-02	7.27E-02	8.82E-01
f_{17}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{18}	0.00E+00	1.00E-02	0.00E+00	1.96E-03	4.47E-03	8.04E-01
f_{19}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{20}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{21}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{22}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{23}	0.00E+00	3.00E+02	0.00E+00	3.49E+01	1.03E+02	8.82E-01
f_{24}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{25}	0.00E+00	1.00E+02	0.00E+00	3.92E+00	2.00E+01	9.61E-01
f_{26}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{27}	0.00E+00	3.87E+02	3.87E+02	2.05E+02	2.68E+02	1.96E-02
f_{28}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00
f_{29}	1.45E+02	2.26E+02	2.18E+02	1.99E+02	4.21E+01	0.00E+00
f_{30}	3.95E+02	3.95E+02	3.95E+02	3.95E+02	2.10E-01	0.00E+00

Table 3.6 Summary of the results obtained with different replacement strategies - CEC 2016

Algorithm	Always Solved	At Least One Time Solved	Statistical Tests			Score
			$\uparrow$	$\downarrow$	$\longleftrightarrow$	
DE-EDM	13	21	51	1	8	100.00
RTS	2	7	3	47	10	19.74
HGSADC	3	15	21	27	12	44.12

3.4.3 Effect of the Elite Population

The replacement strategy proposed in this chapter, which is integrated into the standard version of DE, considers an additional elite population. The motivation behind this population is to maintain information of some promising regions which at some point might be discarded as members of next generations. Preliminary analyses have shown that this inconvenient is more present in highly multi-modal problems and for short runs (e.g. $< 2.5 \times 10^5$ function evaluations). In order, to have better understanding of this elite population, this section introduces an extension of DE-EDM termed as DE-EDM-II. The only difference between DE-EDM and DE-EDM-II is that the latter leaves out the elite population; instead, only target vectors and trial vectors are taken into account. This section adopts the same configuration than in the previous section. Therefore, as pointed out in Algorithm 1 the usual selection of DE is no longer considered (line 7). Furthermore, the replacement strategy (Algorithm 2) only receives as input the target vectors and the trial vectors.

Tables 3.8 and 3.9 report a summary of the results achieved for the contests CEC 2016 and CEC 2017, respectively. Note that these tables report similar information than in previous sections as well as the average error. This is calculated taking into account all the functions in all the runs, which is tagged as *Mean Error* in the last column. Since long-term executions are considered, these tables show that the performance of diversity-aware methods (DE-EDM and DE-EDM-II) is similar. In the results of CEC 2016, DE-EDM-II achieved

Table 3.7 Summary of the results obtained with different replacement strategies - CEC 2017

Algorithm	Always Solved	At Least One Time Solved	Statistical Tests			Score
			↑	↓	⟷	
DE-EDM	21	28	49	0	11	100.00
RTS	4	12	2	49	9	30.91
HGSADC	6	18	23	25	12	40.86

Table 3.8 Summary of the results obtained by DE-EDM, DE-EDM-II and state-of-the-art algorithms- CEC 2016

Algorithm	Always Solved	At Least One Time Solved	Statistical Tests			Score	Mean Error
			↑	↓	⟷		
DE-EDM-II	9	23	81	32	67	61.03	23.49
DE-EDM	13	21	83	31	66	60.04	24.63
jSO	9	14	57	46	77	33.86	46.21
EBOwithCMAR	7	13	23	89	68	30.40	48.20
UMOEAs-II	8	14	42	71	67	27.43	57.86
Standard-DE	9	17	52	66	62	27.90	60.11
L-SHADE-Epsilon	11	19	55	58	67	27.98	52.91

the best score and the lowest mean error. Nevertheless, the number of wins (81) is slightly lower than the number of wins achieved by DE-EDM (83). Similarly, the number of times that DE-EDM-II solved a problem in all the runs was equal to nine, which is less than DE-EDM by four points. State-of-the-art methods achieved similar results, except for Standard-DE, whose ranking deteriorated compared to previous experiments. Note that the score's deterioration in Standard-DE indicates that DE-EDM-II achieved quality results in some test functions that were not properly solved by DE-EDM. Interestingly, this is not the same for CEC 2017, where DE-EDM achieved the best score and the lowest mean error. In this set of problems, diversity-aware variants are also superior to the state-of-the-art methods. For example, the third method is Standard-DE with a score of 30.58, which is approximately half of the second ranked method DE-EDM -II with 71.72. Note that the problems proposed in CEC 2017 adopt more hybrid functions and composition functions in comparison to the problems proposed in CEC 2016, however some of these kind of problems are better solved by diversity-aware methods, which can be reviewed in the number of times that these methods always solved a test function. These results have pointed out that to achieve quality results, the elite population might be discarded in the design of these kinds of methods; nevertheless, this is only suggested for long runs and in the single-objective case[4]. Although other strategies might be integrated to DE-EDM-II it should be done carefully. Note that the methods proposed in this thesis involve only simple modifications to the most standard population-based meta-heuristics. This was done with the intention of clearly demonstrating the benefits of properly managing diversity in the population.

3.4.4 Analysis of the Diversity-Parameters in DE-EDM-II: Initial Threshold Value (ITV) and Final Moment of Diversity Promotion (FMDP)

At this point, the reader might notice that diversity is explicitly maintained based on a linear model which requires both the initial threshold value (ITV) and the final moment of diversity promotion (FMDP).

In the main experimental section, the FMDP is set to 0.9, which has allowed the discovery of high-quality results. However, the effect of this parameter might be different depending on the configuration and properties of each problem. In order to better understand the final moment of diversity promotion, several FMDP values with several ITV values were tested. Specially, $FMDP = \{0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 1.0\}$ and

[4]Note that the integration of an external population has positive effects in multi-objective optimization as can be confirmed in Chapter 6.

Table 3.9 Summary of the results obtained by DE-EDM, DE-EDM-II and state-of-the-art algorithms- CEC 2017

Algorithm	Always Solved	At Least One Time Solved	Statistical Tests			Score	Mean Error
			↑	↓	⟷		
DE-EDM	21	28	92	9	79	80.82	27.91
DE-EDM-II	21	28	87	14	79	71.72	30.39
Standard-DE	12	21	59	43	78	30.58	88.49
UMOEAsII	9	18	45	57	78	28.10	88.83
jSO	8	19	31	72	77	25.49	94.41
L-SHADE-Epsilon	8	15	8	100	72	23.56	650.28
EBOwithCMAR	11	15	36	63	81	14.50	625.63

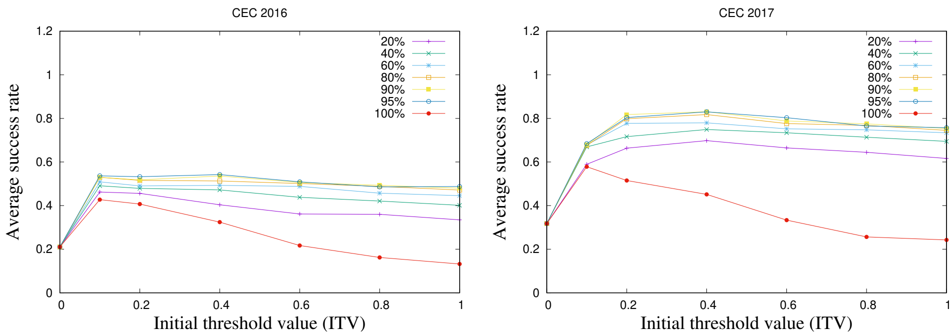

Fig. 3.3 Mean success rate for different ITV and $FMDP$ values of DE-EDM-II. Note that $FMDP$ is converted to a percentage (each line is a configuration of $FMDP$). The CEC 2016 (left side) and CEC 2017 (right side) benchmarks are configured with a population size of 250 and 25×10^6 function evaluations.

$ITV = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 1.0\}$ were considered. The remaining parameters were set as in the main section. Figure 3.3 shows the results achieved in CEC 2016 (left side) and CEC 2017 (right side). The worst results were achieved with $FMDP = 1.0$ (100% of the total function evaluations). This reveals that DE-EDM-II requires a period of time without promoting diversity to properly induce intensification at the end of the execution. Low values (percentages) of $FMDP$ are enough to attain significant benefits, thus adjusting this parameter for shorter stopping criteria might be an alternative. Note that using values of $FMDP$ larger than 0.95 (95%) deteriorates the quality of the final results. In particular, the values of $FMDP$ in the range 0.2 (20%) and 0.95 (95%) and the values of ITV in the range 0.2 to 0.6 offer an acceptable performance, but the best overall results are obtained with values of $FMDP$ close to 0.95 (95%).

3.4.5 Analysis of the Population size in DE-EDM-II

The amount of individuals in the population influences the diversity along the search process. A too small amount of individuals might result in more intensification than the required. In contrast, a large amount of individuals might slow down the convergence resulting in poor solutions. This section presents some results to analyze the effect of the population size with DE-EDM-II. The initial threshold value and final moment of diversity promotion were set to $ITV = 0.4$ and $FMDP = 0.9$, respectively. The remaining configuration was set as in the main section. Since using a larger population size might be computationally expensive for the replacement stage, for each set of test functions (i.e. each year), the population size is increased as long as the average success rate has an improvement. Particularly, in CEC 2016 and CEC 2017, the population sizes tested were $\{50, 100, 250, 500, 1000, 1500, 2000, 3000, 10000\}$ and $\{50, 100, 250, 500, 1000\}$, respectively. Tables 3.10 and 3.11 show a summary of the results obtained by DE-EDM-II for CEC 2016 and CEC 2017, respectively. In the test problems of CEC 2016 (Table 3.10) the best results were achieved with a population size equal to

Table 3.10 Summary of the results obtained by DE-EDM-II with several population sizes in CEC 2016

Population Size	Always Solved	At Least One Time Solved	Mean Error	Average Success Rate
50	8	**23**	30.04	0.47
100	10	21	29.41	0.47
250	8	21	25.53	0.53
500	9	21	19.38	0.57
1000	**11**	20	16.04	0.59
1500	9	21	16.19	0.60
2000	**11**	20	**15.64**	**0.61**
3000	**11**	20	16.93	0.60
10000	7	17	28.62	0.46

Table 3.11 Summary of the results obtained by DE-EDM-II with several population sizes in CEC 2017

Population Size	Always solved	At Least One Time Solved	Mean Error	Average Success Rate
50	17	**28**	35.93	0.76
100	20	**28**	31.28	0.82
250	20	**28**	26.44	**0.83**
500	**21**	27	22.99	0.80
1000	**21**	26	**22.37**	0.80

2000, resulting in an average success rate and mean error of 0.61 and 15.64, respectively. Note that the mean error with a population size equal to 2000 achieved a lower mean error in comparison to the main section[5]. A reasonable explanation for the benefits achieved with larger populations is that the majority of problems are multi-modal resulting in a more appropriate exploration of the search space. Differently, in the test problems of CEC 2017 the average success rate is the best with a population size equal to 250, nevertheless the mean error continued decreasing with larger population sizes. In fact, a population with 1000 individuals achieved a mean error equal to 22.37, which is slightly lower in comparison with the mean error obtained with a population size equal to 250 (26.44). Note that the mean error of all the configurations in both years is superior to the state-of-the-art methods (see Table 3.8 and Table 3.9). These best values reported by other methods are 46.21 (jSO) and 88.39 (Standard-DE) of CEC 2016 and CEC2017, respectively. This findings might suggest that DE-EDM-II might be also quite effective for small populations (e.g. Micro-DE).

3.5 Summary

Premature convergence is one of the most important drawbacks of DE. This chapter proposes a novel variant of DE, the *Differential Evolution with Enhanced Diversity Maintenance* (DE-EDM), which incorporates a novel replacement phase that is based on a recently proposed strategy (RMDDC). Note that RMDDC was successfully used with memetic algorithms in the combinatorial optimization field. In order to attain a faster convergence than in RMDDC, two modifications are proposed for the integration with DE. First, no concepts of the multi-objective field are applied, meaning that a more greedy strategy is devised. Second, an elite population is incorporated. Experimental validation shows the important benefits obtained by DE-EDM against a wide variety of methods in long-term executions. DE-EDM requires setting a new parameter, which is called the initial threshold value. The quality obtained depends on the proper specification of this parameter. However, fixed values in a large range could be used to properly deal with the benchmarks of the CEC 2016 and CEC 2017 competitions, which means that a quite robust proposal is obtained.

[5]Note that in the main section was tested $ITV = 0.3$ instead of $ITV = 0.4$

Some analyzes suggested that the elite population might be discarded in some configurations (e.g. for long runs). Thus, the decision of integrating an elite population is up to the user requirements. Regardless of the elite population, the benefits of explicitly promoting diversity are clear. The final results suggested that DE-EDM-II is quite robust regarding the population size and the initial threshold value, which is also true for DE-EDM, but in the case of the population size, some important benefits could be obtained by properly tuning it. Thus, integrating DE-EDM with methods that alter population size in a dynamic way seems promising.

Chapter 4

Importance of Diversity for A Constrained Menu Planning Problem

Abstract

Planning is the task of finding an arrangement of events to accomplish a proper goal and to satisfy a set of domain constraints (Choon et al., 2016). Menu planning, a more particular domain type of planning, is a complex task that involves finding a combination of menu items by taking into account several kinds of features, such as nutritional, and economical, among others. In order to deal with the menu planning as an optimization problem, these features are transformed into constraints and objectives. Several formulations of this problem have been defined, and meta-heuristics have been significantly successful in solving them. In recent years, memetic algorithms with explicit control of diversity have led to the achievement of high-quality solutions in several combinatorial problems. The main motivation of this chapter is to show that these types of methods are versatile and viable for the menu planning problem. In particular, two variants of menu planning are modeled as single-objective constrained optimization problems. While the first considers the total cost of a meal plan, the second measures the level of repetition with respect to meals. Although each formulation is analyzed independently, both of them are addressed by the same memetic algorithm, which integrates an iterated local search. Each formulation integrates a similar local search, and both follow the same methodology to fulfill the nutrient requirements. In addition, a novel crossover is also proposed. The importance of incorporating an explicit control of diversity is analyzed in both formulations. In the first one is carried out by using several well-known strategies to control the diversity as well as our proposal. In the second case, a flexible application which supports user specification of meals and nutrients intake is developed. Note that the last application is currently considered by nutritionists at the "Integral Center of Care for Children and Teenagers" (CIANNA) [1].

[1] Centro Integral de Atención a Niñas, Niños y Adolescentes

The results show that in both formulations, the incorporation of an explicit method to manage diversity, ad-hoc operators, and efficient local search is mandatory to achieve high-quality solutions.

4.1 Introduction

According to the *World Health Organization*, nutrition is one of the main pillars of health and development. In general, healthy people are stronger, more productive, and better able to break the cycle of poverty and hunger and to develop their full potential. Malnutrition carries considerable risks to human health. Currently, the world faces a double burden of malnutrition that includes both malnutrition and over-nutrition or overweight. The increase in overweight and obesity around the world is one of the main challenges for public health. People of all ages and conditions face this type of malnutrition, as a result of which the rates of diabetes, cardiovascular diseases, and other diseases related to the diet are increasing dramatically. This problem might be critical for some ages, for instance, with children, which is even more delicate. In any case, the only way to avoid this situation is to follow a healthy and balanced diet. Controlling the feeding of children might be an impossible task for public health organizations. However, educational institutions and government shelters (e.g. CIANNA) can take steps to improve the nutrition of children by offering balanced plans, which might be a challenging task. While some school cafeterias offer at least one meal per day (usually lunch), some government shelters have to design an entirely day of meals. Note that in both situations, the set of dishes or food should be proposed and periodically reviewed by an expert to verify that the menus fulfill the appropriate healthy nutritional recommendations.

In practice, most healthy and balanced diets are manually designed by a nutrition expert. An expert trained professionally in the field of nutrition or food science is able to define specific meals that are ideally suited to a person (or group of people) nutritional requirements. However, by considering general criteria for children nutrition and a predefined set of meal choices, it is possible to properly formulate the problem of automatically generating healthy and balanced plans. Such a problem is usually denoted in the literature as *Menu Planning Problem* (MPP) (Choon et al., 2016).

The MPP involves generating a diet plan, usually daily, weekly or monthly, consisting of healthy menus for the main meals of the day. The different meals that comprise every menu in the plan should be fully specified. For example, for lunch, we could specify the starter, the main course, the dessert, the drink and the side dishes. The general plan must fulfill general recommendations for healthy and balanced diets. Such recommendations must consider human energy requirements depending on physiological features of the person (e.g., age, gender, height) and lifestyle (e.g., amount and frequency of physical exercise) in order to establish an adequate proportion for macronutrients (carbohydrates, proteins and fats) and micronutrients (vitamins and minerals) intakes (Aberg, 2009).

The most important and common consideration for menu planning problems, in general is to ensure a healthy and balanced menu from a nutritional point of view. These nutritional requirements are usually grouped into multiple constraints, as they must provide the minimum and maximum recommended amounts of different nutrients (Kahraman and Seven, 2005). Other issues –as the variety, quality (seasonal, fresh, etc.) or cooking method of the food– also impact on the healthiness of the diet. Studies have shown that food variety and/or diversity are healthy and related to diet quality (Steyn et al., 2006). The variety of nutrients is obtained by eating all kinds of foods, so all food groups must be considered (cereal, fruits, vegetables, meats, etc.), since no single food contains all the necessary nutrients by itself. However, some food groups provide more nutrients than others; therefore, it is important to eat as diverse as possible and in the right proportions.

In this chapter is proposed a novel Memetic Algorithm (MA) with explicit control of diversity to address two formulations of the MPP. The reason for including explicit control of diversity is that, in recent years, the results of several well-known combinatorial optimization problems have been further improved by including these control mechanisms (Romero Ruiz and Segura, 2018). Most of such methods have been shown to be successful in the very long term, that is, considering execution of several days or even weeks (Segura et al., 2015a). In this chapter, the viability of using these methods, when the computational power is more restricted, is also studied. In particular, executions with a user-variable stopping criterion are considered (Romero Ruiz and Segura, 2018). The benefits of incorporating an explicit strategy to control the diversity are clear in both formulations. In order, to better validate the hypothesis of this thesis, in each formulation, a novel crossover operator and a local search are also proposed and tested. In the first formulation (school cafeterias), the crossover operator is compared with the traditional uniform crossover and with some state-of-the-art methods regarding the promotion of diversity. The results suggest that the novel crossover operator increases the robustness of the proposal, and its advantages appear when dealing with large instances. Similarly, in the second formulation an extension of this novel crossover operator and local search are presented. Note that this last formulation supports the user-specifications of the constraints and meals, which is currently considered as a desktop application at CIANNA. This last was carried out in collaboration with the "Software Development Department" (GDS)[2] at CIMAT. Evidence reveals the robustness and benefits of explicitly promoting diversity for short-runs and with different algorithmic parameterizations. Note that in both cases this was a collaborative development that was addressed with two different teams.

The rest of this chapter is organized as follows. The mathematical formulations of the menu planning problem are detailed in Section 4.2. Section 4.3 describes the memetic algorithm framework proposed in this chapter, which can be easily adapted to other formulations. The proper description of the crossover operators and the iterated local search applied to the school cafeterias is described in section 4.3.2. In Section 4.3.3 the experimental validation for the school cafeteria formulation is detailed. The extension of the crossover operator and local search for the CIANNA formulation is detailed in Section 4.3.4. Section 4.3.5 presents a parameter analysis of the latter formulation. Finally, our summary is given in Section 4.3.16.

4.2 Menu Planning Formulation for Nutriment Requirements

The Menu Planning Problem (MPP) is a classic optimization problem that for some formulations can be categorized as NP-Complete (Choon et al., 2016; Gazan et al., 2018). The principle behind MPP is to find an optimal combination of menu items that meets particular nutritional, structural and variety requirements for a sequence of days (Balintfy, 1964). In this chapter, the nutritional quality of the meal plans is treated as the problem constraints. As is usual in MPP, both formulations considers several nutrients such as the total intakes for energy, fats, carbohydrates, and proteins. These nutrients are calculated for each generated meal plan. Therefore, for each of the nutrients, its nutrient intake has to be within the acceptable range of its recommendations; otherwise, the solution will be considered non-feasible. Besides, the set of constraints is modeled by two sub-sets of constraints: global constraints and daily constraints.

In the school cafeteria formulation the meal plan only considers lunch, thus recommendation ranges apply to quantities taken at lunch time. Differently, in the CIANNA's formulation a meal plan includes the meals of a whole day, which are breakfast, morning snack, starter (lunch), main course (lunch), evening snack and dinner. However, in both formulations for each nutrient h, r_h denotes the recommended amount of ingest per

[2] Gerencia de Desarrollo de Software (CIMAT)

day. Additionally, there is a set R of ranges or pairs (r_{min_h}, r_{max_h}) that describes the feasible space. The amount of constraints or nutrients to fulfill is denoted as $|H|$, where elements r_{min_h} and r_{max_h} represent the minimum and maximum **proportion** of a nutrient h which is allowed within the global plan for n days.

Formally, a feasible individual $\vec{s}$ –from a nutritional point of view– should fulfill the following constraints:

$$\forall h \in H : \alpha_{min_h} \leq in(\vec{s}, h) \leq \alpha_{max_h} \tag{4.1}$$

where $in(\vec{s}, h)$ denotes the global intake of element h in the whole menu, and the minimum and maximum global limits of the feasible space are defined as $\alpha_{min_h} = n \times r_{min_h} \times r_h$ and $\alpha_{max_h} = n \times r_{max_h} \times r_h$, respectively. Similarly, the intake for every single day $d, in(\vec{s}, h, d)$, are also evaluated to be in the established daily ranges:

$$\forall h \in H : \beta_{min_h} \leq in(\vec{s}, h, d) \leq \beta_{max_h} \tag{4.2}$$

where the minimum and maximum daily limits of the feasible space are defined as $\beta_{min_h} = r_{min_h} \times r_h$ and $\beta_{max_h} = r_{max_h} \times r_h$, respectively.

As described, some constraints are considered per day and some other ones are considered globally for the whole menu. In our approach, a menu optimizer requires the definition of an infeasibility degree (id) to compare solutions, which usually considers the relative importance of the constraints. Lets denote by HD the elements that are considered for the daily constraints and HG the ones that are considered globally. Then, the infeasibility degree $id(\vec{s})$ is calculated as follows:

$$id(\vec{s}) = did(\vec{s}) + gid(\vec{s}) \tag{4.3}$$

$$\begin{aligned} did(\vec{s}) &= \sum_{h \in HD} \sum_{d=1}^{n} (max(in(\vec{s}, h, d), \beta_{max_h}) - \beta_{max_h})^2 + \sum_{h \in HD} \sum_{d=1}^{n} (\beta_{min_h} - min(in(\vec{s}, h, d), \beta_{min_h}))^2 \\ gid(\vec{s}) &= \sum_{h \in HG} (max(in(\vec{s}, h), \alpha_{max_h}) - \alpha_{max_h})^2 \times \omega + \sum_{h \in HG} (\alpha_{min_h} - min(in(\vec{s}, h), \alpha_{min_h}))^2 \times \omega \end{aligned} \tag{4.4}$$

In Eqn. (4.3), $did(\vec{s})$ quantifies the infeasibility of daily constraints. Note that the distances to the acceptable ranges are squared in order to heavily penalize the constraints that are strongly unfulfilled. Finally, $gid(\vec{s})$ quantifies the infeasibility of global constraints. As in previous case, distances are squared. Furthermore, since global constraints are considered to be more important, the result is multiplied by ω. In this thesis the value $\omega = 10^6$ is taken into account.

4.2.1 Menu Planning Formulation for School Cafeterias

The main goal of this formulation is to automatically design meal plans for school cafeterias; therefore, only lunch menus are considered through the plan, which excludes all other daily meals. Moreover, the plans are designed for large groups of children, so specific user preferences or food incompatibilities are not considered for the automatic generation of menus. However, from the obtained plans, the nutritionist(s) would be able to apply a minimum set of changes to avoid possible allergies, intolerance, or food incompatibilities.

In this formulation, it was necessary to design a database of courses and ingredients with information on their respective nutritional specifications and nutritional advice. To this end, the *Spanish Database of Food Composition* have been taken into account. Note that this database was developed with *the Spanish Agency for Food Safety and Nutrition*, approved by the Ministry of Science and Innovation of the Government of Spain.

This database contains, in a very complete way, nutritional information of hundreds of foods per 100 grams of edible product. This formulation also considers several consensus documents and guides on school diets and allergens endorsed by the Spanish Government *Ministry of Education, Culture and Sports* and the *Ministry of Health, Social Services and Equality*. Similarly, general recommendations on the intake of nutrients —- given in the *White Book on Child Nutrition*– were considered. The recommended daily amount of nutrients for children between 4 and 12 years old does not differ significantly by gender, and in practice, such a distinction is not made in school cafeterias. The set of nutrients and corresponding ranges considered in this work are presented in Table 4.1.

Finally, we need to define the optimization goal for the MPP, where a solution consists of a meal plan in which there are as many meals –consisting of a starter, main course and dessert– as there are days for which the plan is going to be designed. In school cafeterias, meal prices should be reasonable, so we have fixed the menu price as a goal to minimize. In fact, cost is one of the most common optimization objectives regarding the MPP literature. Note at this point that the total cost of a meal plan is calculated as the sum of the costs of all its meals. Moreover, the cost of a meal is calculated as the sum of the costs of all ingredients –considering their respective amounts– required to prepare that meal.

The total cost of a meal plan for n days is calculated as follows:

$$C(\vec{s}) = \sum_{j=1}^{n} c_{st_j}(\vec{s}) + c_{mc_j}(\vec{s}) + c_{ds_j}(\vec{s}) \tag{4.5}$$

where $c_{st_j}(\vec{s})$, $c_{mc_j}(\vec{s})$, and $c_{ds_j}(\vec{s})$ are the costs of the starter, main course, and dessert, respectively, for day $j \in \{1,...,n\}$. According to Eqn. (4.3) and Eqn. (4.5), the mathematical formulation of this optimization problem is defined as follows:

$$\underset{\vec{s}}{\text{minimize}} \quad C(\vec{s}) + \lambda \times id(\vec{s}) \tag{4.6}$$

where n is the number of days, and the penalization factors are defined as $\omega = 10^6$, $\lambda = 10^4$.

4.2.2 Menu Planning Formulation for CIANNA

One of the most popular formulations of the MPP considers the total cost of the menu plan as the main objective to optimize. However, this depends on the user's requirements; for instance, a different objective might integrate some features such as preferences for certain foods, the level of acceptance, and the time of cooking. The formulation described in this section is designed to consider the level of repetition for specific courses, and the level of repetition for food categories. The motivation behind the formulation is to promote a varied plan from a nutritional point of view and to avoid excessive repetition of some foods. The main goal of this formulation is to automatically design meal plans for a government shelter; therefore, all meals of the day are considered through the plan. Specifically, six meals are considered which are the breakfast, morning snack, starter (lunch), main course (lunch), evening snack and dinner. Also, note that two options for the starter and the main course must be provided on each day.

Following the specifications of the nutritionists, this optimizer is the core part of a desktop application, in which two files must be provided by the user: the database of dishes (with their nutrients intake) and nutrients constraints (minimum and maximum amounts). Table 4.2 shows an example of the nutrient restrictions that the user must provide. The level of repetition can be interpreted as a score which evaluates the repetitions of

Table 4.1 Minimum and Maximum Ranges for Nutrients Intake

Nutrient	r_{min}	r_{max}	r_h
Energy (per day)	0.85	1.15	1012
Fats (per day)	0.75	1.25	31.72
Proteins (per day)	0.75	1.25	27.08
Energy (*n* days)	0.90	1.10	1012
Fats (*n* days)	0.90	1.10	31.72
Proteins (*n* days)	0.90	1.10	27.08
Folic acid	0.70	1.30	135
Phosphorus	0.70	1.30	562.5
Magnesium	0.70	1.30	112.5
Selenium	0.70	1.30	25.75
Sodium	0.70	1.30	870
Vitamin A	0.70	1.30	450
Vitamin B1	0.70	1.30	0.41
Vitamin B2	0.70	1.30	0.63
Vitamin B6	0.70	1.30	0.54
Vitamin B12	0.70	1.30	2.28
Vitamin C	0.70	1.30	27
Vitamin D	0.70	1.30	4.65
Vitamin E	0.70	1.30	6.3
Iodine	0.70	1.30	67.5
Zinc	0.70	1.30	6.75

courses through the meal plan. This level of repetition, which has to be minimized, is defined as follows:

$$L_{Rep}(\vec{s}) = L_{Meals}(\vec{s}) + L_{Cat}(\vec{s}) \tag{4.7}$$

where $\vec{s}$ is a solution, $L_{Rep}(\vec{s})$ is the level of repetition, $L_{Meals}(\vec{s})$ is the level of repetition regarding the meals and $L_{Cat}(\vec{s})$ is the level of repetition regarding the categories.

Following the requirements of the nutritionist, for each day two options of starters and main courses should be part of the menu. A simple way to integrate two meals options in one day and to maintain the previous definition of feasibility is by considering two configurations per day. Thus, some meals are shared by both configurations only leaving starters and main courses different in each configuration. $L_{Meals}(\vec{s})$ is defined in Eqn. (4.8) where $S1$ and $S2$ are the set of meals that are present in each configuration, d'_i is a measurement that assess the variability for each meal and for each configuration. In particular, each configuration is conformed by $S1 = \{$ breakfast, snack morning, starter1, main course1, snack evening, dinner $\}$ and $S2 = \{$ breakfast, snack morning, starter2, main course2, snack evening, dinner $\}$.

$$L_{Meals}(\vec{s}) = \sum_{m \in S1} d'_m(\vec{s}) + \sum_{m \in S2} d'_m(\vec{s}) \tag{4.8}$$

The measurement $d'_m(\vec{s})$ can be any metric that induces variety of items. The intention is to maximize the minimum number of days in which a course is repeated and especially the repetition of meals in close days. Note that this measurement is calculated for each meal, resulting in a trade-off between meals; for instance, the minimization of repetition for starters might induce a maximization of repetition for main courses, and the other

Table 4.2 Minimum and Maximum Ranges for Nutrients Intake

TYPE OF NUTRIENT	NUTRIENT	MIN	MAX
DAILY	KILOCALORIES	2000	2350
DAILY	FIBER G	15	30
DAILY	SATURATED FATS G	0	7
DAILY	CHOLESTEROL MG	50	455
DAILY	SODIUM MG	1000	4000
DAILY	VEGETABLE EQUIVALENT	1	3
DAILY	FRUIT EQUIVALENT	2	4
DAILY	CEREAL WITHOUT FAT EQUIVALENT	8	9
DAILY	CEREAL WITH FAT EQUIVALENT	2	3
DAILY	LEGUMES EQUIVALENT	1	2
DAILY	VERY LOW IN FAT EQUIVALENT	1	2
DAILY	LOW IN FAT EQUIVALENT	1	2
DAILY	MODERATED IN FAT EQUIVALENT	1	2
DAILY	HIGH IN FAT EQUIVALENT	1	2
DAILY	FULL CREAM MILK EQUIVALENT	1	2
DAILY	MILK WITH SUGAR EQUIVALENT	0	1
DAILY	FATS AND OILS EQUIVALENT	3	4
DAILY	FATS AND OILS WITH PROTEINS EQUIVALENT	1	2
DAILY	SUGAR WITHOUT FAT EQUIVALENT	3	4
DAILY	SUGAR WITH FAT EQUIVALENT	0	1
GLOBAL	IRON MG	11	18
GLOBAL	CALCIUM MG	750	1300
GLOBAL	FOLIC ACID G	150	800

way around. However, this strongly depends on the database and the constraints provided by the nutritionist. The formulation prioritizes meals according to the priorities specified in Table 4.3, where meals are sorted according to their priorities from less important to more important. Note that the bottom meals are the most important meals (i.e. Main Course). The measurement $d'_m(\vec{s})$ is formally defined in Eqn. (4.9), where l_m and u_m are the minimum and maximum values of a meal m (Table 4.3). Similarly, n denotes the number of days, and $d_m(\vec{s})$ represents a metric that considers the lowest number of days that passed between two intakes of the meal m. Eqn. (4.10) formalizes $d_m(\vec{s})$. dcn is the lowest number of days that passed between two intakes of the meal (m) and $count(dcn_m(\vec{s}))$ counts the number of times that such a minimum distance happened. Therefore, a higher value of this metric means a lower level of repetition, which is desired.

$$d'_m(\vec{s}) = l_m + \left(\frac{n - d_m(\vec{s})}{n}\right) \times (u_m - l_m) \tag{4.9}$$

$$d_m(\vec{s}) = dcn_m(\vec{s}) + \frac{n - count(dcn_m(\vec{s}))}{n} \tag{4.10}$$

Similarly, the level or repetition for the categories is calculated. This feature is indicated in the database with an integer which has to be provided by the nutritionist. This information grants flexibility to the user herewise it allows the association of courses as desired. In the sample provided by the experts only two categories are specified (tags $\{1,2\}$). In cases where a course belongs to all the categories a flag with value 0 is assigned. Note that the level of repetition regarding categories has the lowest priority located in the top of Table 4.3 with a suffix "Cat". Equivalently to $L_{meals}(\vec{s})$ the level of repetition is separately calculated by each meal and by each configuration in Eqn. (4.11). Therefore, this metric assess the minimum number of days in which a course of a category is repeated. This is formalized in the Eqn. (4.12) and Eqn. (4.13), where $dcnCat_m(\vec{s})$ assess the lowest dcn of days in which any meal was repeated given a category.

$$L_{Cat}(\vec{s}) = \sum_{m \in S1} dc'_m(\vec{s}) + \sum_{m \in S2} dc'_m(\vec{s}) \tag{4.11}$$

Table 4.3 Ranges Considered for the Normalization. The Suffix “Cat” Refers to Level of Repetition in the Categories and the Remaining is for Courses in General.

Meal	l_i	u_i
Morning Snack Cat, Evening Snack Cat	0	1
Breakfast Cat, Dinner Cat	1	10
Starter Cat	10	10^2
Main Course Cat	10^2	10^3
Morning Snack, Evening Snack	10^3	10^4
Breakfast, Dinner	10^4	10^5
Starter	10^5	10^6
Main Course	10^6	10^7

$$dc'_m(\vec{s}) = l_m + \left(\frac{n - dc_m(\vec{s})}{n}\right) \times (u_m - l_m) \tag{4.12}$$

$$dc_m(\vec{s}) = dcnCat_m(\vec{s}) + \frac{n - count(dcnCat_m(\vec{s}))}{n} \tag{4.13}$$

Following the specifications of nutritionists, there is one more feature that classifies courses into favorite courses and non-favorite. Favorite courses must not be repeated in less than three weeks, and for the non-favorite courses, they should not be repeated in less than four weeks. This classification is integrated within the *dcn* distance; only consider the cases where this distance gives a lower level of repetition than desired.

Finally, the mathematical formulation of this optimization problem is defined in Eqn. (4.14), where the penalization factor of feasibility is defined as $\lambda = 10^{20}$. Notice that these penalization constants are large enough so that the repetition levels' values are lower, so the nutritional requirements are the most important part of the optimization.

$$\underset{\vec{s}}{\text{minimize}} \quad L_{Rep}(\vec{s}) + \lambda \times id(\vec{s}) \tag{4.14}$$

4.3 A Memetic Algorithm for the Menu Planning Problem

This section describes the main procedure of a memetic algorithm which is specifically designed to address the MPP. Note that the procedures presented in this sections can be used to optimize several kinds of MPP with global and daily nutritional constraints.

Our general proposal (Memetic Algorithm for Menu Planning - MAMP) is a first-generation MA that applies Iterated Local Search (ILS) as the improvement strategy. The general algorithm is quite a standard memetic Algorithm (see Algorithm 3). Note that the representation of each solution depends on its formulation, however it can be generalized as follows. Each solution is encoded using an array of $n \times m$ elements, where n is the number of days and m is the number of meals per day, Each day is represented by a contiguous sub-array of meals, i.e. each sub-array represents a set of meals. For instance, an individual $\vec{s}$ can be defined as $\vec{s} = [d_1, d_2, ..., d_n]$, where n is the number of days, and each day is a sub-array of m-elements $d_i = [e_1, e_2, ..., e_m]$. In this way, and under few modifications, different formulations might be integrated to this optimizer. The initialization (line 1) is based on a random uniform selection of menu items. The learning strategy (line 3 and line 9) is a standard iterated local search. Parent selection (line 6) is performed with the binary tournament selection. Then, the mating pool is recombined with any crossover operation (line 7). Note that in the case of applying a steady-state survivor selection, *CP* contains only one individual. Otherwise, it contains *N* individuals.

Algorithm 3 Memetic Algorithm for the Menu Planning - MAMP

1: **Initialization**: Generate an initial population P_0 with p individuals.
2: **Evaluation**: Evaluate all individuals in the population.
3: **Iterated Local Search**: Perform an Iterated Local Search for every individual in the population.
4: Assign $t := 0$
5: **while** (not stopping criterion) **do**
6:     **Mating Selection**: Perform binary tournament selection on P_t in order to fill the mating pool.
7:     **Variation**: Apply crossover operator to the mating pool to create a child population *CP*.
8:     **Evaluation**: Evaluate all individuals in *CP*.
9:     **Iterated Local Search**: Perform an Iterated Local Search for every individual in *CP*.
10:     **Survivor Selection**: Apply a survivor selection strategy taking into account P_t and *CP*
11:     $t := t + 1$
12: **end while**

Finally, a survivor selection strategy is applied (line 10). In order, to fully define our proposal, the details of the crossover operators, iterated local search, and survivor selection are required. These are given in the next subsections.

The survivor selection strategy integrated in this memetic algorithm is a minor modification of the one described in Chapter 3. This strategy (BNP) is described in Algorithm 4. One of the principles of the BNP strategy is to avoid the selection of too close individuals. Specifically, the approach tries to avoid the selection of pairs of individuals that are closer than a D^t value. Since in the initial phases it is important to explore, while in the last phases the procedure should intensify, the D^t value varies during the execution. Particularly, the variable D^t is initialized in base of the content of the first population (line 6), and it is decreased in a linear way.

In each execution of the survivor operator, N individuals from the previous population and offspring are selected to survive (lines 7 to 25). BNP iteratively selects the best element that is not penalized, i.e. those individuals with a distance larger than D^t from the already selected individuals (lines 22 and 23). Then all the remaining individuals with distance at most D^t to the previously selected ones are penalized (lines 10 to 13). If it is not possible to find a non-penalized individual, the individual with the largest distance to the currently selected individuals is taken (line 19 and line 20). Note that the principles of BNP are similar to the ones that guided the design of the *Replacement with Multi-Objective Based Dynamic Diversity Control Strategy* (RMDDC) (Segura et al., 2016a). The main difference is that in RMDDC the same importance is given to quality and diversity, so a multi-objective selection is used, which increases the complexity (both in terms of understanding and computational steps) of the approach.

The survivor operator requires a distance-like function between individuals, which has to be ad-hoc to the problem. Given two individuals $\vec{x} = \{x_1, x_2, ..., x_n\}$ and $\vec{y}_2 = \{y_1, y_2, ..., y_n\}$, where each individual is a multi-set of days, i.e. each individual is conformed by n days and each day (x_i or y_i) is a set of meals, the distance between $\vec{x}$ and $\vec{y}$ is defined in Eqn. (4.15), where the cardinality of the intersection between two multi-sets measures the similarity between $\vec{x}$ and $\vec{y}$. Note that the order of intake is not taken into account for calculating $Dist(\vec{x}, \vec{y})$.

$$Dist(\vec{x}, \vec{y}) = n - |\{x_1, x_2, ..., x_n\} \cap \{y_1, y_2, ..., y_n\}| \tag{4.15}$$

Algorithm 4 BNP Survivor Selection Technique

Input: Population, Offspring

Penalized = $\emptyset$
CurrentIndividuals = Population $\cup$ Offspring
Best = Best Individual in CurrentIndividuals
NewPop = {Best}
CurrentIndividuals = CurrentIndividuals - {Best}
Update D^t taking into account the elapsed time (T_e), stopping criterion (T_s) and initial threshold value of D^t (ITV): $D^t = ITV - ITV * (\frac{T_e}{T_s})$. We took $ITV = 0.5 \times M$, where M is the mean distance in the first population.
while (|NewPop| < N) **do**
for all $I \in CurrentIndividuals$ **do**
I.DCN = distance to the closest individual of I in NewPop
if I.DCN < D^t **then**
Penalized = Penalized $\cup$ {I}
CurrentIndividuals = CurrentIndividuals - {I}
end if
end for
for all $I \in$ Penalized **do**
$I.DCN$ = distance to the closest individual of I in NewPop
end for
if CurrentIndividuals is empty **then**
Selected I with largest $I.DCN$ in Penalized
Penalized = Penalized - { Selected }
else
Selected = Best Individual in CurrentIndividuals
CurrentIndividuals = CurrentIndividuals - {Selected}
end if
NewPop = NewPop $\cup$ {Selected}
end while
return Population = NewPop

4.3.1 Crossover Operators for MPP

This section describes two crossover operators that recombine two solutions to create two more solutions. The first one is the classic Uniform Crossover (UX). This operator is illustrated in Algorithm 5, where n denotes the number of days and m denotes the number of meals per day.

The second operator is called the Similarity Based Crossover (SX). This operator is described in Algorithm 6 and is a newly designed ad-hoc crossover operator. The principle that guided its design is based on the daily constraints. Since there are some daily constraints in the problem, it make sense to inherit complete daily-meals. Thus, candidate solutions might be interpreted as multi-sets of daily-meals. SX identifies the set of daily-meals that are included in both parents and their corresponding cardinality (lines 1 to 10). Then, each common daily-meal is directly copied to both children (lines 11 to 16). Thereafter, for the remaining daily-meals, which appear only in one of the parents, a uniform crossover is used. Specifically, daily-meals are extracted in random order (lines 19 and 21) and with a 0.5 probability they are interchanged prior to including them in the children multi-set. Thus, the main aim of this crossover is to first identify the similarities among parents and then preserve such similarities in the children. Finally, the multi-set of each children is converted to array format (lines 29 and 30). The concept of the SX operator is illustrated in Figure 4.1, where each color represents a daily-meal. Particularly, the daily-meals with red, blue and orange colors are present in both parents, so such colors are inherited to both children, and the remaining colors are assigned randomly.

Since the learning strategy or Iterated Local Search (ILS) varies between formulations, this procedure is detailed later.

Algorithm 5 Uniform Crossover – UX

```
Input: Parent1, Parent2
Child1 = Parent1
Child2 = Parent2
for d in {0,1,...,n−1} do
    if (Random([0,1]) ≤ 0.5 ) then
        for i in {0,1,,...,m−1} do
            Swap(Child1[d × m + i], Child2[d × m + i])
        end for
    end if
end for
```

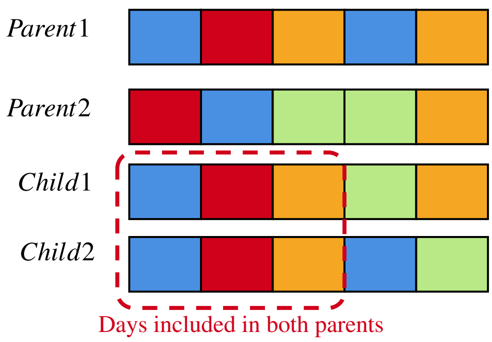

Fig. 4.1 Concept of the Similarity Crossover (SX) operator.

4.3.2 Adaptations of MAMP for the School Cafeterias Formulation

In this section the details for the school cafeteria formulation are given. A solution consists of a meal plan in which there are as many meals –consisting of a starter, main course, and dessert– as there are days for which the plan is going to be designed. Note that in this formulation, the constraints and the objective function (Eqn. 4.5) depend on the complete meals that are selected, but not on the order considered for them. For instance, given a solution $\vec{s}$, interchanging the meals of the first day and second day, does not produce any effect on the calculation of the infeasibility degree and objective function.

Representation of the solutions

Each solution is encoded using an array with $3 \times n$ elements, where n is the number of days to plan. Positions i that fulfill $i \ mod \ 3 = 0$ refer to the starter on day $\lfloor i/3 \rfloor + 1$. Positions i that fulfill $i \ mod \ 3 = 1$ refer to the main course on day $\lfloor i/3 \rfloor + 1$. Finally, the positions i that fulfill $i \ mod \ 3 = 2$ refer to the dessert on day $\lfloor i/3 \rfloor + 1$. Figure 4.2 illustrates an example that consists of two days considering this representation.

Day 1			Day 2		
Starter $0 \ mod \ 3 = 0$	Main Course $1 \ mod \ 3 = 1$	Dessert $2 \ mod \ 3 = 2$	Starter $3 \ mod \ 3 = 0$	Main Course $4 \ mod \ 3 = 1$	Dessert $5 \ mod \ 3 = 2$

Fig. 4.2 Representation of two days in a solution for the first formulation.

Algorithm 6 Similarity Based Crossover – SX

```
Input: P1, P2 // Parents
P1Days = ∅ //multi-set
P2Days = ∅ //multi-set
C1Days = ∅ //multi-set
C2Days = ∅ //multi-set
for day in P1 do
    P1Days = P1Days ∪ {day}
end for
for day in P2 do
    P2Days = P2Days ∪ {day}
end for
for day in ( {P1Days} ∩ {P2Days}) do
    C1Days = C1Days ∪ {day}
    C2Days = C2Days ∪ {day}
    P1Days = P1Days - {day}
    P2Days = P2Days - {day}
end for
commonDays = |C1Days|
for i in {commonDays, ..., n − 1} do
    day1 = P1Days.selectRandom()
    P1Days = P1Days - {day1}
    day2 = P2Days.selectRandom()
    P2Days = P2Days - {day2}
    if (Random([0,1]) ≤ 0.5) then
        Swap(day1, day2)
    end if
    C1Days = C1Days ∪ {day1}
    C2Days = C1Days ∪ {day2}
end for
Child1 = convertToArray(C1Days)
Child2 = convertToArray(C2Days)
```

Iterated Local Search for Feasibility and Costs

Our proposal of this formulation uses a quite simple variant of ILS (Algorithm 7), where in each iteration a local search is applied (line 3). Each neighbor is generated by altering a single menu item. Thus, the number of neighbors of any solution is $n \times (STO + MCO + DSO - 3)$, where STO, MCO, DSO are the number of options for starters, main courses, and desserts, respectively. Neighbors are explored in a random way, and any modification that improves the current solution is accepted. The first-improvement hill climbing stops when a local optimum is reached. Note that, as usual, since a constrained optimization problem is used, the way to compare solutions is the following. Given two solutions $\vec{x}$ and $\vec{y}$, if their infeasibility degree are different, the one with lower infeasibility degree is better. Otherwise, if their infeasibility degree are equal, then the one with a better objective –minimum cost in this case– is better.

The perturbation (line 7) is always applied to the best solution found so far and it operates as follows. In the cases where the solution to perturb contains some days with constraints unfulfilled, all these days are perturbed by reassigning all its meal items randomly. If all the daily constraints are fulfilled, but some global constraints are unfulfilled, the perturbation operates as follows. First, the macronutrient or micronutrient element that contributes higher to the infeasibility degree is detected. If for this element the consumption is higher than the desired, one of the six days with a higher consumption of such element is selected randomly. Similarly, if the consumption is lower than desired, one of the six days with lower consumption of such element is

Algorithm 7 Iterated Local Search for Feasibility and Cost - ILS

input: $\vec{s}$ (Initial Solution), Iterations

$\vec{Best} = \vec{s}$
for i in $\{1,2,...,$Iterations$\}$ **do**
  $\vec{s'}$ = First-Improvement-Hill-Climbing($\vec{s}$)
  if ($\vec{s'}$ is Better than $\vec{Best}$) **then**
    $\vec{Best} = \vec{s'}$
  end if
  $\vec{s}$=Perturb($\vec{Best}$)
end for
return $\vec{Best}$

selected randomly and perturbed by reassigning all its meals items randomly. Finally, for cases in which all the constraints are fulfilled, a single day is selected at random and it is perturbed by reassigning its three meal items randomly.

4.3.3 Experimental Assessment for the School Cafeterias Formulation

This section is devoted to the experimental validation of the memetic algorithm for the MPP. Six different survivor selection strategies have been used to perform the validation. The first five strategies have been described extensively in Section 2.3.3. The first is the generational strategy with elitism (GEN_ELIT). In this case, the whole offspring survives. Additionally, if the best solution of the previous generation is better than any of the offspring, it survives.

Results for three different cases are presented. The same database and nutrition recommendations are used in all of them as previously described. Thus, the only difference is the number of days to plan. Particularly, $n = 20, 40, 60$ were used. The database contains 18 starters, 33 main courses and 13 desserts, which means that the search space size is approximately 10^{77}, 10^{155} and 10^{233}, respectively.

Since all the proposals are stochastic, each execution was repeated 30 times. All experiments were performed by setting a common stopping criterion for all the algorithms. In the cases of using $n = 20$, the stopping criterion was set to 1 hour, whereas it was increased to 2.5 and 5 hours, for the cases $n = 40$ and $n = 60$, respectively. All the proposals were able to generate feasible solutions in all the iterated executions. Thus, comparisons are performed in base of the attained cost. In order to compare the different methods, the series of statistical tests described in (Segura et al., 2015c) were used. This guideline takes into account the ANOVA, *Shapiro-Wilk, Levene, Welch and Kruskal-Wallis* tests. They were applied assuming a significance level of 5%.

Note that six different survivor selection strategies and two different crossover operators were implemented. Thus, the study involves the application of twelve algorithms. In order to refer to each method, a label with two parts separated with a dash is used. The first part denotes the survivor selection strategy, whereas the second part denotes the crossover strategy. Some survivor selection strategies require additional parameters. They were established considering some preliminary experiments, and the final parameterization applied is shown in Table 4.4. The only additional parameter, which was required by all proposals, is the population size, and the iterations per local search. It was set to 10 individuals and 100 iterations with the aim of reducing the time required by the executions. Preliminary experiments showed that higher values achieve similar results, but require additional time.

Table 4.5 show a summary of the results attained by each proposal. In particular, the best (minimum), worst (maximum) and mean cost are shown for the three different instances and for the twelve algorithms. To

Table 4.4 Parameterization of the Survivor Selection Strategies

Survivor Selection	Parameterization
BNP	ITV=0.5
RMDDC	ITV=0.5
HGSADC	$N_{Close} = 3$, $N_{Elit} = 3$
RTS	CF=5
CDRW	No parameters required
GEN_ELIT	No parameters required

facilitate analysis, pairwise statistical analyzes among all algorithms were performed using the set of statistical tests mentioned above. Table 4.6 shows a summary of the results of the statistical tests. Columns labelled with a $\uparrow$ show the amount of times that each algorithm was the winner in the pair-wise statistical tests. The times each algorithm loses are shown in the columns labeled with a $\downarrow$. whereas the columns tagged with $\leftrightarrow$ refer to the number of times when the differences were not statistically significant. Finally, the score is calculated as the times than an algorithm wins minus the times that an algorithm loses, and it gives and overall performance measure of each proposal.

Taking into account the results obtained in the “20 Days” instance, several of the algorithms are similar. In fact, eight algorithms achieved the same results in all executions. The proposals that used the CDRW or GEN_ELIT as replacement strategy were the only ones that reported some not so high-quality results in some executions. The overall score shows a tie among the remaining eight algorithms. This means that for small cases using an ad-hoc crossover operator is not really important for the performance, and any strategy to delay convergence is good enough to reach high-quality solutions.

In the case of the “40 Days” instance, larger differences appear among the proposals. In this case, the BNP-SX strategy is the one that achieves the lowest mean. Moreover, statistical tests report that it is superior to most alternatives. In fact, only when compared to RMDDC-SX, differences were not statistically significant and its score is the highest one. It is also remarkable that in this case, using and ad-hoc crossover operator such as the SX contributes importantly to the performance. The sum of the scores of the methods that apply SX is 6, whereas the sum of the methods that apply UX is -6. Thus, to attain proper solutions, it is important to select both a proper replacement strategy and crossover operators.

Finally, in the case of the “60 Days” instance, the BNP-SX strategy is again the one that attains the lowest mean. However, in this case, statistical tests show that RMDDC-SX and HGSADC-SX achieve the same score. In this case, differences in the performance among those proposals that apply SX and those that apply UX are clearer. Particularly, the sum of the scores of the methods that apply SX is 22 and for those that apply UX is -22. Thus, in order to attain scalable algorithms, ad-hoc crossover operators are required. Again, for this instance, results show that both a proper replacement and an ad-hoc operator are required to develop robust optimizers capable of attaining high-quality solutions in most of the executions.

Taking into account all the results, the BNP-SX and RMDDC-SX strategies were the clear winners, with a slight superiority for the BNP-SX strategy. The common feature of BNP and RMDDC is that they relate the amount of diversity maintained in the population to the stopping criterion. In fact, they are the only methods that start with a population with a high level of diversity which is explicitly and gradually decreased during the whole execution.

Figure 4.3 shows the evolution of diversity for each method considering each problem and with the SX operator. In the three problems, it is clear that RMDDC-SX and RTS-SX maintained higher levels of diversity along the execution, which is not the best behavior for short runs. Although BNP-SX always begins with high levels of diversity, the decrease is faster than that in RMDDC-SX. Note that GEN_ELIT-SX, CDRW-SX and

Table 4.5 Summary of the Cost Attained by the Different Proposals

	20 Days			40 Days			60 Days		
	Best	Worst	Mean	Best	Worst	Mean	Best	Worst	Mean
BNP-SX	20.265	20.265	20.265	40.038	40.067	40.059	59.721	59.788	59.733
BNP-UX	20.265	20.265	20.265	40.038	40.282	40.072	59.721	60.189	59.832
RMDDC-SX	20.265	20.265	20.265	40.038	40.282	40.068	59.721	59.941	59.738
RMDDC-UX	20.265	20.265	20.265	40.038	40.427	40.098	59.721	60.512	60.118
HGSADC-SX	20.265	20.265	20.265	40.038	40.339	40.088	59.721	59.942	59.740
HGSADC-UX	20.265	20.265	20.265	40.038	40.282	40.080	59.721	60.370	59.846
RTS-SX	20.265	20.265	20.265	40.038	40.339	40.151	59.721	59.960	59.784
RTS-UX	20.265	20.265	20.265	40.067	40.427	40.154	59.721	60.370	59.873
CDRW-SX	20.265	20.460	20.297	40.067	40.434	40.260	59.721	60.543	59.813
CDRW-UX	20.265	20.460	20.271	40.067	40.409	40.183	59.721	60.189	59.852
GEN_ELIT-SX	20.265	20.469	20.297	40.067	40.434	40.249	59.721	59.942	59.779
GEN_ELIT-UX	20.265	20.460	20.290	40.038	40.434	40.233	59.721	60.105	59.843

Table 4.6 Statistical Comparison of the Different Proposals for the Three Instances

	20 Days				40 Days				60 Days			
	↑	↓	↔	Score	↑	↓	↔	Score	↑	↓	↔	Score
BNP-SX	3	0	8	3	10	0	1	10	9	0	2	9
BNP-UX	3	0	8	3	6	1	4	5	1	3	7	-2
RMDDC-SX	3	0	8	3	8	0	3	8	9	0	2	9
RMDDC-UX	3	0	8	3	4	2	5	2	0	11	0	-11
HGSADC-SX	3	0	8	3	5	1	5	4	9	0	2	9
HGSADC-UX	3	0	8	3	5	2	4	3	1	3	7	-2
RTS-SX	3	0	8	3	3	3	5	0	1	3	7	-2
RTS-UX	3	0	8	3	2	5	4	-3	1	4	6	-3
CDRW-SX	0	8	3	-8	0	8	3	-8	1	3	7	-2
CDRW-UX	0	0	11	0	0	6	5	-6	1	3	7	-2
GEN_ELIT-SX	0	8	3	-8	0	8	3	-8	2	3	6	-1
GEN_ELIT-UX	0	8	3	-8	0	7	4	-7	1	3	7	-2

HGSAD-SX converged at the 20% of the total time in all the problems, as a result these methods lose their exploration capabilities.

4.3.4 Adaptation of MAMP for the CIANNA's Formulation

This section describes some additional details of the CIANNA's formulation as well as the iterated local search procedure used to intensify. In this formulation all the meals of the day are considered. Thus, a solution consists of a meal plan in which there are as many meals as days. Specifically, a day consists of a breakfast, morning snack, starter (lunch), main course (lunch), evening snack, and dinner. Note that for simplicity, the starter and main course are considered independently, so in this context each one can be interpreted as a meal. As previously explained, the meal plan must integrate two options for starters and main courses. Therefore, a day consists of eight meals, which are breakfast, morning snack, starter1, starter2, main course1, main course2, evening snack and dinner. However, the constraints must be fulfilled and evaluated independently for each option, that is, a feasible solution must be obtained for each association of meals: { breakfast, morning snack, starter1, main course1, evening snack, dinner }, and { breakfast, morning snack, starter2, main course2, evening snack, dinner }. Note that in this formulation, the objective function depends on the selected meals and in the order in which they are selected. Thus, given a solution $\vec{s}$, interchanging the meals of two different days might affect the objective function.

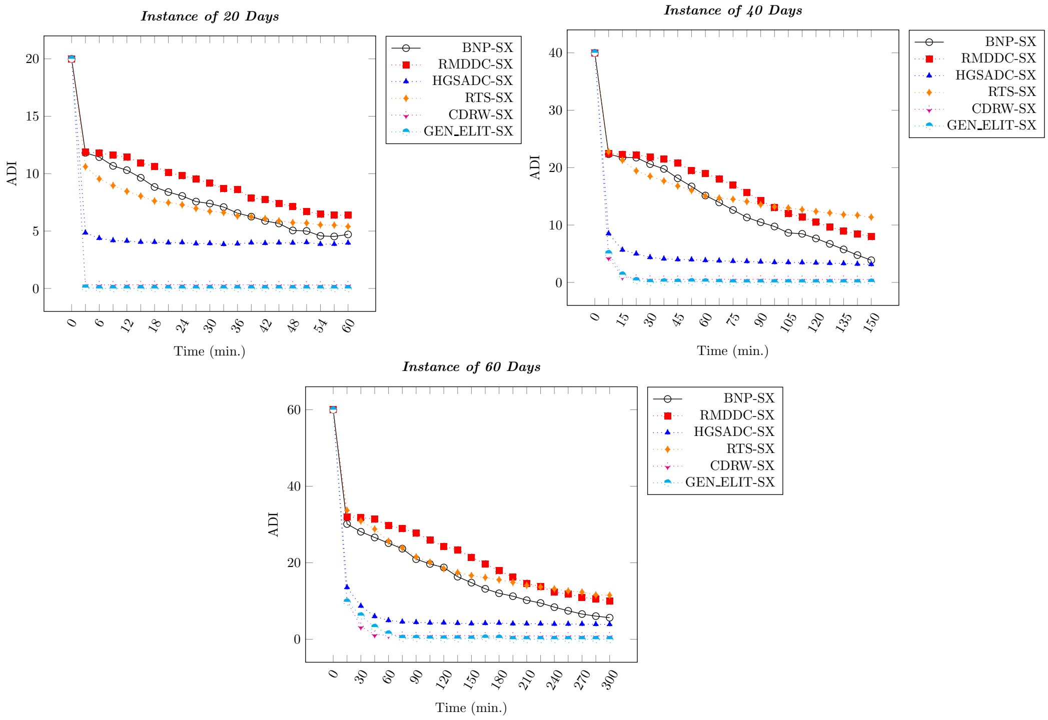

Fig. 4.3 Evolution of the diversity of the survivor selection methods and the SX operator. 20 days (top-left), 40 days (top-right) and 60 days (bottom-middle).

Breakfast	Morning Snack	Starter1	Starter2	Main Course1	Main Course2	Evening Snack	Dinner
$i \ mod \ 8 = 0$	$i \ mod \ 8 = 1$	$i \ mod \ 8 = 2$	$i \ mod \ 8 = 3$	$i \ mod \ 8 = 4$	$i \ mod \ 8 = 5$	$i \ mod \ 8 = 6$	$i \ mod \ 8 = 7$

Fig. 4.4 Representation of a solution for the second formulation (one day).

Representation of the solutions

Each solution is encoded using an array with $8 \times n$ elements, where n is the number of days to plan. Positions i that fulfill *i mod 8 = 0* refer to the breakfast on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 1* refer to the morning snack on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 2* refer to the starter1 on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 3* refer to the starter2 on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 4* refer to the main course1 on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 5* refer to the main course2 on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 6* refer to the evening snack on day $\lfloor i/8 \rfloor + 1$. Positions i that fulfill *i mod 8 = 7* refer to dinner on day $\lfloor i/8 \rfloor + 1$. This is illustrated in figure 4.4, where the meals of a day are listed.

Iterated Local Search for Feasibility and Level of Repetition

The ILS designed for this formulation can be considered as an extension of the one described in Section 4.3.2. This extension integrates two additional mechanisms to increase the quality of individuals. At this point, the reader might have noticed that considering eight meals for a day might impact the performance of the algorithm in terms of scalability. In the previous formulation, the number of variables (calculated as $n \times 3$) for days 20, 40, and 60 were 60, 120 and 180, respectively. However, the performance of some optimizers might deteriorate

Algorithm 8 Iterated Local Search for Feasibility and Repetition Level - ILS

input: $\vec{s}$ (Initial Solution), Iterations
1: $\vec{Best} = \vec{s}$
2: **for** i in $\{1, 2, ..., \text{Iterations}\}$ **do**
3: $\quad \vec{s'}$ =First-Improvement-Hill-Climbing($\vec{s}$)
4: $\quad \vec{s^*}$=First-Improvement-Hill-Climbing-Swaps($\vec{s'}$)
5: $\quad$ **if** ($\vec{s^*}$ is Better than $\vec{Best}$) **then**
6: $\quad\quad \vec{Best} = \vec{s^*}$
7: $\quad$ **end if**
8: $\quad \vec{x}$=Perturb($\vec{Best}$)
9: $\quad \vec{s}$=Local-Search-Per-Day($\vec{x}$)
10: **end for**
11: **return** $\vec{Best}$

as the size of an individual is increased. Moreover, the convergence of the methods might be very slow, and in some cases a feasible solution might be not attained for short runs. Therefore, finding a feasible solution might be a concern, as a matter of fact for this formulation the sizes of a solution (calculated as $n \times 8$) considering 20, 40 and 60 days are 160, 320, and 480, respectively. A way to alleviate this issue is by applying an ILS (Algorithm 7) but only considering the daily constraints. Specifically, after the perturbation considered in the local search. Note that this local search per day can be implemented very efficiently because the contribution of each nutrient can be calculated directly as a difference with the daily infeasibility degree ($in(\vec{s}, h, d)$ - Eqn. 4.2)[3]. Therefore, finding a day that fulfills daily constraints can be quite fast.

Additionally, since this formulation is part of a desktop application and the optimizer has to provide feasible and high-quality solutions for very short runs, additional movements to boost the repetition level are considered. These movements can be promoted with an additional local search. Therefore two local searches are integrated in the main procedure of the ILS, which are the *First-Improvement-Hill-Climbing* and the *First-Improvement-Hill-Climbing-Swaps*. The former has been explained in section 4.3.2, where each neighbour is generated by altering a single menu item. Thus, the number of neighbors of any solution is $n \times (BF + SM + ST1 + ST2 + MC1 + MC2 + SE + DN - 8)$, where BF, SM, $ST1$, $ST2$, $MC1$, $MC2$, SE, DN are the number of options for breakfast, snack morning, starter1, starter2, main course1, main course2, snack evening and dinner, respectively. Note that the neighbors are explored in a random way, and any modification that improves the current solution is accepted. In this local search if a solution is feasible then the level of repetition is considered. Thus, this process allows the integration of new items (meals) that improve the level of repetition.

In the second local search (Algorithm 9) each neighbor is generated by interchanging two single days (line 7). This local search is applied to the solution until there is no improvement in terms of repetition level (line 3). Thus, the number of neighbors of any solution is $n \times (n-1)$, where n is the number of days. Note that this local search is no longer required to update the feasibility degree, since interchanging days does not affect the feasibility. The ILS for feasibility and repetition level is described in Algorithm 8, which integrates the local search *First-Improvement-Hill-Climbing-Swaps* (line 4) and also improves a perturbed solution with an ILS per day (line 9). In preliminary analysis, this modified ILS allowed us to find feasible solutions of cases with 60 days (480 items) in less than ten seconds.

The crossover operator devised for this formulation is an extension of the Similarity Based Crossover (SX). In this formulation, the order of the days is relevant, therefore SX has to preserve the order of days in some way. In this extension each common day is directly copied to both children in the same positions. In addition, the remaining days, which appear only in one of the parents, are assigned to the children in the unassigned positions. This principle preserves the order of the days and in cases where a parent is paired with itself produces the

[3]This technique is better known as incremental evaluation.

Algorithm 9 First-Improvement-Hill-Climbing-Swaps

```
input: s⃗ (Initial Solution), Iterations
Best⃗ = s⃗
improves=true
while improves do
    improves=false
    for i in {1,2,...,n} do
        for j in {i+1,i+2,...,n} do
            swap(s_i, s_j)
            s⃗=Calculate-Repetition-Level(s⃗)
            if (s⃗ is Better than Best⃗) then
                Best⃗ = s⃗
                Improves=True
            else
                swap(s_i, s_j)
            end if
        end for
    end for
end while
return Best⃗
```

same individuals. Note that this last situation induces intensification, because in such cases ILS is applied to the same solution several times. This is illustrated in Figure 4.5, where each day is represented by a color, the rectangles with gray border (blue, red and orange) in the children represents the days that are present in both parents whose position is preserved in both children.

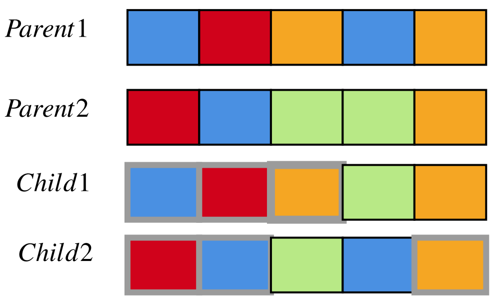

Fig. 4.5 Example of Similarity Crossover for Level of Repetition (SX). The days of a created solution with gray border are in both parents.

4.3.5 Experimental Assessment for CIANNA's Formulation

This section presents the experimental assessment of the Memetic Algorithm for the Menu Planning where the objective is the level of repetition of meals. Following the results obtained in Section 4.3.3, this section only takes into account BNP as a survivor strategy. Moreover, the parameterization of the BNP survivor strategy is analyzed. In this experimental validation an example of a database and nutrition constraints is taken into account. Note that both files must be provided by the nutritionist. Specifically, three different cases were tested where the only difference is the number of days to plan. In particular, $n = 20, 40, 60$ days were considered. In this case the database contains 79 breakfasts, 37 morning snacks, 20 starters, 51 main courses, 49 evening snacks and 59 dinners, meaning the search space size is about $(79 \times 37 \times 20 \times 20 \times 51 \times 51 \times 49 \times 59)^n \approx 10^{12.94 \times n}$, resulting with $10^{258.8}$,$10^{517.7}$, and $10^{776.6}$ for 20, 40, and 60 days, respectively.

Since the proposal is stochastic, each execution was repeated 35 times. All experiments were performed by setting a common stopping criterion, which is set to 5 hours. Note that this amount of time is enough to attain feasible solutions. The population size was set to 10 individuals. The total time considered for each ILS

Table 4.7 Summary of the Three Instances with Several Crossover Probabilities

	20 Days			**40 Days**			**60 Days**		
Crossover Probability	**Best**	**Worst**	**Mean**	**Best**	**Worst**	**Mean**	**Best**	**Worst**	**Mean**
0.0	8231940.67	8240963.20	8236076.29	9731462.68	9774736.37	9740487.25	10246485.16	10290554.63	10269241.71
0.2	8231580.58	8240195.81	8235416.45	9731215.74	9776137.02	9739333.12	**10233129.64**	10289394.64	**10265204.05**
0.4	8231580.58	8240285.98	8235209.07	9731266.95	9762316.93	9737245.14	10244736.90	10301144.90	10277009.99
0.6	8230318.44	8240603.17	8235882.92	9731215.73	9758991.44	**9736906.60**	10249329.64	10299097.15	10274015.48
0.8	8230318.44	8239230.67	**8235170.42**	9731834.51	9768583.76	9741324.00	10264637.38	10316144.39	10284395.12
1.0	**8230048.44**	8240940.58	8235693.25	**9731153.31**	9780383.87	9748675.83	10265374.63	10360796.63	10300944.10

application was as follows. For each run, a random individual is generated and the time required to achieve a feasible solution is recorded ($Time_{FS}$). Then, the total time considered in each ILS application is considered to be ten times the value of $Time_{FS}$.

4.3.6 On the Effect of the Crossover Probability

This section analyzes the effect of the probability of crossover for the three instances. Note that memetic algorithms does not apply a mutation operator, and instead it integrates a local search. In order, to have a better understanding of the proposed algorithm six parameterizations of crossover probabilities were tested. Specifically, the values tested were {0.0, 0.2, 0.4, 0.6, 0.8, 1.0}. Table 4.7 shows a summary of the results attained by each crossover probability. Particularly, the best (minimum), worst (maximum) and mean of the level of repetition is shown for the three instances and for six configurations. For columns "Best" and "Mean" the higher scores are in bold. A lower value of crossover probability induces higher levels of intensification since in average a lower amount of individuals are recombined. For instance a crossover probability equal to 0 does not apply a recombination to the individuals, as an result a ILS applied to the same individuals. In comparison, higher probability values promote higher levels of exploration, which might slow down the convergence of the population. The 20-day (160 variables) instance achieved better results with higher probability values; differently, in the 60-day (480 variables) instance the best results were reported with a probability of 0.2. Interestingly, for all size problems, a probability value equal to 0 was never reported as the best mean, showing the effectiveness of the SX. This result is expected in the sense that for a very large search space, a large computational effort is required even to obtain acceptable local optima, so more intensification is usually required.

Note also that these results suggest that the three instances are quite challenging because of the difference reported between the best and the mean. Therefore, for these problems, the results might be improved extending the stopping criterion, particularly in the instance of 60 days. In the other formulation every execution had obtained similar results, so considering the level of repetition seems to highly complicate solving the problem.

4.3.7 Analysis of the Initial Threshold Value

This section is devoted to analyze the performance of the BNP when different ITV values are used. Note that the maximum distance is the number of days. Additionally, when ITV is set to 0, no individuals are penalized so the BNP has a behavior somewhat similar to GEN_ELIT. In order, to better understand the effect of the ITV, several values of it with three crossover probabilities were tested. Specifically, the values $ITV = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and crossover probabilities $Px = \{0.2, 0.6, 0.8\}$ were tested. Note that these probabilities are the ones that achieved the best results in the previous section. The stopping criterion was set to 5 hours for all instances. Figure 4.6 shows the box-plots of the repetition level for each instance with several crossover probabilities. For the problem that consists of 20 days (top-left), the best range of ITV is defined in $[0.2, 0.6]$ where the median of the box-plots is similar ≈ 8235000. In the case of the instance with 40 days (top-right), the advantages of promoting diversity are sightly lower; however, there is still an improvement in the same range. Note that in problems of 20 and 40 days the worst parameterization is $ITV = 0$ where diversity is not promoted. Promoting more exploration with high values of ITV also affects the performance of the algorithm. Deteriorated results with ITV values in the previous range might be an indicator that the stopping criterion is not large enough. In those cases, extending the execution time might highlight the benefits of maintaining diversity, which is the case for the problem with 60 days (bottom-middle). Moreover, in those cases it is not necessary to promote diversity, instead a higher level of intensification should be promoted. Note that for the 60-day problem, the best results are reported with lower values of crossover probabilities. Applying lower probability values induces more intensification because the ILS is applied to most similar individuals. Note that a similar situation arises with the problem of 40 days, that is, some of the best results are found with lower crossover probabilities. However, the crossover probability has a greater effect than the ITV values in the most challenging problems and for short runs. This can be seen in the problem with 60 days, where values of ITV in $[0.0, 0.6]$ achieve similar results, but the effect of the probability has a significant effect on their performance. Thus, in cases where the problem is too difficult and the stopping criterion cannot be extended, it might be better to test lower crossover probability values.

This analysis suggests that running the desktop application for long-term (e.g. on days or weeks) might be beneficial for menu plans with sizes greater than 20 days, which makes sense since a menu plan with a size of two months would worth it.

4.3.8 Analysis of the Stopping Criterion

Usually, the strategies that explicitly maintain diversity have the effect of slowing down the convergence of the population. This tendency is more present in problems with a large number of variables. This might be the case for the 60-day problem and for short runs. This section analyzes the performance of our method with three stopping criteria. Specifically, 35 runs were tested for 360 minuter (5 hours), 1440 minutes (24 hours), and 2280 minutes (48 hours). Figure 4.7 reports the box-plots of the three problems with four configurations. These four configurations are $ITV = \{0.2, 0.4\}$ and $Px = \{0.2, 0.6\}$. In all the configurations, increasing the stopping criterion improved the performance of the methods in terms of level of repetition. For example, in the problem of 20 days, the configuration that achieved the best level of repetition is with $\{Px = 0.6, ITV = 0.2\}$, and the worst results with respect to the median belong to the configuration $\{Px = 0.2, ITV = 0.4\}$. Note that as the stopping criterion is extended the variability of the results are lower, which is especially the case for the problem of 40 days, where the best level of repetition among all the configurations is close to 9731153.303 Finally, for the problem of 60 days the configuration that achieve the best results is $\{PX = 0.2, ITV = 0.2\}$, in fact lower probability values reported the best level of repetition. These findings suggest that for a menu plan that consists on more than 40 days, extending the stopping criterion or applying low probabilities is better. However, this optimizer is still a good choice for very short runs, since it is able to create feasible solutions very fast. Note that the required time to attain a feasible solution depends on the number of days. To better understand the time required to create a feasible solution, Table 4.8 reports the mean time to achieve a feasible solution for different number of days. Note that each day configuration (row) was taken from averaging 35 runs. Thus, the time required to achieve a feasible solution is not a concern. Also note that the average levels of repetitions are far from the values reported by our proposal.

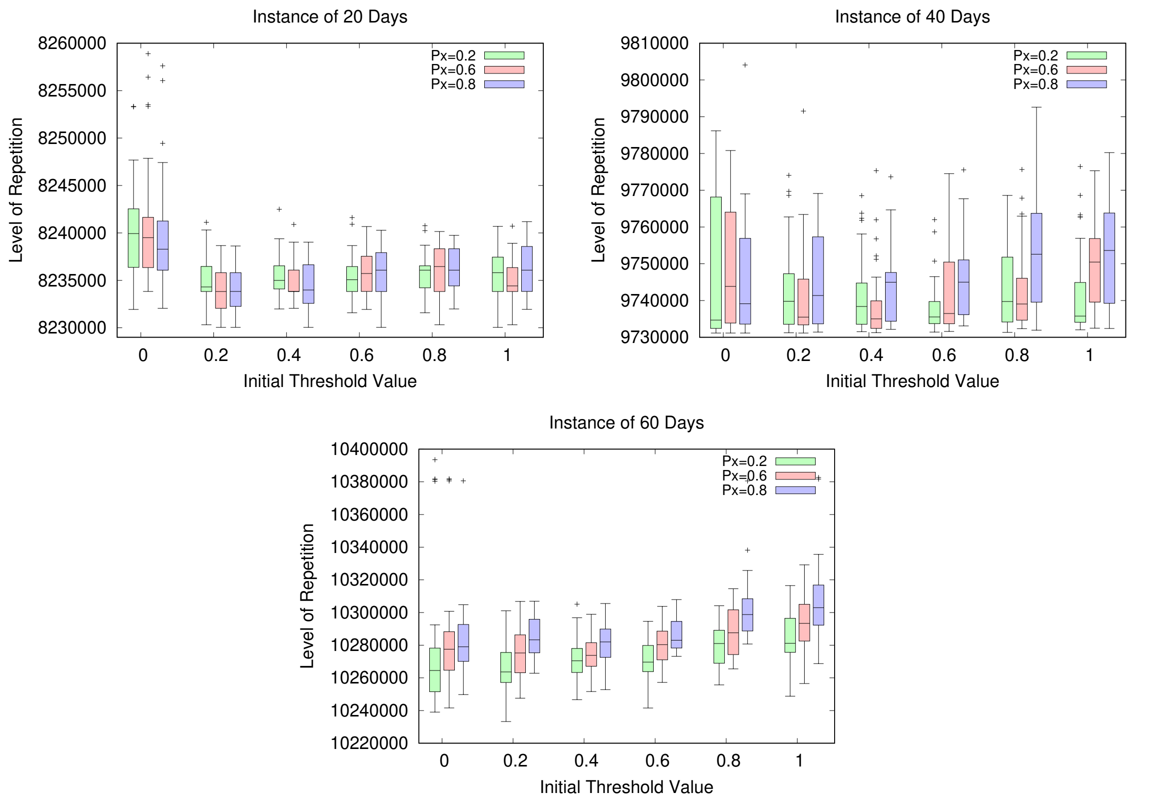

Fig. 4.6 Box-plots of the Level of Repetition for 35 runs for instances with 20, 40, and 60 days. Three crossover probability values (Px) are reported with several initial threshold values.

Table 4.8 Average Time to Achieve a Feasible Solution for Different Days

Days	Time (Sec.)	Repetition Level
20	0.56	10737733.85
40	1.16	11001240.70
60	1.75	11088443.11
80	2.56	11122154.46
100	3.01	11137855.89

4.3.9 Menu Planning Problem in a Desktop Application

This section describes the desktop application developed to deal with the CIANNA's formulation. The main goal of this project was to develop a desktop software that automatically assigns a healthy menu plan with a lower level of repetition. Note that the nutritionists must establish some characteristics of the menu plan, as are the number of days and the nutritional constraints.

Since the amount of data is limited, this desktop application does not integrate database managers; instead, the process considers Excel files to store the database.

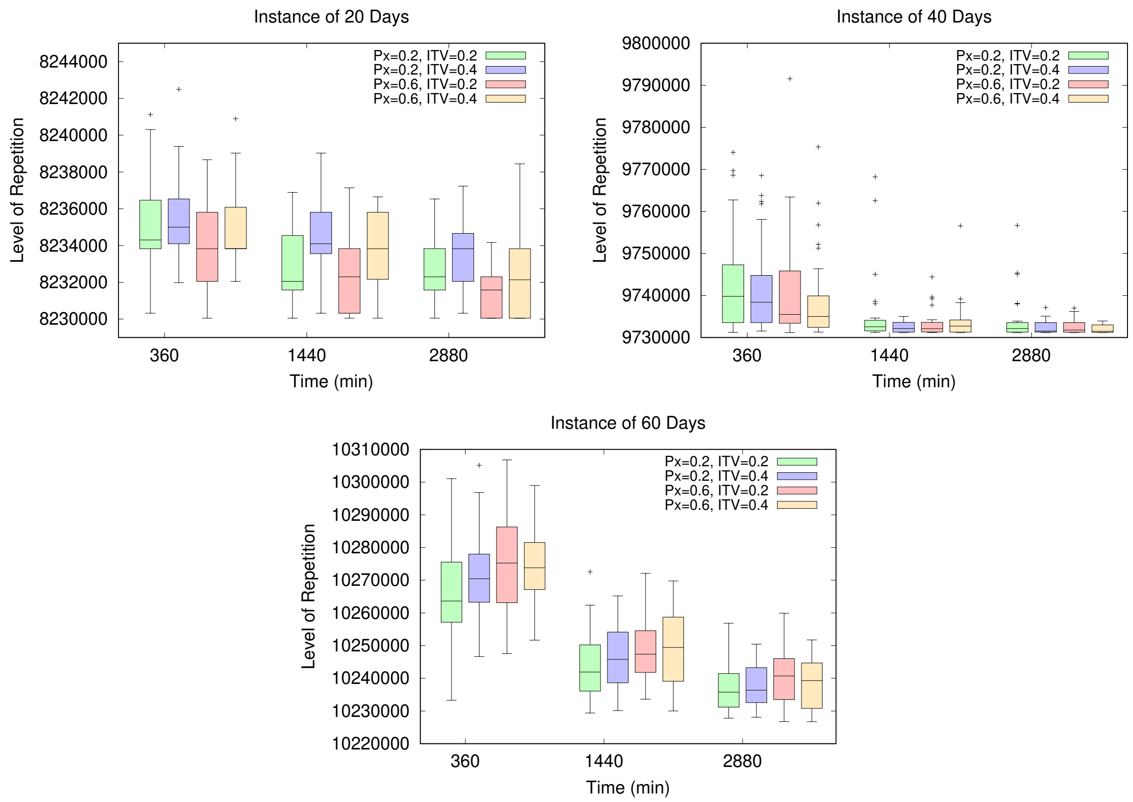

Fig. 4.7 Box-plots of the Level of Repetition for 35 runs for instances with 20, 40, and 60 days. Four configurations with three different stopping criterion were taken into account.

4.3.10 Software Requirements

- This desktop software has to elaborate a menu plan given the database of courses and nutritional constraints.
- There are two types of constraints: daily and global.
- The desktop software has to be flexible in terms of constraints. In one run the problem might consider only constraints regarding fruits and vegetables, and in others constraints of Kilo calories.
- The menu plan of one day considers:
  - Breakfast
  - Morning Snack
  - Starter (two options)
  - Main course (two options)
  - Evening Snack
  - Dinner

- For each course of the database it has to specify its associated meal (or time of the day), as well as its nutritional intake.
- Each nutritional restriction must be indicated in the same unit of measurement.
- The courses are associated with a category in which the level of repetition has to be minimized.
- The courses are classified in favorite and non-favorite, where the favorite courses shouldn't repeat for three weeks and non-favorites shouldn't repeat in four weeks.

4.3.11 Input for the Desktop Software

This software requires two files as input, which has to be in Excel format. The first defines the database of courses, and the second contains the nutritional constraints.

4.3.12 Database Format File

In this file the first row is the header, then each following row defines a course. The first column assigns a name to the course, and the second column specifies the meal (time of the day) to which such a course belongs. Additionally, other two columns are required: "Category" and "isFavorite". The category column is an integer value, and is used by nutritionists to associate courses. In this case two categories (1 and 2) are considered, however a course might belong to both categories, such cases has to be assigned with a value of 0. The field "isFavorite" takes values 0 and 1 (Boolean), which indicates if a course can be repeated for three or four weeks. Additional information can be specified in columns as desired.

4.3.13 Format of the Nutritionist Constraints

In this file, the dietary restrictions are specified. This data is detailed in four columns that are defined as follows:

- Type of constraint: it can be either daily or global.
- Nutrient: refers to the name of the nutrient associated with the constraint. Note that it must be the same as the ones defined in the database of courses, i.e. there must be a column with this name.
- Minimum: this is the minimum amount of nutrient to ingest.
- Maximum: this is the maximum amount of nutrient to ingest.

4.3.14 Format of the Output

The resulting file has an Excel format that contains the menu plan. This menu plan is detailed with each day per row. The first column of this file defines the meals, and the remaining columns provides the amount of nutrient intakes. Note that two options are provided for the starter and main course. Thus, for each one of the two options and in each column nutrient, the total nutrient intake is reported in format "*(amount1, amount2)*", then the minimum and maximum amount of nutrient intake allowed in format "*[min, max]*".

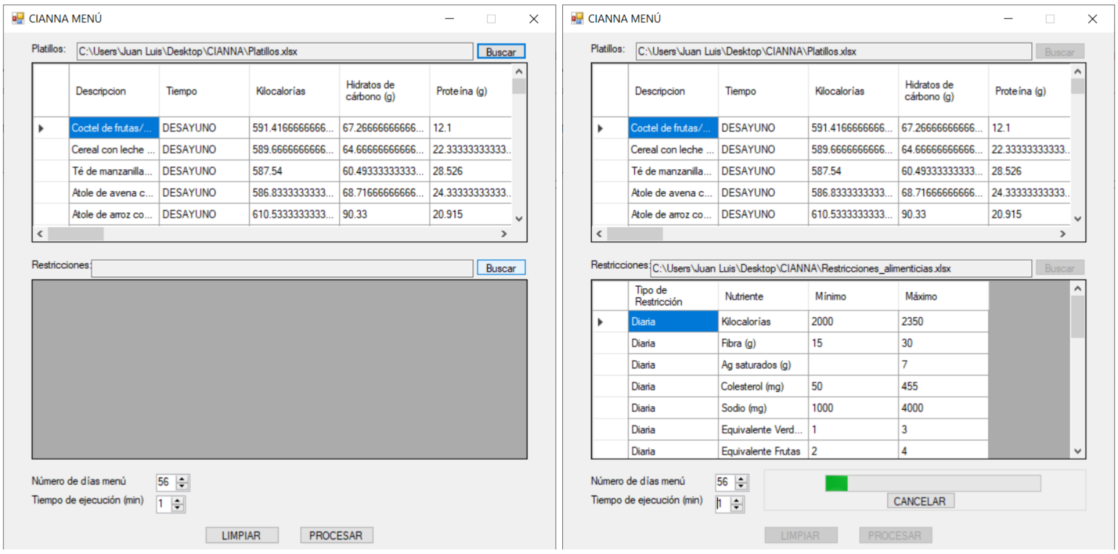

Fig. 4.8 Desktop Application for the MPP in the CIANNA's Formulation.

4.3.15 Software Description

The process of installation of this software is quite standard[4]. The software desktop only works with Windows. The installation of the desktop program must be in a local directory. The interface of this program is shown in Figure 4.8. The database of courses and nutritional constraints is uploaded in that window. At the bottom of the windows there are two fields that refer to the number of days of the menu plan and the total time of execution in minutes. Finalizing the execution of the optimizer, the system will open a window with the suggested menu plan in Excel.

4.3.16 Summary

This chapter presents the application of diversity-aware meta-heuristics to two formulations of the Menu Planning Problem. While the first formulation deals with the minimization of the cost of courses, the second minimizes the level of repetition between days. We have proposed a memetic algorithm, which integrates a novel ad-hoc crossover operator, as well as a novel diversity control mechanism in the form of a replacement scheme, to deal with the menu planning problem. The contribution of the novel operator and the replacement scheme is analyzed by comparing them to the well-known uniform crossover and additional replacement schemes. The experimental validation shows that incorporating explicitly diversity control mechanisms into the memetic algorithm allows improvements in terms of robustness and quality.

In the case of the formulation that minimizes the level of repetition of courses between days, the analyzes are performed only with the replacement *Best Non-Penalized*. Since this problem is more complex, an ILS tailored to the formulation was developed. The results suggest that finding a suitable value for ITV is not a difficult task on a few days menu plan. For menu plans of many days, using the optimizer with long runs is important to decrease the level of repetition. However, feasible solutions are obtained in really short times. The flexibility of our proposal to find feasible solutions for very short runs is shown in several cases. Some

[4]The source code can be downloaded from https://github.com/joelchaconcastillo/CIANNA_Pruebas.git.

specifications regarding the documentation of the desktop application are also described. Note that this last application is currently considered at the CIANNA, where approximately a hundred children live.

Finally, for both formulations, it was not only important to include a proper diversity control mechanism to improve the performance of the memetic algorithm, but also considering a tailored crossover operator.

Part III

Proposals and Achievements in Multi-Objective Optimization

Chapter 5

VSD-MOEA: A Dominance-Based Multi-Objective Evolutionary Algorithm with Explicit Variable Space Diversity Management

Abstract

Most state-of-the-art Multi-Objective Evolutionary Algorithms (MOEAs) promote the preservation of diversity of the objective function space but neglect the diversity of decision variable space. The aim of this chapter is to show that explicitly managing the amount of diversity maintained in the decision variable space is useful to increase the quality of MOEAs when taking into account metrics of the objective space. Our novel Variable Space Diversity based MOEA (VSD-MOEA) explicitly considers the diversity of both decision variable and objective function space. This information is used with the aim of properly adapting the balance between exploration and intensification during the optimization process. In particular, at the initial stages, decisions made by the approach are more biased by information on the diversity of the variable space, whereas it gradually grants more importance to the diversity of objective function space as the evolution progresses. The latter is achieved through a novel density estimator. The new method is compared with state-of-art MOEAs using several benchmarks with two and three objectives. This novel proposal yields much better results than state-of-the-art schemes when considering metrics applied on objective function space, exhibiting a more stable and robust behavior.

5.1 Introduction

In single-objective EAs, it has been shown that taking into account the diversity of decision variable space to properly balance between exploration and exploitation is highly important to attain high quality solutions (Herrera et al., 1996). Diversity can be taken into account in the design of several components, such as in the variation stage (Herrera and Lozano, 2003; Mitchell, 1998), the replacement phase (Segura et al., 2015c) and/or the population model (Koumousis and Katsaras, 2006). The explicit consideration of diversity usually leads to improvements in terms of avoiding premature convergence,which means that taking into account diversity in the design of EAs is especially important when dealing with long-runs. Recently, some diversity management algorithms that combine the information on diversity, stopping criterion and elapsed generations have been devised. They have yielded a gradual loss of diversity that depends on the time or evaluations granted to the execution (Segura et al., 2015c). Specifically, the aim of such a methodology is to promote exploration in the initial generations and gradually alter the behavior toward intensification. These schemes have provided highly promising results.

One of the goals when designing Multi-objective Evolutionary Algorithms (MOEAs) is to obtain a well-spread set of solutions in objective function space. As a result, most state-of-the-art MOEAs consider the diversity of the objective space explicitly. However, this is not the case for the diversity of decision variable space. Maintaining some degree of diversity in the objective space implies that full convergence is not achieved in the decision variable space (Castillo et al., 2017; Kukkonen and Lampinen, 2009). In some way, the decision variable space inherits some degree of diversity due to the way in which the objective space is taken into account. However, this is just an indirect way of preserving diversity of decision variable space, so in some cases the level of diversity might not be large enough to ensure a proper degree of exploration. For instance, it has been shown that with some of the WFG test problems, in most state-of-the-art MOEAs the *distance parameters* quickly converge, meaning that the approach focuses just on optimizing the *position parameters* for a long period of the optimization process (Castillo et al., 2017). Thus, while some degree of diversity is maintained, a situation similar to premature convergence occurs, meaning that genetic operators might no longer be able to generate better trade-offs.

In light of the differences between the state-of-the-art single-objective EAs and MOEAs, this chapter proposes a novel MOEA, the Variable Space Diversity based MOEA (VSD-MOEA), which relies on explicitly managing the amount of diversity in the decision variable space. Similarly to the successful methodology applied in single-objective optimization, the stopping criterion and the number of evaluations evolved are used to vary the amount of diversity desired in decision variable space. The main difference with respect to the single-objective case is that diversity of the objective function space is simultaneously considered by using a novel objective space density estimator. Particularly, the approach grants more importance to the diversity of decision variable space in the initial stages, and it gradually grants more importance to the diversity of objective function space as the evolution progresses. In fact, in the last stage of execution, the diversity of the decision variable space is neglected. Thus, in the last phases, the proposed approach is quite similar to current state-of-the-art approaches. To the best of our knowledge, this is the first MOEA whose design follows this dynamic principle.

Since there currently exists quite a large number of different MOEAs (Tan et al., 2006), three popular schemes have been selected to validate our proposal, including one that promotes diversity in the variable space to deal with multi-modal multi-objective optimization. This validation was performed using several well-known benchmarks and proper quality metrics. This chapter clearly shows the important benefits of properly taking into account the diversity of decision variable space. In particular, the advantages become more evident in the

Algorithm 10 Main Procedure of VSD-MOEA

```
Initialization: Generate an initial population P_0 with p individuals.
Evaluation: Evaluate all individuals in the population.
Assign t := 0
while (not stopping criterion) do
    Mating selection: Fill the mating pool by performing binary tournament selection on P_t, based on the non-dominated
    ranks (ties are broken randomly).
    Variation: Apply SBX and Polynomial-based mutation to the mating pool to create an offspring population Q_t with
    p individuals.
    Evaluation: Evaluate all individuals in Q_t.
    Survivor selection: Generate P_{t+1} by applying the replacement scheme described in Algorithm 11, using P_t and Q_t
    as inputs.
    t := t + 1
end while
```

most complex problems. Note that this is consistent with the single-objective case, where the most important benefits have been obtained in complex multi-modal cases (Segura et al., 2016a). It is also important to clarify that, despite considering the variable space diversity, our work is not a niche-based proposal for multi-modal optimization (Deb and Tiwari (2008), Zhou et al. (2009), Li et al. (2016b),Liang et al. (2016)). Instead, this work aims to show that explicitly managing the amount of diversity maintained in the decision variable space is useful to increase the quality of MOEAs when taking into account metrics of the objective space.

The rest of this chapter is organized as follows. The proposal VSD-MOEA is detailed in Section 5.2. Section 5.3 is devoted to the experimental validation of the proposal. Finally, a summary and conclusions of this chapter are given in Section 5.4. Note also that some supplementary materials are also provided in the Appendix D. They include details of the experimental results with additional performance measures and some additional experiments, as well as an explanatory video. The proper definition of the kind of multi-objective problems addressed in this chapter can be found in the Appendix B.

5.2 Algorithmic Proposal

This section provides a full description of our proposal called *Variable Space Diversity based MOEA* (VSD-MOEA) [1]. The novelty of VSD-MOEA appears in the replacement phase, which incorporates the use of variable space diversity and a novel objective space density estimator. The main principle behind the design of the novel replacement is to use the stopping criterion and elapsed generations with the aim of gradually moving from exploration to exploitation during the search process. In this chapter, our decision was to incorporate it into a dominance-based approach. Note that this category has been particularly suitable for problems with two and three objectives. Thus, some of our design decisions might not be suitable for dealing with many-objective optimization problems.

The general framework of VSD-MOEA is quite standard. Algorithm 10 shows the pseudo-code of VSD-MOEA. Parents are selected using a binary tournament selection based on dominance ranking with ties broken randomly. The variation stage is based on applying the well-known Simulated Binary Crossover (SBX) and polynomial-based mutation operators (Deb et al., 1995, 1996). Thus, the contribution appears in the replacement phase. Note that t is used to denote the number of the current generation. The rest of this section is devoted to describe the replacement phase, including the novel objective space density estimator.

[1]The source code in C++ of our approach is freely available at https://github.com/carlossegurag/VSD-MOEA

5.2.1 Replacement Phase of VSD-MOEA

The replacement phase of EAs is in charge of deciding, for each generation, which members of the previous population together with their corresponding offspring will survive. The novel replacement scheme presented here promotes a gradual movement from exploration to exploitation, which has been a highly beneficial principle in the design of single-objective optimizers (Segura et al., 2015c). Specifically, the replacement phase operates as follows. First, the members of the previous population and offspring are merged in a multi-set with $2 \times p$ individuals. Then, an iterative process that selects an additional individual sequentially is used to pick the p survivors. In order to take into account the diversity of decision variable space, the Distance to Closest Survivor (DCS) of each individual is calculated at each iteration. Thus, the DCS of an individual $\vec{y} \in \Omega$ is calculated as $\min_{\vec{s} \in \mathbb{S}} Distance(\vec{y}, \vec{s})$, where $\mathbb{S}$ is the multi-set containing the currently selected survivors. Normalized Euclidean distances are considered (Barrett, 2005), so in order to calculate distances between two individuals $\vec{a}, \vec{b} \in \Omega$, Eqn. (5.1) is applied. Note that each variable is normalized and the sum is divided by the number of variables. Thus, this measure is expected to be less sensitive to the dimensionality of the decision variable space and to the domain of the variables (Ning et al., 2008). In the first iteration, the $\mathbb{S}$ multi-set is empty, so the DCS of each individual is infinity.

$$Distance(\vec{a}, \vec{b}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{a_i - b_i}{x_i^{(U)} - x_i^{(L)}} \right)^2} \tag{5.1}$$

Note that individuals with larger DCS values are those that contribute more significantly to promoting exploration. In order to avoid an excessive decrease in the degree of exploration, individuals with a DCS value below a certain threshold are penalized. Then, among the non-penalized individuals, an objective space density estimator is used to select the additional survivor of the iteration. In our case, the novel density estimator described in the next subsection is used. Note that it might happen that all individuals are penalized, in which case the individual with the largest DCS is selected to survive.

In order to better understand the penalty method, it can be visualized in the following way. After selecting each survivor, a hyper-sphere centered at a candidate solution — in decision variable space — is created. Then, all the individuals that are inside the hyper-sphere are penalized, with the objective space density estimator only taking into account the survivors and the non-penalized individuals. This is illustrated in Fig. 5.1, which represents a state where three individuals have been selected to survive and an additional survivor must be picked. The left side shows individuals in decision variable space. Current survivors are marked with a red border. Each of them is surrounded by a dashed blue circle of radius D_t. In this scenario, the penalized individuals are the ones with numbers 4, 5, and 6. In objective function space — right side — the penalized individuals are shown in gray, indicating that the objective space density estimator is not considering them.

Since using a large radius for the hyper-spheres induces a large degree of exploration, it makes sense to alter this value during the optimization process. This is precisely one of the key elements of our proposal. The sizes of the hyper-spheres are modified dynamically by taking into account the stopping criterion and elapsed generations. Specifically, the radius is linearly decreased starting from an initial distance. This means that in the initial phases, exploration is promoted. However, as the size of the radius decreases, only very close individuals are penalized, meaning that more exploitation is allowed. Note that this method requires a parameter that is the initial radius of the hyper-spheres or the initial threshold value. This parameter is denoted by ITV. Assigning a large value to this parameter might result in many individuals being penalized, which might thus maintain non-useful diversity. However, a value that is too small might not prevent fast convergence, meaning

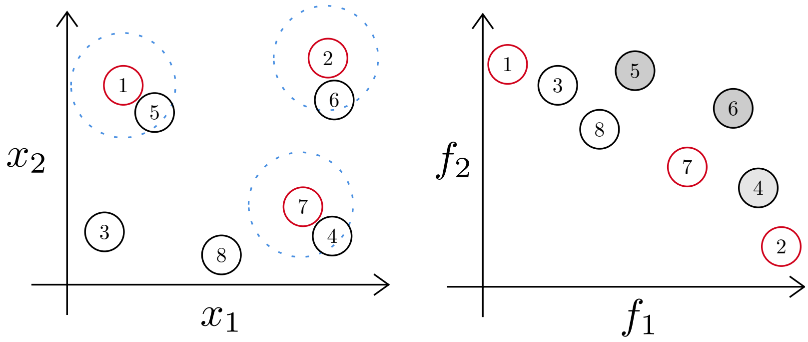

Fig. 5.1 Penalty Method of the Replacement Phase - The left side represents the decision variable space, and the right side the objective function space.

Algorithm 11 Replacement Phase of VSD-MOEA

1: Input: P_t (Population of current generation), Q_t (Offspring of current Generation), p (Population Size) and ITV (Initial Threshold Value)
2: Output: P_{t+1}
3: $R_t := P_t \cup Q_t$
4: $P_{t+1} := \emptyset$
5: $Penalized := \emptyset$
6: $D_t := ITV - ITV \times \frac{G_{Elapsed}}{0.5 \times G_{End}}$
7: **while** $|P_{t+1}| \leq$ p **do**
8:     Compute DCS of individuals in R_t, using P_{t+1} as a reference set
9:     Move the individuals in R_t with $DCS < D_t$ to $Penalized$
10:     **if** R_t is empty **then**
11:         Compute DCS of individuals in $Penalized$, using P_{t+1} as a reference set
12:         Move the individual in $Penalized$ with the largest DCS to R_t
13:     **end if**
14:     Identify the first front (F) in $R_t \cup P_{t+1}$ with at least one individual $\vec{y} \in R_t$
15:     Use the novel density estimator (Algorithm 12) to select a new survivor from F and move it from R_t to P_{t+1}
16: **end while**
17: **return** P_{t+1}

the approach might behave as a traditional non-diversity based MOEA. The robustness of the proposal with respect to this additional parameter is studied in our experimental validation.

The algorithm 11 formalizes the replacement phase of VSD-MOEA. First, the population of the previous generation (P_t) and the offspring (Q_t) are merged in R_t (line 3). At each iteration, the multi-set R_t contains the remaining non-penalized individuals that might be selected to survive. The population of survivors (P_{t+1}) and the set containing the penalized individuals are initialized to the empty set (lines 4 and 5). Then, the threshold value (D_t) that is used to penalize individuals that are too close is calculated (line 6). Note that ITV denotes the initial threshold value, $G_{Elapsed}$ is the number of generations that have evolved, and G_{End} is the stopping criterion, that is, the number of generations that will evolve during the execution of VSD-MOEA. The linear decrease is calculated so that after 50% of the total number of generations, the value of D_t is below 0, meaning that no penalties are applied. This means that in the first half of the algorithm run, more exploration is induced than in traditional MOEAs. Please refer to the Supplementary Material (Appendix D.4) for the sensitivity analysis of the final moment of diversity promotion. Then, an iterative process that selects an individual sequentially is executed until the survivor set contains p individuals (line 7). The iterative process works as follows. First, the DCS value of each remaining non-penalized individual is calculated (line 8). Then, those individuals with a DCS value lower than D_t are moved to the set of penalized individuals (line 9). If all the remaining individuals are penalized (line 10), it means that the amount of exploration is lower than desired. Thus, the individual with the largest DCS value is recovered, i.e., moved to the set of non-penalized individuals (lines 11 and 12), and thus survives. Finally, the objective function space is considered. Specifically, candidate non-penalized individuals and current survivors are merged. Then, the well-known non-dominated sorting procedure proposed in Deb et al. (2002a) is executed on this set, stopping as soon as a front with at least one candidate individual is found, i.e., with an individual of R_t (line 14). Then, taking the identified front as an input, a novel objective space density estimator is used to select the next survivor, which is moved from R_t to P_{t+1} (line 15). The specific way in which each individual's contribution to the diversity of the objective space is measured is described in the next section.

It is important to note that the pseudo-codes presented in this chapter are designed for explanatory purposes, and their corresponding implementations are not necessarily straightforward. For example, in order to calculate the DCS values on line 11, there is no need to iterate over all the solutions in P_{t+1}. Instead, when P_{t+1} is updated by including an additional individual, the distances are updated.

5.2.2 A Novel Density Estimator for Objective Function Space

Since the dominance definition is not related to the preservation of diversity in objective function space, dominance-based MOEAs usually incorporate objective-space density estimators to promote the survival of diverse individuals. As previously described, our density estimator selects a new survivor from the front identified in line 14 of Algorithm 11. This front (referred in Algorithm 12 as F) contains at least one individual belonging to R_t, and it might also contain some elements of P_{t+1}. The aim behind the selection of the next survivor is to pick an individual of the input front that contributes significantly in terms of the quality and diversity of the objective space.

Algorithm 12 describes the selection of the next survivor. First, the sets FP and FR are identified (lines 3 and 4). FP contains the current survivors that are in F (already selected for the next generation P_{t+1}), whereas FR contains the remaining non-penalized individuals that are in F. Then, similarly to most state-of-the-art algorithms, a step to promote the selection of extreme-points is included (Sun et al., 2018). Note that selecting the extreme-points following the criteria of the best solution for each objective might cause some drawbacks related to accepting a small improvement in one objective at the expense of significant degradation in other objectives. This issue can be solved by applying augmented functions (Deb and Abouhawwash, 2015; Sun et al., 2018), which was our design choice. In particular, and similarly to Sun et al. (2018), the m extreme points are selected by decomposing the m-objective MOP into m single-objective problems. Specifically, the k^{th} extreme point is the one that minimizes Eqn. (5.2). The second term, which is multiplied by the penalization parameter ρ, is a measure of overall quality and is used to avoid extreme points that exhibit very poor qualities in some of the objectives. Obviously, the setting of the parameter ρ is problem-dependent, but for benchmark problems where the objective functions are between 0 and some units, the suggested value is 10^{-4} (Deb and Abouhawwash, 2015), so this was the value used in our validation. The lines 6 to 11 identify, for each objective k, the candidate solutions that minimize the Augmented Function (AF). If some of the extreme points belong to the non-penalized candidates (FR), then one of them is randomly selected as a survivor and the process ends (lines 12 to 14).

$$AF_k(\vec{x}) = f_k(\vec{x}) + \rho \times \sum_{j=1}^{m} f_j(\vec{x}) \tag{5.2}$$

Algorithm 12 Density Estimator

1: Input: P_{t+1} (Survivors), R_t (Candidates), F (Current front)
2: Output: $\vec{y} \in R_t$
3: $FP := P_{t+1} \cap F$
4: $FR := R_t \cap F$
5: $Extreme\_points := \emptyset$
6: **for** $k \in$ Number of objectives (m) **do**
7:     Select the best individual $\vec{y} \in F$ according to the k objective using Eqn. 5.2.
8:     **if** $\vec{y} \in FR$ **then**
9:         $Extreme\_points = Extreme\_points \cup \vec{y}$
10:     **end if**
11: **end for**
12: **if** $|Extreme\_points| > 0$ **then**
13:     $\vec{s^*}$:= a random individual from $Extreme\_points$
14:     **return** $\vec{s^*}$
15: **end if**
16: $MaxID := 0$
17: **for** $\vec{y} \in FR$ **do**
18:     $Improvement := \min_{\vec{s} \in FP} ID(\vec{y}, \vec{s})$ (Eqn. 5.3)
19:     **if** $Improvement > MaxID$ **then**
20:         $MaxID := Improvement$
21:         $\vec{s^*} := \vec{y}$
22:     **end if**
23: **end for**
24: **return** $\vec{s^*}$

In cases where the individuals that optimize each AF are already in P_{t+1}, a contribution to the diversity and quality of the objective space is calculated for each individual in FR (lines 16 to 24). This contribution is calculated by taking into account the current survivors of the front (FP). Specifically, the Improvement Distance (ID) defined for the indicator IGD+ (Ishibuchi et al., 2015) is used. The ID of an individual $\vec{x}$ with respect to an individual $\vec{y}$ is calculated by taking into account only the objectives where $\vec{x}$ is better. Specifically, Eqn. (5.3) is used.

$$ID(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (\max\{0, f_i(\vec{y}) - f_i(\vec{x})\})^2} \tag{5.3}$$

The contribution of each member $\vec{y} \in FR$ is calculated as $\min_{\vec{s} \in FP} ID(\vec{y}, \vec{s})$. Then, the individual with the highest contribution is selected as the next survivor (lines 19 to 22). Note that the process of selecting the individual with the best contribution ($\vec{s^*}$) is defined in Eqn. (5.4).

$$\vec{s^*} = \arg\max_{\vec{y} \in FR} \min_{\vec{s} \in FP} ID(\vec{y}, \vec{s}) \tag{5.4}$$

Finally, note that ID has not been used earlier as a density estimator. The logic is to avoid the selection of individuals that are far from the remaining ones, just because there is a worsening in some of the objective functions. As will be shown in the experimental validation, this novel density estimator provides important benefits compared to more traditional density estimators.

5.3 Experimental Validation

This section describes the experimental validation carried out to study the performance and gain a clear understanding of the specifics of our proposed VSD-MOEA. Our results clearly show that controlling the diversity of the decision variable space provides a way to further improve the results obtained using the current state-of-the-art MOEAs. First, we discuss some technical specifications involving the benchmark problems and algorithms implemented. We then present a comparison between VSD-MOEA and state-of-the-art algorithms for long runs. Then, four additional experiments to fully validate VSD-MOEA are included. These analyses are designed to test the scalability in the decision variable space, the performance with different stopping criteria, the novel diversity management mechanism in the objective space, and the implications of the initial penalty threshold.

This work takes into account some of the most popular and widely used benchmarks in the multi-objective field, which are defined in the Appendix C. These problems are the WFG (Huband et al., 2005, 2006), DTLZ (Deb et al., 2005), and UF (Zhang et al., 2008) test suites configured in a standard way. The WFG test problems were used with two and three objectives and configured with 24 parameters, 20 of them corresponding to the distance parameters and 4 to the position parameters. In the DTLZ test problems, the number of variables was set to $n = m + r - 1$, where $r = \{5, 10, 20\}$ for DTLZ1, DTLZ2 to DTLZ6 and DTLZ7, respectively. The UF benchmark comprises seven problems with two objectives (UF1-7) and three problems with three objectives (UF8-10). All of them were configured with 30 variables. Note that the experiment used to analyze the scalability considers different numbers of variables. The experimental validation includes three well-known state-of-the-art MOEAs and VSD-MOEA. The MOEAs that are considered are MOEA/D [2], R2-EMOA [3], and CPDEA [4]. Note that CPDEA is the only method that includes an archive; thus the final results are obtained from the archive. In the remaining methods, following the original implementations, the solutions are taken from the last population. In all cases, the maximum size of the solution set is equal to the population size. Note that in the Supplementary Material (Appendix D.5) some results considering the incorporation of passive archives (Schütze and Hernández, 2021) in all the methods are discussed. Similar conclusions are attained, so the performance is more dependent on the selection and replacement phases, which are the main distinguishing features of each algorithm, than on the specific way of selecting the final solution set. Also note that, in the case of MOEA/D, several variants have been devised. The MOEA/D implementation considered is the one that obtained first place in the 2009 IEEE Congress on Evolutionary Computation's MOP Competition (Zhang et al., 2009).

Given that all the algorithms considered are stochastic, each execution was repeated 35 times with different seeds in all the experiments. The hypervolume indicator (HV) is used to compare results. Note that in the supplementary material (Appendix D), the results are also compared in terms of IGD+ with similar conclusions. The reference point used to calculate the HV is chosen to be a vector whose values are sightly larger (ten percent) than the Nadir point, as suggested in Ishibuchi et al. (2017). Normalized HV is used to facilitate the interpretation of the results (Li et al., 2014), and the reported value is calculated as the ratio between the normalized HV obtained and the maximum attainable normalized HV. In this way, a value equal to one means a perfect approximation. Note that a value equal to one is not attainable because MOEAs yield a discrete approximation. In order to statistically compare the HV ratios attained by the different algorithms, the guidelines proposed in del Amo and Pelta (2013); Derrac et al. (2011) are followed. Given a set of approaches and their

[2] https://github.com/P-N-Suganthan/CEC2009-MOEA/blob/master/Codes-of-Accepted-Papers.rar
[3] http://inriadortmund.gforge.inria.fr/r2emoa/
[4] https://github.com/yiping0liu/CPDEA

Table 5.1 Crossover probability applied in each MOEA

	2 objectives	3 objectives
VSD-MOEA	0.4	0.4
CPDEA	1.0	0.6
MOEA/D	1.0	1.0
R2-EMOA	1.0	0.2

Table 5.2 Parameterization applied in each MOEA

Algorithm	Configuration
MOEA/D	Max. updates by subproblem (η_r) = 2, tour selection = 10, neighbor size = 10, period utility updating = 30 generations, local selection probability (δ) = 0.9
VSD-MOEA	Initial threshold value (ITV) = 0.4
R2-EMOA	Equally distributed weight vectors (ρ) = 1, offspring by iteration = 1
CPDEA	Nearest neighbors (K) = 3, weight of standard deviation (η) = 2

corresponding results, Kruskal-Wallis is used first as an Omnibus test to detect if there are any significant differences. In cases where there are significant differences, pair-wise statistical test are used to detect them; specifically, the Mann-Whitney post-hoc test with Hommel's correction of p-values. An algorithm X is said to beat algorithm Y when the differences between them are statistically significant, and the mean HV ratio obtained by X is higher than the mean achieved by Y. Note that in both tests, a significance level of 5% was considered.

The common configuration in all experiments was as follows: the population size was set to 100, the stopping criterion was set to 2.5×10^6 function evaluations, and the genetic operators were Simulated Binary Crossover (SBX) and polynomial-based mutation (Deb et al., 1995, 1996). The crossover and mutation distribution indexes were fixed to 2 and 50, respectively. The mutation probability was set to $1/n$. In order to select the crossover probabilities, five parameterizations were tested ($\{0.2, 0.4, 0.6, 0.8, 1.0\}$). These configurations were executed with all the aforementioned benchmarks in each algorithm. Then, the mean HV ratios were calculated independently for the problems with two and three objectives, and the parameterization that attained the largest mean was selected for the validation. Table 5.1 shows the crossover probability selected for each algorithm, whereas the specific parameterization required for each algorithm is included in Table 5.2. Note that for specific parameterizations, the default values provided by the authors are maintained. In the case of VSD-MOEA, the initial threshold value (ITV) was set to 0.4, which is the recommended value for single-objective optimization (Castillo and Segura, 2019). In subsequent experiments, the implications of the ITV on the quality of the results are analyzed. Also note that scalarization functions are required in MOEA/D and R2-EMOA. In both cases, the Tchebycheff approach is used. The procedure for generating the weight vectors differs in MOEA/D and R2-EMOA. R2-EMOA was applied with 501 and 496 weight vectors for two and three objectives, respectively (Trautmann et al., 2013). In contrast, MOEA/D requires the same number of weight vectors as the population size. They were generated using the uniform design (UD) and the good lattice point (GLP) method (Berenguer and Coello, 2015; Ma et al., 2014).

5.3.1 Comparison Against State-of-the-art MOEAs for Long Runs

Our first experiment aims to compare the performance for long runs of VSD-MOEA against state-of-the-art proposals. Long runs (2.5×10^6 function evaluations) are considered because this is the kind of execution where diversity-based EAs have been more successful. Experiments with shorter and longer runs are discussed in Section 5.3.3.

Table 5.3 Summary of the hypervolume ratio results attained for problems with two objectives. The higher the normalized hypervolume value, the better the algorithm.

	VSD-MOEA			**CPDEA**			**MOEA/D**			**R2-EMOA**		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
WFG1	**0.993**	0.994	0.002	0.963	0.965	0.013	0.993	0.993	0.001	0.980	0.989	0.018
WFG2	**0.996**	0.998	0.008	0.993	0.996	0.009	0.965	0.965	0.000	0.966	0.966	0.005
WFG3	**0.992**	0.992	0.000	0.973	0.973	0.002	0.992	0.992	0.000	0.991	0.991	0.000
WFG4	0.990	0.990	0.000	0.964	0.964	0.003	0.988	0.988	0.000	**0.991**	0.991	0.000
WFG5	**0.880**	0.881	0.003	0.862	0.862	0.002	0.877	0.876	0.003	**0.882**	0.882	0.002
WFG6	0.884	0.884	0.012	0.787	0.788	0.003	**0.918**	0.919	0.020	**0.914**	0.914	0.015
WFG7	0.990	0.990	0.000	0.973	0.974	0.002	0.988	0.988	0.000	**0.991**	0.991	0.000
WFG8	**0.904**	0.947	0.053	**0.875**	0.881	0.026	0.808	0.808	0.007	0.803	0.804	0.005
WFG9	**0.946**	0.961	0.027	0.791	0.791	0.002	**0.912**	0.949	0.070	**0.894**	0.953	0.079
DTLZ1	0.992	0.992	0.000	0.991	0.991	0.000	**0.993**	0.993	0.000	0.992	0.992	0.000
DTLZ2	0.990	0.990	0.000	0.983	0.983	0.001	0.989	0.989	0.000	**0.992**	0.992	0.000
DTLZ3	0.990	0.990	0.000	0.988	0.988	0.000	0.989	0.989	0.000	**0.992**	0.992	0.000
DTLZ4	**0.990**	0.990	0.000	0.979	0.980	0.003	0.989	0.989	0.000	0.678	0.991	0.362
DTLZ5	0.990	0.990	0.000	0.983	0.983	0.001	0.989	0.989	0.000	**0.992**	0.992	0.000
DTLZ6	**0.990**	0.990	0.000	0.807	0.820	0.088	0.989	0.989	0.000	0.685	0.667	0.088
DTLZ7	0.996	0.996	0.000	0.995	0.995	0.000	0.996	0.996	0.000	**0.997**	0.997	0.000
UF1	**0.989**	0.990	0.003	0.976	0.976	0.003	0.980	0.981	0.005	0.881	0.881	0.030
UF2	**0.987**	0.988	0.004	0.968	0.968	0.001	**0.986**	0.986	0.004	0.979	0.979	0.003
UF3	**0.876**	0.878	0.014	0.755	0.757	0.049	0.616	0.609	0.065	0.556	0.557	0.040
UF4	0.891	0.891	0.003	0.850	0.849	0.004	0.883	0.884	0.005	**0.900**	0.901	0.003
UF5	0.589	0.579	0.050	**0.676**	0.671	0.070	0.294	0.206	0.247	0.306	0.332	0.152
UF6	**0.854**	0.852	0.030	**0.839**	0.848	0.043	0.526	0.538	0.143	0.558	0.545	0.113
UF7	**0.985**	0.985	0.002	0.967	0.968	0.004	0.957	0.979	0.121	0.756	0.944	0.225
Mean	**0.943**	0.945	0.009	0.910	0.912	0.014	0.896	0.895	0.030	0.855	0.880	0.050

Table 5.4 Statistical Tests and Deterioration Level of the HV ratio for problems with two objectives

	$\uparrow$	$\downarrow$	$\leftrightarrow$	**Score**	**Deterioration**
VSD-MOEA	51	11	7	40	0.147
CPDEA	16	51	2	-35	0.911
MOEA/D	27	34	8	-7	1.290
R2-EMOA	31	29	9	2	1.643

Table 5.3 shows the HV ratio obtained for the benchmark functions with two objectives. Specifically, the mean, median, and standard deviation of the HV ratio are shown for each method and problem tested. The last row shows the results considering all the test problems together. For each test problem, the data for the method that yielded the largest mean are shown in **boldface**. Furthermore, all methods that were not statistically inferior to the method with the largest mean are also shown in **boldface**. From here on, the methods shown in **boldface** for a given problem are referred to as the winning methods. Based on the number of test problems where each method is in the group of the winning methods for the cases with two objectives, the best methods are VSD-MOEA and R2-EMOA with 13 and 10 wins, respectively. Thus, VSD-MOEA is the most competitive method in terms of this measure. More impressive is the fact that the mean HV ratio attained by VSD-MOEA, when all the problems are considered simultaneously, is much higher than the one attained by R2-EMOA. Note that the decision variable space diversity-aware methods (CPDEA and VSD-MOEA) reported the largest total mean. However, there is a huge difference between VSD-MOEA and CPDEA, showing the benefits of decreasing the importance given to the decision variable space diversity as the evolution progresses. Inspecting the data carefully, it is clear that in the cases where VSD-MOEA loses (attains lower HV), the difference with respect to the best method is not very large. For instance, the difference between the mean HV ratio attained by the best method and by VSD-MOEA was never larger than 0.1. However, all the other methods exhibited a deterioration greater than 0.1 in several cases. Specifically, it happened in 4, 3 and 6 problems for CPDEA, MOEA/D and R2-EMOA, respectively. This means that even if VSD-MOEA loses in some cases, its deterioration is always small, exhibiting a much more robust behavior than any other method.

Table 5.5 Summary of the hypervolume ratio results attained for problems with three objectives. The higher the normalized hypervolume value, the better the algorithm.

	VSD-MOEA			CPDEA			MOEA/D			R2-EMOA		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
WFG1	0.785	0.789	0.017	0.430	0.428	0.033	**0.968**	0.968	0.001	0.928	0.926	0.009
WFG2	**0.988**	0.989	0.001	0.923	0.924	0.006	0.967	0.976	0.034	0.905	0.962	0.070
WFG3	0.989	0.989	0.000	0.906	0.905	0.013	**0.993**	0.992	0.000	0.992	0.992	0.000
WFG4	**0.920**	0.920	0.001	0.795	0.796	0.013	0.861	0.861	0.003	0.906	0.905	0.001
WFG5	0.834	0.832	0.004	0.801	0.801	0.004	0.795	0.795	0.001	**0.842**	0.843	0.002
WFG6	0.837	0.835	0.007	0.766	0.767	0.005	0.811	0.810	0.012	**0.860**	0.860	0.007
WFG7	**0.919**	0.919	0.001	0.774	0.778	0.021	0.865	0.865	0.000	0.905	0.905	0.001
WFG8	**0.863**	0.864	0.035	0.672	0.682	0.039	0.779	0.779	0.002	0.820	0.820	0.002
WFG9	**0.822**	0.824	0.038	0.727	0.727	0.005	**0.810**	0.837	0.047	**0.804**	0.772	0.048
DTLZ1	**0.965**	0.965	0.001	**0.964**	0.964	0.001	0.950	0.950	0.000	0.940	0.940	0.001
DTLZ2	**0.930**	0.930	0.001	0.864	0.864	0.017	0.899	0.899	0.000	0.915	0.915	0.001
DTLZ3	**0.930**	0.930	0.001	0.830	0.916	0.239	0.899	0.899	0.000	0.912	0.915	0.004
DTLZ4	**0.930**	0.930	0.001	0.859	0.858	0.006	0.899	0.899	0.000	0.652	0.577	0.257
DTLZ5	**0.986**	0.986	0.000	0.977	0.977	0.002	0.978	0.978	0.000	**0.986**	0.986	0.000
DTLZ6	**0.986**	0.986	0.000	0.660	0.643	0.115	0.978	0.978	0.000	0.775	0.760	0.082
DTLZ7	**0.965**	0.965	0.001	0.940	0.941	0.004	0.914	0.914	0.000	0.852	0.852	0.014
UF8	**0.918**	0.920	0.011	0.699	0.711	0.045	0.778	0.777	0.006	0.853	0.905	0.104
UF9	**0.962**	0.965	0.011	0.784	0.793	0.053	0.792	0.747	0.071	0.844	0.783	0.076
UF10	**0.602**	0.581	0.095	0.122	0.121	0.060	0.309	0.270	0.150	0.268	0.209	0.132
Mean	**0.902**	0.901	0.012	0.763	0.768	0.036	0.855	0.852	0.017	0.840	0.833	0.043

Table 5.6 Statistical Tests and Deterioration Level of the HV ratio for problems with three objectives

	↑	↓	↔	**Score**	**Deterioration**
VSD-MOEA	47	6	4	41	0.231
CPDEA	5	46	6	-41	2.752
MOEA/D	24	29	4	-5	1.158
R2-EMOA	27	22	8	5	1.522

In order to better clarify these findings, pair-wise statistical tests were conducted among each method tested in each test problem. For the two-objective cases, Table 5.4 shows the number of times that each method won (column ↑), lost (column ↓) and tied (column ↔), as well as a **Score** that is calculated as the difference between the number of times that each method won and the number of times that each method lost. Additionally, for each method A, we calculated the sum of the differences between the mean HV ratio attained by the best method (the ones with the highest mean) and method A, for each problem where A was not in the group of winning methods. This value is shown in the Deterioration column. The data confirm that although VSD-MOEA loses in some cases, the overall number of wins and losses favors VSD-MOEA. More importantly, the total deterioration is much lower in the case of VSD-MOEA, confirming that when VSD-MOEA loses, the deterioration is not that large.

Tables 5.5 and 5.6 show the same information for problems with three objectives. In this case, the superiority of VSD-MOEA is even clearer. Taking into account the mean of all the test problems, VSD-MOEA again obtained a much larger mean HV ratio than the other methods. Specifically, VSD-MOEA obtained a value of 0.902, while the second ranked algorithm (MOEA/D) obtained a value of 0.855. In this case, the difference between the mean HV ratio obtained by the best method and by VSD-MOEA was larger than 0.1 in only one problem (WFG1). Since WFG1 is a uni-modal and biased problem, a large balance towards intensification is required, so promoting further exploration is not helpful. On the contrary, all other methods exhibited a deterioration greater than 0.1 in several cases. In particular, this happened in 3, 5 and 9 problems for MOEA/D, R2-EMOA and CPDEA, respectively. Interestingly, CPDEA achieved the worst results in the three-objective case. As the number of objectives increases, more individuals are required to attain proper discrete approximations. The aim of attaining high diversity in the decision variable space appears to significantly affect the quality attained in the objective space, probably indicating that, as the number of objectives increases, EMMAs might

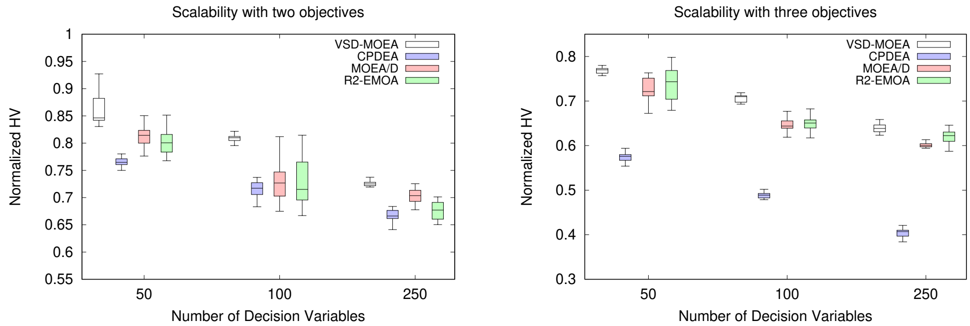

Fig. 5.2 Box-plots of the HV ratio for 35 runs for the two-objective (left side) and three-objective (right side) problems, considering different numbers of variables

be less adequate[5]. Thus, as the number of objectives increases, the dynamic balance promoted by VSD-MOEA seems more important to improve performance. VSD-MOEA is much superior to the other methods both in terms of total deterioration and of total wins and losses (see Table 5.6 and the data shown in **boldface** in Table 5.5). In particular, VSD-MOEA was in the group of winning methods for 15 out of 19 test problems, while the second best ranked algorithm (MOEA/D) was in the group of winning methods for only 4 test problems.

Regarding the kind of problem where VSD-MOEA yields the most impressive improvements, it is clear that this happens in the MOPs that exhibit some features that hinder the optimization process. VSD-MOEA excelled in problems with a strong non-separability (e.g., WFG8, WFG9, UF1-3), high multi-modality (e.g., WFG9, UF6, UF7), irregular Pareto geometries (e.g., WFG2, UF6, UF7, UF9) and complicated Pareto set shapes (e.g., UF6, UF8, UF10). Conversely, the most important decay in performance (in fact, the only one) appeared in the WFG1 problem. Promoting further intensification is normally useful in uni-modal problems, and this is especially the case for biased problems, where small perturbations in decision space provoke large movements in objective function space. Thus, methods that promote the maintenance of further decision variable space diversity do not contribute positively in such kinds of problems.

5.3.2 Decision Variable Scalability Analysis

In order to study the scalability of VSD-MOEA in terms of the number of decision variables, all of the algorithms already described were tested with the same benchmark problems, but considering 50, 100, and 250 variables. Note that in the WFG test problems, the number of position parameters (k) and distance parameters (l) must be specified. The ratio between the number of each kind of variable was kept as in the default configuration. Thus, the number of distance parameters was set to 42, 84, and 210 when using 50, 100 and 250 variables, respectively. The rest of the decision variables were position parameters, meaning that they were 8, 16 and 40, respectively. The stopping criterion was also set for 2.5×10^6 function evaluations. Figure 5.2 shows the box-plots of the mean HV ratios attained with the four algorithms for the problems with two objectives and three objectives and the different dimensionalities. As expected, the HV ratio decreases as the number of variables increases in every algorithm; However, the superiority of VSD-MOEA is maintained in all cases. Additionally, the variability in the case of VSD-MOEA is quite low, which is a typical feature of diversity-aware methods (Segura et al., 2016a). One negative aspect regarding the performance of VSD-MOEA is that, as the

[5] Four additional EMMAs were tested with a similar behavior, but some additional analyses are required to draw more general conclusions.

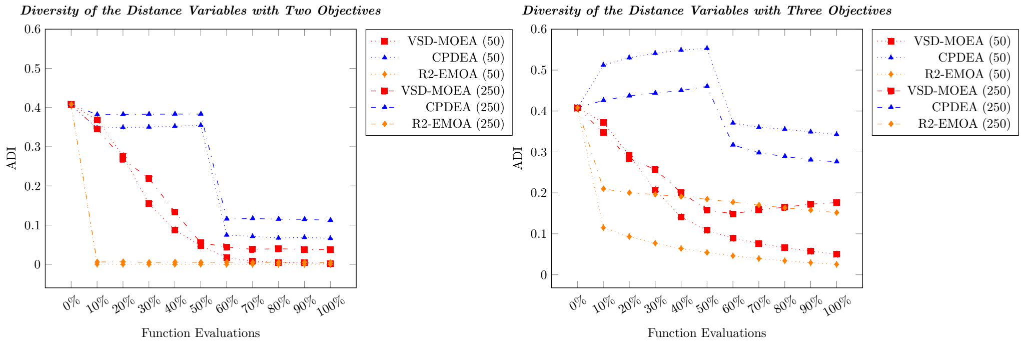

Fig. 5.3 Evolution of ADI for problems WFG1-WFG7 with two and three objectives considering only the distance variables

number of decision variables increases, closer HV ratios are attained by other algorithms. This is especially true in the three-objective problems, where the advantages with respect to R2-EMOA are not as significant.

In order to better understand the reasons for this behavior, we selected the problems WFG1 to WFG7. The WFG test problems divide the variables into two kinds of parameters (this framework uses the term parameter instead of variable): the distance parameters and the position parameters. Note that a parameter i is a distance parameter when modifying $x_i \in \vec{x}$ results in a new solution that dominates $\vec{x}$, is equivalent to $\vec{x}$, or is dominated by $\vec{x}$. However, if i is a position parameter, modifying x_i always results in a vector that is incomparable or equivalent to $\vec{x}$ (Huband et al., 2005). Additionally, note that we selected problems WFG1-WFG7 because their distance parameter values associated to all Pareto optimal solutions have exactly the same values:

$$x_{\{i=k+1:n\}} = 2i \times 0.35 \tag{5.5}$$

This is very important because for these cases, state-of-the-art MOEAs might provoke a rapid convergence in *distance parameters* (already described in Section 2.2.4), resulting in an effect similar to premature convergence in the single-objective case (Castillo et al., 2017; Kukkonen and Lampinen, 2009).

For each algorithm, the average (mean) Euclidean distance among individuals (ADI) in the population was calculated by considering only the distance parameters (ADI is defined in Appendix B.1.6). Figure 5.3 shows how the ADI evolves for the two-objective (left side) and three-objective (right side) problems. The behavior of MOEA/D — which is not included — is similar to that of R2-EMOA in terms of how the ADI evolves. Moreover, to avoid saturating these figures, only the information for VSD-MOEA, CPDEA and R2-EMOA with 50 and 250 variables is shown. The first finding is that, as expected, the decision variable space diversity-aware MOEAs converge much slower than the remaining algorithms. Accordingly, the difference between the diversity maintained in the first generation and that maintained after 10% of the execution, is much larger in R2-EMOA than in VSD-MOEA and CPDEA. In the case of VSD-MOEA, the decrease in ADI is quite linear until the halfway point of execution. This is due to the way in which the threshold distance value (D_t) is calculated. Unlike in CPDEA, where the decrease in ADI is abrupt because it operates in two differentiated stages.

A closer inspection of the data reveals other important aspects. In the two-objective case, increasing the number of variables causes the diversity in R2-EMOA to increase slightly. However, the amount of diversity is low even when using 250 variables, meaning that incorporating mechanisms to increase diversity might be very helpful. In the three-objective case, increasing the number of variables yields a significant increase in the resulting ADI, which means that fast convergence is not an important issue. These results show that, as the

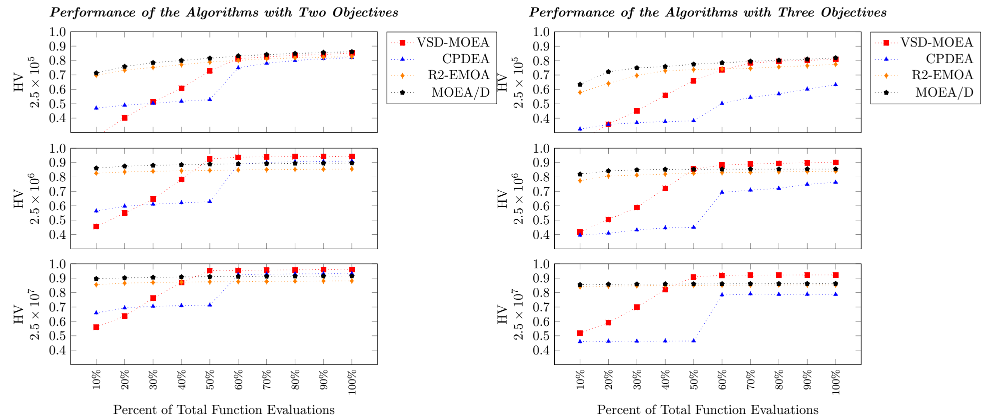

Fig. 5.4 Performance of MOEAs for the problems with two objectives (left side) and three objectives (right side) considering three values for the stopping criterion: short-term (first row), medium-term (second row) and long-term (third row).

Table 5.7 Final mean of HV ratio attained for the three stopping criteria reported in Figure 5.4.

	Two objectives				Three objectives			
	VSD-MOEA	**CPDEA**	**MOEA/D**	**R2-EMOA**	**VSD-MOEA**	**CPDEA**	**MOEA/D**	**R2-EMOA**
2.5×10^5	0.854	0.820	**0.861**	0.826	0.809	0.632	**0.819**	0.775
2.5×10^6	**0.943**	0.910	0.896	0.855	**0.902**	0.763	0.855	0.840
2.5×10^7	**0.962**	0.932	0.915	0.881	**0.923**	0.788	0.863	0.853

number of objectives and variables increases, MOEAs tend to maintain a higher variable space diversity in an implicit way, meaning that explicitly controlling the variable space diversity is probably not as important.

5.3.3 Analysis of the Stopping Criterion

Decision variable space diversity-aware methods usually excel in long runs. As a result, a large stopping criterion was used in our previous experiments. This section considers the performance of the algorithms with several stopping criteria, i.e., maximum number of function evaluations. Furthermore, the trend in HV during execution is inspected with the aim of better understanding the optimization process.

Three different values for the stopping criterion were explored. The values considered were 2.5×10^5, 2.5×10^6, and 2.5×10^7. These values are referred to as short-term, middle-term, and long-term executions, respectively. Figure 5.4 plots the mean HV ratio obtained with each MOEA with two and three objectives versus the number of evaluations. Each figure is divided into three graphs that correspond to the short-term, medium-term, and long-term. Additionally, Table 5.7 shows the final mean HV attained by each model for the different stopping criteria.

The gradual shift from exploration to exploitation attained by VSD-MOEA is clearly seen in the graphs. Instead of reaching a very fast increase in HV, the increase is quite linear in the first half of the run due to the high degree of exploration promoted in this phase. In short-term executions this does not yield important benefits, so the HV ratio attained by VSD-MOEA is similar to those attained by state-of-the-art algorithms. However, as the stopping criterion increases, the benefits of VSD-MOEA become clear. Thus, as in the single-objective case,

Table 5.8 Summary of the hypervolume ratio results attained with VSD-MOEA using three density estimators

	ID			L_2			CD		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
Two objectives	**0.943**	0.945	0.091	0.939	0.940	0.095	0.917	0.915	0.099
Three objectives	**0.902**	0.901	0.096	0.848	0.848	0.167	0.795	0.797	0.186

Table 5.9 Statistical Tests of the HV ratio of the state-of-the-art algorithms and VSD-MOEA with three density estimators (ID, L_2 and CD)

	Two-objectives				Three-objectives			
	$\uparrow$	$\downarrow$	$\leftrightarrow$	**score**	$\uparrow$	$\downarrow$	$\leftrightarrow$	**score**
ID	74	14	27	60	80	7	8	73
L_2	70	13	32	57	60	26	9	34
CD	20	74	21	-54	10	66	19	-56
CPDEA	21	77	17	-56	8	70	17	-62
MOEA/D	43	54	18	-11	38	38	19	0
R2-EMOA	50	46	19	4	45	34	16	11

variable-space diversity-aware methods are especially useful for relatively long-term executions. However, since a faster decrease in diversity is promoted when using short-term executions, the performance is not degraded, meaning it can be used in quite different environments. Thus, the performance in the short-term is similar to current state-of-the-art algorithms, whereas in the long-term, truly significant advances are attained. Finally, note that in the case of CPDEA, there is no significant improvement in the long-term case. Thus, the lower gradual shift from exploration to exploitation promoted in CPDEA is not as helpful.

5.3.4 Analysis of the Novel Density Estimator for the Objective Space

This section considers the impact on the performance of the novel density estimator for the objective space that is proposed in VSD-MOEA. For this analysis, three different density measurements are integrated in VSD-MOEA. In particular, in addition to the Improvement Distance (ID) (Ishibuchi et al., 2015) already described, the Euclidean distance(L_2) (Kukkonen and Deb, 2006a,b) and the NSGA-II crowding distance (CD) are also taken into account. Note that in order to incorporate L_2 and CD, the only modification appears in Eqn.(5.4), where ID is modified by the corresponding distance. VSD-MOEA was executed 35 times to solve the whole benchmark, configured as in the first experiment. Table 5.8 shows the mean and median of the HV ratios obtained for the whole benchmark with the three measurements. In the two-objective problems, L_2 and ID yield quite similar results, whereas CD attains lower values. Note, however, that VSD-MOEA attained a higher mean HV ratio than the remaining state-of-the-art methods with all the density estimators (see Table 5.3), so even with density estimators that are not as adequate, VSD-MOEA excels. The case of three-objective problems is different. In this case, the results attained with ID are quite superior to the remaining ones. This might indicate that the benefits appear in these problems only because of the application of ID; however, this is not the case. In fact, VSD-MOEA with L_2 also exhibits a very good performance in most problems. However, in the case of WFG1, it presents a huge degradation that greatly affects the overall results.

In order to better illustrate the benefits of VSD-MOEA, pair-wise statistical tests were applied by considering VSD-MOEA with the three density measurements and the state-of-the-art methods for the whole benchmark. Table 5.9 shows the number of wins and losses, as well as the score. The benefits of VSD-MOEA are clear when using both ID and L_2. However, the additional advantages provided by ID are remarkable, especially for the problems with three objectives.

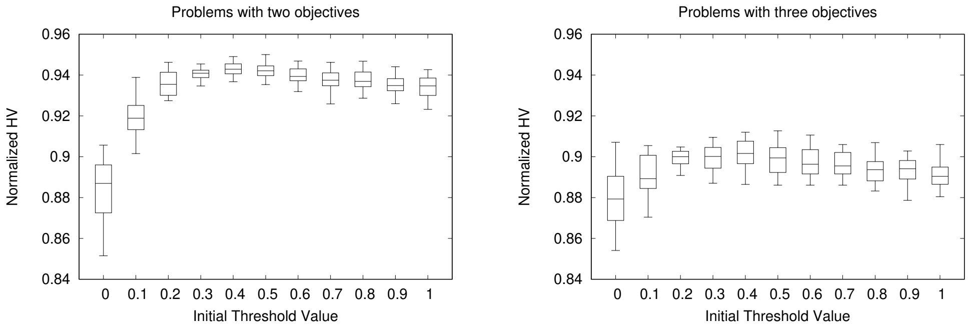

Fig. 5.5 Box-plots of the HV ratio for 35 runs for the two-objective (left side) and three-objective (right side) problems, considering different initial threshold values

5.3.5 Analysis of the Initial Threshold Value

One of the disadvantages of including a strategy for controlling diversity is that this is usually done at the expense of incorporating additional parameters in the EA designed. In the case of VSD-MOEA, the Initial Threshold Value (ITV) must be set. The higher this value, the greater the exploration of the decision variable space. Note that in all previous experiments, $ITV = 0.4$ was used. This is the value suggested in Castillo and Segura (2019) for a single-objective optimization strategy that was designed with principles similar to those applied in VSD-MOEA. This section is devoted to analyzing the performance of VSD-MOEA when using different ITV values. Note that, since normalized distances are used, the maximum attainable difference is 1. Furthermore, when ITV is set to 0, no individual is penalized based on its contribution to diversity in the decision variable space, so VSD-MOEA would behave like a more traditional MOEA, which means that there is no explicit promotion of diversity in the decision variable space. As a result, the values $ITV = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ were tested. As in previous experiments, the whole set of benchmark problems was used and the stopping criterion was set to 2.5×10^6 function evaluations.

Figure 5.5 shows the box-plots of the mean HV ratios obtained for both the two-objective and the three-objective cases. In comparison to state-of-the-art algorithms, when ITV is set to 0, VSD-MOEA yielded competitive results in the two-objective case and the best results in the three-objective case (see Tables 5.3 and 5.5). Specifically, the mean values were 0.884 and 0.880 for two and three objectives, respectively. This means that the novel density estimator put forth in this chapter is also helpful for methods that do not explicitly take into account the variable space diversity. However, the increase in performance when using other ITV values is clear. The HV ratio obtained quickly increases as higher ITV values up to 0.4 are used. Then, with values in the range $[0.5, 1.0]$, the performance decreases slightly. There is a wide range of values where the performance is very good (e.g. $ITV \geq 0.2$), meaning that the behavior of VSD-MOEA is quite robust. Thus, properly setting this parameter is not a complex task.

5.4 Summary

EAs have been one of the most popular approaches for dealing with complex optimization problems. Their design is a highly complex task that requires the definition of several components. Looking at the differences between single-objective and multi-objective optimizers, it is worth noting that several state-of-the-art single-objective optimizers explicitly consider the diversity of the variable space, particularly when dealing with long-term executions, whereas this is not the case for MOEAs. Moreover, single-objective optimizers that take diversity into account to induce a gradual shift between exploration and intensification have been particularly successful.

This chapter proposes a novel MOEA, called VSD-MOEA, that takes into account the diversity of both decision variable space and objective function space. The main novelty is that the importance given to the different diversities is adapted during the optimization process. In particular, in VSD-MOEA more importance is given to the diversity of the decision variable space in the initial stages, but as the evolution progresses, it assigns more importance to the diversity of the objective function space, meaning that a gradual shift between exploration and intensification is promoted. This is performed using a penalty method that is integrated into the replacement phase. Also included is a novel density estimator based on IGD+ that is used to select from the non-penalized individuals.

The experimental validation carried out shows a notable improvement in VSD-MOEA compared to state-of-the-art MOEAs both in two-objective and three-objective problems. Moreover, our proposal not only improves the state-of-the-art algorithms in long-term and medium-term executions, but it also offers a competitive performance in short-term executions. Scalability analyzes show that as the number of objectives and decision variables increases, the implicit variable space maintained by state-of-the-art MOEAs also increases. Thus, for a large number of objectives and decision variables, explicitly considering the diversity of decision variable space is less helpful. Furthermore, analysis of the initial threshold value, which is an additional parameter required by VSD-MOEA, shows that finding a proper value for this parameter is not a difficult task. Finally, the analysis shows that the novel density estimator has a significant impact on performance, especially in the problems with three objectives. The main conclusion is that state-of-the-art solvers can be significantly improved by explicitly taking into account the diversity of decision variable space and by reducing the importance given to this kind of diversity as the evolution progresses.

Finally, we should mention that integrating the design principles put forth in this chapter with interactive approaches (Deb, 2001), where the decision maker guides the search, is complex because the stopping criterion is usually not known *a priori*. Similarly, applying the principles studied in this chapter to cases where the stopping criterion is set by quality, seems complex. Thus, developing strategies to allow for the integration of the design principle studied in this research in such settings seems a worthwhile area of research.

Chapter 6

AVSD-MOEA/D: Archived Variable Space Diversity Based on Decomposition

Most current Multi-Objective Evolutionary Algorithms (MOEAs) do not directly manage the population's diversity in the variable space. Usually, these kinds of mechanisms are only considered when the aim is to attain diverse solutions in the variable space. This is a remarkable difference with respect to single-objective optimizers, where even when no diverse solutions are required, the benefits of diversity-aware techniques are well-known. The aim of this chapter is to show that the quality of current MOEAs in terms of objective space metrics can be enhanced by integrating mechanisms to explicitly manage the diversity in the variable space. The key is to consider the stopping criterion and elapsed period in order to dynamically alter the importance granted to the diversity in the variable space and to the quality and diversity in the objective space, which is an important difference with respect to niching-based MOEAs. Specifically, more importance is given to the variable space in the initial phases, a balance that is shifted towards the objective space as the evolution progresses. This chapter presents a novel MOEA based on decomposition that relies on these principles by means of a novel replacement phase. Extensive experimentation shows the clear benefits provided by our design principle. The main contributions of this chapter are the design of a decomposition-based MOEA which includes an external archive, the consideration of new biased test problems and the integration of DE operators. This last point aims to prove that the superiority of diversity-aware algorithms proposed in this thesis are quire robust regarding the crossover operators.

6.1 Introduction

In recent years, the development of MOEAs has grown dramatically (Coello et al., 2007; Van Veldhuizen and Lamont, 1998), resulting in effective and broadly applicable algorithms. However, some function features provoke significant degradation of the performance of MOEAs (Huband et al., 2006), meaning that better design principles are still required. The current state-of-the-art MOEAs consider in some way the diversity in the objective space. In some cases, this is done explicitly through density estimators (Beume et al., 2007), whereas in other cases, this is done indirectly (Zhang and Li, 2007). Since optimizing most objective space quality indicators implies attaining a well-spread set of solutions in the objective space, not considering this kind of diversity would result in fairly ineffective optimizers. A quite different condition appears with respect to diversity in the variable space. Since objective space quality metrics do not consider at all the diversity in the variable space, most MOEA designers disregard this diversity.

Alternatively, several state-of-the-art single-objective methods introduce mechanisms to vary the trend of the diversity in the variable space, even if obtaining a diverse set of solutions is not the aim of the optimization (Črepinšek et al., 2013). Instead, this is done to induce a better balance between exploration and exploitation. In fact, the proper management of diversity is considered one of the cornerstones for proper performance (Herrera-Poyatos and Herrera, 2017). Thus, these differences between the design principles applied in single-objective and multi-objective evolutionary algorithms are surprising. Moreover, practitioners have shown that modern MOEAs suffer some drawbacks involving stagnation and premature convergence in subsets of variables (Castillo et al., 2017; Liu et al., 2016, 2014; Usta and Kantar, 2011). As a result, this chapter aims to validate the hypothesis of this thesis, which establishes that incorporating mechanisms to manage the diversity in the variable space might yield important benefits to the field of multi-objective optimization. Note that unlike other proposals, such as niching-based MOEAs (Mahfoud, 1995; Srinivas and Deb, 1994; Tanabe and Ishibuchi, 2019), this chapter is not primarily interested in obtaining a diverse set of solutions in the variable space; rather, in this chapter is stated that the quality of the results in terms of objective-space indicators can be improved further with these kinds of mechanisms.

Since controlling diversity in the variable space is so important in single-objective domains to attain a proper balance between exploration and intensification (Lin and Gen, 2009), a large number of related methods have been devised (Črepinšek et al., 2013). Recent research on single-objective optimization has shown that important advances can be achieved when the balance between exploration and intensification is managed by relating the diversity of the population to the stopping criterion and the elapsed execution period. Specifically, these methods reduce the importance given to preserving diversity as the end of the optimization is approached.

One of the main problems in incorporating the above principle into the design of MOEAs is that measures of the variable and objective spaces must be considered simultaneously. This design principle is based on reducing the importance of diversity in the variable space as generations evolve, so we maintain this decision and indirectly increase the importance granted to diversity and quality in the objective space as the execution progresses. In order to show the validity of our hypothesis, this chapter proposes the *Archived Variable Space Diversity MOEA based on Decomposition* (AVSD-MOEA/D). AVSD-MOEA/D simplifies MOEA/D-DE (Zhang et al., 2009) by deactivating the dynamical resource allocation scheme and disregarding the notion of neighborhood; at the same time, it is extended by including a novel replacement strategy that applies the design principle discussed. Our proposal is validated by taking into consideration MOEA/D-DE (Zhang et al., 2009), NSGA-II (Deb et al., 2002a), R2-EMOA (Trautmann et al., 2013) and NSGA-III (Deb and Jain, 2013). Remarkable benefits are achieved in terms of robustness and scalability.

The rest of this chapter is organized as follows. The AVSD-MOEA/D, which is our proposal, is detailed in section 6.2. Section 6.3 is devoted to an extensive experimental validation with several interesting test problems. Finally, a summary of this chapter is presented in Section 6.4. Note that a proper definition of the sort of problems addressed in this chapter can be found in the Appendix B.

Algorithm 13 Main procedure of AVSD-MOEA/D

Input: N (population and archive size), λ (a set of N weight vectors for decomposition in the population), Λ (a set of weight vectors for the R2-based archive), ITV (initial threshold value), CR (Crossover Rate), F (scale factor), p_m (mutation probability)
Output: A^t (archive with N solutions)
Initialization: Generate an initial population P^0 with N individuals
Evaluation: Evaluate each individual in P^0 and assemble the reference vector $\vec{z}^*$ with the best objective values
Archive Initialization: $A^0 = P^0$
Assign $t = 0$
while (not stopping criterion) **do**
  for each individual $\vec{i} \in P^t$ **do**
    DE variation: Generate solution $\vec{j} \in Q^t$ by applying DE/rand/1/bin with F as the mutation scale factor and CR as the crossover rate, using $\vec{i}$ as the target vector
    Mutation: Apply polynomial mutation to $\vec{j}$ with probability p_m
    Evaluation: Evaluate the new individual $\vec{j}$ and update the reference vector $\vec{z}^*$ with the best objective values.
  end for
  Survivor selection: Generate P^{t+1} by applying the replacement scheme described in Algorithm 15, using P^t, Q^t, λ, $\vec{z}^*$ and ITV as input
  Update Archive: Create A^{t+1} by applying Algorithm 14 using A^t, Q^t, Λ, and $\vec{z}^*$ as input.
  $t = t + 1$
end while
return A^t

6.2 Algorithmic Proposal

This section describes our proposal, the *Archived Variable Space Diversity* MOEA *based on Decomposition* (AVSD-MOEA/D)[1]. The main novelty and motivation behind AVSD-MOEA/D is the incorporation of an explicit way to manage diversity in the variable space, the goal being to improve the behavior in terms of objective space metrics, especially in long-term executions, which is the environment where diversity-aware techniques have excelled (Segura et al., 2015c). Although AVSD-MOEA/D is inspired by MOEA/D, it has been simplified, so in some ways it resembles more mature decomposition-based MOEAs, such as WBGA (Hajela et al., 1993). For example, the notion of neighborhood of subproblems is not used and the dynamic resource allocation that is generally applied in modern variants of MOEA/D is deactivated. The main reason for the simplification is to show that even a simple MOEA that incorporates our design principles can improve further more complex state-of-the-art algorithms.

Our proposal decomposes the MOP into several single-objective problems. Notwithstanding that any scalarization approach can be employed, our strategy applies the *Achievement Scalarizing Function* (ASF), which has reported some of the most effective results in recent years (Hernández Gómez and Coello Coello, 2015). Let $\vec{\lambda}_1, ..., \vec{\lambda}_N$ be a set of weight vectors and $\vec{z}^*$ a reference vector. The MOP is decomposed into N scalar optimization sub-problems as shown in Eqn. (6.1).

$$g^{asf}(\vec{x}|\vec{\lambda}_j, \vec{z}^*) = \max_{1 \leq i \leq m} \left\{ \frac{|f_i(\vec{x}) - z_i^*|}{\lambda_{j,i}} \right\} \tag{6.1}$$

The main novelty of AVSD-MOEA/D appears in the survivor selection scheme. According to some of the most successful single-objective diversity-aware algorithms (Segura et al., 2016a), the replacement strategy relates the degree of diversity in the variable space to the stopping criterion and the elapsed generations. The

[1]The source code in C++ is freely available at https://drive.google.com/drive/folders/1760kQ6OVU6NHNB_BiNPmPZ_xbx40gEft?usp=sharing

Algorithm 14 Procedure to update the R2-based archive

1: Input: A^t (external archive in the current generation), Q^t (offspring of current generation), Λ (weight vectors for $R2$), $\vec{z}^*$ (reference vector)
2: Output: A^{t+1}
3: $R^t = A^t \cup Q^t$
4: $A^{t+1} = \emptyset$
5: **while** $|A^{t+1}| < N$ **do**
6: $\quad \forall \vec{x} \in R^t : r(\vec{x}) = R2(A^{t+1} \cup \{\vec{x}\}; \Lambda, \vec{z}^*)$
7: $\quad \vec{x}^* = argmin(r(\vec{x}) : \vec{x} \in R^t)$
8: $\quad A^{t+1} = A^{t+1} \cup \{\vec{x}^*\}$
9: $\quad R^t = R^t \setminus \{\vec{x}^*\}$
10: **end while**

aim of this relationship is to gradually bias the search from exploration to exploitation as the optimization evolves. Specifically, diversity is explicitly promoted less and less until half of the total generations. Then, in the remaining generations, AVSD-MOEA/D behaves similarly to most popular MOEAs, i.e. the diversity in the variable space is no longer considered explicitly.

The main procedure of AVSD-MOEA/D is shown in Algorithm 13. Its general template is quite standard. The variation step is similar to those used in typical MOEAs, meaning it is based on crossover and mutation and any operators might be used. Specifically, N individuals are created in each generation by randomly selecting at each step three individuals to apply the $DE/rand/1/bin$ operator. Note that each member of the population is used as the target vector once. Then, a polynomial mutation is applied to the output of the DE operator. As in most current MOEA/D variants, the initial population is generated randomly, the number of weight vectors ($|\lambda|$) is equal to the population size, and the reference vector $\vec{z}^*$ used for ASF consists of the best objective values achieved. The weight vectors used in this chapter are based on a uniform design strategy and are specified in the experimental validation section. Finally, the survivor selection stage is applied. This is quite different from traditional techniques, in the sense that parent (P^t) and offspring (Q^t) are merged, which means that unlike in MOEA/D, the position of each individual is not important, and a diversity-aware selection is performed. Since this is the novelty of this chapter, its working operation is given in detail.

Note that AVSD-MOEA/D incorporates an external archive to store the best solutions. While in MOEA/D this is considered optional, in our approach it is quite important because the penalty approach included to pick up the survivors of the population in the replacement phase might discard elite solutions for some weight vectors. Since methods based on the R2 indicator (Trautmann et al., 2013) have reported quite high-quality solutions, our archive is based on the R2 indicator applying the ASF and the weights Λ to generate the utility functions. Therefore, the weight vectors (Λ) considered to update the archive have a different distribution than the ones considered to update the population (λ). In each iteration, the archive selects N candidate solutions by combining its contents with the offspring of each generation (see line 14 in Algorithm 13). This is done by following a simple greedy approach (Algorithm 14). Specifically, it iteratively selects the individual that minimizes the $R2$ (lines 6-9) considering the individuals already selected with ties broken randomly. Note that since $R2$ is weakly Pareto-compliant, it might happen that some non-dominated individuals do not contribute to $R2$, so incorporating more complex archiving strategies might be helpful (Schütze and Hernández, 2021). However, our preliminary experiments showed that this is not overly important for proper performance, so we decided to maintain its simplicity.

Algorithm 15 Replacement Phase of AVSD-MOEA/D

1: Input: P^t (population of current generation), Q^t (offspring of current generation), λ (a set of weight vectors), $\vec{z}^*$ (reference vector), and ITV (initial threshold value).
2: Output: P^{t+1}
3: $R^t = P^t \cup Q^t$
4: $P^{t+1} = \emptyset$
5: $Penalized = \emptyset$
6: $r\lambda = \lambda$
7: $D^t = ITV - ITV \times \frac{G_{Elapsed}}{0.5 * G_{End}}$
8: **while** $|P^{t+1}| < N$ **do**
9: Compute DCS in R^t using P^{t+1} as reference set
10: Move the individuals in R^t with $DCS < D^t$ to $Penalized$
11: **if** R^t is empty **then**
12: Compute DCS of each individual in $Penalized$ set employing P^{t+1} as reference set
13: Move the individual in $Penalized$ with the largest DCS to R^t
14: **end if**
15: Identify the pair of non-penalized individual $\vec{r}_i \in R^t$ and weight vector $\vec{r\lambda}_j \in r\lambda$ with the minimum scalarizing function value according to $g^{asf}(\vec{r}_i | \vec{r\lambda}_j, \vec{z}^*)$
16: Move individual $\vec{r}_i$ to P^{t+1}
17: Erase the weight vector $\vec{r\lambda}_j$ from $r\lambda$
18: **end while**
19: **return** P^{t+1}

6.2.1 Novel Replacement Phase of AVSD-MOEA/D

The purpose of the replacement phase (see Algorithm 15) is to select the set of survivors of the next generation. The survivor selection described in this chapter incorporates similar design principles to those applied in the diversity-aware single-objective optimizer DE-EDM (Castillo and Segura, 2019) (Chapter 3). It operates as follows. First, the parent and offspring populations are merged in a multi-set to establish the candidate set R^t (line 3). A key of the scheme is to promote the selection of individuals with a large enough contribution to diversity in the variable space. Specifically, the contribution of an individual $\vec{x}$ is calculated as $\min_{\vec{p} \in P^{t+1}} Distance(\vec{x}, \vec{p})$, where P^{t+1} is the multi-set of the already picked survivors and the normalized Euclidean distance specified in Eqn. (6.2) is applied. Note that in the pseudocode, the tag DCS (*Distance to Closest Solution*) is used to denote the contribution to diversity.

$$Distance(\vec{a}, \vec{b}) = \left(\frac{1}{n} \sum_{i=1}^{n} \left(\frac{a_i - b_i}{x_i^{(U)} - x_i^{(L)}} \right)^2 \right)^{1/2} \tag{6.2}$$

In order to promote the selection of distant individuals, a threshold D^t is dynamically calculated (line 7) and individuals with a DCS value lower than the threshold are considered undesirable individuals. Note that the calculation of D^t depends on an initial threshold value (ITV), which is a parameter of our proposal, on the number of generations that have evolved ($G_{Elapsed}$) and on the stopping criterion (G_{End}), that is, the number of generations to evolve. Specifically, the value decreases linearly as generations evolve. Since survivors with larger DCS values provoke exploration steps, while survivors with short DCS values promote intensification steps, this linear decrease promotes a gradual transition from exploration to exploitation. Also note that after 50% of the total number of generations, the D^t value is below 0, meaning that no penalties are applied and a more traditional strategy focused only on the objective values is used to perform the selection steps.

The strategy iteratively selects an individual from the candidate set (R^t) to enter the new population (P^{t+1}) until it is filled with N individuals (lines 8-18). In particular, the aim is to select a proper individual for each weight vector, while at the same time fulfilling the condition imposed for the contribution to diversity in the variable space. In order to satisfy this last condition, non-selected individuals with a DCS lower than D^t are moved from R^t to the *Penalized* set (lines 9-10), and in each iteration an individual belonging to R^t is selected to survive. The set of weight vectors considered by our strategy are initially placed in $r\lambda$ (line 6). In each iteration, the individual $\vec{r}_i \in R^t$ with the best scalarizing function for any of the weight vectors in $r\lambda$ is identified (line 15). Then, this individual is selected as a survivor (line 16) and the weight vector used is erased from $r\lambda$ (line 17). Note that N individuals are selected, which means that each weight vector is used to select exactly one individual. Also note that it might happen that R^t is empty prior to selecting N individuals. This means that the diversity is lower than expected, so in order to increase the amount of exploration, the individual with the largest DCS value in the *Penalized* set is selected to survive (lines 11-14).

6.3 Experimental Validation

In this section, we provide the validation of our proposal against state-of-the-art MOEAs and explain the reasons behind the superiority of AVSD-MOEA/D. Since methods that include strategies to explicitly delay convergence usually require additional computational resources to excel, long-term analyses are presented. In order to draw proper conclusions, four experiments were carried out. First, a comparison between AVSD-MOEA/D and four state-of-the-art MOEAs is presented. This comparison focuses on showing the benefits of AVSD-MOEA/D for benchmarks configured in standard ways. Second, an analysis to test the scalability on the number of decision variables is carried out. Third, the robustness of AVSD-MOEA/D in terms of the initial threshold value (ITV) is analyzed. Finally, the ability of dealing with different levels of difficulty in biased test problems is studied. Note that in order to test the quality of the approximations, the hypervolume and attainment surfaces are used. Additionally, our analyses also include some studies to better understand the implications of AVSD-MOEA/D on the diversity in the variable space. These analyzes provide a better understanding of the reasons behind the proper performance of AVSD-MOEA/D and highlight the significant differences between AVSD-MOEA/D and other strategies in terms of the dynamics of the population.

In the first three analyses, our validation takes into account three of the most popular benchmarks in multi-objective optimization: WFG (Huband et al., 2006), DTLZ (Deb et al., 2005), and UF (Zhang et al., 2008) [2]. The last experiment takes into account the BT problems (Li et al., 2016a), which were designed with the aim of incorporating difficult features of real-world MOPs into benchmarks. Unless otherwise stated, standard configurations were used. Specifically, the WFG test problems were used with two and three objectives with 24 parameters[3], 20 of them corresponding to distance parameters and 4 to position parameters. The DTLZ test problems were also used with two and three objectives, and the number of variables in each case was set to $n = m + k - 1$, where $k = \{5, 10, 20\}$ for DTLZ1, DTLZ2 to DTLZ6 and DTLZ7, respectively. The UF benchmark and BT test problems were configured with 30 variables.

Regarding our comparisons, the set of state-of-the-art MOEAs used to validate our proposals is comprised of four popular and complementary MOEAs: NSGA-II (Deb et al., 2002a), MOEA/D-DE (Zhang et al., 2009), R2-EMOA (Trautmann et al., 2013) and NSGA-III (Deb and Jain, 2013). Given that all the algorithms are stochastic, each execution was repeated 35 times in every experiment. The reference point used to calculate

[2] These test problems are defined in the Appendix C.

[3] In the WFG context, the term *parameter* is equivalent to variable.

Table 6.1 Parameterization of the variation phase applied in each MOEA

	2 objectives		**3 objectives**	
	CR	*F*	*CR*	*F*
AVSD-MOEA/D	0.0	0.75	0.0	0.75
MOEA/D-DE	0.75	0.75	0.5	0.5
R2-EMOA	0.75	0.5	0.5	0.5
NSGA-II	0.75	0.5	0.0	0.25
NSGA-III	0.75	0.25	0.5	0.75

the HV is chosen to be a vector whose values are sightly larger (ten percent) than the nadir point, as suggested in Ishibuchi et al. (2017). The value reported is computed as the ratio between the normalized HV attained (Li et al., 2014) and the maximum attainable normalized HV. In this way, a value equal to one means a perfect approximation. Note that this value is not attainable because MOEAs yield discrete approximations. Finally, to statistically compare the HV ratios, a guideline similar to that proposed in (Durillo et al., 2010) was used, which entails the use of the Shapiro-Wilk, Levene, ANOVA, Welch and Kruskal-Wallis tests. They were applied assuming a significance level of 5%. An algorithm $A1$ is said to beat an algorithm $A2$ when the differences between the HV ratios attained are statistically significant, and the mean and median HV ratios obtained by $A1$ are higher than the mean and median achieved by $A2$.

An important step to perform fair comparisons is the parameterization of algorithms. Note that the variation operators used in each algorithm in their original variants differ. Using the original variation operators to perform comparisons is not fair, and probably would offer more conclusions about the effectiveness of the operators than about the general framework proposed in each MOEA. However, there might also be a dependency between the general framework and the proper variation operators. We thus decided to use a common simple framework for the variation step, but to allow a different parameterization for each algorithm. Specifically, the variation phase first applies the classic DE scheme known as DE/rand/1/bin with parameters F and CR, and then it applies a polynomial mutation with probability p_m and a distribution index equal to 50. Note that the use of the additional mutation in variants based on MOEA/D is quite important (Zhang et al., 2009). An additional common parameter is the population size. Since the hypervolume is highly dependent on the number of solutions used to approximate the Pareto front, all MOEAs were configured with a common population size equal to 100 individuals.

In order to set the parameters CR, F, and p_m, 40 parameterizations were tested for each algorithm. They were generated by combining four values of F (0.25, 0.5, 0.75 and 1.0), five values of CR (0.0, 0.25, 0.5, 0.75, 1.0) and two values of p_m (0.0, $\frac{1}{n}$). These configurations were executed by setting the stopping criterion to 2.5×10^6 function evaluations, with all the aforementioned benchmarks. The mean of the resulting HV ratios were calculated independently for the problems with two and three objectives. Then, in the experiments that follow, the parameter configuration that attained the largest mean was used. Table 6.1 shows the configuration of CR and F selected for each MOEA. Note that all of them yielded better results when mutation was enabled, so the benefits reported in Zhang et al. (2009) also appeared for other MOEAs. Note that in the case of AVSD-MOEA/D, CR was set to 0, which reduces the strength of the perturbation performed by DE. Since AVSD-MOEA/D maintains a larger degree of diversity than other methods, low disruptive operators seem to be more helpful.

Note also that there are some additional parameters that are specific to some of the MOEAs. They were set to typical values used in literature. Table 6.2 shows this additional parameterization. Note also that scalarization functions are used in MOEA/D-DE, R2-EMOA, NSGA-III and AVSD-MOEA/D. In all of those cases, the ASF approach is used. However, the weight vectors used in R2-EMOA are different from those of the remaining methods because in R2-EMOA, using a larger number of weight vectors than the size of the population is

Table 6.2 Configuration of the specific parameters of each MOEA

Algorithm	Configuration
MOEA/D-DE	Max. updates by subproblem (η_r) = 2, tour selection = 10, neighbor size = 20, period utility updating = 50 generations, local selection probability (δ) = 0.9
R2-EMOA	$\rho = 1$, offspring by iteration = 1
AVSD-MOEA/D	$ITV = 0.4$

Table 6.3 Summary of the hypervolume ratios attained for problems with two objectives

	AVSD-MOEA/D			**MOEA/D-DE**			**NSGA-II**			**NSGA-III**			**R2-EMOA**		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
WFG1	0.995	0.982	0.020	0.957	0.842	0.058	0.994	0.966	0.026	0.993	**0.989**	0.011	0.993	0.921	0.039
WFG2	0.999	**0.999**	0.000	0.996	0.996	0.000	0.998	0.998	0.000	0.997	0.990	0.013	0.998	0.998	0.000
WFG3	0.993	**0.993**	0.000	0.992	0.992	0.000	0.980	0.978	0.001	0.992	0.992	0.000	0.992	0.991	0.000
WFG4	0.991	**0.991**	0.000	0.988	0.988	0.000	0.979	0.975	0.002	0.988	0.986	0.003	0.988	0.973	0.007
WFG5	0.933	**0.905**	0.008	0.891	0.882	0.004	0.883	0.878	0.002	0.895	0.888	0.003	0.890	0.885	0.003
WFG6	0.959	0.922	0.020	0.988	0.963	0.019	0.977	0.974	0.001	0.956	0.934	0.013	0.991	**0.990**	0.001
WFG7	0.991	**0.991**	0.000	0.988	0.988	0.000	0.980	0.977	0.001	0.988	0.988	0.000	0.991	**0.991**	0.000
WFG8	0.963	**0.954**	0.004	0.846	0.833	0.004	0.825	0.815	0.003	0.829	0.826	0.001	0.835	0.832	0.001
WFG9	0.978	**0.976**	0.002	0.974	0.954	0.039	0.941	0.873	0.071	0.798	0.796	0.001	0.975	0.939	0.051
DTLZ1	0.993	**0.993**	0.000	0.993	**0.993**	0.000	0.992	0.991	0.000	0.993	**0.993**	0.000	0.992	0.992	0.000
DTLZ2	0.991	0.991	0.000	0.989	0.989	0.000	0.987	0.986	0.001	0.989	0.989	0.000	0.992	**0.992**	0.000
DTLZ3	0.991	0.991	0.000	0.989	0.989	0.000	0.989	0.989	0.000	0.989	0.989	0.000	0.992	**0.992**	0.000
DTLZ4	0.991	**0.991**	0.000	0.989	0.989	0.000	0.987	0.903	0.231	0.989	0.989	0.000	0.992	0.803	0.320
DTLZ5	0.991	0.991	0.000	0.989	0.989	0.000	0.987	0.986	0.001	0.989	0.989	0.000	0.992	**0.992**	0.000
DTLZ6	0.991	**0.991**	0.000	0.989	0.986	0.014	0.989	0.989	0.000	0.989	0.989	0.000	0.992	0.985	0.021
DTLZ7	0.997	**0.997**	0.000	0.996	0.996	0.000	0.997	0.996	0.000	0.996	0.996	0.000	0.997	**0.997**	0.000
UF1	0.995	**0.995**	0.000	0.987	0.986	0.001	0.989	0.988	0.001	0.992	0.991	0.001	0.992	0.992	0.000
UF2	0.995	**0.995**	0.000	0.990	0.988	0.001	0.984	0.982	0.001	0.986	0.981	0.003	0.988	0.987	0.001
UF3	0.938	0.906	0.016	0.991	**0.990**	0.001	0.988	0.985	0.004	0.985	0.968	0.019	0.991	0.982	0.005
UF4	0.979	**0.977**	0.001	0.914	0.904	0.006	0.892	0.882	0.005	0.880	0.876	0.003	0.902	0.893	0.003
UF5	0.990	**0.975**	0.009	0.715	0.439	0.137	0.792	0.734	0.087	0.777	0.654	0.067	0.792	0.733	0.092
UF6	0.962	**0.938**	0.013	0.928	0.748	0.175	0.870	0.720	0.069	0.820	0.708	0.043	0.827	0.691	0.091
UF7	0.993	**0.993**	0.000	0.991	0.990	0.001	0.980	0.976	0.002	0.983	0.975	0.002	0.992	0.982	0.006
Mean	0.983	**0.976**	0.004	0.960	**0.931**	0.020	0.956	**0.937**	0.022	0.948	**0.934**	0.008	0.960	**0.936**	0.028

beneficial. As in the official code, R2-EMOA was applied with 501 and 496 weight vectors for two and three objectives, respectively (Trautmann et al., 2013). In the remaining cases — including AVSD-MOEA/D — the number of weight vectors was equal to the population size, and they were generated with the uniform design strategy described in Wagner et al. (2013). In the case of AVSD-MOEA/D, a second set of weight vectors (Λ) was considered for the external archive. Since the archive is based on $R2$, it considers the same weight vectors as R2-EMOA.

6.3.1 Performance of MOEAs in long-term executions

One of the aims behind the design of AVSD-MOEA/D is to profit from long-term executions. Therefore, in this section, we present the results attained by the different algorithms when setting the stopping criterion to 2.5×10^7 function evaluations. The results presented in this section are discussed in terms of the HV ratios and attainment surfaces. A similar analysis in terms of the indicator IGD+ (Ishibuchi et al., 2015) was carried out. This analysis provided similar conclusions, so this is included in the supplementary material in the Appendix E. Table 6.3 shows the HV ratios obtained for the benchmark functions with two objectives. For each method and problem, the best, mean, and standard deviation of the HV ratio values are reported. Furthermore, in order to summarize the results attained by each method, the last row shows the mean for the whole set of problems.

Table 6.4 Statistical Tests and Deterioration Level of the HV ratio for problems with two objectives

	↑	↓	↔	Score	Deterioration
AVSD-MOEA/D	78	13	1	65	0.160
MOEA/D-DE	41	50	1	-9	1.181
NSGA-II	21	66	5	-45	1.057
NSGA-III	35	52	5	-17	1.119
R2-EMOA	47	41	4	6	1.066

For each test problem, the method that yielded the largest mean and those that were not statistically inferior to the best are shown in **boldface**. Similarly, the method that yielded the best HV value among all the runs is underlined. From now on, the methods shown in **boldface** for a given problem are referred to as the winning methods. AVSD-MOEA/D, R2-EMOA, MOEA/D-DE, NSGA-III and NSGA-II belonged to the winning methods in 17, 6, 2, 2 and 0 problems, respectively. The superiority of AVSD-MOEA/D is clear both in terms of this metric and in terms of the mean HV. In particular, AVSD-MOEA/D reached a mean value equal to 0.976, while all remaining methods reached values between 0.931 and 0.937. A careful inspection of the data shows that in those cases where AVSD-MOEA/D loses, the difference with respect to the best method is low. In fact, the difference between the mean HV ratio attained by the best method and by AVSD-MOEA/D is never greater than 0.1. However, in all other methods, there were several problems in which the distance from the best approach was greater than 0.1. Specifically, it occurred in 4, 4, 4 and 5 problems for R2-EMOA, MOEA/D-DE, NSGA-II, and NSGA-III, respectively. This means that AVSD-MOEA/D wins in most cases and that when it loses, the difference is always small. Note also that in terms of standard deviation, AVSD-MOEA/D yields much lower values than all other algorithms, which means that it is quite robust.

In order to better clarify these findings, pair-wise statistical tests were applied between each method tested in each test problem. For the two-objective cases, Table 6.4 shows the number of times that each method statistically won (column ↑), lost (column ↓) or tied (column ↔). The **Score** column shows the difference between the number of times that each method won and the number of times that each method lost. Additionally, for each method μ, we calculated the sum of the differences between the mean HV ratio attained by the best method (the ones with the highest mean) and method μ, for each problem where μ was not in the group of winning methods. This value is shown in the *Deterioration* column. The data confirm that although AVSD-MOEA/D loses in some pair-wise tests, the overall numbers of wins and losses clearly favor AVSD-MOEA/D. More importantly, the total deterioration is much lower in the case of AVSD-MOEA/D, confirming that when AVSD-MOEA/D loses, the differences are low.

Tables 6.5 and 6.6 show the same information for the problems with three objectives. In this case, the number of times that each method belonged to the winning groups were 17, 2, 0, 0 and 0 for AVSD-MOEA/D, R2-EMOA, MOEA/D-DE, NSGA-III and NSGA-II, respectively. Thus, AVSD-MOEA/D yielded quite superior results. Considering the whole set of problems, AVSD-MOEA/D obtained a much larger mean HV ratio than the other ones. Moreover, the difference between the mean HV ratio obtained by the best method and by AVSD-MOEA/D was never greater than 0.1. However, all the other methods exhibited a deterioration in excess of 0.1 in several cases. In particular, this happened in 2, 2, 2 and 6 problems for MOEA/D-DE, R2-EMOA, NSGA-III, and NSGA-II, respectively. Interestingly, AVSD-MOEA/D is quite superior in both total deterioration and the score generated from pair-wise statistical tests. In fact, its deterioration for the entire problem set is just 0.006. Beating all state-of-the-art algorithms on such a large number of problem benchmarks is a quite significant achievement, and shows the robustness of AVSD-MOEA/D. Our results show that the superiority of AVSD-MOEA/D persists, and even increases, when problems with three objective functions are considered. In order to better illustrate the important impact contributed by AVSD-MOEA/D on the results obtained, the

Table 6.5 Summary of the hypervolume ratios attained for problems with three objectives

	AVSD-MOEA/D			**MOEA/D-DE**			**NSGA-II**			**NSGA-III**			**R2-EMOA**		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
WFG1	0.985	**0.982**	0.007	0.972	0.937	0.030	0.966	0.959	0.008	0.970	0.967	0.008	0.981	0.965	0.017
WFG2	0.991	**0.991**	0.000	0.981	0.979	0.001	0.974	0.967	0.003	0.972	0.971	0.001	0.963	0.963	0.000
WFG3	0.995	**0.994**	0.000	0.990	0.990	0.000	0.986	0.975	0.006	0.966	0.954	0.007	0.992	0.992	0.000
WFG4	0.943	**0.941**	0.001	0.899	0.898	0.001	0.892	0.876	0.008	0.897	0.897	0.000	0.911	0.906	0.002
WFG5	0.901	**0.872**	0.011	0.831	0.831	0.000	0.828	0.812	0.009	0.833	0.827	0.003	0.849	0.846	0.001
WFG6	0.912	0.888	0.011	0.887	0.862	0.013	0.851	0.822	0.013	0.880	0.858	0.012	0.902	**0.893**	0.006
WFG7	0.943	**0.942**	0.001	0.899	0.898	0.001	0.892	0.865	0.010	0.897	0.897	0.000	0.906	0.904	0.001
WFG8	0.910	**0.902**	0.003	0.816	0.812	0.003	0.759	0.748	0.006	0.810	0.807	0.002	0.827	0.824	0.001
WFG9	0.910	**0.894**	0.006	0.875	0.862	0.005	0.819	0.732	0.019	0.858	0.749	0.027	0.886	0.880	0.002
DTLZ1	0.967	**0.967**	0.000	0.953	0.953	0.000	0.950	0.941	0.004	0.953	0.953	0.000	0.942	0.941	0.001
DTLZ2	0.945	**0.944**	0.000	0.914	0.914	0.000	0.905	0.892	0.008	0.913	0.913	0.000	0.916	0.915	0.001
DTLZ3	0.945	**0.944**	0.000	0.914	0.914	0.000	0.901	0.883	0.009	0.913	0.913	0.000	0.916	0.915	0.001
DTLZ4	0.945	**0.944**	0.000	0.914	0.914	0.000	0.908	0.813	0.238	0.913	0.903	0.059	0.916	0.893	0.127
DTLZ5	0.985	0.985	0.000	0.979	0.979	0.000	0.986	0.984	0.001	0.967	0.959	0.005	0.986	**0.986**	0.000
DTLZ6	0.985	**0.985**	0.000	0.979	0.959	0.038	0.984	0.955	0.127	0.958	0.948	0.007	0.986	0.985	0.008
DTLZ7	0.970	**0.968**	0.001	0.922	0.922	0.000	0.941	0.924	0.025	0.929	0.912	0.008	0.907	0.848	0.020
UF8	0.922	**0.916**	0.003	0.891	0.862	0.032	0.747	0.695	0.035	0.890	0.835	0.101	0.893	0.877	0.016
UF9	0.957	**0.951**	0.003	0.947	0.813	0.071	0.822	0.735	0.069	0.954	0.936	0.043	0.942	0.862	0.077
UF10	0.831	**0.787**	0.041	0.681	0.435	0.147	0.543	0.483	0.084	0.624	0.458	0.127	0.579	0.561	0.042
Mean	0.944	**0.937**	0.005	0.908	**0.881**	0.018	0.877	**0.845**	0.036	0.900	**0.877**	0.022	0.905	**0.892**	0.017

Table 6.6 Statistical Tests and Deterioration Level of the HV ratio for problems with three objectives

	↑	↓	↔	**Score**	**Deterioration**
AVSD-MOEA/D	74	2	0	72	0.006
MOEA/D-DE	33	38	5	-5	1.075
NSGA-II	9	64	3	-55	1.745
NSGA-III	22	50	4	-28	1.149
R2-EMOA	45	29	2	16	0.851

50% attainment surfaces (Knowles, 2005) for WFG8 and UF5 are shown in Figure 6.1[4]. Note that the selected problems are among the most difficult ones. WFG8 presents strong dependencies among all the variables, and UF5 is a multi-modal biased problem whose Pareto optimal front is discrete and consists of 21 points. In both problems AVSD-MOEA/D was the only one that converged adequately to the Pareto front in at least the 50% of the runs. Moreover, since the standard deviation is quite low, the proper convergence also appears even if larger percentage values are used to generate the attainment surfaces.

In order to better understand the reasons behind the benefits of AVSD-MOEA/D against state-of-the-art MOEAs, the trend of the HV values and the diversity along the run is analyzed. Note that in some MOPs, variables can be classified into two types: distance variables and position variables (see Section 2.2.4). This is important because in some cases, MOEAs do not maintain a large enough diversity in the distance variables (Castillo et al., 2017), so analyzing the diversity trend for these kinds of variables provides an useful insight into the dynamics of the population.

In order to show the behavior of the different schemes, we selected WFG5 and UF5. They are complementary in the sense that in WFG5, all Pareto solutions exhibit constant values for distant variables, which is not the case in UF5. Moreover, in UF5, the optimal regions are isolated in the variable space, meaning that a larger diversity is required. For each algorithm, the diversity is calculated as the mean Euclidean distance between individuals (ADI) in the population considering only the distance variables (see Appendix B.1.6). Figures 6.2 and 6.3 show the evolution of the ADI (top) and the mean of HV (bottom) for WFG5 and UF5, respectively. In the WFG5 problem, the distance variables quickly converged to a small region in state-of-the-art MOEAs.

[4]The attainment surfaces are explained in Appendix B.1.9

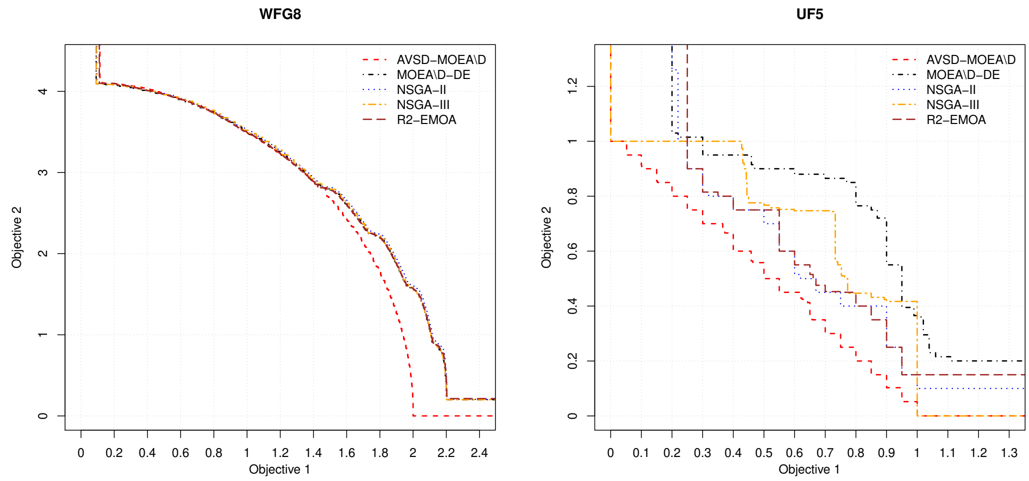

Fig. 6.1 50% attainment surfaces achieved for WFG8 and UF5 test problems.

Thus, the differential evolution operator loses it exploring power and as a result, those MOEAs were unable to significantly improve the quality of the approximations as the evolution progresses. On the contrary, in the case of AVSD-MOEA/D, the decrease in ADI is quite linear until the midpoint of the execution, and the increase in HV is gradual. The final HV attained by AVSD-MOEA/D is the largest, which shows the important benefit of gradually decreasing the diversity.

As expected, explicitly promoting diversity is also beneficial for problems with disconnected optimal regions. As the data in Figure 6.3 show, the advantage of promoting diversity in the UF5 test problem is clear. In this case, state-of-the-art algorithms maintain a non-negligible degree of diversity in the distance variables for the entire search. However, a large degree of diversity is required to obtain the 21 optimal solutions, and these MOEAs do not maintain the required amount of diversity and, as a result, they miss many of the solutions. In the case of AVSD-MOEA/D, enforcing a large degree of diversity in the initial phases promotes more exploration, which makes it possible to find additional optimal regions. Once these regions are located, they are not discarded, which means that a higher level of diversity is maintained throughout the execution. This way, AVSD-MOEA/D not only attained better HV values for the first 10% of the total function evaluations, but it also kept looking for promising regions. In fact, its HV values improved significantly until the midpoint of the execution period, that is, the final moment when diversity was explicitly promoted. Then, an additional increase was obtained due to intensification in the regions identified. This analysis shows that the dynamic of the population depends on the problem at hand. The behavior of AVSD-MOEA/D with all the problems tested was similar to those already presented. Scenarios where the optimal regions consists of constant values for the distance variables behave like WFG5, whereas the behavior in those cases where the optimal regions consist of non-constant values for the distance variables is more similar to the UF5 case. Note, however, that in these cases, different levels of diversity are required, so the behavior is not as homogeneous.

6.3.2 Analysis of Scalability in the Decision Variables

In order to gain a deeper insight into the benefits of our proposal, an analysis of the scalability in terms of the number of decision variables is presented. Given the computational cost associated with this experiment, it

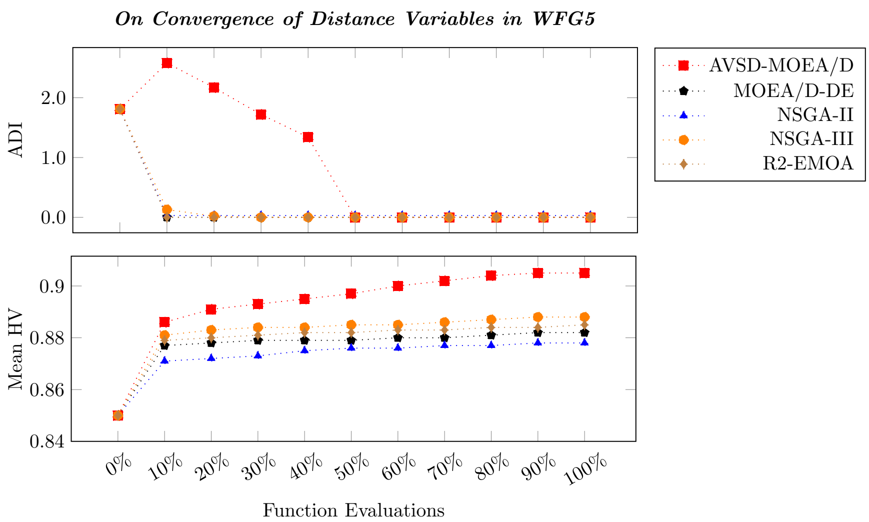

Fig. 6.2 Diversity of distance variables (top) and mean of HV ratios (bottom) vs. elapsed function evaluations in the two-objective WFG5 test problem. The results reported were taken from 35 runs.

was performed only for the decomposition-based algorithms. AVSD-MOEA/D and MOEA/D-DE were applied to the same benchmark problems as in the previous experiment, but considering 50, 100, 250 and 500 variables. Note that in the WFG test problems, the number of position variables and distance variables must be specified. The number of distance variables was set to 42, 84, 210 and 418 when using 50, 100, 250, and 500 variables, respectively. The remaining decision variables were position variables, that is, there were 8, 16, 40 and 82 such variables, respectively. Therefore, the relationship between the number of position and distance variables was kept fixed. In addition, the stopping criterion was set to 2.5×10^7 function evaluations. Figure 6.4 shows the mean HV ratio for the selected algorithms, considering the problems with two and three objectives. As expected, the HV ratio decreased as the number of variables increased. However, the performance of AVSD-MOEA/D is quite robust, and its decrease is less pronounced than that of MOEA/D-DE, meaning that AVSD-MOEA/D is more helpful as complexity increases. In fact, in our previous analyzes, AVSD-MOEA/D also stood out in the most complex cases, such as WFG8 and UF5.

6.3.3 Analysis of the Initial Threshold Value

One of the main potential downsides of including a strategy to explicitly promote diversity is that this is usually at the cost of incorporating additional parameters. In the case of AVSD-MOEA/D, it requires setting the Initial Threshold Value (ITV). Given that in single-objective cases, values close to 0.4 had yielded proper performance (Castillo and Segura, 2019; Romero Ruiz and Segura, 2018) (see Chapter 3), $ITV = 0.4$ was used in previous experiments. This section provides a more detailed study of the implications of this parameter.

In order to better understand the importance of ITV, the entire set of benchmark problems was tested with different values of ITV. As in previous experiments, the stopping criterion was set to 2.5×10^7 function evaluations. Since normalized distances are used, the maximum attainable distance between individuals is 1.0. Setting ITV to 0 implies that diversity is not promoted in the variable space, so several values in the range $[0,1]$ were considered. Specifically, the values $ITV = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ were tested. Figure 6.5 shows the mean HV ratio obtained for both the two-objective and the three-objective

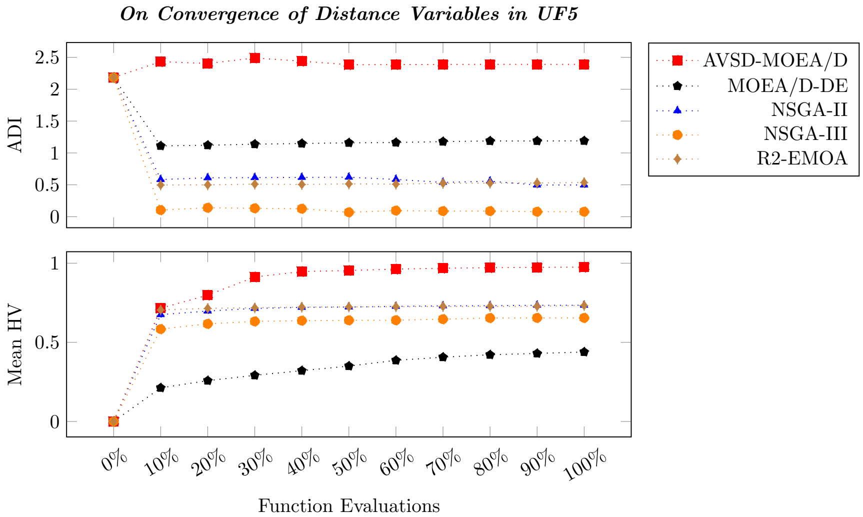

Fig. 6.3 Diversity of distance variables (top) and mean of HV ratios (bottom) vs. elapsed function evaluations in the bi-objective UF5 test problem. The results reported were taken from 35 runs.

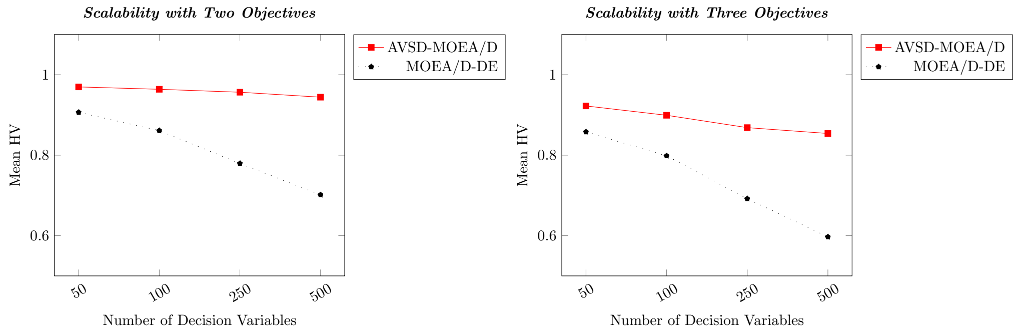

Fig. 6.4 Mean of the HV ratio for 35 runs of two-objective and three-objective problems for different numbers of variables

case with the ITV values tested. The AVSD-MOEA/D performed worst when ITV was set to 0. The HV ratio quickly increased as higher ITV values up to 0.2 were used. Larger values yielded quite similar performances. Therefore, a wide range of values (from 0.2 to 1.0) showed very good performance,which means that the behavior of AVSD-MOEA/D is quite robust. Thus, properly setting this parameter is not a complex task.

In order to better understand the implications of ITV on the dynamics of the population, Figure 6.6 shows, for AVSD-MOEA/D, the evolution of diversity in the distance variables in the WFG9 case for three different values of ITV. When setting $ITV = 0$, the diversity is reduced quite quickly, resulting in premature convergence. The result is a hypervolume that is not too high. However, when $ITV = 0.4$ and $ITV = 1$ are used, the loss of diversity is slowed down, and the resulting hypervolume is quite large. Note that setting $ITV = 1$ promotes greater diversity, so the hypervolume increases slower than when $ITV = 0.4$. However, the degree of exploration in both cases is enough to yield high-quality solutions. The behavior is quite similar in every problem, which explains the stability of the algorithms for different values of ITV. Note that for shorter periods, setting a proper ITV value is probably much more important. However, for long-term executions at least, practically any value higher than 0.2 yields similar solutions, which we regard as a highly positive feature.

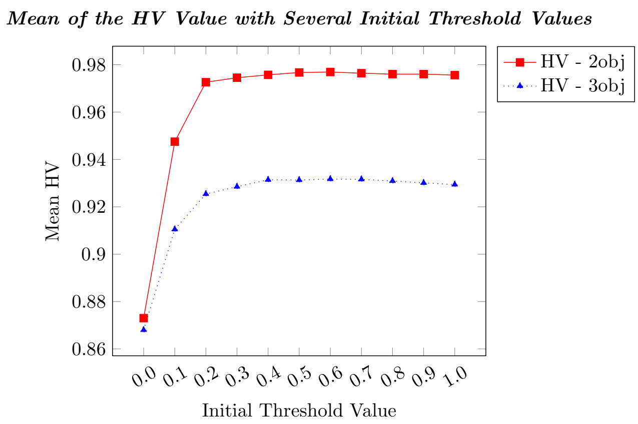

Fig. 6.5 Mean of HV values for all the problems with several initial threshold values

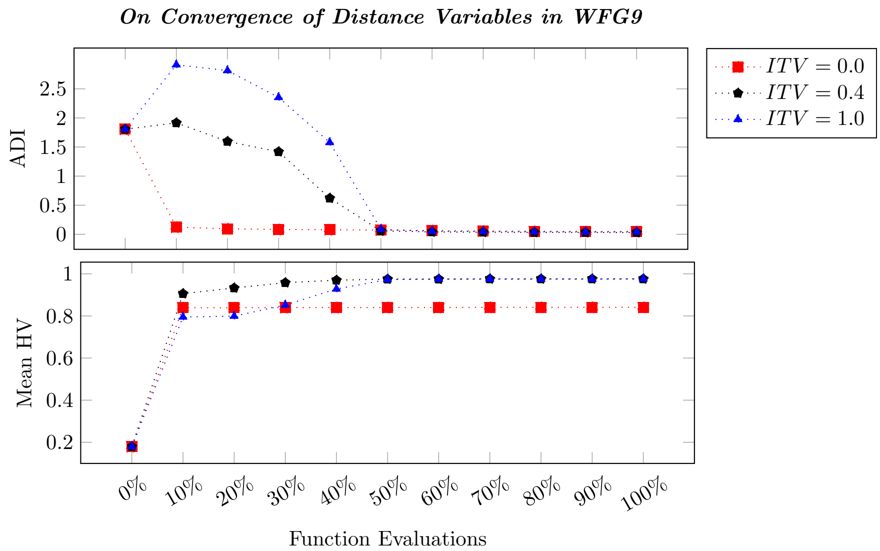

Fig. 6.6 Diversity of distance variables (top) and mean of HV ratios (bottom) vs. elapsed function evaluations in the bi-objective WFG9 test problem. The results reported were taken from 35 runs.

6.3.4 On the Convergence of MOEAs in Test Problems with Bias Features

As pointed out in Deb (1999); Huband et al. (2006); Li et al. (2016a), the bias feature is one of the most challenging difficulties that MOEAs might face. Recently, the BTs test problems were proposed to facilitate the study of the ability of MOEAs to deal with bias. In this context bias means that small variations in the decision space around the Pareto set cause significant changes in the objective space Huband et al. (2006). Two kinds of bias are included in this set of problems: position-related bias and distance-related bias. A position-related bias means that a small change in a position-related variable implies a significant displacement along the Pareto front, and a distance-relate bias means that a small variation in a distance-related variable of one solution in the Pareto set causes a significant change on the proximity to the Pareto front. Note that these are definitions proposed by the authors; however, there are more factors to consider. One of the aims behind the design of BTs was to propose problems with a controllable bias. In particular, the amount of bias caused by distance-related variables is modified with the parameter θ, while the parameter γ is used to control the bias caused by position-related

Table 6.7 Initial distance-related bias values (θ_0) that are assigned to each BT Problem

Problem	**Bias (θ_0)**	**Problem**	**Bias (θ_0)**	**Problem**	**Bias (θ_0)**
BT1	10^{-10}	**BT4**	10^{-8}	**BT7**	10^{-3}
BT2	$\frac{1}{5}$	**BT5**	10^{-10}	**BT8**	10^{-3}
BT3	10^{-8}	**BT6**	10^{-4}	**BT9**	10^{-9}

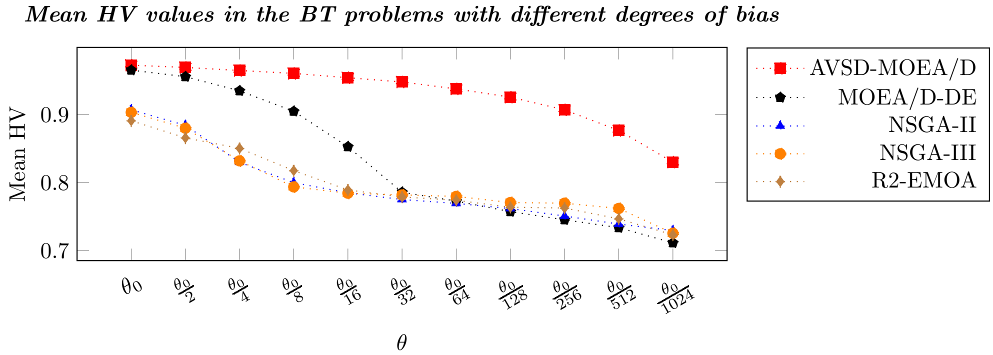

Fig. 6.7 Mean of HV values for eight BTs problems against several degrees of bias (θ).

variables. Particularly, as the parameter values decrease, the amount of bias, and consequently, the difficulty, increases.

Our first analysis involved executing the five tested algorithms with the θ and γ values proposed in Li et al. (2016a). Both decomposition-based approaches (AVSD-MOEA/D and MOEA/D-DE) attained important benefits compared to the remaining alternatives. However, differences between the decomposition-based approaches were not very large. In order to analyze the capability of the MOEAs to deal with different levels of bias features and get more interesting conclusions, an extended analysis that involves using different θ values was designed. The main aim behind the experiment is to analyze the robustness of the different optimizers against increasing bias. For each problem, the base case (θ_0) considers the distance-related bias applied in Li et al. (2016a) (see Table 6.7), as well as the γ established in that chapter. Then, the distance-related bias value (θ) is iteratively decreased by a factor of two, until it reaches the value $\frac{\theta_0}{1024}$. Note that the BT2 problem considers a quite large θ_0 value. The reason is that it performs a transformation based on an exponential that cannot be applied with small θ values with enough accuracy with the standard double representation. Thus, this problem was discarded.

Figure 6.7 shows the mean HV ratio obtained with the different values of θ, considering the remaining eight BT problems. As mentioned previously, with the original configuration (bias equal to θ_0), AVSD-MOEA/D is just sightly better than MOEA/D-DE. However, as soon as θ is decreased to $\frac{\theta_0}{32}$, the performance of MOEA/D-DE decays significantly. However, the mean HV attained by AVSD-MOEA/D is larger than 0.9 with values of θ lower than or equal to $\frac{\theta_0}{256}$, which is quite superior to the state of the art MOEAs, whose values in that region are approximately 0.75. Figure 6.8 shows the 50% attainment surface of BT6, BT7 and BT8 with a bias equal to $\frac{\theta_0}{32}$. In BT6, MOEA/D-DE converged to a region that AVSD-MOEA/D did not reach. However, AVSD-MOEA/D identifies a much larger region that is not reached by MOEA/D-DE, meaning that in this case, promoting diversity in the decision space results in an increased diversity in the objective space. In addition, AVSD-MOEA/D converges quite properly in complicated non-linear problems, as is the case of BT7. Finally, the most important superiority appears in the BT8 case. Note that in this case, besides the bias, the problem is multimodal, so this

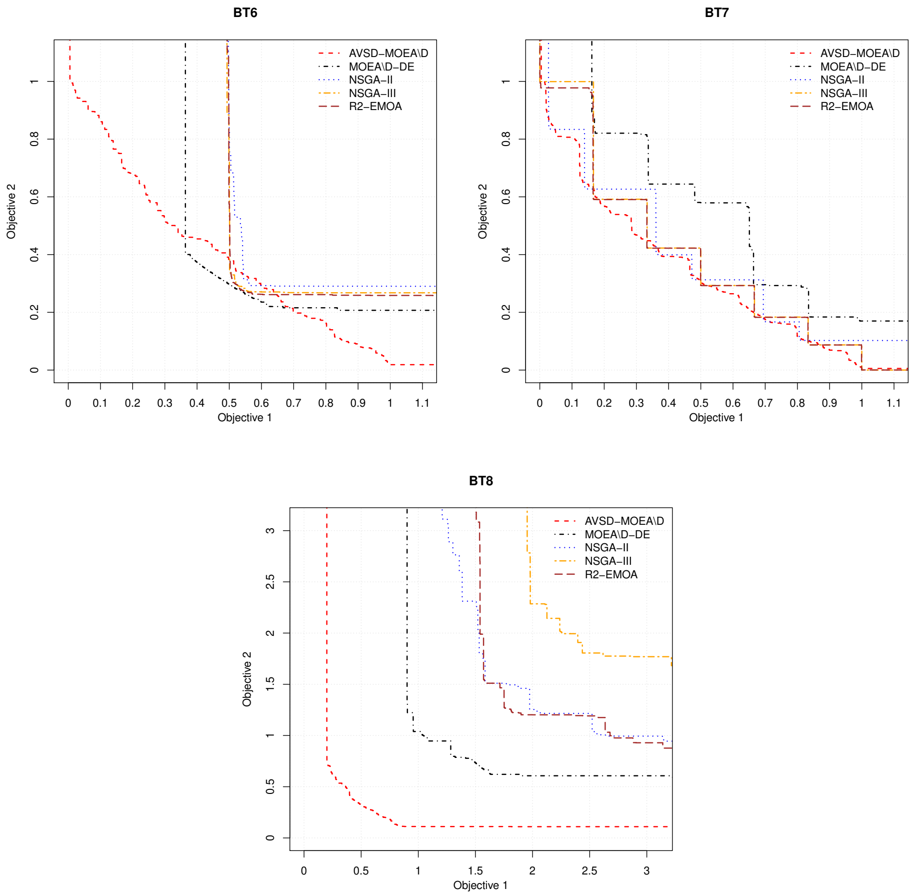

Fig. 6.8 50% attainment surfaces achieved for BT6, BT7 and BT8 with $\theta = \frac{\theta_0}{32}$

is probably the most complex problem considered in this research. More information about BT's problem can be reviewed in Appendix C.

6.4 Summary

Premature convergence is one of the most typical drawbacks of EAs. MOEAs indirectly promote the preservation of diversity in the variable space because of the implicit relationship between the diversity maintained in the objective space and the one maintained in the variable space. However, for many problems, the degree of diversity maintained is not sufficient to ensure the exploratory power of genetic operators, reducing the probability of locating the optimal regions. In single-objective optimization, many of the state-of-the-art algorithms explicitly manage diversity. Specifically, schemes that relate the degree of diversity to the elapsed period of execution and the stopping criterion have excelled. This chapter shows that this design principle is also helpful in the area of multi-objective optimization, where the optimization of many of the most complex popular benchmark problems can be improved further by applying this design principle.

In order to prove the hypothesis expressed in this thesis, a replacement operator based on the design principle aforementioned is applied to generate a decomposition-based MOEA that takes into account the diversity in both the variable and objective spaces. This is done using a dynamic penalty method. Note that since the aim of the approach is to improve the results when considering metrics in the objective space, the importance given to the diversity in the variable space is reduced as the evolution progresses, meaning that in the later phases, our proposal behaves more similarly to the traditional MOEAs. Additionally, taking into account recent advances, and to ensure that our proposal maintains high-quality solutions despite the penalty scheme, an external archive based on the R2-indicator is integrated. As a result, the proposal presented in this chapter is called the *Archived Variable Space Diversity MOEA based on Decomposition* (AVSD-MOEA/D).

The experimental validation carried out shows the remarkable improvement provided by AVSD-MOEA/D compared to the state-of-the-art MOEAs with both two-objective and three-objective problems. Scalability analyzes show that as the number of decision variables increases, the benefits of including proper diversity management are even more important, so performance differences with respect to state-of-the-art MOEAs increase. In fact, the most remarkable benefits emerge for the most complex cases, such as biased multi-modal problems. Moreover, the analysis of the initial threshold value, which is an additional parameter required by AVSD-MOEA/D, shows that the method is quite robust, which makes finding a proper parameter value an easy task. Finally, to better understand the reasons behind the huge superiority of our proposal, some analyses involving the dynamics of the populations are provided. In comparison to state-of-the-art algorithms, our proposal clearly slows down convergence.

Chapter 7

A Diversity-Aware Indicator-Based Multi-Objective Evolutionary Algorithm with Incorporation of Preference Information

Abstract

Indicator-based MOEAs are algorithms quite popular with desirable mathematical properties that under simple modifications can integrate preference information imposed by a Decision Maker (DM). This chapter shows that explicitly managing the amount of diversity maintained in the decision variable space is beneficial to better fulfill the preference of a DM. Without loss of generality, the proposed framework is integrated with two indicators: the hypervolume-indicator and the R2-indicator. The motivation behind these indicators is that both are Pareto-compliant and weakly Pareto-compliant, respectively. As a result the order induced by the dominance relation is respected. The indicator-based MOEA proposed in this chapter explicitly considers the diversity in both the decision variable space and objective space. The algorithmic proposal is designed in such a way that at the initial stages the diversity in the decision variable has greater relevance and as the evolution progresses, the optimizer gradually grants greater relevance to the preference information imposed by the DM. The experimental validation shows that this principle has positive effects on the final results.

7.1 Introduction

Multi-criteria Decision Making (MCDM) is an aspect of multi-objective optimization that involves methods to help a decision maker (DM) in making better decisions in a conflicting scenario (Ojha et al., 2019). Since MOEAs have been fruitful methods to approximate MOPs, these strategies are used in the MCDM field, better known as MOEAs with preference information. Although, MOEAs are designed to search for non-dominated solutions that better approximate the entire Pareto front, in some circumstances (e.g. a very large Pareto front) it is better to focus on a specific region of the Pareto front using some preferences. Such preference information is used to guide the search towards the region of the Pareto front which is of interest to the Decision Maker (DM). In the literature, indicator-based algorithms have been successfully applied to satisfy DM preferences (Trautmann et al., 2013; Zitzler et al., 2007; Zitzler and Künzli, 2004). A quality indicator is a function that evaluates the quality of Pareto-front approximations (Guerreiro et al., 2020; Zitzler and Künzli, 2004).

Although the benefits attained by MOEAs are impressive, DM should be aware that under some circumstances Evolutionary Algorithms (EAs) might fail to attain high-quality solutions for some problems. A well-known drawback that has appeared in the application of EAs and is even more present in the most challenging problems is the premature convergence. Premature convergence might be alleviated with the consideration of techniques to control the diversity in the population. For instance, in single-objective optimization (see Chapter 3), it has been shown that taking into account the diversity of the decision variables to properly balance exploration and exploitation along the run is favorable. Diversity promotion techniques might involve one or several components of an EA, such as variation stage and replacement phase (see Chapter 2). The explicit management of diversity favors the avoidance of premature convergence, which is a common drawback with long runs. Recently, some diversity management algorithms that consider the diversity, stopping criterion and elapsed generations simultaneously have been devised. These explicit strategies allow a gradual loss of diversity as the execution progresses, and they have allowed the attainment of high-quality solutions

Most state-of-the-art MOEAs consider the diversity of the objective space explicitly, which is mandatory since in multi-objective optimization, obtaining a well-spread set of solutions is one of the aims of the optimization. Nevertheless, the diversity of decision variable space is usually not considered at all because it is not evaluated when comparing the approximations. However, it has been shown that the most popular MOEAs might quickly converge to a subspace of the decision variable space (see Chapter 2.2.4). Therefore, in some circumstances (e.g. for long runs), most MOEAs allocate most computational resources to explore better a subset of variables that usually represent the sparsity in the objective space. Thus, the remaining variables are not properly analyzed, resulting in a poor convergence to the Pareto front in some cases.

In light of the disadvantages that the loss of diversity in decision variable space might causes to MOEAs with preference information, this chapter presents a replacement-phase operator that explicitly manages the amount of diversity maintained along the run. In this way the diversity is explicitly managed by a replacement operator where the stopping criterion and the number of evaluations (or elapsed time) are used to manage the amount of diversity that can be loss in the decision variable space. This mechanism is introduced with minor modifications and without loss of generality in two MOEAs that consider two different indicators: the Hypervolume (HV) (Beume et al., 2007) and the R2 (Trautmann et al., 2013). Particularly, this approach follows the same principles that in Chapter 5 and Chapter 6, which gives more importance to the diversity of decision variable space in the initial stages, and it gradually grants more importance to the diversity of objective function space as the evolution progresses.

Two quite popular schemes have been selected to validate our proposal. This validation was performed using several well-known test problems and two well-known indicator-based MOEAs. This chapter clearly shows the important benefits of taking into account the diversity of the decision variable space for MCDM. In particular, the advantages become more evident in the most complex problems. It is also important to clarify that this work is oriented to show that managing explicitly the amount of diversity maintained in the decision variable space is useful to increase the quality of MOEAs regarding the preference information imposed by the DM.

The rest of this chapter is organized as follows. The proposal is detailed in Section 7.2. Section 7.3 is devoted to the experimental validation of the proposal. Finally, a summary and some conclusions are given in Section 7.4. The proper definition of the kind of multi-objective problems addressed in this chapter can be found in the Appendix B.

7.2 Algorithmic Proposal

This chapter proposes an indicator-based framework that process individuals in a steady-state manner. This framework integrates a replacement scheme that properly manages the diversity in the decision variable space. This replacement scheme adopts a similar principle as the one proposed in Chapter 3 where the stopping criterion and elapsed generations are considered with the aim of gradually moving from exploration to exploitation during the search process. Although this framework is mainly designed and tested for steady-state MOEAs it might be also applied to generational algorithms with minor modifications.

The main procedure follows the same outline than the most popular indicator-based MOEAs. Algorithm 16 shows the pseudo-code of our proposal. This procedure is quite standard and applies the usual stages appeared in an EA, which are *Mating selection*, *Variation*, and *Survivor selection*. First, in mating selection, two parents are chosen using binary tournament selection based on dominance ranking with ties broken randomly. Then, given the previously selected individuals, a new individual is created in the variation stage with the chosen operators. In this case, the Simulated Binary Crossover (SBX) and polynomial-based mutation operators are taken into account (Deb et al., 1995, 1996). Finally, the survivor selection stage is applied. Regarding steady-state MOEAs this survivor selection introduces a *reduce* operation in which an updating criterion is considered to discard one individual. This last process preserves μ individuals as the population of the subsequent iteration. For instance, in SMS-EMOA and R2-EMOA, an individual is discarded from the front with the worst rank. Whenever the worst front (F_v) comprises more than one individual, the individual with the worst contribution is eliminated. The contribution of each individual can be computed according to Eqn. (7.1), where $\vec{s}$ is any individual that belongs to the last front (F_v) and $I(.)$ is an indicator function that accomplishes the requirement of the DM. Note that any indicator might be integrated into this framework. In particular, this chapter analyzes the hypervolume-indicator (HV) and the R2-indicator.

$$\Delta I(\vec{s}, F_v) = I(F_v) - I(F_v \setminus \vec{s}) \tag{7.1}$$

The main contribution of our proposal is the integration of a diversity-aware replacement scheme. Note that in this context the stopping criterion is taken as the total number of evaluations.

7.2.1 Survivor Selection

Most of the first EAs were based in generational approaches where at each generation the offspring replace the previous populations regardless of the fitness (Eshelman, 1991; Segura et al., 2015c). Recent schemes integrate a selection stage known as a *replacement strategy*. This replacement selects which individuals survive to the next generation (Eiben et al., 2003). Note that all the algorithms that are considered in this chapter are

Algorithm 16 Main procedure of the steady-state MOEA

1: **Initialization**: Generate an initial population P_0 with μ individuals.
2: **Evaluation**: Evaluate all individuals in the population.
3: Assign $t := 0$
4: **while** (not stopping criterion) **do**
5: **Mating selection**: Select two individuals $\vec{y_1}$ and $\vec{y_2}$ by performing binary tournament selection on P_t, based on the non-dominated ranks (ties are broken randomly).
6: **Variation**: Generate an offspring $\vec{z} \in \mathbb{R}^n$ with $\vec{y_1}$ and $\vec{y_2}$ by applying SBX and Polynomial-based mutation.
7: **Evaluation**: Evaluate offspring $\vec{z}$ and update the reference vector $\vec{r}$ and ideal vector $\vec{i}$.
8: **Survivor selection**: Generate P_{t+1} by applying the replacement scheme described in Algorithm 17, using P_t and $\vec{z}$ as input.
9: $t := t+1$
10: **end while**

known as *steady-state*, where just a single individual is generated in each generation of the algorithm (Alba and Cotta, 2006). The replacement strategy presented in this section promotes a gradual change from exploration to exploitation, which is shown in previous chapters, and has led to remarkable results. Particularly, this replacement approach is applied at each generation of the main procedure and it works in the following manner.

Initially, a new individual is merged with the previous population, resulting in a temporary multi-set of size $\mu + 1$. Then, μ survivors that will become the parents of the next generation are iteratively selected. The principle followed to select these survivors is based on the individuals that better fulfill the requirements of diversity in the decision variable space and quality in the objective space. This diversity in the decision space of each individual is calculated with the Distance to the Closest Survivor (DCS). Given an individual $\vec{z}$ and a multi-set $\mathbb{S}$ that contains the currently selected survivors, DCS$(\vec{z}, \mathbb{S})$ is calculated according to Eqn. (7.2).

$$DCS(\vec{z}, \mathbb{S}) = \min_{\vec{s} \in \mathbb{S}} Distance(\vec{z}, \vec{s}) \tag{7.2}$$

Although any distance might be used, the Normalized Euclidean distance is considered (Barrett, 2005). This distance has shown to be quite effective for continuous domains in single-objective optimization (see Chapter 3). The normalized Euclidean distance not only normalizes each variable but also normalizes the whole measure by dividing it by the number of variables, resulting in a unitary measurement. As a consequence it is less sensitive to the magnitude of variable bounds and to the number of variables (Ning et al., 2008). Therefore, the distance between any pair of individuals $\vec{a}, \vec{b} \in \Omega$ is calculated according to Eqn. (7.3).

$$Distance(\vec{a}, \vec{b}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{a_i - b_i}{u_i - l_i} \right)^2} \tag{7.3}$$

At the first iteration of the replacement scheme, the $\mathbb{S}$ multi-set is empty, so the DCS of each individual is infinity; therefore, the individual that better fulfills the quality indicator is chosen as survivor. Then, at each step an individual with sufficient contribution to diversity and the best quality in the objective space is chosen as survivor. Note that the degree of exploration can be promoted by maintaining a diverse population that can be preserved by selecting individuals with a larger DCS. In fact, abrupt loss of exploration can be avoided by forcing the selection of individuals with a DCS above a certain threshold value. Since our intention is to promote exploration, individuals with a DCS value below a threshold are penalized. Then, among the non-penalized individuals, an indicator function is used to select the additional survivor. At this point, it might arise that all

Algorithm 17 Replacement Strategy

1: Input: P_t (Population of current iteration), $\vec{z}$ (Offspring of current iteration), μ (Population Size) and ITV (Initial Threshold Value).
2: Output: P_{t+1}.
3: $R_t := P_t \cup \vec{z}$
4: $P_{t+1} := \emptyset$
5: $Penalized := \emptyset$
6: $\tau := ITV - ITV \times \frac{T_{Elapsed}}{0.5 \times T_{End}}$
7: **while** $\mid P_{t+1} \mid \leq \mu$ **do**
8: Compute DCS of individuals in R_t, using P_{t+1} as a reference set.
9: Move the individuals in R_t with $DCS < \tau$ to $Penalized$.
10: **if** R_t is empty **then**
11: Compute DCS of individuals in $Penalized$, using P_{t+1} as a reference set.
12: Move the individual in $Penalized$ with the largest DCS to R_t.
13: **end if**
14: Identify the first front (F) in $R_t \cup P_{t+1}$ with at least one individual $\vec{y} \in R_t$.
15: According the contribution of each individual (Eqn 7.1) select a new survivor from F and move it from R_t to P_{t+1}.
16: **end while**
17: return P_{t+1}

individuals are penalized (caused by a high threshold value). In these cases, the individual with the largest DCS is selected to survive. Note that the threshold value used to penalize is decreased as the time elapses.

The algorithm 17 formalizes the proposed replacement scheme. This replacement strategy requires as input the population of the current iteration (P_t), the new individual $\vec{z}$, the population size μ, and an Initial Threshold Value (ITV). The output of this algorithm are the μ individuals of the next iteration (P_{t+1}). First, the population of the previous generation (P_t) and the new individual ($\vec{z}$) are merged in R_t (line 3). At each iteration, the multi-set R_t contains the remaining non-penalized individuals that might be selected to survive. The population of survivors (P_{t+1}) and the multi-set that contains the penalized individuals are initially empty (lines 4 and 5). Then, the threshold value (τ) which is applied to manage the exploration degree is calculated (line 6). Note that the threshold value follows a linear model that requires an initial threshold value (ITV), elapsed time ($T_{Elapsed}$) and the stopping criterion (T_{End}). In this way, this threshold is linearly decreased until the 50% of the total time and after that moment this threshold is below 0, meaning that no penalties are applied. Since no penalties occur after the first half of the algorithm's run, it follows the same principle as the steady state MOEAs. Then, an iterative process that selects one individual each time is executed until the survivor set contains μ individuals (lines 7 - 16). The iterative process works as follows. First, the DCS value of each remaining non-penalized individual is calculated (line 8). Then, those individuals with a DCS value lower than τ are moved to the set of penalized individuals (line 9). If all remaining individuals are penalized (line 10), it means that the amount of exploration is lower than desired. Thus, the individual with the largest DCS value is retaken, i.e., moved to the set of non-penalized individuals (lines 11 and 12), and thus survives.

Finally, the objective function space is considered. In particular, candidate non-penalized individuals and current survivors are merged. Then, the well-known non-dominated sorting procedure proposed in Deb et al. (2002a) is executed on this set, stopping as soon as a front with at least one candidate individual is found, that is, with an individual of R_t (line 14). Then, taking the identified front as an input, the contribution of each individual (Eqn. 7.1) is used to select the next survivor, i.e. the individual that is not already a survivor, and it is moved from R_t to P_{t+1} (line 15). Note that the contribution of each individual to the objective space depends on the specifications given by the DM. The pseudo-codes presented in this chapter are designed for explanatory purposes, and their corresponding implementations are not necessarily straightforward. Also note that some indicators might be efficient calculated with two and three objectives.

7.3 Experimental Validation

This section is devoted to validate our framework which is integrated with two popular indicator functions. The results obtained in this section clearly confirm that controlling diversity in the decision variable space is a way to improve indicator-based MOEAs in terms of preference information. First, the technical specifications involving the benchmark problems and algorithms are described. Then, an extensive experimentation of four MOEAs with different indicator functions is carried out. Later, some preference information is integrated into the same MOEAs and tested in some problems. Finally, an analysis of the initial threshold value is performed.

Regarding the multi-objective optimization problems, some of the most popular benchmarks are taken into account. These problems are WFG (Huband et al., 2006), DTLZ (Deb et al., 2005), UF (Zhang et al., 2008), and IMB (Liu et al., 2016). Note that each one was configured as is suggested by theirs authors. Particularly, the WFG test problems were used with two and three objectives and they were configured with 24 parameters, 20 of them corresponding to distance parameters and 4 to position parameters. In the DTLZ test problems, the number of variables was set to $n = m + k - 1$, where $k = \{5, 10, 20\}$ for DTLZ1, DTLZ2 to DTLZ6 and DTLZ7, respectively. The UF and IMB test problems were configured with 30 and 10 variables, respectively[1]. The experimental validation includes two well-known state-of-the-art MOEAs and two algorithmic proposals that result from integrating HV and R2 into our framework. In particular, the state-of-the-art MOEAs comprises the *S-metric Selection Evolutionary Multi-objective Optimization Algorithm* (SMS-EMOA) [2], and the *R2-Indicator Based Evolutionary Multi-objective Optimization Algorithm* (R2-EMOA) [3]. Similarly, the two proposals are the *Variable Space Diversity based* SMS-EMOA (VSD-SMS-EMOA), and *Variable Space Diversity based* R2-EMOA (VSD-R2-EMOA). Given that all the algorithms considered are stochastic, each execution was repeated 35 times with different seeds in all the experiments. The performance of each algorithm is measured according to its indicator function. In particular, for SMS-EMOA and VSD-SMS-EMOA the hypervolume indicator (HV) is used. The reference point used to calculate the HV is chosen to be a vector whose values are sightly larger (ten percent) than the Nadir point, as suggested in Ishibuchi et al. (2017). The normalized HV is used to facilitate the interpretation of the results (Li et al., 2014), and the reported value is calculated as the ratio between the HV obtained and the maximum attainable HV. In this way, a value equal to one means a perfect approximation. Note that a value equal to one is not attainable because MOEAs yield a discrete approximation.

Similarly, for R2-EMOA and VSD-R2-EMOA the normalized R2 is used, this value is computed as the ratio between the minimum attainable R2 and the R2 obtained. In this case a value equal to one means a perfect approximation and it can only be obtained if the approximation coincides with the Pareto Front on the same direction than the weight vectors. Without loss of generality, the R2 indicator integrates the Achievement Scalarizing Function (ASF) (Wagner et al., 2013). Note that ASF requires a set of weight vectors that follows the preference information according to the DM. In this case the preference information was the same than the one proposed in Trautmann et al. (2013), where a method to generate the weight vectors according the DM is

[1]More information of all the test problems can be reviewed in the Appendix C.
[2]Source code taken from https://ls11-www.cs.tu-dortmund.de/rudolph/hypervolume/start
[3]Source code taken from http://inriadortmund.gforge.inria.fr/r2emoa/

Table 7.1 Statistical information of HV ratio results for problems with two objectives

	VSD-SMS-EMOA						SMS-EMOA					
Crossover Probability	↑	↓	↔	**Det.**	**Mean**	**Median**	↑	↓	↔	**Det.**	**Mean**	**Median**
0.2	16	9	4	0.016	0.920	0.920	9	16	4	2.067	0.850	0.860
0.4	16	10	3	0.045	0.922	0.921	10	16	3	1.662	0.854	0.867
0.6	15	10	4	0.062	0.923	0.926	10	15	4	2.012	0.854	0.872
0.8	17	10	2	0.070	0.921	0.921	10	17	2	1.569	0.858	0.872
1.0	14	10	5	0.112	0.915	0.915	10	14	5	1.928	0.852	0.871
Total	**78 (54%)**	**49 (34%)**	**18 (12%)**	**0.061**	**0.920**	**0.920**	**49 (34%)**	**78 (54%)**	**18 (12%)**	**1.848**	**0.853**	**0.868**

described[4]. Therefore, the distribution of weight vectors used in this work are equally distributed and consists of 501 and 496 weight vectors for two and three objectives, respectively.

In order to statistically compare the indicator values attained by each pair of algorithms, the guidelines proposed in del Amo and Pelta (2013); Derrac et al. (2011) are followed. Given a set of approaches and their corresponding results, first, the Kruskal-Wallis is used as an Omnibus test to detect if there are any significant differences. In cases where there are significant differences, pair-wise statistical test are used to detect them; specifically, the Mann-Whitney post-hoc test with Hommel's correction of p-values. Note that in both tests, a significance level of 5% was considered.

The common configuration in all experiments was as follows: the population size was set to 100, the stopping criterion was set to 5×10^6 function evaluations, and the genetic operators were Simulated Binary Crossover (SBX) and polynomial-based mutation Deb et al. (1995, 1996). The crossover and mutation distribution indexes were fixed to 2 and 50, respectively. The mutation probability was set to $1/n$. In order, to have a better understanding of each algorithm five parameterizations of crossover probabilities were tested ($\{0.2, 0.4, 0.6, 0.8, 1.0\}$). These configurations were executed with all the aforementioned benchmarks in each algorithm. Note that, for the specific parameterizations, the default values provided by the authors are maintained. In the case of VSD-SMS-EMOA and VSD-R2-EMOA, the initial threshold value (ITV) was set equal to 0.4, which is the recommended value (see Chapter 5). Recall that the implications of ITV on preference information are analyzed in detail at the end of this section.

7.3.1 Comparison Against State-of-the-art Indicator-based MOEAs

In this experiment, each proposal is compared with the state-of-the-art algorithms that do not incorporate explicit management of diversity. Several configurations are generated by altering the crossover probability. Long runs are considered since it is the kind of execution where diversity-based EAs have been more successful. Note that this section only reports a summary and the complete information can be consulted online[5].

Tables 7.1 and 7.2 show a summary of the HV ratio obtained for the benchmark functions with two objectives (29 test problems) and three objectives (23 test problems), respectively. In this summary, each row details the results attained for each configuration (i.e. crossover probability) with all the test problems. Regarding each indicator, these tables include statistical information to compare carefully the results attained by each algorithm. Statistical tests were generated through pair-wise comparisons of both algorithms in each problem. In this way, each row reports the number of times that each method won (column ↑), lost (column ↓)

[4]Note that the weight vectors taken in this work correspond to $\rho = 1$.

[5]The file with the result of each problem and each crossover probability can be reviewed at https://drive.google.com/file/d/1qVxOlkgszzq8GQK9t3vupcatO-PO_Pzy/view

Table 7.2 Statistical information of HV ratio results for problems with three objectives

	VSD-SMS-EMOA						SMS-EMOA					
Crossover Probability	↑	↓	↔	**Det.**	**Mean**	**Median**	↑	↓	↔	**Det.**	**Mean**	**Median**
0.2	11	8	4	0.022	0.935	0.936	8	11	4	1.345	0.878	0.877
0.4	11	10	2	0.030	0.932	0.935	10	11	2	1.140	0.884	0.884
0.6	11	8	4	0.034	0.931	0.934	8	11	4	1.189	0.880	0.887
0.8	12	7	4	0.034	0.929	0.933	7	12	4	1.032	0.886	0.887
1.0	11	7	5	0.043	0.921	0.922	7	11	5	0.816	0.882	0.885
Total	**56 (48%)**	**40 (35%)**	**19 (17%)**	**0.033**	**0.930**	**0.932**	**40 (35%)**	**56 (48%)**	**19 (17%)**	**1.104**	**0.882**	**0.884**

and tied (column ↔), as well as the Mean and Median. Additionally, for each problem, we calculated the sum of the differences between the mean HV ratio attained and the best mean HV ratio. Note that this difference is only taken into account if there is a statistical significance difference, i.e. ties are not considered. This value is shown in the Deterioration column. The last row shows the total results belonging to all the crossover probabilities together. The data confirm that although VSD-SMS-EMOA loses in some cases, the overall number of wins and losses favors VSD-SMS-EMOA. More importantly, the total deterioration is quite lower in the case of VSD-SMS-EMOA, confirming that when VSD-SMS-EMOA loses, the deterioration is not that large. Interestingly, VSD-SMS-EMOA added more wins independently in each configuration.

Table 7.1 shows the results attained in problems with two objectives on the methods that integrates the HV indicator. Among them, the best method is VSD-SMS-EMOA with 78 wins against SMS-EMOA with 49 wins. Thus, the benefits attained by VSD-SMS-EMOA are quite evident. Furthermore, the mean HV ratio achieved by VSD-SMS-EMOA in each configuration (crossover probability) is always much higher than that achieved by SMS-EMOA. In fact, the total mean HV ratio values (last row) are 0.920 and 0.853 for VSD-SMS-EMOA and SMS-EMOA, respectively. Furthermore, in the case of VSD-SMS-EMOA, the total mean values and the total median values are quite similar, indicating its stability as an optimizer, which is particularly beneficial in MCDM. When analyzing the data, it is clear that in the cases where VSD-SMS-EMOA loses (attains lower HV), the difference with respect to SMS-EMOA is not very large. For example, the level of deterioration attained by VSD-SMS-EMOA was never higher than 0.15, while SMS-EMOA always reached a higher level of deterioration. This means that even if VSD-MOEA loses in some cases, its deterioration is always small, exhibiting a much more stable behavior according the preference information.

Regarding HV values, Table 7.2 reports the same information for the three-objective case. Taking into account the total mean of all the test problems, VSD-SMS-EMOA again obtained a much larger total mean HV ratio than SMS-EMOA. Specifically, VSD-MOEA obtained a value equal to 0.930, whereas SMS-EMOA obtained a value equal to 0.884. In this case, the deterioration level obtained by VSD-SMS-EMOA was never greater than 0.05 while SMS-EMOA always reached a higher deterioration level. Regarding the percentage reported in the statistical tests, the percentage of ties increased from the two-objective case to the three-objective case with values of 12% and 17%, respectively. On the contrary, the percentage of wins of VSD-SMS-EMOA is clearly higher in the two-objective case compared to the three-objective case whose values were 54% and 48% each. These findings suggest that, for this set of problems, the three-objective case requires less levels of exploration than the problems of two objectives. In fact, SMS-EMOA attained a lower deterioration level for the three-objective case (1.104) compared with the two-objective case (1.848). Similarly, the deterioration level value of VSD-SMS-EMOA was reduced approximately by a half from the two-objective case to the three objective-case, i.e. from 0.061 to 0.033. However, the benefits obtained by the promotion of diversity in the three-objective case are remarkable.

Table 7.3 Statistical information of R2 ratio results for problems with two objectives

	VSD-R2-EMOA						R2-EMOA					
Crossover Probability	↑	↓	↔	**Det.**	**Mean**	**Median**	↑	↓	↔	**Det.**	**Mean**	**Median**
0.2	16	6	7	0.002	0.936	0.939	6	16	7	3.574	0.812	0.805
0.4	17	7	5	0.054	0.939	0.940	7	17	5	3.497	0.820	0.811
0.6	17	7	5	0.070	0.939	0.941	7	17	5	3.729	0.813	0.805
0.8	15	8	6	0.083	0.937	0.939	8	15	6	3.723	0.811	0.809
1.0	14	6	9	0.091	0.930	0.931	6	14	9	3.702	0.805	0.805
Total	**79** (55%)	**34** (23%)	**32** (22%)	**0.060**	**0.936**	**0.938**	**34** (23%)	**79** (55%)	**32** (22%)	**3.645**	**0.812**	**0.807**

Table 7.4 Statistical information of R2 ratio results for problems with three objectives

	VSD-R2-EMOA						R2-EMOA					
Crossover Probability	↑	↓	↔	**Det.**	**Mean**	**Median**	↑	↓	↔	**Det.**	**Mean**	**Median**
0.2	11	5	7	0.055	0.952	0.953	5	11	7	1.105	0.906	0.91
0.4	10	6	7	0.058	0.951	0.951	6	10	7	1.046	0.905	0.90
0.6	10	8	5	0.066	0.949	0.951	8	10	5	0.901	0.913	0.93
0.8	12	7	4	0.062	0.947	0.948	7	12	4	0.956	0.908	0.91
1.0	11	8	4	0.078	0.940	0.940	8	11	4	0.946	0.902	0.90
Total	**54** (47%)	**34** (30%)	**27** (23%)	**0.064**	**0.948**	**0.949**	**34** (30%)	**54** (47%)	**27** (23%)	**0.991**	**0.907**	**0.911**

Tables 7.3 and 7.4 show the same information regarding the R2-indicator and with the methods VSD-R2-EMOA and R2-EMOA. In this case, the percentage of ties is similar for two objectives and three objectives with values of 22% and 23%, respectively. Similarly to the indicator HV, the total mean increased with the number of objectives for both methods. Again, VSD-R2-EMOA obtained the highest total mean values in both the two-objective case and the three-objective case. Specifically, considering two objectives VSD-R2-EMOA reached a total mean value equal to 0.936 and R2-EMOA a value equal to 0.812. Similarly, taking into account three objectives, VSD-R2-EMOA reached a total mean value equal to 0.948 and R2-EMOA a value equal to 0.907. Regarding the total mean and deterioration, R2-EMOA attained better results considering three objectives than with two objectives. In fact, its percentage of wins also increased. R2-EMOA was clearly outperformed by VSD-R2-EMOA.

In general, the result of taking into account the diversity in the decision variable space with both indicators is quite evident. The results reported in these section are with the intention that the reader perceives a general overview of the four methods. Note that, there are some kind of problems in which the diversity-aware methods yields remarkable improvements. Such MOPs present some features that hinder the optimization process. In particular, diversity-aware methods excelled in problems with a strong non-separability (e.g., WFG8, WFG9, UF1-3, IMB7-9), high multi-modality (e.g., WFG9, UF6, UF7), irregular Pareto geometries (e.g., WFG2, UF6, UF7, UF9), complicated Pareto set shapes (e.g., UF6, UF8, UF10), and imbalance mapping (e.g. IMB2, IMB3, IMB5). On the contrary, diversity-aware methods might fail in problems with high irregular variable linkages (e.g. WFG6, IMB1 and IMB4) and high biased problems (e.g. WFG1). Note that for biased problems, small perturbations in the decision space provoke large displacements in the objective function space. Thus –depending in the configuration– the methods that promote the maintenance of diversity in the decision variable space might not contribute positively in biased problems. Note that this drawback might not be present for some

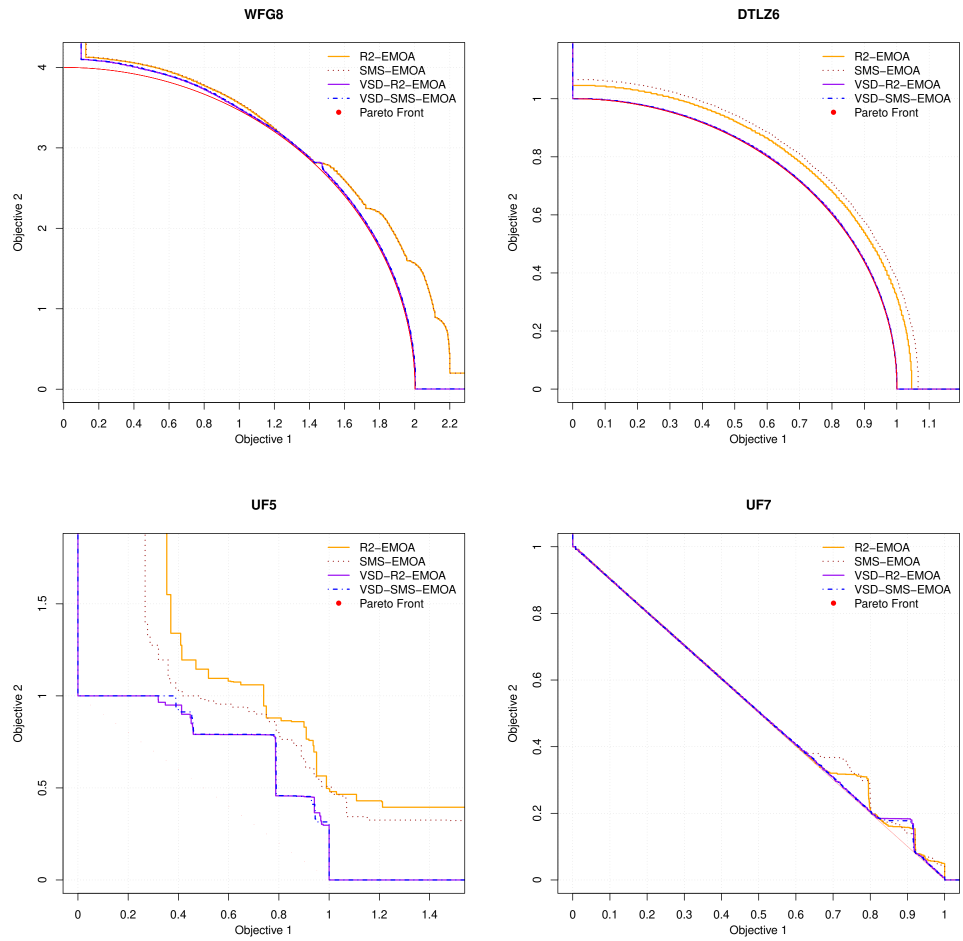

Fig. 7.1 50% attainment surfaces achieved for WFG8, UF3, UF5, and UF7

configurations; for instance, biased problems might be adequately approximated with either higher crossover probabilities [6] or with a higher probability of mutation.

In order to better illustrate the important impact contributed by diversity-aware methods on the attained results, the 50% attainment surfaces (Knowles, 2005) for WFG8, DTLZ6, UF5 and UF7 are shown in Figure 7.1. Note that for a fair comparison, the configuration reported on each method was the one that attained the best mean values according to their indicator functions. Each of the selected problems introduces different properties that might represent a different challenge for standard optimizers. In particular, WFG8 presents strong dependencies among all the variables, DTLZ6 is a uni-modal problem with a strong bias, UF5 is a multi-modal biased problem whose Pareto optimal front is discrete and consists of only 21 points, and UF7 presents irregular linkages between the variables. Among the four methods, the diversity-aware methods were the only ones that converged adequately to the Pareto front at least 50% of the runs on the WFG8 and DTLZ6 problems. Regarding the UF7 test problem, although all the methods converged for a long region of the Pareto front at least the 50% of the runs, the diversity-aware methods covered more the bottom right region of the Pareto

[6] This is confirmed on the WFG1 test problem, which can be consulted on the complete results, https://drive.google.com/file/d/1qVxOlkgszzq8GQK9t3vupcatO-PO_Pzy/view.

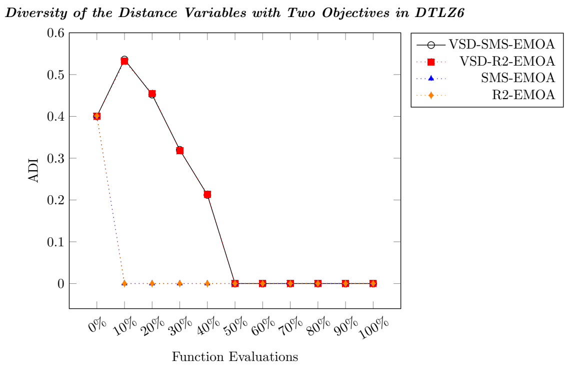

Fig. 7.2 Diversity of distance variables vs. elapsed function evaluations in the DTLZ6 test problem for two (left) an three (right) objectives. The reported results were taken from 35 runs.

front. Note that taking at least 50% of the runs, none of the methods converged to the Pareto front on the UF5, nevertheless the diversity-aware algorithms converged better than the state-of-the-art MOEAs.

In order to better understand the reasons behind the benefits of diversity-aware methods against state-of-the-art MOEAs the diversity along the run is analyzed. Note that variables can be classified into two types: distance variables and position variables (see Chapter 2.2.4). In some conditions MOEAs do not maintain a large enough diversity in the distance variables, so inspecting the diversity trend for these kinds of variables provides an useful insight into the dynamics of the population.

In order to show the behavior of the different schemes, we selected the previous four MOPs. In particular, in DTLZ6, UF5, and UF7 the first variable is a position variable, and the remaining are distance variables. In WFG8 the first four variables are defined as position parameters, however its strong dependencies affects the interaction between position and distance variables, i.e. a modification of the position variables also might affect the proximity to the Pareto front. Moreover, in UF5 the optimal regions are isolated in the variable space, meaning that a larger diversity is required. For each algorithm, the diversity is calculated as the mean Euclidean distance between individuals (ADI) in the population by considering only the distance variables. Figure 7.2 shows the evolution of ADI in the distance variables in DTLZ6 for two objectives (left side) and three objectives (right side). Note that for the state-of-the-art MOEAs, the distance variables quickly converged to a small region. Thus, on those cases the recombination operators loses its exploring power, as a result those methods were unable to better converge to the Pareto front as the evolution progresses. Conversely, in the diversity-aware methods, the decrease in ADI is quite linear to the midpoint of the execution achieving better approximations. Note that the evolution of ADI is quite similar considering two objectives and three objectives. Similarly, the UF problems (Figure 7.3) show the same behavior. Nevertheless, those problems do not converge completely since their Pareto front is mapped from different regions of the distance variables. As shown in UF5 (top left), explicitly promoting diversity is also beneficial for problems with disconnected optimal regions. Finally, regarding diversity-aware methods, the diversity maintained in WFG8 (bottom-middle) is quite different to the remaining problems. In this problem, the strong dependencies between the position variables and the distance variables are evident. As can be shown at the bottom of Figure 7.3 the promotion of diversity allowed to locate distinct distance variables that result in high-quality individuals. Note that in this kind of problems, it is expected that at the end of the execution, a level of diversity is maintained in both position and distance variables. However, the state-of-the-art MOEAs quickly converged to a small region, which results in a poor approximation of the Pareto front.

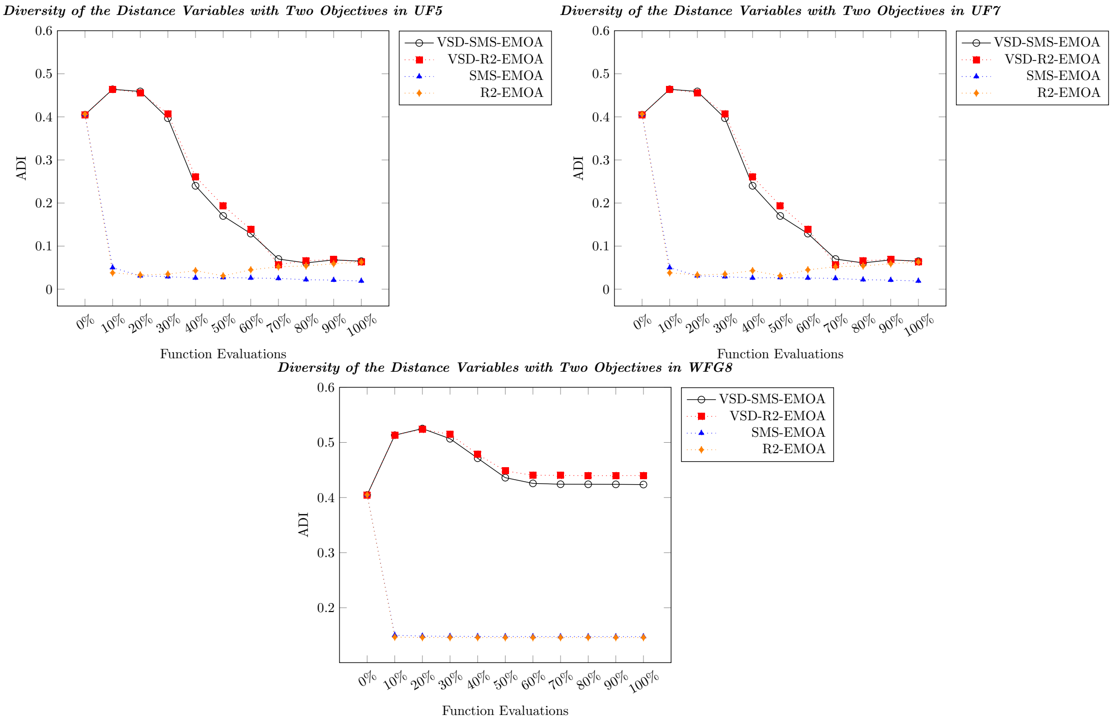

Fig. 7.3 Diversity of distance variables vs. elapsed function evaluations in the bi-objective problems UF5 (top-left), UF7 (top-right) and WFG8 (bottom-middle). The reported results were taken from 35 runs.

7.3.2 On the Integration of the Preference Information

Integrating DM's preference into quality indicators is possible and, in some cases, quite simple. However, expressing such preferences is not always intuitive for a DM. For the two-objective case, there are several visualization tools that might facilitate to DM the integration of preference information into indicator-based algorithms. In the previous section the attainment surfaces allowed to visually compare the results attained by the MOEAs in specific problems. This section illustrates the integration of preference information to the same problems (WFG8, DTLZ6, UF5 and UF7). At the end of this section another visual approach to compare such outcomes through the visualization of the empirical attainment functions (EAFs) differences, as well as numerical results are shown.

The integration of preference information into the HV and the R2-indicator can be done with some minor modifications (see Appendix B.1.8). For this case and based on the attainment surfaces shown in Figure 7.1, we define the preference information as follows. The favored regions consist of all the solutions that meet $\{\forall \vec{x} \mid f_1(\vec{x}) > 0.6 \wedge F(\vec{x}) \in PF\}$.

In order, to incorporate this preference information into the methods based on HV, this section considers the weighted hypervolume indicator defined in the Appendix Eqn. (B.17), where the weighted function is defined in Eqn (7.4).

$$w(\vec{z}) = \begin{cases} 1, & \text{if } f_1 > 0.6 \\ 0.001, & \text{otherwise} \end{cases} \tag{7.4}$$

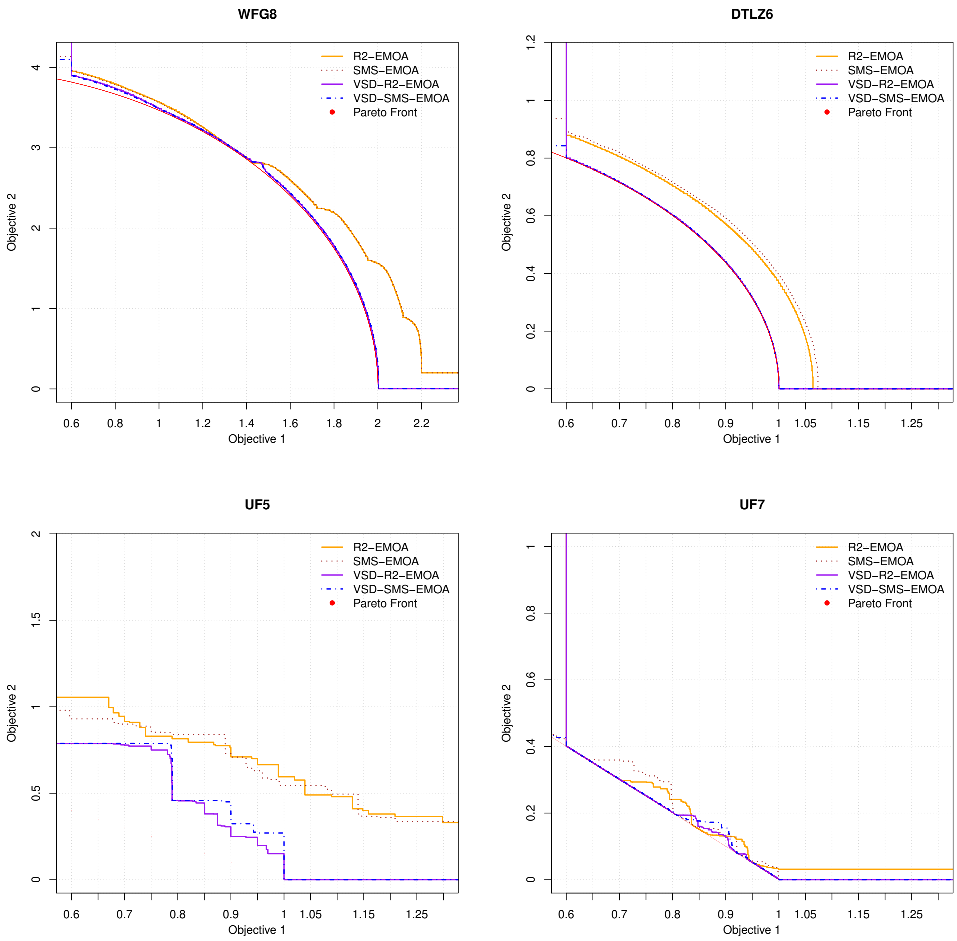

Fig. 7.4 50% attainment surfaces achieved for WFG8, DTLZ6, UF5 and UF7 with preference information.

Similarly, the inclusion of preference information in the R2 indicator is possible by the definition of a reference vector. Specifically, given a reference vector $\vec{r} = (0.6, 0)$, a set of solutions A, and a set of weight vectors $\vec{\lambda} = (\lambda_1, ..., \lambda_M) \in \Lambda$ the R2 indicator is defined as follows:

$$R2(A, \Lambda, \vec{r}) = \frac{1}{|\Lambda|} \sum_{\vec{\lambda} \in \Lambda} \min_{\vec{a} \in A} \left\{ \max_{j \in 1,..,M} \left\{ \frac{r_j - a_j}{\lambda_j} \right\} \right\} \tag{7.5}$$

Figure 7.4 shows the 50% attainment surfaces for WFG8, DTLZ6, UF5 and UF7 integrating the preference information. Regarding WFG8 and DTLZ7, VSD-SMS-EMOA and VSD-R2-EMOA achieved the preferred region of the Pareto front at least 50% of the runs. Moreover, all the methods covered better the preferred region of the Pareto front on the UF5 and UF7 problems than without preference information. In particular, VSD-SMS-EMOA and VSD-R2-EMOA converged much better than SMS-EMOA and R2-EMOA. In fact, the former two methods covered almost all the preferred region of the Pareto front in UF7.

Empirical Attainment Functions (EAF) describe the frequency that a specific region was attained by a multi-set of approximation fronts. Visualization of the difference in the EAF is an intuitive way to compare

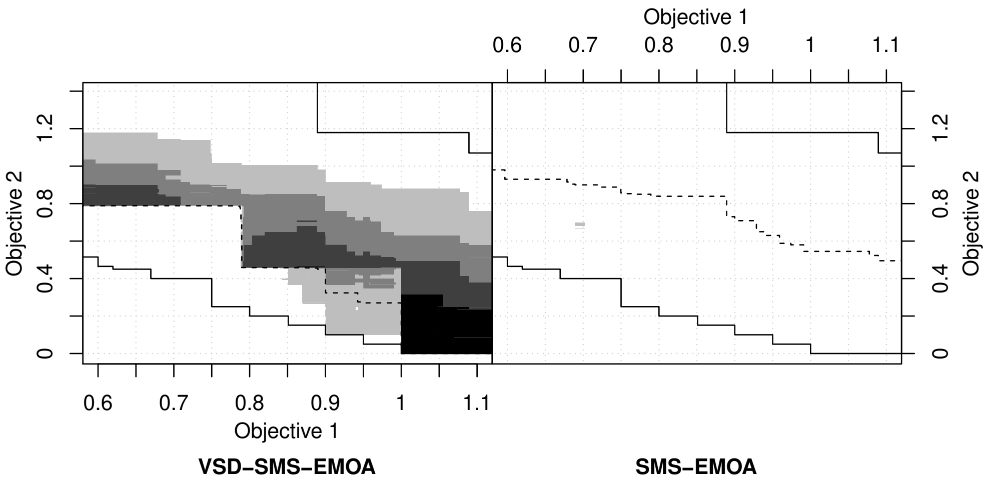

Fig. 7.5 Attainment difference plot of VSD-SMS-EMOA (left side) and SMS-EMOA (right side) on the problem UF5.

the performance of two-objective optimizers (Zitzler et al., 2008). Figures 7.5 and 7.6 show the Empirical Attainment Function (EAF) difference plot in UF5 for MOEAs based on HV and based on R2, respectively. The shaded areas represent the regions of the objective space where one algorithm achieved more than another algorithm. The more darker a region is, the more number of time an algorithm attained such region along the runs. In this visualization are also shown the best and worst regions among all runs represented by a solid line. Therefore, the best and worst results are indicated by the lower and upper lines, respectively. Finally, dashed lines represent the regions achieved by each algorithm at least 50% of the runs. In both figures, the advantages of diversity-aware methods are clear. Moreover, it can be noticed that VSD-SMS-EMOA has a tendency to attain some specific regions (nearby $f_1 = 0.6, 0.8, 1.0$) while VSD-R2-EMOA attained different regions of the objective space, this behavior is consistent with the properties of each indicator function.

Tables 7.5 and 7.6 show the statistics of the four methods and their respective indicators. In particular, these tables report *Worst*, *Best*, *Mean*, and *Median* of the 35 runs. Note that before computing HV values the individuals with $f_1 \leq 0.6$ are removed. These results confirm that for these problems and this preference information, the diversity-aware algorithm converged better into the region of preference than the state-of-the-art MOEAs. Specifically, for the DTLZ6 test problem, both diversity-aware methods always converged to the Pareto front, which can be confirmed by its worst approximation of 0.989 and 0.997 of VSD-SMS-EMOA and VSD-R2-EMOA, respectively. An important result in WFG8 is that the worst result of VSD-SMS-EMOA (0.848) is better than the best result attained by SMS-EMOA (0.844), which clearly shows a better convergence with these kind of problems (i.e. strong non-separability) and with integration of preference information. Finally, note that in terms of median, the most difficult problems (UF5, UF7) are better solved by VSD-SMS-EMOA and VSD-R2-EMOA. Regarding UF5, VSD-SMS-EMOA and SMS-EMOA achieved a median equal to 0.632 and 0.292, respectively. Similarly in UF7, VSD-R2-EMOA and R2-EMOA attained a median equal to 0.983 and 0.720, respectively.

Based on these findings the diversity-aware methods proved to improve the convergence and coverage of the Pareto front in indicator-based algorithms with the integration of preference information.

Table 7.5 Statistics of problems with preference information for HV methods. Individuals with $f_1 \leq 0.6$ are removed before computing HV.

	VSD-SMS-EMOA				SMS-EMOA			
Problem	**Worst**	**Best**	**Mean**	**Median**	**Worst**	**Best**	**Mean**	**Median**
WFG8	0.848	0.984	0.955	0.981	0.832	0.844	0.838	0.839
DTLZ6	0.989	0.996	0.995	0.996	0.528	0.928	0.762	0.771
UF5	0.371	0.857	0.632	0.632	0.000	0.748	0.297	0.292
UF7	0.897	0.981	0.946	0.949	0.448	0.985	0.912	0.966

Table 7.6 Statistics of problems with preference information on R2 methods

	VSD-R2-EMOA				R2-EMOA			
Problem	**Worst**	**Best**	**Mean**	**Median**	**Worst**	**Best**	**Mean**	**Median**
WFG8	0.353	0.987	0.889	0.977	0.739	0.742	0.741	0.741
DTLZ6	0.997	0.997	0.997	0.997	0.786	0.997	0.877	0.878
UF5	0.139	0.790	0.594	0.672	0.063	0.596	0.207	0.128
UF7	0.881	0.991	0.973	0.983	0.170	0.968	0.699	0.720

7.3.3 Analysis of the Initial Threshold Value

The downside of integrating our strategy that controls the diversity is that it incorporates additional parameters that must be set before each execution. In the case of diversity-aware MOEAs, the Initial Threshold Value (ITV) must be set. The higher this value is, the greater the exploration on the first stages. Note that in all previous experiments, $ITV = 0.4$ was used. As the reader has probability noticed on Figure 7.3, in average this level of diversity is the starting point for standard MOEAs, that is, on average the initial populations have similar levels of diversity. In addition, this value is suggested in Chapter 3 for a single-objective optimizer that was designed with similar principles. This section is devoted to analyze the performance of VSD-SMS-EMOA and VSD-R2-EMOA when different ITV values are used. Note that, since normalized distances are used, the maximum attainable difference is 1. Furthermore, when ITV is set to 0, no individual is penalized, so VSD-SMS-EMOA and VSD-R2-EMOA behave in a similar way to traditional MOEAs. Thus there is no explicit promotion of diversity in decision variable space. In order, to better understand the effect of ITV, several values of it with two crossover probabilities were tested. Specifically, the values $ITV = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and crossover probabilities $Px = \{0.2, 1.0\}$ were tested. Similar than in the previous experiment, all the test problems were considered for the two-objective and three-objective case as well the stopping criterion was set to 5×10^6 function evaluations.

Figure 7.7 shows the box-plots of the mean ratio with respect to each indicator function. Specifically, top figures describe the performance on the HV indicator and bottom figures describe the R2 indicator. Similarly, the figures on the left side belong to the two-objective cases, and the figures on the right side are for the three-objective cases. When ITV is set to 0, VSD-SMS-EMOA and VSD-R2-EMOA yielded similar results than the state-of-the-art MOEAs in both the two-objective and three-objective cases. Note that the performance with both indicator functions increases rapidly as the higher values of ITV are set to 0.4. Then, performance appears to decrease as ITV increases over 0.4. However, there is a wide range of values where performance is outstanding (e.g. $ITV \geq 0.2$). Regarding the crossover probabilities, both MOEAs behaves similar with both configurations, meaning that promoting diversity has a positive impact regardless of the probability. Nevertheless, in both indicator functions, lower crossover probabilities attain sightly better results in comparison with higher crossover probabilities.

This findings confirms that regarding the initial threshold value and the crossover probability the behavior of both VSD-SMS-EMOA and VSD-R2-EMOA is robust. Therefore, setting those parameters should not be complicated for practitioners.

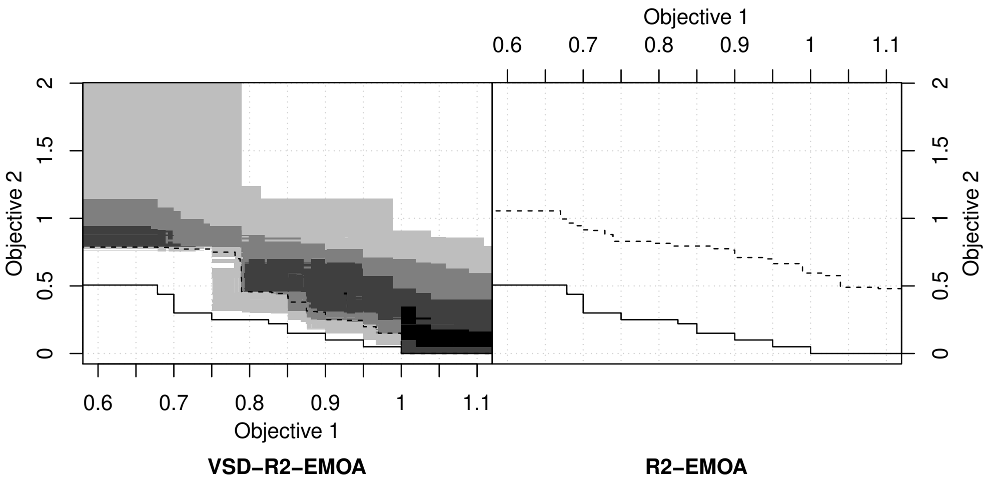

Fig. 7.6 Attainment difference plot of VSD-R2-EMOA (left side) and R2-EMOA (right side) on the problem UF5.

7.4 Summary

Evolutionary Multi-objective Algorithms are one of the most popular approaches for dealing with complicated black-box optimization problems. Their popularity and effectiveness has encouraged the design of a vast number of algorithms. Among them, indicator-based algorithms are adequate for their versatility to include decision makers' preferences.

This chapter proposes a general steady-state indicator-based algorithm. In particular, without loss of generality, this methodology is integrated with the hypervolume-indicator and R2-indicator generating the methods called VSD-SMS-EMOA and VSD-R2-EMOA, respectively. In both algorithms, the diversity takes into account both the decision variable space and the objective function space according to the preference information. These optimizers are important for the preservation of diverse populations at different levels and are adapted during the optimization process. Specifically, more importance is given to the diversity of the decision variable space at the initial stages, but as the time elapses, the objective space has more importance, therefore a gradual shift between exploration and intensification is promoted. This dynamical process is performed by using a penalty method that is integrated into the replacement strategy.

The experimental validation carried out showed a remarkable improvement in VSD-SMS-EMOA and VSD-R2-EMOA against the state-of-the-art MOEAs for both two-objective and three-objective problems. The benefits of our proposals are shown and analyzed with attainment surfaces and attainment difference plots. These results illustrate that VSD-SMS-EMOA and VSD-R2-EMOA are the MOEAs that better approximate these problems according to the preference information. Finally, the analysis of the initial threshold value, which is a new parameter, shows that the method is quite robust regardless of the crossover probability and the indicator-function, thus selecting a proper value is not a tedious task.

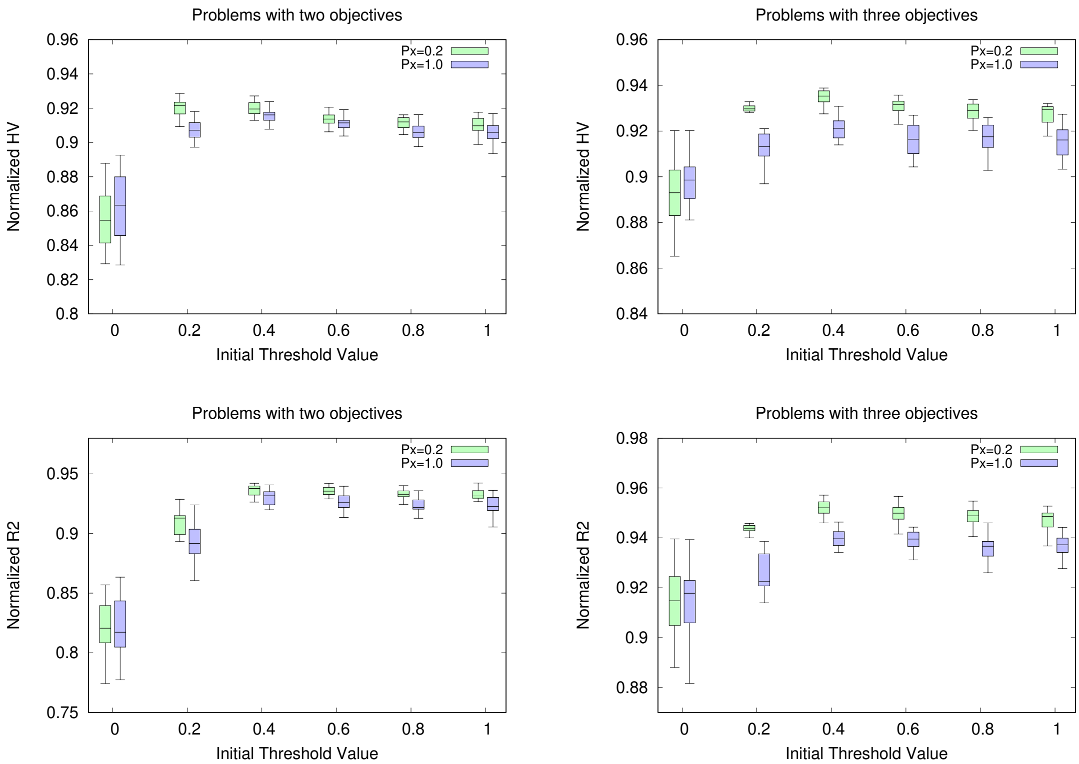

Fig. 7.7 Box-plots of the HV ratio and R2 ratio for 35 runs for the two-objective (left side) and three-objective (right side) problems, considering different initial threshold values

Chapter 8

Conclusions and Future Work

Population-based algorithms have been one of the most common approaches to address complex optimization problems, such as real-world problems. Their application is a complex task that requires the design of several components. However, their flexibility and effectiveness have allowed to attain some of the best known solutions for several classic optimization problems. Nevertheless, their performance is steered by several aspects such as how the diversity evolves through the optimization process.

In this thesis the way in which the evolution of diversity influences the performance of several well-known population-based strategies with some of the most popular benchmarks is analyzed. Particularly, this work generalizes a replacement method that explicitly maintains diversity in the decision variable space along the optimization process. As a consequence, this strategy has boosted the performance of several population-based algorithms either in single-objective or multi-objective optimization. In the remainder of this chapter, we conclude with the most remarkable results obtained in this work, and propose several lines for future research.

Differential Evolution has proven to be a popular strategy broadly applied for continuous optimization problems; however, one of its most important drawbacks is the premature convergence. This drawback can be alleviated by properly managing the diversity of solutions. Chapter 3 proposes two variants of DE, the *Differential Evolution with Enhanced Diversity Maintenance* (DEEDM and DEEDM-II). The main difference between them is that the former integrates an elite population. According to an exhaustive experimental validation, not including an elite population can be compensated by tuning the diversity parameters. Note that both variants follow the same principle of a novel replacement phase known as *Best Non Penalized*, which has already been successful with memetic algorithms in the combinatorial optimization field. The experimental validation have shown the important benefits obtained by both variants in long-term executions. This validation also considers both the classic DE variant and the more modern DE algorithms.

The benefits of properly managing diversity are also shown for real-world problems, which is the case of the *Menu Planning Problem* (MPP) discussed in Chapter 4. Specifically, this chapter addresses two formulations of the MPP; the minimization of the total cost and the minimization of the level of repetition among courses between days. Both formulations are optimized by a memetic algorithm that integrates a novel ad-hoc crossover operator, and a strategy to control the amount of diversity maintained in the replacement scheme. The experimental validation showed that incorporating a mechanism that explicitly controls the diversity into a memetic algorithms allows to improve in terms of the quality of the solutions attained at the end of the run. In the second formulation, some improvements for shortening the time required to attain feasible solutions from approximately one hour to ten seconds were devised. This formulation was incorporated into a desktop application, which is currently considered by nutritionists at the *Centro Integral de Atención a Niñas, Niños y Adolescentes*. Note that for both formulations, it is important to include a mechanism to manage the diversity and to consider tailored crossover operators and intensification phases.

Multi-objective Evolutionary Algorithms (MOEAs) are not exempted to the premature loss of diversity. Nevertheless, these algorithms implicitly maintain a minimum amount of diversity to better cover the Pareto front. In this thesis, three MOEAs that are dominance-based, indicator-based, and decomposition-based are proposed. In Chapter 5 the *Dominance-Based Multi-Objective Evolutionary Algorithm* (VSD-MOEA) is presented. VSD-MOEA takes into account the diversity of both the decision variable space and the objective function space. The main novelty is that the importance given to different diversities is adapted during the optimization process. In particular, in VSD-MOEA more importance is given to the diversity of the decision variable space in the initial stages, but as the evolution progresses, it assigns more importance to the diversity of the objective function space, meaning that a gradual shift between exploration and intensification is promoted. This is performed using a penalty method that is integrated into the replacement phase. VSD-MOEA also includes a novel density estimator based on IGD+ used to select from the non-penalized individuals. The experimental validation carried out shows a remarkable improvement in the VSD-MOEA compared to state-of-the-art MOEAs. Moreover, this proposal not only improves the state-of-the-art algorithms in long-term and medium-term executions, but it also offers competitive performance in short-term executions. Scalability analyzes show that as the number of objectives and decision variables increases, the implicit variable space maintained by state-of-the-art MOEAs also increases. Thus, for a large number of objectives and decision variables, explicitly considering the diversity of decision variable space is less helpful.

Chapter 6 presents the *Archived Variable Space Diversity MOEA based on Decomposition* (AVSD-MOEA/D), which follows the same design principle presented in Chapter 5. Specifically, a novel replacement operator based on the aforementioned design principle is applied to generate a decomposition-based MOEA which takes into account the diversity in both the variable and objective spaces. This is done by using a dynamic penalty method. Note that in the later phases, AVSD-MOEA/D behaves more similarly to traditional MOEAs. Additionally, taking into account recent advances and to ensure that AVSD-MOEA/D maintains high-quality solutions despite the penalty scheme, an external archive based on the R2-indicator is incorporated. The experimental validation carried out shows the remarkable improvement provided by AVSD-MOEA/D in comparison to state-of-the-art MOEAs with a diverse kind of test problems. Furthermore, the superiority of AVSD-MOEA/D is maintained with the benchmark of biased test problems. Finally, to better understand the reasons behind the huge superiority, some analyses involving the dynamics of the populations are provided. In comparison to state-of-the-art algorithms, AVSD-MOEA/D clearly slows down convergence.

Indicator-based Multi-objective Evolutionary Algorithms are also quite popular MOEAs that can be easily adapted to integrate the decision's maker preferences. Chapter 7 proposes a generalized steady-state indicator-based algorithm that integrates preference information. Without loss of generality, the hypervolume and R2 indicators are taken into account. Their respective proposals are *Variable Space Diversity based SMS-EMOA* (VSD-SMS-EMOA) and *Variable Space Diversity based R2-EMOEA* (VSD-R2-EMOA). Similarly to Chapter 5 and Chapter 6, in both algorithms, the diversity is taken into account simultaneously in the decision variable space and the objective function space with the integration of preference information. The experimental validation carried out shows a remarkable improvement in VSD-SMS-EMOA and VSD-R2-EMOA compared to state-of-the-art MOEAs. These results illustrate that explicitly promoting diversity in the decision variable space also favors problems that integrate preference information.

Unfortunately, the principle followed in the majority of chapters requires to set a new parameter, which is called the initial threshold value (a.k.a. initial distance factor). However, fixing values in a large range could be used to properly deal with a diverse set of problems, resulting in a quite robust strategy.

The main conclusion of this thesis is that most state-of-the-art solvers can be significantly improved by explicitly taking into account the diversity of the decision variables, and by reducing the importance given to this kind of diversity as the evolution progresses. As a consequence of this work, there is enough evidence to confirm the established hypothesis, where it is claimed that the performance of standard population-based algorithms can be improved with a proper balance between exploration and exploitation by considering the elapsed period and stopping criterion.

Several extensions might be explored to further improve these proposals. First, with the aim of facilitating the application of the replacement method, we would like to develop adaptive and/or self-adaptive schemes to avoid setting the initial threshold value. Second, with the aim of reducing the number of evaluations required to attain high-quality solutions, some of the strategies that are used in the field of optimization with expensive functions might be integrated. Third, including the diversity management put forth in this thesis in constrained multi-objective optimization, multi-modal multi-objective optimization, continuation improvement methods and many-objective optimization seems plausible. Fourth, incorporation of a local search engine mainly in the last stages of optimization might bring additional benefits. Finally, we should mention that integrating the design principles put forth in this thesis to cases where the stopping criterion is set by quality seems complex. Thus, developing strategies to allow for the integration of the design principles studied in this research in such settings seems a worthwhile area of research.

References

Aberg, J. (2009). An evaluation of a meal planning system: Ease of use and perceived usefulness.

Alba, E. (2005). *Parallel metaheuristics: a new class of algorithms*, volume 47. John Wiley & Sons.

Alba, E. and Cotta, C. (2006). Evolutionary algorithms. *Handbook of bioinspired algorithms and applications*, pages 3–19.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009). Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 555–562.

Auger, A. and Hansen, N. (2005). A restart cma evolution strategy with increasing population size. In *2005 IEEE congress on evolutionary computation*, volume 2, pages 1769–1776. IEEE.

Awad, N., Ali, M. Z., and Reynolds, R. G. (2015). A differential evolution algorithm with success-based parameter adaptation for cec2015 learning-based optimization. In *2015 IEEE congress on evolutionary computation (CEC)*, pages 1098–1105. IEEE.

Awad, N. H., Ali, M. Z., and Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 372–379. IEEE.

Awad, N. H., Ali, M. Z., Suganthan, P. N., and Reynolds, R. G. (2016). An ensemble sinusoidal parameter adaptation incorporated with l-shade for solving cec2014 benchmark problems. In *2016 IEEE congress on evolutionary computation (CEC)*, pages 2958–2965. IEEE.

Babalola, A. E., Ojokoh, B. A., and Odili, J. B. (2020). A review of population-based optimization algorithms. In *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, pages 1–7. IEEE.

Back, T. (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE World Congress on Computational Intelligence*, pages 57–62. IEEE.

Balintfy, J. L. (1964). Menu planning by computer. *Communications of the ACM*, 7(4):255–259.

Banchs, R. E. (1997). Simulated Annealing. *Research Progress Report*.

Barrett, P. (2005). Euclidean distance: Raw, normalised, and double-scaled coefficients. *Tech. Rep.*

Bechikh, S., Datta, R., and Gupta, A. (2016). *Recent advances in evolutionary multi-objective optimization*, volume 20. Springer.

Beheshti, Z. and Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, 5(1):1–35.

Berenguer, J. A. M. and Coello, C. A. C. (2015). Evolutionary many-objective optimization based on kuhn-munkres' algorithm. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 3–17. Springer.

Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, 8(1):10–15.

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

Beyer, H.-G. and Deb, K. (2001). On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on evolutionary computation*, 5(3):250–270.

Blickle, T. and Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394.

Bolufé-Röhler, A., Estévez-Velarde, S., Piad-Morffis, A., Chen, S., and Montgomery, J. (2013). Differential evolution with thresheld convergence. In *2013 IEEE Congress on Evolutionary Computation*, pages 40–47. IEEE.

Bosman, P. A. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE transactions on evolutionary computation*, 7(2):174–188.

Brest, J., Maučec, M. S., and Bošković, B. (2016). il-shade: Improved l-shade algorithm for single objective real-parameter optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1188–1195. IEEE.

Brest, J., Maučec, M. S., and Bošković, B. (2017). Single objective real-parameter optimization: Algorithm jso. In *2017 IEEE congress on evolutionary computation (CEC)*, pages 1311–1318. IEEE.

Brest, J., Zamuda, A., Boskovic, B., Maucec, M. S., and Zumer, V. (2008). High-dimensional real-parameter optimization using self-adaptive differential evolution algorithm with population size reduction. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2032–2039. IEEE.

Brest, J., Zumer, V., and Maucec, M. S. (2006). Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In *2006 IEEE international conference on evolutionary computation*, pages 215–222. IEEE.

Bui, L. T., Abbass, H. A., and Branke, J. (2005). Multiobjective optimization for dynamic environments. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2349–2356. IEEE.

Castillo, J. C. and Segura, C. (2019). Differential evolution with enhanced diversity maintenance. *Optimization Letters*, pages 1–20.

Castillo, J. C., Segura, C., Aguirre, A. H., Miranda, G., and León, C. (2017). A multi-objective decomposition-based evolutionary algorithm with enhanced variable space diversity control. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1565–1571.

Chakraborty, U. K. (2008). *Advances in differential evolution*, volume 143. Springer.

Chan, K. and Ray, T. (2005). An evolutionary algorithm to maintain diversity in the parametric and the objective space. In *International Conference on Computational Robotics and Autonomous Systems (CIRAS), Centre for Intelligent Control, National University of Singapore*.

Chen, G., Low, C. P., and Yang, Z. (2009). Preserving and exploiting genetic diversity in evolutionary programming algorithms. *IEEE Transactions on Evolutionary Computation*, 13(3):661–673.

Chen, S., Bolufé-Röhler, A., Montgomery, J., and Hendtlass, T. (2019). An analysis on the effect of selection on exploration in particle swarm optimization and differential evolution. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 3037–3044. IEEE.

Chen, S. and Montgomery, J. (2013). Particle swarm optimization with thresheld convergence. In *2013 IEEE congress on evolutionary computation*, pages 510–516. IEEE.

Chiang, T.-C. and Lai, Y.-P. (2011). Moea/d-ams: Improving moea/d by an adaptive mating selection mechanism. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1473–1480. IEEE.

Choon, N. H., Cheah, Y.-N., Goh, O. S., Choo, Y.-H., Basiron, H., and Kumar, Y. J. (2016). A review on automated menu planning approaches. *J. Comput. Sci.*, 12(12):582–596.

Coello, C. A. C. (2015). Multi-objective evolutionary algorithms in real-world applications: Some recent results and current challenges. In *Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences*, pages 3–18. Springer.

Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. (2007). Mop evolutionary algorithm approaches. *Evolutionary Algorithms for Solving Multi-Objective Problems: Second Edition*, pages 61–130.

Coello, C. A. C. and Sierra, M. R. (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican international conference on artificial intelligence*, pages 688–697. Springer.

Constantino, O. H. and Segura, C. (2021). A parallel memetic algorithm with explicit management of diversity for the job shop scheduling problem. *Applied Intelligence*, pages 1–13.

Črepinšek, M., Liu, S.-H., and Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, 45(3):1–33.

Cuate, O. and Schütze, O. (2019). Variation rate to maintain diversity in decision space within multi-objective evolutionary algorithms. *Mathematical and Computational Applications*, 24(3):82.

Czyzżak, P. and Jaszkiewicz, A. (1998). Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47.

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

Das, S. and Suganthan, P. N. (2010). Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems. *Jadavpur University, Nanyang Technological University, Kolkata*, pages 341–359.

Das, S. and Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31.

De Jong, K. (1975a). Analysis of the behavior of a class of genetic adaptive systems. *Technical Report No. 185, Department of Computer and Communication Sciences, University of Michigan*.

De Jong, K. (2016). Evolutionary computation: a unified approach. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 185–199.

De Jong, K. A. (1975b). *An analysis of the behavior of a class of genetic adaptive systems.* University of Michigan.

Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary computation*, 7(3):205–230.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience series in systems and optimization. Wiley.

Deb, K. and Abouhawwash, M. (2015). An optimality theory-based proximity measure for set-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(4):515–528.

Deb, K., Agrawal, R. B., et al. (1995). Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148.

Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. In *Proceedings of the third international conference on Genetic algorithms*, pages 42–50.

Deb, K., Goyal, M., et al. (1996). A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45.

Deb, K. and Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002a). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2002b). Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 825–830. IEEE.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer.

Deb, K. and Tiwari, S. (2008). Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3):1062–1087.

del Amo, I. G. and Pelta, D. A. (2013). Srcs: a technique for comparing multiple algorithms under several factors in dynamic optimization problems. In *Metaheuristics for Dynamic Optimization*, pages 61–77. Springer.

Derrac, J., García, S., Molina, D., and Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18.

Dhiman, G. and Kumar, V. (2019). Knrvea: A hybrid evolutionary algorithm based on knee points and reference vector adaptation strategies for many-objective optimization. *Applied Intelligence*, 49(7):2434–2460.

Diaz, J. E. and López-Ibáñez, M. (2021). Incorporating decision-maker's preferences into the automatic configuration of bi-objective optimisation algorithms. *European Journal of Operational Research*, 289(3):1209–1222.

Durillo, J. J., Nebro, A. J., Coello, C. A. C., García-Nieto, J., Luna, F., and Alba, E. (2010). A study of multiobjective metaheuristics when solving parameter scalable problems. *IEEE Transactions on Evolutionary Computation*, 14(4):618–635.

Eiben, Á. E., Hinterding, R., and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on evolutionary computation*, 3(2):124–141.

Eiben, A. E. and Schippers, C. A. (1998). On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50.

Eiben, A. E., Smith, J. E., et al. (2003). *Introduction to evolutionary computing*, volume 53. Springer.

Elsayed, S., Hamza, N., and Sarker, R. (2016). Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems. In *2016 IEEE congress on evolutionary computation (CEC)*, pages 2966–2973. IEEE.

Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2011a). Differential evolution with multiple strategies for solving cec2011 real-world numerical optimization problems. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1041–1048. IEEE.

Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2011b). Ga with a new multi-parent crossover for solving ieee-cec2011 competition problems. In *2011 IEEE congress of evolutionary computation (CEC)*, pages 1034–1040. IEEE.

Elsayed, S. M., Sarker, R. A., Essam, D. L., and Hamza, N. M. (2014). Testing united multi-operator evolutionary algorithms on the cec2014 real-parameter numerical optimization. In *2014 IEEE congress on evolutionary computation (CEC)*, pages 1650–1657. IEEE.

Erlich, I., Rueda, J. L., Wildenhues, S., and Shewarega, F. (2014). Evaluating the mean-variance mapping optimization on the ieee-cec 2014 test suite. In *2014 IEEE congress on evolutionary computation (CEC)*, pages 1625–1632. IEEE.

Eshelman, L. J. (1991). The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Foundations of genetic algorithms*, volume 1, pages 265–283. Elsevier.

Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., and Muhuri, P. K. (2021). Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review*, pages 1–80.

Fogel, D. B. (1994). An introduction to simulated evolutionary optimization. *IEEE transactions on neural networks*, 5(1):3–14.

Fonseca, C., Paquete, L., Stützle, T., López-Ibáñez, M., and Chiarandini, M. (2015). Package 'eaf'. *Manual*.

Fonseca, C. M. and Fleming, P. J. (1996). On the performance assessment and comparison of stochastic multiobjective optimizers. In *International Conference on Parallel Problem Solving from Nature*, pages 584–593. Springer.

Gabor, T. and Belzner, L. (2017). Genealogical distance as a diversity estimate in evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1572–1577.

Galán, S. F., Mengshoel, O. J., and Pinter, R. (2013). A novel mating approach for genetic algorithms. *Evolutionary computation*, 21(2):197–229.

Gazan, R., Brouzes, C. M., Vieux, F., Maillot, M., Lluch, A., and Darmon, N. (2018). Mathematical optimization to explore tomorrow's sustainable diets: a narrative review. *Advances in Nutrition*, 9(5):602–616.

Glover, F. and Kochenberger, G. (2005). Handbook of metaheuristics (international series in operations research and management science). *JOURNAL-OPERATIONAL RESEARCH SOCIETY*, 56(5):614–614.

Goldberg, D. E., Richardson, J., et al. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49. Hillsdale, NJ: Lawrence Erlbaum.

Gómez, R. H., Coello, C. A. C., and Alba, E. (2016). A parallel version of sms-emoa for many-objective optimization problems. In *International Conference on Parallel Problem Solving from Nature*, pages 568–577. Springer.

Gonzalez-Fernandez, Y. and Chen, S. (2015). Leaders and followers—a new metaheuristic to avoid the bias of accumulated information. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 776–783. IEEE.

Guerreiro, A. P., Fonseca, C. M., and Paquete, L. (2020). The hypervolume indicator: Problems and algorithms. *arXiv preprint arXiv:2005.00515*.

Guo, S.-M., Tsai, J. S.-H., Yang, C.-C., and Hsu, P.-H. (2015). A self-optimization approach for l-shade incorporated with eigenvector-based crossover and successful-parent-selecting framework on cec 2015 benchmark set. In *2015 IEEE congress on evolutionary computation (CEC)*, pages 1003–1010. IEEE.

Hajela, P., Lee, E., and Lin, C.-Y. (1993). Genetic algorithms in structural topology optimization. In *Topology design of structures*, pages 117–133. Springer.

Hansen, M. and Jaszkiewicz, A. (1998). Evaluating the quality of approximations of the non-dominated set. institute of mathematical modeling, technical university of denmark. Technical report, IMM Technical Report IMM-REP-1998-7.

Hansen, M. P. and Jaszkiewicz, A. (1994). *Evaluating the quality of approximations to the non-dominated set*. Citeseer.

Harik, G. R. et al. (1995). Finding multimodal solutions using restricted tournament selection. In *ICGA*, pages 24–31. Citeseer.

Harik, G. R., Lobo, F. G., et al. (1999). A parameter-less genetic algorithm. In *GECCO*, volume 99, pages 258–267.

Hernández Gómez, R. (2018). Parallel hyper-heuristics for multi-objective optimization= hiper-heurísticas paralelas para optimización multi-objetivo. *Thesis*.

Hernández Gómez, R. and Coello Coello, C. A. (2015). Improved metaheuristic based on the r2 indicator for many-objective optimization. In *Proceedings of the 2015 annual conference on genetic and evolutionary computation*, pages 679–686.

Herrera, F. and Lozano, M. (2003). Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. *Soft computing*, 7(8):545–562.

Herrera, F., Lozano, M., et al. (1996). Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorithms and Soft Computing*, 8(1996):95–125.

Herrera-Poyatos, A. and Herrera, F. (2017). Genetic and memetic algorithm with diversity equilibrium based on greedy diversification. *arXiv preprint arXiv:1702.03594*.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Huang, V., Qin, A. K., Deb, K., Zitzler, E., Suganthan, P. N., Liang, J., Preuss, M., and Huband, S. (2007). Problem definitions for performance assessment of multi-objective optimization algorithms. .

Huband, S., Barone, L., While, L., and Hingston, P. (2005). A scalable multi-objective test problem toolkit. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer.

Huband, S., Hingston, P., Barone, L., and While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506.

Hussain, K., Salleh, M. N. M., Cheng, S., and Shi, Y. (2019). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review*, 52(4):2191–2233.

Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2017). Reference point specification in hypervolume calculation for fair comparison and efficient search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 585–592.

Ishibuchi, H., Imada, R., Setoguchi, Y., and Nojima, Y. (2018). Reference point specification in inverted generational distance for triangular linear pareto front. *IEEE Transactions on Evolutionary Computation*, 22(6):961–975.

Ishibuchi, H., Masuda, H., Tanigaki, Y., and Nojima, Y. (2015). Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pages 110–125. Springer.

Ishibuchi, H., Yoshida, T., and Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE transactions on evolutionary computation*, 7(2):204–223.

Jerebic, J., Mernik, M., Liu, S.-H., Ravber, M., Baketarić, M., Mernik, L., and Črepinšek, M. (2021). A novel direct measure of exploration and exploitation based on attraction basins. *Expert Systems with Applications*, 167:114353.

Jiang, S., Zhang, J., Ong, Y.-S., Zhang, A. N., and Tan, P. S. (2014). A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics*, 45(10):2202–2213.

Kahraman, A. and Seven, H. A. (2005). Healthy daily meal planner. In *Proceedings of the 7th annual workshop on genetic and evolutionary computation*, pages 390–393.

Knowles, J. (2005). A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 552–557. IEEE.

Knowles, J. and Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 1, pages 711–716. IEEE.

Koumousis, V. K. and Katsaras, C. P. (2006). A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Transactions on Evolutionary Computation*, 10(1):19–28.

Kukkonen, S. and Deb, K. (2006a). A fast and effective method for pruning of non-dominated solutions in many-objective problems. In *Parallel Problem Solving from Nature-PPSN IX*, pages 553–562. Springer.

Kukkonen, S. and Deb, K. (2006b). Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1179–1186. IEEE.

Kukkonen, S. and Lampinen, J. (2009). Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In *2009 IEEE Congress on Evolutionary Computation*, pages 1943–1950. IEEE.

Kumar, A., Misra, R. K., and Singh, D. (2017). Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In *2017 IEEE congress on evolutionary computation (CEC)*, pages 1835–1842. IEEE.

Lacroix, B., Molina, D., and Herrera, F. (2013). Dynamically updated region based memetic algorithm for the 2013 cec special session and competition on real parameter single objective optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1945–1951. IEEE.

Lampinen, J., Zelinka, I., et al. (2000). On stagnation of the differential evolution algorithm. In *Proceedings of MENDEL*, pages 76–83.

Lewis, H. R. (1983). Computers and intractability. a guide to the theory of np-completeness.

Li, H. and Zhang, Q. (2008). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE transactions on evolutionary computation*, 13(2):284–302.

Li, H., Zhang, Q., and Deng, J. (2016a). Biased multiobjective optimization and decomposition algorithm. *IEEE transactions on cybernetics*, 47(1):52–66.

Li, K., Deb, K., Zhang, Q., and Kwong, S. (2014). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716.

Li, X. (2005). Efficient differential evolution using speciation for multimodal function optimization. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 873–880.

Li, X., Epitropakis, M. G., Deb, K., and Engelbrecht, A. (2016b). Seeking multiple solutions: An updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation*, 21(4):518–538.

Liang, J., Qu, B., Suganthan, P., and Chen, Q. (2014). Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization. *Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 29:625–640.

Liang, J. J., Qu, B. Y., and Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 635:490.

Liang, J. J. and Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer with local search. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 522–528. Ieee.

Liang, J. J., Yue, C., and Qu, B.-Y. (2016). Multimodal multi-objective optimization: A preliminary study. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2454–2461. IEEE.

Liao, T. and Stuetzle, T. (2013). Benchmark results for a simple hybrid algorithm on the cec 2013 benchmark set for real-parameter optimization. In *2013 IEEE Congress on Evolutionary Computation*, pages 1938–1944. IEEE.

Lin, L. and Gen, M. (2009). Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation. *Soft Computing*, 13(2):157–168.

Lin, M.-H., Tsai, J.-F., and Yu, C.-S. (2012). A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering*, 2012.

Liu, H.-L., Chen, L., Deb, K., and Goodman, E. D. (2016). Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms. *IEEE Transactions on Evolutionary Computation*, 21(3):408–425.

Liu, H.-L., Gu, F., and Zhang, Q. (2013). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE transactions on evolutionary computation*, 18(3):450–455.

Liu, M., Zheng, J., Wang, J., Liu, Y., and Lei, J. (2014). An adaptive diversity introduction method for dynamic evolutionary multiobjective optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3160–3167. IEEE.

Liu, S.-H., Mernik, M., and Bryant, B. R. (2009). To explore or to exploit: An entropy-driven approach for evolutionary algorithms. *International journal of knowledge-based and intelligent engineering systems*, 13(3-4):185–206.

Liu, Y., Ishibuchi, H., Yen, G. G., Nojima, Y., and Masuyama, N. (2019). Handling imbalance between convergence and diversity in the decision space in evolutionary multimodal multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 24(3):551–565.

Locatelli, M. and Vasile, M. (2015). (non) convergence results for the differential evolution method. *Optimization Letters*, 9(3):413–425.

Locatelli, M. and Wood, G. R. (2005). Objective function features providing barriers to rapid global optimization. *Journal of Global Optimization*, 31(4):549–565.

López-Camacho, E., García Godoy, M. J., Nebro, A. J., and Aldana-Montes, J. F. (2014). jmetalcpp: optimizing molecular docking problems with a c++ metaheuristic framework. *Bioinformatics*, 30(3):437–438.

López-López, I., Sosa-Gómez, G., Segura, C., Oliva, D., and Rojas, O. (2020). Metaheuristics in the optimization of cryptographic boolean functions. *Entropy*, 22(9):1052.

Loshchilov, I. (2013). Cma-es with restarts for solving cec 2013 benchmark problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 369–376. Ieee.

Lozano, M., Herrera, F., and Cano, J. R. (2008). Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information sciences*, 178(23):4421–4433.

Ma, X., Qi, Y., Li, L., Liu, F., Jiao, L., and Wu, J. (2014). Moea/d with uniform decomposition measurement for many-objective problems. *Soft Computing*, 18(12):2541–2564.

Mahdi, W., Medjahed, S. A., and Ouali, M. (2017). Performance Analysis of Simulated Annealing Cooling Schedules in the Context of Dense Image Matching. *Computación y Sistemas*, 21(3):493–501.

Mahfoud, S. W. (1995). *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign.

Mahfoud, S. W. et al. (1992). Crowding and preselection revisited. In *PPSN*, volume 2, pages 27–36. Citeseer.

Martin, W., Lienig, J., and Cohoon, J. P. (1997). C6. 3 island (migration) models: evolutionary algorithms based on punctuated equilibria. *B ack et al. BFM97], Seiten C*, 6:101–124.

Mengshoel, O. J., Galán, S. F., and De Dios, A. (2014). Adaptive generalized crowding for genetic algorithms. *Information Sciences*, 258:140–159.

Mengshoel, O. J. and Goldberg, D. E. (1999). Probabilistic crowding: Deterministic crowding with probabilistic replacement. *thesis*.

Miettinen, K. (2012). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

Mitchell, M. (1998). *An introduction to genetic algorithms*. MIT press.

Molina, D., LaTorre, A., and Herrera, F. (2018). An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions. *Cognitive Computation*, 10(4):517–544.

Molina, D., Moreno-García, F., and Herrera, F. (2017). Analysis among winners of different ieee cec competitions on real-parameters optimization: Is there always improvement? In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 805–812. IEEE.

Montgomery, J. (2009). Differential evolution: Difference vectors and movement in solution space. In *2009 IEEE Congress on Evolutionary Computation*, pages 2833–2840. IEEE.

Montgomery, J. and Chen, S. (2010). An analysis of the operation of differential evolution at high and low crossover rates. In *IEEE congress on evolutionary computation*, pages 1–8. IEEE.

Montgomery, J. and Chen, S. (2012). A simple strategy for maintaining diversity and reducing crowding in differential evolution. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE.

Morales-Castañeda, B., Zaldivar, D., Cuevas, E., Fausto, F., and Rodríguez, A. (2020). A better balance in metaheuristic algorithms: Does it exist? *Swarm and Evolutionary Computation*, 54:100671.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308–313.

Nesterov, Y. and Nemirovskii, A. (1994). *Interior-point polynomial algorithms in convex programming*. SIAM.

Ning, T., Tranquillo, J. V., Grare, A. C., and Saraf, A. (2008). Computing correlation integral with the euclidean distance normalized by the embedding dimension. In *2008 9th International Conference on Signal Processing*, pages 2708–2712. IEEE.

Nourani, Y. and Andresen, B. (1998). A Comparison of Simulated Annealing Cooling Strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373.

Ojha, M., Singh, K. P., Chakraborty, P., and Verma, S. (2019). A review of multi-objective optimisation and decision making using evolutionary algorithms. *International Journal of Bio-Inspired Computation*, 14(2):69–84.

Okabe, T., Jin, Y., Olhofer, M., and Sendhoff, B. (2004). On test functions for evolutionary multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 792–802. Springer.

Omidvar, M. N., Li, X., Mei, Y., and Yao, X. (2013). Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on evolutionary computation*, 18(3):378–393.

Osman, I. H. and Laporte, G. (1996). Metaheuristics: A bibliography.

Pandey, H. M., Chaudhary, A., and Mehrotra, D. (2014). A comparative review of approaches to prevent premature convergence in ga. *Applied Soft Computing*, 24:1047–1077.

Pescador-Rojas, M., Gómez, R. H., Montero, E., Rojas-Morales, N., Riff, M.-C., and Coello, C. A. C. (2017). An overview of weighted and unconstrained scalarizing functions. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 499–513. Springer.

Pétrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE international conference on evolutionary computation*, pages 798–803. IEEE.

Poláková, R., Tvrdík, J., and Bujok, P. (2016). L-shade with competing strategies applied to cec2015 learning-based test suite. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 4790–4796. IEEE.

Price, W. (1983). Global optimization by controlled random search. *Journal of Optimization Theory and Applications*, 40(3):333–348.

Purshouse, R. C. (2003). *On the evolutionary optimisation of many objectives*. PhD thesis, Citeseer.

Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). Moea/d with adaptive weight adjustment. *Evolutionary computation*, 22(2):231–264.

Qin, A. K., Huang, V. L., and Suganthan, P. N. (2008). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE transactions on Evolutionary Computation*, 13(2):398–417.

Qin, A. K. and Suganthan, P. N. (2005). Self-adaptive differential evolution algorithm for numerical optimization. In *2005 IEEE congress on evolutionary computation*, volume 2, pages 1785–1791. IEEE.

Reynoso-Meza, G., Sanchis, J., Blasco, X., and Herrero, J. M. (2011). Hybrid de algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In *2011 IEEE congress of evolutionary computation (CEC)*, pages 1551–1556. IEEE.

Rodríguez Villalobos, C. A. and Coello Coello, C. A. (2012). A new multi-objective evolutionary algorithm based on a performance assessment indicator. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 505–512.

Romero Ruiz, E. and Segura, C. (2018). Memetic algorithm with hungarian matching based crossover and diversity preservation. *Computación y Sistemas*, 22(2).

Rudolph, G., Naujoks, B., and Preuss, M. (2007). Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets. In *Evolutionary Multi-Criterion Optimization*, pages 36–50. Springer.

Rueda, J. L. and Erlich, I. (2013). Hybrid mean-variance mapping optimization for solving the ieee-cec 2013 competition problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 1664–1671. IEEE.

Rueda, J. L. and Erlich, I. (2015). Testing mvmo on learning-based real-parameter single objective benchmark optimization problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1025–1032. IEEE.

Rueda Torres, J. and Erlich, I. (2016). Solving the cec2016 real-parameter single objective optimization problems through mvmo-phm: Technical report. *contest*.

Sá, Â. A., Andrade, A. O., Soares, A. B., and Nasuto, S. J. (2008). Exploration vs. exploitation in differential evolution. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 57. Citeseer.

Salah, A., Hart, E., and Sim, K. (2016). Validating the grid diversity operator: an infusion technique for diversity maintenance in population-based optimisation algorithms. In *European Conference on the Applications of Evolutionary Computation*, pages 11–26. Springer.

Sallam, K. M., Sarker, R. A., Essam, D. L., and Elsayed, S. M. (2015). Neurodynamic differential evolution algorithm and solving cec2015 competition problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1033–1040. IEEE.

Sareni, B. and Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE transactions on Evolutionary Computation*, 2(3):97–106.

Schutze, O., Esquivel, X., Lara, A., and Coello, C. A. C. (2012). Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

Schütze, O. and Hernández, C. (2021). Archiving in evolutionary multi-objective optimization: A short overview. *Archiving Strategies for Evolutionary Multi-objective Optimization Algorithms*, pages 17–20.

Segura, C., Aguirre, A. H., Peña, S. I. V., and Rionda, S. B. (2017a). The importance of proper diversity management in evolutionary algorithms for combinatorial optimization. In *NEO 2015*, pages 121–148. Springer.

Segura, C., Botello Rionda, S., Hernández Aguirre, A., and Valdez Peña, S. I. (2015a). A novel diversity-based evolutionary algorithm for the traveling salesman problem. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 489–496.

Segura, C., Coello, C. A. C., and Hernández-Díaz, A. G. (2015b). Improving the vector generation strategy of differential evolution for large-scale optimization. *Information Sciences*, 323:106–129.

Segura, C., Coello, C. A. C., Miranda, G., and León, C. (2013a). Using multi-objective evolutionary algorithms for single-objective optimization. *4OR*, 11(3):201–228.

Segura, C., Coello, C. A. C., Segredo, E., and Aguirre, A. H. (2015c). A novel diversity-based replacement strategy for evolutionary algorithms. *IEEE transactions on cybernetics*, 46(12):3233–3246.

Segura, C., Coello, C. A. C., Segredo, E., Miranda, G., and León, C. (2013b). Improving the diversity preservation of multi-objective approaches used for single-objective optimization. In *2013 IEEE congress on evolutionary computation*, pages 3198–3205. IEEE.

Segura, C., Hernández-Aguirre, A., Luna, F., and Alba, E. (2016a). Improving diversity in evolutionary algorithms: New best solutions for frequency assignment. *IEEE transactions on evolutionary computation*, 21(4):539–553.

Segura, C., Miranda, G., Segredo, E., and Chacón, J. (2019). A novel memetic algorithm with explicit control of diversity for the menu planning problem. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2191–2198. IEEE.

Segura, C., Peña, S. I. V., Rionda, S. B., and Aguirre, A. H. (2016b). The importance of diversity in the application of evolutionary algorithms to the sudoku problem. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 919–926. IEEE.

Segura, C., Segredo, E., and Miranda, G. (2017b). The importance of the individual encoding in memetic algorithms with diversity control applied to large sudoku puzzles. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2152–2160. IEEE.

Shang, K., Ishibuchi, H., He, L., and Pang, L. M. (2020). A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20.

Shi, R., Lin, W., Lin, Q., Zhu, Z., and Chen, J. (2019). Multimodal Multi-objective Optimization Using a Density-Based One-By-One Update Strategy. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 295–301. IEEE.

Shir, O. M., Preuss, M., Naujoks, B., and Emmerich, M. (2009). Enhancing decision space diversity in evolutionary multiobjective algorithms. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 95–109. Springer.

Singh, G. and Deb, K. (2006). Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1305–1312.

Squillero, G. and Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, 329:782–799.

Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248.

Steyn, N. P., Nel, J. H., Nantel, G., Kennedy, G., and Labadarios, D. (2006). Food variety and dietary diversity scores in children: are they good indicators of dietary adequacy? *Public health nutrition*, 9(5):644–650.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., and Tiwari, S. (2005). Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL report*, 2005005(2005):2005.

Sun, Y., Yen, G. G., and Yi, Z. (2018). Igd indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 23(2):173–187.

Takahama, T. and Sakai, S. (2006). Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites. In *2006 IEEE international conference on evolutionary computation*, pages 1–8. IEEE.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

Talke, I. S. et al. (2012). *Number theoretic methods in designing experiments*. PhD thesis, Montana State University-Bozeman, College of Letters & Science.

Tamayo-Vera, D., Bolufé-Röhler, A., and Chen, S. (2016). Estimation multivariate normal algorithm with thresheld convergence. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3425–3432. IEEE.

Tan, K. C., Khor, E. F., and Lee, T. H. (2006). *Multiobjective evolutionary algorithms and applications*. Springer Science & Business Media.

Tanabe, R. and Fukunaga, A. (2013). Evaluating the performance of shade on cec 2013 benchmark problems. In *2013 IEEE Congress on evolutionary computation*, pages 1952–1959. IEEE.

Tanabe, R. and Fukunaga, A. S. (2014). Improving the search performance of shade using linear population size reduction. In *2014 IEEE congress on evolutionary computation (CEC)*, pages 1658–1665. IEEE.

Tanabe, R. and Ishibuchi, H. (2019). A review of evolutionary multimodal multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 24(1):193–200.

Tang, K., Yáo, X., Suganthan, P. N., MacNish, C., Chen, Y.-P., Chen, C.-M., and Yang, Z. (2007). Benchmark functions for the cec'2008 special session and competition on large scale global optimization. *Nature inspired computation and applications laboratory, USTC, China*, 24:1–18.

Tanweer, M. R., Suresh, S., and Sundararajan, N. (2015). Improved srpso algorithm for solving cec 2015 computationally expensive numerical optimization problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1943–1949. IEEE.

Thierens, D. (1998). *Selection schemes, elitist recombination, and selection intensity*, volume 1998. Utrecht University: Information and Computing Sciences.

Tian, Y., Xiang, X., Zhang, X., Cheng, R., and Jin, Y. (2018). Sampling reference points on the pareto fronts of benchmark multi-objective optimization problems. In *2018 IEEE congress on evolutionary computation (CEC)*, pages 1–6. IEEE.

Toffolo, A. and Benini, E. (2003). Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary computation*, 11(2):151–167.

Trautmann, H., Wagner, T., and Brockhoff, D. (2013). R2-emoa: Focused multiobjective search using r2-indicator-based selection. In *International Conference on Learning and Intelligent Optimization*, pages 70–74. Springer.

Trivedi, A., Srinivasan, D., Sanyal, K., and Ghosh, A. (2016). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462.

Tseng, L.-Y. and Chen, C. (2008). Multiple trajectory search for large scale global optimization. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3052–3059. IEEE.

Tsutsui, S., Ghosh, A., Corne, D., and Fujimoto, Y. (1997). A real coded genetic algorithm with an explorer and an exploiter populations. In *ICGA*, pages 238–245. Citeseer.

Ulrich, T., Bader, J., and Zitzler, E. (2010). Integrating decision space diversity into hypervolume-based multi-objective search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 455–462.

Ursem, R. K. (2002). Diversity-guided evolutionary algorithms. In *International Conference on Parallel Problem Solving from Nature*, pages 462–471. Springer.

Usta, I. and Kantar, Y. M. (2011). Mean-variance-skewness-entropy measures: A multi-objective approach for portfolio selection. *Entropy*, 13(1):117–133.

Van Veldhuizen, D. A. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Institute of Technology.

Van Veldhuizen, D. A. and Lamont, G. B. (1998). Multiobjective evolutionary algorithm research: A history and analysis. Technical report, Citeseer.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research*, 40(1):475–489.

Voß, S., Martello, S., Osman, I. H., and Roucairol, C. (2012). *Meta-heuristics: Advances and trends in local search paradigms for optimization*. Springer Science & Business Media.

Wagner, T., Trautmann, H., and Brockhoff, D. (2013). Preference articulation by means of the r2 indicator. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 81–95. Springer.

Wang, Y. and Li, B. (2008). A restart univariate estimation of distribution algorithm: sampling under mixed gaussian and lévy probability distribution. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3917–3924. IEEE.

Wang, Z., Zhang, Q., Zhou, A., Gong, M., and Jiao, L. (2015). Adaptive replacement strategies for moea/d. *IEEE transactions on cybernetics*, 46(2):474–486.

Wei, Z., Gao, W., Li, G., and Zhang, Q. (2021). A penalty-based differential evolution for multimodal optimization. *IEEE Transactions on Cybernetics*.

Wineberg, M. and Chen, J. (2004). The shifting balance genetic algorithm as more than just another island model ga. In *Genetic and Evolutionary Computation Conference*, pages 318–329. Springer.

Wu, G., Mallipeddi, R., and Suganthan, P. (2016). Problem definitions and evaluation criteria for the cec 2017 competition and special session on constrained single objective real-parameter optimization.

Wu, G., Mallipeddi, R., and Suganthan, P. N. (2017). Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization. *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*.

Xu, J. and Zhang, J. (2014). Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis. In *Proceedings of the 33rd Chinese control conference*, pages 8633–8638. IEEE.

Yang, M., Li, C., Cai, Z., and Guan, J. (2014). Differential evolution with auto-enhanced population diversity. *IEEE transactions on cybernetics*, 45(2):302–315.

Yang, M., Li, C., Cai, Z., and Guan, J. (2015). Differential evolution with auto-enhanced population diversity. *IEEE transactions on cybernetics*, 45(2):302–315.

Yang, Q., Chen, W.-N., Li, Y., Chen, C. P., Xu, X.-M., and Zhang, J. (2016). Multimodal estimation of distribution algorithms. *IEEE transactions on cybernetics*, 47(3):636–650.

Yin, X. and Germay, N. (1993). A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Artificial neural nets and genetic algorithms*, pages 450–457. Springer.

Yu, W.-J., Shen, M., Chen, W.-N., Zhan, Z.-H., Gong, Y.-J., Lin, Y., Liu, O., and Zhang, J. (2013). Differential evolution with two-level parameter adaptation. *IEEE Transactions on Cybernetics*, 44(7):1080–1099.

Zadorojniy, A., Masin, M., Greenberg, L., Shir, O. M., and Zeidner, L. (2012). Algorithms for Finding Maximum Diversity of Design Variables in Multi-Objective optimization. *Procedia Computer Science*, 8:171–176.

Zaharie, D. (2003). Control of population diversity and adaptation in differential evolution algorithms. In *Proc. of MENDEL*, volume 9, pages 41–46.

Zhang, J. and Sanderson, A. C. (2009). Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958.

Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.

Zhang, Q., Liu, W., and Li, H. (2009). The performance of a new version of moea/d on cec09 unconstrained mop test instances. In *2009 IEEE congress on evolutionary computation*, pages 203–208. IEEE.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S., et al. (2008). Multiobjective optimization test instances for the cec 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, 264:1–30.

Zhao, L., Sun, C.-j., Huang, X.-c., and Zhou, B.-x. (2016). Differential evolution with strategy of improved population diversity. In *2016 35th Chinese Control Conference (CCC)*, pages 2784–2787. IEEE.

Zhou, A., Zhang, Q., and Jin, Y. (2009). Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE transactions on evolutionary computation*, 13(5):1167–1189.

Zitzler, E., Brockhoff, D., and Thiele, L. (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195.

Zitzler, E., Knowles, J., and Thiele, L. (2008). Quality assessment of pareto set approximations. *Multiobjective optimization*, pages 373–404.

Zitzler, E. and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *International conference on parallel problem solving from nature*, pages 832–842. Springer.

Zitzler, E., Laumanns, M., and Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimization. *Metaheuristics for multiobjective optimisation*, pages 3–37.

Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301. Springer.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132.

Appendix A

Single-Objective Test Problems

A.1 Some Definitions

This section is devoted to the definition of the test functions proposed in contests of the Congress in Evolutionary Computation (CEC) (Liang et al., 2013; Wu et al., 2016) [1]. The kind of test functions proposed in these contests are black-box minimization problems formalized in Definition 3.2.1. In particular, the set of test functions belonging to years 2016 and 2017 are taken into account. The set of functions tested in both years integrate the definition of some "basic functions" assembling more challenging functions, which are termed as *hybrid functions* and *composition functions*. The lower bound and upper bound are the same for all the functions in both contests, which are $x_i^{(L)} = -100$ and $x_i^{(U)} = 100$, respectively. In other words, the search range (a.k.a. decision variable space) for each black-box optimization problem is $\Omega \in [-100, 100]^n$. Additionally, some functions require different metadata as are displacement vectors to shift the global optimum ($\vec{o}_i = [o_{i1}, ..., o_{in}]^T$), or rotation matrices to impose linkages among variables. The displacement of each test function $\vec{o}_i$ is randomly distributed in $[-80, 80]^n$, and each basic function has a different rotation matrix. Note that in the hybrid functions and composition functions the variables are divided into sub-components randomly. The rotation matrix for each sub-components are generated from standard normally distributed entries by Gram-Schmidt ortho-normalization with condition number (c) equal to 1 or 2.

A.1.1 Hybrid Functions

Considering that in real-word optimization problems, different sub-components of the variables may have different properties, in this set of hybrid functions, the variables are randomly divided into some sub-components. Each sub-component defines a different basic function.

$$F(\vec{x}) = \sum_{i=1}^{N} g_i(\mathbf{M_i z_i}) + F^*(\vec{x}) \tag{A.1}$$

where

- $F(\vec{x})$: hybrid function
- $g_i(\vec{x})$: i-th basic function used to construct the hybrid function
- N: number of basic functions
- $\vec{z} = [\vec{z}_1, ..., \vec{z}_N]$, a sub-component of variables.
- $\vec{z}_1 = [y_{s_1}, ..., y_{s_{n1}}]$, $\vec{z}_2 = [y_{s_{n1+1}}, ..., y_{s_{n1+n2}}], ..., \vec{z}_N = [y_{s_{sum_{i=1}^{N-1} n_i+1}}, ..., y_{s_n}]$
- $\vec{y} = \vec{x} - \vec{o}_i$, and S is a random permutation of the variables
- p_i: used to control the percentage of $g_i(\vec{x})$
- n_i: dimension for each basic function $\sum_{i=1}^{N} n_i = n$
- $n_1 = \lceil p_1 n \rceil, n_2 = \lceil p_2 n \rceil, ..., n_{N-1} = \lceil p_{N-1} n \rceil, n_N = n - \sum_{i=1}^{N-1} n_i$

[1] The test problems of CEC 2016 and CEC 2017 can be downloaded at https://github.com/P-N-Suganthan/CEC2014 and https://github.com/P-N-Suganthan/CEC2017-BoundContrained, respectively. Note that set problems of 2016 and 2014 are the same.

The hybrid functions have the same properties which are:

- Multi-modal or uni-modal, depending on the basic function
- Non-separability by sub-component
- Different properties for different variables sub-components

A.1.2 Composition Functions

The composition functions have additional interesting properties in comparison to hybrid functions. This set of functions are a mixture of sub-components, where each component might be either a basic function or a hybrid function. The main difference with hybrid functions is that each sub-component can be configured in a different manner. Although this set of functions might be more complicated mathematically, some optimizers might not have problems solving them. In composition functions, a bias has to be set for each component. Since this bias defines the minimum value, only one component should has a bias equal to zero. Also note that each component is multiplied by a weight (λ_i), which defines the height of its attraction basin.

The local optimum that has the smallest bias value is the global optimum. The composition functions merge the properties of the sub-functions and maintains continuity around the global/local optima. Note that the functions $F_i^{'} = F_i - F_i^*$ are considered in g_i. In this way, the function values of the global optima of g_i are equal to 0 for all composition functions. With hybrid functions as the basic functions, the composition function might have different properties for different sub-components. In order to test the algorithm's tendency to converge to the search center, a local optimum is set to the origin as a trap for each composition function included in this benchmark suite.

$$F(\vec{x}) = \sum_{i=1}^{N} \{\omega_i \times [\lambda_i g_i(\vec{x}) + bias_i]\} + F^* \tag{A.2}$$

where

- $F(\vec{x})$: composition function
- $g_i(\vec{x})$: i-th basic function used to construct the composition function.
- N: number of basic functions
- o_i: new shifted optimum position for each $g_i(\vec{x})$, define the position of the global and local optima's position
- $bias_i$: defines which optimum is global optimum
- σ_i: used to control the coverage range of each $g_i(\vec{x})$, a small σ_i gives a narrow range for that $g_i(\vec{x})$
- λ_i: used to control the height of each $g_i(\vec{x})$
- ω_i: weight value for each $g_i(\vec{x})$, calculated as

$$\omega_i = \left(\sum_{j=1}^{n} (x_j - o_{ij})^2\right)^{-1/2} exp\left(-\frac{\sum_{j=1}^{n} (x_j - o_{ij})^2}{2n\sigma_i}\right)$$

A.2 Problem Definitions for the Congress in Evolutionary Computation 2016

In recent years, several kinds of novel optimization algorithms have been proposed to solve real-parameter optimization problems, including the CEC 05 and CEC 2013 Special Session on Real-Parameter Optimization. This set of problems are proposed taking into account the feedback of the contest in CEC 2013. The benchmark proposed in this contest considers basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. Note that in this contest the surrogate models or meta-models are not allowed. This special session is devoted to the approaches, algorithms and techniques for solving real-parameter single-objective optimization without making use of the exact equation of the test functions. In addition, the set of problems taken into account in 2016 are exactly the same than in the contest of 2014. Particularly this benchmark comprises thirty test functions (see Table A.1), which are derived of fourteen basic functions, which are defined as follows.

Table A.1 Summary of the CEC'16 Test Functions

	No.	Functions	$F(\vec{x}^*)$
Unimodal Functions	1	Rotated High Conditioned Elliptic Function	100
	2	Rotated Bent Cigar Function	200
	3	Rotated Discus Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Ackley's Function	500
	6	Shifted and Rotated Weierstrass Function	600
	7	Shifted and Rotated Griewank's Function	700
	8	Shifted Rastringin's Function	800
	9	Shifted and Rotated Rastringin's Function	900
	10	Shifted Schwefel's Function	1000
	11	Shifted and Rotated Schwefel's Function	1100
	12	Shifted and Rotated Katsuura Function	1200
	13	Shifted and Rotated HappyCat Function	1300
	14	Shifted and Rotated HGBat Function	1400
	15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
	16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid Function	17	Hybrid Function 1 (N=3)	1700
	18	Hybrid Function 2 (N=3)	1800
	19	Hybrid Function 3 (N=4)	1900
	20	Hybrid Function 4 (N=4)	2000
	21	Hybrid Function 5 (N=5)	2100
	22	Hybrid Function 6 (N=5)	2200
Composition Functions	23	Composition Function 1 (N=5)	2300
	24	Composition Function 2 (N=3)	2400
	25	Composition Function 3 (N=3)	2500
	26	Composition Function 4 (N=5)	2600
	27	Composition Function 5 (N=5)	2700
	28	Composition Function 6 (N=5)	2800
	29	Composition Function 7 (N=3)	2900
	30	Composition Function 8 (N=3)	3000
Search Range $[-100, 100]^n$			

Definition of Basic Functions

Rotated High Conditioned Elliptic Function

$$f_1(\vec{x}) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$$

Bent Cigar Function

$$f_2(\vec{x}) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$$

Discus Function

$$f_3(\vec{x}) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$$

Rosenbrock's Function

$$f_4(\vec{x}) = \sum_{i=1}^{n-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$$

Ackley's Function

$$\begin{aligned} f_5(\vec{x}) = & -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) \\ & - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e \end{aligned}$$

Weierstrass Function

$$\begin{aligned} f_6(\vec{x}) = & \sum_{i=1}^{n}\left(\sum_{k=0}^{20} [0.5^k cos(2\pi 3^k (x_i + 0.5))]\right) \\ & - n\sum_{k=0}^{20} [0.5^k cos(\pi 3^k)] \end{aligned}$$

Griewank's Function

$$f_7(\vec{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Rastrigin's Function

$$f_8(\vec{x}) = \sum_{i=1}^{n} \left(x_i^2 - 10cos(2\pi x_i) + 10\right)$$

Modified Schwefel's Function

$$\begin{aligned} f_9(\vec{x}) = & 418.9829 \times n - \sum_{i=1}^{n} g(z_i), \\ z_i = & x_i + 4.209687462275036e+002 \end{aligned}$$

$$g(z_i) = \begin{cases} z_i sin(|z_i|^{1/2}) & |z_i| \leq 500 \\ (500 - mod(z_i, 200))sin\left(\sqrt{|500 - mod(z_i, 500)|}\right) - \frac{(z_i - 500)^2}{10000n} & z_i > 500 \\ (mod(|z_i|, 500) - 500)\, sin\left(\sqrt{|mod(z_i, 500) - 500|}\right) - \frac{(z_i + 500)^2}{10000n} & z_i < -500 \end{cases}$$

Katsuura Function

$$f_{10}(\vec{x}) = \frac{10}{n^2} \prod_{i=1}^{n} \left(1 + i \sum_{j=1}^{32} \frac{|2^j x_i - round(2^j x_i)|}{2^j}\right)^{\frac{10}{n^{1.2}}} - \frac{10}{n^2}$$

HappyCat Function

$$\begin{aligned} f_{11}(\vec{x}) = & \left|\sum_{i=1}^{n} x_i^2 - n\right|^{1/4} \\ & + \left(0.5\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i\right)/n + 0.5 \end{aligned}$$

HGBat Function

$$f_{12}(\vec{x}) = \left| \left(\sum_{i=1}^{n} x_i^2\right)^2 - \left(\sum_{i=1}^{n} x_i\right)^2 \right|^{1/2} + \left(0.5\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i\right)/n + 0.5$$

Expanded Griewank's plus Rosenbrock's Function

$$f_{13}(\vec{x}) = \left(\sum_{i=1}^{n-1} f_7(f_4(x_i, x_{i+1}))\right) + f_7(f_4(x_n, x_1))$$

Expanded Scaffer's F6 Function

$$f_{14}(\vec{x}) = \left(\sum_{i=1}^{n-1} g(x_i, x_{i+1})\right) + g(x_n, x_1)$$

$$g(a,b) = 0.5 + \frac{(sin^2(\sqrt{a^2+b^2}) - 0.5)}{(1+0.001(a^2+b^2))^2}$$

A.2.1 Definitions of the CEC 2016 Test suite

Rotated High Conditioned Elliptic Function
Test Problem 1

$$F_1(\vec{x}) = f_1(\mathbf{M}(\vec{x} - \vec{o}_1)) + F_1^*$$

- Non-separable
- Quadratic ill-conditioned

Rotated Bent Cigar Function
Test Problem 2

$$F_2(\vec{x}) = f_2(\mathbf{M}(\vec{x} - \vec{o}_2)) + F_2^*$$

- Non-separable
- Smooth but narrow ridge

Rotated High Conditioned Elliptic Function
Test Problem 3

$$F_3(\vec{x}) = f_3(\mathbf{M}(\vec{x} - \vec{o}_3)) + F_3^*$$

- Non-separable
- With one sensitive direction

Shifted and Rotated Rosenbrock's Function
Test Problem 4

$$F_4(\vec{x}) = f_4\left(\mathbf{M}\left(\frac{2.048(\vec{x} - \vec{o}_4)}{100}\right) + 1\right) + F_4^*$$

- Non-separable
- Having very narrow valley from local optimum to global optimum

Shifted and Rotated Ackley's Function
Test Problem 5

$$F_5(\vec{x}) = f_5(\mathbf{M}(\vec{x} - \vec{o}_5)) + F_8^*$$

- Non-separable

Shifted and Rotated Weierstrass Function
Test Problem 6

$$F_6(\vec{x}) = f_6\left(\mathbf{M}\left(\frac{0.5\,(\vec{x} - \vec{o}_6)}{100}\right)\right) + F_6^*$$

- Non-separable
- Continuous but differentiable only on a set of points

Shifted and Rotated Griewank's Function
Test Problem 7

$$F_7(\vec{x}) = f_7\left(\mathbf{M}\left(\frac{600(\vec{x}-\vec{o}_7)}{100}\right)\right) + F_7^*$$

- Rotated
- Non-separable

Shifted Rastrigin's Function
Test Problem 8

$$F_8(\vec{x}) = f_8\left(\frac{5.12(\vec{x}-\vec{o}_8)}{100}\right) + F_7^*$$

- Separable
- Local optima's number is huge

Shifted and Rotated Rastrigin's Function
Test Problem 9

$$F_9(\vec{x}) = f_8\left(\mathbf{M}\left(\frac{5.12(\vec{x}-\vec{o}_9)}{100}\right)\right) + F_9^*$$

- Non-separable
- Local optima's number is huge

Shifted 'Schwefel's Function
Test Problem 10

$$F_{10}(\vec{x}) = f_9\left(\frac{1000(\vec{x}-\vec{o}_{10})}{100}\right) + F_{10}^*$$

- Separable
- Local optima's number is huge and the second better local optimum is far from the global optimum

Shifted and Rotated Schwefel's Function
Test Problem 11

$$F_{11}(\vec{x}) = f_9\left(\mathbf{M}\left(\frac{1000(\vec{x}-\vec{o}_{11})}{100}\right)\right) + F_{11}^*$$

- Non-separable
- Local optima's number is huge and second better local optimum is far from the global optimum

Shifted and Rotated Katsuura Function
Test Problem 12

$$F_{12}(\vec{x}) = f_{10}\left(\mathbf{M}\left(\frac{5(\vec{x}-\vec{o}_{12})}{100}\right)\right) + F_{12}^*$$

- Non-separable
- Continuous everywhere yet differentiable nowhere

Shifted and Rotated HappyCat Function
Test Problem 13

$$F_{13}(\vec{x}) = f_{11}\left(\mathbf{M}\left(\frac{5(\vec{x}-\vec{o}_{13})}{100}\right)\right) + F_{13}^*$$

- Non-separable

Shifted and Rotated HGBat Function
Test Problem 14

$$F_{14}(\vec{x}) = f_{12}\left(\mathbf{M}\left(\frac{5(\vec{x}-\vec{o}_{14})}{100}\right)\right) + F_{14}^*$$

- Non-separable

Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function
Test Problem 15

$$F_{15}(\vec{x}) = f_{13}\left(\mathbf{M}\left(\frac{5(\vec{x}-\vec{o}_{15})}{100}\right) + 1\right) + F_{15}^*$$

- Non-separable

Shifted and Rotated Expanded Scaffer's F6 Function
Test Problem 16

$$F_{16}(\vec{x}) = f_{14}\left(\mathbf{M}(\vec{x}-\vec{o}_{16}) + 1\right) + F_{16}^*$$

- Non-separable

Hybrid Function 1
Test Problem 17

$N = 3$, $\vec{p} = [0.3, 0.3, 0.4]$

g_1: Modified Schwefel's Function f_9

g_2: Rastrigin's Function f_8

g_3: High Conditioned Elliptic Function f_1

Hybrid Function 2
Test Problem 18

$N = 3$, $\vec{p} = [0.3, 0.3, 0.4]$

g_1: Bent Cigar Function Function f_2

g_2: HGBat Function f_{12}

g_3: Rastrigin's Function f_8

Hybrid Function 3
Test Problem 19

$N = 4$, $\vec{p} = [0.2, 0.2, 0.3, 0.3]$

g_1: Griewank's Function f_7

g_2: Weierstrass Function f_6

g_3: Rosenbrock's Function f_4

g_4: Scaffer's F6 Function f_{14}

Hybrid Function 4
Test Problem 20

$N = 4, \quad \vec{p} = [0.2, 0.2, 0.3, 0.3]$
g_1: HGBat Function f_{12}
g_2: Discus Function f_3
g_3: Expanded Griewank's plus Rosenbrock's Function f_{13}
g_4: Rastrigin's Function f_8

Hybrid Function 5
Test Problem 21

$N = 5, \quad \vec{p} = [0.1, 0.2, 0.2, 0.2, 0.3]$
g_1: Scaffer's F6 Function f_{14}
g_2: HGBat Function f_{12}
g_3: Rosenbrock's Function f_4
g_4: Modified Schwefel's Function f_9
g_5: High Conditioned Elliptic Function f_1

Hybrid Function 6
Test Problem 22

$N = 5, \quad \vec{p} = [0.1, 0.2, 0.2, 0.2, 0.3]$
g_1: Katsuura Function f_{10}, g_2: HappyCat Function f_{11}
g_3: Expanded Griewank's plus Rosenbrock's Function f_{13}
g_4: Modified Schwefel's Function f_9, g_5: Ackley's Function f_5

Composition Function 1
Test Problem 23

$N = 5, \vec{\sigma} = [10, 20, 30, 40, 50]$
$\vec{\lambda} = [1, 1e-6, 1e-26, 1e-6, 1e-6]$
$\vec{bias} = [0, 100, 200, 300, 400]$
g_1 : Rotated Rosenbrock's Function F_4'
g_2 : High Conditioned Elliptic Function F_1'
g_3 : Rotated Bent Cigar Function F_2'
g_4 : Rotated Discus Function F_3'
g_5 : High Conditioned Elliptic Function F_1'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 2
Test Problem 24

$N = 3, \vec{\sigma} = [20, 20, 20]$
$\vec{\lambda} = [1, 1, 1]$
$\vec{bias} = [0, 100, 200]$
g_1 : Schwefel's Function F_{10}'
g_2 : Rotated Rastrigin's Function F_9'
g_3 : Rotated HGBat Function F_14'

- Multi-modal
- Non-separable
- Different properties around different local optima

Composition Function 3
Test Problem 25

$N = 3, \vec{\sigma} = [10, 30, 50]$
$\vec{\lambda} = [0.25, 1e-7]$
$\vec{bias} = [0, 100, 200]$
g_1 : Rotated Schwefel's Function F_{11}'
g_2 : Rotated Rastrigin's Function F_9'
g_3 : Rotated High Conditioned Elliptic Function F_1'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 4
Test Problem 26

$N = 5, \vec{\sigma} = [10, 10, 10, 10, 10]$
$\vec{\lambda} = [0.25, 1, 1e-7, 2.5, 10]$
$\vec{bias} = [0, 100, 200, 300, 400]$
g_1 : Rotated Schwefel's Function F_{11}'
g_2 : Rotated HappyCat Function F_{13}'
g_3 : Rotated High Conditioned Elliptic Function F_1'
g_4 : Rotated Weierstrass Function F_6'
g_5 : Rotated Griewank's Function F_7'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 5
Test Problem 27

$$N = 5, \vec{\sigma} = [10, 10, 10, 20, 20]$$
$$\vec{\lambda} = [10, 10, 2.5, 25, 1e-6]$$
$$\vec{bias} = [0, 100, 200, 300, 400]$$

g_1 : Rotated HGBat Function F'_{14}

g_2 : Rotated Rastrigin's Function F'_9

g_3 : Rotated Schwefel's Function F'_{11}

g_4 : Rotated Weierstrass Function F'_6

g_5 : Rotated High Conditioned Elliptic Function F'_1

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 6
Test Problem 28

$$N = 5, \vec{\sigma} = [10, 20, 30, 40, 50]$$
$$\vec{\lambda} = [2.5, 10, 2.5, 5e-4, 1e-6]$$
$$\vec{bias} = [0, 100, 200, 300, 400]$$

g_1 : Rotated Expanded Griewank's plus Rosenbrock's FunctionF'_{15}

g_2 : Rotated HappyCat Function F'_{13}

g_3 : Rotated Schwefel's Function F'_{11}

g_4 : Rotated Expanded Scaffer's F6 Function F'_{16}

g_5 : Rotated High Conditioned Elliptic Function F'_1

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 7
Test Problem 29

$$N = 3, \vec{\sigma} = [10, 30, 50]$$
$$\vec{\lambda} = [1, 1, 1]$$
$$\vec{bias} = [0, 100, 200]$$

g_1 : Hybrid Function 1 F'_{17}

g_2 : Hybrid Function 2 F'_{18}

g_3 : Hybrid Function 3 F'_{19}

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima
- Different properties for different sub-components

Composition Function 8
Test Problem 30

$$N = 3, \vec{\sigma} = [10, 30, 50]$$
$$\vec{\lambda} = [1, 1, 1]$$
$$\vec{bias} = [0, 100, 200]$$

g_1 : Hybrid Function 4 F'_{20}

g_2 : Hybrid Function 5 F'_{21}

g_3 : Hybrid Function 6 F'_{22}

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima
- Different properties for different sub-components

A.3 Problem Definitions for the Congress in Evolutionary Computation 2017

In the last decade various kinds of novel optimization algorithms have been proposed to solve real-parameter optimization problems, including the CEC'05, CEC'13, CEC'14 and CEC'16 Special Session on Real-Parameter Optimization. The CEC'17 Test suite is designed with the same structure as the CEC'14 test suite. In particular, this set of problems are designed given the feedback of the community about CEC'14. Note that in this contest are excluded the surrogate or meta-models methods. Thus, this special session is devoted to the approaches, algorithms and techniques for solving real parameter single objective optimization without making use of the exact definition of the test functions. Particularly this benchmark comprises thirty test functions (see Table A.2), which are derived of twenty basic functions, which are defined as follows.

Table A.2 Summary of the CEC'17 Test Functions

	No.	Functions	$F(\vec{x}^*)$
Unimodal Functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Sum of Different Power Function	200
	3	Shifted and Rotated Zakharov Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Rastringin's Function	500
	6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	7	Shifted and Rotated Lunacek Bi_Rastringin Function	700
	8	Shifted and Rotated Non-Continuous Rastringin's Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefel's Function	1000
Hybrid Function 1	11	Hybrid Function 1 (N=3)	1100
	12	Hybrid Function 2 (N=3)	1200
	13	Hybrid Function 3 (N=3)	1300
	14	Hybrid Function 4 (N=4)	1400
	15	Hybrid Function 5 (N=4)	1500
	16	Hybrid Function 6 (N=4)	1600
	17	Hybrid Function 7 (N=5)	1700
	18	Hybrid Function 8 (N=5)	1800
	19	Hybrid Function 9 (N=5)	1900
	20	Hybrid Function 10 (N=6)	2000
Composition Functions	21	Composition Function 1 (N=3)	2100
	22	Composition Function 2 (N=3)	2200
	23	Composition Function 3 (N=4)	2300
	24	Composition Function 4 (N=4)	2400
	25	Composition Function 5 (N=5)	2500
	26	Composition Function 6 (N=5)	2600
	27	Composition Function 7 (N=6)	2700
	28	Composition Function 8 (N=6)	2800
	29	Composition Function 9 (N=3)	2900
	30	Composition Function 10 (N=3)	3000
Search Range $[-100, 100]^n$			

Definition of Basic Functions

Bent Cigar Function

$$f_1(\vec{x}) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$$

Sum of Different Power Function

$$f_2(\vec{x}) = \sum_{i=1}^{n} |x_i|^{i+1}$$

Zakharov Function

$$f_3(\vec{x}) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5x_i\right)^2 + \left(\sum_{i=1}^{n} 0.5x_i\right)^4$$

Rosenbrock's Function

$$f_4(\vec{x}) = \sum_{i=1}^{n-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$$

Rastrigin's Function

$$f_5(\vec{x}) = \sum_{i=1}^{n} \left(x_i^2 - 10cos(2\pi x_i) + 10\right)$$

Expanded Scaffer's F6 Function

$$f_6(\vec{x}) = \left(\sum_{i=1}^{n-1} g(x_i, x_{i+1})\right) + g(x_n, x_1)$$

$$g(a,b) = 0.5 + \frac{(sin^2(\sqrt{a^2+b^2}) - 0.5)}{(1 + 0.001(a^2+b^2))^2}$$

Lunacek bi-Rastrigin Function

$$f_7(\vec{x}) = \min\left\{\sum_{i=1}^{n}(\hat{x}_i - \mu_0)^2, n + s\sum_{i=1}^{n}(\hat{x}_i - \mu_1)^2\right\} + 10(n - \sum_{i=1}^{n} cos(2\pi\hat{z}_i))$$

$$\mu_0 = 2.5, \quad \mu_1 = -\sqrt{\frac{\mu_0^2 - 1}{s}}, \quad s = 1 - \frac{1}{2\sqrt{n+20} - 8.2},$$

$$y = \frac{10(\vec{x} - \vec{o})}{100}, \quad \hat{x}_i = 2sign(x_i^*)y_i + \mu_0, \quad \vec{z} = \Lambda^{100}(\vec{\hat{x}} - \mu_0)$$

Non-continuous Rotated Rastrigin's Function

$$f_8(\vec{x}) = \sum_{i=1}^{n}(z_i^2 - 10cos(2\pi z_i) + 10) + f_{13}^*,$$

$$\vec{z} = \mathbf{M}_1\Lambda^1 0\mathbf{M}_2 T_{asy}^{0.2}(T_{osz}(\vec{y})), \quad y_i = \begin{cases} \hat{x}_i & |\hat{x}_i| \leq 0.5 \\ round(2\hat{x}_i) & |\hat{x}_i| > 0.5 \end{cases} \vec{\hat{x}} = \mathbf{M}_1 \frac{5.12(\vec{x} - \vec{o})}{100}, \quad T_{asy}^{\beta} = \begin{cases} x^{1+\beta\frac{i-1}{n-1}\sqrt{x_i}} & x_i > 0 \\ x_i & otherwise \end{cases} T_{osz} = sign(x_i)exp(\hat{x}_i + 0.049(sin(c_2\hat{x}_i) + sin(c_2\hat{x}_i)))$$

$$\hat{x}_i = \begin{cases} log(|x_i|) & x_i \neq 0 \\ 0 & otherwise \end{cases}, sign(x_i) = \begin{cases} -1 & x_i < 0 \\ 0 & x_i = 0 \\ 1 & otherwise \end{cases}, c_1 = \begin{cases} 10 & x_i > 0 \\ 5.5 & otherwise \end{cases}, \quad c_2 = \begin{cases} 7.9 & x_i > 0 \\ 3.1 & otherwise \end{cases}$$

Levy Function

$$f_9(\vec{x}) = sin^2(\pi w_1) + \sum_{i=1}^{n-1}(w_i - 1)^2[1 + 10sin^2(\pi w_i + 1)] + (w_n - 1)^2[1 + sin^2(\pi w_0)]$$

$$w_i = 1 + \frac{x_1 - 1}{4}$$

Modified Schwefel's Function

$$f_{10}(\vec{x}) = 418.9829 \times n - \sum_{i=1}^{n} g(z_i),$$

$$z_i = x_i + 4.209687462275036e + 002$$

$$g(z_i) = \begin{cases} z_i sin(|z_i|^{1/2}) & |z_i| \leq 500 \\ (500 - mod(z_i, 200))sin\left(\sqrt{|500 - mod(z_i, 500)|}\right) - \frac{(z_i - 500)^2}{10000n} & z_i > 500 \\ (mod(|z_i|, 500) - 500)\, sin\left(\sqrt{|mod(z_i, 500) - 500|}\right) - \frac{(z_i + 500)^2}{10000n} & z_i < -500 \end{cases}$$

High Conditioned Elliptic Function

$$f_{11}(\vec{x}) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$$

Discus Function

$$f_{12}(\vec{x}) = 10^6 x_1^2 + \sum_{i=2}^{n} x_i^2$$

Ackley's Function

$$f_{13}(\vec{x}) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$$

Weierstrass Function

$$f_{14}(\vec{x}) = \sum_{i=1}^{n}\left(\sum_{k=0}^{20}[0.5^k cos(2\pi 3^k (x_i + 0.5))]\right) - n\sum_{k=0}^{20}[0.5^k cos(\pi 3^k)]$$

Griewank's Function

$$f_{15}(\vec{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Katsuura Function

$$f_{16}(\vec{x}) = \frac{10}{n^2}\prod_{i=1}^{n}\left(1 + i\sum_{j=1}^{32}\frac{|2^j x_i - round(2^j x_i)|}{2^j}\right)^{\frac{10}{n^{1.2}}} - \frac{10}{n^2}$$

HappyCat Function

$$f_{17}(\vec{x}) = \left|\sum_{i=1}^{n} x_i^2 - n\right|^{1/4} + \left(0.5\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i\right)/n + 0.5$$

HGBat Function

$$f_{12}(\vec{x}) = \left|\left(\sum_{i=1}^{n} x_i^2\right)^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right|^{1/2} + \left(0.5\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i\right)/n + 0.5$$

A.3.1 Definitions of the CEC 2017 Test suite

Shifted and Rotated Bent Cigar
Test Problem 1

$$F_1(\vec{x}) = f_1(\mathbf{M}(\vec{x} - \vec{o}_1)) + F_1^*$$

- Non-separable
- Smooth but narrow ridge

Shifted and Rotated Sum of Different Power Function
Test Problem 2

$$F_2(\vec{x}) = f_2(\mathbf{M}(\vec{x} - \vec{o}_2)) + F_2^*$$

- Non-separable
- Symmetric

Shifted and Rotated Zakharov Function — Test Problem 3

$$F_3(\vec{x}) = f_3(\mathbf{M}(\vec{x}-\vec{o}_3)) + F_3^*$$

- Non-separable

Shifted and Rotated Rosenbrock's Function — Test Problem 4

$$F_4(\vec{x}) = f_4\left(\mathbf{M}\left(\frac{2.048(\vec{x}-\vec{o}_4)}{100}\right)+1\right) + F_4^*$$

- Non-separable
- Local optima's number is huge

Shifted and Rotated Ackley's Function — Test Problem 5

$$F_5(\vec{x}) = f_5(\mathbf{M}(\vec{x}-\vec{o}_5)) + F_5^*$$

- Non-separable
- Local optima's number is huge and second better local optimum is far from the global optimum

Shifted and Rotated Schaffer's F7 Function — Test Problem 6

$$F_6(\vec{x}) = f_{20}\left(\mathbf{M}\left(\frac{0.5\,(\vec{x}-\vec{o}_6)}{100}\right)\right) + F_6^*$$

- Non-separable
- Asymmetrical
- Local optima's number is huge

Shifted and Rotated Lunacek Bi-Rastrigin's Function — Test Problem 7

$$F_7(\vec{x}) = f_7\left(\mathbf{M}\left(\frac{600(\vec{x}-\vec{o}_7)}{100}\right)\right) + F_7^*$$

- Rotated
- Non-separable
- Asymmetrical
- Continuous everywhere yet differentiabl nowhere

Shifted and Rotated Non-Continuous Rastrigin's Function — Test Problem 8

$$F_8(\vec{x}) = f_8\left(\frac{5.12(\vec{x}-\vec{o}_8)}{100}\right) + F_8^*$$

- Non-separable
- Asymmetrical
- Local optima's number is huge

Shifted and Rotated Levy Function — Test Problem 9

$$F_9(\vec{x}) = f_9\left(\mathbf{M}\left(\frac{5.12(\vec{x}-\vec{o}_9)}{100}\right)\right) + F_9^*$$

- Non-separable
- Local optima's number is huge

Shifted and Rotated Schwefel's Function — Test Problem 10

$$F_{10}(\vec{x}) = f_{10}\left(\frac{1000(\vec{x}-\vec{o}_{10})}{100}\right) + F_{10}^*$$

- Non-separable
- Local optima's number is huge and second better local optimum is far from the global optimum

Hybrid Function 1 — Test Problem 11

$N = 3$, $\vec{p} = [0.2, 0.4, 0.4]$

g_1: Zakharov Function f_3

g_2: Rosenbrock Function f_4

g_3: Rastrigin's Function f f_5

Hybrid Function 2 — Test Problem 12

$N = 3$, $\vec{p} = [0.3, 0.3, 0.4]$

g_1: High Conditioned Elliptic Function f_{11}

g_2: Modified Schwefel's Function f_{10}

g_3: Bent Cigar Function f_1

Hybrid Function 3 — Test Problem 13

$N = 3$, $\vec{p} = [0.3, 0.3, 0.4]$

g_1: Bent Cigar Function f_1

g_2: Rosenbrock Function f_4

g_3: Lunache Bi-Rastrigin Function f_7

Hybrid Function 4 — Test Problem 14

$N = 4$, $\vec{p} = [0.2, 0.2, 0.2, 0.4]$

g_1: High Conditioned Elliptic Function f_{11}

g_2: Ackley's Function f_{13}

g_3: Schaffer's F7 Function f_{20}

g_4: Rastrigin's Function f_5

Hybrid Function 5 — Test Problem 15

$N = 4$, $\vec{p} = [0.2, 0.2, 0.3, 0.3]$

g_1: Bent Cigar Function f_1

g_2: HGBat Function f_{18}

g_3: Rastrigin's Function f_5

g_4: Rosenbrock's Function f_4

Hybrid Function 6 — Test Problem 16

$N = 4$, $\vec{p} = [0.2, 0.2, 0.3, 0.3]$

g_1: Expanded Schaffer F6 Function f_6

g_2: HGBat Function f_{18}

g_3: Rosenbrock's Function f_4

g_4: Modified Schwefel's Function f_{10}

Hybrid Function 7
Test Problem 17

$N = 5, \quad \vec{p} = [0.1, 0.2, 0.2, 0.2, 0.3]$
g_1: Katsuura Function f_{16}
g_2: Ackley's Function f_{13}
g_3: Expanded Griewank's plus Rosenbrock's Function f_{19}
g_4: Modified Schwefel's Function f_{10}
g_5: Rastrigin's Function f_5

Hybrid Function 8
Test Problem 18

$N = 5, \quad \vec{p} = [0.2, 0.2, 0.2, 0.2, 0.2]$
g_1: High Conditioned Elliptic Function f_1
g_2: Ackley's Function f_{13}
g_3: Rastrigin's Function f_5
g_4: HGBat Function f_{18}
g_5: Discus Function f_{12}

Hybrid Function 9
Test Problem 19

$N = 5, \quad \vec{p} = [0.2, 0.2, 0.2, 0.2, 0.2]$
g_1: Bent Cigar Function f_1
g_2: Rastrigin's Function f_5
g_3: Expanded Griewank's plus Rosenbrock's Function f_{19}
g_4: Weierstrass Function f_{14}
g_5: Expanded Schaffer's F6 Function f_6

Hybrid Function 10
Test Problem 20

$N = 6, \quad \vec{p} = [0.1, 0.1, 0.2, 0.2, 0.2, 0.2]$
g_1: Happycat Function f_{17}
g_2: Katsuura Function f_{16}
g_3: Ackley's Function f_{13}
g_4: Rastrigin's Function f_5
g_5: Modified Schwefel's Function f_{10}
g_5: Schaffer's F7 Function f_{20}

Composition Function 1
Test Problem 21

$$N = 3, \vec{\sigma} = [10, 20, 30]$$
$$\vec{\lambda} = [1, 1e-6, 1]$$
$$\vec{bias} = [0, 100, 200]$$
g_1 : Rotated Rosenbrock's Function F_4'
g_2 : High Conditioned Elliptic Function F_{11}'
g_3 : Rastrigin's Function F_5'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 2
Test Problem 22

$$N = 3, \vec{\sigma} = [10, 20, 30]$$
$$\vec{\lambda} = [1, 10, 1]$$
$$\vec{bias} = [0, 100, 200]$$
g_1 : Rastrigin's Function F_5'
g_2 : Griewank's Function F_{15}'
g_3 : Modifed Schwefel's Function F_{10}'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 3
Test Problem 23

$$N = 4, \vec{\sigma} = [10, 20, 30, 40]$$
$$\vec{\lambda} = [1, 10, 1, 1]$$
$$\vec{bias} = [0, 100, 200, 300]$$
g_1 : Rosenbrock's Function F_4'
g_2 : Ackley's Function F_{13}'
g_3 : Modified Schwefel's Function F_{10}'
g_4 : Rastrigin's Function F_5'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 4
Test Problem 24

$$N = 4, \vec{\sigma} = [10, 20, 30, 40]$$
$$\vec{\lambda} = [10, 1-6, 10, 1]$$
$$\vec{bias} = [0, 100, 200, 300]$$
g_1 : Ackley's Function F_{13}'
g_2 : High Conditioned Elliptic Function F_{11}'
g_3 : Girewank Function F_{15}'
g_4 : Rastrigin's Function F_5'

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

<table>
<tr><th>Composition Function 5
Test Problem 25</th><th>Composition Function 6
Test Problem 26</th></tr>
<tr><td>

$N = 5, \vec{\sigma} = [10, 20, 30, 40, 50]$
$\vec{\lambda} = [10, 1, 10, 1e-6, 1]$
$\vec{bias} = [0, 100, 200, 300, 400]$
g_1 : Rastrigin's Function F_5'
g_2 : Happycat Function F_{17}'
g_3 : Ackley Function F_{13}'
g_4 : Discus Function F_{12}'
g_5 : Rosenbrock's Function F_4'
• Multi-modal
• Non-separable
• Asymmetrical
• Different properties around different local optima

</td><td>

$N = 5, \vec{\sigma} = [10, 20, 20, 30, 40]$
$\vec{\lambda} = [1e-26, 10, 1e-6, 10, 5e-4]$
$\vec{bias} = [0, 100, 200, 300, 400]$
g_1 : Expanded Scaffer's F6 Function F_6'
g_2 : Modified Schwefel's Function F_{10}'
g_3 : Griewank's Function F_{15}'
g_4 : Rosenbrock's Function F_4'
g_5 : Rastrigin's Function F_5'
• Multi-modal
• Non-separable
• Asymmetrical
• Different properties around different local optima

</td></tr>
</table>

Composition Function 7
Test Problem 27

$$N = 6, \vec{\sigma} = [10, 20, 30, 40, 50, 60]$$
$$\vec{\lambda} = [10, 10, 2.5, 1e-26, 1e-6, 5e-4]$$
$$\vec{bias} = [0, 100, 200, 300, 400, 500]$$

g_1 : HGBat Function F'_{18}
g_2 : Rastrigin's Function F'_5
g_3 : Modified Schwefel's Function F'_{10}
g_4 :Bent-Cigar FunctionF'_1
g_5 :High Conditioned Elliptic FunctionF'_{12}
g_6 :Expanded Scaffer's F6 FunctionF'_6

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 8
Test Problem 28

$$N = 6, \vec{\sigma} = [10, 20, 30, 40, 50, 60]$$
$$\vec{\lambda} = [10, 10, 1e-6, 1, 1, 5e-4]$$
$$\vec{bias} = [0, 100, 200, 300, 400, 500]$$

g_1 : Ackley's Function F'_{13}
g_2 : Griewank's Function F'_{15}
g_3 : Discus Function F'_{12}
g_4 : Rosenbrock's Function F'_4
g_5 : HappyCat Function F'_{17}
g_6 :Expanded Scaffer's F6 Function F'_6

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima

Composition Function 9
Test Problem 29

$$N = 3, \vec{\sigma} = [10, 30, 50]$$
$$\vec{\lambda} = [1, 1, 1]$$
$$\vec{bias} = [0, 100, 200]$$

g_1 : Hybrid Function 5 F'_5
g_2 : Hybrid Function 6 F'_6
g_3 : Hybrid Function 7 F'_7

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima
- Different properties for different sub-components

Composition Function 10
Test Problem 30

$$N = 3, \vec{\sigma} = [10, 30, 50]$$
$$\vec{\lambda} = [1, 1, 1]$$
$$\vec{bias} = [0, 100, 200]$$

g_1 : Hybrid Function 5 F'_5
g_2 : Hybrid Function 8 F'_8
g_3 : Hybrid Function 9 F'_9

- Multi-modal
- Non-separable
- Asymmetrical
- Different properties around different local optima
- Different properties for different sub-components

Appendix B

Fundamental Concepts in Multi-Objective Optimization

B.1 Fundamental Concepts

This section provides some basic concepts and definitions from multi-objective optimization. Note that this section only gives some fundamental concepts and more information can be reviewed in Coello (2015); Deb (2001); Hernández Gómez (2018); Purshouse (2003); Rodríguez Villalobos and Coello Coello (2012).

Multi-objective Optimization problems (MOPs) involve the simultaneous optimization of several objective functions that are usually in conflict with each other (Deb, 2001). A continuous box-constrained minimization [1] MOP, which is the kind of problems addressed in this thesis, is formally defined in Definition B.1.1, where n corresponds to the dimensionality of the decision variable space, $\vec{x}$ is a vector of n decision variables $\vec{x} = (x_1, ..., x_n) \in \mathbb{R}^n$, which are constrained by $\vec{x}^{(L)}$ and $\vec{x}^{(U)}$, i.e. the lower bound and the upper bound, and m is the number of objective functions to optimize. The feasible space bounded by $\vec{x}^{(L)}$ and $\vec{x}^{(U)}$ is denoted by Ω. Each solution is mapped to the objective space with the function $F : \Omega \rightarrow \mathbb{R}^m$, which consists of m real-valued objective functions, and $\mathbb{R}^m$ is called the *objective space*.

Definition B.1.1 (**Multi-Objective Optimization Problem**). *Minimize the m components of a vector function $\vec{f}$ with respect to a vector variable $\vec{x} = (x_1, x_2, ..., x_n)$ in a universe Ω, i.e.,*

$$\begin{aligned} \min \quad & \vec{f} = [f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x})] \\ \text{subject to} \quad & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, ..., n. \end{aligned}$$

In multi-objective optimization, it is not possible to directly compare two solutions as in single-objective optimization. Therefore, the Pareto dominance relation can be considered to compare two solutions (Knowles and Corne, 2002). This is formalized in Definition B.1.2. Note that in the binary relation of dominance three scenarios might occur, either $\vec{x} \prec \vec{y}$, $\vec{y} \prec \vec{x}$ or neither of both (i.e. both are incomparable). The set of solutions that are not dominated by any other is called the non-dominated set. This is performed in Definition B.1.3.

Definition B.1.2 (**Pareto Dominance**). *Given two solutions $\vec{x}$, $\vec{y} \in \Omega$, vector $\vec{x}$ is said to dominate vector $\vec{y}$ (denoted by $\vec{x} \prec \vec{y}$) if and only if:*

$$\forall i \in \{1, ..., m\} : f_i(\vec{x}) \leq f_i(\vec{y}) \wedge \exists i \in \{1, ..., m\} : f_i(\vec{x}) < f_i(\vec{y})$$

Definition B.1.3 (**Non-Dominated Set**). *The Non-Dominated set of a specific set $A \subseteq \Omega$ is given by:*

$$ND(A) = \{\vec{a} \in A : \nexists \vec{b} \in A \wedge \vec{b} \prec \vec{a}\}$$

The best solutions of a MOP are the Non-Dominated set of the feasible space Ω. Each solution of *Omega* is termed as *Pareto optimal* if it fulfills Definition B.1.4. Therefore, the set of all solutions that are not dominated by any other feasible vector is termed the *Pareto optimal set* (Definition B.1.5). Similarly, the *Pareto front* (Definition B.1.6) is the image (i.e. the corresponding objective function values) of the Pareto optimal set. The goal of multi-objective optimizers is to obtain a proper approximation of the Pareto front, i.e. a set of well-distributed solutions that are close to the Pareto front.

Definition B.1.4 (**Pareto Optimal**). *Let $\vec{x^*} \in \Omega$, we say that $\vec{x^*}$ is a "Pareto optimal" solution, if there is no other solution $\vec{y} \in \Omega$ such that: $\vec{y} \prec \vec{x}$.*

[1]Without loss of generality, in this thesis only minimization problems are considered.

Definition B.1.5 (Pareto Optimal Set). *The "Pareto Optimal Set" (PS) is defined by*

$$PS = \{\vec{x} \in \Omega | \vec{x} \text{ is a Pareto Optimal solution}\}.$$

Definition B.1.6 (Pareto Optimal Front). *The "Pareto Front" (PF) is defined by:*

$$PF = \{\vec{f} = [f_1(\vec{x}), ..., f_m(\vec{x})] | \vec{x} \in PS\}.$$

The Pareto optimal front of a MOP is bounded by two special vectors, which are the *Ideal Objective Vector* (Definition B.1.7) and the *Nadir Objective Vector* (Definition B.1.9). In some algorithms (e.g. SMS-EMOA), the approximation of the Ideal Objective Vector considers all the individuals that have been created so far. Differently, the Nadir Objective Vector is approximated taking into account the non-dominated set of the current population.

Definition B.1.7 (Ideal Objective Vector). *The "Ideal Objective Vector" denoted $\vec{z^*} \in \mathbb{R}^m$ minimizes all the objectives functions. Each component is defined by:*

$$z_i^* = \min\{f_i(\vec{x}) | \vec{x} \in \Omega\} \forall i \in \{1, ..., m\}.$$

Definition B.1.8 (Utopian Objective Vector). *The "Utopian Objective Vector" is an infeasible objective vector whose components are formed by $\vec{z_i^{**}} = z_i^* - \varepsilon_i \forall i \in \{1, ..., m\}$, where ε_i is some small positive scalar.*

Definition B.1.9 (Nadir Objective Vector). *The "Nadir Objective Vector" $\vec{z^{nad}} \in \mathbb{R}^m$ is constructed using the worst values of the Pareto optimal front. Each component is defined by:*

$$z_i^{nad} = \max\{f_i(\vec{x}) | \vec{x} \in PS\} \forall i \in \{1, ..., m\}.$$

B.1.1 Goals of Multi-Objective Evolutionary Algorithms

According to the previous definitions, the essential objective of an optimizer is to find the Pareto front. Nevertheless, this is not possible in all the kind of MOPs (e.g. continuous Pareto Fronts) because an infinity number of points might be required. Alternatively, the optimizers aim to find a discrete approximation that better fulfills some requirements that can be imposed by the decision maker. Commonly, the approximation of an optimal set is carried out by the following three goals:

- Convergence to the Pareto optimal front.
- An uniform distribution of solutions along the Pareto front.
- Coverage of the entire Pareto front.

B.1.2 Performance indicators

A quality indicator is a function that evaluates Pareto-front approximations by mapping them to a single real value (Guerreiro et al., 2020; Zitzler and Künzli, 2004; Zitzler et al., 2003). These quality indicators usually measure both convergence and diversity simultaneously. According to Zitzler et al. (2003) a quality indicator is formalized in Definition B.1.10.

Definition B.1.10 (Quality Indicator). *An $z-$ry quality indicator I is a function $I : \Omega^z \rightarrow \mathbb{R}$, which assigns each vector $(A_1,...,A_z)$ of z approximation sets a real value $I(A_1,...,A_z)$.*

Unary quality indicators are most commonly used in the literature. What makes them attractive is their capability of assigning quality values to an approximation set independent of other sets under consideration. This allows the comparison between approximations sets. Note that the comparison between sets is also possible with the weak-Pareto dominance relation. In particular, a set A weakly dominates B if and only if Eqn. (B.1) is fulfilled.

$$\{A \preceq B \Longleftrightarrow \forall \vec{b} \in B, \exists \vec{a} \in A : \vec{a} \preceq \vec{b}\} \tag{B.1}$$

As consequence, an unary performance indicator I is *weakly Pareto Compliant* if and only if Eqn. (B.2) is fulfilled.

$$A \preceq B \Rightarrow I(A) \geq I(B). \tag{B.2}$$

This unary indicator might also be *Pareto compliant* if and only if Eqn. (B.3) is accomplished.

$$(A \prec B \land A \neq B) \Rightarrow I(A) > I(B) \tag{B.3}$$

Note that in this context the quality indicator I must be maximized. In other words, an unary indicator is Pareto compliant if it "respects" the order induced by the dominance relation. Thus, the measure of $I(A)$ should be better than $I(B)$ if A dominates B. In order to provide fair comparisons, the indicators should be Pareto-compliant or at least weakly Pareto-compliant. The quality indicators that are not Pareto-compliant might mislead the search process (Zitzler et al., 2003).

B.1.3 Hypervolume Indicator

The Hypervolume (HV) or S metric is one of the most popular Pareto-compliant quality indicators. The HV rewards the convergence and distributions of the solutions towards the Pareto front. This indicator favors solutions in regions where the Pareto front has knee-points and captures regions with different fair trade-offs between different objectives (Dhiman and Kumar, 2019). Therefore, the regions located on the flanks (or flat regions) of the Pareto front are less rewarded by this indicator. However it still maintains the external solutions (Beume et al., 2007). The hypervolume measures the portion of the objective space that is dominated by an approximated set.

Formally, the HV can be defined as follows (Diaz and López-Ibáñez, 2021; Shang et al., 2020). Given an approximation set $A \subset \mathbb{R}^m$ and a reference vector $\vec{r} \in \mathbb{R}^m$, the hypervolume of the point set A can be calculated according to Eqn. (B.4).

$$HV(A,\vec{r}) = \int^{\vec{r}} \alpha_A(\vec{z})d\vec{z} \tag{B.4}$$

where $\alpha_A(\vec{z})$ is the attainment function (a.k.a. characteristic function), $\alpha_A : \mathbb{R}^m \rightarrow \{0,1\}$ specifying which points are weakly dominated by A formally defined in Eqn. (B.5).

$$\alpha_A(\vec{z}) = \begin{cases} 1, & \text{if } A \preceq \{\vec{z}\} \\ 0, & \text{otherwise} \end{cases} \tag{B.5}$$

The HV is widely integrated into the design of MOEAs to guide the optimization process (Beume et al., 2007). Thus, the hypervolume contribution of an individual solution measures the influence of a solution to the approximation set. The contribution of a solution $\vec{a} \in A$ to the approximation set A in terms of HV is defined in Eqn. (B.6).

$$\Delta HV(\vec{a}, A, \vec{r}) = HV(A, \vec{r}) - HV(A \setminus \vec{a}, \vec{r}) \tag{B.6}$$

The nice mathematical properties of HV have allowed the design of a large amount of MOEAs. Nevertheless, its computational costs increases exponentially with the number of objectives. Note that there are several works that aim to compute HV more efficiently (Shang et al., 2020).

B.1.4 Generational Distance and Inverted Generational Distance

The Generational Distance (GD) (Van Veldhuizen, 1999; Zitzler et al., 2003) and Inverted Generational Distance (IGD) (Coello and Sierra, 2004; Ishibuchi et al., 2015) are two well-known performance indicators. These unary indicators evaluate the quality of an objective vector set using a reference set. The term "Inverted Generational Distance" was first used in Coello and Sierra (2004). However, similar indicators have been presented before (Bosman and Thierens, 2003; Czyzżak and Jaszkiewicz, 1998; Ishibuchi et al., 2003). Recently, a mathematical analysis of convergence with both indicators was carried out (Schutze et al., 2012). This analysis compared sets with different magnitudes and concluded that these metrics have undesirable effects as the magnitude of the sets is increased. To alleviate this shortcoming a sightly change in both definitions makes both indicators more stable. This modified version of GD and IGD is as follows. Let $Z = \{\vec{z}_1, ..., \vec{z}_{|Z|}\}$ be a given reference vector set where $|Z|$ is the cardinality of Z, and a non-dominated objective vector set $A = \{\vec{a}_1, ..., \vec{a}_{|A|}\}$ with cardinality $|A|$ then the GD is defined in Eqn. (B.7) where d_i is a distance (B.8) from $\vec{a}_i \in A$ to its nearest reference vector $\vec{z}_i \in Z$, and p is an integer parameter (usually is $p = 2$ or $p = 3$).

$$GD(A, Z) = \left(\frac{1}{|A|} \sum_{\vec{a} \in A} d(\vec{a}, Z)^p \right)^{1/p} \tag{B.7}$$

$$d(\vec{a}, Z)^p = \min_{\vec{z} \in Z} \sqrt[p]{\sum_{i=1}^{m} (a_i - z_i)^2} \tag{B.8}$$

Similarly, IGD is defined in Eqn. (B.9) where $\hat{d}_j$ is the Euclidean distance from $\vec{z}_j$ to its nearest objective vector to A.

$$IGD(A, Z) = \left(\frac{1}{|Z|} \sum_{\vec{z} \in Z} \hat{d}(\vec{z}, A)^p \right)^{1/p} \tag{B.9}$$

$$\hat{d}(\vec{z}, A)^p = \min_{\vec{a} \in A} \sqrt[p]{\sum_{i=1}^{m} (z_i - a_i)^2} \tag{B.10}$$

Since GD andIGD are non Pareto-compliant indicators, in Ishibuchi et al. (2015) the Inverted Generational Distance Plus (IGD+) was proposed, which is an improvement of IGD that is weakly Pareto compliant. The definition of IGD+ is given in Eqn. (B.11).

$$IGD^{+}(A,Z) = \frac{1}{|Z|}\left(\sum_{\vec{z}\in Z}\hat{d}^{+}(\vec{z},A)^{p}\right)^{1/p} \tag{B.11}$$

The underlying idea of this last performance indicator is that the distance $\hat{d}^{+}$ considers the Pareto dominance relation indicated in Eqn (B.12).

$$\hat{d}^{+}(\vec{z},A)^{p} = \min_{\vec{a}\in A}\sqrt[p]{\sum_{i=1}^{m}(max\{a_i - z_i, 0\})^2} \tag{B.12}$$

Note that all these indicators are to be minimized and the optimum is zero.

B.1.5 Distance to the Closest Neighbor

The Distance to the Closest Neighbor (DCN) is a metric used to measure the contribution to diversity of an individual (Bui et al., 2005; Segura et al., 2013b). Given an individual $\vec{i}\in\Omega$ and a set Q, the contribution to the diversity of $\vec{i}$ is the minimum distance between $\vec{i}$ and any individual of Q, this is defined in Eqn. (B.13). Note that any distance might be applied, and in the continuous domain, the Euclidean distance is the most popular one.

$$DCN(\vec{i},Q) = \min_{\vec{q}\in Q} d(\vec{i},\vec{q}) \tag{B.13}$$

B.1.6 Average Distance to All Individuals

The average (mean) Euclidean distance among all the individuals (ADI) in the population is usually taken as a diversity measurement (Bui et al., 2005). The average distance of an individual $\vec{i}\in\Omega$, takes into account the average of distances between an individual $\vec{i}$ and the remaining individuals in the population Q.

B.1.7 R2 Indicator

The R2 indicator belongs to the R-indicator family and was introduced by Hansen et al. (Hansen and Jaszkiewicz, 1994). In this indicator, the Pareto front approximations are assessed based on a set of utility functions, allowing the comparison of approximations by means. Note that with any variant of the R-indicator family, certain regions of the Pareto front can be determined by the specification of utility functions. The R-indicator family defines three variants of functions, each one modifies the way in which the utilities are evaluated and combined. Among these three variants, the second variant ($R2$) is quite popular because of its Weakly Pareto compatibility (Hernández Gómez and Coello Coello, 2015; Zitzler et al., 2008). The $R2$ indicator is a popular approach in many-objective optimization, frequently preferred instead to the hypervolume indicator given its low computational cost and that it produces more uniform solutions. However, this indicator requires a set of weight vectors whose proper distribution depends on the Pareto geometry. Although the R-indicator family has solid mathematical basis, in practice an approximation of it is applied (Rodríguez Villalobos and Coello Coello, 2012). The definition of this approximation is formulated as follows.

Given a set U of utility functions, which have an associated set of weight vectors $\vec{\lambda}\in\Lambda$, and an approximation set A, the unary version of the $R2$ indicator is given by Eqn. (B.14). Note that $|U|$ is the carnality of the utility functions and lower values are preferred in this indicator.

$$R2(A,\Lambda) = -\frac{1}{|U|}\sum_{u\in U}\max_{\vec{a}\in A}\{u(\vec{a},\vec{\lambda})\} \tag{B.14}$$

In literature several choices for the utility function u can be found. Some of them are *Weighted Sum (WS)*, *Chebyshev Function (TCHE)*, *Penalty Boundary Intersection (PBI)*, *Augmented Achievement Scalarizing Function (AASF)* and *Achievement Scalarizing Function (*ASF*)*. This thesis takes into account ASF (Miettinen, 2012). Therefore, given a reference vector $\vec{r}$ (ideal objective vector), a set of solutions A and a set of weight vectors $\vec{\lambda} = (\lambda_1, ..., \lambda_M) \in \Lambda$, the R2 is defined in Eqn. (B.15).

$$R2(A,\Lambda,\vec{r}) = \frac{1}{|\Lambda|}\sum_{\vec{\lambda}\in\Lambda}\min_{\vec{a}\in A}\left\{\max_{j\in 1,..,M}\left\{\frac{r_j - a_j}{\lambda_j}\right\}\right\} \tag{B.15}$$

Similarly to the hypervolume indicator, the contribution of a solution $\vec{a} \in A$ to the R2 indicator can be calculated according Eqn. (B.16).

$$\Delta R2(\vec{a},A,\Lambda,\vec{r}) = R2(A,\Lambda,\vec{r}) - R2(A\setminus\vec{a},\Lambda\vec{r}) \tag{B.16}$$

An important inconvenience of the R2 indicator lies in its instability with the scale of the objectives. In practice, objectives are normalized "online" with the ideal objective vector and the worst objective vector.

B.1.8 Performance Indicators in Multi-criteria Decision Making

The mathematical properties of the indicator-base MOEAs have allowed the successful integration of preferences imposed by the decision maker (Trautmann et al., 2013; Zitzler et al., 2007; Zitzler and Künzli, 2004). Two of the most popular algorithms that integrate preferences with slight modifications are Hypervolume (HV) (Guerreiro et al., 2020; Shang et al., 2020; Zitzler and Thiele, 1998) and the R2 indicator (Hansen and Jaszkiewicz, 1998). Note that the performance indicators considered by these MOEAs are Pareto-compliant and weakly Pareto-compliant, respectively. Therefore, these two methods are highly recommended to achieve the preferences of the DM and the goals of multi-objective optimization (Wagner et al., 2013). Certain regions of the objective space might be preferred by the DM, which can be carried out with the weighted hypervolume indicator (HV^w) proposed by Auger et al. (2009). The weighted hypervolume indicator integrates the DM preferences by assigning a weight function to the objective space. Therefore, two sets with the same HV may differ in terms of HV^w. In particular, the HV^w assigns a positive weight to each point in the objective space, $w : \mathbb{R}^m \to \mathbb{R}_{>0}$, resulting in the definition of the Eqn. (B.17), where $\alpha_A(\vec{z})$ is the attainment function (a.k.a. characteristic function) previously described (Eqn. B.5).

$$HV^w(A,\vec{r}) = \int^{\vec{r}} w(\vec{z})\alpha_A(\vec{z})d\vec{z} \tag{B.17}$$

Regarding the R2-indicator, there are three popular strategies for integrating preferences (Wagner et al., 2013). These strategies consist of fixing the reference vector, restricting the weight space, and skewing the weight vector distribution. Among them, the reference vector, as a parameter of R2, has the most relevant influence on the final result. Especially, in case it is chosen too close to the Pareto front, i.e. it does not dominate the whole Pareto front, the optimal distributions regarding R2 will never cover the whole extent of the Pareto front, independent of the number of weight vectors. However, the number of weight vectors that intersect the Pareto front decreases with increasing distance from the front. The recommendation is to apply a very conservative reference vector (far better than the approximated Pareto front), and use more weight vectors which increases the probability to intersect the complete Pareto front.

B.1.9 Attainment Surfaces

A visualization tool that compares the performance of several optimizers along multiple runs are the attainment surfaces. Visual analysis of several approximation sets is useful for several reasons:

- Some performance indicators are not always capable of discerning order relations between the approximation set of even a single pair of optimization runs.
- The decision maker may have preferences towards certain regions or shapes of the Pareto front that are unknown before optimization (for a posteriori analyzes).
- In some cases, the performance indicators are not able to discern adequately when one approximation set should be judged better that another.
- Visualizing the approximation set might give insights of the strengths and weakness of an optimizer.

For these reasons, in this thesis, several experimental validations include both performance indicators and plots of approximated sets (summary of attainment surfaces).

An attainment surface is used to describe an approximation set, it combines the information about the quality and the distribution of the corresponding non-dominated points across the trade-off surface (Fonseca and Fleming, 1996). Given a non-dominated set, an attainment surface can be defined as the boundary that splits the dominated region of the non-dominated region. Note that the visualization of attainment surfaces is usually considered for two-objective problems and rarely considered for three-objective problems (or more). The closer the non-dominated approximated solutions are between them, the closer the real trade-off surface the attainment surface will be. As is expected closer approximations might require a large amount of solutions. An advantage of using attainment surfaces is that it allows to understand the distribution of the location over multiple runs. Illustratively, an attainment surface can be drawn in the two-objective case connecting all the goals (approximations) with straight lines. In this context, an approximation or solution is termed as a *goal*.

In stochastic optimization is mandatory to perform multiple runs to properly assess an optimizer for a specific problem, this is also the case for multi-objective optimization. These multiple approximation sets can be represented by attainment surfaces that preserve the information about how the approximations are distributed. The attainment surfaces of several approximations can be represented in a *summary attainment surface*, which divides the objective space in regions that were attained only in a certain fractions of the runs. The formal definition of a summary attainment surface is as follows. A summary attainment surface is defined as the union of all tightest goals (solutions) that have been attained (independently) in precisely s of the runs of a sample of n runs, for any $s \in \{1, ..., n\}$, and for any dimension (Knowles, 2005). Following this definition a 50% attainment surface (the median) illustrates the regions of the objective space that were attained only by a half of the runs.

A similar concept is the Empirical Attainment Function (EAF), which describes the frequency that a specific region was attained by a multi-set of approximated fronts. The main difference between an empirical attainment function and an attainment surface is that the former describes a probabilistic distribution of the outcomes in a simple and elegant way, while the latter describes the outcomes by only surfaces (one attainment surface by outcome) (Da Fonseca et al., 2001). Note that an EAF can be estimated through several attainment surfaces. In other words, an EAF represents the boundary separating points that are known to be attainable in at least a fraction (quantile) of the runs from those that are not. Another visualization tool that allows a pair-wise comparison are the *Empirical Attainment Function Differences*, which plot the differences between the empirical attainment functions of two data sets as a two-panel plot (Fonseca et al., 2015).

Appendix C

Multi-Objective Test Problems

The multi-objective problems used in the experimental validation of this thesis are defined in this section. In particular, we adopted the Deb-Thiele-Laumanns-Zitzled (DTLZ)(Deb et al., 2005), the Walking-Fish-Group (WFG) (Huband et al., 2005), the Unconstrained Test Problems (UF) (Zhang et al., 2008), Biases Test Problems (BT) (Li et al., 2016a), and Imbalance Test Problems (IMB) (Liu et al., 2016). All the minimization problems presented in this section are scalable in the decision variable, and some of them are also scalable with respect to the number of objectives. Particularly, the experimental validation presented in Chapter 5 considered the DTLZ, WFG and UF test problems. Chapter 6 also considered the BT problems. Finally, Chapter 7 takes into account the same test problems replacing the BT test problems with the IMB test problems. Among all the test problems, the Pareto geometry is quite diverse. Specifically, some geometries are linear, concave, convex, mixed, degenerated and disconnected. They also include some aspects such as separability and multi-frontality which make them more challenging. Table C.1 and Table C.2 summarizes the main features of these test problems.

C.1 Deb-Thiele-Laumanns-Zitzler Test Problems (DTLZ)

The DTLZ suite of benchmark problems, created by Deb et al. (2005), was one of the first multi-objective test problems that can be scaled to any number of objectives. Nine test problems are included in the DTLZ test suite[1], of which the first seven are defined in this section. The remaining two are not considered because they have side constraints. All DTLZ1-DTLZ6 are scalable with respect to the number of distance parameters, but have a fixed number of $m-1$ position parameters, where m is the number of objectives. Note also that the objective functions of DTLZ1-DTLZ4 have multiple global optima. In all cases, the range of the decision variables is $[0,1]$. The number of decision variables is given by $n = m+k-1$, where k is the number of distance-related parameters. The distance vector of the k entries is defined as $\vec{y} = \{x_m, x_{m+1}, ..., x_n\}$, considering the decision vector $\vec{x} = \{x_1, ..., x_{m-1}, x_m, ..., x_n\}$. The authors (Deb et al., 2005) suggested k-values of 5 for DTLZ1, 10 for DTLZ2-DTLZ6, and 20 for DTLZ7. In the case of DTLZ4 $\alpha = 100$ is suggested.

The Pareto optimal solutions correspond to $\vec{y} = (0.5, 0.5, ..)^T$ for DTLZ1-DTLZ5 and $\vec{y} = (0, 0, ..)^T$ for DTLZ6-7. Moreover, the Pareto front shape is described by $\sum_{i=1}^{m} f_i = 0.5$ for DTLZ1, and $\sum_{i=1}^{m} f_i^2 = 1$ for DTLZ2-DTLZ4.

[1]Note that the number of DTLZ problems depends on which paper is being referred to. In this thesis we are using the problems defined in the technical report Deb et al. (2005). A conference paper version (Deb et al., 2002b) of the technical report also exists, in which only seven of the nine original problems are taken into account

DTLZ1

$$f_1(\vec{x}) = 0.5(1+g(\vec{y}))\prod_{i=1}^{m-1} x_i$$

$$f_{j=2:m-1}(\vec{x}) = 0.5(1+g(\vec{y}))(1-x_{m-j+1})\prod_{i=1}^{m-j} x_i$$

$$f_m(\vec{x}) = 0.5(1+g(\vec{y}))(1-x_1)$$

$$g(\vec{y}) = 100\left(k+\sum_{i=1}^{k}(y_i-0.5)^2 - cos(20\pi(y_i-0.5))\right)$$

DTLZ2

$$f_1(\vec{x}) = (1+g(\vec{y}))\prod_{i=1}^{m-1} cos(x_i\pi/2)$$

$$f_{j=2:m-1}(\vec{x}) = (1+g(\vec{y}))\left(\prod_{i=1}^{m-j} cos(x_i\pi/2)\right) sin(x_{m-j+1}\pi/2)$$

$$f_m(\vec{x}) = (1+g(\vec{y}))sin(x_1\pi/2)$$

$$g(\vec{y}) = \sum_{i=1}^{k}(y_i-0.5)^2$$

DTLZ3

$$f_1(\vec{x}) = (1+g(\vec{y}))\prod_{i=1}^{m-1} cos(x_i\pi/2)$$

$$f_{j=2:m-1}(\vec{x}) = (1+g(\vec{y}))\left(\prod_{i=1}^{m-j} cos(x_i\pi/2)\right) sin(x_{m-j+1}\pi/2)$$

$$f_m(\vec{x}) = (1+g(\vec{y}))sin(x_1\pi/2)$$

$$g(\vec{y}) = 100\left(k+\sum_{i=1}^{k}(y_i-0.5)^2 - cos(20\pi(y_i-0.5))\right)$$

DTLZ4

$$f_1(\vec{x}) = (1+g(\vec{y}))\prod_{i=1}^{m-1} cos(x_i^{\alpha}\pi/2)$$

$$f_{j=2:m-1}(\vec{x}) = (1+g(\vec{y}))\left(\prod_{i=1}^{m-j} cos(x_i^{\alpha}\pi/2)\right) sin(x_{m-j+1}^{\alpha}\pi/2)$$

$$f_m(\vec{x}) = (1+g(\vec{y}))sin(x_1^{\alpha}\pi/2)$$

$$g(\vec{y}) = \sum_{i=1}^{k}(y_i-0.5)^2$$

DTLZ5

$$f_1(\vec{x}) = (1+g(\vec{y}))\prod_{i=1}^{m-1} cos(\theta_i\pi/2)$$

$$f_{j=2:m-1}(\vec{x}) = (1+g(\vec{y}))\left(\prod_{i=1}^{m-j} cos(\theta_i\pi/2)\right)$$

$$sin(\theta_{m-j+1}\pi/2)$$

$$f_m(\vec{x}) = (1+g(\vec{y}))sin(\theta_1\pi/2)$$

$$\theta_i = \begin{cases} x_i, & i=1 \\ \frac{1+2g(\vec{y})}{2(1+g(\vec{y}))}x_i, & \text{otherwise} \end{cases}$$

$$g(\vec{y}) = \sum_{i=1}^{k}(y_i-0.5)^2$$

DTLZ6

$$f_1(\vec{x}) = (1+g(\vec{y}))\prod_{i=1}^{m-1} cos(\theta_i\pi/2)$$

$$f_{j=2:m-1}(\vec{x}) = (1+g(\vec{y}))\left(\prod_{i=1}^{m-j} cos(\theta_i\pi/2)\right)$$

$$sin(\theta_{m-j+1}\pi/2)$$

$$f_m(\vec{x}) = (1+g(\vec{y}))sin(\theta_1\pi/2)$$

$$\theta_i = \begin{cases} x_i, & i=1 \\ \frac{1+2g(\vec{y})}{2(1+g(\vec{y}))}x_i, & \text{otherwise} \end{cases}$$

$$g(\vec{y}) = \sum_{i=1}^{k} y_i^{0.1}$$

DTLZ7

$$f_{j=1:m-1}(\vec{x}) = x_j$$

$$f_m(\vec{x}) = (1+g(\vec{y}))\left(m - \sum_{i=1}^{m-1}\left[\frac{f_i}{1+g(\vec{y})}(1+sin(3\pi f_i)\right]\right)$$

$$g(\vec{y}) = 1+\frac{9}{k}\sum_{i=1}^{k} y_i$$

Table C.1 Properties of the test problems (part I)

Problems	Separability	Modality	Geometry	Bias	Number of Objectives
DTLZ1	separable	multi	linear	no	scalable
DTLZ2	separable	uni	concave	no	scalable
DTLZ3	separable	multi	concave	no	scalable
DTLZ4	separable	uni	concave	polynomial	scalable
DTLZ5	unknown	uni	arc, degenerated	parameter dependent	scalable
DTLZ6	unknown	uni	arc, degenerated	parameter dependent	scalable
DTLZ7	$f_{1:m-1}$ not applicable f_m separable	$f_{1:m_1}$ Uni f_m Multi	disconnected, mixed	no	scalable
WFG1	separable	uni	convex, mixed	polynomial, flat	scalable
WFG2	non-separable	$f_{1:m-1}$ uni, f_m multi	convex, disconnected	no	scalable
WFG3	non-separable	uni	linear, degenerated	no	scalable
WFG4	separable	multi	concave	no	scalable
WFG5	separable	deceptive	concave	no	scalable
WFG6	non-separable	uni	concave	no	scalable
WFG7	separable	uni	concave	parameter dependent	scalable
WFG8	non-separable	uni	concave	parameter dependent	scalable
WFG9	non-separable	multi, deceptive	concave	parameter dependent	scalable
BT1	separable	uni	convex	parameter dependent	two-objective
BT2	separable	uni	convex	parameter dependent	two-objective
BT3	separable	uni	convex	parameter dependent (position-related)	two-objective
BT4	separable	uni	convex	parameter dependent (position-related)	two-objective
BT5	separable	uni	convex, disconnected	parameter dependent	two-objective
BT6	non-separable	uni	convex	parameter dependent	two-objective
BT7	non-separable	uni	convex	parameter dependent	two-objective
BT8	non-separable	multi	convex	parameter dependent	two-objective
BT9	separable	uni	concave	parameter dependent	three-objective

Table C.2 Properties of the test problems (part II)

Problems	Separability	Modality	Geometry	Bias	Number of Objectives
UF1	non-separable	multi	convex	no	two-objective
UF2	non-separable	multi	convex	no	two-objective
UF3	non-separable	multi	convex	no	two-objective
UF4	non-separable	multi	concave	no	two-objective
UF5	non-separable	multi	only 21 points, lineal	no	two-objective
UF6	non-separable	multi	lineal	no	two-objective
UF7	non-separable	multi	lineal	no	two-objective
UF8	non-separable	multi	concave	no	three-objective
UF9	non-separable	multi	plane, disconnected	no	three-objective
UF10	non-separable	multi	concave	no	three-objective
IMB1	non-separable	uni	convex	polynomial (unfavored part)	two-objective
IMB2	non-separable	uni	linear	polynomial (unfavored part)	two-objective
IMB3	non-separable	uni	concave	polynomial (unfavored part)	two-objective
IMB4	non-separable	uni	plane, simplex	polynomial (unfavored part)	three-objective
IMB5	non-separable	uni	concave	polynomial (unfavored part)	three-objective
IMB6	non-separable	uni	plane, simplex	polynomial (unfavored part)	three-objective
IMB7	non-separable	uni	convex	polynomial (unfavored part)	two-objective
IMB8	non-separable	uni	linear	polynomial (unfavored part)	two-objective
IMB9	non-separable	uni	concave	polynomial (unfavored part)	two-objective
IMB10	non-separable	uni	plane, simplex	polynomial (unfavored part)	three-objective

C.2 Walking Fish Group Test Problems

Originally, the WFG toolkit Huband et al. (2005, 2006) provides a variety of predefined shape and transformation functions to allow the design of test problems, via a series of composable transformations, which features will be present in the test problem. As an example, their authors suggested nine benchmark multi-objective problems with that toolkit. However, the community popularly considered these nine problems as the Walking-Fish-Group (WFG) test problems to be part of their experimental validations.

The nine multi-objective test problems are scalable with respect to both objectives and variables. The notation adopted in this section is the same as that proposed in the documents Hernández Gómez (2018); Huband et al. (2006). The WFG Toolkit defines a problem in terms of an underlying vector of parameters $\vec{x}$. The vector $\vec{x}$ is always associated with a simple underlying problem that defines the fitness space. The vector $\vec{x}$ is derived, by a series of transition vectors, from a vector of working parameters $\vec{z}$. Each transition vector adds complexity to the underlying problem, such as multi-modality and non-separability. The algorithm directly manipulates $\vec{z}$, through which $\vec{x}$ is indirectly manipulated. Note that m is the number of objectives, $\vec{x}$ is a set of m underlying parameters, $\vec{z}$ is a set of $k+l=n$ working parameters, where the first k and the last l working parameters are position- and distance-related parameters, respectively. All $x_i \in \vec{x}$ will have domain $[0,1]$, and the domain of all $z_i \in \vec{z}$ is $[0,2i]$. Note that in the optimization process only the vector $\vec{z}$ is modified which is transformed in $\vec{x}$. For WFG1-WFG7, a solution is Pareto optimal iff $z_{i=k+1:n} = 2i \times 0.35$. For WFG8, it is required that all of:

$$\begin{aligned} z_{i=k+1:n} &= 2i \times 0.35^{\left(0.02+49.48\left(\frac{0.98}{49.98}-(1-2u)|\lfloor 0.5-u\rfloor+\frac{0.98}{49.98}|\right)\right)^{-1}}, \\ u &= r\_sum(\{z_1,...,z_{i-1}\},\{1,..,1\}) \end{aligned} \tag{C.1}$$

To obtain a Pareto optimal solution, the position should first be determined by setting $z_{i:k}$ appropriately. The required distance-related parameter values can then be calculated by first determining z_{k+1} (which is trivial given that z_k have been set), then z_{k+2}, and so on, until z_n has been calculated.

In the case of WFG9, for a solution to be Pareto optimal, it is required that all of

$$\begin{aligned} z_{i=k+1:n} &= 2i \times \begin{cases} 0.35^{(0.02+1.96u)^{-1}}, & i \neq n \\ 0.35, & i = n \end{cases} \\ u &= r\_sum(\{z_{i+1},...,z_n\},\{1,...,1\}) \end{aligned} \tag{C.2}$$

which can be found by first determining z_n, then z_{n-1}, and so on, until the required value for z_{k+1} is determined. Once the optimal values for z_{k+1} are determined, the position-related parameters can be varied arbitrarily to obtain different Pareto optimal solutions.

Each problem is composed by a number of transformation functions that map parameters with domain $[0,1]$ into the range $[0,1]$. There are three types of transformation functions: *bias*, *shift*, and *reduction functions*. Bias and shift functions only employ one parameter, whereas reduction functions can employ many. Bias transformations impact the search process by biasing the fitness landscape. Shift transformations move the location of optima, and are used to apply a linear shift, or to produce deceptive and multi-modal problems. Reduction transformations are used to produce non-separability of the problem. These three transformations are defined as follows.

Bias: Polynomial

When $\alpha > 1$ or when $\alpha < 1$, y is biased towards zero or towards one, respectively.

$$b\_poly(y, \alpha) = y^{\alpha}, \tag{C.3}$$

where $\alpha > 0$ and $\alpha \neq 1$.

Bias: Flat Region

Values of y between B and C (the whole flat region) are mapped to the value A.

$$b\_flat(y,A,B) = A + min(0, \lfloor y - B \rfloor)\frac{A(B-y)}{B} - min(0, \lfloor C - y \rfloor)\frac{(1-A)(y-C)}{1-C}, \tag{C.4}$$

where A, B, $C \in [0,1]$, $B < C$, $B = 0 \Rightarrow A = 0 \wedge C \neq 1$, and $C = 1 \Rightarrow A = 1 \wedge B \neq 0$.

Bias: Parameter Dependent

A, B, C, and the secondary parameter vector $\vec{y}'$ together determine the degree to which $\vec{y}$ is biased by being raised to an associated power: the values of $u(\vec{y}') \in [0, 0.5]$ are linearly mapped onto $[B, B + (C - B)A]$, and the values of $u(\vec{y}') \in [0.5, 1]$ are linearly mapped onto $[B + (C - B)A, C]$.

$$b\_param(y, \vec{y}', A, B, C) = y^{B+(C-B)\left(A-(1-2u(\vec{y}'))|\lfloor 0.5-u(\vec{y}')\rfloor + A|\right)}, \tag{C.5}$$

where $A \in (0,1)$, and $0 < B < C$.

Shift: Linear

$A \in (0,1)$ is the value for which y is mapped to zero.

$$s\_linear(y,A) = \frac{|y-A|}{|\lfloor A-y \rfloor + A|} \tag{C.6}$$

Shift: The deceptive

A is the value at which y is mapped to zero, and the global minimum of the transformation. B is the "aperture" size of the basin leading to the global minimum at A, and C is the value of the deceptive minima (there are always two deceptive minima).

$$\begin{aligned} s\_decept(y,A,B,C) =& 1 + (|y-A| - B) \times \\ & \left(\frac{\lfloor y - A + B \rfloor (1 - C + \frac{A-B}{B})}{A-B} + \frac{\lfloor A + B - y \rfloor (1 - C + \frac{1-A-B}{B})}{1-A-B} + \frac{1}{B} \right), \end{aligned} \tag{C.7}$$

where $A \in (0,1)$, $0 < B \ll 1$, $0 < C \ll 1$, $A - B > 0$, and $A + B < 1$.

Shift: Multi-modal

A controls the number of minima, B controls the magnitude of the "hill sizes" of the multi-modality, and C is the value for which y is mapped to zero. When $B = 0$, $2A+1$ values of y (one at C) are mapped to zero, and when $B \neq 0$, there are $2A$ local minima, and one global minimum at C. Larger values of A and smaller values of B create more difficult problems.

$$s\_multy(y,A,B,C) = \frac{1 + cos\left[(4A+2)\pi\left(0.5 - \frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)\right] + 4B\left(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)^2}{B+2}, \tag{C.8}$$

where $A \in \{1,2,...\}$, $B \geq 0$, $(4A+2)\pi > 4B$, and $C \in (0,1)$.

Reduction: Weighted Sum

By varying the constants of the weight vector $\vec{w}$, optimizers can be forces to treat parameters differntely.

$$r\_sum(\vec{y},\vec{w}) = \frac{\left(\sum_{i=1}^{|\vec{y}|} w_i y_i\right)}{\sum_{i=1}^{|\vec{y}|} w_i}, \tag{C.9}$$

where $|\vec{w}| = |\vec{y}|$, and $w_1,..,w_{|\vec{y}|} > 0$.

Reduction: Non-separable

A controls the degree of non-separability (noting that $r\_nonsep(\vec{y},1) = r\_sum(\vec{y},\{1,...,\})$.

$$r\_non\_sep(\vec{y},A) = \frac{\sum_{j=1}^{|\vec{y}|}\left(y_j + \sum_{k=0}^{A-2}|y_j - y_{1+(j+k) mod |\vec{y}|}\right)}{\frac{|\vec{y}|}{A}\lceil A/2 \rceil (1 + 2A - 2\lceil A/2 \rceil)}, \tag{C.10}$$

where $A \in \{1,...,|\vec{y}|\}$, and $|\vec{y}| mod A = 0$.

WFG1

$$
\begin{aligned}
f_1(\vec{x}) &= x_m + 2\prod_{i=1}^{m-1}(1-cos(x_i\pi/2))\\
f_{j=2:m-1}(\vec{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}(1-cos(x_i\pi/2))\,(1-sin(x_{m-j+1}\pi/2))\right)\\
f_m(\vec{x}) &= x_m + 2m\left(1-x_1-\frac{cos(10\pi x_1+\pi/2)}{10\pi}\right)\\
x_{i=1:m-1} &= r\_sum\left(\{y_{(i-1)k/(m-1)},...,y_{ik/(m-1)}\},\right.\\
&\quad \left.\{2(i-1)k/(m-1)+1,...,2ik/(m-1)\}\right)\\
x_m &= r\_sum(\{y_{k+1},...,y_n\},\{2(k+1),...,2n\})\\
y_{i=1:n} &= b\_poly(y_i',0.02)\\
y_{i:k}' &= y_i''\\
y_{i=k+1:m}' &= b\_flat(y_i'',0.8,0.75,0.85)\\
y_{i=1:k}'' &= z_i/(2i)\\
y_{i=k+1:n}'' &= s\_linear(z_i/(2i),0.35)
\end{aligned}
$$

WFG2

$$
\begin{aligned}
f_1(\vec{x}) &= x_m + 2\prod_{i=1}^{m-1}(1-cos(x_i\pi/2))\\
f_{j=2:m-1}(\vec{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}(1-cos(x_i\pi/2))\,(1-sin(x_{m-j+1}\pi/2))\right)\\
f_m(\vec{x}) &= x_m + 2m\left(1-x_1cos^2(5x_1\pi)\right)\\
x_{i=1:m-1} &= r\_sum\left(\{y_{(i-1)k/(m-1)+1},...,y_{ik/(m-1)}\},\{1,...,1\}\right)\\
x_m &= r\_sum(\{y_{k+1},...,y_n\},\{1,...,1\})\\
y_{i=1:k} &= y_i'\\
y_{i=k+1:k+l/2} &= r\_nonsep\left(\left\{y_{k+2(i-k)-1}',y_{k+2(i-k)}'\right\},2\right)\\
y_{i=1:k}' &= z_i/(2i)\\
y_{i=k+1:n}' &= s\_linear(z_i/(2i),0.35)
\end{aligned}
$$

WFG3

$$
\begin{aligned}
f_1(\vec{x}) &= x_m + 2\prod_{i=1}^{m-1}x_i\\
f_{j=2:m-1}(\vec{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}x_i\right)(1-x_{m-j+1})\\
f_m(\vec{x})) &= x_m + 2m(1-x_1)\\
x_{i=1} &= u_i\\
x_{i=2:m-1} &= x_m(u_i-0.5)+0.5\\
x_m &= r\_sum\left(\{y_{k+1},...,y_{k+l/2}\},\{1,...,1\}\right)\\
u_i &= r\_sum\left(\{y_{(i-1)k/(m-1)+1},...,y_{ik/(m-1)}\},\{1,...,1\}\right)\\
y_{i=1:k} &= y_i'\\
y_{i=k+1:k+l/2} &= r\_nonsep\left(\left\{y_{k+2(i-k)-1}',y_{k+2(i-k)}'\right\},2\right)\\
y_{i=i:k}' &= z_i/(2i)\\
y_{i=k+1:n}' &= s\_linear(z_i/(2i),0.35)
\end{aligned}
$$

WFG4

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

WFG5

$$
\begin{aligned}
f_1(\vec{x}) &= x_m + 2\prod_{i=1}^{m-1}sin(x_i\pi/2)\\
f_{j=2:m-1}(\vec{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}sin(x_i\pi/2)\right)cos(x_{m-j+1}\pi/2)\\
f_m(\vec{x}) &= x_m + (2m)(cos(x_1\pi/2))\\
x_{i=1:m-1} &= r\_sum\left(\{y_{(i-1)k/(m-1)+1},...,y_{ik/(m-1)}\},\{1,...,1\}\right)\\
x_m &= r\_sum(\{y_{k+1},...,y_n\},\{1,...,1\})\\
y_{i=1:n} &= s\_decept(z_i/(2i),0.35,0.001,0.05)
\end{aligned}
$$

WFG6

$$
\begin{aligned}
f_1(\vec{x}) &= x_m + 2\prod_{i=1}^{m-1}sin(x_i\pi/2)\\
f_{j=2:m-1}(\vec{x}) &= x_m + 2j\left(\prod_{i=1}^{m-j}sin(x_i\pi/2)\right)cos(x_{m-j+1}\pi/2)\\
f_m(\vec{x}) &= x_m + (2m)(cos(x_1\pi/2))\\
x_{i=1:m-1} &= r\_nonsep\left(\{y_{(i-1)k/(m-1)+1},...,y_{ik/(m-1)}\},k/(m-1)\right)\\
x_m &= r\_nonsep(\{y_{k+1},...,y_n\},l)\\
y_{i=1:k} &= z_i/(2i)\\
y_{i=k+1:n} &= s\_linear(z_i/(2i),0.35)
\end{aligned}
$$

WFG7

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

WFG8

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

WFG9

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

C.3 Unconstrained Test Problems

Due largely to the nature of Multi-objective Evolutionary Algorithms (MOEAs), their behaviors and performances are mainly studied experimentally. In the last two decades (Deb, 1999; Deb et al., 2002b; Huang et al., 2007; Huband et al., 2006; Li and Zhang, 2008; Okabe et al., 2004; Zitzler et al., 2000), several continuous multi-objective optimization problem test suites have been proposed in the evolutionary community, which have played a crucial role in developing and studying MOEAs. However, to steer the design of algorithms, more test instances are needed to resemble complicated real-life problems. In this section a set of unconstrained test instances that were presented for the CEC09 algorithm contest are described (Zhang et al., 2008). The UF benchmark comprises seven problems with two objectives (UF1-UF7) and three problems with three objectives (UF8-UF10). The authors Li and Zhang (2008) suggest to configure these problems with 30 variables. Note that in all the problems decision variables are grouped as follows.

In cases where the problem is two-objective:

$$\begin{aligned} J_1 =& \{j | j \text{is odd and } 2 \leq j \leq n\}, \\ J_2 =& \{j | j \text{is even and } 2 \leq j \leq n\} \end{aligned} \tag{C.11}$$

In cases where the problem is made up of three objectives:

$$\begin{aligned} J_1 =& \{j | 3 \leq j \leq n, \text{and } j-1 \text{ is a multiplication of } 3\}, \\ J_2 =& \{j | 3 \leq j \leq n, \text{and } j-2 \text{ is a multiplication of } 3\}, \\ J_3 =& \{j | 3 \leq j \leq n, \text{and } j \text{ is a multiplication of } 3\}. \end{aligned} \tag{C.12}$$

Similarly, the Pareto Set in the problems UF1, UF4, and UF7 is defined by setting $0 \leq x_1 \leq 1$ and $x_j = sin\left(6\pi x_1 \frac{j\pi}{n}\right)$ for the remaining variables ($j = 2, .., n$).

UF1

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

- The search space is $[0,1] \times [-1,1]^{-1}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}$, $f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = sin\left(6\pi x_1 \frac{j\pi}{n}\right)$

UF2

$$\begin{aligned} f_1 =& x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} (x_j - g_1(x_j))^2 \\ f_2 =& 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} (x_j - g_2(x_j))^2 \\ g_1(x_j) =& \{0.3x_1^2 cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1\} cos(6\pi x_1 + \frac{j\pi}{n}), \\ g_2(x_j) =& \{0.3x_1^2 cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1\} sin(6\pi x_1 + \frac{j\pi}{n}) \end{aligned}$$

- The search space is $[0,1] \times [-1,1]^{n-1}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}$, $f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = \begin{cases} g_1(x_j) & j \in J_1 \\ g_2(x_j) & j \in J_2 \end{cases}$

UF3

$$\begin{aligned} f_1 =& x_1 + \frac{2}{|J_1|} \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right) + 2 \right) \\ f_2 =& 1 - \sqrt{x_1} + \frac{2}{|J_2|} \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right) + 2 \right) \\ y_j =& x_j - x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})} \end{aligned}$$

- The search space is $[0,1]^n$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}$, $f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = x_1^{0.5(1 + \frac{3(j-2)}{n-2})}$, $j = 2, ..., n$.

UF4

$$\begin{aligned} f_1 =& x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} h(y_j), \\ f_2 =& 1 - x_1^2 + \frac{2}{|J_2|} \sum_{j \in J_2} h(y_j), \\ y_i =& x_j - sin(6\pi x_1 + \frac{j\pi}{n}), \\ h(t) =& \frac{|t|}{1 + e^{2|t|}}. \end{aligned}$$

- The search space is $[0,1] \times [-2,2]^{n-1}$.
- The Pareto Front is $f_2 = 1 - f_1^2$, $f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = sin\left(6\pi x_1 \frac{j\pi}{n}\right)$

UF5

$$f_1 = x_1 + \left(\frac{1}{2N}+\varepsilon\right)|sin(2N\pi x_1)| + \frac{2}{|J_1|}\sum_{j\in J_1} h(y_j),$$
$$f_2 = 1 - x_1 + \left(\frac{1}{2N}+\varepsilon\right)|sin(2N\pi x_1)| + \frac{2}{|J_2|}\sum_{j\in J_2} h(y_j),$$
$$y_j = x_j - sin(6\pi x_1 + \frac{j\pi}{n}),$$
$$h(t) = 2t^2 - cos(4\pi t) + 1.$$

- The search space is $[0,1]\times[-1,1]^{n-1}$.
- The Pareto Front are $2N+1$ points, $\left(\frac{i}{2N}, 1-\frac{i}{2N}\right)$, $N=10$, $\varepsilon=0.1$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = sin\left(6\pi x_1 \frac{j\pi}{n}\right)$, $j=2,..,n$.

UF6

$$f_1 = x_1 + max\left\{0, 2\left(\frac{1}{2N}+\varepsilon\right)sin(2N\pi x_1)\right\} + \frac{2}{J_1}\left(4\sum_{j\in J_1} y_j^2 - 2\prod_{j\in J_1} cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right)+2\right)$$
$$f_2 = 1 - x_1 + max\left\{0, 2\left(\frac{1}{2N}+\varepsilon\right)sin(2N\pi x_1)\right\} + \frac{2}{J_2}\left(4\sum_{j\in J_2} y_j^2 - 2\prod_{j\in J_2} cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right)+2\right),$$
$$y_j = x_j - sin(6\pi x_1 + \frac{j\pi}{n}).$$

- The search space is $[0,1]\times[-1,1]^{n-1}$.
- The Pareto Front are N disconnected parts with $f_2 = 1-f_1$, $f_1 \in \cup_{i=1}^{N}\left(\frac{2i-1}{2N}, \frac{2i}{2N}\right)$ and an isolated point at $(0,1)^T$, $N=2$, $\varepsilon=0.1$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = sin\left(6\pi x_1 \frac{j\pi}{n}\right)$, $j=2,..,n$

UF7

$$f_1 = \sqrt[5]{x_1} + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2$$
$$f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2,$$
$$y_j = x_j - sin(6\pi x_1 + \frac{j\pi}{n}).$$

- The search space is $[0,1]\times[-1,1]^{n-1}$.
- The Pareto Front is $f_2 = 1-f_1$, $f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1]$, $x_j = sin\left(6\pi x_1 \frac{j\pi}{n}\right)$, $j=2,..,n$.

UF8

$$f_1 = cos(0.5x_1\pi)cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1}(x_j - g(x_j))^2,$$
$$f_2 = cos(0.5x_1\pi)sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_2}(x_j - g(x_j))^2,$$
$$f_3 = sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3}(x_j - g(x_j))^2,$$
$$g(x_j) = 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right).$$

- The search space is $[0,1]^2\times[-2,2]^{n-2}$.
- The Pareto Front is $f_1^2+f_2^2+f_3^2 = 1$, $(f_1,f_2,f_3)\in[0,1]$.
- The Pareto set is $x_1, x_2 \in [0,1]$, $x_j = g(x_j)$, $j=3,...,n$.

UF9

$$f_1 = 0.5[max\{0,(1+\varepsilon)(1-4(2x_1-1)^2)\}+2x_1]x_2 + \frac{2}{|J_1|}\sum_{j\in J_1}(x_j-g(x_j))^2,$$
$$f_2 = 0.5[max\{0,(1+\varepsilon)(1-4(2x_1-1)^2)\}-2x_1+2]x_2 + \frac{2}{|J_2|}\sum_{j\in J_2}(x_j-g(x_j))^2,$$
$$f_3 = 1 - x_2 + \frac{2}{|J_3|}\sum_{j\in J_3}(x_j-g(x_j))^2,$$
$$g(x_j) = 2x_2 sin\left(2\pi x_1 + \frac{j\pi}{n}\right).$$

- The search space is $[0,1]^2\times[-2,2]^{n-2}$.
- The Pareto Front is $f_2 = 1-f_1-f_3$ and has two parts:
  - $f_3 \in [0,1] \cup f_1 \in [0, \frac{1}{4}(1-f_3)]$.
  - $f_1 \in [\frac{3}{4}(1-f_3), 1]$.
- The Pareto Set has two disconnected parts $x_1 \in [0,0.25]\cup[0.75,1]$, $x_j = g(x_j)$.

UF10

$$f_1 = cos(0.5x_1\pi)cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1}[4y_j^2 - cos(8\pi y_j)+1]$$
$$f_2 = cos(0.5x_1\pi)sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_1}[4y_j^2 - cos(8\pi y_j)+1]$$
$$f_3 = sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3}[4y_j^2 - cos(8\pi y_j)+1],$$
$$y_j = x_j - 2x_2 sin(2\pi x_1 + \frac{j\pi}{n}).$$

- The search space is $[0,1]^2\times[-2,2]^{n-2}$.
- Its Pareto Front is $f_1^2+f_2^2+f_3^2 = 1$, $(f_1,f_2,f_3)\in[0,1]$.
- Its Pareto Set is $x_1, x_2 \in [0,1]$, $x_j = 2x_2 sin(2\pi x_1 + \frac{j\pi}{n})$, $j=3,...,n$.

C.4 Biased Test Problems

The bias feature is one of the most challenging difficulties that MOEAs might face (Deb, 1999; Huband et al., 2006; Li et al., 2016a). Recently, the BTs test problems were proposed to facilitate the study of the ability of MOEAs for dealing with bias (Li et al., 2016a). In this context bias means that small variations in the decision space around the Pareto Set cause significant changes in the objective space (Huband et al., 2006). Two kinds of biases are included in this set of problems: position-related bias and distance-related bias. The former means that a small change in a position-related variable of a Pareto set solution, implies a significant move along the Pareto front, whereas the the later implies that a small variation in a distance-related variable of one solution in the Pareto set causes a significant changes on the proximity to the Pareto front. One of the aims behind the design of BTs was to propose problems with controllable bias. Particularly, the amount of bias provoked by distance-related variables is modified with the parameter θ, whereas the parameter γ is used to control the bias provoked by position-related variables. Note that as the parameter values decrease, the amount of bias and consequently, the difficulty, increases. The BT benchmark includes eight problems with two objectives (BT1-BT8) and one problem with three objective (BT9). Their authors suggest to configure this problems with 30 variables. The detailed formulations of the nine multi-objective test problems with bias feature are as follows. The two-objective problems (BT1-BT8) the functions I_1 and I_2 are the same.

$$I_k = \{j | mod(j,2) = k-1, \quad j = 2, ..., n\} \tag{C.13}$$

For the three-objective problem (BT9) I_1, I_2, and I_3 are defined by:

$$I_k = \{j | mod(j,3) = k-1, \quad j = 3, ..., n\} \tag{C.14}$$

Position-Related Bias Functions

The following two polynomial transformations are proposed to map all position-related variables x_j from $[0,1]$ to $[0,1]$. Particularly, these transformations are applied to the variables defined in $j \in \{1, ..., m-1\}$, where m is the number of objectives.

$$\begin{aligned} S_1(x_j|\gamma) =& |x_j|^\gamma \\ S_2(x_j|\gamma) =& \begin{cases} \frac{1-(1-4x_j)^\gamma}{4} & x_j \in [0, 0.25), \\ \frac{1+(4x_j-1)^\gamma}{4} & x_j \in [0.25, 0.5), \\ \frac{3-(3-4x_j)^\gamma}{4} & x_j \in [0.5, 0.75), \\ \frac{3+(4x_j-3)^\gamma}{4} & x_j \in [0.75, 1]. \end{cases} \end{aligned} \tag{C.15}$$

Distance-Related Bias Functions

For every component of $\vec{x} = (x_m, ..., x_n)^T$, the following two nonlinear transformations are used to generate the distance-related bias:

$$\begin{aligned} D_1(x_j|\theta) =& x_j^2 + \frac{1 - exp(x_j^2/\theta)}{5}, \\ D_2(x_j|\theta) =& x_j^2 + \frac{|x_j|^\theta}{5}. \end{aligned} \tag{C.16}$$

Note that the minimal values of D_1 and D_2 are zero when $x_j = 0$. In addition, these variables are defined in variables $j \in \{m, ..., n\}$, where n is the number of decision variables.

BT1

$$f_1(\vec{x}) = x_1 + \sum_{j\in I_1} D_1(y_j|10^{-10}),$$
$$f_2(\vec{x}) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_1(y_j|10^{-10}),$$
$$y_j = x_j - sin(j\pi/(2n)).$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = sin(j\pi/(2n))$.

BT2

$$f_1(\vec{x}) = x_1 + \sum_{j\in I_1} D_2\left(y_j|\frac{1}{5}\right),$$
$$f_2(\vec{x}) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_2\left(y_j|\frac{1}{5}\right),$$
$$y_j = x_j - sin(j\pi/(2n)).$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = sin(j\pi/(2n))$.

BT3

$$f_1(\vec{x}) = S_1(x_1|0.02) + \sum_{j\in I_1} D_1(y_j|10^{-8}),$$
$$f_2(\vec{x}) = 1 - \sqrt{S_1(x_1|0.02)} + \sum_{j\in I_2} D_1\left(y_j|10^{-8}\right),$$
$$y_j = x_j - sin(j\pi/(2n)).$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = sin(j\pi/(2n))$.

BT4

$$f_1(\vec{x}) = S_2(x_1|0.06) + \sum_{j\in I_1} D_1\left(y_j|10^{-8}\right),$$
$$f_2(\vec{x}) = 1 - \sqrt{S_2(x_1|0.06)} + \sum_{j\in I_2} D_1\left(y_j|10^{-8}\right),$$
$$y_j = x_j - sin(j\pi/(2n)).$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = sin(j\pi/(2n))$.

BT5

$$f_1(\vec{x}) = x_1 + \sum_{j\in I_1} D_1(y_j|10^{-10}),$$
$$f_2(\vec{x}) = (1-x_1)[1 - x_1 sin(8.5\pi x_1)] + \sum_{j\in I_2} D_1(y_j|10^{-10}),$$
$$y_j = x_j - sin(j\pi/(2n)).$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = (1-f_1)(1-f_1 sin(8.5\pi f_1)), f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = sin(j\pi/(2n))$.

BT6

$$f_1(\vec{x}) = x_1 + \sum_{j\in I_1} D_1(y_j|10^{-4}),$$
$$f_2(\vec{x}) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_1(y_j|10^{-4}),$$
$$y_j = x_j - x_1^{0.5+1.5(j-1)/(n-1)}.$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = 1 - \sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = x_1^{0.5+1.5(j-1)/(n-1)}$.

BT7

$$f_1(\vec{x}) = x_1 + \sum_{j\in I_1} D_1(y_j|10^{-3}),$$
$$f_2(\vec{x}) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_1(y_j|10^{-3}),$$
$$y_j = x_j - sin(6\pi x_1).$$

- The search space is $[0,1]\times[-1,1]^{29}$.
- The Pareto Front is $f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = sin(6\pi x_1)$.

BT8

$$f_1(\vec{x}) = \sum_{j\in I_1} Q\left(D_1(y_j|10^{-3})\right),$$
$$f_3(\vec{x}) = 1 - \sqrt{x_1} + \sum_{j\in I_2} Q\left(D_1(y_j|10^{-3})\right),$$
$$y_j = x_j - x_1^{0.5+1.5(j-1)/(n-1)},$$
$$Q(z) = 4z^2 - cos(8\pi z) + 1.$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]$.
- The Pareto Set is $x_1 \in [0,1], x_j = x_1^{0.5+1.5(j-1)/(n-1)}$.

BT9

$$f_1(\vec{x}) = cos(0.5x_1\pi)cos(0.5x_2\pi) + 10\sum_{j\in I_1} D_1(y_j|10^{-9}),$$
$$f_2(\vec{x}) = cos(0.5x_1\pi)sin(0.5x_2\pi) + 10\sum_{j\in I_1} D_1(y_j|10^{-9}),$$
$$f_3(\vec{x}) = sin(0.5x_1\pi) + 10\sum_{j\in I_3} D_1(y_j|10^{-9}),$$
$$y_j = x_j - sin(j\pi/(2n)).$$

- The search space is $[0,1]^{30}$.
- The Pareto Front is $f_1^2+f_2^2+f_3^2 = 1$, $f_i \in [0,1]$.
- The Pareto Set is $(x_1,x_2) \in [0,1], x_j = sin(j\pi/(2n))$.

C.5 Imbalance Test Problems

Most state-of-the-art algorithms are designed based on the principle of convergence first and diversity second. These algorithms are effective when optimizing balanced problems. However, it has been found recently (Liu et al., 2013) that these algorithms face great difficulties in optimizing imbalance problems. A Multi-objective Optimization Problem (MOP) is defined to be an imbalance problem, if a specific subset (*favored subset*) of its Pareto Front satisfies the following conditions:

- The complexity of the optimization subproblem corresponding to the favored subset is significantly lower than the complexity of the subproblem corresponding to the other part (*unfavored subset*) of the Pareto Front.
- The Pareto Set of the favored subset dominates a significantly larger part of the feasible variable space than the Pareto Set of the unfavored subset.

Note that in this sort of problems, the favored Pareto Front is relatively easy to find and the corresponding favored Pareto Set dominates most of the feasible search space. Following these definitions, *Imbalance Multi-objective Test Suite* (IMB) were proposed (Liu et al., 2016). These problems are similar to the DTLZ test problem construction procedure. The structure of IMB is the following:

$$\begin{aligned} \min f_1(\vec{x}) &= (1+g(\vec{x}_{II}))\Phi_1(\vec{x}_I), \\ \min f_2(\vec{x}) &= (1+g(\vec{x}_{II}))\Phi_2(\vec{x}_I), \\ &\dots \\ \min f_m(\vec{x}) &= (1+g(\vec{x}_{II}))\Phi_m(\vec{x}_I). \end{aligned} \tag{C.17}$$

where m is the dimension of the objective space, $\vec{x}_I = (x_1, x_2, ..., x_{m-1})^T$, $\vec{x}_{II} = (x_m, x_{m+1}, ..., x_n)^T$, and $\vec{x} = (\vec{x}_I, \vec{x}_{II})$. Note that $\vec{x}_I$ and $\vec{x}_{II}$ are the position-related and distance-related parameters, respectively. The parameter n is the dimension of the search space. The function $g(\vec{x}_{II})$ is a positive function having a minimum value of zero. The functions $\Phi_i(\vec{x}_{II})$ are also positive functions and are mutually conflicting. This construction ensures that the Pareto Set occurs for $\vec{x}^*_{II}$ that makes $g(\vec{x}^*_{II}) = 0$. The Pareto Set for x_I occurs from the trade-off among Φ_i conflicting functions. All these test functions are to be minimized by default, and their authors suggest $n = 10$ decision variables for all problem. The domain of all the problems is $x_i \in [0,1] \forall i \in \{1,...,n\}$

IMB1

$$\begin{aligned} f_1(\vec{x}) &= (1+g(\vec{x}))x_1, \\ f_2(\vec{x}) &= (1+g(\vec{x}))(1-\sqrt{x_1}), \\ g(\vec{x}) &= \begin{cases} 0, & 0 \le x_1 \le 0.2, \\ \sum_{i=2}^{n} 0.5(-0.9t_i^2 + |t_i|^{0.6}) & otherwise. \end{cases} \\ t_i &= x_i - sin(0.5\pi x_1). \end{aligned}$$

- The search space is $[0,1]^{10}$.
- The Pareto Front $f_2 = 1 - \sqrt{f_1}$, $f_1 \in [0,1]$.
- The favored subset is $f_1 \in [0,0.2]$, $x_1 \in [0,0.2]$.
- The unfavored subset is $f_1 \in [0.2,1]$, $x_1 \in (0.2,1]$, and $x_i = sin(0.5\pi x_1)$.

IMB2

$$\begin{aligned} f_1(\vec{x}) &= (1+g(\vec{x}))x_1, \\ f_2(\vec{x}) &= (1+g(\vec{x}))(1-x_1), \\ g(\vec{x}) &= \begin{cases} 0, & 0.4 \le x_1 \le 0.6, \\ \sum_{i=2}^{n} 0.5(-0.9t_i^2 + |t_i|^{0.6}) & otherwise. \end{cases} \\ t_i &= x_i - sin(0.5\pi x_1). \end{aligned}$$

- The search space is $[0,1]^{10}$.
- The Pareto Front $f_2 = 1 - f_1$, $f_1 \in [0,1]$.
- The favored subset is $f_1 \in [0.4,0.6]$, $x_1 \in [0.4,0.6]$.
- The unfavored subset is $f_1 \in [0,0.4) \cup (0.6,1]$, $x_1 \in [0,0.4) \cup (0.6,1]$, $x_i = sin(0.5\pi x_1)$.

IMB3

$$\begin{aligned} f_1(\vec{x}) &= (1+g(\vec{x}))cos(\pi x_1/2), \\ f_2(\vec{x}) &= (1+g(\vec{x}))sin(\pi x_1/2), \\ g(\vec{x}) &= \begin{cases} 0, & 0.8 \le x_1 \le 1, \\ \sum_{i=2}^{n} 0.5(-0.9t_i^2 + |t_i|^{0.6}) & otherwise. \end{cases} \\ t_i &= x_i - sin(0.5\pi x_1). \end{aligned}$$

- The search space is $[0,1]^{10}$.
- The Pareto Front $f_1^2 + f_2^2 = 1$, $f_1 \in [0,1]$.
- The favored subset is $f_1 \in [0.8,1]$, $x_1 \in [0.8,1]$.
- The unfavored subset is $f_1 \in [0,0.8)$, $x_1 \in [0,0.8)$, $x_i = sin(0.5\pi x_1)$.

IMB4

$$\begin{aligned} f_1(\vec{x}) &= (1+g(\vec{x}))x_1x_2, \\ f_2(\vec{x}) &= (1+g(\vec{x}))x_1(1-x_2), \\ f_3(\vec{x}) &= (1+g(\vec{x}))(1-x_1), \\ g(\vec{x}) &= \begin{cases} 0, & 2/3 \le x_1 \le 1, \\ 2cos\left(\frac{\pi x_1}{2}\right) \sum_{i=3}^{n}(-0.9t_i^2 + |t_i|^{0.6}) & otherwise \end{cases} \\ t_i &= x_i - (x_1 + x_2)/2. \end{aligned}$$

- The search space is $[0,1]^{10}$.
- The Pareto Front $f_1 + f_2 + f_3 = 1$, $(f_1, f_2, f_3) \in [0,1]$.
- The favored subset is $f_3 \in [0,1/3]$, $x_1 \in [2/3,1]$.
- The unfavored subset is $f_3 \in (1/3,1]$, $x_1 \in [0,2/3)$, $x_i = sin(0.5\pi x_1)$

IMB5

$$
\begin{aligned}
f_1(\vec{x}) &= (1+g(\vec{x}))cos(\pi x_1/2)cos(\pi x_2/2),\\
f_2(\vec{x}) &= (1+g(\vec{x}))cos(\pi x_1/2)sin(\pi x_2/2),\\
f_3(\vec{x}) &= (1+g(\vec{x}))sin(\pi x_1/2),\\
g(\vec{x}) &= \begin{cases} 0, & 0 \le x_1 \le 0.5,\\ 2cos\left(\frac{\pi x_1}{2}\right)\sum_{i=3}^{n}(-0.9t_i^2+|t_i|^{0.6}) & otherwise. \end{cases}\\
t_i &= x_i-(x_1+x_2)/2.
\end{aligned}
$$

- The search space is $[0,1]^{10}$.
- The Pareto Front is $f_1^2+f_2^2+f_3^2=1$, $(f_1,f_2,f_3)\in[0,1]$.
- The favored subset is $f_3\in[0,\sqrt{2}/2]$, $x_1\in[0,0.5]$, and the point at $(0,0,1)^T$.
- The unfavored subset is $f_3\in(\sqrt{2}/2,1]$, $x_1\in(0.5,1]$, $x_i=(x_1+x_2)/2$

IMB6

$$
\begin{aligned}
f_1(\vec{x}) &= (1+g(\vec{x}))x_1x_2,\\
f_2(\vec{x}) &= (1+g(\vec{x}))x_1(1-x_2),\\
f_3(\vec{x}) &= (1+g(\vec{x}))(1-x_1),\\
g(\vec{x}) &= \begin{cases} 0, & 0 \le x_1 \le 0.75,\\ 2cos\left(\frac{\pi x_1}{2}\right)\sum_{i=3}^{n}(-0.9t_i^2+|t_i|^{0.6}) & otherwise. \end{cases}\\
t_i &= x_i-(x_1+x_2)/2.
\end{aligned}
$$

- The search space is $[0,1]^{10}$.
- The Pareto Front is $f_1+f_2+f_3=1$, $(f_1,f_2,f_3)\in[0,1]$.
- The favored subset is $f_3\in[0,0.75]$, $x_1\in[0,0.75]$ and the point $(0,0,1)^T$.
- The unfavored subset is $f_3\in(0.75,1]$, $x_1\in(0.75,1]$, $x_j=(x_1+x_2)/2$.

IMB7

$$
\begin{aligned}
f_1(\vec{x}) &= (1+g(\vec{x}))x_1,\\
f_2(\vec{x}) &= (1+g(\vec{x}))(1-\sqrt{x_1}),\\
g(\vec{x}) &= \begin{cases} \sum_{i=2}^{n}(-0.9s_i^2+|s_i|^{0.6}), & 0.5 \le x_1 \le 0.8,\\ \sum_{i=2}^{n}|t_i|^{0.6} & otherwise. \end{cases}\\
s_i &= x_i-sin(0.5\pi x_1)\\
t_i &= x_i-0.5.
\end{aligned}
$$

- The search space is $[0,1]^{10}$.
- The Pareto Front is $f_2=1-\sqrt{f_1}$, $f_1\in[0,1]$.
- The favored subset is $f_1\in[0,0.5)\cup(0.8,1]$, $x_1\in[0,0.5)\cup(0.8,1]$, $x_j=0.5$
- The unfavored subset is $f_1\in[0.5,0.8]$, $x_1\in[0.5,0.8]$, $x_j=sin(0.5\pi x_1)$.

IMB8

$$
\begin{aligned}
f_1(\vec{x}) &= (1+g(\vec{x}))x_1,\\
f_2(\vec{x}) &= (1+g(\vec{x}))(1-x_1),\\
g(\vec{x}) &= \begin{cases} \sum_{i=2}^{n}(-0.9s_i^2+|s_i|^{0.6}), & 0.5 \le x_1 \le 0.8,\\ \sum_{i=2}^{n}|t_i|^{0.6} & otherwise. \end{cases}\\
s_i &= x_i-sin(0.5\pi x_1),\\
t_i &= x_i-0.5.
\end{aligned}
$$

- The search space is $[0,1]^{10}$.
- The Pareto Front is $f_2=1-f_1$, $f_1\in[0,1]$.
- The favored subset is $f_1\in[0,0.5)\cup(0.8,1]$, $x_1\in[0,0.5)\cup(0.8,1]$, $x_j=0.5$
- The unfavored subset is $f_1\in[0.5,0.8]$, $x_1\in[0.5,0.8]$, $x_j=sin(0.5\pi x_1)$.

IMB9

$$
\begin{aligned}
f_1(\vec{x}) &= (1+g(\vec{x}))cos(\pi x_1/2),\\
f_2(\vec{x}) &= (1+g(\vec{x}))sin(\pi x_1/2),\\
g(\vec{x}) &= \begin{cases} \sum_{i=2}^{n}(-0.9s_i^2+|s_i|^{0.6}), & 0.5 \le x_1 \le 0.8,\\ \sum_{i=2}^{n}|t_i|^{0.6} & otherwise. \end{cases}\\
s_i &= x_i-sin(0.5\pi x_1),\\
t_i &= x_i-0.5.
\end{aligned}
$$

- The search space is $[0,1]^{10}$.
- The Pareto Front is $f_2=\sqrt{1-f_1}$, $f_1\in[0,1]$.
- The favored subset is $f_1\in[0,0.309)\cup(0.707,1]$, $x_1\in[0,0.5)\cup(0.8,1]$, $x_j=0.5$.
- The unfavored subset is $f_1\in[0.309,0.707]$, $x_1\in[0.5,0.8]$, $x_j=sin(0.5\pi x_1)$.

IMB10

$$
\begin{aligned}
f_1(\vec{x}) &= (1+g(\vec{x}))x_1x_2,\\
f_2(\vec{x}) &= (1+g(\vec{x}))x_1(1-x_2),\\
f_3(\vec{x}) &= (1+g(\vec{x}))(1-x_1),\\
g(\vec{x}) &= \begin{cases} \sum_{i=3}^{n}2(-0.9s_i^2+|s_i|^{0.6}), & 0.2 \le (x_1,x_2) \le 0.8,\\ \sum_{i=3}^{n}|t_i|^{0.6} & otherwise. \end{cases}\\
s_i &= x_i-(x_1+x_2)/2,\\
t_i &= x_i-x_1x_2.
\end{aligned}
$$

- The search space is $[0,1]^{10}$.
- The Pareto Front is $f_1+f_2+f_3=1$, $(f_1,f_2,f_3)\in[0,1]$.
- The favored subset is $f_1\in[0.04,0.64]$, $f_3\in[0.2,0.8]$, $(x_1,x_2)\in[0,0.2)\cup(0.8,1]$, $x_j=(x_1+x_2)/2$.
- The unfavored subset is $f_1\in[0,0.04)\cup(0.64,1]$, $f_3\in[0,0.2)\cup(0.8,1]$, $x_1\in[0.2,0.2)\cup(0.8,1]$, $x_j=sin(0.5\pi x_1)$.

Appendix D

Supplementary Section for "VSD-MOEA: A Dominance-Based Multi-Objective Evolutionary Algorithm with Explicit Variable Space Diversity Management"

This section contains supplementary materials which aim to provide a clearer understanding of the specifics of VSD-MOEA (Chapter 5). Section D.1 describes multimedia material that provides a visualization of the internal behavior of VSD-MOEA in comparison to other state-of-the-art schemes. In Section D.2, some of the results described in the Chapter 5 are analyzed in terms of the Modified Inverted Generational Distance (IGD+), with the conclusions being quite similar to those obtained in the Chapter 5 with the hypervolume (HV). The importance of decay models for promoting diversity in VSD-MOEA is analyzed in Section D.3. Then, Section D.4 provides an analysis of the importance of the final moment of diversity promotion. Finally, Section D.5 provides a comparison of MOEAs with $R2$-based passive archives. Similar conclusions are drawn with or without passive archives. Note that, as in Chapter 5, each experiment was repeated 35 times.

D.1 Multimedia Material

This section provides a description of the video included as Supplementary Material[1]. The main aim of this video is to show the large differences in the way in which VSD-MOEA and state-of-the-art MOEAs explore the search space. The video shows a single execution of VSD-MOEA, CPDEA and R2-EMOA to solve WFG5. Note that MOEA/D was also executed and its behavior was similar to that of R2-EMOA. Thus, and in order to avoid saturating the video, it was not included. In order to allow a proper visualization, WFG5 was configured with two decision variables, which means that one distance parameter and one position parameter are used. The main peculiarity of WFG5 is its deceptiveness (Huband et al., 2005, 2006). In this configuration, a solution $\vec{\mathbf{x}}$ is part of the Pareto set when $x_2 = 1.4$. However, most of the descent directions point towards two local optimal fronts, which appear when $x_2 = 0$ or $x_2 = 4$. The stopping criterion was set to $100,000$ function evaluations, and the remaining parameters were set as in the main document. It is worth noting that VSD-MOEA explicitly promotes diversity in the decision variable space until the halfway point of the execution.

The video is divided into two sides. The left-side represents the objective space and the right-side represents the decision variable space. In the decision variable space, each local optimal region is highlighted with a horizontal blue line, and the global optimal region is highlighted with a horizontal red line. The video shows that after just a few function evaluations, R2-EMOA has converged prematurely to the local optimal regions. In the case of CPDEA, at approximately 5% of the total number of function evaluations, most of the solutions are located in the global optimal region. Thus, promoting diversity in the decision variable space is helpful for locating the proper region. Finally, note that at 5% of the total number of function evaluations, VSD-MOEA has located few individuals in the global optimal region; however, it keeps exploring in an effort to find better regions. Later, at approximately 40% of the total number of function evaluations, the number of individuals in the global optimal region starts to increase and by the midpoint of the execution, most of the solutions are located in the global optimal region. Thus, the results obtained by R2-EMOA are clearly outperformed by diversity-aware MOEAs. Note that in these conditions, it might appear that the search performed by CPDEA is better than the one performed by VSD-MOEA, because CPDEA found the optimal region faster. However, since it converged relatively soon, the search performed by VSD-MOEA is preferred when the problem is more complex (i.e., with more decision variables). In fact, Chapter 5 and subsequent sections show that, in terms of HV and IGD+, VSD-MOEA outperforms CPDEA for WFG5 with a larger number of decision variables.

Finally, it is worth mentioning that both R2-EMOA and VSD-MOEA place a large number of points in the knee of the Pareto front, at the expense of leaving some holes near the boundaries. This is because both the $R2$ metric and the objective space density estimator used in VSD-MOEA prefer these regions. To underscore the

[1]Alternatively, the video can be accessed at https://youtu.be/ZpaV1Wc5LuE

Table D.1 Summary of the IGD+ results attained for problems with two objectives. The lower the IGD+ value, the better the algorithm.

	VSD-MOEA			CPDEA			MOEA/D			R2-EMOA		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
WFG1	**0.009**	0.008	0.004	0.047	0.043	0.017	0.009	0.009	0.002	0.032	0.017	0.032
WFG2	**0.014**	0.004	0.040	0.021	0.006	0.048	0.177	0.177	0.000	0.172	0.177	0.028
WFG3	**0.009**	0.009	0.000	0.027	0.027	0.002	0.010	0.010	0.000	0.011	0.011	0.000
WFG4	0.007	0.007	0.000	0.024	0.024	0.002	0.009	0.009	0.000	**0.007**	0.007	0.000
WFG5	**0.069**	0.069	0.002	0.081	0.081	0.001	0.071	0.071	0.002	**0.069**	0.068	0.002
WFG6	0.070	0.071	0.007	0.132	0.132	0.001	**0.048**	0.048	0.016	**0.050**	0.049	0.009
WFG7	0.007	0.007	0.000	0.019	0.018	0.002	0.009	0.009	0.000	**0.007**	0.007	0.000
WFG8	**0.072**	0.050	0.031	0.081	0.078	0.018	0.129	0.129	0.004	0.134	0.133	0.003
WFG9	**0.032**	0.026	0.015	0.130	0.130	0.002	**0.053**	0.029	0.044	**0.066**	0.028	0.050
DTLZ1	0.002	0.002	0.000	0.002	0.002	0.000	**0.001**	0.001	0.000	0.002	0.002	0.000
DTLZ2	0.003	0.003	0.000	0.004	0.004	0.000	0.003	0.003	0.000	**0.002**	0.002	0.000
DTLZ3	0.003	0.003	0.000	0.003	0.003	0.000	0.003	0.003	0.000	**0.002**	0.002	0.000
DTLZ4	**0.003**	0.003	0.000	0.005	0.005	0.001	0.003	0.003	0.000	0.205	0.002	0.235
DTLZ5	0.003	0.003	0.000	0.004	0.004	0.000	0.003	0.003	0.000	**0.002**	0.002	0.000
DTLZ6	**0.003**	0.003	0.000	0.048	0.044	0.022	0.003	0.003	0.000	0.080	0.084	0.023
DTLZ7	0.003	0.003	0.000	0.004	0.004	0.000	0.004	0.004	0.000	**0.002**	0.002	0.000
UF1	**0.010**	0.010	0.003	0.013	0.013	0.002	0.010	0.009	0.003	0.061	0.062	0.011
UF2	0.013	0.012	0.005	0.017	0.017	0.001	**0.009**	0.008	0.003	0.016	0.016	0.003
UF3	**0.062**	0.060	0.008	0.124	0.123	0.030	0.209	0.212	0.040	0.241	0.237	0.024
UF4	0.034	0.034	0.001	0.047	0.047	0.001	0.034	0.034	0.001	**0.031**	0.031	0.001
UF5	**0.173**	0.174	0.018	**0.154**	0.160	0.036	0.444	0.426	0.251	0.368	0.299	0.155
UF6	**0.057**	0.048	0.031	**0.099**	0.074	0.055	0.354	0.369	0.161	0.339	0.363	0.130
UF7	**0.010**	0.010	0.002	0.014	0.013	0.002	0.027	0.012	0.084	0.173	0.037	0.161
Mean	**0.029**	0.027	0.007	0.048	0.046	0.011	0.071	0.069	0.027	0.090	0.071	0.038

Table D.2 Statistical Tests and Deterioration Level of the IGD+ for problems with two objectives

	↑	↓	↔	**Score**	**Deterioration**
VSD-MOEA	49	11	9	38	0.050
CPDEA	16	50	3	-34	0.483
MOEA/D	27	33	9	-6	1.012
R2-EMOA	31	29	9	2	1.070

superiority of VSD-MOEA and CPDEA for WFG5, note that most of the solutions obtained by R2-EMOA are dominated by those solutions located by diversity-aware MOEAs. The main feature of VSD-MOEA, which is to delay convergence, is clearly shown in this video.

D.2 Comparison Against State-of-the-art MOEAs for Long Runs

This section presents the results obtained by VSD-MOEA and state-of-the-art schemes in terms of IGD+(Ishibuchi et al., 2015). The same parameterization is used as in Section 5.3, meaning that the stopping criterion was set to 2.5×10^6 function evaluations. The structure of the tables is the same as in Chapter 5. Thus, the only modification is that instead of using the hypervolume, IGD+ is adopted in this case.

Table D.1 shows the IGD+ values obtained for the benchmark functions with two objectives. Specifically, the mean, median and standard deviation of the IGD+ values are provided for each method and test problem considered. The last row shows the results considering all the test problems. For each test problem, the data for the method that yielded the lowest mean are shown in **boldface**. Furthermore, all methods that were not statistically inferior to the method that yielded the lowest mean are shown in **boldface**. From here on, the

Table D.3 Statistical Tests and Deterioration Level of the IGD+ for problems with three objectives

	↑	↓	↔	**Score**	**Deterioration**
VSD-MOEA	49	8	0	41	0.308
CPDEA	3	45	9	-42	2.696
MOEA/D	20	29	8	-9	0.891
R2-EMOA	29	19	9	10	1.129

methods shown in **boldface** for a given problem are referred to as the winning methods. Based on the number of functions where each method is in the group of winning methods for cases with two objectives, the best methods are VSD-MOEA and R2-EMOA with 13 and 10 wins, respectively. Thus, VSD-MOEA is the most competitive method in terms of this measure. More impressive is the fact that the mean IGD+ achieved by VSD-MOEA, when all problems are considered simultaneously, is much lower than that achieved by R2-EMOA. In fact, the total means of CPDEA, MOEA/D, and R2-EMOA are 0.048, 0.071, and 0.090, respectively. By contrast, VSD-MOEA yielded a much lower value of 0.029. When the data are carefully inspected, it is clear that in the cases where VSD-MOEA loses, the difference with respect to the best method is not very large. For instance, the difference between the mean IGD+ achieved by VSD-MOEA and by the best method was never greater than 0.05. However, all the other methods exhibited a deterioration greater than 0.05 in several cases. Specifically, it occurred in 3, 5 and 7 problems for CPDEA, MOEA/D and R2-EMOA, respectively. This means that even if VSD-MOEA loses in some cases, its deterioration is always small, exhibiting a much more robust behavior than any other method. Exactly the same situation appeared when analyzing the data in terms of hypervolume.

In order to better clarify these findings, pair-wise statistical tests were performed between each method tested in each test problem. Table D.2 shows the results, with the same meaning as in Section 5.3. The calculated data confirm that although VSD-MOEA loses in some cases, the overall numbers of wins and losses favor VSD-MOEA. More importantly, the total deterioration is considerably lower in the case of VSD-MOEA, confirming that when VSD-MOEA loses, the deterioration is not very high.

Tables D.3 and D.4 show the same information for problems with three objectives. Taking into account the mean of all the test problems, VSD-MOEA again yielded a much lower mean IGD+ than the other methods. Specifically, VSD-MOEA obtained a value of 0.087, whereas the second-ranked algorithm (MOEA/D) obtained a value of 0.113. As with HV, the difference between the mean IGD+ obtained by VSD-MOEA and by the best method was larger than 0.05 in only one problem (WFG1). All remaining methods exhibited a deterioration greater than 0.05 in several cases. In particular, this happened in 5, 7 and 12 problems for MOEA/D, R2-EMOA and CPDEA, respectively. Moreover, VSD-MOEA is considerably superior to the other methods not only in terms of total deterioration, but also in terms of total wins and losses. VSD-MOEA was in the group of winning methods for 13 out of 19 test problems, while the second-best-ranked algorithm (R2-EMOA) was in the group of winning methods for only 4 test problems. These conclusions are again quite similar to those drawn for the hypervolume in the main document.

D.3 On the Effect of Diversity Decay Models

VSD-MOEA uses a linear decay model for the promotion of diversity in the decision space, i.e., for setting the diversity penalty threshold (D). In the optimization field, there are several methods that also require some decay models for their parameters. For instance, several cooling schemes have been tested with simulated annealing (Bertsimas and Tsitsiklis, 1993). This section analyzes the impact on performance of different models. Specifically, the Linear (Eqn. D.1), Geometric (Eqn. D.2), Exponential (Eqn. D.3) and Logarithmic (Eqn. D.4)

Table D.4 Summary of the IGD+ results attained for problems with three objectives. The lower the IGD+ value, the better the algorithm.

	VSD-MOEA			CPDEA			MOEA/D			R2-EMOA		
	Mean	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**	**Mean**	**Median**	**Std**
WFG1	0.374	0.368	0.030	1.129	1.141	0.106	**0.090**	0.089	0.004	0.160	0.160	0.013
WFG2	**0.046**	0.045	0.005	0.163	0.164	0.015	0.088	0.077	0.044	0.185	0.120	0.079
WFG3	0.040	0.040	0.000	0.212	0.216	0.035	**0.030**	0.030	0.000	0.033	0.034	0.000
WFG4	**0.106**	0.106	0.001	0.235	0.232	0.016	0.158	0.158	0.002	0.119	0.120	0.001
WFG5	**0.162**	0.162	0.002	0.201	0.202	0.005	0.206	0.206	0.001	0.169	0.169	0.002
WFG6	0.169	0.171	0.006	0.249	0.249	0.007	0.197	0.197	0.009	**0.155**	0.155	0.006
WFG7	**0.107**	0.107	0.001	0.281	0.283	0.021	0.157	0.157	0.000	0.120	0.120	0.002
WFG8	**0.150**	0.147	0.024	0.348	0.340	0.034	0.222	0.222	0.001	0.182	0.182	0.002
WFG9	**0.162**	0.158	0.035	0.261	0.262	0.006	0.186	0.164	0.039	0.192	0.219	0.041
DTLZ1	0.015	0.015	0.000	0.017	0.017	0.000	0.016	0.016	0.000	**0.015**	0.015	0.000
DTLZ2	0.028	0.028	0.000	0.051	0.051	0.006	0.033	0.033	0.000	**0.027**	0.027	0.000
DTLZ3	0.028	0.028	0.000	0.105	0.032	0.226	0.033	0.033	0.000	**0.027**	0.027	0.000
DTLZ4	**0.028**	0.028	0.000	0.054	0.054	0.003	0.033	0.033	0.000	0.279	0.257	0.238
DTLZ5	**0.003**	0.003	0.000	0.005	0.005	0.000	0.005	0.005	0.000	0.003	0.003	0.000
DTLZ6	**0.003**	0.003	0.000	0.064	0.068	0.022	0.005	0.005	0.000	0.044	0.047	0.019
DTLZ7	**0.031**	0.032	0.001	0.048	0.047	0.003	0.053	0.053	0.000	0.107	0.109	0.007
UF8	**0.037**	0.035	0.006	0.100	0.097	0.014	0.089	0.090	0.002	0.075	0.043	0.068
UF9	**0.028**	0.027	0.003	0.145	0.135	0.036	0.157	0.212	0.078	0.136	0.207	0.086
UF10	**0.139**	0.138	0.031	0.451	0.436	0.087	0.395	0.443	0.119	0.411	0.448	0.115
Mean	0.087	0.086	0.008	0.217	0.212	0.034	0.113	0.117	0.016	0.128	0.130	0.036

models are considered (Banchs, 1997; Mahdi et al., 2017; Nourani and Andresen, 1998), where r corresponds to the ratio of the elapsed period, i.e., $\frac{G_{Elapsed}}{G_{End}}$. The graph in Figure D.1 shows the decay model configurations that are tested in this section.

$$D(r) = max\left(0, ITV - \frac{ITV \times r}{0.5}\right) \tag{D.1}$$

$$D(r) = ITV \times \alpha^{r} \tag{D.2}$$

$$D(r) = ITV \times e^{-6r^{\beta}} \tag{D.3}$$

$$D(r) = \frac{ITV \times 0.05}{log_2(1+r)} + \beta \tag{D.4}$$

Tables D.5 and D.6 show the HV ratio obtained for the benchmark functions with two and three objectives, respectively. Specifically, the mean and median of the HV ratio is shown for all the problems. For the two-objective case, the exponential model with $\beta = 1.5$ achieved the highest mean with a value of 0.945. However, the results attained by the linear, geometric and exponential models are quite similar, ranging between 0.937 and 0.945. The worst results are those achieved by the logarithmic model, which decreases the level of diversity very early in the optimization process. In the three-objective case, the linear model reached the largest mean. Once again, the results achieved by the linear, geometric and exponential models are quite similar, ranging between 0.894 and 0.902. The logarithmic method also yielded inferior results in the three-objective problems. Finally, it is also important to note that the worst results achieved by VSD-MOEA with any of the decay models, both in the two-objective and three-objective cases, are better than the second-ranked state-of-the-art algorithm (see Tables 5.3 and 5.4). Thus, while setting a proper decay model is important, all these different kinds of diversity promotion were successful in further improving the results attained by the state-of-the-art. Moreover,

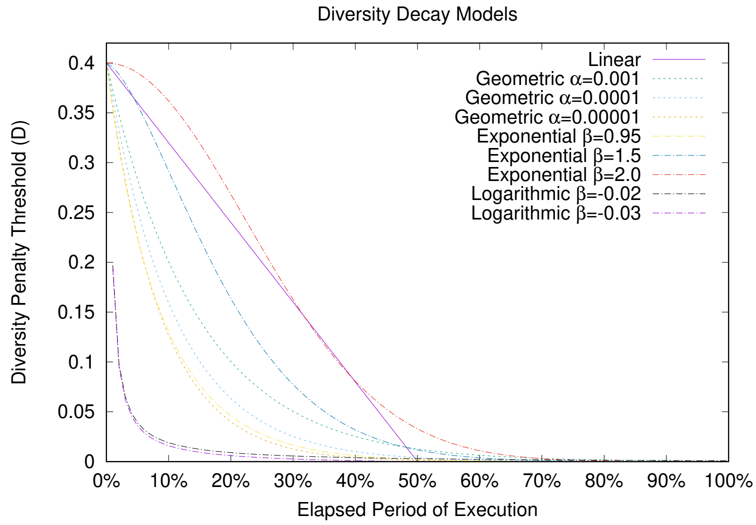

Fig. D.1 Diversity decay models tested in the replacement phase of VSD-MOEA.

Table D.5 Summary of the hypervolme ratio results attained with different decay models for the problems with two objectives

	LINEAR	**GEOMETRIC**				**EXPONENTIAL**			**LOGARITHMIC**	
		$\alpha = 1\times10^{-2}$	$\alpha = 1\times10^{-3}$	$\alpha = 1\times10^{-4}$	$\alpha = 1\times10^{-5}$	$\beta = 0.95$	$\beta = 1.5$	$\beta = 2.0$	$\beta = -0.02$	$\beta = -0.03$
Mean	0.943	0.943	0.941	0.939	0.937	0.942	**0.945**	0.944	0.918	0.918
Median	0.945	0.943	0.940	0.941	0.938	0.942	0.945	0.947	0.916	0.916

VSD-MOEA is quite robust in the sense that only those decay models that quickly decrease diversity promotion cause a relatively significant deterioration in performance.

D.4 Analysis of the Final Moment of Diversity Promotion

In Chapter 5, the decay model applied in VSD-MOEA was the linear one (Eqn. D.5), configured with the Final Moment of Diversity Promotion ($FMDP$) set to 0.5, meaning that the diversity in the decision variable space is explicitly promoted only in the first half of the algorithm’s run. The $FMDP$ parameter is used to specify the final moment of diversity promotion. Specifically, when r is larger than $FMDP$, diversity in the decision variable space is not promoted.

$$D(r) := max\left(0, ITV - ITV \times \frac{r}{FMDP}\right) \tag{D.5}$$

In order to gain a better insight into the effect of the final moment of diversity promotion, several values of $FMDP$ with a fixed value $ITV = 0.4$ were tested. Specifically, $FMDP = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ were considered both for the two-objective and three-objective cases. The remaining parameters were set as in the main document and previous sections. Figure D.2 shows, for each value $FMDP$, the box-plots of the resulting HV ratio. In the two-objective case, VSD-MOEA performs quite well with any $FMDP$ in the range $[0.5, 1.0]$. The worst results were achieved with $FMDP = 0.1$, meaning that reducing diversity quickly is not adequate. However, even in this case, the median value is higher than the one attained by the second-ranked algorithm in Table 3 of the main document. There is also a slight decrease in performance when setting

Table D.6 Summary of the hypervolme ratio results attained with different decay models for the problems with three objectives

	LINEAR	**GEOMETRIC**				**EXPONENTIAL**			**LOGARITHMIC**	
		$\alpha = 1\times10^{-2}$	$\alpha = 1\times10^{-3}$	$\alpha = 1\times10^{-4}$	1×10^{-5}	$\beta = 0.95$	$\beta = 1.5$	$\beta = 2.0$	$\beta = -0.02$	$\beta = -0.03$
Mean	**0.902**	0.895	0.899	0.899	0.898	0.898	0.899	0.894	0.877	0.877
Median	0.901	0.895	0.899	0.899	0.898	0.899	0.901	0.894	0.877	0.873

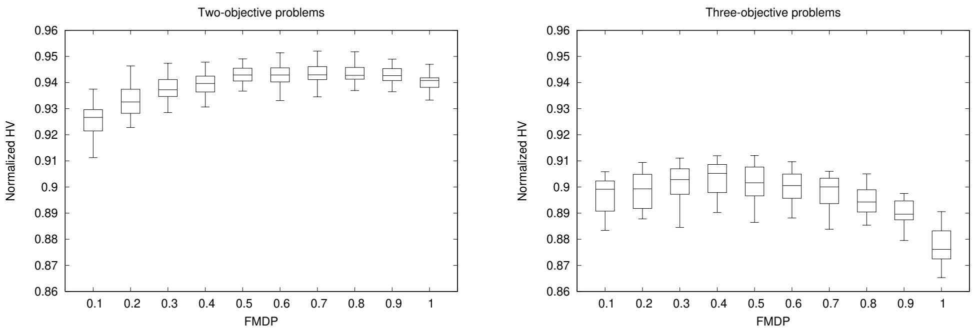

Fig. D.2 Box-plots of the HV ratio for 35 runs for the two-objective (left side) and three-objective (right side) problems, considering different final moments of diversity promotion

$FMDP = 1$. The reason is that executing for a period without promoting diversity is important to increase the amount of intensification.

In the three-objective case, the performance is more sensitive to the $FMDP$ setting. Using $FMDP$ values larger than 0.4 deteriorates HV, and this deterioration is especially important for $FMDP = 1$. Therefore, the explicit promotion of diversity should be carried out for a shorter period of time, with the aim of increasing the period devoted to intensification. Particularly, any value in the range 0.2 to 0.6 offers acceptable performance. Note also that any $FMDP$ yielded larger HV medians than any of the state-of-the-art methods reported in Table 5.5, which highlights the importance of properly promoting diversity in decision variable space.

D.5 Effect of Passive Archives

As explained in the Chapter 5, official codes were used in our experimental validation. This means that in every case, except in CPDEA, the set of results was taken from the last population. By contrast, in the case of CPDEA, an archive is used that simultaneously considers diversity in the objective and decision variable space. Since this might affect the results, in this section, a bounded passive archive was incorporated into all the methods. Then, the solution is taken from the archive. As in Chapter 5, the solution set is restricted to 100 elements. After any solution is evaluated, the solution is inserted in the archive if it is not dominated by any of its members. Additionally, any dominated solutions are erased from the archive. Then, if there are 101 remaining solutions, one is selected to be erased. Particularly, the solution with a lower effect on the $R2$-indicator is selected. Note that the $R2$-indicator is configured similarly as in R2-EMOA.

Regardless of the incorporation of the archive, the remaining configuration is the same as in the main document and previous sections. Tables D.7 and D.8 show some statistics for the two-objective and three-objective cases. Information with the same meaning as in the main document is shown. Particularly, the data include the mean, median, wins, loses, ties, score and deterioration for each algorithm. The benefits obtained

Table D.7 Mean, Median, Statistical Tests and Deterioration Level of the HV for problems with two objectives, considering a passive archive

	Mean	Median	↑	↓	↔	Score	Deterioration
VSD-MOEA	0.944	0.946	**55**	11	3	44	0.133
CPDEA	0.919	0.921	27	35	7	-8	0.719
MOEA/D	0.898	0.897	20	41	8	-21	1.279
R2-EMOA	0.855	0.880	23	38	8	-15	1.657

Table D.8 Mean, Median, Statistical Tests and Deterioration Level of the HV for problems with three objectives, considering a passive archive

	Mean	Median	↑	↓	↔	Score	Deterioration
VSD-MOEA	0.909	0.908	48	4	5	44	0.224
CPDEA	0.832	0.837	18	31	8	-13	1.572
MOEA/D	0.872	0.869	16	33	8	-17	0.975
R2-EMOA	0.840	0.833	17	31	9	-14	1.652

by incorporating the archive are not significant. As expected, the only method where the benefits are more noticeable is CPDEA. The reason is that, in its original version, it tries to simultaneously maximize diversity in the objective space and decision variable space. In the case of VSD-MOEA, MOEA/D and R2-EMOA, the differences are minimal. Since the statistics in terms of score and deterioration are also quite similar as in Chapter 5 (without archive), similar conclusions are drawn with or without the incorporation of archives; specifically, the advantages of VSD-MOEA are remarkable.

Appendix E

Supplementary Section for “AVSD-MOEA/D: Archived Variable Space Diversity Based on Decomposition”

E.1 Performance of MOEAs in long-term executions

This section presents the supplementary material of Chapter 6. Particularly, the material of this section complements the results presented in the subsection 6.3.1, by discussing the results in terms of the *Modified Inverted Generational Distance* (IGD+). The conclusions found from these additional analyses are similar to those obtained in Chapter 6.

In the following, we present a set of tables analogous to those presented in the main document, with the only difference that the hypervolume ratio is substituted for the indicator IGD+. Since the indicator IGD+ considers distances between a high-quality reference set and the approximations, the aim is to minimize the attained value. In order to generate high-quality reference sets, the information given in Ishibuchi et al. (2018); Li et al. (2016a); Talke et al. (2012); Tian et al. (2018); Zhang et al. (2008) was taken into account. In order to facilitate future comparisons, the reference fronts are publicly available[1].

Table E.1 shows the IGD+ values for the benchmark functions with two objectives. For each method and problem, the best, mean, and standard deviation are reported. In order to summarize the results obtained by each method, the last row reports the mean for the whole set of problems. Furthermore, for each problem, the method that produced the lowest mean and those that were not statistically inferior to the best are shown in **boldface**. Note that the same statistical procedure as in the main document was used. In the following, for each problem, the methods shown in **boldface** are referred to as the winning methods. AVSD-MOEA/D, R2-EMOA, MOEA/D-DE, NSGA-III, and NSGA-II belonged to the winning methods in 20, 5, 2, 2 and 0 problems, respectively. Particularly, AVSD-MOEA/D attained a mean value equal to 0.014, while the remaining methods attained values about three times larger, i.e. their values are between 0.036 and 0.044. Analyzing the results for all problems in which AVSD-MOEA/D does not belong to the winning methods, the maximum difference to the best method is never greater than 0.05. However, this amount of deterioration appeared in 4, 4, 5 and 5 problems for MOEA/D-DE, NSGA-III, R2-EMOA and NSGA-II, respectively. Thus, the advantages of AVSD-MOEA/D are clear.

It is also notable that the worst approximations attained by AVSD-MOEA/D appeared in WFG6 and UF3, and in those cases, the values of IGD + are equal to 0.040 and 0.037, respectively, which means that they are adequate approximations. In contrast, the worst values attained by the remaining methods are 0.146, 0.202, 0.220 and 0.251 corresponding to NSGA-III (UF5), NSGA-II (UF6), R2-EMOA (UF6) and MOEA/D-DE (UF5), respectively. Thus, AVSD-MOEA/D attains the best approximations in most cases and when it does not, the difference with respect to other schemes is always small.

In order to better clarify these findings, pair-wise statistical tests were applied between each method tested in each benchmark problem. For the two-objective cases, Table E.2 shows the number of times that each method statistically won (column ↑), lost (column ↓) or tied (column ↔). The **Score** column shows the difference between the number of times that each method won and the number of times that each method lost. Additionally, for each method μ, we calculated the sum of the differences between the mean IGD+ attained by μ and the best method (the ones with the lowest mean), for each problem where μ was not in the group of winning methods. This value is shown in the *Deterioration* column. The data confirm that although AVSD-MOEA/D loses in some pair-wise tests, the overall numbers of wins and losses clearly favor AVSD-MOEA/D. More importantly, the total deterioration is much lower in the case of AVSD-MOEA/D, confirming that when AVSD-MOEA/D loses, the differences are low.

[1] https://drive.google.com/drive/folders/1760kQ6OVU6NHNB_BiNPmPZ_xbx40gEft?usp=sharing.

Table E.1 Summary of the IGD+ attained for problems with two objectives

	AVSD-MOEA/D			MOEA/D-DE			NSGA-II			NSGA-III			R2-EMOA		
	Best	**Mean**	**Std**	**Best**	**Mean**	**Std**	**Best**	**Mean**	**Std**	**Best**	**Mean**	**Std**	**Best**	**Mean**	**Std**
WFG1	0.006	0.020	0.024	0.048	0.195	0.076	0.008	0.039	0.030	0.009	**0.014**	0.012	0.009	0.094	0.049
WFG2	0.003	**0.003**	0.000	0.008	0.008	0.000	0.004	0.004	0.000	0.007	0.045	0.066	0.004	0.006	0.001
WFG3	0.008	**0.008**	0.000	0.010	0.010	0.000	0.021	0.022	0.001	0.010	0.010	0.000	0.010	0.011	0.000
WFG4	0.006	**0.006**	0.000	0.009	0.009	0.000	0.014	0.016	0.001	0.009	0.010	0.001	0.008	0.014	0.003
WFG5	0.037	**0.056**	0.005	0.065	0.067	0.001	0.071	0.072	0.001	0.060	0.065	0.002	0.065	0.067	0.001
WFG6	0.024	0.047	0.013	0.009	0.022	0.011	0.014	0.016	0.001	0.026	0.039	0.008	0.007	**0.007**	0.000
WFG7	0.006	**0.006**	0.000	0.009	0.009	0.000	0.013	0.015	0.001	0.009	0.009	0.000	0.007	0.007	0.000
WFG8	0.034	**0.048**	0.005	0.110	0.115	0.002	0.119	0.124	0.002	0.116	0.117	0.001	0.116	0.118	0.001
WFG9	0.009	**0.011**	0.001	0.012	0.028	0.025	0.031	0.077	0.046	0.123	0.126	0.001	0.011	0.035	0.033
DTLZ1	0.001	**0.001**	0.000	0.001	**0.001**	0.000	0.002	0.002	0.000	0.001	**0.001**	0.000	0.002	0.002	0.000
DTLZ2	0.002	**0.002**	0.000	0.003	0.003	0.000	0.003	0.004	0.000	0.003	0.003	0.000	0.002	**0.002**	0.000
DTLZ3	0.002	**0.002**	0.000	0.003	0.003	0.000	0.003	0.003	0.000	0.003	0.003	0.000	0.002	**0.002**	0.000
DTLZ4	0.002	**0.002**	0.000	0.003	0.003	0.000	0.003	0.058	0.150	0.003	0.003	0.000	0.002	0.124	0.207
DTLZ5	0.002	**0.002**	0.000	0.003	0.003	0.000	0.003	0.004	0.000	0.003	0.003	0.000	0.002	**0.002**	0.000
DTLZ6	0.002	**0.002**	0.000	0.003	0.004	0.003	0.003	0.003	0.000	0.003	0.003	0.000	0.002	0.004	0.005
DTLZ7	0.002	**0.002**	0.000	0.004	0.004	0.000	0.003	0.003	0.000	0.004	0.004	0.000	0.002	**0.002**	0.000
UF1	0.003	**0.003**	0.000	0.007	0.007	0.000	0.005	0.006	0.000	0.004	0.004	0.000	0.004	0.004	0.000
UF2	0.003	**0.003**	0.000	0.006	0.007	0.001	0.009	0.011	0.001	0.009	0.014	0.003	0.008	0.009	0.001
UF3	0.030	0.042	0.007	0.004	**0.005**	0.000	0.006	0.008	0.002	0.008	0.022	0.018	0.005	0.010	0.004
UF4	0.007	**0.007**	0.000	0.027	0.031	0.002	0.034	0.037	0.001	0.038	0.039	0.001	0.030	0.033	0.001
UF5	0.016	**0.026**	0.006	0.140	0.251	0.063	0.094	0.129	0.032	0.103	0.146	0.022	0.094	0.135	0.067
UF6	0.017	**0.024**	0.012	0.036	0.225	0.151	0.078	0.202	0.060	0.078	0.130	0.074	0.081	0.220	0.103
UF7	0.003	**0.003**	0.000	0.004	0.004	0.000	0.008	0.010	0.001	0.010	0.016	0.002	0.004	0.012	0.005
Mean	0.010	**0.014**	0.003	0.023	**0.044**	0.015	0.024	**0.038**	0.014	0.028	**0.036**	0.009	0.021	**0.040**	0.021

Table E.2 Statistical Tests and Deterioration Level of the IGD+ for problems with two objectives

	↑	↓	↔	**Score**	**Deterioration**
AVSD-MOEA/D	78	13	1	65	0.085
MOEA/D-DE	36	50	6	-14	0.770
NSGA-II	21	65	6	-44	0.620
NSGA-III	39	49	4	-10	0.581
R2-EMOA	44	41	7	3	0.675

Tables E.3 and E.4 show the same information for problems with three objectives. In this case, the number of times that each method belonged to the winning groups were 15, 3, 1, 1 and 1 for AVSD-MOEA/D, R2-EMOA, MOEA/D-DE, NSGA-III and NSGA-II, respectively. Furthermore, the difference between the mean IGD+ obtained by AVSD-MOEA/D and the best method was never greater than 0.005. However, all the other methods exhibited a deterioration in excess of 0.005 in several cases. In particular, this happened in 13, 17, 18 and 18 problems for R2-EMOA, MOEA/D-DE, NSGA-III and NSGA-II respectively. Interestingly, AVSD-MOEA/D is quite superior both in total deterioration and in the score generated from pair-wise statistical tests. In fact, its total deterioration for the entire set of problems is just 0.005. In contrast, the total deterioration of R2-EMOA, MOEA/D-DE, NSGA-III and NSGA-II are 0.656, 0.774, 0.844 and 1.260, respectively. Although these conclusions are quite similar to those drawn in the main document for hypervolume, expressing these results in terms of IGD+ emphasizes the outstanding performance of AVSD-MOEA/D.

Table E.3 Summary of the IGD+ attained for problems with three objectives

	AVSD-MOEA/D			**MOEA/D-DE**			**NSGA-II**			**NSGA-III**			**R2-EMOA**		
	Best	**Mean**	**Std**	**Best**	**Mean**	**Std**	**Best**	**Mean**	**Std**	**Best**	**Mean**	**Std**	**Best**	**Mean**	**Std**
WFG1	0.073	**0.085**	0.010	0.083	0.136	0.043	0.108	0.129	0.012	0.092	0.096	0.010	0.079	0.104	0.023
WFG2	0.055	**0.057**	0.001	0.062	0.069	0.004	0.096	0.135	0.021	0.097	0.113	0.018	0.119	0.120	0.000
WFG3	0.026	**0.027**	0.000	0.032	0.032	0.000	0.047	0.095	0.030	0.084	0.098	0.012	0.033	0.034	0.000
WFG4	0.088	**0.092**	0.001	0.133	0.133	0.000	0.132	0.142	0.009	0.133	0.133	0.000	0.119	0.124	0.002
WFG5	0.122	**0.137**	0.006	0.185	0.185	0.000	0.181	0.192	0.008	0.182	0.185	0.001	0.166	0.169	0.002
WFG6	0.112	0.130	0.009	0.140	0.158	0.009	0.159	0.183	0.012	0.145	0.162	0.009	0.120	**0.128**	0.005
WFG7	0.089	**0.091**	0.001	0.133	0.133	0.000	0.130	0.158	0.010	0.133	0.133	0.000	0.117	0.119	0.001
WFG8	0.122	**0.128**	0.003	0.191	0.193	0.001	0.242	0.254	0.006	0.194	0.198	0.002	0.174	0.178	0.002
WFG9	0.100	**0.103**	0.001	0.135	0.138	0.001	0.178	0.252	0.017	0.149	0.237	0.022	0.129	0.133	0.002
DTLZ1	0.015	0.015	0.000	0.014	**0.014**	0.000	0.019	0.021	0.001	0.014	**0.014**	0.000	0.015	0.015	0.000
DTLZ2	0.023	**0.024**	0.000	0.029	0.029	0.000	0.033	0.037	0.002	0.029	0.029	0.000	0.026	0.027	0.000
DTLZ3	0.023	**0.023**	0.000	0.029	0.029	0.000	0.035	0.039	0.002	0.029	0.029	0.000	0.026	0.027	0.000
DTLZ4	0.023	**0.023**	0.000	0.029	0.029	0.000	0.032	0.107	0.200	0.029	0.042	0.075	0.026	0.045	0.106
DTLZ5	0.004	0.004	0.000	0.005	0.005	0.000	0.003	**0.003**	0.000	0.008	0.010	0.002	0.003	**0.003**	0.000
DTLZ6	0.004	0.004	0.000	0.005	0.009	0.007	0.003	0.010	0.029	0.010	0.013	0.002	0.003	**0.003**	0.001
DTLZ7	0.033	**0.033**	0.000	0.059	0.059	0.000	0.040	0.060	0.056	0.050	0.061	0.005	0.075	0.113	0.047
UF8	0.030	**0.032**	0.001	0.040	0.054	0.016	0.089	0.111	0.026	0.040	0.075	0.066	0.042	0.050	0.008
UF9	0.029	**0.031**	0.001	0.038	0.169	0.071	0.103	0.164	0.058	0.032	0.046	0.041	0.034	0.110	0.085
UF10	0.060	**0.072**	0.010	0.105	0.309	0.091	0.229	0.273	0.043	0.154	0.276	0.055	0.254	0.261	0.017
Mean	0.054	**0.058**	0.002	0.076	**0.099**	0.013	0.098	**0.124**	0.029	0.084	**0.103**	0.017	0.082	**0.093**	0.016

Table E.4 Statistical Tests and Deterioration Level of the IGD+ for problems with three objectives

	↑	↓	↔	**Score**	**Deterioration**
AVSD-MOEA/D	69	5	2	64	0.005
MOEA/D-DE	35	34	7	1	0.774
NSGA-II	6	65	5	-59	1.260
NSGA-III	22	48	6	-26	0.844
R2-EMOA	46	26	4	20	0.656