Universidad Veracruzana

Artificial Intelligence Research Center

A Bio-inspired algorithm to solve Dynamic Multi-Objective Optimization Problems

Submitted by

María-Guadalupe Martínez-Peñaloza

as the fulfillment of the requirement for the degree of

Ph.D. on Artificial Intelligence

Advisor

Dr. Efrén Mezura-Montes

January 2018

Abstract

Many different applications in engineering, science and industry have a considerable degree of complexity, and sometimes this complexity may be based on the presence of multiple conflicting objective functions, which must be simultaneously optimized. These kinds of problems are the so-called multi-objective optimization problems (MOPs). However, in everyday life, most optimization problems are not static in nature and usually have at least one objective that can change over time. In recent years, MOPs in dynamic environments have attracted some research efforts. However, most research focuses on either static multi-objective optimization or dynamic single-objective optimization. Therefore, not much research has been done on Dynamic Multi-Objective Optimization (DMO). Evolutionary algorithms and Artificial Immune System (AIS) have been popular to solve dynamic single objective optimization problems. Nevertheless, such combination has been scarcely explored when solving DMOPs. On the other hand, a few Differential Evolution(DE)-based algorithms have been proposed. In this thesis, two Differential Evolution-based algorithms to solve dynamic multi-objective optimization problems (DMOPs) are proposed. The novelty of these algorithms with respect to other approaches is the fact that the algorithms take advantage of DE and AIS to track the changes in the environment and respond quickly when a change is detected. Three main issues of the algorithms are explored: (1) the general performance of both algorithms in comparison with other well-known algorithms, (2) their sensitivity to different change severities and frequencies, and (3) the role of their change reaction mechanism based on an immune response. For such purpose, different performance metrics, four unary and one binary, are computed in a comparison against other state-of-the-art dynamic multi-objective evolutionary algorithms (DMOEAs) when solving a novel suite of test

problems. The statistically validated results indicate that the proposed approaches are robust to change frequency and severity variations and can track the environmental changes finding a good distribution of solutions.

Acknowledgments

I would like to thank sincerely my thesis director Dr. Efrén Mezura Montes for his guidance and all the support provided during the doctoral program. Thanks for showing me the way to do research.

I also want to thank the reviewers of this document for their comments and suggestions which helped to enhance its quality. Thank you to Dra. Alicia Morales Reyes, Dra. Ericka Janet Rechy Ramírez, Dra. Marcela Quiroz Castellanos, Dr. Gregorio Toscano Pulido, and Dr. Nicandro Cruz Ramírez.

I want to thank Dr. Hernán Aguirre for received me during the international doctoral stay at Shinshu University of Nagano, Japan.

In my national research stay in the National Institute of Astrophysics, Optics, and Electronics, I would like to thank Dra. Alicia Morales Reyes for her advice and friendship during my stay.

I would like to thank my parents, Martha and Pablo and my brothers Víctor and Pablo for their endless love, encouragement and for always believing in me.

Finally, I want to acknowledge support from CONACyT through scholarship No. 258800 and the University of Veracruz to pursue graduate studies.

This research work was derived from the CONACyT project entitled ” Dinamismo y Modelos Subrogados en Algoritmos Bio-Inspirados para Optimización con Restricciones´´(Ref. 220522), whose Principal Investigator is Dr. Efrén Mezura Montes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In Engineering, Science and Industry there are many cases where a research goal can be translated into an optimization problem. For example, mechanical engineers are interested in designing mechanical components for the purpose of achieving either a minimum manufacturing cost or maximum component life. In production plants, engineers are interested in designing optimum schedules or different machine operations to minimize the idle time of machines and the overall job completion. Civil engineers, on the other hand, are involved in designing structures in order to achieve a minimum overall cost or maximum safety or both [17, 23]. Therefore, all the above-mentioned examples involve the minimization or maximization of different tasks (collectively known as optimization) of an objective [23].

Real-world optimization problems have a considerable degree of complexity, and this complexity may be based on the presence of multiple conflicting objective functions, which must be simultaneously optimized. This kind of optimization problems is known as Multi-objective Optimization Problems (MOPs). In the case of single-objective optimization, a single optimal solution should be reached (the global optimum). In contrast, in multi-objective optimization, a set of solutions with different trade-offs among the objectives is usually achieved.

There are several mathematical programming methods which have shown to be effective for solving MOPs. However, there are cases where these methods can not guarantee that the solution obtained is optimum. Also, some mathematical programming methods can be inefficient or even inapplicable for particular problems.

Because of these reasons, the use of meta-heuristics as Evolutionary Algorithms (EAs) to solve MOPs has become increasingly popular.

Evolutionary algorithms have shown being good candidates to solve MOPs in a single run, compared to classical methods such as gradient descent and simulated annealing [26],[20]. Therefore, in the last few years, there have been significant contributions to Multi-objective Evolutionary Algorithms (MOEAs) design. Different MOEAs currently proposed are capable of attaining the multi-objective optimization goals with high efficacy regarding convergence and diversity of solutions [24, 94, 101].

In recent years, MOPs in dynamic environments have attracted some research efforts. Therefore, the so-called Dynamic MOPs (DMOPs) are gaining attention [77]. Initially, not much research had been done on Dynamic Multi-Objective Optimization (DMO) [2, 33], but in the last few years, more researchers focused on solving DMOPs using nature-inspired meta-heuristics. Evolutionary algorithms and Artificial Immune Systems (AIS) have been popular to solve dynamic single objective optimization problems [13, 42, 73, 91, 105]. Nevertheless, such combination has been scarcely explored when solving DMOPs [8].

Differential Evolution (DE) has been widely applied to solve static optimization problems. Furthermore, it has shown a high convergence rate with a higher degree of robustness than other meta-heuristics [68]. In addition, different from other meta-heuristics, the DE algorithm does not use a fixed distribution, instead, it generates a diverse set of search directions based on the distribution of solutions in the current population. This last feature seems to be one of its main advantages because it allows a better exploration of the search space [68]. Despite of the excellent performance of DE solving static optimization problems, it has been little applied in DMO [95]. Therefore, an important goal of this thesis aims to analyze the behavior of DE solving dynamic multi-objective problems.

In DMO, maintaining population diversity is a very important task, if the population diversity is lost prematurely, then tracking the new optimal positions in the environment becomes more difficult. AIS have shown a competitive performance solving multi-objective optimization problems in dynamic environments. The cycle of the immune response against foreign components in the organism (antigens) has different dynamic characteristics like adaptation, diversity maintenance, dynamism,

and detection. This is the main motivation to use immune ideas for solving dynamic optimization problems.

One of the most important features that should be considered for the design of dynamic optimization algorithms is that such algorithms must present not only a fast convergence level but also a good mechanism to promote diversity [107]. However, convergence in dynamic optimization could lead to several problems. For example, the optimization algorithm could find it difficult to find the global optima due to the lack of diversity once it has already converge in a particular region of the problem landscape [77]. Since the dynamic characteristics of AIS enhance algorithms population diversity and the good performance of DE as global optimization algorithm promotes a good convergence, it is expected that the combination of these two meta-heuristics leads to the design of a new competitive algorithm able to track the changes in the environment.

On the other hand, in multi-objective optimization, the survival selection mechanism plays an important role to determine the quality of the solutions that are able to survive through the optimization process. Different survival selection mechanisms have been proposed being the most popular the Pareto-based selection mechanism. However, different studies suggest that MOEAs based on Pareto have difficulties when solving MOPS with more than three objectives [54]. So, there has been a lot of research regarding the design of alternative selection mechanisms as the use of performance metrics to guide the search. Even though different works regarding MOEAs based on performance metrics or indicators have been proposed for static multi-objective optimization, to the best of the author's knowledge, there is not an approach which uses a performance metric in the selection mechanism to guide the search when solving DMOPs. For the reasons described above, in this thesis, the design of a dynamic optimization algorithm based on an indicator is also proposed.

In this thesis, two new Dynamic Multi-Objective Evolutionary Algorithms (DMOEAs), namely Immune Generalized Differential Evolution (Immune GDE3) and Distance-based Immune Generalized Differential Evolution (DIDGDE) are proposed. The novelty of these algorithms with respect to other approaches is the fact that the algorithms take advantage of DE and AIS to track the changes in the environment and respond quickly when a change is detected.

1.1 Problem statement

As mentioned above, optimization problems that occur in situations of everyday life are normally not static in nature. Some examples of these problems are: scheduling, robot path planning, air traffic control, routing in telecommunication networks, etc. [1, 11, 78, 89, 104]. These problems are called dynamic or non-stationary. A dynamic optimization problem may also involve more than one objective to be optimized and those problems are called Dynamic Multi-objective Optimization Problems (DMOPs).

Optimization in a changing environment is a challenging task, especially when multiple objectives need to be optimized. The search then requires a fast convergence in the current problem conditions and also quick responses after changes [8]. In this way, it is very important to design approaches that could detect a change in the environment and then finding the new Pareto optimal front as soon as possible in the presence of new changes. In addition, the study on this optimization area is still limited due to a lack of standard benchmark problems and appropriated performance metrics [5, 44, 45, 47, 48].

There has been research regarding different mechanisms to design dynamic multi-objective evolutionary algorithms [5, 77]. However, some of those mechanisms have several limitations, e.g., mechanisms which introduce diversity during the optimization process usually depend on the ability of the optimization algorithm, and sometimes such algorithm presents difficulties tracking the new positions of the Pareto optimal front. Furthermore, they may not efficiently work when the changes of the problem are severe or fast. The use of multiple populations in the design of DMOEAs can affect the performance of the optimization algorithm; approaches based on prediction mechanisms depend on how well the predictors are trained; the use of memory-based approaches has the disadvantage that they can generate redundant information and may not necessarily promote diversity [77]. Therefore, in this thesis, the design of two DMOEAs able to track the changes in the environment as quickly as possible and to obtain solutions that are spread along the Pareto front as uniformly as possible, is presented.

On the other hand, an important issue in dynamic multi-objective optimization is the

performance comparison of different algorithms. Therefore, different performance unary metrics traditionally used to evaluate the performance of multi-objective optimization evolutionary algorithms (MOEAs) have been adapted to work with DMO [77]. However, previous studies have shown in general that unary indicators are not capable of indicating whether the quality of an approximation set is better than another, even if several sets of unary indicators are used [109]. Hence, binary quality indicators enhance the empirical evidence, on which it is possible to detect whether an algorithm performs better than another. Therefore, in conjunction with unary indicators, binary ones can be used to complement the performance evaluation of an algorithm [109]. Due to this reason, among the metrics used to evaluate the performance of DMOEAs, to the best of the author's knowledge, a binary metric has not been yet adapted to compare DMOPs. In this thesis, a binary metric called C-metric is also adapted to evaluate the performance of dynamic MOEAs.

1.2 Hypothesis

From the previous paragraphs, the main hypothesis for this research is the following:

A model based on Differential Evolution and inspired by an immune response, particularly in the clonal selection algorithm, will lead to the design of a Dynamic Multi-objective Evolutionary Algorithm for solving DMOPs with a highly competitive performance. So that, it has the following features.

- It has good performance concerning the two following aspects:
  - It is able to track the new positions of the Pareto optimal front in each time step, obtaining solutions that are, as close as possible, to the Pareto Optimal Front.
  - It can produce solutions with a uniform distribution along the Pareto front in each time step.
- It can efficiently work solving problems with different change frequencies and change severities.
- It takes the advantages of different mechanisms to deal with dynamic environments to obtain competitive results.

1.3 Goals

1.3.1 Main goal

The main goal of this thesis is to advance the state-of-the-art in dynamic multi-objective optimization, particularly regarding the design of a dynamic multi-objective nature-inspired algorithms, which combine the advantages of MOEAs and other meta-heuristics as AIS. Also, it is of our interest to design DMOEAs that use performance metrics to guide the optimization process when solving DMOPs.

1.3.2 Specific goals

The specific goals of this thesis are the following:

- To gain a deep knowledge of the state-of-the-art regarding DMOEAs, including their main mechanisms and their advantages and disadvantages. The aim is to identify possible improvements that lead to the design of a new DMOEA, which is more efficient than state-of-the-art DMOEAs used in this research field.

- To advance knowledge within dynamic multi-objective optimization by developing an efficient DMOEA which combines the advantages of MOEAs and other meta-heuristics as AIS. This DMOEA must comply with desirable features (efficiency, fast convergence, good tracking ability, good distribution of solutions).

- To analyze the feasibility of using a performance metric to guide the search in the optimization process in the dynamic optimization area. Therefore, a DMOEA based on a performance metric should be designed. This DMOEA also must comply with desirable features listed in the above item.

- To identify representative state-of-the-art DMOEAs for performance assessment.

- To analyze the efficiency of the proposed approaches to solving DMOPs with different frequencies and severities of change.

- To perform a comparative study between the proposed approaches and other popular state-of-the-art DMOEAs.

- To validate the performance of the proposed approaches with respect to other popular state-of-the-art DMOEAS. This validation will be done with a set of benchmarks functions representative of dynamic multi-objective optimization. Performance will be assessed using performance measures usually adopted in multi-objective optimization that quantifies the performance of DMOEAs.

- To analyze the ability of the designed algorithms to track the changes in the environment.

- To understand the role that the immune response plays in the performance of the proposed approaches.

- To analyze the behavior of DE solving dynamic multi-objective problems.

1.4 Expected contributions

- A new DMOEA based on DE which combines different mechanisms to deal with changes in the environment.

- A new DMOEA that uses a performance metric to guide the search in the optimization process.

- A detailed empirical study of the proposed approaches. This study must be based on an in depth statistical analysis that considers well-known DMOEAs, standard test problems and performance measures commonly adopted in the specialized literature.

- An adaptation of a binary metric to evaluate the performance of DMOEAs in dynamic multi-objective optimization.

1.5 Publications

In the following, the products obtained during the development of this thesis are presented.

1.5.1 Journal papers

- María-Guadalupe Martínez-Peñaloza and Efrén Mezura-Montes. Immune Generalized Differential Evolution for Dynamic Multi-Objective Environments: An empirical study, *Knowledge-Based Systems*, vol. 142, pages:192–219, 2018.

- Héctor Cervantes-Culebro, Carlos A. Cruz-Villar, María-Guadalupe Martínez-Peñaloza, Efrén Mezura-Montes, Constraint-Handling Techniques for the Concurrent Design of a Five-Bar Parallel Robot, *IEEE Access*, 5(1):23010–23021, 2017.

- María-Guadalupe Martínez-Peñaloza, Efrén Mezura-Montes, Hernan Aguirre and Alicia Morales-Reyes. Distance-based Immune Generalized Differential Evolution algorithm for Dynamic Multi-Objective Optimization, *Applied Soft Computing*, (to be submitted).

1.5.2 International conference papers

- María-Guadalupe Martínez-Peñaloza and Efrén Mezura-Montes, Immune Generalized Differential Evolution for Multi-objective Dynamic Optimization Problems, in *Proceedings of the IEEE Congress on Evolutionary Computation*, pages: 1918-1925, IEEE Press, 2015.

1.6 Methodology

Firstly, the DMO literature was reviewed to determine the limitations with regards to: (1) the development of DMO algorithms, especially concerning evolutionary algorithms, (2) benchmark problems for DMO, and (3) performance metrics to evaluate the performance of the developed DMOEAs.

Secondly, a bio-inspired algorithm to solve DMOPs must be proposed and validated empirically using a novel suite of test problems. Figure 1.1 presents the general methodology followed in the development of this thesis.

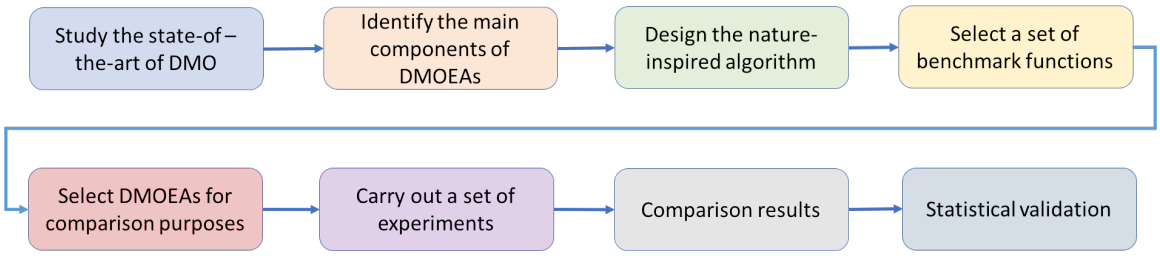

Figure 1.1: General methodology adopted in the thesis.

1.7 Structure of the document

This document is organized in seven chapters. The first four chapters (including this one) describe basic concepts required, on the one hand, to understand the contributions of this thesis, and, on the other hand, to support the contents of the following chapters.

In Chapter 2, basic concepts related to optimization are presented. Additionally, mathematical definitions of single-objective and multi-objective optimization problems are introduced. Finally, in Chapter 2, some mathematical programming methods traditionally used for solving MOPs are also presented. Chapter 3, presents a brief introduction to MOEAs and describes some of the most popular state-of-the-art MOEAs. In addition, some performance metrics to evaluate MOEAs are also introduced. The mathematical definition of DMOPs is introduced in Chapter 4. Furthermore, the related work concerning DMOEAs is also given in this chapter. In Chapters 5 and 6, the contributions of this thesis are presented. Two new DMOEAs are proposed, evaluated and compared with respect to other well-known DMOEAs. In Chapter 5, the proposal and empirical validation of a novel DMOEA called Immune GDE3 is presented. In Chapter 6, a DMOEA based on Inverted Generational Distance indicator is presented. This DMOEA is an improved version of Immune GDE3 algorithm. Finally, Chapter 7, provides the concluding remarks and possible directions for future work.

Chapter 2

Optimization Background

This chapter presents a theoretical overview related to optimization. Therefore, the main goal of this chapter is that the reader familiarizes with the basic concepts, definitions, and notations that are used throughout this thesis document.

2.1 Optimization concepts

Optimization refers to the process of finding the minimum or maximum possible solution to a given problem. Each optimization problem contains one or more objective functions, a set of decision variables and most of them also contains a set of constraints. The objective function is the mathematical function that expresses the objective to be minimized or maximized. The objective function is also known as the fitness function, the cost function or optimization criterion. Each objective function has a vector of decision variables $\vec{x}$ that influence the value of the objective function. When an optimization problem has constraints, the set of constraints restricts the possible values that can be assigned to the decision variables. The set of all possible values of $\vec{x}$ that satisfy the problem's constraints forms the feasible region $\mathcal{F}$ which is a subset of the search space. Throughout this thesis, without loss of generality, minimization is assumed.

According to the number of objectives to be optimized, optimization problems can be classified as single-objective optimization problems (SOPs) and multi-objective optimization problems (MOPs). The former are problems which involve only one

objective function and the latter problems which involve more than one objective function.

2.2 Single-objective optimization

Definition 1. Single-objective optimization problem: Mathematically, a SOP can be defined as: Find the vector $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ which minimizes the function $f(\vec{x})$ subject to $\vec{x} \in \mathcal{F}$, where $\mathcal{F} \subseteq \mathbb{R}^n$ is the feasible region which satisfies the m inequality constraints:

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

and the p equality constraints:

$$h_j(\vec{x}) = 0 \quad j = 1, 2, \ldots, p$$

The feasible solution $\vec{x}^* \in \mathcal{F}$ that corresponds to the smallest value of $f(\vec{x})$ in all the search space is known as *global optimum*.

To understand the complexity involved in solving an optimization problem, the following definitions are introduced.

Definition 2. Global minimum: Given a function $f(\vec{x})$ defined on a set $\mathcal{F}$, a solution $\vec{x}^* \in \mathcal{F}$ is called global minimum [1] of the objective function f, if only and if:

$$\forall \vec{x} \in \mathcal{F} : f(\vec{x}^*) \leq f(\vec{x}) \tag{2.1}$$

Definition 3. Local minimum: Given a function $f(\vec{x})$ defined on a set $\mathcal{F}$, a solution $\vec{x}^l \in \mathcal{F}$ is called local minimum, if and only if:

$$\forall \vec{x} \in \mathcal{F} : f(\vec{x}^l) \leq f(\vec{x}), \text{ such as: } \left\| \vec{x} - \vec{x}^l \right\| < \epsilon \tag{2.2}$$

where $\epsilon > 0$ and the value $f(\vec{x}^l)$ is called local minimum.

[1] The global minimum may not be unique i.e., an optimization problem can have more than one global minimum.

2.3 Multi-objective optimization

In the field of optimization, usually real-world problems have a considerable degree of complexity, and this complexity may be based on the presence of multiple conflicting objectives to be optimized. This kind of optimization problems are known as multi-objective optimization problems (MOPs).

Multi-objective optimization is the process of simultaneously optimizing a vector function whose elements represent the objective functions which are normally in conflict with each other. Therefore, solving MOPs implies finding trade-offs among all the objective functions. In this kind of problems, a set of optimal solutions is obtained instead of a single one as in the case of SOPs. This is because in multi-objective optimization it is not possible to find a single optimal solution which optimizes all the objective functions simultaneously [19].

In the following, some general concepts and notations regarding to Multi-objective optimization are presented.

Definition 4. Multi-objective optimization problem: Mathematically, a general multi-objective optimization problem (MOP) can be formally defined as:

Find $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ which minimizes:

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T \tag{2.3}$$

Subject to:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \ldots, m$$
$$h_j(\vec{x}) = 0 \quad j = 1, 2, \ldots, p$$

where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables. The decision variables can be continuous or discrete, in this work, we are only interested in continuous domains. $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, k$ are the objective functions and $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$ are the inequality and equality constraint functions of the problem, respectively. The set of constraints define the feasible region $\mathcal{F} \subseteq \mathbb{R}^n$. Therefore, any decision vector $\vec{x} \in \mathcal{F}$ is considered a feasible solution of the MOP.

Definition 5. Decision variable space: The decision variable space is the n-

dimensional space of the decision variables, in which each coordinate axis corresponds with one component of vector $\vec{x}$.

Definition 6. Objective functions: The objective functions evaluate how good a given solution is. The objective functions are usually denoted as $f_i : \mathbb{R}^n \to \mathbb{R}$. In MOPs, more than one objective functions are solved and they are denoted by an objective vector: $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T$, where $\vec{f}(\vec{x}) : \mathbb{R}^n \to \mathbb{R}^k$.

Definition 7. Objective function space: The objective function space is the k-dimensional space of the objective functions, in which each coordinate axis corresponds with one component of vector $\vec{f}(\vec{x})$.

Figure 2.1 represents the above definitions for a MOP with three decision variables and two objective functions. $\vec{f}$ is a function that maps a vector $\vec{x} \in X$ in the decision variable space to a vector $\vec{z} \in Z$ in the objective function space.

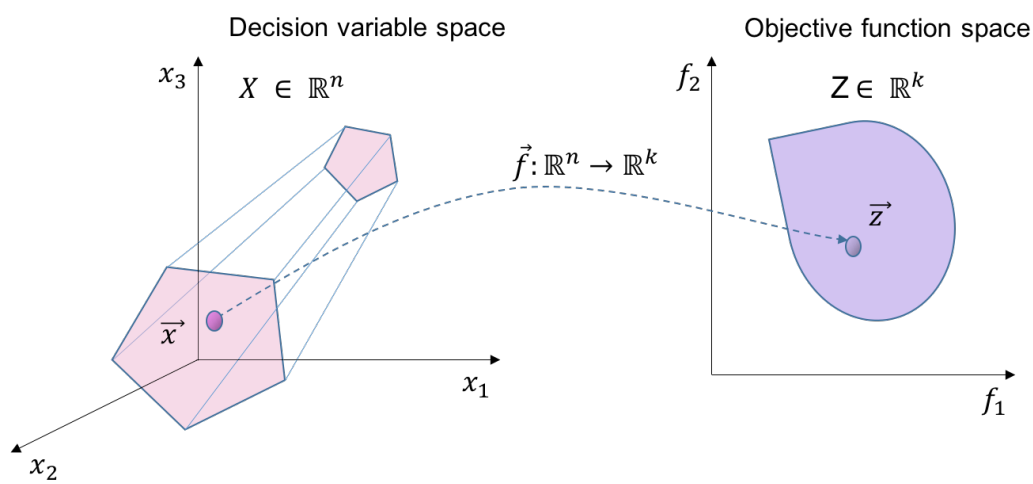

Figure 2.1: Mapping between decision variables space and objective function space.

2.3.1 Optimality in multi-objective optimization

In single-objective optimization, it is possible to determine if one solution is better than another solution by comparing their function values. Therefore, a single optimal solution is obtained (the global optimum). On the other hand, in MOPs, the aim is to find a set of optimal solutions which represent the best possible trade-offs among

all the objectives. Therefore, in multi-objective optimization the definition of optimality changes. The notion of optimality most commonly adopted in multi-objective optimization is normally referred to as Pareto optimality and it was originally proposed by Edgeworth in 1881 [29] and later generalized by Pareto in 1896 [74]. In the following, some important concepts related to Pareto optimality are presented.

Definition 8. Pareto Dominance: Let f_i be an objective function. Then, a decision vector $\vec{x} = [x_1, \ldots, x_n]^T$ is said to dominate $\vec{y} = [y_1, \ldots, y_n]^T$ (denoted by $\vec{x} \prec \vec{y}$) if and only if, $f_i(\vec{x}) \leq f_i(\vec{y})$ for all $i \in \{1, \ldots, k\}$ and $f_i(\vec{x}) < f_i(\vec{y})$ in at least one f_i (see Figure 2.2).

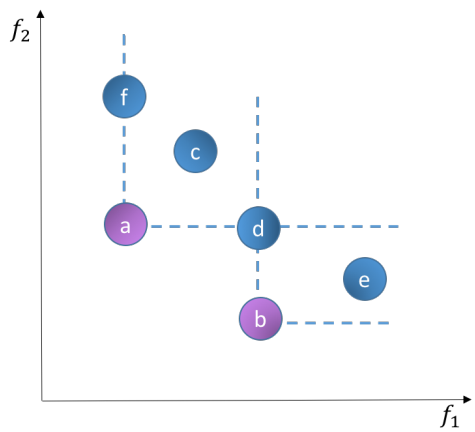

Figure 2.2: Pareto dominance for a MOP with two objective functions. The solution $a \prec c$ such that, a is better in f_1 and f_2, $a \prec d$, such that a is equal to d in f_2 but it is better than d in f_1, $b \prec d$ such that, b is equal to d in f_1 but it is better than d in f_2, $b \prec e$ such that, b is better than e in both functions; a and b are incomparable, therefore a and b are Pareto optimal solutions.

Definition 9. Pareto Optimality: A vector of decision variables $\vec{x}^* \in \mathcal{F}$ is Pareto optimal, if there does not exist other solution $\vec{x} \in \mathcal{F}$ such that $\vec{x} \prec \vec{x}^*$. If $\vec{x}^*$ is Pareto optimal, the objective vector, $\vec{f}(\vec{x}^*)$, is also Pareto optimal.

The set of all the Pareto optimal decision vectors forms the Pareto optimal set (POS) and their corresponding objective vectors form the Pareto optimal front (POF). The POS and the POF are defined as follows:

Definition 10. Pareto optimal set: For a given MOP, the POS ($\mathcal{P}^*$) is defined as: $\mathcal{P}^* = \{\vec{x} \in \mathcal{F} | \nexists \vec{y} \in \mathcal{F} : \vec{y} \prec \vec{x}\}$

Definition 11. Pareto optimal front: For the objective vector, $\vec{f}(\vec{x})$ and the POS ($\mathcal{P}^*$), the POF is defined as: $\mathcal{PF}^* = \left\{\vec{f}(\vec{x}) | \vec{x} \in \mathcal{P}^*\right\}$

Figure 2.3 illustrates the POS and the POF of a MOP with two decision variables and two objective functions.

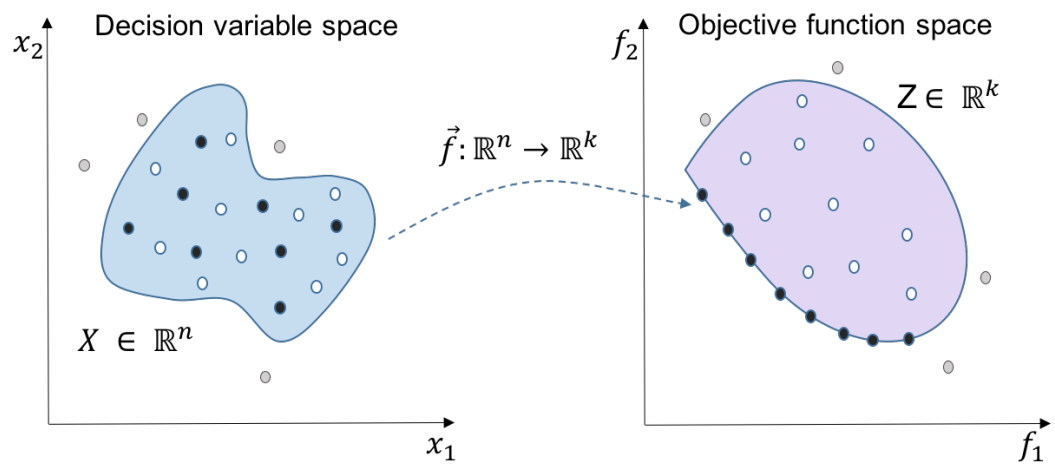

Figure 2.3: Illustration of the Pareto optimal set and its mapping to the Pareto front. Black points are non-dominated solutions and they define the Pareto optimal set in decision variable space and the Pareto front in objective space. White points are dominated vectors and gray points are infeasible solutions.

Definition 12. Ideal objective vector: The ideal objective vector, denoted by $\vec{z}^* = [z_1^*, z_2^*, \ldots, z_k^*]^T$ is obtained by minimizing each of the objective functions individually subject to the constraints (if any), i.e., $z_i^* = \min f_i(\vec{x})$ subject to $\vec{x} \in \mathcal{F}$.

Definition 13. Nadir objective vector: The components of the nadir objective vector denoted by $\vec{z}^{nad} = \left[z_1^{nad}, z_2^{nad}, \ldots, z_k^{nad}\right]^T$ are the upper bounds of the Pareto optimal set.

In Figure 2.4 the above definitions are illustrated for a MOP with two objectives.

In general, it is not possible to find an analytical expression of the line or surface that defines the Pareto front of a MOP. Therefore, the most common procedure to

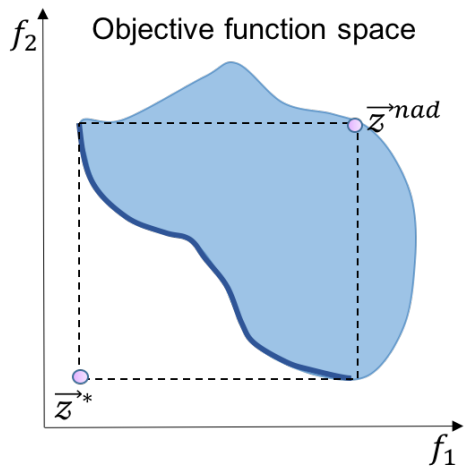

Figure 2.4: Illustration of ideal vector $\vec{z}^*$ and nadir vector $\vec{z}^{nad}$ for a MOP with two objective functions.

generate the Pareto front for a given MOP is to compute a sufficient number of points within the feasible region $\mathcal{F}$, and then select the non-dominated vectors from them.

Since the size of the Pareto optimal set might be infinite, the main goal when solving a MOP is to find an approximation of the true POF such that:

- The distance between the found POF and the true POF is minimized.
- The set of non-dominated solutions maintains a distribution as diverse as possible along the found POF.

2.4 Optimization methods to solve MOPs

Over the years, a large number of optimization methods for solving MOPs have been proposed. These methods can be classified in many ways according to different taxonomies (for example, enumerative, deterministic and stochastic methods [19]). Enumerative methods evaluate each possible solution within some finite search space. Therefore, they are the simplest search strategy. It is easy to note that enumerative methods are not suitable for large search spaces. On the other hand, deterministic methods incorporate problem domain knowledge. Most of them are

graph/tree search algorithms. Deterministic methods are often ineffective solving irregular MOPs, i.e., when the MOP is high-dimensional, discontinuous, multimodal or NP-hard [19]. Since many real-world MOPs are irregular, stochastic search methods such as the Bio-inspired algorithms have been developed as alternative approaches for solving them (this kind of methods will be described in the next chapter).

2.4.1 Mathematical programming methods to solve MOPs

As was aforementioned, in multi-objective optimization a Pareto optimal set is obtained instead of a single one optimal solution. However, it is preferable to obtain one point as a solution for the MOP. The decision maker (DM) is responsible for choosing only one solution from all those available. The operations research community has proposed several stochastic and deterministic optimization methods to solve MOPs. Mathematical programming methods as well as multi-criteria decision making methods, are commonly classified based on how and when the DM is required to provide preference information. Cohon and Marks [21] proposed one of the most popular classifications of mathematical programming techniques within the operations research community. This classification is presented below.

- **A priori methods:** The DM defines the importance of the objective functions before starting the search.

- **A posteriori methods:** First, the approximate Pareto front is generated by an optimizer, and then, the DM selects the most preferred one(s) according to her/his preferences.

- **Interactive methods:** Both optimizer method and DM work progressively. The optimizer produces solutions and the DM provides preference information so that the most preferred solutions can be found.

In the following, a brief description of the most popular Multi-Criteria Decision Making methods according to the above classification is presented.

A priori methods

- **Goal Programming:** This method was development by Charnes and Cooper [15]. It is considered one of the first methods explicitly designed for multi-objective optimization. In this method, the DM has to assign aspiration levels $\vec{z}_i (i = 1, \ldots, k)$ to be achieved for each objective function. Therefore, the objective function tries to minimize any deviations from these aspiration levels to the objectives. Both, the objective function and the aspiration level form a goal. Several variants of this method have been proposed (for example, weighted and lexicographic approaches). The weighted approach can be formulated as follows:

$$\min \sum_{i=1}^{k} w_i |f_i(\vec{x}) - \vec{z}_i| \,, \text{ subject to: } \quad \vec{x} \in \mathcal{F} \tag{2.4}$$

  where w_i are weights previously pre-defined by the DM for the ith objective function $f_i(\vec{x})$, and $\mathcal{F}$ represents the feasible region. On the other hand, in the lexicographic approach, the DM must specify a lexicographic order on the goals in addition to the aspiration levels. Goal programming is a widely used method to solve MOPs. However, one disadvantage of this method is that the specification of the weighted coefficients or the lexicographic ordering may be difficult. Furthermore, goal programming is not appropriate when a user wants to obtain more than one trade-off solutions. More details about this method can be found in [69].

- **Lexicographic Method:** In this method, the DM has to rank objective functions in order of importance (from best to worst). After that, the optimum solution $\vec{x}^*$ is obtained, by minimizing the objective functions. First, the most important objective function is minimized proceeding with the remaining objective functions according to their order of importance. In each optimization step, the optimal solution found of each objective is added as a constraint for subsequent optimizations. The addition of the constraint guarantees that the most important objective function preserves its optimal value. Suppose that f_1 and f_k are the most and least important objective functions, respectively.

Then, the first problem can be formulated as follows:

$$\text{Minimize:} \quad f_1(\vec{x}), \quad \text{subject to:} \quad g_j(\vec{x}) \leq 0; \quad j = 1, 2, \ldots, m \tag{2.5}$$

and its solution $\vec{x}_1^*$ and $f_1^* = f_1(\vec{x}_1^*)$ is obtained. Then, the second problem is formulated as follows:

$$\text{Minimize:} \quad f_2(\vec{x}), \tag{2.6}$$

subject to:

$$g_j(\vec{x}) \leq 0; \quad j = 1, 2, \ldots, m$$
$$f_1(\vec{x}) = f_1^*$$

and the solution obtained by this problem is $\vec{x}_2^*$ and $f_2^* = f_1(\vec{x}_1^*)$. The procedure continues until all k objectives have been considered. For more information of this method interested readers can be referred to [30].

A posteriori methods

- **Weighting Method:** This method was introduced by Gass and Saaty [36]. The aim of this method is to transform the MOP into a single-objective problem. In the weighting method, each objective function is associated with one weighting coefficient. The goal of the method is to minimize the weighted sum of all the objectives. The weighting coefficients w_i are real numbers. Such that, the MOP is transformed into a SOP as follows:

$$\min \sum_{i=1}^{k} w_i f_i(\vec{x}), \quad \text{subject to:} \quad \vec{x} \in \mathcal{F} \tag{2.7}$$

where $w_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$. The weighting method is a simple way to generate different Pareto optimal solutions. However, the main disadvantage of the weighting method is that not all of the Pareto optimal points can be found if the problem has discontinuous objective functions or it is a nonconvex problem. More details of this method can be found in [36],[69].

- **ϵ-Constraint Method:** This method was proposed by Haimes et al. [41]. The main idea of this method consists of minimizing the most preferred objective function at a time, considering the remaining objectives as constraints bound by some allowable levels ϵ_j. The non-dominated solutions of the problem can be obtained by varying these ϵ_j levels solving the next problem:

$$\text{Minimize:} \quad f_i(\vec{x}) \tag{2.8}$$

subject to:

$$f_j(\vec{x}) \leq \epsilon_j \quad \text{for all} \quad j = 1, \ldots, k, j \neq i, \\ \vec{x} \in \mathcal{F}$$

where $i = 1, \ldots, k$. In order to apply the ϵ-constraint method, a preliminary analysis is recommended to identify proper starting values for ϵ_j. Usually, a mathematical programming technique for single-objective optimization is used to optimize each objective function. Therefore, the ϵ-constraint method is very expensive because it needs to perform k optimizations for all f_i objective functions.

Interactive methods

- **Tchebycheff Method:** This is an interactive weighting vector space reduction method. It was proposed by Steuer [87]. In this method, an utopian vector below the ideal vector should be established. Then, a weighted Tchebycheff metric is used to minimize the distance between the utopian vector and the feasible region. Thus, different solutions are obtained with different weighting vectors in the metric. The solution space is reduced by working with sequences of smaller and smaller subsets of the weighting vector space. Therefore, the main idea of the Tchebycheff Method consists in developing a sequence of progressively smaller subsets of the Pareto optimal set until a final solution is obtained. At each iteration, different objective vectors are presented to the decision maker and he or she must select the most preferred one. The feasible region is then reduced and alternatives from the reduced space are presented

to the DM for selection.

- **GUESS Method:** This method was proposed by Buchannan et al. [9]. It requires that the ideal objective vector $\vec{z}^*$ and the nadir vector $\vec{z}^{nad}$ are available. The general idea of this method consists in maximizing the minimum weighted deviation from the nadir vector. Therefore, the decision maker needs to specify a reference point, usually known as a guess point $\vec{z}^h$, then a solution with equal proportional achievements is generated. After that, the decision maker specifies a new reference point; the process continues until the decision maker is satisfied with the solution proposed. In this method, the scales of the objective functions are normalized. In the GUESS method, the decision maker can examine what kind of an effect his or her input point has on the solution obtained and then modify the input according to his or her preferences. The main disadvantage of this method is the requirement of the nadir vector which is not easy to determine, and usually, it is only an approximation to the true nadir vector. For more details about this method, readers can be referred to [69].

2.5 Summary

This chapter discussed different aspects of optimization relevant to this thesis. Section 2.1 described optimization problems and their main features with regards to the problem's objective functions, decision variables and constraints. The mathematical definition of single-objective optimization problems was presented in Section 2.2. Section 2.3 defined a multi-objective optimization problem. In order to re-defined the optima for a MOP, the concepts of a POS and POF were presented in Section 2.3.1. Finally, in Section 2.4, some of the most popular mathematical programming methods used to solve MOPs were discussed.

Mathematical programming methods for solving MOPs have shown to be effective in many domains. However, they have several disadvantages. For instance, they are sensitive to the shape of the Pareto optimal front and/or they required previous knowledge of the problem being solved. In addition, mathematical programming methods have in common that they need to perform several independent runs to

obtain an approximation of the Pareto optimal set.

Evolutionary Algorithms (EAs) have been found to offer several advantages in comparison with traditional programming methods solving MOPs. The next chapter discusses some of these advantages and presents a review of some of the most popular EAs in the context of the solution of MOPs.

Chapter 3

Multi-Objective Evolutionary Algorithms

Classical multi-objective programming methods are only some of the optimization methods used to solve MOPs. However, they have some disadvantages. Some of them are the following:

- Since those algorithms restate a MOP into a SOP, only one Pareto optimal solution can be found. Therefore, if the user is interested in finding multiple Pareto-optimal solutions, it is necessary to run those algorithms many times. Thus, in order to find N solutions, at least N single-objective optimization problems need to be solved.
- For some MOPs, classical methods can not be applicable or may have poor performance, for example, problems in which the objective functions are non-differentiable.
- Many of them are sensitive to the shape or continuity of the Pareto front.
- Most classical algorithms require some knowledge about the problem to be solved.

Such disadvantages have motivated the use of alternative approaches to tackle different kinds of MOPs. Among those approaches, Evolutionary Algorithms (EAs) have become a popular alternative to classical optimization methods.

EAs are stochastic search and optimization methods inspired by the natural evolution process. They are techniques that operate on a set of solutions (population of solutions). At each iteration, EAs implement a selection mechanism to choose the best solutions and perform a reproduction process to generate new solutions.

In 1967, Rosenberg [81] introduced the use of genetic algorithms to solve MOPs. However, it was until 1984, when Schaffer [82] proposed the first implementation of what it is now called a Multi-objective Evolutionary Algorithm (MOEA). After that, different MOEAs have been proposed and applied to several optimization problems [24, 49, 57, 101, 111]. Some of the major advantages of MOEAs, as compared to other methods, are that they require little problem specific knowledge and they are not very susceptible to the shape or continuity of the Pareto front. In addition, MOEAs are easy to implement and use, and they can generate several elements of the Pareto optimal set in a single run.

According to different authors, finding an approximate Pareto front is, by itself, a bi-objective problem whose objectives are [20, 111]:

- Minimize the distance of the generated solutions to the POF, and
- Maximize the diversity among the solutions in the obtained POF as much as possible.

Single objective EAs and MOEAs share a similar structure. However, since MOEAs deal with more than one objective at the same time, MOEAs must use a fitness assignment mechanism which considers the two objectives presented above. MOEAs can be classified in several ways [20]. However, for the purposes of this thesis, a simple high-level classification based on their selection mechanism is adopted. In the following, some of the most popular MOEAs are presented.

3.1 MOEAs based on Pareto

- **Multiple Objective Genetic Algorithm (MOGA):** This algorithm was proposed in 1993 by Fonseca and Fleming [34]. It is based on the ranking scheme proposed by Goldberg [39]. In such ranking scheme, first, all individuals in the population are ranked based on non-dominance. Thus, the rank of an

individual x at generation t is equal to the number of individuals by which it is dominated plus one, i.e. $rank(x,t) = 1 + p(x,t)$, where $p(x,t)$ is the number of individuals that dominate individual x in the objective space. After the ranking scheme, the fitness of all individuals is computed. In this algorithm, in order to obtain a good distribution of solutions along the Pareto front, fitness sharing is implemented. The sharing mechanism used by MOGA calculates its value depending on the current maximum and minimum values of the objectives and the population size.

- **Non-dominated Sorting Genetic Algorithm (NSGA):** In this algorithm the ranking scheme proposed by Goldberg is also implemented but in a more straightforward way. The NSGA algorithm was proposed by Srinivas and Deb [86]. It ranks the population in different layers or fronts based on non-dominance. Thus, before applying the selection mechanism, the population is ranked. In such ranking scheme, the first front is composed by the non-dominated individuals of the current population. The second front, is composed by the non-dominated individuals excluding individuals in the first rank. Therefore, each front is computed ignoring the individuals that have already been ranked. The ranking scheme continues until all individuals in the population are ranked. Since individuals in the first front are the best ranked individuals, they have a higher selection probability than the rest of the population.

- **Non-dominated Sorting Genetic Algorithm-II (NSGA-II):** This algorithm was proposed by Deb et al. [24] and it is an improved version of the NSGA algorithm. In the NSGA-II, individuals of the current population are ranked and sorted according to its non-domination level. After that, NSGA-II applies evolutionary operators to create an offspring pool. Once the offspring has been created, the parents and offspring are combined to create a new population. After that, the Pareto-ranking is performed on the new population, i.e. the combined population is sorted according to the individuals' level of non-domination. For each ranking level, the sum of the Euclidean distances between the two neighboring solutions from either side of the solution along each of the objectives is computing. This value is the so-called crowding distance

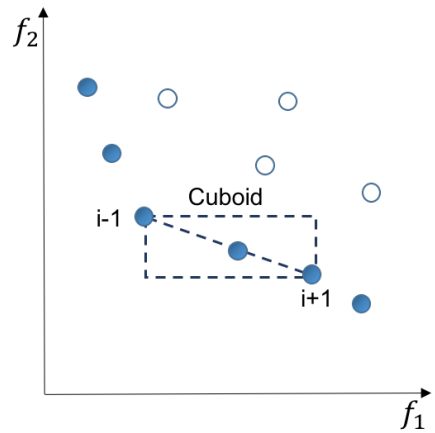

Figure 3.1: Representation of Crowding Distance.

(see Figure 3.1). During selection, the NSGA-II uses a crowded-comparison operator which guides the selection process toward to a uniformly spread-out Pareto-optimal set. Such crowded-comparison operator takes into consideration both the non-domination rank of an individual in the population and its crowding distance. Between two solutions with different non-domination rank, the solution with the lower (better) rank is preferred. Otherwise, if both solutions belong to the same front, then the one that resides in the less crowded region is preferred. Since the NSGA-II uses an elitist mechanism that consists in combining the best parents with the best offspring obtained i.e., a $(\mu + \lambda)$-selection, it does not use an external memory as other MOEAs. In the last few years, the NSGA-II algorithm has been the most popular MOEA, and it is commonly adopted to compare the performance of newly introduced MOEAs. In Figure 3.2 the general behavior of the NSGA-II algorithm is presented.

- **Strength Pareto Evolutionary Algorithm (SPEA):** The SPEA algorithm was introduced by Zitzler and Thiele [111]. This MOEA uses a secondary population (an external archive, the so-called external non-dominated set) containing non-dominated solutions previously found. At each iteration, non-dominated solutions are copied to the external non-dominated set, removing the dominated solutions. In SPEA, the fitness of each individual in the primary population is computed by using the individuals of the external archive.

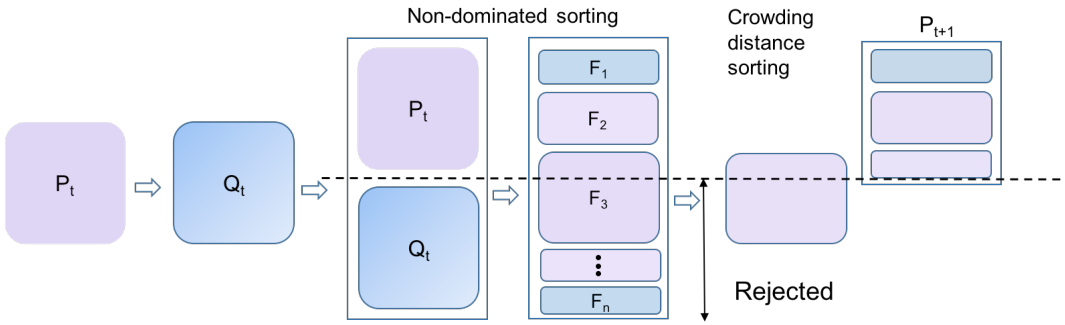

Figure 3.2: General procedure of NSGA-II algorithm.

First, a strength value is calculated for each individual in the external set. This strength value is similar to the ranking value of MOGA [34]. After that, the fitness of each individual in the primary population is computed as the sum of the strengths of all the external non-dominated solutions that dominate it. In SPEA, the fitness assignment considers both, closeness to the true Pareto front and even distribution of solutions at the same time. On the other hand, SPEA uses a clustering technique called average linkage method [70] maintaining the size of the external non-dominated set below a certain threshold to avoid that its size grows too large.

- **Strength Pareto Evolutionary Algorithm 2 (SPEA2):** This MOEA is an improved version of SPEA algorithm proposed by Zitzler et al. [108]. It has three main differences with respect to the SPEA previously described: (1) it has a fine-grained fitness assignment strategy which takes into consideration, for each individual, the number of individuals that dominate it and the number of individuals to which it dominates; (2) it uses an adaptation of the k-th nearest neighbor method, the so-called nearest neighbor density estimation technique, which guides the search more efficiently, and (3) it incorporates an enhanced archive truncation method that guarantees the preservation of boundary solutions.

- **Generalized Differential Evolution (GDE3):** This algorithm was proposed by Kukkonen et al. [57] as an extension of Differential Evolution algorithm to solve MOPs with constraints. Since GDE3 is the MOEA algorithm

used as a basis in our proposed approach, in this section a detailed description of such algorithm is presented. First, a brief description of DE is presented, followed by the detailed description of GDE3.

Differential Evolution (DE) is an evolutionary algorithm with a mechanism to generate multiple search directions based on the distribution of solutions (vectors) in the current population [68]. The population of solutions in DE is represented as: $x_{i.G}, i = 1, \ldots, NP$, where $x_{i.G}$ represents one vector i at generation G, NP is the population size.

There are different DE variants which are distinguished by the way of generating new vectors. The most popular of them is called *DE/rand/1/bin*, where "rand" means the criterion to choose the base vector, "1" refers to the number of vector differences to be computed, and "bin" is the type of the crossover operator, in this case, binomial crossover [75]. This DE variant is described below.

At the moment of reproduction, each vector in the population called *target vector or parent vector*, $\vec{x}_{i,G}$, generates one offspring, *trial vector* $\vec{u}_{i,G}$, using a vector called mutant as follows:

First, a search direction is defined by calculating a *difference vector* between a pair of vectors chosen randomly from the current population $\vec{x}_{r_1,G}$ and $\vec{x}_{r_2,G}$. This difference vector is scaled using a user-defined parameter called *scale factor* $F > 0$ [75]. After that, this scaled vector is added to a third vector (called base vector) $\vec{x}_{r_0,G}$ chosen randomly as well. The three vectors used for this operation are different from each other and different from the target. The result of these operations is a new vector known as *mutant vector* $\vec{v}_{i,G}$ (mutation operation) as shown in Equation 3.1.

$$\vec{v}_{i,G} = \vec{x}_{r_0,G} + F(\vec{x}_{r_1,G} - \vec{x}_{r_2,G}) \tag{3.1}$$

After the mutant vector is generated, it will be recombined with the target vector, based on another user-defined parameter called *crossover probability* $0 \leq CR \leq 1$, to generate a *trial (child) vector*. CR defines the similarity between the mutant and trial vector (Equation 3.2)

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & if(rand_j \quad \leq \quad CR) \quad or \quad (j = J_{rand}) \\ x_{i,j,G} & otherwise \end{cases} \tag{3.2}$$

In Equation 3.2, $rand_j$ generates a random real number with uniform distribution between 0 and 1, $j \in 1, \ldots, D$ is the $j-th$ variable of the D-dimensional vector and $J_{rand} \in [1, D]$ is a random integer which prevents a target vector copy as its trial.

To finish the evolutionary process, the best vector between the target and trial, considering the objective function value and assuming minimization, is chosen to remain in the population for the next generation according to Equation 3.3:

$$x_{i,G+1} = \begin{cases} u_{i,G} & if(f(u_{i,G}) \quad \leq \quad f(x_{i,G})), \\ x_{i,G} & otherwise \end{cases} \tag{3.3}$$

A graphical example of the variant described above is shown in Fig. 3.3.

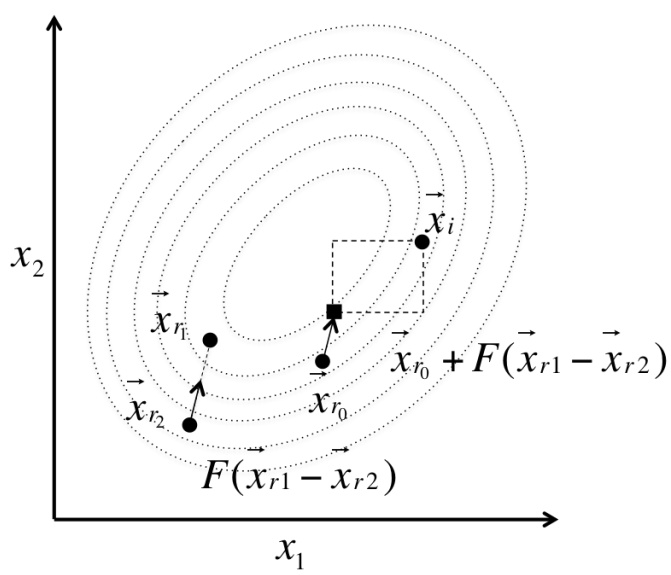

Figure 3.3: DE/rand/1/bin graphical example [68].

Regarding GDE3 algorithm, different previous versions of GDE have been proposed in [59] and [58]. The first one was an expansion of DE to deal with con-

strained multi-objective problems. Its basic idea was only the modification of the selection rule of the basic DE. Crowding distance and non-dominated concepts were introduced in the second version. Finally, the last version (adopted in our proposed approach), is an improved version of the two previous GDE approaches to solve problems with M objectives and K constraints functions. Therefore, it handles any number of M objectives and any number of K constraints, including the cases where $M = 0$ (constraint satisfaction problem) and $K = 0$ (unconstrained problem). One main characteristic in GDE3 is the selection mechanism used, which is based on the following criteria:

- Between two infeasible vectors, the trial vector is selected if it dominates the target in constraint violation space, otherwise the target is selected.
- Between one feasible vector and one infeasible vector, the feasible vector is selected.
- Between two feasible vectors, the trial is selected if it dominates the target. If the target dominates the trial vector, then the target is selected. If neither vector dominates each other, both vectors are temporarily selected for the next generation.

When two non-dominated feasible vectors are chosen to survive, the population size might grow through generations. When this occurs, it is necessary to reduce the population to its original size. For such purpose, the algorithm uses the same selection mechanism of NSGA-II, which consists in truncating the population based on crowding distance (CD) and non-dominance criteria. This allows the best members of the population to remain for the next generation, and the worst members are removed to decrease the size of the population to the original size. The non-dominated sorting is modified as well, to handle constraints, and the selection based on crowding distance is improved to provide a better distribution of solutions. In the original selection mechanism, for selecting n solutions out of N (current population), the solutions are first sorted according to objective values. Then crowding distances (CDs) are calculated. Finally, solutions are sorted according to the crowding distance values. However, there are cases where this approach does not provide good results

[56]. Therefore, to deal with this problem, instead of selecting n solutions with the largest crowding distance values, the selection mechanism implemented in GDE3 first calculates CDs of every solution (n) of the non-dominated set. After that, $N - n$ solutions with the smallest crowding distance values are removed one by one. Removing these solutions from the front, crowding values from the remaining solutions change and these values are updated every time solutions are eliminated. Algorithm 1 shows the pseudocode of the GDE3 selection mechanism.

Algorithm 1 Pseudocode of the Selection mechanism of GDE3 [56]

```
Input: a non-dominated set S, the size n of a desired pruned set
Output: elements of a heap H
for i = 1 to |S| do
    Chose a member x_i ∈ S and calculate it CD value;
    Create a data structure D containing information about neighbors on either side
  of the member x_i along each objective;
end for
Using CDs values as ordering key, create an ascending heap H from the members of S;
while |H| > n do
    Remove a solution x_i with a minimum CD value from H and update H;
    Update D with the correct neighbor information for the neighbors of the removed
  element;
    ∀ neighbors of the removed element: calculate a new CD;
    Replace old CD value in H with the new one and update H;
end while
```

In unconstrained single objective problems, GDE3 performs equal to the original DE. The general pseudocode of GDE3 is shown in Algorithm 2.

3.2 MOEAs based on decomposition

- **Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D):** This algorithm was proposed by Zhang and Li [101]. It is well-known that a Pareto optimal solution to a MOP, under certain conditions, could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the objective functions. Therefore, an approximation of the Pareto optimal front can be decomposed into a number of

Algorithm 2 General pseudocode of GDE3 algorithm [57]

1: **Input:** D, G_{max}, NP, CR, F, and initial bounds $\bar{x}^{(lo)}$, $\bar{x}^{(hi)}$
2: Initialize population $x_{i,0} | i = \{1, 2, , \ldots, NP\}, j = \{1, 2, , \ldots, D\}, G = 0, m = 0$;
3: Evaluate the initial population;
4: **for** G= 1 to G_{max} **do**
5: **for** i= 1 to NP **do**
6: Choose $r_0 \neq r_1 \neq r_2 \neq i$ from current population $\mathbf{P}$;
7: Generate $j_{rand} = radint(1, D)$;
8: **for** j= 1 to D **do**
9: **if** j= j_{rand} or $rand_j[0, 1) < CR$ **then**
10: $u_{j,i,G} = \vec{x}_{j,r0,G} + F \cdot (\vec{x}_{j,r1,G} - \vec{x}_{j,r2,G})$ Mutation operator;
11: **else**
12: $u_{j,i,G} = x_{j,i,G}$;
13: **end if**
14: **end for**
15: **if** $\vec{u}_{i,G} \preceq_c \vec{x}_{i,G}$ **then**
16: $\vec{x}_{i,G+1} = \vec{u}_{i,G}$;
17: **else**
18: $\vec{x}_{i,G+1} = \vec{x}_{i,G}$;
19: **end if**
20: **if** $\forall_j : g_j(\vec{u}_{i,G}) \leq 0 \wedge \vec{x}_{i,G+1} == \vec{x}_{i,G} \wedge \vec{x}_{i,G} \not\prec \vec{u}_{i,G}$ **then**
21: $m = m + 1$;
22: $\vec{x}_{NP+m,G+1} = \vec{u}_{i,G}$;
23: **end if**
24: **end for**
25: **while** $m > 0$ **do**
26: Select $\vec{x} \in \{\vec{x}_{1,G+1}, \vec{x}_{2,G+1}, \ldots, \vec{x}_{NP+m,G+1}\}$:

$$\begin{cases} \forall_i \vec{x} \not\prec_c \vec{x}_{i,G+1} \\ \wedge \\ \forall (\vec{x}_{i,G+1} : \vec{x}_{i,G+1} \not\prec_c \vec{x}) \quad CD(\vec{x}) \leq CD(\vec{x}_{i,G+1}) \end{cases}$$

27: Remove $\vec{x}$;
28: $m = m - 1$;
29: **end while**
30: **end for**

scalar optimization subproblems. This is the basic idea behind many traditional mathematical programming methods (see Chapter 2 of this document). The same idea is adopted by MOEA/D, i.e., it decomposes the MOP into a number of scalar objective optimization subproblems (SOPs). In the evolution process, neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Subproblem i is a neighbor of subproblem j if the weight vector of subproblem i is close to that of subproblem j. MOEA/D optimizes these subproblems using information only from its neighboring subproblems. In MOEA/D an external archive is also used to store the non-dominated solutions found during the search. In a simple version of MOEA/D, each subproblem keeps one solution in its memory, which could be the best solution found so far for the subproblem. A new solution is generated by applying genetic operators among pairs of solutions from its neighboring subproblems, after that, MOEA/D updates its memory if the new solution is better than the old one for the subproblem. One advantage of MOEA/D different from NSGA-II and SPEA2 algorithms is that MOEA/D uses the well-distributed set of weight vectors for guiding the search. Therefore, the diversity of the population in MOEA/D is implicitly maintained. In contrast, NSGA-II and SPEA2 use density estimators, crowding distance and neighboring solutions, respectively.

3.3 Indicator-based MOEAs

Since a MOP can be solved using different MOEAs, different approximations to the Pareto-optimal set can be found. Therefore, several performance measures have been proposed to evaluate and compare the outcome sets of MOEAs. In this sense, more recently designed MOEAs have considered the use of performance measures or indicators to guide the search during the evolution process [7, 31, 110]. The main motivation for the design of MOEAs based on indicators has been to overcome the poor performance of the Pareto-based selection schemes when dealing with MOPs having more than three objectives [54]. Such indicators can be incorporated into a MOEA in different ways: (1) as an archiving algorithm, (2) as a selection mechanism,

and (3) as a set preference relation. However, for the purposes of this thesis, we are only focussed on MOEAs that incorporate the indicators as a selection mechanism. Here we refer to this type of MOEAs as Indicator-based MOEAs. In the following, a brief description of the most popular Indicator-based MOEAs is presented.

- **Indicator-Based Evolutionary Algorithm (IBEA):** This algorithm was proposed by Zitzler and Künzli [110]. The main idea of IBEA is to first define the optimization goal in terms of a binary performance metric and then use the metric in the selection process. Since IBEA is considered as a general framework of Indicator-based MOEA, any binary metric can be used in the fitness assignment function. IBEA assigns the fitness of each individual in the population using: $f(\vec{x}) = \sum_{\vec{y} \in P \setminus \{\vec{x}\}} -e^{-I(\{\vec{x}\},\{\vec{y}\})/k}$, where P is the current population, $\vec{x}, \vec{y} \in P$, $I(\{\vec{x}\},\{\vec{y}\})$ represents the binary quality indicator, and k is the scalar factor which is defined by the user and depends on the problem being solved.

  In general, IBEA performs binary tournaments for mating selection and implements environmental selection by iterative removing the worst individuals from the current population, in terms of the selected binary quality indicator. When the worst individual is eliminated, IBEA updates the fitness values of the remaining individuals with: $f(\vec{x}) = f(\vec{x}) + e^{-I(\{\vec{x}'\},\{\vec{x}\})/k}$, where $\vec{x}'$ is the eliminated individual.

- **S-metric Selection-Evolutionary Multi-Objective Algorithm (SMS-EMOA):** This algorithm was originally proposed by Emmerich et al. [31]. In this algorithm, the hypervolume (or S-metric) contribution is used in the environmental selection process. SMS-EMOA creates an initial population, then, only one solution by iteration is created using the operators (crossover and mutation) of the NSGA-II algorithm and inserted into the current population. After that, SMS-EMOA applies Pareto ranking. Then, for each solution in the current population, its contribution to the hypervolume is computed. Since the maximization of the hypervolume metric attains both, convergence to the Pareto-optimal front and a good distribution of solutions along the Pareto-front approximation, the solution with the less contribution to the HV metric is then

discarded. At the beginning of the evolutionary process, some solutions in the current population can be dominated and, therefore, they do not contribute to the hypervolume metric of the Pareto-front approximation. In such cases, Beume et al. [7] proposed that the SMS-EMOA algorithm computes the hypervolume contribution for each rank layer of solutions in the Pareto-ranking. Therefore, the discarded solution will be selected as the less contributing in the hypervolume metric but in the highest rank layer.

- **Hypervolume Estimation Algorithm for Multi-objective optimization (HypE):** HypE is another novel HV-based MOEA that can be used for solving MOPs with an arbitrary number of objectives. It was proposed by Bader et al. [6]. In HypE, to reduce the computational cost of HV calculations, instead of calculating the exact HV values, the Monte Carlo simulation has been adopted to estimate the approximate HV values. In the environmental selection of HypE, the non-dominated sorting is used to divide the population into several fronts, after that, the solutions on the last front are distinguished by their contributions to the HV values of the population.

3.4 Other meta-heuristics

There exist other multi-objective meta-heuristics that have been proposed. Next, some of them are briefly described:

- **Particle Swarm Optimization (PSO):** The PSO algorithm was proposed by Kennedy and Eberhart [53]. This algorithm is inspired by the choreography of bird flocks. The implementation of the algorithm adopts a population of individuals called particles. During the optimization process, the behavior of each individual is affected by either the best local (i.e., within a certain neighborhood) or the best global individual. Different from traditional evolutionary algorithms, in order to accelerate converge, PSO introduces the use of an operator that sets the velocity of a particle to a particular direction. This can be seen as a directional mutation operator. Another difference is that PSO allows individuals to benefit from their past experiences. PSO algorithm

has been successfully used for both continuous nonlinear and discrete binary optimization [32]. Since the high speed of convergence that PSO presents solving single-objective optimization problems, it has also been extended for multi-objective optimization. To deal with MOPs, normally, mechanisms very similar to those adopted with MOEAs (namely, Pareto ranking scheme, mutation operators and external archives) have been adopted in multi-objective particle swarm optimizers (MOPSOs). Several multi-objective versions of PSO have been proposed (see for example [79]).

- **Artificial Immune Systems (AIS):** The AIS are computational paradigms inspired by the biological immune system [14]. Therefore, they belong to the nature-inspired meta-heuristics. One of the main goals of the immune system consists essentially of a process where different cells interact each other to protect the organisms from intruding pathogens or bacteria (antigens). The immune system is capable of distinguishing between normal and foreign components in the organism. Cells that are recognized as foreign material are known as antigens.

  The most popular paradigm of the immune system that attempts to explain the process whereby the antigens are eliminated is called Clonal Selection Principle [14]. In Clonal Selection, the molecules called antibodies play an important role, when an antigen is detected, the antibodies that best recognize an antigen will proliferate by cloning. The new cloned cells experiment an hypermutation process, according to the affinity to the antigen, antibodies with highest affinity experiment lowest mutation and antibodies with the lowest affinity have high mutation rates. The hypermutation process in the immune response is important because it allows the creation of new antibodies and maintains the diversity. Once these clonation and hypermutation processes finish, the immune system has improved the antibodies' affinity, which results on the antigen neutralization and elimination. Finally, the immune system needs to return to its normal condition, eliminating the excess cells. However, there are cells, called memory cells, which can be activated when the organism is later attacked by a similar type of antigen, and these cells present a better efficient response [14].

In the same way of PSO, AIS have also been extended for solving MOPs. Thus, very similar mechanisms to those adopted by MOEAs have been used to proposed multi-objective optimization algorithms inspired on the artificial immune system (MOAIS). The first multi-objective optimization algorithm inspired on the AIS was called MISA (Multi-objective Immune System Algorithm) proposed by Cruz and Coello [18]. In MISA the Pareto ranking scheme was used to deal with MOPs and the authors attempted to follow the clonal selection principle very closely, then the performance of MISA was improved in a successive version [22]. After the proposal of MISA, several MOAIS have been proposed (see for example [12] and [35]), and this remains as a very active research area.

3.5 Performance assessment of MOEAs

In multi-objective optimization, the comparison of the performance of different MOEAs is an important issue. Different from single-objective optimization, where the quality of a solution can be defined using the objective function values: the smaller (to minimize) or the larger (to maximize) value corresponds to a better solution, in multi-objective optimization, other aspects should be considered to evaluate the performance of MOEAs.

As pointed before, MOEAs should be designed to satisfy the two main goals of multi-objective optimization: (1) minimize the distance of the approximated POF produced to the true POF, and (2) maximize the diversity among the solutions in the Pareto front approximation as much as possible. Therefore, to assess the performance of the MOEAs, several performance measures have been proposed which considered the two above issues. In the following, the performance measures which are referred to in this thesis are briefly described. In such definitions, we considered POF as the optimal Pareto front and POF* as an approximate Pareto front.

- **Inverted Generational Distance (IGD) [77]:** IGD metric measures both diversity and convergence of found solutions by an algorithm [99], [103], [51]. Hence, if the diversity of an algorithm is not good, then its IGD value can be affected. The IGD is computed as in Equation 3.4:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{3.4}$$

where n is the number of uniformly distributed points in the POF and d_i is the Euclidean distance between the ith solution member in the POF and its nearest member in the approximated Pareto front (POF*) obtained by an algorithm. A value of $IGD = 0$ is preferred because a low value indicates that the solutions generated by an algorithm are very close and cannot miss any part of the whole POF. Therefore, any other value will indicate how "far" an algorithm is from the POF in a given test problem. In the case of IGD, the POF is used as a reference, and each one of its elements is compared with respect to the POF*.

- **Hypervolume (HV):** This performance measure was originally proposed by Zitzler and Thiele in [93]. It quantifies both convergence and spread of non-dominated solution along the POF. Let POF* be a Pareto front approximation obtained by an algorithm and a reference point in objective space z_{ref}, the hypervolume indicator measures the space covered by the set of solutions Q_t of the POF* in the objective space. The hypervolume corresponds to the non-overlapping volume of all the hypercubes formed by the reference point z_{ref} and each solution in the POF*. The hypervolume measure of POF* is calculated as indicated in Equation 6.4:

$$HV = \bigcup_{i=1}^{Q} \{vol_i | vec_i \in POF^*\} \tag{3.5}$$

  where vec_i is a non-dominated vector from POF*, and vol_i is the volume for the hypercube formed by the reference point and the non-dominated vector vec_i.

  For static multi-objective optimization problems, a high HV value, indicates that the approximation is close to the POF and has good spread towards the extreme portions of the POF.

- **Spacing (S)**: The spacing metric was introduced by Schott [83]. It measures

how evenly the solutions obtained by an algorithm approaximation (POF*) are distributed in the objective space, and it is calculated as indicated in Equation 3.6.

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(D_i - \overline{D})^2} \tag{3.6}$$

$$\overline{D} = \frac{1}{n}\sum_{i=1}^{n} D_i \tag{3.7}$$

where D_i is the Euclidean distance between the ith member in POF* and any other solution in POF*, and $\overline{D}$ is the average of all D_i values. If $S = 0$, the non-dominated solutions of POF* are uniformly spread or spaced [25].

- **Maximum Spread (MS)**: The maximum spread performance metric, first introduced by Zitzler [113], measures to what extent the extreme solutions in POF* have been reached. Goh and Tan [37] proposed a modified version of MS which measures how much the obtained POF* covers the Pareto optimal front (POF), and it is calculated as indicated in Equation 3.8

$$MS = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left[\frac{\min\left[\overline{POF_k}, \overline{POF_k^*}\right] - \max\left[\underline{POF_k}, \underline{POF_k^*}\right]}{\overline{POF_k^*} - \underline{POF_k^*}}\right]^2} \tag{3.8}$$

where $\overline{POF_k}$ and $\underline{POF_k}$ are the maximum and minimum values of the kth objective in POF, respectively. $\overline{POF_k^*}$ and $\underline{POF_k^*}$ are the maximum and minimum values of the kth objective in POF*. Larger values of MS indicate a good spread of POF*, and MS will have a value equal to one when POF* covers the whole POF.

- **Two-set Coverage (C-Metric)**: The Two-set-coverage metric was introduced by Ziztler et al. [111]. It is a binary performance metric which estimates the coverage proportion, in terms of percentage of dominated solutions, between two POF*. Given two approximate Pareto fronts POF*, X' and X'', both containing only feasible non-dominated solutions, the C-Metric is formally defined as indicated in Equation 3.9.

$$C(X', X'') = \frac{|\{a'' \in X''; \quad \exists a' \in X' : \quad a' \preceq a''\}|}{|X''|} \tag{3.9}$$

Where X' and X'' are POF* obtained by different algorithms, If all the points in X' dominate or are equal to all points in X''', then by definition C = 1, C = 0 otherwise. Therefore, the C-metric value means the portion of solutions in X'' being dominated by any solution in X'. Note that the domination operator is not a symmetric operator, i.e. $C(X', X'')$ it is not necessarily equal to $1 - C(X'', X')$. Therefore, if many algorithms are compared against each other, the C-metric needs to be evaluated in both directions of the different POF* approximations for each possible combination of algorithms.

3.6 Summary

This chapter provided a brief summary of the most popular multi-objective evolutionary algorithms. Also, a classification of MOEAs based on their selection mechanism was presented. In Section 3.1, different MOEAs based on Pareto ranking were described, including the GDE3 algorithm which is a basis of the dynamic MOEA proposed in this thesis. The MOEA/D algorithm was presented in Section 3.2. Section 3.3 discussed indicator-based MOEAs. Section 3.4 provided information about other nature-inspired algorithms, namely PSO and AIS, which are required as background for later chapters in the thesis. Finally, different performance metrics for performance assessment of MOEAs were described in Section 3.5.

Since real-world optimization problems are not static in nature and change over time, the next chapter introduces both, dynamic single-objective optimization problems and dynamic multi-objective optimization problems. Furthermore, the state-of-the-art regarding dynamic multi-objective evolutionary algorithms is provided.

Chapter 4

Dynamic Evolutionary Multi-Objective Optimization

In many real-world situations, an optimization problem may present changes in the objective function(s) and/or constraints. Such changes, lead to a change in the search landscape and/or in the feasible space. Therefore, the optima of the problem can change in position or value, or the optima can disappear while a new optimum appears. These kinds of problems are usually known as dynamic optimization problems. According to the number of objectives to be optimized, dynamic optimization problems can also be classified into dynamic single-objective optimization problems (DSOPs) and dynamic multi-objective optimization problems (DMOPs).

4.1 Dynamic Single-Objective Optimization Problem

Mathematically, a DSOP can be formulated as follows:

Definition 14. Dynamic single-objective optimization problem: Find $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ which minimizes:

$$f(\vec{x}, t) \tag{4.1}$$

subject to:

$$g_i(\vec{x}, t) \leq 0, \quad i = 1, 2, \ldots, m$$
$$h_j(\vec{x}, t) = 0, \quad j = 1, 2, \ldots, p$$

where $\vec{x}$ is the vector of decision variables and t represents the time step.

Since the optima change over time, the goal of a dynamic optimization algorithm is to find the optimum $\vec{x}^*$ and track its trajectory as soon as possible.

4.2 Dynamic Multi-objective Optimization

Real-world optimization problems may have more than one objective, and they can present environmental changes. This kind of problems are usually known as dynamic multi-objective optimization problems (DMOPs). Since solving DMOPs is a fundamental part of our research interest, in this section, the theory and definitions with regards to DMOPs are presented.

Definition 15. Dynamic multi-objective optimization problem: Mathematically, a DMOP can be formulated as follows [77]:

Minimize:

$$\vec{f}(\vec{x}, t) = [f_1(\vec{x}, t), f_2(\vec{x}, t), \ldots, f_k(\vec{x}, t)]^T \tag{4.2}$$

subject to:

$$g_i(\vec{x}, t) \leq 0 \quad i = 1, 2, \ldots, m$$
$$h_j(\vec{x}, t) = 0 \quad j = 1, 2, \ldots, p$$

where $\vec{x}$ is the vector of decision variables, $\vec{f}$ is the set of objective functions to be minimized with respect to the variable time t, t is the discrete time instance defined as $t = (1/n_t) \lfloor (\tau/\tau_\tau) \rfloor$, where n_t, τ_τ and τ represent the severity of change, the frequency of change, and the iteration counter, respectively. The functions g and h, represent the set of constraints, which define the feasible region $\mathcal{F}$ of the feasible space solutions that change with respect to the time t.

Definition 16. Frequency of change τ_τ: In a time-dependent problem, the frequency of change determines how often the environment changes. Usually, it is measured as the number of generations or the number of fitness functions evaluations from one landscape change to the next [80].

Definition 17. Severity of change n_t: The severity of change means how fundamental the changes are in terms of their magnitude, i.e., it measures the difference of the landscape change by comparing the landscape before and after a change [80].

When solving DMOPs, the change frequency and change severity parameters control the changes in the environment. The change frequency and severity analysis are important tasks in DMO, because they allow evaluating the performance of Dynamic Multi-objective Evolutionary Algorithms (DMOEAs) for a specific type of environment, i.e., whether a DMOEA performs well in fast-changing environments, slow-changing environments, or both; gradually changing environment, severely changing environment, or a combination of these listed environment types [47].

4.2.1 Dynamic environment types

Based on the relation between the POF and the POS, Farina et al. [33] classified the dynamic environments for DMOPs in four types:

- **Type I:** The POS changes, whereas the POF (optimal objective values) does not change.
- **Type II:** Both POS and POF change.
- **Type III:** POS does not change, whereas POF changes.
- **Type IV:** Both POS and POF remain unchanged with time, but other changes in the problem definition induce dynamicity.

These four environment types are summarized in Table 4.1

In this work, DMOPs with the first three types of changes indicated above, without constraints, were considered for our research. When a change occurs in the environment, the POF can change over time in different ways [33]:

Table 4.1: Dynamic environment types

.

POF	**POS**	
	No change	Change
No change	Type IV	Type I
Change	Type III	Type II

- Existing solutions in the POF becomes dominated and therefore are not part of the POF any more.
- The shape of the POF can change over time from convex to nonconvex and/or viceversa. The shape of the POF changes from a continuous front to a disconnected front. These kind of changes are common with either type II or type III DMOPs.
- The shape of the POF remains the same, but its location in the objective space changes over time. This kind of change occurs with type I DMOPs.
- The density of the solutions in the POF changes over time. This kind of change can occur in all types of DMOPs.

4.3 Dynamic Multi-objective Optimization using Evolutionary Algorithms

As mentioned previously, DMOPs are optimization problems with multiple objectives with at least one objective changing over time. In the same way of multi-objective optimization, the objectives in a DMOP are in conflict with one another. Therefore, a DMOP has a set of trade-off solutions called the POF. The main difficulty in dynamic multi-objective optimization is that the POF of a DMOP may change when the environment changes. Therefore, the search then requires a fast convergence in the current problem conditions and also quick responses after changes. In this way, in order to solve a DMOP, an optimization algorithm must be able to track the changing POF over time, apart from convergence and diversity.

In dynamic multi-objective optimization, as in dynamic single-objective optimization, it is important to maintain diversity in the population in order to improve the process of tracking the moving optima. For this reason, several mechanisms have been proposed to keep diversity in the population. Diversity can be either increased after a change or maintained throughout the run. Aside from diversity approaches, other approaches have been proposed to solve DMOPs, such as multiple population approaches, prediction-based approaches, memory-based approaches, etc. In this section, a review of some of the most popular EAs proposed to solve DMOPs is provided.

4.3.1 Diversity-based approaches

Convergence in dynamic multi-objective optimization could lead to different problems. One of them is that, when optimization algorithms have already converged to a particular area in the problem landscape, it could be difficult to find the new POF since convergence during the run promotes a lack of diversity. One way to overcome this problem is introducing diversity after detecting a change. Another way is to maintain a good diversity level over the search process. Some of the most widely used techniques to introduced or maintain diversity are: (a) hyper mutate the previous population, (b) reuse the previous population or the non-dominated solutions from the previous populations, (c) randomly generate new solutions (random immigrant), and (d) apply different crossover and mutation operators [107].

Diversity introduction approaches may not be effective when the problem changes are severe, fast or random. On the other hand, maintaining diversity could perform well when DMOPs have slow changes since it provides time for the optimization algorithm to converge. However, it might not be effective when the DMOP has only small changes [77].

One of the most important dynamic multi-objective evolutionary algorithm belonging to this kind of approaches is the so-called Dynamic NSGA-II (DNSGA-II) proposed by Deb [27]. To propose DNSGA-II Deb et al. extended the well-known multi-objective optimization algorithm NSGA-II to handle DMOPs. One of the modifications in the original algorithm is adding a way to detect problem changes by randomly re-evaluating 10% of the individuals in the population for each generation.

If there is a change in the objectives or constraint violation values, the problem is considered to be changed. Two dynamic versions of the proposed DNSGA-II called DNSGAII-A and DNSGAII-B were proposed. Their main difference is only the way of generating the initial population after a change. In the first case, the population is reinitialized while in the second the population is mutated depending on the type of change in the environment. The two versions were tested on a two-objective dynamic problem and applied to the problem of dynamic hydrothermal power scheduling.

More recently, Zeng et al. [100], proposed a dynamic orthogonal multi-objective evolutionary algorithm called DOMOEA. This approach selects randomly between a linear crossover operator and an orthogonal crossover operator. The linear operator is employed as a diversity maintenance scheme and the orthogonal operator is used to enhance the fitness of the population while the problem remains stabilized between changes. The weakness of this algorithm is that it is used mainly when the environment change occurs with a low frequency.

Based on AIS ideas, Shang et al. [84], proposed a clonal selection algorithm for DMOPs called CSADMO (Clonal Selection Algorithm for Dynamic Multi-objective Optimization). CSADMO uses clonal selection, a self-adaptive dynamic process of the immune system. A non-uniform mutation operator and the crowding distance measure are used by CSADMO to increase the diversity in the population.

In [95], Wang et al. introduced an adaptive immigration scheme which is then integrated to the differential evolution algorithm. The immigration strategy can be divided into correlated immigration, uncorrelated immigration and hybrid immigration. In the correlated immigration, the immigrants are generated from the previous solutions. However, for uncorrelated immigration, the immigrants are generated randomly. In the immigration scheme used by the authors, the immigrants are generated by both the correlated immigration and the uncorrelated immigration, so that it is called hybrid immigration.

Another way to maintain diversity consists in using diversity as an additional objective function [10]. In [16], Chen et al. proposed the Individual Diversity Multi-objective Optimization EA (IDMOEA) to explicitly maintain genetic diversity by considering it as an additional objective in the optimization process. The algorithm uses a new diversity preserving evaluation method which is called Individual Diversity

Evolutionary Method (IDEM). The goal of IDEM is to add a useful selection pressure addressed towards both, the optimal POS and the diversity maintenance.

Based on swarm intelligence, Díaz-Manríquez et al. [65], proposed the DPSO (Dynamic Particle Swarm Optimization) algorithm. It incorporates a hyperplane distribution and Pareto dominance to solve DMOPs. When a change is detected, DPSO reinitializes (in different ways) the velocity parameter of PSO and the file where non-dominated solutions are stored.

4.3.2 Multi-population approaches

In multiple population approaches, to detect changes or new optimal solutions, different parts of the search space are simultaneously explored by different subpopulations. For instance, one population can be in charge of the current solutions while another population would explore different regions. In this kind of approaches, different tasks should be assigned to each subpopulation and the subpopulations should not converge to the same location in the search space. Multiple population approaches have the advantage of being able to track multiple optima, recall previous optima, effective in solving multi-modal problems and can adequately adapt whenever the problem changes. However, one disadvantage of this kind of approaches is that the number of sub-population could affect the performance of the optimization algorithm [77].

In [38], Goh and Tan proposed a Dynamic Competitive-Cooperative Coevolutionary Algorithm (dCOEA). In dCOEA, the cooperative and competitive mechanisms work together to promote an adaptive problem decomposition to solve static and dynamic multi-objective optimization problems. The algorithm uses a temporal memory to exploit evolutionary results.

On the other hand, in [64], a similar approach called PNSCCDMO (novel cooperative coevolutionary dynamic MO optimization algorithm based on non-dominated sorting and prediction) was proposed. This algorithm also allows the decomposition of the optimization problem and each subcomponent will cooperate to evolve for better solutions. Another feature is that this algorithm is based on non-dominated sorting and a modified linear regression prediction strategy to get a rapid response to the new changes in the environment.

Another approach inspired on cooperative co-evolutionary algorithms was proposed by Xu et al. [98]. In such work, a cooperative co-evolutionary strategy based on environment sensitivities was used to solve DMOPs. In order to apply such strategy, first, the authors proposed to split all the decision variables into two subcomponents according to their interrelation with the environment. After that, two sub-populations are used to cooperatively optimize the two subcomponents, with the goal of generating the optimal solutions of the DMOPs. Finally, two prediction methods are employed to reinitialize the two subpopulations in order to respond to the changes in the environment. The strategy described above was incorporated into NSGA-II and a multi-objective particle swarm optimization algorithm, in order to propose two different DMOEAs called DNSGA-II-CO and DMOPSO-CO, respectively.

Helbing and Engelbrecht [46], proposed a Dynamic Vector Evaluated Particle Swarm Optimization algorithm (DVEPSO) to deal with DMOPs. In DVEPSO, one swarm is dedicated to solve one objective function and the algorithm detects if there is a change in the environment by re-evaluating sentry particles. If a change is detected, a portion of the particles is reinitialized by changing their position and re-evaluating their personal and neighborhood best.

Recently, Shang et al. [85], proposed a DMOEA called quantum immune clonal coevolutionary algorithm (QICCA). QICCA is a multi-population algorithm that uses the immune clonal function and clonal selection function of AIS. In QICCA, the coevolutionary competitive and cooperative operations were designed and incorporated into the immune clonal algorithm to enhance the uniformity and diversity of the non-dominated solutions and to enhance the exchange of information between the various populations. Moreover, a quantum updating operation was proposed to improve the search ability of the populations, ensuring a better distribution of solutions in the POF.

4.3.3 Prediction-based approaches

Aside from diversity maintenance, introduction techniques or multi-populations approaches, the prediction-based approaches are also used to solve DMOPs. In such approaches, a learning model is built to estimate the current change and the knowl-

edge is used to predict the next change in the problem landscape. The knowledge is also used to generate new individuals that best match the estimation of the next change.

In [43], Hatzakis and Wallace introduced the Dynamic Queuing Multi-objective Optimizer (D-QMOO). This algorithm exploits past information in order to predict the behavior of a DMOP in the future. When a change is detected, the location of the optimal solution is estimated by using a self-regressive model to predict the location of the optimal solutions.

Liu [62], proposed a Dynamic Multi-objective Evolutionary Algorithm with Core Estimation of Distribution (CDDMEA) that uses core estimation of a distribution model to predict the Pareto optimal solutions of the next environment. The performance of CDDMEA was compared against DNSGA-II-A. The author claimed that CDDMEA is better than DNSGA-II-A. However, the experimental results need a more thorough investigation through the use of different benchmark problems and more performance metrics.

Zhou et. al. [107], proposed a population prediction strategy (PPS) to predict the population of an EA after a change in the environment occurred. This approach consists in dividing the POS into two parts, namely a center point and manifold. When a change is detected, the next center point is predicted using a sequence of center points. Similarly, previous manifolds are used to estimate the next manifold. After that, PPS initializes the whole population by combining the predicted center and the manifold. The performance of PPS was evaluated by comparing the performance of three instances of RM-MEDA [102], which incorporate different mechanisms to respond to environmental changes, including the PPS strategy.

More recently, a new DMOEA based on Kalman filter (KF) predictions was developed by Muruganantham et al. [71]. The predictions help to guide the search towards the changed optima; thereby, the DMOEA quickly tracks the changing POF. Besides, a scoring scheme was devised to hybridize the KF prediction with a random reinitialization method.

In [97], a directed search strategy (DSS) was developed to improve the performance of NSGAII for solving DMOPs. The proposed DSS was incorporated into NSGAII with a differential evolution operator and form a new DMOA called NSGAII-

DE-DSS. In NSGAII-DE-DSS, two prediction-based methods were designed. The first one reinitializes the population based on the predicted moving directions once an environment change is detected. The second aims to accelerate the convergence by generating solutions in predicted regions of the Pareto set according to the moving direction of the non-dominated solutions between two consecutive generations.

Biswas et al. [8], proposed a variant of MOEA/D algorithm called MOEA/D-BR (MOEA/D + PS-based nearest distance + Basic Re-initialization strategy). The algorithm proposed uses the prediction model and the reinitialization scheme proposed in [106].

Based on a steady-state principle, Jiang et al. [52] proposed a new algorithm called steady-state and generational evolutionary algorithm (SGEA), which combines the fast and continuously tracking ability of steady-state algorithms and good diversity preservation of generational algorithms to solve DMOPs. MOEA/D-BR and SGEA algorithms will be described in detail later because these algorithms are used in the experimental design proposed for this thesis.

Prediction-based approaches could work very efficiently if their prediction is correct each time. But the problem with this type of approaches depends on how well the predictors are trained. Furthermore, the prediction approaches may not be always successful. Therefore, there is a need to combine prediction-based approaches with a diversity maintenance mechanism.

4.3.4 Memory-based approaches

Another commonly used strategy in dynamic optimization is to use memory schemes which implicitly or explicitly store relevant information from past generations to guide the future search. Using memory approaches is especially useful when optimal solutions repeatedly return to previous locations or when the environment has periodical changes. An implicit memory scheme uses a redundant coding representation that can store more information. The most common implicit memory scheme used in EAs is diploidy [40]. On the other hand, in explicit memory schemes, the information is stored in a memory separate from the population.

One disadvantage of memory-based approaches is that memory is very dependent on diversity and should be used in combination with diversity-maintenance ap-

proaches. Furthermore, redundancy of information using memory approaches could become a problem and may not necessarily promote diversity [77].

In [104], Zhang et al. proposed an algorithm based on an artificial immune system called Dynamic Constrained Multi-objective Optimization Artificial Immune System (DCMOAIS). DCMOAIS uses a T-module to detect changes in the environment and creates the initial population by using information from previous results. Once a change is detected, a B-module is used to find the Pareto optimal solution of the current environment. Then, the M-module stores the generated non-dominated solutions into a memory to which the T-module will use to generate the initial population whenever a new environment change is detected.

Wang and Li [96] proposed a new multi-strategy ensemble MOEA (MS-MOEA) to solve DMOPs. In such algorithm, the convergence speed is accelerated using a new offspring generation mechanism based on adaptive genetic and differential operators. MS-MOEA uses a Gaussian mutation operator to cope with premature convergence and a memory-like strategy to handle population reinitialization when a change takes place. The archive update in MS-MOEA is performed using the Fast Hypervolume (FH) strategy which consists in introducing the new solution in the archive only if it dominates an existing solution.

More recently in [4], Azzouz et al. proposed and adaptive hybrid population management strategy using memory, local search (LS) and random strategies to tackle DMOPs. The proposed strategy is based on a new technique that measures the change severity, according to which, it adjusts the number of memory, LS, and random solutions to be used. In addition, the authors also proposed a dynamic version of NSGA-II, called Dy-NSGA-II, within which they incorporated the aforementioned strategies.

4.3.5 Other approaches

In this section, other approaches that have been introduced for dynamic multi-objective optimization are described. It was decided to present such approaches apart from the proposed classification because they use different mechanisms to deal with DMOPs or they have been applied without using any additional mechanism for dynamism handling.

One of the first algorithms proposed to solve DMOPs was introduced by Farina et al. [33], and it was namely HMCEDA (Hybridized Minimal Cost Evolutionary Deterministic Algorithm). HMCEDA is a hybrid algorithm which uses an (1+1) evolution strategy as global optimization algorithm of the DMOP. The (1+1) evolution strategy (ES) is an EA that applies Gaussian mutation at each generation to one parent to create one offspring. In HMCEDA, once (1+1) ES starts to converge, a gradient-based algorithm or a simplex Nelder-Mead [72] search algorithm is used. HMCEDA can obtain some solutions close to the POF. However, in some cases, HMCEDA failed to converge towards the changing POF and struggled to find a diverse set of solutions. In addition, time consumptions are expensive.

In [63], Liu and Wang proposed a DMOEA called DMEA (Dynamic Evolutionary Algorithm). In such approach, the total time period of the DMOP is divided into smaller multiple time subperiods where each one is seen as a fixed environment. In DMEA, for each subperiod the DMOP is considered as a static MOP and for each of the MOPs, a MOEA is used to optimize the problem.

Inspired by the clonal selection principle, a micro-cloning local exploitation operator and an adaptive change reaction strategy were designed by Qian et al. to propose a DMOEA called mcDMOA [76]. The micro-cloning operator is applied to enhance the exploitation and exploration capabilities of the proposed algorithm, while the adaptive change reaction strategy is used to accelerate the ability of the algorithm to track the changing Pareto front.

4.4 Summary

In the first part of this chapter, the mathematical definitions of both DSOPs and DMOPs were introduced. In addition, different concepts regarding dynamic multi-objective optimization were presented.

This chapter also presented the state-of-the-art of dynamic multi-objective evolutionary algorithms (DMOEAs) that have been proposed for DMO. Four main categories of DMOEAs were identified, namely diversity-based approaches, multi-population approaches, prediction-based approaches and memory-approaches.

From the literature review presented in this chapter, it can be seen that many

DMOEAs have been proposed to solve DMOPs using evolutionary algorithms. However, other meta-heuristics as PSO and AIS have also been adapted for DMO. Moreover, most of the designed algorithms are focused on incorporating techniques that introduce or maintain diversity throughout the optimization process. Even though these kind of techniques are the most traditional and simplest way to deal with DMOPs, different studies have demonstrated that they can not be effective under different conditions commonly presented in DMOPs, for instance, when the problem changes are severe, fast, recurrent or random. For such reason, other mechanisms to solve DMOPs as the use of multi-populations, prediction techniques and memory-based approaches have also been proposed [77]. Therefore, the design of algorithms that uses some of these mechanisms or a combination of them is gaining attention.

Although different EAs have been used and adapted for DMO, only three algorithms that use Differential Evolution operators have been proposed to solve DMOPs [97],[96]. However, as it can be seen in the description of such DMOEAs, in two of them the global optimization algorithm used is based on other EAs as NSGA-II and MOEA/D. Therefore, DE operators only were adapted to replace the original variation operators of the algorithms. On the other hand, in [95] a DMOEA based on DE was introduced. However, its empirical validation its very limited.

Aside from DMO, in dynamic single-objective optimization, EAs and AIS have been widely used solving DSOPs [13, 42, 73, 91, 105]. Nevertheless, such combination has been scarcely explored when solving DMOP's. Moreover, in DMO, to the best of the author's knowledge, there is one EA which has not been yet extended to solve DMOPs. This is the so-called Generalized Differential Evolution 3 (GDE3) [57]. In this thesis, GDE3 is adapted to deal with dynamic objectives and some AIS concepts are also considered for such task.

In the following chapters of this thesis, two new DE-based algorithms for DMO, namely Immune GDE3 and DIDGE are proposed. Chapter 5 introduces the extensions made to GDE3 to propose the Immune GDE3 algorithm and analyzes its performance compared to state-of-the-art algorithms. Chapter 6 introduces the use of distance metrics to guide the search when solving DMOPs. Furthermore, in Chapter 6, DIGDE algorithm and its empirical validation are presented.

Chapter 5

Immune Generalized Differential Evolution

In the last few years, the development of dynamic multi-objective evolutionary algorithms is gaining attention. As it can be seen from Chapter 4, several approaches have been proposed. However, few works based on DE and inspired on AIS have been introduced. In addition, DE has especially attracted the interest from researches due to its excellent performance solving static optimization problems [68]. Nevertheless, DE had been little applied in dynamic optimization especially in Dynamic multi-objective optimization (DMO) [95]. On the other hand, AIS present different dynamic characteristics like adaptation, diversity promotion, dynamism, and detection. Moreover, to the best of the author's knowledge, no method is currently available for solving DMOPs using a combination of DE and AIS. Because of these reasons, in this chapter, a new dynamic multi-objective evolutionary algorithm called Immune GDE3 is introduced and validated. The proposed approach is based on the GDE3 algorithm and incorporates ideas from the clonal selection algorithm to form a complete competitive DMOEA.

5.1 Immune GDE3

To deal with dynamic multi-objective optimization, Immune GDE3 uses the GDE3 as search algorithm, when a change is detected in the environment, an immune re-

sponse based on the clonal selection principle [66] is activated. According to the framework proposed in [107], change detection, change reaction and MOP optimizer are the three main components in most existing dynamic multi-objective evolutionary algorithms, including the Immune GDE3 algorithm (See Algorithm 3). In Immune GDE3, the description of these three components is presented in the following subsections.

Algorithm 3 A General DMOEA Framework [107]

```
Set time step t = 0;
Initialize a population P^t;
while not terminate do
    if a change is detected then
        Set t = t + 1;
        Some change reaction;
    else
        Optimize the t − th MOP by using an multiobjective evolutionary algorithm;
    end if
end while
```

5.1.1 Change detection

To detect when a change has occurred, a reevaluating solution method is used, such that, at each iteration, Immune GDE3 selects a percentage of solutions from the current population to reevaluate them; their objective values are compared against their previous values (a tolerance of 1.0e-4 is used to detect changes). If there is a change in any objective function value, then it is established that there was a change in the DMOP. Therefore, the reevaluation solution method used in this work is only applied in the objective functions. The percentage of solutions selected is a user defined parameter. It is important to emphasize that in this kind of DMOPs, the number of iterations (generations) of the algorithm and the time t are related, i.e. to detect changes, a certain number of generations must be computed (represented by the τ_t parameter) and after that, the change detection mechanism is applied. The pseudocode of the change detection mechanism is detailed in Algorithm 4.

Algorithm 4 *ChangeDetectionMechanism ( )*

```
Input: Current population P
Select a percentage of solutions (SS) of the current population;
repeat
    Select a solution x_{i,t} from the set of solutions (SS);
    Evaluate x_{i,t} at time t;
    if any value is not the same as those of its previous evaluation then
        Change reaction: Immune Response (Algorithm 5);
    end if
until At least one change is detected
```

5.1.2 Change reaction

When an environment change is detected, an immune response is used to respond, so that, a population reinitialization takes place, through either the hypermutation of previous solutions or the clonation of some individuals based on clonal selection principle. The first one allows to maintain the diversity in the population and the second promotes a fast convergence using history solutions, stored in memory.

The immune response works as follows:

1. Initialize a memory so that it is empty.

2. Reinitialize a percentage (ζ =20) of the population.The population in the new environment has old solutions and reinitialized solutions, i.e., a percentage of the population is replaced by random solutions.

3. Determinate for each individual in the population the dominance relations.

4. Split the individuals of the current population into antigens and antibodies. For this approach, the dominated individuals are considered like antigens and non-dominated individuals are considered like antibodies. In an immune approach, the antibodies are capable of recognizing and eliminating antigens (worst individuals in the population). The percentage of antibodies and antigens depends on the problem conditions, i.e., the number of non-dominated solutions in each iteration.

5. Based on the cloning criteria proposed in [22], the same number of clones must be created for each antibody, and that value is determined such that, 60% of

the population is created as clones. Once memory is full, clones are created following the next rules [22]:

- Zero clones are created if an individual stored in memory is repeated or if it belongs to crowded regions.
- The number of clones is duplicated if the individual belongs to a position whose number of solutions contained in the memory is below the average of all occupied positions.
- The number of clones is half reduced if the number of solutions in memory of an added individual position is above the average of all occupied positions.

Therefore, the number of clones created at each step depends on the number of individuals in the current population and the space in memory.

6. Clone all non-dominated antibodies based on the information from the previous step, after that, copy them into memory.

7. Determine the mutation rates for hypermutation process. The algorithm uses an affinity measure (Euclidean distance) to compare antibodies against antigens. The antigens with high affinity are considered significantly similar to antibodies and they do not need to be modified significantly. For the hypermutation process, a non-uniform hypermutation operator is applied to the antigens as follows: the mutation rate changes linearly over time, for the antigens with highest affinity the mutation rate changes from 0.3 to 0.5, and for the rest of antigens (lowest affinity) from 0.5 to 0.9. For the affinity measure a threshold of 1.0e-4 was used. The parameters for the mutation rates were inherited from the original paper of the clonal criteria adopted in this work [22].

8. The process is repeated from step three until all the antigens were selected.

9. When the immune response ends, the best antibodies and their corresponding clones are stored in the memory.

In Immune GDE3 a memory is used as an elitism mechanism in order to maintain the best individuals found along the optimization process. The individuals stored in memory are non-dominated individuals not only with respect to each other but also with respect to all of the previous individuals which attempted to enter into the memory. To store the individuals in memory with a uniform distribution, the adaptive grid proposed by Knowles and Corne is used [55]. The number of non-dominated individuals stored is fixed. Therefore, the size of memory is limited, and the memory will get full at some moment. When this occurs, a criterion to allow the storage of other non-dominated solutions is required. A region density approach is used to deal with this problem, i.e., the individuals that belong to less dense regions have preference to be stored. For the implementation of the adaptive grid, the memory should be divided according to a number of subdivisions specified as a parameter (25 grid subdivision as it was proposed in [22]).

The mutation rates, the percentage of clones created, the threshold for the affinity measure and the memory size are also parameters of the proposed approach, and their values were adopted according to the experimental design proposed in [22]. An important advantage of the immune response used by Immune GDE3 is that the number of evaluations does not increase significantly because in step 2 only 20 % of the population is reinitialized. Therefore, the algorithm only computes the function evaluations of the reinitialized members.

Algorithm 5 and Fig. 5.1 show the general behavior of the immune response implemented in Immune GDE3.

5.1.3 MOP optimization

In multi-objective optimization, the design of algorithms capable of solving problems with different characteristics, for example, solving problems with any number of objectives and any number of constraints, is an important task. The third evolution version of the Generalized Differential Evolution, GDE3, is a multi-objective optimization algorithm designed for multiple objectives and constraints without introducing any extra control parameters to the original DE. On the other hand, different studies have shown that GDE3 improved the diversity and convergence of the solutions over other traditional methods like NSGA-II [57],[56]. Therefore, for

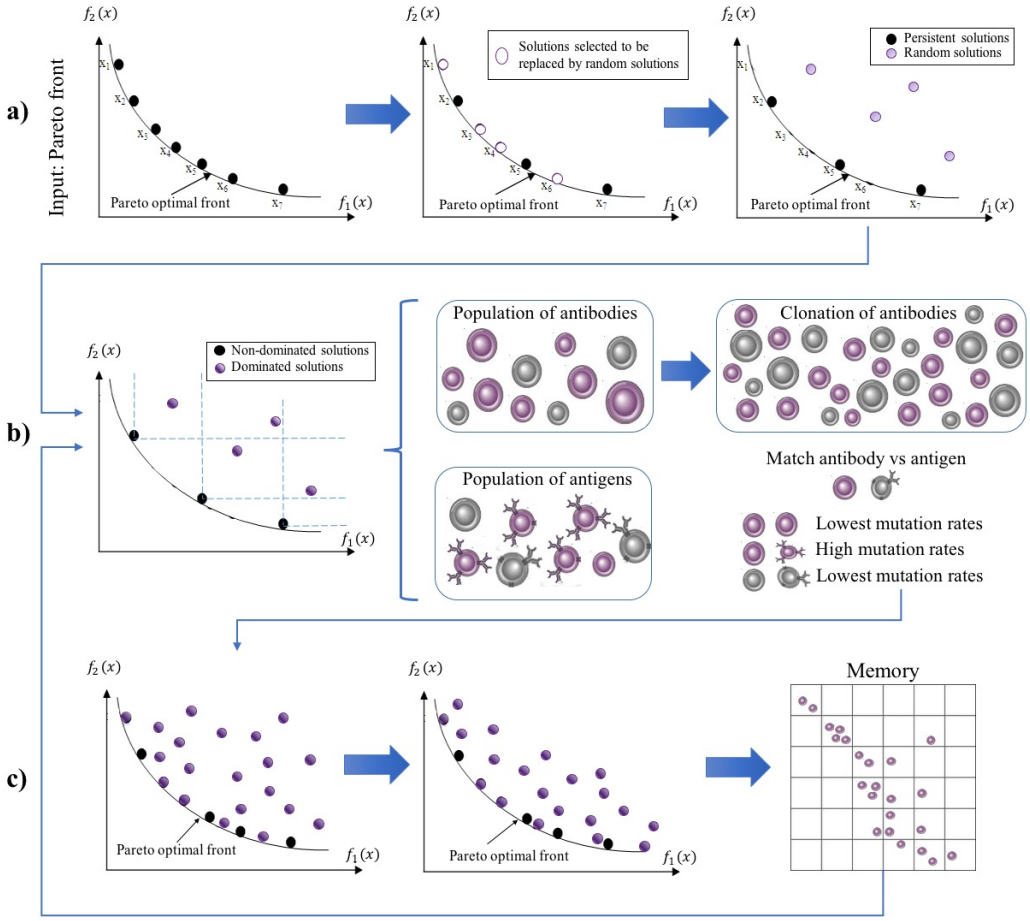

Figure 5.1: Immune Response of Immune GDE3 algorithm.

Algorithm 5 *ChangeReactionMechanism ( )*

```
Input: Current population
Initialize a memory so that it is empty;
Replace a percentage of the population by random solutions;
repeat
    Compute the dominance relations;
    Divide the population in antibodies and antigens;
    Clone all the antibodies;
    Select an antigen A from Population of antigens (PA);
    Take (randomly) R antibodies from Population of antibodies (PS);
    for each antibody r in R do
        Compare the antibody r against the selected antigen A;
        Compute its match score (affinity measure: Euclidean distance);
        Highest affinity antigens experiment the lowest mutation rates;
        The lowest affinity antibodies have high mutation rates;
    end for
until All the antigens were selected
return P
```

the optimization process, the GDE3 algorithm was selected as a core framework, in order to design a DMOEA suitable to solve different kind of dynamic multi-objective search spaces.

It is necessary to emphasize that after the immune response is carried out, the population size may grow. Then, some individuals need to be eliminated to maintain the original population size. To do that, GDE3 uses the crowding distance method to truncate the population.

Fig. 5.2 shows the flowchart of the Immune GDE3 algorithm. As it can be seen in it, the immune response is activated when a change is detected in the environment. After that, the GDE3 algorithm continues its operation by applying the non-dominated sorting to the new population. The process stops when a maximum number of generations is reached.

5.2 Experimental Design

In order to asses the performance of Immune GDE3, three experiments were designed. Each of them is described below:

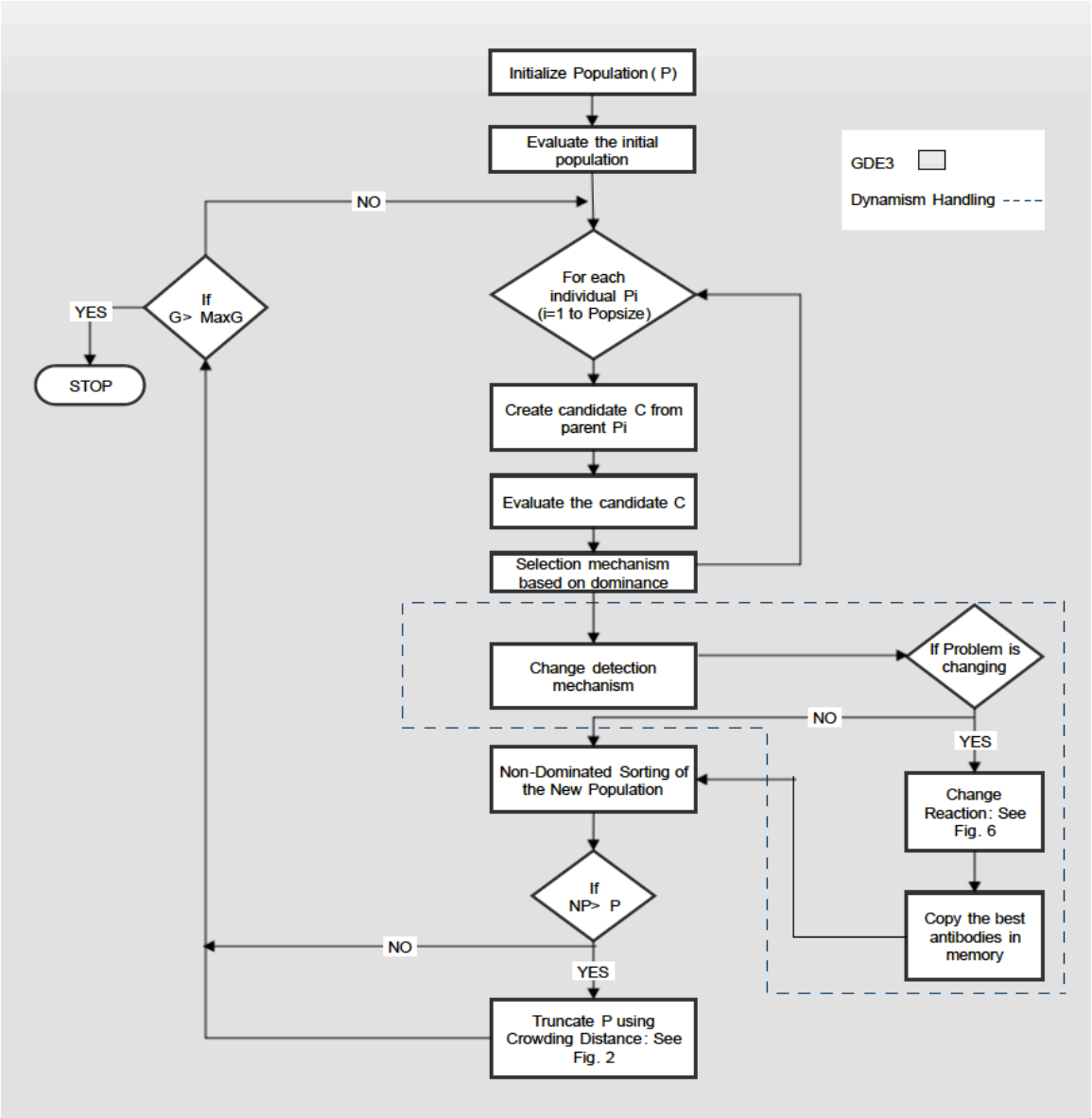

Figure 5.2: Flowchart of Immune-GDE3.

5.2.1 Experiment 1: Immune GDE3 performance analysis

The aim of the first experiment is to evaluate the performance of Immune GDE3 against other well-known dynamic MOEAs. Each one of them has different mechanisms to deal with dynamism and to solve the multi-objective optimization problem. A brief description of each one of them is presented below:

- **DNSGA II-A and DNSGA II-B** [27]: These two variations of NSGA-II, track the new Pareto optimal frontiers in dynamic environments. When there is a change in the problem, on the one hand, the first version (DNSGA II-A) uses the addition of random solutions. A percentage of the new population is replaced with randomly created solutions. This approach helps to introduce diversity when there is a change in the problem and it performs better in problems undergoing a large change in the objectives and constraints. In the second version (DNSGA II-B), instead of introducing random solutions, a percentage of the population is replaced with mutated solutions of existing solutions chosen randomly. In this case, the introduced solutions are related to the existing population. For this reason, this method works better in problems with less severe changes in the problem.

- **DPSO-4** [65]: This algorithm is a dynamic version of the original PSO. In this approach, when a change occurs, the reinitialization of particles is considered. Therefore, the current position of each particle after the change will be taken as the best position of the particle, in PSO known as *pbest*, and the new leader, i.e. the best position of the whole swarm (*gbest*) should be identified. After that, the particles are reevaluated and the resulting non-dominated solutions are stored in an archive, which could be updated using a hyper-plane distribution. One important characteristic of this method is the reinitialization of the velocity of each particle to zero when a change is detected in order to properly follow the POF.

- **MOEA/D-BR (MOEA/D + PS-based nearest distance + Basic Reinitialization Strategy)** [8]: It is a variant of the MOEA/D algorithm (Multi-objective Evolutionary Algorithm based on Decomposition). In this approach,

controlled Extrapolation, POF based nearest distance and a re-initialization scheme proposed in [106] were incorporated to the MOEA/D to form a complete algorithm which can work with DMOPs. This method is mainly used to map a current population member to its past states.

- **SGEA (Steady-State and Generational Evolutionary Algorithm)** [52]: It is a recently proposed algorithm to solve DMOPs, which detects environmental changes and responds in a steady-state manner. Once a change is detected, SGEA reuses a number of well-distributed solutions from past generations and relocates them close to new positions of the Pareto front based on the information collected from previous environments. Therefore, the new environment population consists of 50% old solutions and 50% reinitialized solutions. Old solutions are selected by a method called farthest first selection method. This method selects 50% old solutions that maximize diversity in the objective space. On the other hand, reinitialized solutions are produced by predicting the new location of the changed POS. To make a correct or at least reasonable prediction, SGEA computes two main things: a moving direction and a movement step-size.

As it can be seen, each selected algorithm uses different strategies of dealing with changes in the environment. In addition, these algorithms have different mechanisms to work with multi-objective problems. On the one hand, the algorithms that use the non-dominated sorting criteria (DNSGA-II versions), then the Swarm intelligence algorithm (DPSO-4), the most traditional algorithm based on decomposition of multi-objective problems (MOEA/D-BR) and finally, the steady-state and generational evolutionary algorithm (SGEA).

The parameter settings for the proposed approach and the parameters of the algorithms selected for comparison are inherited from referenced papers. Table 5.1 shows the parameter values used for the experiments carried out.

The performance of each algorithm is measured using different performance metrics adopted from multi-objective optimization. Those metrics will be explained later.

The obtained results and the discussion remarks of this experiment are presented in Section 5.3.1.

Table 5.1: Parameter values for the tested algorithms.

All algorithms	Problem parameters	τ_t=5, n_t=5
	Change detection	10% of the population randomly selected
	Number of runs	30
	Stopping criteria	Maximum generations = 300
	Population size	200 (bi-objective problems) 300 (tri-objective problem)
DNSGA-II versions	Crossover ratio	0.8
	Mutation ratio	0.6
	Percentage of replaced or hyper muted solutions	ζ =20%
DPSO-4	Particles	20
	Memory size	100
	Velocity vector	W=0.4, C1=1.49, C2=1.49
MOEA/D-BR	Neighborhood size	20
SGEA	Crossover ratio	1.0
	Mutation ratio	$1/n$ (number of decision variables)
	Archive size	100
Immune GDE3	F	0.5
	CR	0.8
	Mutation rates	0.9-0.3
	Memory size	100

5.2.2 Experiment 2: Change frequency and severity analysis

Due to the importance of the change frequency and severity analysis, the second experiment was designed to analyze the effects of different change severities and also change frequencies in the performance of Immune GDE3 and the most competitive algorithm of the previous experiment. The tests were conducted at different combinations of change severities and frequencies, i.e., (n_t, τ_t)= (5,5), (5,10), (10,10), and (10,5). The different combinations of n_t and τ_t parameters were selected according to the experimental designed proposed in [51].

5.2.3 Experiment 3: The role of immune response

The third experiment was divided in two parts. In the first part of this experiment, a comparison of Immune GDE3 against other dynamic GDE3 versions was carried out. In preliminary tests, Immune GDE3 showed promising results [66]. However, with those previously reported results, it is hard to find out why Immune GDE3 performs better than the other compared algorithms. The main goal of this experiment is to

analyze the role of the immune response in the GDE3 algorithm to solve DMOPs. Therefore, this experiment was designed to understand if the competitive performance of Immune GDE3 is due to either the immune response or other components of the algorithm. The other three GDE3 versions considered in this experiment are described below:

- GDE3-A and GDE3-B: These two GDE3 variants use DNSGA-II mechanism to track environmental changes. When a change is detected, in the first version (GDE3-A), a percentage of the new population is replaced with random immigrants. In the second version (GDE3-B), a percentage of the population is replaced with randomly mutated solutions. Similar to Immune GDE3, the optimization algorithm in GDE3-A and GDE3-B is the GDE3 algorithm. In addition, these versions use the same criteria to detect the environment changes. The most significant difference is the change reaction mechanism involved in each one of them.

- GDE3-BR: This version uses the prediction strategy and the re-initialization scheme incorporated to MOEA/D-BR. Once a change is detected, the new locations of a number of Pareto solutions in the decision space are predicted. After that, the new individuals in the population for the changed problem are sampled around those predicted points.The change detection also consists in the re-evaluation of solutions.

As can be seen, the new proposed versions for this experiment use GDE3 as a search algorithm. However, the immune response was removed from Immune GDE3 and the change reaction mechanism was replaced by other mechanisms found in the specialized literature.

On the other hand, the second part of this experiment was designed to evaluate the effect of the immune response in the less competitive algorithm of Experiment one. The change detection and change reaction mechanisms were removed from that algorithm, and they were replaced by the immune response of the Immune GDE3 algorithm. Therefore, the aim of this part of the experiment consists in analyzing the improvement capability of the immune response when added to another multi-objective optimization evolutionary algorithm.

The parameter configuration of the compared algorithms in this experiment are taken from the first experiment (n_t=5, τ_t=5). The percentage of replaced and hyper mutated solutions is the same of DNSGA-II versions. For GDE3, parameters values are F=0.5 and CR=0.8. These parameters were determined empirically. Therefore, they were selected because they obtained the better results for most of the test problems.

5.2.4 Test instances

Benchmark problems play an important role in assessing the performance of an algorithm. Furthermore, benchmark problems contribute to analyzing and identifying the strengths and weaknesses of a DMOEA, guiding the algorithm design. However, one of the main issues in the field of dynamic multi-objective optimization is that there are no standard test suite of benchmark problems to evaluate the performance of DMOAs. In addition, problems with different characteristics like constrained problems, many-objective problems and high dimensionality problems are still required [47],[48].

On the three experiments carried out, two different set of functions were selected from the specialized literature. On the one hand, a benchmark for DMOPs recently proposed by Biswas et al. was solved. This set of test functions introduces dynamicity into the POS and POF through an angular shift or slope change of the POF, shape variation of a Polynomial or Trigonometric POS and curvature change of a spherical POF. This benchmark has the following characteristics [8]:

- It contains 9 multi-objective dynamic functions (UDF1 to UDF9).
- All functions have two objectives, except UDF7 which has three objectives.
- The POF Geometry is continuous, discrete and time-varying curvature and/or location.
- All are unconstrained dynamic functions.
- All objective functions are to be minimized.
- The functions have different dimensions of the search space.

Table 5.2: Summary of main features for benchmark set used for experiments

Test Problem	**Problem Type**	**Number of Objectives**	n	**POF Nature**	**POS Nature**
FDA1	Type I	2	11	Continuos	Trigonometric
FDA2	Type III	2	13	Continuos	Trigonometric
FDA3	Type II	2	10	Continuos	Trigonometric
UDF1	Type II	2	10	Continuos	Trigonometric
UDF2	Type II	2	10	Continuos	Polynomial
UDF3	Type III	2	10	Discrete	Trigonometric
UDF4	Type II	2	10	Continuos	Trigonometric
UDF5	Type II	2	10	Continuos	Polynomial
UDF6	Type III	2	10	Discrete	Trigonometric
UDF7	Type III	3	10	Continuos	Trigonometric
UDF8	Type II	2	10	Continuos	Trigonometric
UDF9	Type II	2	10	Continuos	Polynomial

The other test functions selected were the FDA functions given by Farina *et. al.* [33]. These functions were created by adapting the static problems from the ZDT [112] and DTLZ [26] test suites. The POF and/or the POS change with time, while the number of decision variables, the number of objectives and the boundaries of the search space keep fixed throughout the run. Another feature of the FDA test suite is the linear correlation among the decision variables. The dynamism in these functions is introduced only with two types of changes: shift and shape (curvature) change of the POF, and shift of the POS throughout with change in radius of a spherical POF. Farina's test suite has been extensively used by DMOPs researchers to evaluate the performance of their algorithms.

The main features of the the benchmark problems used in our experimental design are summarized in Table 5.2 and the details can be found in [8] and [33].

5.2.5 Performance metrics

The obtained results of the three experiments were evaluated using different performance metrics adopted for DMOPs and one binary metric was also adapted to evaluate the performance in dynamic environments.

For comparison purposes, 30 independent runs were carried out by each algorithm. The obtained results in the runs were used to compute the values of four performance metrics to assess the performance of evolutionary multi-objective optimization algorithms. The metrics have been adapted to work with DMOPs. These

measures are Inverted Generational Distance (IGD), Hypervolume (HV), Spacing (S) and Two-Set Coverage (C-metric). S and C-metric are used to analyze distribution and coverage, respectively. IGD and HV measure proximity to the Pareto Optimal front. A brief description of each metric for evaluating the performance of DMOEAs is presented below.

- **Inverted Generational Distance (IGD) [77]:** It is one of the most commonly adopted performance metric for DMOPs. Some authors suggest that it measures both diversity and convergence of found solutions by an algorithm [99], [103], [51].

  For DMOPs, this metric works in a similar way to the static version (see section 3.5). However, to evaluate the performance of a DMOP, an average of the IGD values in each time step over a run is calculated, which is computed as in Equation 5.1 [107]:

$$\overline{IGD} = \frac{1}{T_s} \sum_{t=1}^{T_s} IGD(t) \tag{5.1}$$

  where IGD(t) refers to the IGD at time instance t, which is calculated before the next change.

- **Hypervolume (HV) [93]**: For static multi-objective optimization problems, when HV value is larger, the performance of an algorithm is better. However, for DMOPs, since Pareto optimal front (POF) is time-varying, it becomes meaningless to compare different $HV(t)$ values through generations. The way to overcome this problem is to compute the ratio of $HV(t)$, i.e., $HVR(t)$, of POF* approximation and POF. The HVR is computed as indicated in Equation 5.2 [61]:

$$HVR(t) = \frac{HV(POF^*)}{HV(POF)} \tag{5.2}$$

  In this work, the reference point z_{ref} is the worst value in each objective dimension of all solutions at iteration t. The maximum value of HVR(t) must be 1 when POF* is equal to the POF. HVR(t) also requires that the POF

is known and it is dependent on the sampling distribution of solutions in the POF. To measure how well an algorithm performs over a run according to HV, the average of HVR(t) at each time step needs to be computed (See Equation 5.3).

$$\overline{HV} = \frac{1}{T_s}\sum_{t=1}^{T_s} HVR(t) \tag{5.3}$$

- **Spacing (S)** [51]: To analyze the distribution of solutions in a POF* obtained by a DMOEA, the spacing metric was adopted. In the same way of the previous metrics, for DMOPs the average of the spacing metric over all time steps in a run is computed as in Equation 5.4.

$$\overline{S} = \frac{1}{T_s}\sum_{t=1}^{T_s} S(t) \tag{5.4}$$

  where S(t) refers to the S at time instance t, which is calculated before the next change.

- **Two-set Coverage (C-Metric)** [111]: For DMO, this binary performance metric estimates the coverage proportion, in terms of percentage of dominated solutions, between two DMOEAs. In DMO, two comparisons are applied at each time step in a run before the changes took place. The average of obtained results at each time step is computed as in Equation 5.5. Where C(t) is the value of C-metric at time instance t.

$$\overline{C} = \frac{1}{T_s}\sum_{t=1}^{T_s} C(t) \tag{5.5}$$

5.3 Results and Discussion

The twelve aforementioned test problems (summarized in Table 5.2) were tackled in the three proposed experimental frameworks. The statistical results of the four performance metrics were computed over 30 independent runs.

For IGD, HV and Spacing, statistical validation was made with 95%-confidence Kruskal-Wallis (KW) test and the Bergmann-Hommels post-hoc test, as suggested in [28]. Two-set coverage metric compares among all assessed algorithmic techniques. Therefore, a pairwise comparison was carried out considering Immune GDE3 as reference. Results were validated by a 95% confidence rank-sum Wilcoxon test. This section presents the obtained results and the discussion remarks of each experiment.

5.3.1 Results of Experiment 1: Immune GDE3 performance analysis

Experiment 1 compares Immune GDE3 against state-of-the-art DMOEAs. The change severity and change frequency were $n_t = 5$ and $\tau_t = 5$, respectively. These values are the most used in the specialized literature of DMOPs. Lower values of these two parameters imply more difficulty to the problems in their dynamic behavior [8].

For the obtained results of this experiment, the algorithm that outperformed the competitors had the best rank with the number one, and the one that outperformed the least had the worst rank. Using this ranking approach the final rank of each algorithm and the statistical results under each performance metric (according to the mean of the metric result) is indicated in Table 5.3. For UDF7 test problem, DPSO results are not reported, because DPSO algorithm does not work with problems with three or more objectives.

Proximity metrics discussion

Regarding IGD metric, a lowest value is preferred, Table 5.3 presents the $\overline{IGD}$ values and ranking of the algorithms for all test instances. From the table, it was observed that the algorithm that obtains the lowest values (rank one) in most test problems is Immune GDE3. Therefore, Immune GDE3 outperformed all algorithms in nine of twelve functions. This good performance can be attributed to the fast convergence in Immune GDE3, which helps the algorithm to track the changes in the environment as quickly as possible. As it can be seen at the end of Table 5.3, the results of the Bonferroni-Dunn post-hoc test confirm such finding. On the other

hand, by comparing the algorithm performance in FDA1, FDA3 and UDF6 test problems, where apparently SGEA and MOEA/D-BR were better than Immune GDE3, according to the statistical test, no significant differences are observed, i.e., the algorithms have similar behavior. The statistical results also show that SGEA was better than Immune GDE3 in UDF6. However, for the rest of the metrics such behavior is not observed. If a closer look is taken, for UDF6, it can be observed that the discrete Pareto optimal front nature increase the difficulty of the DMOP, thus affects the performance of all the algorithms. The results obtained by this metric indicate not only the good performance of Immune GDE3 to track the Pareto optimal front in most of the test instances but also the good distribution of its solutions.

Results of Hypervolume metric were similar than IGD results, Table 5.3 shows the good performance of Immune GDE3. For this metric, Immune GDE3 obtained better results than the rest of the algorithms in eleven of twelve test problems. However, when it was compared with MOEA/D-BR, the statistical test showed no significant differences between these two algorithms in nine of twelve test problems (See Table 5.3). Therefore, Immune GDE3 and MOEA/D-BR have similar behavior in most of the test problems. Otherwise, when Immune GDE3 was compared against SGEA, the results of the statistical test showed that SGEA was similar only in three test problems.

Comparing with DPSO algorithm, it was observed that, for IGD Immune GDE3 was better in nine of eleven test problems. Regarding HV, Immune GDE3 obtained better results in eight of eleven test problems. Both metrics measure proximity to the POF, even though results showed different performance depending on the tested problem. For IGD no significant differences were found in FDA3 and UDF5, while HV results showed no significant differences in FDA2, UDF4 and UDF9.

From Table 5.3, it can be clearly observed that the Immune GDE3 algorithm is better when it is compared with DNSGA-II-A and DNSGA-II-B. The results of the statistical test at the end of the Table confirm such finding.

Table 5.3: $\overline{IGD}$, $\overline{HV}$, and $\overline{S}$ mean, standard deviation values and performance rankings of the first experiment. Resume of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test . "+" means that Immune GDE3 outperformed the algorithm in the corresponding row. "−" means that the algorithm in the corresponding row outperformed Immune GDE3. No significant differences between Immune GDE3 and the version in the corresponding row are indicated with "=".

Test Prob.	**IGD**			**HV**			**S**		
	Algorithm	Mean St. Dev.		Algorithm	Mean St. Dev.		Algorithm	Mean St. Dev.	
FDA1	SGEA(1)	0.04826 ±(6.0373E-3)	=	Immune-GDE3 (1)	0.94760 ±(1.2433E-4)		SGEA(1)	0.12182 ±(1.7812E-3)	=
	Immune-GDE3 (2)	0.05506 ±(1.0464E-3)		MOEA/D-BR (2)	0.91595 ±(1.8250E-2)	+	MOEA/D-BR (2)	0.13454 ±(9.9702E-3)	=
	MOEA/D-BR (3)	0.08601 ±(6.1657E-3)	+	SGEA(3)	0.91350 ±(1.5141E-2)	+	Immune-GDE3 (3)	0.13755 ±(1.2220E-2)	
	DNSGA-II-A (4)	0.12084 ±(9.1590E-3)	+	DNSGA-II-B (4)	0.84152 ±(2.0022E-2)	+	DNSGA-II-A (4)	0.18458 ±(2.8821E-2)	+
	DNSGA-II-B (5)	0.12039 ±(8.1615E-3)	+	DNSGA-II-A (5)	0.82670 ±(1.3890E-2)	+	DPSO-4 (5)	0.18886 ±(3.4469E-2)	+
	DPSO-4 (6)	0.13484 ±(6.7136E-3)	+	DPSO-4 (6)	0.79470 ±(1.5505E-2)	+	DNSGA-II-B (6)	0.19944 ±(3.6756E-2)	+
FDA2	Immune-GDE3 (1)	0.01533 ±(2.4733E-3)		Immune-GDE3 (1)	0.94249 ±(1.2731E-2)		Immune-GDE3 (1)	0.04399 ±(6.9097E-3)	
	MOEA/D-BR (2)	0.01557 ±(1.4357E-3)	=	MOEA/D-BR (2)	0.93456 ±(2.0434E-2)	=	DPSO-4(2)	0.06064 ±(1.7331E-2)	=
	SGEA(3)	0.01998 ±(3.5352E-3)	+	DPSO-4 (3)	0.93071 ±(1.8495E-2)	=	SGEA(3)	0.07104 ±(2.7298E-3)	+
	DNSGA-II-A (4)	0.02724 ±(3.1655E-3)	+	DNSGA-II-A (4)	0.89729 ±(1.8394E-2)	+	MOEA/D-BR (4)	0.08532 ±(9.7344E-3)	+
	DNSGA-II-B (5)	0.02604 ±(3.9815E-3)	+	DNSGA-II-B (5)	0.88967 ±(1.9928E-2)	+	DNSGA-II-B (5)	0.09028 ±(3.7050E-2)	+
	DPSO-4 (6)	0.02843 ±(4.6869E-3)	+	SGEA(6)	0.87145 ±(1.2110E-2)	+	DNSGA-II-A (6)	0.09174 ±(2.9774E-2)	+
FDA3	SGEA(1)	0.12583 ±(1.4993E-2)	=	SGEA(1)	0.92566 ±(1.2194E-2)	=	SGEA(1)	0.12652 ±(1.3411E-2)	=
	Immune-GDE3 (2)	0.13446 ±(8.7314E-3)		Immune-GDE3 (2)	0.90627 ±(1.3549E-2)		Immune-GDE3 (2)	0.14233 ±(5.4645E-3)	
	DPSO-4 (3)	0.16005 ±(4.1623E-3)	=	MOEA/D-BR (3)	0.88773 ±(1.3071E-2)	+	MOEA/D-BR (3)	0.20567 ±(1.5007E-2)	+
	MOEA/D-BR (4)	0.16332 ±(1.6654E-2)	+	DPSO-4 (4)	0.85143 ±(1.9964E-2)	+	DPSO-4 (4)	0.21385 ±(1.3056E-2)	+
	DNSGA-II-B (5)	0.18242 ±(2.0071E-2)	+	DNSGA-II-B (5)	0.85073 ±(2.0319E-2)	+	DNSGA-II-A (5)	0.25911 ±(1.6295E-2)	+
	DNSGA-II-A (6)	0.19680 ±(2.6509E-2)	+	DNSGA-II-A (6)	0.84718 ±(1.4987E-2)	+	DNSGA-II-B (6)	0.25956 ±(3.1350E-2)	+
UDF1	Immune-GDE3 (1)	0.12688 ±(4.6714E-3)		Immune-GDE3 (1)	0.92324 ±(1.3869E-2)		Immune-GDE3 (1)	0.08882 ±(1.8400E-2)	
	MOEA/D-BR (2)	0.14659 ±(8.0849E-3)	+	MOEA/D-BR (2)	0.90257 ±(1.3726E-2)	=	MOEA/D-BR (2)	0.08969 ±(1.3500E-2)	=
	DPSO-4 (3)	0.16077 ±(3.6720E-3)	+	DPSO-4 (3)	0.88330 ±(7.5728E-3)	+	SGEA(3)	0.11742 ±(1.7946E-2)	+
	SGEA(4)	0.17174 ±(4.4882E-3)	+	SGEA(4)	0.87701 ±(1.6893E-2)	+	DNSGA-II-B (4)	0.12263 ±(2.6779E-2)	+
	DNSGA-II-B (5)	0.17658 ±(2.4053E-2)	+	DNSGA-II-A (5)	0.86145 ±(1.2560E-2)	+	DPSO-4 (5)	0.12941 ±(1.7208E-2)	+
	DNSGA-II-A (6)	0.19008 ±(2.5501E-2)	+	DNSGA-II-B (6)	0.84651 ±(2.0782E-2)	+	DNSGA-II-A (6)	0.13308 ±(1.3565E-2)	+
UDF2	Immune-GDE3 (1)	0.03520 ±(1.2918E-3)		Immune-GDE3 (1)	0.97536 ±(1.1036E-2)		Immune-GDE3 (1)	0.02493 ±(6.5317E-3)	
	MOEA/D-BR (2)	0.03884 ±(6.9585E-4)	=	MOEA/D-BR (2)	0.95622 ±(8.2531E-3)	=	MOEA/D-BR (2)	0.04363 ±(1.5216E-2)	=
	DPSO-4 (3)	0.04309 ±(2.5694E-3)	+	DNSGA-II-B (4)	0.90056 ±(5.4139E-3)	+	SGEA(3)	0.06210 ±(2.6613E-2)	+
	DNSGA-II-B (4)	0.05547 ±(1.1405E-3)	+	DPSO-4 (3)	0.89903 ±(2.0427E-2)	+	DPSO-4 (4)	0.07989 ±(3.0888E-2)	+
	DNSGA-II-A (5)	0.05769 ±(2.6110E-4)	+	DNSGA-II-A (5)	0.86818 ±(7.5739E-3)	+	DNSGA-II-B (5)	0.13437 ±(1.6419E-2)	+
	SGEA(6)	0.06815 ±(1.8508E-3)	+	SGEA(6)	0.83819 ±(8.6301E-3)	+	DNSGA-II-A (6)	0.13679 ±(1.1456E-2)	+
UDF3	Immune-GDE3 (1)	0.31319 ±(6.9004E-3)		Immune-GDE3 (1)	0.89853 ±(1.7168E-2)		Immune-GDE3 (1)	0.11594 ±(1.0428E-2)	
	MOEA/D-BR (2)	0.42828 ±(1.8934E-2)	+	MOEA/D-BR (2)	0.89556 ±(1.0683E-2)	=	MOEA/D-BR (2)	0.15769 ±(5.6919E-3)	+
	DPSO-4 (3)	0.52859 ±(2.2757E-2)	+	DNSGA-II-B (3)	0.88021 ±(1.8489E-2)	+	SGEA(3)	0.17986 ±(5.9698E-3)	+
	DNSGA-II-B (4)	0.61963 ±(3.6366E-2)	+	SGEA(4)	0.87594 ±(1.7674E-2)	+	DPSO-4 (4)	0.19967 ±(8.4299E-3)	+
	DNSGA-II-A (5)	0.65057 ±(4.4689E-2)	+	DPSO-4 (5)	0.86224 ±(1.4045E-2)	+	DNSGA-II-B (5)	0.22222 ±(8.4713E-3)	+

Continued on next page

Table 5.3 *Continued from previous page*

Test Prob.		IGD				HV				S		
	SGEA(6)	0.65874 ±(4.8253E-2)		+	DNSGA-II-A (6)	0.84892 ±(1.1309E-2)		+	DNSGA-II-A (6)	0.23209 ±(1.6955E-2)		+
	Immune-GDE3 (1)	0.20832 ±(1.6900E-2)			Immune-GDE3 (1)	0.93442 ±(1.4374E-2)			Immune-GDE3 (1)	0.09110 ±(1.0267E-2)		
	SGEA(2)	0.27363 ±(2.9540E-2)		+	DPSO-4 (2)	0.92053 ±(2.0309E-2)		=	SGEA(2)	0.15638 ±(1.2938E-2)		+
UDF4	DPSO-4 (3)	0.34181 ±(2.3781E-2)		+	MOEA/D-BR (3)	0.90249 ±(1.5699E-2)		=	DPSO-4 (3)	0.16577 ±(6.9443E-3)		+
	MOEA/D-BR (4)	0.44645 ±(1.2293E-2)		+	SGEA(4)	0.90015 ±(6.5063E-2)		=	MOEA/D-BR (4)	0.17182 ±(1.1848E-2)		+
	DNSGA-II-A (5)	0.58673 ±(1.4736E-2)		+	DNSGA-II-A (5)	0.87169 ±(1.5132E-2)		+	DNSGA-II-A (5)	0.23277 ±(8.9446E-3)		+
	DNSGA-II-B (6)	0.55904 ±(2.0006E-2)		+	DNSGA-II-B (6)	0.86142 ±(9.9539E-3)		+	DNSGA-II-B (6)	0.23659 ±(1.1837E-2)		+
	Immune-GDE3 (1)	0.02243 ±(7.9638E-4)			Immune-GDE3 (1)	0.98301 ±(7.8875E-3)			Immune-GDE3 (1)	0.02498 ±(7.9183E-3)		
	MOEA/D-BR (2)	0.03452 ±(1.5078E-3)		=	MOEA/D-BR (2)	0.97297 ±(1.0906E-2)		=	MOEA/D-BR (2)	0.08302 ±(1.0824E-2)		+
UDF5	DPSO-4 (3)	0.03562 ±(3.9625E-3)		=	SGEA(3)	0.90920 ±(8.8994E-3)		+	SGEA(3)	0.09637 ±(3.9912E-3)		+
	DNSGA-II-A (4)	0.04414 ±(1.8135E-3)		+	DPSO-4 (4)	0.90741 ±(2.3542E-2)		+	DPSO-4 (4)	0.10250 ±(7.0188E-3)		+
	DNSGA-II-B (5)	0.04827 ±(1.4589E-3)		+	DNSGA-II-B (5)	0.86734 ±(9.8722E-3)		+	DNSGA-II-A (5)	0.12082 ±(5.4924E-3)		+
	SGEA(6)	0.08891 ±(7.0345E-3)		+	DNSGA-II-A (6)	0.85313 ±(1.4637E-2)		+	DNSGA-II-B (6)	0.13851 ±(9.0072E-3)		+
	SGEA(1)	1.093726±(2.5638E-2)		+	Immune-GDE3 (1)	0.86883 ±(1.4723E-2)			MOEA/D-BR (1)	0.08957 ±(1.0217E-2)		+
	MOEA/D-BR (2)	1.23559 ±(2.0391E-2)		=	SGEA(2)	0.86600 ±(1.0413E-2)		=	Immune-GDE3 (2)	0.13245 ±(1.5807E-2)		
UDF6	Immune-GDE3 (3)	1.26128 ±(2.9028E-2)			MOEA/D-BR (3)	0.85665 ±(2.5743E-2)		=	SGEA(3)	0.16315 ±(2.8150E-2)		+
	DPSO-4 (4)	1.30090 ±(1.2724E-2)		+	DPSO-4 (4)	0.80307 ±(1.4785E-2)		+	DPSO-4 (4)	0.16963 ±(1.1722E-2)		+
	DNSGA-II-A (5)	1.53947 ±(3.9231E-2)		+	DNSGA-II-A (5)	0.76906 ±(1.3840E-2)		+	DNSGA-II-B (5)	0.19951 ±(5.1329E-3)		+
	DNSGA-II-B (6)	1.61406 ±(5.0917E-2)		+	DNSGA-II-B (6)	0.76213 ±(1.6744E-2)		+	DNSGA-II-A (6)	0.19989 ±(9.9754E-3)		+
	Immune-GDE3(1)	0.21558 ±(9.3813E-3)			Immune-GDE3 (1)	0.90152 ±(1.6464E-2)			Immune-GDE3 (1)	0.11741 ±(7.6001E-3)		
	MOEA/D-BR (2)	0.23547 ±(9.4661E-3)		=	MOEA/D-BR (2)	0.90121 ±(1.7895E-2)		=	MOEA/D-BR (2)	0.17646 ±(1.4566E-2)		+
UDF7	SGEA(3)	0.46301 ±(4.2824E-2)		+	SGEA(3)	0.85080 ±(5.2626E-3)		+	SGEA(3)	0.19502 ±(2.7150E-2)		+
	DNSGA-II-A (4)	0.62510 ±(3.5058E-2)		+	DNSGA-II-B (4)	0.82240 ±(2.2981E-2)		+	DNSGA-II-A (4)	0.22840 ±(1.3060E-2)		+
	DNSGA-II-B (5)	0.68059 ±(3.7617E-2)		+	DNSGA-II-A (5)	0.82178 ±(1.5377E-2)		+	DNSGA-II-B (5)	0.24617 ±(9.2607E-3)		+
	Immune-GDE3 (1)	0.39940 ±(1.1659E-2)			Immune-GDE3 (1)	0.90305 ±(4.4884E-3)			Immune-GDE3 (1)	0.10252 ±(1.0831E-2)		
	DPSO-4 (2)	0.45922 ±(3.7157E-2)		+	MOEA/D-BR (2)	0.87890 ±(6.9378E-3)		+	MOEA/D-BR (2)	0.18150 ±(2.4492E-2)		+
UDF8	SGEA(3)	0.46714 ±(5.2666E-2)		+	SGEA(3)	0.85820 ±(1.0848E-2)		+	SGEA(3)	0.21670 ±(1.0985E-2)		+
	MOEA/D-BR (4)	0.46778 ±(1.9631E-2)		+	DPSO-4 (4)	0.84816 ±(1.0410E-2)		+	DPSO-4 (4)	0.21753 ±(1.2865E-2)		+
	DNSGA-II-A (5)	0.60448 ±(4.5989E-2)		+	DNSGA-II-A (5)	0.82592 ±(1.2795E-2)		+	DNSGA-II-A (5)	0.22365 ±(1.4291E-2)		+
	DNSGA-II-B (6)	0.62362 ±(2.2379E-2)		+	DNSGA-II-B (6)	0.80866 ±(1.6419E-2)		+	DNSGA-II-B (6)	0.25673 ±(1.3008E-2)		+
	Immune-GDE3 (1)	0.11966 ±(5.8976E-3)			Immune-GDE3 (1)	0.92476 ±(1.4725E-2)			Immune-GDE3 (1)	0.06834 ±(1.3208E-2)		
	MOEA/D-BR (2)	0.14116 ±(1.1506E-2)		+	MOEA/D-BR (2)	0.92115 ±(1.8088E-2)		=	MOEA/D-BR (2)	0.11914 ±(1.0972E-2)		+
UDF9	DPSO-4 (3)	0.15766 ±(1.1598E-2)		+	DPSO-4 (3)	0.90375 ±(1.3593E-2)		=	DPSO-4 (3)	0.15025 ±(1.8451E-2)		+
	DNSGA-II-A (4)	0.23963 ±(2.2069E-2)		+	SGEA(4)	0.85266 ±(1.1640E-2)		+	SGEA(4)	0.18593 ±(2.9243E-2)		+
	DNSGA-II-B (5)	0.23760 ±(2.3084E-2)		+	DNSGA-II-B (5)	0.84724 ±(1.2931E-2)		+	DNSGA-II-B (5)	0.21807 ±(1.5518E-2)		+
	SGEA(6)	0.32329 ±(5.2898E-2)		+	DNSGA-II-A (6)	0.84625 ±(1.7802E-2)		+	DNSGA-II-A (6)	0.25415 ±(1.1683E-2)		+
Summary of the statistical test		+	−	=		+	−	=		+	−	=
	DNSGA-II-A	12	0	0		12	0	0		12	0	0
Immune	DNSGA-II-B	12	0	0		12	0	0		12	0	0
GDE3	DPSO-4	9	0	2		8	0	3		10	0	1
	MOEA/D-BR	7	0	5		3	0	9		9	0	3
	SGEA	9	1	2		9	0	3		10	0	2

According to the results previously discussed, MOEA/D-BR and Immune GDE3 are the two most competitive algorithms and present similar performances. However, to analyze another kind of behavior of both algorithms, the tracking IGD plots for some representative test problems are presented. Fig. 5.3 shows the tracking of IGD values obtained by MOEA/D-BR and Immune GDE3 in each time window for change frequency $\tau_t = 5$, i.e., 200 generations/5 for bi-objective problems and 300 generations/5 for tri-objective problem. The figure gives a close inspection of the tracking ability and robustness of the algorithms.

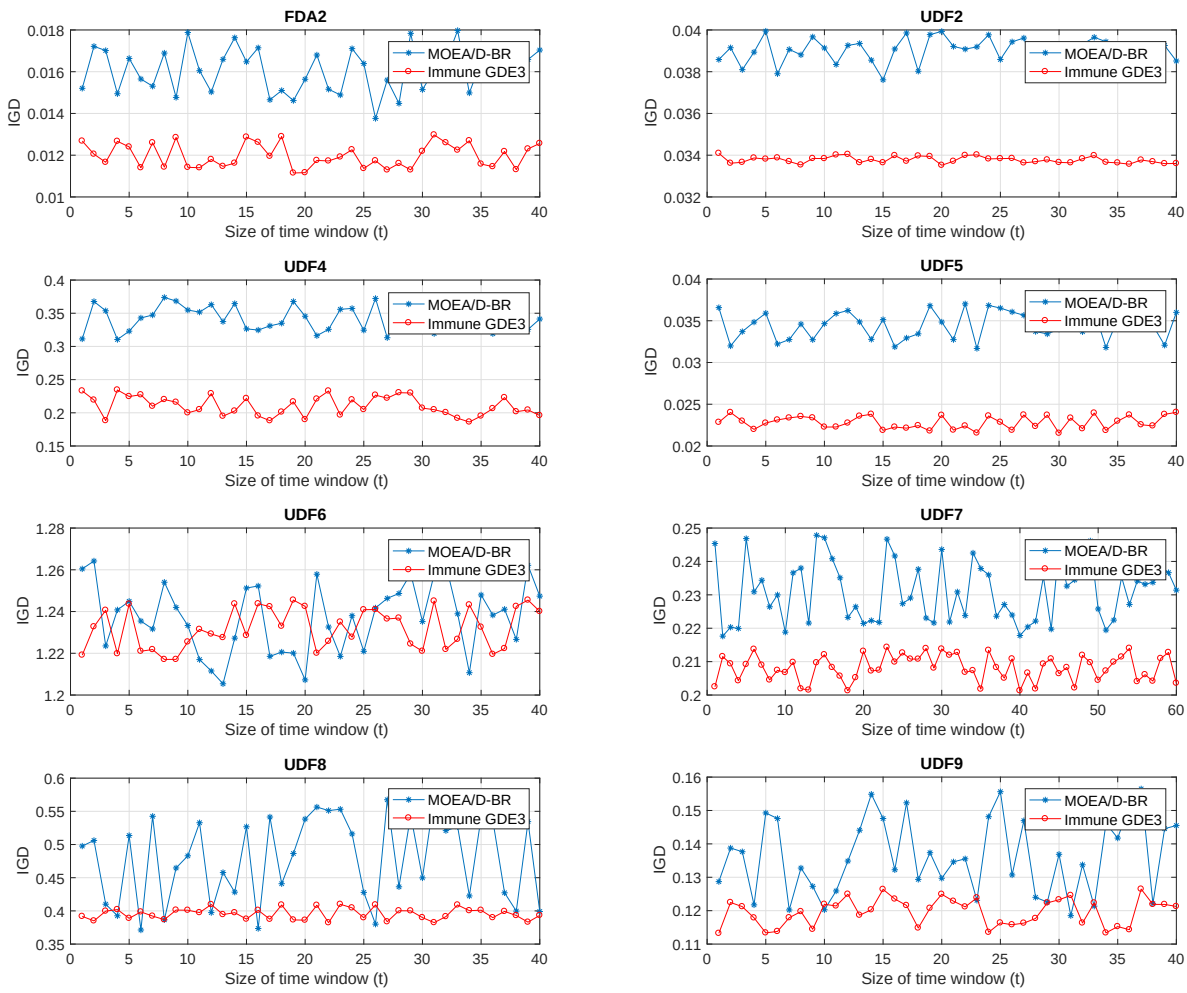

Figure 5.3: Tracking of $\overline{IGD}$ values obtained by the two most competitive algorithms for $n_\tau \tau_t = 5$

From Fig. 5.3, it can be observed that for most test problems, Immune GDE3 obtained better IGD values and the behavior of this metric in each time step is more constant than the values obtained by MOEA/D-BR. For example, for UDF2, UDF5, UDF6 and UFD7 test problems, the statistical test suggest that the performance of both algorithms is similar. However, when the IGD tracking plots are analyzed, it can be observed that the algorithm with the best values of IGD is Immune GDE3. Immune GDE3 also tracks the environmental changes fairly stably in UDF2, UDF4, UDF5, and UDF8. MOEA/D-BR shows robust performance in UDF2, UDF4, and UDF5. This behavior is consistent with those IGD results in Table 5.3.

Distribution metrics discussion

To complement the results obtained by proximity metrics and to analyze how good is the distribution of solutions over the Pareto Front, the statistical results obtained by Spacing and Two-Set Coverage metric (C-metric) are presented.

As regards Spacing metric, it can be seen in Table 5.3, that Immune-GDE3 outperformed the compared algorithms in nine of twelve test problems including the tri-objective test problem (UDF7). In the case of FDA1, FDA3, and UDF6, the obtained results suggested that MOEA/D-BR and SGEA algorithms are better than Immune GDE3. However, Immune GDE3 obtained values closer to those values obtained by MOEA/D-BR and SGEA. Furthermore, the statistical test indicated that there are not significant differences among these three algorithms. At the end of Table 5.3, a resume of the significance test is presented. In such resume table, it can be clearly observed that Immune GDE3 presented good performance in most cases and it is equally competitive than MOEA/D-BR and SGEA, and superior when compared with algorithms based on dominance criteria (DNSGA versions).

For UDF2 and UDF5 test problems, better values on $\overline{S}$ are obtained for all the algorithms. Therefore, in the experiments carried out, for all algorithms, it is easier to maintain the distribution of solutions in DMOPs with continuous POF nature and polynomial POS nature.

In addition, from Table 5.3 it was observed that DPSO in most cases is in the fourth rank. Therefore, the obtained results are below than Immune GDE3. One possible explanation to this is that when a change occurs, DPSO updates the memory

of particles, but if there are a few solutions in the memory, this diversity maintenance technique may not be effective to keep a set uniformly distributed solutions. Furthermore, in the experiments carried out, the change severity is too high, which does not give DPSO time to track the Pareto optimal front. Due to the change severity, a similar reason can also be used to explain the poor performance of DNSGA versions.

Finally, Table 5.4 presents the summary of the Two-Set Coverage metric statistical results. Such values were based on all the pairwise combinations of the 30 independent runs executed by each algorithm (each one of the 30 Pareto fronts of one algorithm was compared with each one the 30 fronts obtained by the other algorithm). As mentioned before, for this metric the optimal value is 1. So that, if all solutions of an algorithm dominate or are equal to all the solutions of the other algorithm, the value for C-metric will be equal to 1, or 0 otherwise. To say that an algorithm is better than another, it is preferable to have values close to 1. Table 5.5, presents the obtained results by the significance test.

Table 5.4 shows the achieved results between each comparison in both directions in a specific test problem. The comparison of algorithms that obtains a higher value of non-dominated solutions is marked in boldface and means that the first algorithm in the comparison is considerably better than the second one. In table 5.4, it can also be observed that, in some cases, the comparisons showed values equal to zero or one, when this occurs means that in all the executions, the first algorithm in the comparison dominates all the solutions of the second algorithm. Therefore, the algorithm that obtains a value equal to one has better performance than the other. The results suggested that, when Immune GDE3 is compared with DNSGA-II-A and DNSGA-II-B, Immune GDE3 always obtains values equal to one in this metric, therefore, as the same way that the results of the metrics presented before, Immune GDE3 outperformed the DNSGA versions.

Regarding the second most competitive algorithm (MOEA/D-BR), the results of this binary metric showed that, Immune GDE3 obtains better results in eight of twelve problems, similar to Spacing where the statistical test showed that Immune-GDE3 is better than MOEA/D-BR in nine of the twelve problems and had similar performance in the remaining test problems including the problems where apparently

MOEA/D-BR is better (Table 5.5). From Table 5.5, it can also be observed that, in contrast with the metrics presented before, Immune GDE3 outperformed DPSO and SGEA in more test problems (ten and eleven, respectively). Taking into account that DPSO does not work with UDF7, only in one test instance (UDF4), both algorithms presented a similar behavior. Regarding SGEA algorithm, the results in Tables 5.4 and 5.5, showed that Immune GDE3 was equal to SGEA only on FDA1 test problem and outperformed SGEA in the remaining test problems.

Table 5.4: Statistical results of $\overline{C}$ mean and standard deviation values for all test problems. The best results are marked in **boldface**.

	Algorithm comparison	Mean	St. dev.		Algorithm comparison	Mean	St. dev.
FDA1	DNSGA-II-A vs DNSGA-II-B	**0.99636**	±(0.0063279)	UDF4	DNSGA-II-A vs DNSGA-II-B	**0.64866**	±(0.0222756)
	DNSGA-II-B vs DNSGA-II-A	0.98354	±(0.0036559)		DNSGA-II-B vs DNSGA-II-A	0.32423	±(0.0178040)
	DNSGA-II-A vs Immune-GDE3	0.01312	±(0.0054813)		DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-A	**0.99834**	±(0.0023691)		Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)
	DNSGA-II-A vs DPSO-4	**1**	±(0.0000000)		DNSGA-II-A vs DPSO-4	0.10374	±(0.0064699)
	DPSO-4 vs DNSGA-II-A	0	±(0.0000000)		DPSO-4 vs DNSGA-II-A	**0.92436**	±(0.0035292)
	DNSGA-II-A vs MOEA/D-BR	0.03410	±(0.0006372)		DNSGA-II-A vs MOEA/D-BR	0.34747	±(0.0063246)
	MOEA/D-BR vs DNSGA-II-A	**0.99233**	±(0.0053674)		MOEA/D-BR vs DNSGA-II-A	**0.68466**	±(0.0089737)
	DNSGA-II-A vs SGEA	0	±(0.0000000)		DNSGA-II-A vs SGEA	0.08384	±(0.0072644)
	SGEA vs DNSGA-II-A	**1**	±(0.0000000)		SGEA vs DNSGA-II-A	**0.95572**	±(0.0026849)
	DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs DPSO-4	**0.97031**	±(0.0124274)		DNSGA-II-B vs DPSO-4	0	±(0.0000000)
	DPSO-4 vs DNSGA-II-B	0.75072	±(0.0231390)		DPSO-4 vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-B vs MOEA/D-BR	0.61354	±(0.0003654)
	MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-B	**0.99437**	±(0.0073637)
	DNSGA-II-B vs SGEA	0	±(0.0000000)		DNSGA-II-B vs SGEA	0.01936	±(0.0004017)
	SGEA vs DNSGA-II-B	**1**	±(0.0000000)		SGEA vs DNSGA-II-B	**0.99720**	±(0.0002847)
	Immune-GDE3 vs DPSO-4	**1**	±(0.0000000)		Immune-GDE3 vs DPSO-4	**0.92535**	±(0.0048849)
	DPSO-4 vs Immune-GDE3	0	±(0.0000000)		DPSO-4 vs Immune-GDE3	0.92435	±(0.0089828)
	Immune-GDE3 vs MOEA/D-BR	**0.99783**	±(0.0005893)		Immune-GDE3 vs MOEA/D-BR	**0.98646**	±(0.0006465)
	MOEA/D-BR vs Immune-GDE3	0.96471	±(0.0004628)		MOEA/D-BR vs Immune-GDE3	0.79365	±(0.0063627)
	Immune-GDE3 vs SGEA	0.97153	±(0.0169668)		Immune-GDE3 vs SGEA	**0.97027**	±(0.0003017)
	SGEA vs Immune-GDE3	**0.99163**	±(0.0118076)		SGEA vs Immune-GDE3	0.92074	±(0.0038792)
	DPSO-4 vs MOEA/D-BR	0	±(0.0000000)		DPSO-4 vs MOEA/D-BR	**0.90373**	±(0.0007465)
	MOEA/D-BR vs DPSO-4	**1**	±(0.0000000)		MOEA/D-BR vs DPSO-4	0.79386	±(0.0064643)
	DPSO-4 vs SGEA	0	±(0.0000000)		DPSO-4 vs SGEA	**0.92047**	±(0.0084501)
	SGEA vs DPSO-4	**1**	±(0.0000000)		SGEA vs DPSO-4	0.91073	±(0.0010746)
	MOEA/D-BR vs SGEA	0.88577	±(0.0211440)		MOEA/D-BR vs SGEA	0.85304	±(0.0070164)
	SGEA vs MOEA/D-BR	**0.96453**	±(0.0141464)		SGEA vs MOEA/D-BR	**0.94017**	±(0.0020744)
FDA2	DNSGA-II-A vs DNSGA-II-B	**0.94833**	±(0.0073894)	UDF5	DNSGA-II-A vs DNSGA-II-B	**0.64407**	±(0.0309839)
	DNSGA-II-B vs DNSGA-II-A	0.89262	±(0.0036559)		DNSGA-II-B vs DNSGA-II-A	0.38646	±(0.0289828)
	DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-A vs Immune-GDE3	0.24564	±(0.0783647)
	Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-A	**0.99865**	±(0.0064742)
	DNSGA-II-A vs DPSO-4	0.34636	±(0.0063246)		DNSGA-II-A vs DPSO-4	0.38747	±(0.0003564)
	DPSO-4 vs DNSGA-II-A	**0.68734**	±(0.0189737)		DPSO-4 vs DNSGA-II-A	**0.64775**	±(0.0087475)
	DNSGA-II-A vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-A vs MOEA/D-BR	0.55074	±(0.0274775)
	MOEA/D-BR vs DNSGA-II-A	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-A	**0.96355**	±(0.0007748)
	DNSGA-II-A vs SGEA	0.31435	±(0.0188654)		DNSGA-II-A vs SGEA	**0.83271**	±(0.0301784)
	SGEA vs DNSGA-II-A	**0.91636**	±(0.0742175)		SGEA vs DNSGA-II-A	0.71021	±(0.0061038)
	DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs DPSO-4	0	±(0.0000000)		DNSGA-II-B vs DPSO-4	0.05807	±(0.0748049)
	DPSO-4 vs DNSGA-II-B	**1**	±(0.0000000)		DPSO-4 vs DNSGA-II-B	**0.85365**	±(0.0074085)
	DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)
	MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs SGEA	0.23355	±(0.0110186)		DNSGA-II-B vs SGEA	**0.84074**	±(0.0064824)
	SGEA vs DNSGA-II-B	**0.93435**	±(0.0048168)		SGEA vs DNSGA-II-B	0.77028	±(0.0020469)
	Immune-GDE3 vs DPSO-4	**0.78634**	±(0.0189737)		Immune-GDE3 vs DPSO-4	**0.96038**	±(0.0032379)

Continued on next page

Table 5.4 *Continued from previous page*

	Algorithm comparison	Mean	St. dev.		Algorithm comparison	Mean	St. dev.
	DPSO-4 vs Immune-GDE3	0.34462	±(0.0063246)		DPSO-4 vs Immune-GDE3	0.59427	±(0.0534791)
	Immune-GDE3 vs MOEA/D-BR	**0.97633**	±(0.0004628)		Immune-GDE3 vs MOEA/D-BR	**0.96047**	±(0.0074297)
	MOEA/D-BR vs Immune-GDE3	0.93784	±(0.0007393)		MOEA/D-BR vs Immune-GDE3	0.89731	±(0.0031941)
	Immune-GDE3 vs SGEA	**0.98284**	±(0.0062702)		Immune-GDE3 vs SGEA	**0.99842**	±(0.0001074)
	SGEA vs Immune-GDE3	0.89351	±(0.0243818)		SGEA vs Immune-GDE3	0.24976	±(0.0040176)
	DPSO-4 vs MOEA/D-BR	0.60358	±(0.0189737)		DPSO-4 vs MOEA/D-BR	0.87075	±(0.0039425)
	MOEA/D-BR vs DPSO-4	**0.95739**	±(0.0006373)		MOEA/D-BR vs DPSO-4	**0.90373**	±(0.0007465)
	DPSO-4 vs SGEA	0.18374	±(0.0125711)		DPSO-4 vs SGEA	**0.88019**	±(0.0091594)
	SGEA vs DPSO-4	**0.98736**	±(0.0341637)		SGEA vs DPSO-4	0.45280	±(0.0010738)
	MOEA/D-BR vs SGEA	**0.96377**	±(0.0122939)		MOEA/D-BR vs SGEA	**0.95017**	±(0.0001037)
	SGEA vs MOEA/D-BR	0.87356	±(0.0417031)		SGEA vs MOEA/D-BR	0.13017	±(0.0030684)
FDA3	DNSGA-II-A vs DNSGA-II-B	**0.64640**	±(0.0040984)	UDF6	DNSGA-II-A vs DNSGA-II-B	**0.57094**	±(0.0084327)
	DNSGA-II-B vs DNSGA-II-A	0.38853	±(0.0028983)		DNSGA-II-B vs DNSGA-II-A	0.38934	±(0.0096609)
	DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-A vs Immune-GDE3	0.02740	±(0.0028095)
	Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-A	**0.99016**	±(0.0002075)
	DNSGA-II-A vs DPSO-4	0.59636	±(0.0024555)		DNSGA-II-A vs DPSO-4	0.23482	±(0.0198886)
	DPSO-4 vs DNSGA-II-A	**0.71078**	±(0.0097367)		DPSO-4 vs DNSGA-II-A	**0.89045**	±(0.0913292)
	DNSGA-II-A vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-A vs MOEA/D-BR	0	±(0.0000000)
	MOEA/D-BR vs DNSGA-II-A	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-A	**1**	±(0.0000000)
	DNSGA-II-A vs SGEA	0	±(0.0000000)		DNSGA-II-A vs SGEA	0.03629	±(0.0394824)
	SGEA vs DNSGA-II-A	**1**	±(0.0000000)		SGEA vs DNSGA-II-A	**0.99628**	±(0.0006264)
	DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs DPSO-4	0	±(0.0000000)		DNSGA-II-B vs DPSO-4	0.10737	±(0.0038955)
	DPSO-4 vs DNSGA-II-B	**1**	±(0.0000000)		DPSO-4 vs DNSGA-II-B	**0.87395**	±(0.0046375)
	DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)
	MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs SGEA	0	±(0.0000000)		DNSGA-II-B vs SGEA	0.07246	±(0.0071534)
	SGEA vs DNSGA-II-B	**1**	±(0.0000000)		SGEA vs DNSGA-II-B	**0.99184**	±(0.0004801)
	Immune-GDE3 vs DPSO-4	**0.87433**	±(0.0038974)		Immune-GDE3 vs DPSO-4	**0.83725**	±(0.0032379)
	DPSO-4 vs Immune-GDE3	0.32434	±(0.0036755)		DPSO-4 vs Immune-GDE3	0.71084	±(0.0083661)
	Immune-GDE3 vs MOEA/D-BR	**0.97882**	±(0.0006793)		Immune-GDE3 vs MOEA/D-BR	0.98037	±(0.0023664)
	MOEA/D-BR vs Immune-GDE3	0.93945	±(0.0035489)		MOEA/D-BR vs Immune-GDE3	**0.98537**	±(0.0031940)
	Immune-GDE3 vs SGEA	**0.96341**	±(0.0051781)		Immune-GDE3 vs SGEA	**0.92404**	±(0.0036442)
	SGEA vs Immune-GDE3	0.90726	±(0.0460324)		SGEA vs Immune-GDE3	0.86027	±(0.0064280)
	DPSO-4 vs MOEA/D-BR	0.65838	±(0.0073666)		DPSO-4 vs MOEA/D-BR	0.77033	±(0.0094832)
	MOEA/D-BR vs DPSO-4	**0.94634**	±(0.0005724)		MOEA/D-BR vs DPSO-4	**0.97902**	±(0.0076381)
	DPSO-4 vs SGEA	**0.91543**	±(0.0424100)		DPSO-4 vs SGEA	0.78462	±(0.0015745)
	SGEA vs DPSO-4	0.27053	±(0.0031291)		SGEA vs DPSO-4	**0.94810**	±(0.0010728)
	MOEA/D-BR vs SGEA	0.83644	±(0.0977426)		MOEA/D-BR vs SGEA	**0.93161**	±(0.0010748)
	SGEA vs MOEA/D-BR	**0.95016**	±(0.0411382)		SGEA vs MOEA/D-BR	0.82760	±(0.0086491)
UDF1	DNSGA-II-A vs DNSGA-II-B	**0.85373**	±(0.0093792)	UDF7	DNSGA-II-A vs DNSGA-II-B	**0.73527**	±(0.0063372)
	DNSGA-II-B vs DNSGA-II-A	0.41749	±(0.0064743)		DNSGA-II-B vs DNSGA-II-A	0.57305	±(0.0024584)
	DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-A vs Immune-GDE3	0.10769	±(0.0050384)
	Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-A	**0.98387**	±(0.0016485)
	DNSGA-II-A vs DPSO-4	0.63462	±(0.0005382)		DNSGA-II-A vs DPSO-4	–	–
	DPSO-4 vs DNSGA-II-A	**0.78045**	±(0.0073890)		DPSO-4 vs DNSGA-II-A	–	–
	DNSGA-II-A vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-A vs MOEA/D-BR	0.23039	±(0.0088579)
	MOEA/D-BR vs DNSGA-II-A	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-A	**0.92737**	±(0.0035292)
	DNSGA-II-A vs SGEA	0.41063	±(0.0350373)		DNSGA-II-A vs SGEA	0.84204	±(0.0026484)
	SGEA vs DNSGA-II-A	**0.82963**	±(0.0051631)		SGEA vs DNSGA-II-A	**0.89247**	±(0.0009626)
	DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-B vs Immune-GDE3	0.01314	±(0.0054813)
	Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-B	**0.98684**	±(0.0009106)
	DNSGA-II-B vs DPSO-4	0	±(0.0000000)		DNSGA-II-B vs DPSO-4	–	–
	DPSO-4 vs DNSGA-II-B	**1**	±(0.0000000)		DPSO-4 vs DNSGA-II-B	–	–
	DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)
	MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs SGEA	0.54257	±(0.0176409)		DNSGA-II-B vs SGEA	0.80187	±(0.0107485)
	SGEA vs DNSGA-II-B	**0.80378**	±(0.0360274)		SGEA vs DNSGA-II-B	**0.91024**	±(0.0038169)
	Immune-GDE3 vs DPSO-4	**1**	±(0.0000000)		Immune-GDE3 vs DPSO-4	–	–
	DPSO-4 vs Immune-GDE3	0	±(0.0000000)		DPSO-4 vs Immune-GDE3	–	–
	Immune-GDE3 vs MOEA/D-BR	**0.99635**	±(0.0000564)		Immune-GDE3 vs MOEA/D-BR	**0.99027**	±(0.0006436)
	MOEA/D-BR vs Immune-GDE3	0.96538	±(0.0005476)		MOEA/D-BR vs Immune-GDE3	0.98039	±(0.0012704)
	Immune-GDE3 vs SGEA	**0.97362**	±(0.0034037)		Immune-GDE3 vs SGEA	**0.97428**	±(0.0003178)
	SGEA vs Immune-GDE3	0.21036	±(0.0260462)		SGEA vs Immune-GDE3	0.90374	±(0.0048192)
	DPSO-4 vs MOEA/D-BR	0.72942	±(0.0056367)		DPSO-4 vs MOEA/D-BR	–	–
	MOEA/D-BR vs DPSO-4	**0.98540**	±(0.0005638)		MOEA/D-BR vs DPSO-4	–	–
	DPSO-4 vs SGEA	**0.92036**	±(0.0063801)		DPSO-4 vs SGEA	–	–
	SGEA vs DPSO-4	0.89015	±(0.0030763)		SGEA vs DPSO-4	–	–
	MOEA/D-BR vs SGEA	**0.94016**	±(0.0010763)		MOEA/D-BR vs SGEA	**0.93917**	±(0.0015714)

Continued on next page

Table 5.4 *Continued from previous page*

	Algorithm comparison	Mean	St. dev.		Algorithm comparison	Mean	St. dev.
	SGEA vs MOEA/D-BR	0.86971	±(0.0100358)		SGEA vs MOEA/D-BR	0.88197	±(0.0019646)
UDF2	DNSGA-II-A vs DNSGA-II-B	0.48047	±(0.0067486)	UDF8	DNSGA-II-A vs DNSGA-II-B	**0.67394**	±(0.0063974)
	DNSGA-II-B vs DNSGA-II-A	**0.69463**	±(0.0084327)		DNSGA-II-B vs DNSGA-II-A	0.44890	±(0.0093876)
	DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)
	DNSGA-II-A vs DPSO-4	0	±(0.0000000)		DNSGA-II-A vs DPSO-4	0.11835	±(0.0094365)
	DPSO-4 vs DNSGA-II-A	**1**	±(0.0000000)		DPSO-4 vs DNSGA-II-A	**0.87494**	±(0.0074989)
	DNSGA-II-A vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-A vs MOEA/D-BR	0.04639	±(0.0098485)
	MOEA/D-BR vs DNSGA-II-A	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-A	**0.97364**	±(0.0009849)
	DNSGA-II-A vs SGEA	**0.89163**	±(0.0036183)		DNSGA-II-A vs SGEA	0.36492	±(0.0123715)
	SGEA vs DNSGA-II-A	0.81063	±(0.0010374)		SGEA vs DNSGA-II-A	**0.84862**	±(0.0037810)
	DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs DPSO-4	0.53273	±(0.0003456)		DNSGA-II-B vs DPSO-4	0.06384	±(0.0007475)
	DPSO-4 vs DNSGA-II-B	**0.85039**	±(0.0063613)		DPSO-4 vs DNSGA-II-B	**0.88905**	±(0.0047429)
	DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)
	MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs SGEA	**0.91013**	±(0.0016591)		DNSGA-II-B vs SGEA	0.29107	±(0.0301831)
	SGEA vs DNSGA-II-B	0.80103	±(0.0030817)		SGEA vs DNSGA-II-B	**0.89201**	±(0.0016884)
	Immune-GDE3 vs DPSO-4	**1**	±(0.0000000)		Immune-GDE3 vs DPSO-4	**0.97994**	±(0.0063279)
	DPSO-4 vs Immune-GDE3	0	±(0.0000000)		DPSO-4 vs Immune-GDE3	0.01980	±(0.0069043)
	Immune-GDE3 vs MOEA/D-BR	**0.98037**	±(0.0046646)		Immune-GDE3 vs MOEA/D-BR	0.97394	±(0.0003274)
	MOEA/D-BR vs Immune-GDE3	0.97458	±(0.0002557)		MOEA/D-BR vs Immune-GDE3	**0.99068**	±(0.0006379)
	Immune-GDE3 vs SGEA	**0.99896**	±(0.0003618)		Immune-GDE3 vs SGEA	**0.96103**	±(0.0004017)
	SGEA vs Immune-GDE3	0.05183	±(0.0015319)		SGEA vs Immune-GDE3	0.89017	±(0.0030649)
	DPSO-4 vs MOEA/D-BR	0.49402	±(0.0096739)		DPSO-4 vs MOEA/D-BR	0.10835	±(0.0046785)
	MOEA/D-BR vs DPSO-4	**0.97359**	±(0.0004365)		MOEA/D-BR vs DPSO-4	**0.90027**	±(0.0006485)
	DPSO-4 vs SGEA	**0.81648**	±(0.0075103)		DPSO-4 vs SGEA	**0.91072**	±(0.0389100)
	SGEA vs DPSO-4	0.64364	±(0.0039049)		SGEA vs DPSO-4	0.87017	±(0.0010040)
	MOEA/D-BR vs SGEA	**0.96301**	±(0.0072917)		MOEA/D-BR vs SGEA	0.89104	±(0.0108420)
	SGEA vs MOEA/D-BR	0.12937	±(0.0010284)		SGEA vs MOEA/D-BR	**0.92074**	±(0.0030680)
UDF3	DNSGA-II-A vs DNSGA-II-B	**0.59075**	±(0.0069464)	UDF9	DNSGA-II-A vs DNSGA-II-B	**0.86383**	±(0.0083839)
	DNSGA-II-B vs DNSGA-II-A	0.32007	±(0.0040489)		DNSGA-II-B vs DNSGA-II-A	0.74949	±(0.0009876)
	DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-A vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-A	**1**	±(0.0000000)
	DNSGA-II-A vs DPSO-4	0.49365	±(0.0057337)		DNSGA-II-A vs DPSO-4	0.25191	±(0.0099375)
	DPSO-4 vs DNSGA-II-A	**0.77256**	±(0.0006464)		DPSO-4 vs DNSGA-II-A	**0.91764**	±(0.0064385)
	DNSGA-II-A vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-A vs MOEA/D-BR	0.02749	±(0.0004838)
	MOEA/D-BR vs DNSGA-II-A	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-A	**0.97937**	±(0.0008364)
	DNSGA-II-A vs SGEA	0.82674	±(0.0107441)		DNSGA-II-A vs SGEA	**0.86488**	±(0.0046794)
	SGEA vs DNSGA-II-A	**0.79817**	±(0.0086173)		SGEA vs DNSGA-II-A	0.72840	±(0.0027941)
	DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)		DNSGA-II-B vs Immune-GDE3	0	±(0.0000000)
	Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)		Immune-GDE3 vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs DPSO-4	0.02799	±(0.0011996)		DNSGA-II-B vs DPSO-4	0.03735	±(0.0048275)
	DPSO-4 vs DNSGA-II-B	**0.98699**	±(0.0029973)		DPSO-4 vs DNSGA-II-B	**0.88906**	±(0.0097374)
	DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)		DNSGA-II-B vs MOEA/D-BR	0	±(0.0000000)
	MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)		MOEA/D-BR vs DNSGA-II-B	**1**	±(0.0000000)
	DNSGA-II-B vs SGEA	**0.86016**	±(0.0015371)		DNSGA-II-B vs SGEA	**0.82049**	±(0.0013885)
	SGEA vs DNSGA-II-B	0.75017	±(0.0010836)		SGEA vs DNSGA-II-B	0.76012	±(0.0010874)
	Immune-GDE3 vs DPSO-4	**0.94644**	±(0.0309839)		Immune-GDE3 vs DPSO-4	**0.96939**	±(0.0047282)
	DPSO-4 vs Immune-GDE3	0.68747	±(0.0289828)		DPSO-4 vs Immune-GDE3	0.23810	±(0.0109375)
	Immune-GDE3 vs MOEA/D-BR	**0.98468**	±(0.0393743)		Immune-GDE3 vs MOEA/D-BR	**0.99037**	±(0.0002738)
	MOEA/D-BR vs Immune-GDE3	0.95375	±(0.0445446)		MOEA/D-BR vs Immune-GDE3	0.92937	±(0.0017393)
	Immune-GDE3 vs SGEA	**0.99167**	±(0.0001683)		Immune-GDE3 vs SGEA	**0.97017**	±(0.0001844)
	SGEA vs Immune-GDE3	0.08351	±(0.0080363)		SGEA vs Immune-GDE3	0.89875	±(0.0017684)
	DPSO-4 vs MOEA/D-BR	0.74364	±(0.8340465)		DPSO-4 vs MOEA/D-BR	0.52049	±(0.0053949)
	MOEA/D-BR vs DPSO-4	**0.96535**	±(0.0006365)		MOEA/D-BR vs DPSO-4	**0.97229**	±(0.0063931)
	DPSO-4 vs SGEA	**0.80930**	±(0.0062849)		DPSO-4 vs SGEA	**0.82545**	±(0.0010651)
	SGEA vs DPSO-4	0.43974	±(0.0010884)		SGEA vs DPSO-4	0.79462	±(0.0017910)
	MOEA/D-BR vs SGEA	**0.96017**	±(0.0003801)		MOEA/D-BR vs SGEA	**0.88017**	±(0.0049914)
	SGEA vs MOEA/D-BR	0.11082	±(0.0408384)		SGEA vs MOEA/D-BR	0.81920	±(0.0010958)

Table 5.5: Resume of Wilcoxon Statistical Test on $\overline{C}$ metric. "+" means that Immune GDE3 outperformed the algorithm in the corresponding row. "-" means that the algorithm in the corresponding row outperformed Immune GDE3. No significant differences between Immune GDE3 and the version in the corresponding row are indicated with "=".

		FDA1	FDA2	FDA3	UDF1	UDF2	UDF3	UDF4	UDF5	UDF6	UDF7	UDF8	UDF9	+	=	−
Immune-GDE3	DNSGA-II-A	+	+	+	+	+	+	+	+	+	+	+	+	12	0	0
	DNSGA-II-B	+	+	+	+	+	+	+	+	+	+	+	+	12	0	0
	DPSO-4	+	+	+	+	+	+	=	+	+	N/A	+	+	10	1	0
	MOEA/D-BR	=	+	+	+	=	+	+	+	+	=	=	+	8	4	0
	SGEA	=	+	+	+	+	+	+	+	+	+	+	+	11	1	0

5.3.2 Results of Experiment 2: Change frequency and severity analysis

The second experiment analyzed the effects of different change severities and frequencies in the performance of Immune GDE3 and MOEA/D-BR. From experiment one, it was observed that, in general, MOEA/D-BR was the second most competitive algorithm. For that reason, it was selected to carry out the analysis of frequency and severity. The experiments were computed with other three problem parameter configurations (n_t, τ_t)= (5,10), (10,10), and (10,5). The twelve test problems, as in Experiment 1, were solved and the obtained results were analyzed according to the performance metrics. In Tables 5.6, 5.7, 5.8, and 5.9 the results of the average and standard deviation of each metric over different change severities and frequencies are presented. These Tables also include a summary of the statistical test applied.

Proximity metrics discussion

Regarding the IGD metric, from Table 5.6, it was observed that Immune GDE3 outperformed MOEA/D-BR in all test problems for the last three configurations of the problem parameters. For configuration (n_t=10, τ_t=10), both algorithms presented better performance than the results obtained in the other combinations of change severity and frequency. This behavior could be attributed to the fact that a higher value of such parameters decreases the difficulty of the problem. From the same Table 5.6, it was also observed that Immune GDE3 presented a better performance than MOEA/D-BR when the problems have a high change frequency. In contrast, MOEA/D-BR obtained good results when the changes are less frequent,

Table 5.6: $\overline{IGD}$ mean and standard deviation values for all test problems over different configurations of problem parameters (n_t, τ_t) and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune GDE3 outperformed MOEA/D-BR. "-" means that MOEA/D-BR outperformed Immune GDE3. No significant differences between Immune GDE3 and MOEA/D-BR are indicated with "=". The best results are marked in **boldface**

Test Problem	**Algorithm**	**(5,5)**			**(5,10)**			**(10,10)**			**(10,5)**		
		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.	
FDA1	Immune-GDE3	**0.055062**	±(1.0464E-3)	+	**0.048831**	±(5.9498E-4)	+	**0.029793**	±(2.3621E-4)	+	**0.031860**	±(4.1996E-3)	+
	MOEA/D-BR	0.086006	±(6.1657E-3)		0.077389	±(4.1316E-3)		0.059057	±(1.2079E-3)		0.067092	±(2.3079E-3)	
FDA2	Immune-GDE3	**0.015333**	±(2.4733E-3)	=	**0.011475**	±(8.0401E-4)	+	**0.009862**	±(4.1849E-4)	+	**0.011642**	±(1.4059E-3)	+
	MOEA/D-BR	0.015573	±(1.4357E-3)		0.016149	±(2.7805E-3)		0.010136	±(3.2106E-3)		0.015224	±(1.5365E-3)	
FDA3	Immune-GDE3	**0.134456**	±(8.7314E-3)	+	**0.111877**	±(8.8516E-3)	+	**0.087729**	±(6.0179E-3)	+	**0.104331**	±(5.1049E-3)	+
	MOEA/D-BR	0.163323	±(1.6654E-2)		0.150646	±(6.0251E-3)		0.134362	±(7.9978E-3)		0.139922	±(5.5347E-3)	
UDF1	Immune-GDE3	**0.126877**	±(4.6714E-3)	+	**0.110879**	±(5.5627E-3)	+	**0.097518**	±(3.2627E-3)	+	**0.111225**	±(2.3434E-3)	+
	MOEA/D-BR	0.146588	±(8.0849E-3)		0.136125	±(3.3260E-3)		0.113874	±(2.5029E-3)		0.126736	±(5.5535E-3)	
UDF2	Immune-GDE3	**0.035197**	±(1.2918E-3)	=	**0.021089**	±(5.8055E-3)	+	**0.016623**	±(3.6932E-3)	+	**0.021952**	±(2.3790E-3)	+
	MOEA/D-BR	0.038836	±(6.9585E-4)		0.042809	±(2.6436E-3)		0.031538	±(4.7244E-3)		0.037472	±(5.0232E-3)	
UDF3	Immune-GDE3	**0.313187**	±(6.9004E-3)	+	**0.395608**	±(9.0219E-3)	+	**0.212443**	±(2.8595E-3)	+	**0.356359**	±(1.2050E-2)	+
	MOEA/D-BR	0.428278	±(1.8934E-2)		0.456038	±(1.9513E-2)		0.411316	±(1.3250E-2)		0.414984	±(1.0310E-2)	
UDF4	Immune-GDE3	**0.208323**	±(1.6900E-2)	+	**0.194531**	±(1.5068E-2)	+	**0.150834**	±(1.8599E-3)	+	**0.192119**	±(1.9370E-2)	+
	MOEA/D-BR	0.341806	±(2.3781E-2)		0.328705	±(1.7042E-2)		0.321557	±(2.0079E-2)		0.331420	±(2.1199E-2)	
UDF5	Immune-GDE3	**0.022426**	±(7.9638E-4)	=	**0.020183**	±(1.9755E-4)	=	**0.016473**	±(3.1760E-3)	+	**0.019792**	±(1.3847E-3)	+
	MOEA/D-BR	0.034523	±(1.5078E-3)		0.029316	±(1.1285E-3)		0.029786	±(4.5544E-3)		0.034826	±(2.9085E-3)	
UDF6	Immune-GDE3	1.261284	±(2.9028E-2)	=	**1.206605**	±(1.5330E-2)	+	**0.992931**	±(8.0603E-3)	+	**1.168535**	±(1.0141E-2)	+
	MOEA/D-BR	**1.235593**	±(2.0391E-2)		1.222786	±(1.7668E-2)		1.184477	±(5.8419E-2)		1.226356	±(1.6007E-2)	
UDF7	Immune-GDE3	**0.215578**	±(9.3813E-3)	=	**0.185661**	±(1.3057E-2)	+	**0.172417**	±(9.5688E-3)	+	**0.201356**	±(4.8354E-3)	+
	MOEA/D-BR	0.235467	±(9.4661E-3)		0.222149	±(1.5211E-2)		0.209706	±(9.5929E-3)		0.227930	±(5.2235E-3)	
UDF8	Immune-GDE3	**0.399400**	±(1.1659E-2)	+	**0.295218**	±(1.9109E-3)	+	**0.217609**	±(2.7210E-2)	+	**0.256169**	±(2.9525E-3)	+
	MOEA/D-BR	0.467142	±(5.2666E-2)		0.417166	±(5.7478E-2)		0.358355	±(3.2235E-2)		0.412264	±(5.2692E-2)	
UDF9	Immune-GDE3	**0.119658**	±(5.8976E-3)	+	**0.119730**	±(5.3391E-3)	+	**0.100523**	±(6.2796E-3)	+	**0.105342**	±(5.0707E-3)	+
	MOEA/D-BR	0.141160	±(1.1506E-2)		0.163020	±(4.7898E-3)		0.138796	±(1.0648E-2)		0.150768	±(1.1389E-2)	
Summary of the statistical test	Immune GDE3	+		7			11			12			12
	vs	-		0			0			0			0
	MOEA/D-BR	=		5			1			0			0

i.e. the algorithm needs more time to converge to the Pareto optimal front. Table 5.6 also included the results obtained from the configuration (n_t=5, τ_t=5). As it can be observed from the previous experiment, Immune GDE3 obtained better results than MOEA/D-BR in most problems. However, the statistical test showed that the behavior of both algorithms was similar in seven of twelve problems. The results obtained with this experiment showed that Immune GDE3 was less sensitive to the variation of the change severity and frequency parameters. In addition, the performance of Immune GDE3 was better in comparison with MOEA/D-BR. The Bonferroni-Dunn post-hoc test results confirm such findings.

For the Hypervolume results, the behavior of both algorithms in this experiment

was very similar to that observed for the IGD results. Immune GDE3 obtained better results in most test problems. The variation of problem parameters improved the performance of Immune GDE3. In contrast, the results of MOEA/D-BR were affected. Less severity and less frequent changes enhanced the performance of both algorithms. However, the results obtained by MOEA/D-BR do not outperform Immune GDE3 results. Analyzing more thoroughly the obtained results, from Table 5.7 it can also be observed that, for new configurations of frequencies and severities in UDF4, UDF5, and UDF9 test problems, the statistical test showed that Immune GDE3 results are not significantly different to MOEA/D-BR results. These test problems are problems where both, Pareto optimal set and Pareto optimal front changes. On the other hand, according to the Hypervolume metric results in test problems UDF3, UDF6, and UDF7, Immune GDE3 obtained better results than MOEA/D-BR when the Pareto optimal set does not change over time. Therefore, one disadvantage of Immune GDE3 regarding proximity metrics is that its performance tends to decrease when solving problems with changing Pareto optimal sets (POSs) and with more than two objectives. Such behavior can occur in problems where the severity of change is considerable, and the changes are very frequent.

Distribution metrics discussion

In spite of the variation of problem parameters, the results of distribution metrics showed a better performance of Immune GDE3 in most test problems. In contrast with proximity metrics, for both distribution metrics in configuration (n_t=5, τ_t=5), the results obtained by Immune GDE3 were better than those of MOEA/D-BR. According to the Spacing results, the variation of problem parameters also improved significantly the performance of Immune GDE3. As it can be seen in Table 5.8, with the exception of the first configuration, Immune GDE3 obtained significantly better results with respect to MOEA/D-BR in all test problems, i.e., the distribution of solutions of Immune GDE3 was better when the severity and frequency parameters are changed.

As regards to the binary metric, C-metric measures the percentage of non-dominated solutions in an algorithm with respect to another algorithm. The obtained results suggested a good performance of both algorithms in the original configuration

Table 5.7: $\overline{HV}$ mean and standard deviation values for all test problems over different configurations of problem parameters (n_t, τ_t) and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune GDE3 outperformed MOEA/D-BR. "-" means that MOEA/D-BR outperformed Immune GDE3. No significant differences between Immune GDE3 and MOEA/D-BR are indicated with "=". The best results are marked in **boldface**

Test Problem	**Algorithm**	**(5,5)**			**(5,10)**			**(10,10)**			**(10,5)**		
		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.	
FDA1	Immune-GDE3	**0.947598**	±(1.2433E-4)	+	**0.928808**	±(2.2660E-2)	=	**0.970845**	±(8.8909E-3)	+	**0.960830**	±(8.2739E-3)	+
	MOEA/D-BR	0.915952	±(1.8250E-2)		0.912243	±(9.3628E-3)		0.938588	±(1.3300E-2)		0.910441	±(1.0145E-2)	
FDA2	Immune-GDE3	**0.942487**	±(1.2731E-2)	=	**0.963906**	±(9.3028E-3)	+	**0.979100**	±(9.9910E-3)	+	**0.967054**	±(1.6193E-2)	+
	MOEA/D-BR	0.934556	±(2.0434E-2)		0.905675	±(2.5070E-2)		0.930355	±(1.4331E-2)		0.927364	±(2.2449E-2)	
FDA3	Immune-GDE3	**0.906268**	±(1.3549E-2)	+	**0.935259**	±(1.1321E-2)	+	**0.963346**	±(1.3653E-2)	+	**0.951589**	±(1.5050E-2)	+
	MOEA/D-BR	0.887728	±(1.3071E-2)		0.876368	±(1.9911E-2)		0.898755	±(1.5213E-2)		0.884185	±(1.4845E-2)	
UDF1	Immune-GDE3	**0.923244**	±(1.3869E-2)	=	**0.952185**	±(6.9927E-3)	+	**0.969338**	±(1.3314E-2)	+	**0.950051**	±(1.5183E-2)	+
	MOEA/D-BR	0.902568	±(1.3726E-2)		0.895724	±(1.4137E-2)		0.907699	±(1.3394E-2)		0.893775	±(1.5449E-2)	
UDF2	Immune-GDE3	**0.975362**	±(1.1036E-2)	=	**0.983420**	±(8.2933E-3)	+	**0.993274**	±(3.1250E-3)	+	**0.985798**	±(5.7181E-3)	+
	MOEA/D-BR	0.956223	±(8.2531E-3)		0.943379	±(4.7765E-3)		0.950705	±(9.5585E-3)		0.947104	±(7.5593E-3)	
UDF3	Immune-GDE3	**0.898526**	±(1.7168E-2)	=	**0.944853**	±(1.2774E-2)	+	**0.960542**	±(1.1923E-2)	+	**0.938894**	±(1.4168E-2)	+
	MOEA/D-BR	0.895563	±(1.0683E-2)		0.897970	±(8.1838E-3)		0.910003	±(1.1155E-2)		0.885575	±(9.2959E-3)	
UDF4	Immune-GDE3	**0.934419**	±(1.4374E-2)	=	**0.945811**	±(9.6780E-3)	+	**0.972065**	±(1.2397E-2)	+	**0.951553**	±(1.0932E-2)	+
	MOEA/D-BR	0.902486	±(1.5699E-2)		0.889678	±(1.5807E-2)		0.915648	±(1.5669E-2)		0.894837	±(1.6574E-2)	
UDF5	Immune-GDE3	**0.983015**	±(7.8875E-3)	=	**0.986187**	±(4.0528E-3)	+	**0.994307**	±(3.8465E-3)	=	**0.994307**	±(2.4172E-3)	+
	MOEA/D-BR	0.972967	±(1.0906E-2)		0.953340	±(1.0908E-2)		0.975223	±(8.5292E-3)		0.953644	±(9.4308E-3)	
UDF6	Immune-GDE3	**0.868829**	±(1.4723E-2)	=	**0.934086**	±(7.4488E-3)	+	**0.952580**	±(8.9889E-3)	+	**0.935324**	±(1.3588E-2)	+
	MOEA/D-BR	0.864445	±(2.5743E-2)		0.864810	±(2.1600E-2)		0.889254	±(2.2972E-2)		0.861071	±(1.7646E-2)	
UDF7	Immune-GDE3	**0.901520**	±(1.6464E-2)	=	**0.945925**	±(9.4995E-3)	+	**0.964274**	±(8.5407E-3)	+	**0.942143**	±(9.3416E-3)	+
	MOEA/D-BR	0.901212	±(1.7895E-2)		0.881447	±(1.9348E-2)		0.904925	±(1.7147E-2)		0.883290	±(1.4403E-2)	
UDF8	Immune-GDE3	**0.903051**	±(4.4884E-3)	+	**0.936333**	±(1.1041E-2)	+	**0.961585**	±(6.2486E-3)	+	**0.936436**	±(9.9165E-3)	+
	MOEA/D-BR	0.878897	±(6.9378E-3)		0.880893	±(1.7555E-2)		0.894767	±(1.3142E-2)		0.860880	±(1.4674E-2)	
UDF9	Immune-GDE3	**0.924763**	±(1.4725E-2)	=	**0.952338**	±(1.0925E-2)	+	**0.968181**	±(9.7412E-3)	+	0.901031	±(1.0209E-2)	=
	MOEA/D-BR	0.921146	±(1.8088E-2)		0.893129	±(1.4687E-2)		0.919203	±(1.0137E-2)		**0.903308**	±(1.1092E-2)	
Summary of the statistical test	Immune-GDE3	+		3			11			11			11
	vs	−		0			0			0			0
	MOEA/D-BR	=		9			1			1			1

Table 5.8: $\overline{S}$ mean and standard deviation values for all test problems over different configurations of problem parameters (n_t, τ_t) and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune GDE3 outperformed MOEA/D-BR. "−" means that MOEA/D-BR outperformed Immune GDE3. No significant differences between Immune GDE3 and MOEA/D-BR are indicated with "=". The best results are marked in **boldface**

Test Problem	**Algorithm**	**(5,5)**			**(5,10)**			**(10,10)**			**(10,5)**		
		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.	
FDA1	Immune-GDE3	0.137552	±(1.2220E-2)	=	**0.025033**	±(6.4731E-3)	+	**0.017481**	±(3.9889E-3)	+	**0.050171**	±(6.4195E-3)	+
	MOEA/D-BR	**0.134540**	±(9.9702E-3)		0.099722	±(1.1576E-2)		0.045790	±(3.5958E-3)		0.092845	±(8.6534E-3)	
FDA2	Immune-GDE3	**0.043993**	±(6.9097E-3)	+	**0.026975**	±(9.4391E-3)	+	**0.020736**	±(6.5629E-3)	+	**0.030845**	±(5.7175E-3)	+
	MOEA/D-BR	0.085317	±(9.7344E-3)		0.080080	±(1.2033E-2)		0.061372	±(9.7295E-3)		0.064074	±(8.8591E-3)	
FDA3	Immune-GDE3	**0.142330**	±(5.4645E-3)	+	**0.133573**	±(9.0981E-3)	+	**0.065943**	±(8.1560E-3)	+	**0.096434**	±(8.6304E-3)	+
	MOEA/D-BR	0.205672	±(1.5007E-2)		0.173396	±(1.2050E-2)		0.147248	±(9.4471E-3)		0.160816	±(1.0114E-2)	
UDF1	Immune-GDE3	**0.088819**	±(8.4001E-3)	=	**0.067456**	±(9.6953E-3)	+	**0.030420**	±(7.7871E-3)	+	**0.050265**	±(9.3228E-3)	+
	MOEA/D-BR	0.089691	±(1.3500E-2)		0.075875	±(1.1019E-2)		0.069938	±(1.1472E-2)		0.100073	±(1.1170E-2)	
UDF2	Immune-GDE3	**0.024929**	±(6.5317E-3)	=	**0.020446**	±(5.2018E-3)	+	**0.009071**	±(7.7264E-4)	+	**0.014738**	±(3.0603E-3)	+
	MOEA/D-BR	0.043629	±(1.5216E-2)		0.030671	±(6.8962E-3)		0.023947	±(8.2712E-3)		0.029495	±(7.1050E-3)	
UDF3	Immune-GDE3	**0.115938**	±(1.0428E-2)	+	**0.082287**	±(5.2974E-3)	+	**0.064997**	±(7.1295E-3)	+	**0.069602**	±(6.8790E-3)	+
	MOEA/D-BR	0.157694	±(5.6919E-3)		0.131308	±(1.1971E-2)		0.105714	±(8.4053E-3)		0.130333	±(1.2652E-2)	
UDF4	Immune-GDE3	**0.091096**	±(1.0267E-2)	+	**0.073403**	±(1.0969E-2)	+	**0.042969**	±(8.4732E-3)	+	**0.052019**	±(6.3966E-3)	+
	MOEA/D-BR	0.171819	±(1.1848E-2)		0.150753	±(1.2261E-2)		0.106371	±(1.1433E-2)		0.121100	±(1.0974E-2)	
UDF5	Immune-GDE3	**0.024981**	±(7.9183E-3)	+	**0.070373**	±(1.1751E-2)	+	**0.041697**	±(8.1481E-3)	+	**0.054058**	±(7.3107E-3)	+
	MOEA/D-BR	0.083017	±(1.0824E-2)		0.151248	±(1.1783E-2)		0.106060	±(1.1248E-2)		0.118747	±(1.2771E-2)	
UDF6	Immune-GDE3	**0.089574**	±(1.0217E-2)	+	**0.058960**	±(1.2508E-2)	+	**0.044052**	±(9.6669E-3)	+	**0.065002**	±(8.7641E-3)	+
	MOEA/D-BR	0.132454	±(1.5807E-2)		0.111002	±(1.0324E-2)		0.090693	±(1.1628E-2)		0.115420	±(1.4133E-2)	
UDF7	Immune-GDE3	**0.117410**	±(7.6001E-3)	+	**0.074564**	±(7.8961E-3)	+	**0.058277**	±(7.3233E-3)	+	**0.067845**	±(4.8613E-3)	+
	MOEA/D-BR	0.176457	±(1.4566E-2)		0.128057	±(1.1451E-2)		0.112580	±(1.3723E-2)		0.135562	±(1.2408E-2)	
UDF8	Immune-GDE3	**0.102524**	±(1.0831E-2)	+	**0.066819**	±(9.8009E-3)	+	**0.036642**	±(7.9870E-3)	+	**0.055002**	±(1.2664E-2)	+
	MOEA/D-BR	0.181500	±(2.4492E-2)		0.130808	±(1.2127E-2)		0.101421	±(1.5587E-2)		0.145612	±(1.2201E-2)	
UDF9	Immune-GDE3	**0.068341**	±(1.3208E-2)	+	**0.050860**	±(1.0565E-2)	+	**0.020028**	±(6.9827E-3)	+	**0.044439**	±(9.2070E-3)	+
	MOEA/D-BR	0.119138	±(1.0972E-2)		0.095114	±(1.1128E-2)		0.073072	±(1.3588E-2)		0.114383	±(1.2963E-2)	
Summary of the statistical test	Immune-GDE3	+		9			12			12			12
	vs	−		0			0			0			0
	MOEA/D-BR	=		3			0			0			0

Table 5.9: $\overline{C}$ mean and standard deviation values for all test problems over different configurations of problem parameters (n_t, τ_t) and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test . "+" means that Immune GDE3 outperformed MOEA/D-BR. "−" means that MOEA/D-BR outperformed Immune GDE3. No significant differences between Immune GDE3 and MOEA/D-BR are indicated with "=". The best results are marked in **boldface**

Test Problem	**Algorithm**	**(5,5)**			**(5,10)**			**(10,10)**			**(10,5)**		
		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.		Mean	St. dev.	
FDA1	Immune-GDE3 vs MOEA/D-BR	**0.998172**	±(4.4362E-4)	=	**0.998864**	±(2.8681E-4)	+	**0.999527**	±(2.1406E-4)	+	**0.999073**	±(6.2261E-5)	+
	MOEA/D-BR vs Immune-GDE3	0.957717	±(2.7748E-3)		0.918381	±(1.1195E-2)		0.924294	±(1.9926E-2)		0.886633	±(1.2974E-2)	
FDA2	Immune-GDE3 vs MOEA/D-BR	**0.993593**	±(2.8622E-3)	+	**0.992464**	±(1.3749E-3)	+	**0.998948**	±(4.7650E-4)	+	**0.989350**	±(6.1540E-4)	+
	MOEA/D-BR vs Immune-GDE3	0.878797	±(1.3513E-2)		0.920525	±(4.0009E-3)		0.917033	±(1.2609E-2)		0.890612	±(1.1278E-2)	
FDA3	Immune-GDE3 vs MOEA/D-BR	**0.975394**	±(3.5148E-3)	+	**0.982105**	±(1.1593E-3)	+	**0.990654**	±(4.5009E-3)	+	**0.993083**	±(2.3690E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.800582	±(1.7528E-2)		0.806515	±(2.1632E-2)		0.840118	±(2.3950E-2)		0.816525	±(7.2115E-3)	
UDF1	Immune-GDE3 vs MOEA/D-BR	**0.994410**	±(2.0067E-3)	+	**0.995974**	±(1.3592E-3)	+	**0.997587**	±(1.3556E-3)	+	**0.997012**	±(9.9663E-4)	+
	MOEA/D-BR vs Immune-GDE3	0.823839	±(1.3659E-2)		0.816665	±(2.9042E-2)		0.854602	±(1.0090E-2)		0.792722	±(3.0194E-2)	
UDF2	Immune-GDE3 vs MOEA/D-BR	**0.985741**	±(2.6844E-3)	=	**0.992413**	±(2.2375E-3)	+	**0.995233**	±(2.1778E-3)	+	**0.987917**	±(2.1424E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.962914	±(7.6618E-3)		0.822160	±(1.4852E-2)		0.856756	±(1.7087E-2)		0.786150	±(1.0563E-2)	
UDF3	Immune-GDE3 vs MOEA/D-BR	**0.987248**	±(3.7431E-3)	+	**0.990409**	±(3.3405E-4)	+	**0.999930**	±(3.7802E-5)	+	**0.993750**	±(1.4272E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.860455	±(9.6686E-3)		0.913036	±(1.0546E-2)		0.906851	±(2.0068E-2)		0.855270	±(9.2544E-3)	
UDF4	Immune-GDE3 vs MOEA/D-BR	**0.988880**	±(5.0691E-3)	+	**0.992855**	±(1.5444E-3)	+	**0.998935**	±(5.3612E-4)	+	**0.987561**	±(1.8892E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.773828	±(5.8370E-2)		0.779123	±(5.9639E-3)		0.811210	±(2.0083E-2)		0.784044	±(1.2012E-2)	
UDF5	Immune-GDE3 vs MOEA/D-BR	**0.961725**	±(9.3726E-4)	+	**0.994973**	±(1.7718E-3)	+	**0.997674**	±(1.3696E-3)	+	**0.984572**	±(1.1385E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.892246	±(5.5369E-3)		0.842911	±(3.8983E-3)		0.891091	±(9.4304E-3)		0.893447	±(5.0208E-3)	
UDF6	Immune-GDE3 vs MOEA/D-BR	**0.989206**	±(4.7177E-3)	+	**0.994297**	±(2.1119E-3)	+	**0.999658**	±(2.0045E-4)	+	**0.992869**	±(2.3189E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.779203	±(1.4342E-2)		0.807311	±(4.3781E-2)		0.837987	±(3.5337E-2)		0.798030	±(1.9983E-2)	
UDF7	Immune-GDE3 vs MOEA/D-BR	0.974187	±(2.2222E-3)	=	**0.990862**	±(1.0140E-3)	+	**0.998881**	±(3.4431E-4)	+	**0.994303**	±(1.2911E-3)	+
	MOEA/D-BR vs Immune-GDE3	**0.974815**	±(2.5236E-3)		0.902326	±(1.5909E-2)		0.892530	±(2.2927E-2)		0.870319	±(1.9367E-2)	
UDF8	Immune-GDE3 vs MOEA/D-BR	0.983829	±(3.4514E-3)	=	**0.986707**	±(2.2274E-3)	+	**0.987418**	±(1.6308E-3)	=	**0.995063**	±(1.4771E-3)	+
	MOEA/D-BR vs Immune-GDE3	**0.985947**	±(2.2279E-3)		0.895627	±(1.5831E-2)		0.985501	±(2.1721E-3)		0.868734	±(1.7605E-2)	
UDF9	Immune-GDE3 vs MOEA/D-BR	**0.991359**	±(8.6822E-4)	+	**0.994123**	±(8.9832E-4)	+	**0.992920**	±(2.8231E-3)	+	**0.981228**	±(3.7699E-3)	+
	MOEA/D-BR vs Immune-GDE3	0.908888	±(6.3919E-3)		0.867226	±(3.8292E-3)		0.880611	±(6.0838E-3)		0.875512	±(3.3347E-3)	
Summary of the statistical test	Immune-GDE3 vs MOEA/D-BR	+		8			12			11			12
		−		0			0			0			0
		=		4			0			1			0

of parameters (See Table 5.9). However, the percentage of non-dominated solutions of Immune GDE3 always outperformed the results of MOEA/D-BR in the rest of configurations. In contrast with the spacing metric, the results of MOEA/D-BR were less competitive than Immune GDE3 results. The variation of change frequency and change severity affects the performance of MOEA/D-BR significantly. In addition, and in the same way of proximity results, for MOEA/D-BR it is more difficult to obtain a better percentage of non-dominated solutions in the presence of many changes in the environment. On the other hand, the performance of Immune GDE3 was not affected when the severity of change was increased.

In order to visualize the algorithms' tracking ability over the changing environments, Pareto-front plots of the two best algorithms over three representative test

problems (FDA1, FDA2, and UDF2) are shown in Fig. 5.4. Each of the three problems selected represents a different type of DMOP, either type I, type III or type II, respectively. Fig. 5.4 shows that Immune GDE3 is very capable of tracking environmental changes, obtaining solutions with good distribution over the POF. Such behavior confirms the obtained results by proximity and distribution metrics.

5.3.3 Results of Experiment 3: The role of immune response

Finally, the first part of the third experiment compared Immune GDE3 against other versions of the GDE3 algorithm that use a different mechanism to react after environmental changes. This part of the experiment was designed to analyze the role of the immune response like change reaction mechanism within GDE3. As it was mentioned before, three different mechanisms were selected to evaluate the performance of the immune response. These mechanisms are the same mechanisms implemented in the algorithms selected in experiment one. For comparison purposes, the four metrics already used in the other two experiments were adopted here to analyze the obtained behaviors. In the two previous experiments, the configuration parameters (n_t=5 and τ_t=5) showed to add difficulty to the test problems. Due to this reason, in the same way of the first experiment, the parameter configuration n_t=5 and τ_t=5 was selected to compute the results of the algorithms used in this third experiment.

Table 5.10 summarizes the obtained results by each algorithm in each test problem. At the end of the table, the results of the Bergmann-Hommels post-hoc test are included. As it can be seen, similarly to the two previous experiments, Immune version of GDE3 obtained good results in most test problems.

Proximity metrics discussion

Regarding the IGD metric, from Table 5.10, it can be seen that Immune GDE3 obtained better results in ten of the twelve test problems and such differences were statistically validated. When Immune GDE3 was compared against GDE3-A, Immune GDE3 was better in most test problems, with the exception of FDA2, where the statistical test showed not significant difference between these two algorithms.

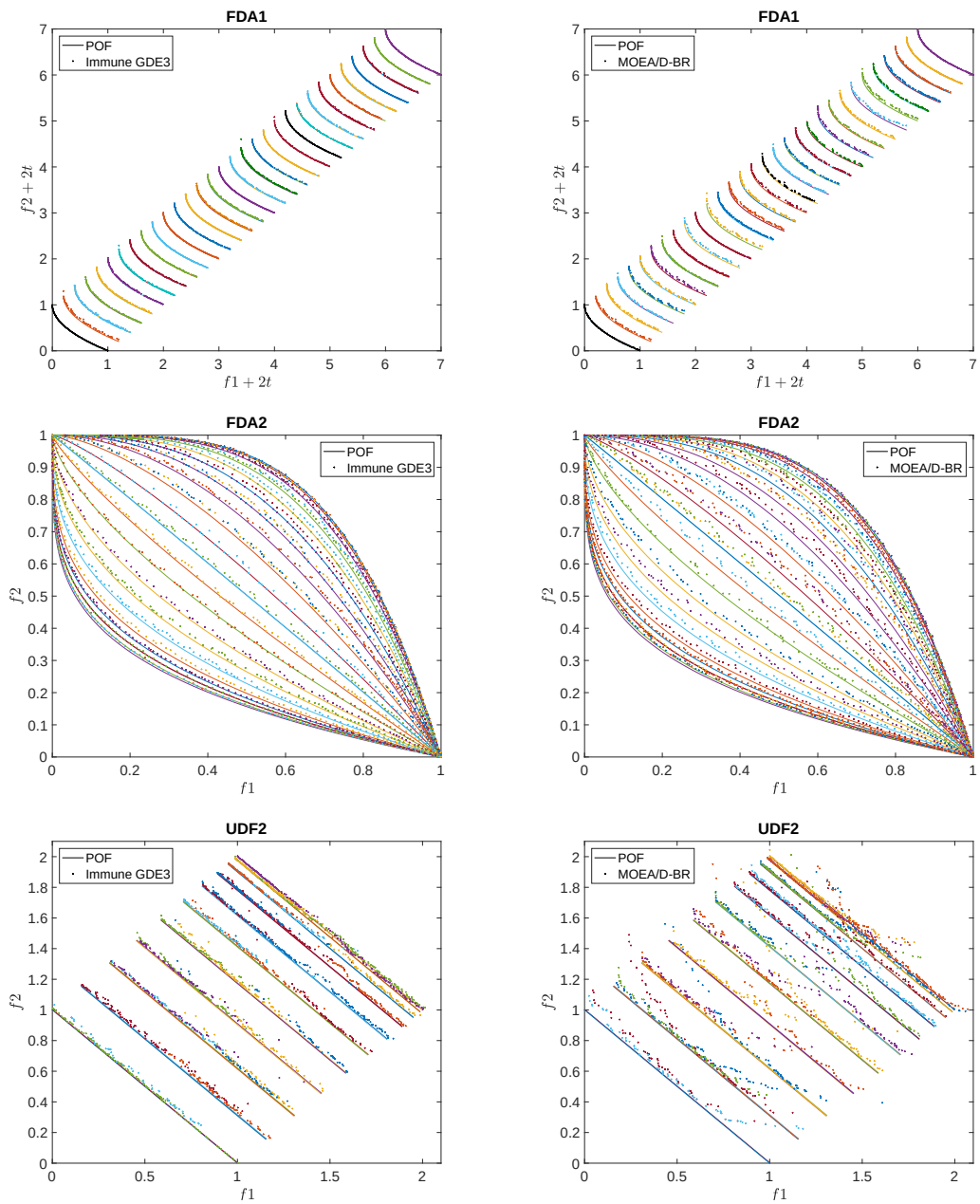

Figure 5.4: Obtained POFs by the two best algorithms on three representative test problems with $n_t = 10$ and $\tau_t = 10$.

Furthermore, when Immune GDE3 was compared against GDE3-B, Immune GDE3 outperformed GDE3-B in all test problems.

The prediction version of GDE3 (GDE3-BR) provided a different behavior with respect to the other two GDE3 versions. GDE3-BR was more competitive than GDE3-A and GDE3-B. However, the results obtained by GDE3-BR did not outperform those of Immune GDE3. The statistical test indicated that the performance of Immune GDE3 was better in seven test problems and similar to GDE3-BR in the remaining five.

As regards to HV, Immune GDE3 provided better results to those obtained by GDE3-A in eleven test problems. Moreover, Immune GDE3 was better than GDE3-B in all test problems and provided a similar performance as the observed for the IGD metric, when it was compared with GDE3-BR. For this metric, the immune response mechanism was more competitive in Type II and Type III problems, i.e., problems with changes in the POS and POF or changes only in POF.

Table 5.10: $\overline{IGD}$, $\overline{HV}$ and $\overline{S}$ mean and standard deviation values for all test problems on the third experiment and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune GDE3 outperformed the algorithm in the corresponding row. "−" means that the algorithm in the corresponding row outperformed Immune GDE3. No significant differences between Immune GDE3 and the version in the corresponding row are indicated with "=". The best results are marked in **boldface**

Test Problem		**IGD**			**HV**			**S**		
	Algorithm	Mean	St. dev.		Mean	St. dev.		Mean	St. dev.	
	Immune-GDE3	**0.055062**	±(1.0464E-3)		**0.947598**	±(1.2433E-4)		0.137552	±(1.2220E-2)	
FDA1	GDE3-A	0.092881	±(8.0243E-3)	+	0.871685	±(1.8488E-2)	+	0.163171	±(1.5182E-2)	+
	GDE3-B	0.112551	±(6.4481E-3)	+	0.813126	±(2.1174E-2)	+	0.185558	±(2.7120E-2)	+
	GDE3-BR	0.076914	±(9.7872E-3)	+	0.903858	±(1.4179E-2)	+	**0.123808**	±(1.0414E-2)	=
	Immune-GDE3	0.015333	±(2.4733E-3)		**0.942487**	±(1.2731E-2)		**0.043993**	±(6.9097E-3)	
FDA2	GDE3-A	0.017761	±(1.0292E-3)	=	0.905606	±(1.3413E-2)	+	0.094181	±(8.5774E-3)	+
	GDE3-B	0.024153	±(2.6093E-3)	+	0.895457	±(1.6187E-2)	+	0.111749	±(9.9340E-3)	+
	GDE3-BR	**0.014908**	±(1.9716E-3)	=	0.925525	±(9.3340E-3)	=	0.086791	±(4.6064E-3)	+
	Immune-GDE3	**0.134456**	±(8.7314E-3)		**0.906268**	±(1.3549E-2)		**0.142330**	±(5.4645E-3)	
FDA3	GDE3-A	0.188140	±(1.3340E-2)	+	0.883443	±(1.2732E-2)	=	0.227166	±(1.6766E-2)	+
	GDE3-B	0.192935	±(1.3632E-2)	+	0.850149	±(1.5455E-2)	+	0.252608	±(1.4645E-2)	+
	GDE3-BR	0.152803	±(1.2208E-2)	=	0.876620	±(1.3360E-2)	+	0.191743	±(1.1709E-2)	+
	Immune-GDE3	**0.126877**	±(4.6714E-3)		**0.923244**	±(1.3869E-2)		**0.088819**	±(8.4001E-3)	
UDF1	GDE3-A	0.163256	±(9.9488E-3)	+	0.850475	±(9.1815E-3)	+	0.123365	±(1.0451E-2)	+
	GDE3-B	0.181585	±(1.6598E-2)	+	0.845393	±(9.0315E-3)	+	0.125337	±(1.9746E-2)	+
	GDE3-BR	0.160167	±(1.1402E-2)	+	0.906104	±(1.4060E-2)	=	0.123690	±(8.8349E-3)	+
	Immune-GDE3	**0.035197**	±(1.2918E-3)		**0.975362**	±(1.1036E-2)		**0.024929**	±(6.5317E-3)	
UDF2	GDE3-A	0.063155	±(9.9167E-3)	+	0.906550	±(2.1470E-2)	+	0.082639	±(1.6363E-2)	+
	GDE3-B	0.069553	±(7.4715E-3)	+	0.901151	±(1.7944E-2)	+	0.123219	±(9.1922E-3)	+

Continued on next page

Table 5.10 *Continued from previous page*

Test Problem		**IGD**		**HV**		**S**	
	GDE3-BR	0.038409 ±(7.3735E-4)	=	0.900431 ±(2.2435E-2)	+	0.070547 ±(1.2376E-2)	+
	Immune-GDE3	**0.313187**±(6.9004E-3)		**0.898526**±(1.7168E-2)		**0.115938**±(1.0428E-2)	
UDF3	GDE3-A	0.462604 ±(1.3399E-2)	+	0.854321 ±(4.6096E-3)	+	0.155045 ±(1.2896E-2)	+
	GDE3-B	0.486240 ±(1.4030E-2)	+	0.856425 ±(8.2676E-3)	+	0.200869 ±(1.1406E-2)	+
	GDE3-BR	0.446327 ±(9.1276E-3)	+	0.885519 ±(1.3592E-2)	=	0.175363 ±(3.2044E-3)	+
	Immune-GDE3	**0.208323**±(1.6900E-2)		**0.934419**±(1.4374E-2)		**0.091096**±(1.0267E-2)	
UDF4	GDE3-A	0.314340 ±(1.5834E-2)	+	0.842836 ±(1.4427E-2)	+	0.141395 ±(9.9958E-3)	+
	GDE3-B	0.337127 ±(2.6497E-2)	+	0.848353 ±(1.6500E-2)	+	0.182133 ±(1.1223E-2)	+
	GDE3-BR	0.274149 ±(9.4875E-3)	+	0.886287 ±(1.3950E-2)	+	0.131155 ±(5.3718E-3)	+
	Immune-GDE3	**0.022426**±(7.9638E-4)		**0.983015**±(7.8875E-3)		**0.024981**±(7.9183E-3)	
UDF5	GDE3-A	0.060463 ±(6.2296E-3)	+	0.916603 ±(1.1412E-2)	+	0.103387 ±(1.1161E-2)	+
	GDE3-B	0.050155 ±(7.9648E-5)	+	0.895093 ±(9.4002E-3)	+	0.131687 ±(2.3963E-3)	+
	GDE3-BR	0.031787 ±(1.2490E-3)	=	0.928299 ±(1.5916E-2)	+	0.054946 ±(2.7425E-3)	+
	Immune-GDE3	**1.261284**±(2.9028E-2)		**0.868829**±(1.4723E-2)		**0.089574**±(1.0217E-2)	
UDF6	GDE3-A	1.450378 ±(3.8378E-2)	+	0.840490 ±(2.1643E-2)	+	0.166448 ±(1.1814E-2)	+
	GDE3-B	1.483274 ±(4.7225E-2)	+	0.811160 ±(1.5105E-2)	+	0.204857 ±(7.8698E-3)	+
	GDE3-BR	1.355084 ±(2.1906E-2)	+	0.879177 ±(2.9863E-2)	=	0.110381 ±(1.2824E-2)	+
	Immune-GDE3	0.215578 ±(9.3813E-3)		**0.901520**±(1.6464E-2)		**0.117410**±(7.6001E-3)	
UDF7	GDE3-A	0.275805 ±(1.1373E-2)	+	0.841745 ±(1.0698E-2)	+	0.201810 ±(1.4068E-2)	+
	GDE3-B	0.353933 ±(3.0801E-2)	+	0.829512 ±(1.3720E-2)	+	0.214499 ±(1.0275E-2)	+
	GDE3-BR	**0.210719**±(1.4540E-2)	=	0.845727 ±(1.2964E-2)	+	0.170541 ±(5.6473E-3)	+
	Immune-GDE3	**0.399400**±(1.1659E-2)		**0.903051**±(4.4884E-3)		**0.102524**±(1.0831E-2)	
UDF8	GDE3-A	0.527552 ±(2.2211E-2)	+	0.854339 ±(1.1234E-2)	+	0.202146 ±(1.0131E-2)	+
	GDE3-B	0.553863 ±(2.4636E-2)	+	0.843285 ±(1.4723E-2)	+	0.200871 ±(1.1308E-2)	+
	GDE3-BR	0.448665 ±(1.8965E-2)	+	0.861648 ±(1.6859E-2)	+	0.145741 ±(1.1042E-2)	+
	Immune-GDE3	**0.119658**±(5.8976E-3)		**0.924763**±(1.4725E-2)		**0.068341**±(1.3208E-2)	
UDF9	GDE3-A	0.169351 ±(1.1650E-2)	+	0.893263 ±(1.4521E-2)	+	0.202146 ±(1.0131E-2)	+
	GDE3-B	0.181014 ±(1.0395E-2)	+	0.856388 ±(1.6960E-2)	+	0.237909 ±(1.9791E-2)	+
	GDE3-BR	0.153902 ±(9.5809E-3)	+	0.879565 ±(2.0215E-2)	+	0.116522 ±(8.0706E-3)	+
Summary of the statistical test		+	=	+	=	+	=
Immune	GDE3-A	11	1	11	1	12	0
GDE3	GDE3-B	12	0	12	0	12	0
	GDE3-BR	7	5	8	4	11	1

Distribution metrics discussion

For distribution metrics, Immune GDE3 improved its performance significantly. Regarding the spacing metric, Immune GDE3 was better in all test problems when it was compared with GDE3-A and GDE3-B algorithms, and outperformed GDE3-BR in eleven of twelve problems. The statistical test confirms such finding. The results of the binary metric confirms the spacing results (See Table 5.11). C-metric also showed that Immune GDE3 had a better performance than the rest of the algorithms. The statistical results of each algorithm also showed that the Immune version obtained a higher percentage of non-dominated solutions in comparison with the rest of the algorithms and it was more robust over all the executions.

Table 5.11: $\overline{C}$ mean and standard deviation values for all test problems on the third experiment and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune GDE3 outperformed the algorithm in the corresponding row. "−" means that the algorithm in the corresponding row outperformed Immune GDE3. No significant differences between Immune GDE3 and the version in the corresponding row are indicated with "=". The best results are marked in **boldface**

Test Problem	Algorithm comparison	Mean	St. Dev.	
FDA1	Immune-GDE3 vs GDE3-A	**0.994390**	±(2.4998E-3)	+
	GDE3-A vs Immune-GDE3	0.551367	±(5.5330E-3)	
	Immune-GDE3 vs GDE3-B	**0.996380**	±(1.4650E-3)	+
	GDE3-B vs Immune-GDE3	0.395684	±(1.4470E-2)	
	Immune-GDE3 vs GDE3-BR	**0.995921**	±(1.5127E-3)	+
	GDE3-BR vs Immune-GDE3	0.888748	±(1.8555E-2)	
FDA2	Immune-GDE3 vs GDE3-A	**0.999069**	±(6.1445E-5)	+
	GDE3-A vs Immune-GDE3	0.361984	±(1.2091E-2)	
	Immune-GDE3 vs GDE3-B	**0.998672**	±(7.0641E-4)	+
	GDE3-B vs Immune-GDE3	0.253101	±(2.3331E-2)	
	Immune-GDE3 vs GDE3-BR	**0.997380**	±(5.1268E-4)	+
	GDE3-BR vs Immune-GDE3	0.745288	±(7.4976E-3)	
FDA3	Immune-GDE3 vs GDE3-A	**0.932456**	±(2.1754E-2)	+
	GDE3-A vs Immune-GDE3	0.462439	±(2.1084E-2)	
	Immune-GDE3 vs GDE3-B	**0.940666**	±(1.1722E-2)	+
	GDE3-B vs Immune-GDE3	0.339451	±(1.5392E-2)	
	Immune-GDE3 vs GDE3-BR	**0.991728**	±(9.1912E-4)	+
	GDE3-BR vs Immune-GDE3	0.919124	±(1.8452E-2)	
UDF1	Immune-GDE3 vs GDE3-A	**0.897844**	±(8.8375E-3)	+
	GDE3-A vs Immune-GDE3	0.501080	±(9.8546E-3)	
	Immune-GDE3 vs GDE3-B	**0.913676**	±(1.4559E-2)	+
	GDE3-B vs Immune-GDE3	0.499546	±(2.5266E-2)	
	Immune-GDE3 vs GDE3-BR	**0.951048**	±(1.0765E-2)	+
	GDE3-BR vs Immune-GDE3	0.899346	±(1.3246E-2)	
UDF2	Immune-GDE3 vs GDE3-A	**0.919108**	±(1.0620E-2)	+
	GDE3-A vs Immune-GDE3	0.449362	±(1.1093E-2)	
	Immune-GDE3 vs GDE3-B	**0.958826**	±(1.3780E-2)	+
	GDE3-B vs Immune-GDE3	0.412067	±(2.6819E-2)	
	Immune-GDE3 vs GDE3-BR	**0.952785**	±(1.1656E-2)	+
	GDE3-BR vs Immune-GDE3	0.860022	±(1.1328E-2)	
UDF3	Immune-GDE3 vs GDE3-A	**0.981287**	±(4.2563E-3)	+
	GDE3-A vs Immune-GDE3	0.315093	±(1.5295E-2)	
	Immune-GDE3 vs GDE3-B	**0.993651**	±(2.6273E-3)	+
	GDE3-B vs Immune-GDE3	0.300662	±(1.1365E-2)	
	Immune-GDE3 vs GDE3-BR	**0.962160**	±(8.6604E-3)	+
	GDE3-BR vs Immune-GDE3	0.865029	±(1.2886E-2)	
UDF4	Immune-GDE3 vs GDE3-A	**0.878638**	±(1.1841E-2)	+
	GDE3-A vs Immune-GDE3	0.592988	±(1.6434E-2)	
	Immune-GDE3 vs GDE3-B	**0.894711**	±(1.3125E-2)	+
	GDE3-B vs Immune-GDE3	0.515567	±(1.1246E-2)	
	Immune-GDE3 vs GDE3-BR	**0.899259**	±(8.5955E-3)	+
	GDE3-BR vs Immune-GDE3	0.862596	±(1.9793E-2)	
UDF5	Immune-GDE3 vs GDE3-A	**0.937271**	±(7.8036E-3)	+
	GDE3-A vs Immune-GDE3	0.543775	±(1.8013E-2)	
	Immune-GDE3 vs GDE3-B	**0.915532**	±(1.3661E-2)	+
	GDE3-B vs Immune-GDE3	0.537808	±(3.2329E-2)	
	Immune-GDE3 vs GDE3-BR	**0.907041**	±(1.4571E-2)	+
	GDE3-BR vs Immune-GDE3	0.814343	±(1.1252E-2)	
UDF6	Immune-GDE3 vs GDE3-A	**0.975434**	±(7.4323E-3)	+
	GDE3-A vs Immune-GDE3	0.453048	±(9.4770E-3)	
	Immune-GDE3 vs GDE3-B	**0.958279**	±(1.1995E-2)	+

Continued on next page

Table 5.11 *Continued from previous page*

Test Problem	Algorithm comparison	Mean	St. Dev.	
	GDE3-B vs Immune-GDE3	0.417398	±(1.6414E-2)	
	Immune-GDE3 vs GDE3-BR	**0.968249**	±(1.0679E-2)	+
	GDE3-BR vs Immune-GDE3	0.827830	±(9.6023E-3)	
UDF7	Immune-GDE3 vs GDE3-A	**0.990972**	±(2.1847E-3)	+
	GDE3-A vs Immune-GDE3	0.390610	±(1.5843E-2)	
	Immune-GDE3 vs GDE3-B	**0.958845**	±(1.0558E-2)	+
	GDE3-B vs Immune-GDE3	0.408314	±(1.0560E-2)	
	Immune-GDE3 vs GDE3-BR	**0.971797**	±(7.7882E-3)	+
	GDE3-BR vs Immune-GDE3	0.820085	±(1.5704E-2)	
UDF8	Immune-GDE3 vs GDE3-A	**0.907991**	±(4.5679E-3)	+
	GDE3-A vs Immune-GDE3	0.314790	±(1.7426E-2)	
	Immune-GDE3 vs GDE3-B	**0.930182**	±(1.6492E-2)	+
	GDE3-B vs Immune-GDE3	0.253717	±(1.2210E-2)	
	Immune-GDE3 vs GDE3-BR	**0.892362**	±(1.6992E-2)	+
	GDE3-BR vs Immune-GDE3	0.819519	±(9.8680E-3)	
UDF9	Immune-GDE3 vs GDE3-A	**0.929208**	±(2.5293E-2)	+
	GDE3-A vs Immune-GDE3	0.435045	±(2.0088E-2)	
	Immune-GDE3 vs GDE3-B	**0.925090**	±(2.5217E-2)	+
	GDE3-B vs Immune-GDE3	0.461559	±(1.5869E-2)	
	Immune-GDE3 vs GDE3-BR	**0.955251**	±(6.9224E-3)	+
	GDE3-BR vs Immune-GDE3	0.877668	±(8.7088E-3)	

Results of experiment three: Second part

Regarding the second part of experiment three, from experiment one, it was observed that the two dynamic versions of the DNSGA-II algorithm were the less competitive solving the test problems. Therefore, both, DNSGA-II-A and DNSGA-II-B were selected for comparison purposes and the immune response of Immune GDE3 was added to the original NSGA-II algorithm in order to evaluate the role of the immune response in the improvement capability of such algorithm. In Tables 5.12 and 5.13, the results obtained by each metric are presented.

Table 5.12: $\overline{IGD}$, $\overline{HV}$ and $\overline{S}$ mean and standard deviation values for all test problems on the second part of the third experiment and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune DNSGA-II outperformed the algorithm in the corresponding row. "−" means that the algorithm in the corresponding row outperformed Immune DNSGA-II. No significant differences between Immune DNSGA-II and the version in the corresponding row are indicated with "=". The best results are marked in **boldface**

Test Problem		IGD			HV			S		
	Algorithm	Mean	St. Dev.		Mean	St. Dev.		Mean	St. Dev.	
	DNSGA-II-A	0.120843	±(9.1590E-3)	+	0.826732	±(1.3890E-2)	+	0.184589	±(2.8821E-2)	+
FDA1	DNSGA-II-B	0.120395	±(8.1615E-3)	+	0.841526	±(2.0022E-2)	+	0.199440	±(3.6756E-2)	+

Continued on next page

Table 5.12 *Continued from previous page*

Test Problem		IGD		HV		S	
	Immune DNSGA-II	**0.096054**±(4.2226E-3)		**0.875366**±(1.8750E-3)		**0.160803**±(5.3929E-3)	
FDA2	DNSGA-II-A	0.027240 ±(3.1655E-3)	=	0.897290 ±(1.8394E-2)	=	0.091746 ±(2.9774E-2)	=
	DNSGA-II-B	0.026041 ±(3.9815E-3)	=	0.889673 ±(1.9928E-2)	+	0.090280 ±(3.7050E-2)	=
	Immune DNSGA-II	**0.023510**±(7.5902E-3)		**0.902999**±(8.9232E-3)		**0.086640**±(2.4794E-3)	
FDA3	DNSGA-II-A	0.196840 ±(2.6509E-2)	+	0.847181 ±(1.4987E-2)	+	0.259114 ±(1.6295E-2)	+
	DNSGA-II-B	0.182420 ±(2.0071E-2)	=	0.850735 ±(2.0319E-2)	=	0.259560 ±(3.1350E-2)	+
	Immune DNSGA-II	**0.163086**±(6.8267E-3)		**0.864146**±(4.7337E-3)		**0.202834**±(5.7269E-3)	
UDF1	DNSGA-II-A	0.190081 ±(2.5501E-2)	+	0.861455 ±(1.2560E-2)	+	0.133088 ±(1.3565E-2)	+
	DNSGA-II-B	0.176580 ±(2.4053E-2)	=	0.846510 ±(2.0782E-2)	+	0.122636 ±(2.6779E-2)	=
	Immune DNSGA-II	**0.167543**±(5.7238E-3)		**0.880600**±(1.0391E-2)		**0.114682**±(2.7917E-3)	
UDF2	DNSGA-II-A	0.057697 ±(2.6110E-4)	+	0.868180 ±(7.5739E-3)	+	0.136798 ±(1.1456E-2)	+
	DNSGA-II-B	0.055472 ±(1.1405E-3)	+	**0.900560**±(5.4139E-3)	=	0.134370 ±(1.6419E-2)	+
	Immune DNSGA-II	**0.044413**±(2.7835E-3)		0.899551 ±(9.1902E-3)		**0.108414**±(5.6335E-3)	
UDF3	DNSGA-II-A	0.650578 ±(4.4689E-3)	+	0.848921 ±(1.1309E-2)	+	0.232095 ±(1.6955E-2)	+
	DNSGA-II-B	0.619635 ±(3.6366E-3)	+	0.880210 ±(1.8489E-2)	=	0.222228 ±(8.4713E-3)	+
	Immune DNSGA-II	**0.501796**±(6.3162E-3)		**0.886490**±(2.9400E-3)		**0.183815**±(7.3217E-3)	
UDF4	DNSGA-II-A	0.586730 ±(1.4736E-2)	+	0.871692 ±(1.5132E-2)	=	0.232770 ±(8.9446E-3)	+
	DNSGA-II-B	0.559040 ±(2.0006E-2)	+	0.861425 ±(9.9539E-3)	+	0.236598 ±(1.1837E-2)	+
	Immune DNSGA-II	**0.500398**±(8.4876E-3)		**0.894535**±(1.1690E-2)		**0.187338**±(6.6042E-3)	
UDF5	DNSGA-II-A	**0.044145**±(1.8135E-3)	+	0.853131 ±(1.4637E-2)	+	0.120825 ±(5.4924E-3)	=
	DNSGA-II-B	0.048279 ±(1.4589E-3)	+	0.867340 ±(9.8722E-3)	+	0.138510 ±(9.0072E-3)	+
	Immune DNSGA-II	0.027398 ±(2.2582E-3)		**0.890816**±(7.1747E-3)		**0.102585**±(5.6493E-3)	
UDF6	DNSGA-II-A	1.539477 ±(3.9231E-2)	+	0.769060 ±(1.3840E-2)	+	0.199890 ±(9.9754E-3)	+
	DNSGA-II-B	1.614060 ±(5.0917E-2)	+	0.762134 ±(1.6744E-2)	+	0.199513 ±(5.1329E-3)	+
	Immune DNSGA-II	**1.370507**±(2.7627E-2)		**0.833734**±(6.4772E-3)		**0.170140**±(1.7153E-3)	
UDF7	DNSGA-II-A	0.625160 ±(3.5058E-2)	+	0.821780 ±(1.5377E-2)	+	0.228400 ±(1.3060E-2)	+
	DNSGA-II-B	0.680596 ±(3.7617E-2)	+	0.822403 ±(2.2981E-2)	+	0.246178 ±(9.2607E-3)	+
	Immune DNSGA-II	**0.552439**±(1.1320E-2)		**0.845524**±(5.7777E-3)		**0.202311**±(3.1198E-3)	
UDF8	DNSGA-II-A	0.604487 ±(4.5989E-2)	+	0.825920 ±(1.2795E-2)	=	0.223656 ±(1.4291E-2)	=
	DNSGA-II-B	0.623623 ±(2.2379E-2)	+	0.808660 ±(1.6419E-2)	+	0.256730 ±(1.3008E-2)	+
	Immune DNSGA-II	**0.514712**±(1.5054E-2)		**0.843417**±(3.7189E-3)		**0.207686**±(6.3454E-3)	
UDF9	DNSGA-II-A	0.239630 ±(2.2069E-2)	+	0.846251 ±(1.7802E-2)	+	0.254158 ±(1.1683E-2)	+
	DNSGA-II-B	0.237603 ±(2.3084E-2)	+	0.847240 ±(1.2931E-2)	+	0.218070 ±(1.5518E-2)	+
	Immune DNSGA-II	**0.187841**±(6.7872E-3)		**0.863282**±(5.0849E-3)		**0.171116**±(4.1945E-3)	
Summary of the statistical test		+	=	+	=	+	=
Immune DNSGA-II	DNSGA-II-A	11	1	9	3	9	3
	DNSGA-II-B	9	3	9	3	10	2

Table 5.13: $\overline{C}$ mean and standard deviation values for all test problems on the second part of the third experiment and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that Immune DNSGA-II outperformed the algorithm in the corresponding row. "−" means that the algorithm in the corresponding row outperformed Immune DNSGA-II. No significant differences between Immune DNSGA-II and the version in the corresponding row are indicated with "=". The best results are marked in **boldface**

Test Problem	Algorithm comparison	Mean	St. Dev.	
FDA1	Immune DNSGA-II vs DNSGA-II-A	**0.898187**	±(1.0735E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.821892	±(1.0111E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.901603**	±(1.2007E-3)	+
	DNSGA-II-B vs Immune DNSGA-II	0.801082	±(2.9748E-2)	

Continued on next page

Table 5.13 *Continued from previous page*

Test Problem	Algorithm comparison	Mean	St. Dev.	
FDA2	Immune DNSGA-II vs DNSGA-II-A	**0.920360**	±(1.4240E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.830184	±(1.9685E-3)	
	Immune DNSGA-II vs DNSGA-II-B	**0.882176**	±(2.0748E-3)	+
	DNSGA-II-B vs Immune DNSGA-II	0.820716	±(1.9373E-2)	
FDA3	Immune DNSGA-II vs DNSGA-II-A	**0.930168**	±(9.0167E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.840176	±(1.9064E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.891020**	±(2.9619E-2)	+
	DNSGA-II-B vs Immune DNSGA-II	0.850163	±(1.0729E-2)	
UDF1	Immune DNSGA-II vs DNSGA-II-A	**0.830164**	±(5.1974E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.720174	±(3.7468E-3)	
	Immune DNSGA-II vs DNSGA-II-B	**0.810748**	±(2.0174E-2)	=
	DNSGA-II-B vs Immune DNSGA-II	0.780166	±(1.0732E-2)	
UDF2	Immune DNSGA-II vs DNSGA-II-A	**0.801636**	±(1.0620E-2)	=
	DNSGA-II-A vs Immune DNSGA-II	0.781037	±(1.1093E-2)	
	Immune DNSGA-II vs DNSGA-II-B	0.796018	±(1.2603E-3)	=
	DNSGA-II-B vs Immune DNSGA-II	**0.802073**	±(2.7136E-2)	
UDF3	Immune DNSGA-II vs DNSGA-II-A	**0.901917**	±(1.6478E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.691874	±(2.2965E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.879176**	±(1.0539E-3)	+
	DNSGA-II-B vs Immune DNSGA-II	0.701992	±(1.7017E-2)	
UDF4	Immune DNSGA-II vs DNSGA-II-A	**0.850173**	±(3.7174E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.790827	±(6.0178E-3)	
	Immune DNSGA-II vs DNSGA-II-B	**0.878468**	±(1.2897E-2)	+
	DNSGA-II-B vs Immune DNSGA-II	0.801645	±(1.0737E-2)	
UDF5	Immune DNSGA-II vs DNSGA-II-A	**0.800183**	±(7.0354E-3)	=
	DNSGA-II-A vs Immune DNSGA-II	0.770174	±(2.5213E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.830846**	±(1.2986E-2)	=
	DNSGA-II-B vs Immune DNSGA-II	0.793684	±(2.2268E-2)	
UDF6	Immune DNSGA-II vs DNSGA-II-A	**0.930178**	±(2.4736E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.689319	±(4.2678E-3)	
	Immune DNSGA-II vs DNSGA-II-B	**0.940183**	±(1.3023E-3)	+
	DNSGA-II-B vs Immune DNSGA-II	0.599188	±(1.1501E-2)	
UDF7	Immune DNSGA-II vs DNSGA-II-A	**0.801836**	±(2.1031E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.769982	±(1.5114E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.840177**	±(1.0312E-2)	+
	DNSGA-II-B vs Immune DNSGA-II	0.670948	±(1.7693E-2)	
UDF8	Immune DNSGA-II vs DNSGA-II-A	**0.791837**	±(1.9164E-3)	+
	DNSGA-II-A vs Immune DNSGA-II	0.690197	±(1.7163E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.810081**	±(1.1328E-2)	+
	DNSGA-II-B vs Immune DNSGA-II	0.640174	±(1.3096E-2)	
UDF9	Immune DNSGA-II vs DNSGA-II-A	**0.800187**	±(2.1211E-2)	+
	DNSGA-II-A vs Immune DNSGA-II	0.740199	±(2.5253E-2)	
	Immune DNSGA-II vs DNSGA-II-B	**0.790185**	±(7.3950E-2)	+
	DNSGA-II-B vs Immune DNSGA-II	0.680174	±(4.5443E-2)	

From Tables 5.12 and 5.13 it was observed that the new version of the NSGA-II algorithm called Immune DNSGA-II obtained the best results in all performance metrics when it was compared against DNSGA-II-A and DNSGA-II-B algorithms. Regarding IGD metric, it can also be seen that Immune DNSGA-II was better than DNSGA-II-A and DNSGA-II-B in the most test problems. However, Immune DNSGA-II improved its performance when it was compared with DNSGA-II-A.

On the other hand, the statistical results of the HV metric showed that Immune DNSGA-II was the best algorithm in nine of twelve test problems independently of the DNSGA version compared.

As regards to distribution metrics, the obtained results also showed that Immune DNSGA-II outperformed the other algorithms in the most test problems. The results obtained by S metric showed that Immune DNSGA-II obtained a better distribution of solutions in ten of twelve test problems when comparing it with DNSGA-II-B. From Table 5.13 it can be observed that regarding the C-metric, Immune DNSGA-II improved its performance significantly, i.e., it obtained a higher percentage of non-dominated solutions than the other algorithms.

The results obtained by this experiment showed that the proposed immune response has an important role as a change reaction mechanism. For such reason, the DNSGA-II algorithm improves significantly its performance when its original change reaction mechanisms were replaced by the immune response. However, despite the good performance of Immune DNSGA-II, it does not outperform Immune GDE3.

5.4 Summary

This chapter presented a DMOEA called Immune GDE3 which uses the GDE3 algorithm as a MOP optimizer. The change reaction mechanism in Immune GDE3 consists of an immune response based on clonal selection principle which has the main role in Immune GDE3 to track the changing POFs.

Furthermore, in this chapter, the empirical validation of Immune GDE3 was also presented. For this validation, three different experiments were designed. The first one evaluated the performance of Immune GDE3 against other state-of-the-art DMOEAs; the second experiment consisted of a sensitivity analysis, where different combinations of change severities and change frequencies were selected to analyze the performance of Immune GDE3. Finally, the third experiment was designed to evaluate the role of the immune response in Immune GDE3 algorithm to solve DMOPs.

The experiments were carried out on a set of well-known benchmark problems and four performance metrics were used to assess the performance of the algorithms

used in each experiment. Those performance metrics were: Inverted Generational Distance, Hypervolume, Spacing and Two-set-coverage.

The experiments carried out showed that Immune GDE3 is a very competitive algorithm solving DMOPs and this good performance is mainly attributed to its change reaction mechanism based on an immune response.

The next chapter presents the proposal of another DMOEA called DIGDE which is an improved version of Immune GDE3 and it is based on the Inverted Generational Distance metric.

Chapter 6

An improved Immune GDE3 based on the IGD indicator

In the previous chapter, an Immune Generalized Differential Evolution, which is based on the GDE3 algorithm and incorporates an Immune response as change reaction mechanism was presented. The resulting Immune GDE3 was found to be a competitive DMOEA compared with other popular DMOEAs on the adopted test problems.

An important component of MOEAs is their survival selection mechanism, because its main role is to determine the quality of the candidate solutions that are able to survive at each generation. Different survival selection mechanisms have been designed based on different selection criteria, being the most popular the use of Pareto-based selection (mainly through the use of some Pareto ranking scheme [20]).

As was observed in Section 3.5, several performance metrics have been proposed to evaluate and compare the performance of MOEAs. However, in the last few years, the use of performance metrics to guide the search of a MOEA has also been a recent trend in the design of new MOEAs [7, 50]. Even though different works regarding MOEAs based on performance metrics or indicators have been proposed for static multi-objective optimization (see Section 3.3), to the best of the author's knowledge, there is not any DMOEA which uses a performance metric to guide the search in dynamic multi-objective optimization. Therefore, in this chapter, an improved

version of Immune GDE3, namely Distance-based Immune Generalized Differential Evolution (DIGDE), which is based on Inverted Generational Distance is proposed, and its empirical validation is carried out.

6.1 IGD indicator

Inverted Generational Distance (IGD) is one of the most representative performance metrics used in multi-objective optimization to evaluate the quality of solution set obtained by a MOEA regarding convergence and diversity [99]. To compute IGD, it requires a set of uniformly distributed reference points sampled from the Pareto optimal front (POF) as a priori knowledge. Let N be a set of uniformly distributed points in the Pareto optimal front, and d_i is the Euclidean distance between the ith solution member in the Pareto optimal front and its nearest member of an analyzed algorithm (POF*). The IGD is calculated as in Equation 6.1:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{6.1}$$

IGD obtains an average minimum distance from each point in POF to those in POF*, which measures not only convergence but also the diversity of solution set POF* [99]. As it can be seen in the definition of IGD previously described, the IGD value is obtained by computing the mean of the Euclidean distances between elements of the Pareto optimal front (POF) and elements of the POF*. However, in the calculation of IGD, it is often observed that only the solutions of POF* which are closest to at least one solution of POF are considered by the IGD indicator. Therefore, some solutions of POF* could not contribute to the value of this indicator while other solutions can have several contributions. Those solutions that do not have any contribution to the IGD value are usually known as non-contributing solutions. Consequently, a non-contributing solution to IGD indicator can be defined as follows:

Let POF be a set of all solutions that are uniformly sampled on the Pareto optimal front. A solution s* is considered to be non-contributing solution in set POF* for a given POF if the following condition is satisfied:

$$\nexists x \in POF : d(x, s^*) = \min_{s \in POF^*} d(x, s) \tag{6.2}$$

where d is the Euclidean distance between two points. Therefore, the non-contributing solutions are the solutions which are not nearest neighbor of any point in POF.

Since the Pareto optimal front is usually unknown a priori, an approximation must be constructed to be used as the reference set for the IGD calculations. There are several methods available for building the reference set. One of them consists in using external archives (memory), which store a set of non-dominated solutions with the best convergence and diversity found so far [67, 90]. Since Immune GDE3 uses a memory to store the best solutions obtained through the optimization process, such non-dominated solutions were used as the reference set for computing the IGD indicator.

6.2 Distance-based Immune Generalized Differential Evolution

The aim of the new proposed approach consists in designing an improved version of Immune GDE3 which attempts to minimize the IGD indicator in order to obtain a competitive DMOEA that it would be able to provide solutions with good distribution and as close as possible to the Pareto optimal front.

Immune GDE3 [66] is a DMOP with three main components to deal with changes in the environment: (1) a reevaluation solution method as change detection mechanism, (2) an immune response to respond to changes in the environment and (3) a multi-objective optimization algorithm: GDE3 [58] which is an extension of Differential Evolution (DE) to solve MOPs with constraints.

Crowding distance and Non-dominated concepts are also a fundamental part of Immune GDE3 algorithm. The dominance relations are used in the (1+1)-selection mechanism. Crowding distance and Non-dominated Sorting are used as environmental selection mechanism to reduce the population size to its original size.

Taking as reference the original Immune GDE3, the Distance-based Immune

Generalized Differential Evolution algorithm is proposed. The main idea of DIGDE consists of eliminating the candidate solutions that have the least contribution to the IGD metric value in the population by modifying mainly the Survival Selection mechanism of the Immune GDE3 algorithm.

The basic framework of the proposed approach based on IGD metric, denoted as DIGDE is shown in Algorithm 6. DIGDE starts with an initial population P randomly initialized. Then the objective values are computed for all the solutions. In every generation, DIGDE uses the change detection mechanism of Immune GDE3 to detected changes in the environment. If a change is detected, a change reaction mechanism is carried out. Otherwise, the optimization process continues its normal operation as the same way of GDE3 algorithm. After that, the offspring generation is performed to produce a population of offspring (N). Finally, at the end of each generation, the offspring population (N) and the current population (P) are combined, and an environmental selection mechanism is applied on the combined population in order to preserve good solutions for the next generation. The implementation of each component of DIGDE is explained below:

6.2.1 Dynamism handling

The dynamism handling in DIGDE is based on the change detection and change reaction mechanisms used in Immune GDE3. It is worth remarking that the implementation of those components is the same of the original algorithm (See Sections 5.1.1 and 5.1.2).

6.2.2 Offspring generation

The offspring generation of Immune GDE3 and DIDGE are inspired by DE operators (DE/rand/1/bin variant) [88]. For this reason, both generate multiple search directions based on the distribution of solutions (vectors) in the current population. In order to determine which individuals will take part in the offspring generation, a selection mechanism is applied. Selection is also used to determine which individuals will survive for next generations.

In DIGDE, the solutions (vectors) for offspring generation are randomly selected

Algorithm 6 Framework of DIGDE

```
Input: n(number of variables of the problem), G_max(maximum number of generations), NP (population size), CR, F, and initial bounds x̄^(lo), x̄^(hi)
Output: a series of approximated POFs
Set time t=0;
Generate a initial population P := {x_1, ..., x_NP};
Evaluate the initial population;
Copy the non-dominated individuals of P in memory M;
for G= 1 to G_max do
    for i= 1 to NP do
        Implement a ChangeDetectionMechanism() Algorithm 4;
        if Change is detected then
            Set t = t + 1;
            ChangeReactionMechanism() Algorithm 5;
        else
            Optimize the MOP as a static multi-objective evolutionary algorithm;
        end if
        (y):=GenerateOffspring(P);
        N_{i,G} := y;
    end for
    Update memory using N;
    if PopulationSize > NP then
        (P):=SurvirvalSelection(P ∪ N, M); /* here the IGD indicator is used for environmental selection*/
    end if
end for
return P
```

after the immune response (output population from the immune response), while in Immune GDE3 the selection for offspring generation is carried out before the immune response. The idea of changing the place where the offspring generation is performed in the original algorithm emerges because several studies suggest that mating selection from the parent population (P) and memory population (M) improves the offspring generation. Selecting mating parents from P can maintain good population diversity, selecting parents from M can generate fast converge of the population which is desirable in fast-changing environments [52],[60], [92], [111]. As it was aforementioned, DIGDE uses a memory in the change reaction mechanism. Therefore, the current population after the immune response consists of individuals from P and M.

Both, Immune GDE3 and DE, share a (1+1)-selection mechanism, i.e., the best vector between the target and trial is chosen to remain in the population for the next generation. However, since Immune GDE3 was proposed to solve multi-objective problems, it incorporates the dominance relations to select the best vector. On the other hand, in DIGDE, the (1+1)-selection mechanism is replaced by a $(\mu + \lambda)$-selection mechanism, i.e., all the offspring are generated and they are combined with the parent population to form a single population ($P = P \cup N$, where $\mid P \mid = 2NP$). After that, the best μ individuals from the combined population are selected to survive for the next generation according to the environmental selection described in the next section.

Both, offspring generation and selection mechanism described above are presented in Algorithm 7.

6.2.3 Survival selection

The environmental selection procedure (Algorithm 8), starts with the non-dominated sorting, where the combined population P is divided into several non-dominated fronts $F_1, F_2, \ldots,$ according to objective values. Then, a truncation technique is needed to maintain the original population size. To do that, the solutions in the first k fronts are selected, where k is the maximum value satisfying $\mid F_1 \cup F_2 \cup \ldots \cup F_k \mid < NP$. Afterwards, the worst solutions (solutions with least contribution to IGD) are removed one by one from the worst ranked front until the number of solutions

Algorithm 7 *GenerateOffspring(P)*

Input: P (current population), G (current generation), n(number of variables of the problem), CR, F, and initial bounds $\bar{x}^{(lo)}$, $\bar{x}^{(hi)}$
Output: y (offspring solution)
Choose $r_0 \neq r_1 \neq r_2 \neq i$ from current population $\mathbf{P}$;
$\vec{v}_{i,G} = \vec{x}_{r0,G} + F \cdot (\vec{x}_{r1,G} - \vec{x}_{r2,G})$ Mutation operator;
Generate $j_{rand} = radint(1, n)$;
for j= 1 to n **do**
    if j= j_{rand} or rand $(0, 1) < CR$ **then**
        $u_{i,j,G} = v_{i,j,G}$;
    else
        $u_{i,j,G} = x_{i,j,G}$;
    end if
end for
$trial = \vec{u}_{i,j,G}$;
$y = trial$;
return y

reaches the predefined population size, where the solution to be eliminated is denoted as $s* \in F_k$, which satisfies:

$$s* = \underset{s \in F_k}{argmin} \quad IGD(F_k \setminus \{s\}, M) \tag{6.3}$$

Algorithm 8 *SurvivalSelection(P, M)*

Input: P (combined population), M(memory), NP (population size)
Output: P(population for next generation)
$F \leftarrow NondominatedSort(P)$;
$P \leftarrow F_1 \cup F_2 \cup \ldots \cup F_k$, where k is the maximum value such that $\mid F_1 \cup F_2 \cup \ldots \cup F_k \mid < NP$;
while $\mid P \mid > NP$ **do**
    Find $s*$ in F_k by Equation 6.3;
    $F_k \leftarrow F_k \setminus \{s*\}$;
end while
return P

It is worth remarking that the following modifications were done to Immune GDE3 to adapt the algorithm for dynamic multi-objective optimization using the IGD based selection mechanism:

- The IGD contributions must be computed considering the objective function

values of the current population members and the reference set.

- The offspring generation procedure was incorporated after the change reaction mechanism, different from Immune GDE3 where the offspring generation is carried out before the change reaction.

- The (1+1)-selection mechanism was replaced by an $(\mu + \lambda)$-selection mechanism.

- The crowding distance truncation mechanism was replaced by a contribution IGD indicator mechanism.

6.3 Experimental design of DIGDE

The main goal of the DIGDE experimental design consists of an empirical study of its behavior by considering (a) its performance against other well-known dynamic MOEAs including its previous version (Immune GDE3), and (b) its ability to track changes in the environment using different change frequencies. The experimental design followed in the empirical validation of DIGDE was adopted from [52], because the experimental design proposed in such work represented the most recent reference for solving DMOPs.

6.3.1 Test instances

On the experiments carried out, seventeen test problems were selected from the specialized literature to assess the proposed algorithm against other well-known dynamic MOEAs. Different from the selected benchmark functions for the empirical validation of Immune GDE3, for the empirical validation of DIGDE, two FDA test problems namely, FDA4 and FDA5, and dMOP test problems were added to the original set of test instances. Therefore, the new set of test instances include five FDA test problems [33], three dMOP problems [38], and the nine UDF test problems [8]. Table 6.1 presents the main features of the new set of test instances. The definition and details of those problems can be found in [33], [38], [8].

Table 6.1: Summary of main features for the benchmark set used in DIGDE empirical validation.

Test Problem	**Remarks**	**Problem Type**	**No. of Obj.**	**Search Space**	n
FDA1	POS shifts, POF is static	Type I	2	$[0,1] \times [-1,1]^{n-1}$	11
FDA2	POS shifts, POF shape changes	Type II	2	$[0,1] \times [-1,1]^{n-1}$	13
FDA3	POS shifts, POF changes	Type II	2	$[0,1]^2 \times [-1,1]^{n-2}$	10
FDA4	POS shifts, POF is static	Type I	3	$[0,1]^n$	12
FDA5	POS shifts, POF shifts	Type II	3	$[0,1]^n$	12
dMOP1	POS is static, POF shape changes	Type III	2	$[0,1]^n$	10
dMOP2	POS shifts, POF shape changes	Type II	2	$[0,1]^n$	10
dMOP3	POS changes, POF is static	Type I	2	$[0,1]^n$	10
UDF1	POF shifts, POF shifts	Type II	2	$[0,1] \times [-2,2]^{n-1}$	10
UDF2	POS changes, POF shifts	Type II	2	$[0,1] \times [-1,2]^{n-1}$	10
UDF3	POS is static, POF changes	Type III	2	$[0,1] \times [-1,1]^{n-1}$	10
UDF4	POS shifts, POF shifts	Type II	2	$[0,1] \times [-1,1]^{n-1}$	10
UDF5	POS changes, POF changes	Type II	2	$[0,1] \times [-1,2]^{n-1}$	10
UDF6	POS is static, POF changes	Type III	2	$[0,1] \times [-1,1]^{n-1}$	10
UDF7	POS is static, POF changes	Type III	3	$[0,1]^2 \times [-2,2]^{n-2}$	10
UDF8	Several types of dynamic variation of POS and POF	Type II	2	$[0,1] \times [-1,1]^{n-1}$	10
UDF9	Several types of dynamic variation of POS and POF	Type II	2	$[0,1] \times [-2,2]^{n-1}$	10

6.3.2 Performance metrics

In order to analyze the performance of DIGDE regarding convergence, distribution, and diversity, additionally to the performance metrics used in the empirical validation of Immune GDE3 (see Section 5.2.5), the Hypervolume Difference (HVD) and the Maximum Spread (MS) performance metrics were employed. These two metrics were adopted as follows:

- **Hypervolume Difference (HVD)** [52]: This metric measures the gap between the hypervolume of the Pareto front obtained by an algorithm (POF*) and that of the true POF. It is calculated as indicated in Equation 6.4

$$HVD = HV(POF) - HV(POF^*) \tag{6.4}$$

  where HV(S) is the hypervolume of a set S [93]. In this work, the reference point z_{ref} for the computation of hypervolume is $(z+0.5, z_2+0.5, \ldots, z_M+0.5$, where z_j is the maximum value of the jth objective of the true POF and M is the number of objectives. The value of the reference point was adopted from

the experimental design proposed in [52]. For DMOPs, the average of HVD at each time step over a run needs to be computed.

- **Maximum Spread (MS) [37]**: As the same way of HVD, the average of MS obtained results at each time step is computed to report the final result of the metric.

6.3.3 Compared algorithms

In order to assess the performance of DIGDE, the results obtained by DIGDE were compared with respect to those obtained by five different DMOEAs: DNSGA-II-B, DPSO-4, MOEA/D-CER, SGEA and Immune GDE3. Different from the experimental design proposed to evaluate the performance of Immune GDE3, in the experiments carried out for the empirical validation of DIDGE, only DNSGA-II-B was adopted as it showed slightly better performance than DNSGA-II-A in the previously reported experiments (see Section 5.3.1). Furthermore, the MOEA/D-BR algorithm was replaced by another version of dynamic MOEA/D called MOEA/D-CER. Unlike MOEA/D-BR, which considers the nearest solution in the past POS as the parent of the solution concerned. In MOEA/D-CER, the mapping with the past is based on the nearest distance in the objective function space, i.e., based on the relationships in the POF [8].

6.3.4 Parameter settings

The problem and parameter settings of those DMOEAs considered in the experiments designed for the empirical validation of DIGDE were inherited from the referenced papers and were set as follows:

1. Problem parameters: In order to study the impact of change frequency on algorithms' performance, the severity of change n_t is fixed to 10, and the frequency of change τ_t was set to 5, 10 and 20. These parameters were adopted from the experimental design proposed in [52].

2. Population size: The population size for all the test problems was set to 100.

3. Change detection: For all the algorithms, 10% of the population members were randomly selected for change detection mechanism.

4. Number of runs and stopping criterion: Each algorithm was execute 30 independent runs for each test problem. Due to the fact that in DMOPs the term time is related with the generation counter, each algorithm stops after a specified number of generations, which should cover all possible changes. In order to minimize the effect of static optimization, the first change takes place after the first 50 generations as it was proposed in [52]. Therefore, the total number of generations was set to $3n_t\tau_t + 50$, which ensures there are $3n_t$ changes during the evolution process.

5. Algorithm's parameters: The parameter values of each DMOEA adopted for the experiments are the same parameters used for the experiments carried out in the empirical validation of Immune GDE3 (see Table 5.1). The parameters of DIDGDE are the same of those adopted by Immune GDE3.

6.4 Results and Discussion

In the same way of the empirical validation of Immune GDE3, to study the general performance of DIGDE, 30 independent runs were carried out by each one of the compared algorithms. The obtained results in the runs were used to compute the values of five performance metrics (IGD, HVD, S, MS and C-metric).

Tables 6.2, 6.3, 6.4, 6.5 and 6.6 present the obtained average and standard deviation values of IGD, HVD, S, MS and C-metric results for all test problems. For FDA4, FDA5 and UDF7 test problems, DPSO results are not reported, because DPSO algorithm only works with bi-objective problems.

The statistical validation of IGD, HVD, S, and MS results was carried out using the Kruskal-Wallis (KW) test and the Bergmann-Hommels post-hoc test with 95%-confidence. The Wilcoxon rank-sum test was used to indicate significant differences between the C-metric results with 95%-confidence. Figs. 6.1, 6.3, 6.4, 6.5 and 6.6 present a summary of the statistical tests of each metric.

Table 6.2: $\overline{IGD}$ mean and standard deviation values for all test problems on the empirical validation of DIGDE and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that DIGDE outperformed the algorithm in the corresponding column. "−" means that the algorithm in the corresponding column outperformed DIGDE. No significant differences between DIGDE and the algorithm in the corresponding column are indicated with "=". The best results are marked in **boldface**

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
	(5,10)	0.62299 ±(8.6991E-2) +	0.40824 ±(1.8676E-2) +	0.07645 ±(4.1179E-3) +	0.03234 ±(1.4610E-3) +	0.04886 ±(5.4716E-4) +	**0.01604 ±(5.1278E-4)**
FDA1	(10,10)	0.05765 ±(6.6603E-3) +	0.12486 ±(1.3793E-2) +	0.05934 ±(1.0640E-3) +	**0.01320 ±(2.6544E-3) =**	0.02978 ±(1.6619E-4) +	0.01309 ±(2.1780E-3)
	(20,10)	0.04062 ±(4.6663E-3) +	0.02304 ±(3.0631E-3) +	0.01163 ±(6.4228E-4) +	0.00764 ±(1.8690E-3) +	0.01008 ±(1.6202E-4) +	**0.00375 ±(1.1274E-3)**
	(5,10)	0.02557 ±(1.6028E-3) +	0.01934 ±(1.0898E-3) +	0.01647 ±(2.9283E-3) +	0.01453 ±(1.3881E-3) +	0.01153 ±(8.5212E-4) +	**0.00785 ±(2.3451E-4)**
FDA2	(10,10)	0.01040 ±(2.8388E-4) +	0.01067 ±(3.9143E-4) +	0.01054 ±(3.2397E-4) +	0.00856 ±(2.8976E-4) +	0.00753 ±(7.6300E-4) =	**0.00591 ±(5.8512E-4)**
	(20,10)	0.00608 ±(2.5876E-4) +	0.00899 ±(1.0552E-3) +	0.00829 ±(1.0200E-3) +	0.00612 ±(1.1523E-4) +	0.00483 ±(7.3294E-4) +	**0.00281 ±(2.7758E-3**
	(5,10)	0.24766 ±(9.6644E-3) +	0.18963 ±(1.0883E-2) +	0.15040 ±(6.2962E-3) +	0.06115 ±(3.8965E-2) =	0.11206 ±(8.4817E-3) +	**0.05856 ±(1.7228E-3)**
FDA3	(10,10)	0.10375 ±(2.5686E-3) +	0.10625 ±(4.1923E-3) +	0.12275 ±(2.5433E-3) +	0.03952 ±(3.2663E-2) +	0.08293 ±(4.4161E-3) +	**0.01197 ±(3.2057E-3)**
	(20,10)	0.08720 ±(6.8513E-3) +	0.05366 ±(8.5496E-3) +	0.05127 ±(8.1250E-3) +	0.03420 ±(2.6656E-2) +	0.01390 ±(2.4701E-3) +	**0.00984 ±(2.3210E-3)**
	(5,10)	1.48976 ±(1.0483E-2) +	n/a	0.28703 ±(1.0496E-2) +	0.90946 ±(7.1165E-3) +	0.15779 ±(7.5694E-3) =	**0.14851 ±(5.0570E-3)**
FDA4	(10,10)	0.75427 ±(6.6113E-3) +	n/a	0.20601 ±(5.5054E-3) +	0.28161 ±(6.8366E-3) +	**0.13322 ±(3.8142E-3) =**	0.13588 ±(1.8520E-3)
	(20,10)	0.26119 ±(6.5582E-3) +	n/a	0.16867 ±(5.0890E-3) +	0.12335 ±(1.6531E-3) +	0.11829 ±(3.7750E-3) +	**0.08050 ±(2.1205E-3)**
	(5,10)	1.74441 ±(8.4398E-3) +	n/a	0.58579 ±(1.4686E-2) +	0.52055 ±(1.3708E-3) +	0.41388 ±(1.1710E-2) =	**0.40211 ±(3.9802E-3)**
FDA5	(10,10)	1.00648 ±(8.1048E-3) +	n/a	0.38978 ±(1.8828E-2) +	0.36127 ±(1.8125E-3) =	0.28940 ±(5.9896E-3) =	**0.27002 ±(4.5182E-4)**
	(20,10)	0.48839 ±(4.4864E-3) +	n/a	0.29505 ±(1.1296E-2) +	0.30490 ±(2.5630E-3) +	0.29068 ±(8.3887E-4) +	**0.15092 ±(3.0982E-3)**
	(5,10)	0.13595 ±(4.7950E-3) +	0.11402 ±(1.1749E-2) +	0.01199 ±(1.2659E-3) +	0.01109 ±(8.3652E-3) +	0.01255 ±(3.5428E-4) +	**0.00918 ±(1.6203E-5)**
dMOP1	(10,10)	0.00885 ±(2.8045E-4) +	0.00917 ±(5.6396E-4) +	0.00878 ±(5.9896E-4) +	0.00770 ±(2.7687E-3) +	0.00745 ±(2.5972E-4) +	**0.00420 ±(2.0275E-4)**
	(20,10)	0.00747 ±(2.6866E-4) +	0.00792 ±(4.1984E-4) +	0.00698 ±(9.5141E-5) +	0.00630 ±(1.2165E-3) +	0.00608 ±(1.8168E-4) +	**0.00230 ±(1.6582E-4)**
	(5,10)	0.65051 ±(2.3397E-2) +	0.52055 ±(1.0845E-2) +	0.19393 ±(1.3095E-2) +	**0.03003 ±(1.1135E-4) =**	0.03624 ±(1.5298E-3) =	0.03208 ±(2.0302E-4)
dMOP2	(10,10)	0.11449 ±(2.6286E-3) +	0.16905 ±(6.6161E-3) +	0.15374 ±(1.4819E-2) +	0.01212 ±(1.8319E-5) =	0.01252 ±(3.2109E-4) =	**0.01209 ±(3.2051E-5)**
	(20,10)	0.14885 ±(6.0729E-3) +	0.10841 ±(6.9869E-3) +	0.10092 ±(2.3381E-3) +	0.00621 ±(5.4693E-5) =	0.00801 ±(5.0057E-4) +	**0.00596 ±(2.8512E-6)**
	(5,10)	0.56467 ±(9.0425E-3) +	0.51113 ±(1.9323E-2) +	0.14657 ±(7.7593E-3) +	0.17585 ±(3.1699E-2) +	0.10938 ±(6.3584E-3) +	**0.08032 ±(1.3150E-3)**
dMOP3	(10,10)	0.19230 ±(4.7921E-3) +	0.18841 ±(1.4559E-2) +	0.10385 ±(6.2212E-3) +	0.12691 ±(3.8237E-3) +	0.09749 ±(7.1065E-3) +	**0.06021 ±(1.0518E-3)**
	(20,10)	0.10010 ±(5.1016E-3) +	0.15874 ±(1.0211E-2) +	0.07666 ±(4.7863E-3) +	0.08043 ±(5.9849E-3) +	0.06198 ±(2.9097E-3) +	**0.03152 ±(1.2018E-3)**

Continued on next page

Table 6.2 *Continued from previous page*

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
	(5,10)	0.15460 ±(4.3575E-2) +	0.21334 ±(1.4350E-2) +	0.13314 ±(8.4602E-3) +	0.18100 ±(5.3190E-2) +	0.16138 ±(5.8614E-3) +	**0.11094 ±(2.5120E-3)**
UDF1	(10,10)	0.08368 ±(1.9410E-2) =	0.14577 ±(1.1246E-2) +	0.13085 ±(4.9004E-3) +	0.11617 ±(4.4549E-2) +	0.09670 ±(3.3274E-3) +	**0.05031 ±(1.8451E-3)**
	(20,10)	0.07440 ±(6.9930E-3) +	0.11351 ±(9.1603E-3) +	0.09561 ±(5.5863E-3) +	0.08413 ±(7.2941E-2) +	0.07360 ±(2.0273E-3) +	**0.03844 ±(3.4501E-3)**
	(5,10)	0.15000 ±(5.4306E-3) +	0.13814 ±(5.6220E-2) +	0.08889 ±(4.4838E-3) +	0.11856 ±(6.1656E-2) +	0.03484 ±(5.4327E-3) =	**0.02842 ±(2.8410E-3)**
UDF2	(10,10)	0.10655 ±(3.6593E-3) +	0.08759 ±(3.2898E-2) +	0.03269 ±(2.8550E-3) +	0.08580 ±(2.8780E-2) +	0.01128 ±(3.8278E-3) +	**0.00655 ±(1.8025E-3)**
	(20,10)	0.08345 ±(4.6526E-3) +	0.06632 ±(3.0116E-2) +	0.00899 ±(4.1663E-4) +	0.02116 ±(1.5646E-3) +	0.00738 ±(2.3986E-4) +	**0.00408 ±(1.6152E-3)**
	(5,10)	0.68876 ±(5.8259E-3) +	0.66295 ±(8.8046E-3) +	0.45383 ±(7.5594E-3) +	0.68531 ±(4.0420E-3) +	0.39703 ±(1.9040E-2) +	**0.25105 ±(2.0275E-4)**
UDF3	(10,10)	0.59227 ±(9.2397E-3) +	0.59477 ±(7.3205E-3) +	0.40498 ±(1.1930E-2) +	0.60199 ±(2.5568E-3) +	0.31235 ±(2.9010E-2) +	**0.17205 ±(3.7114E-3)**
	(20,10)	0.49255 ±(5.8185E-3) +	0.51317 ±(1.0874E-2) +	0.35424 ±(3.4430E-3) +	0.40909 ±(9.3294E-3) +	0.20125 ±(6.1389E-3) +	**0.10240 ±(4.9520E-3)**
	(5,10)	0.37554 ±(8.8753E-3) +	0.39641 ±(8.2271E-3) +	0.33847 ±(9.4256E-3) +	**0.18623 ±(4.0053E-1) =**	0.20735 ±(5.8972E-3) +	0.19021 ±(2.0813E-3)
UDF4	(10,10)	0.17321 ±(2.2895E-2) +	0.24426 ±(1.2164E-2) +	0.31726 ±(1.7003E-2) +	**0.16318 ±(2.6759E-1) =**	0.17857 ±(5.2238E-3) =	0.18102 ±(2.2351E-3)
	(20,10)	0.13845 ±(1.3822E-2) +	0.18547 ±(8.6882E-3) +	0.15967 ±(1.8004E-2) +	0.11166 ±(4.3169E-2) +	0.11871 ±(5.1822E-3) +	**0.09021 ±(1.0852E-3)**
	(5,10)	0.29810 ±(1.4654E-2) +	0.16703 ±(1.2308E-2) +	0.02938 ±(1.6744E-4) =	0.16328 ±(3.3695E-2) +	0.02067 ±(8.8276E-4) =	**0.01810 ±(5.5102E-3)**
UDF5	(10,10)	0.11303 ±(3.0661E-2) +	0.11281 ±(8.1360E-3) +	0.03028 ±(3.4347E-3) =	0.09873 ±(1.0825E-2) +	0.01007 ±(4.7987E-3) =	**0.00971 ±(2.2141E-4)**
	(20,10)	0.08073 ±(2.6852E-2) +	0.05793 ±(1.4880E-2) +	0.00820 ±(3.8439E-4) +	0.02406 ±(2.9366E-3) +	0.00888 ±(5.3006E-4) +	**0.00598 ±(3.9421E-4)**
	(5,10)	0.60166 ±(1.6185E-1) +	1.36285 ±(8.7259E-3) +	1.22323 ±(1.7900E-2) +	0.74116 ±(2.8448E-1) +	0.40838 ±(1.2777E-2) =	**0.38526 ±(5.0215E-3)**
UDF6	(10,10)	0.45209 ±(3.0209E-2) +	1.25042 ±(1.8869E-2) +	1.18341 ±(8.6106E-3) +	0.68014 ±(3.3005E-1) +	0.29770 ±(4.6476E-2) +	**0.13102 ±(3.2102E-4)**
	(20,10)	0.41426 ±(2.9141E-2) +	1.04189 ±(4.6806E-2) +	1.05263 ±(3.5941E-2) +	0.56374 ±(1.9105E-2) +	0.12874 ±(3.2273E-3) +	**0.08210 ±(5.1017E-4)**
	(5,10)	0.68339 ±(4.4837E-3) +	n/a	0.23849 ±(4.2897E-3) +	0.55324 ±(2.6480E-3) +	0.20193 ±(2.6969E-3) =	**0.18651 ±(1.1402E-3)**
UDF7	(10,10)	0.52067 ±(2.0654E-2) +	n/a	0.21556 ±(7.4823E-3) +	0.50227 ±(2.4764E-2) +	0.17483 ±(9.1391E-3) +	**0.12014 ±(5.0571E-4)**
	(20,10)	0.47155 ±(4.4886E-3) +	n/a	0.12952 ±(4.6331E-3) +	0.41117 ±(1.8852E-2) +	0.11321 ±(2.3259E-3) +	**0.05997 ±(3.8402E-4)**
	(5,10)	0.52833 ±(4.0274E-3) +	0.50751 ±(1.7687E-2) +	0.42104 ±(5.5450E-2) +	0.40346 ±(3.1179E-3) +	0.28754 ±(1.8807E-2) +	**0.15971 ±(3.2485E-3)**
UDF8	(10,10)	0.37625 ±(1.2407E-2) +	0.32423 ±(7.3664E-3) +	0.33553 ±(2.1371E-2) +	0.29624 ±(4.0600E-3) +	0.19793 ±(1.4719E-2) +	**0.09812 ±(5.9120E-3)**
	(20,10)	0.22232 ±(8.3150E-3) +	0.27848 ±(6.4089E-3) +	0.20442 ±(9.3678E-3) +	0.18898 ±(5.7045E-3) +	0.10231 ±(5.0851E-3) +	**0.05989 ±(3.3289E-3)**
	(5,10)	0.47824 ±(6.1106E-3) +	0.41907 ±(1.0846E-2) +	0.16242 ±(4.8309E-3) +	0.41466 ±(3.9856E-3) +	0.11827 ±(4.3334E-3) +	**0.08113 ±(1.0454E-3)**
UDF9	(10,10)	0.38029 ±(1.0272E-2) +	0.29128 ±(7.1188E-3) +	0.14618 ±(5.5383E-3) +	0.24147 ±(2.3278E-3) +	0.09742 ±(5.7192E-3) +	**0.05941 ±(2.0844E-4)**
	(20,10)	0.21582 ±(5.9646E-3) +	0.18407 ±(9.7614E-3) +	0.08398 ±(6.0741E-3) +	0.11090 ±(5.2469E-3) +	0.01240 ±(1.3085E-3) +	**0.00705 ±(2.0748E-4)**

6.4.1 Proximity metrics discussion

Regarding the IGD metric, it can be observed from Table 6.2 that DIGDE obtains the best results in most test problems (thirteen of seventeen test instances) over the three configurations of problem parameters. Furthermore, both SGEA and Immune GDE3 also obtains competitive results. However, the statistical test shows that they do not outperform DIGDE in any test problem. For example, the results showed in Table 6.2 also suggest that SGEA obtains slightly better results than DIGDE on FDA1 and UDF4 test problems in the parameter configuration (10,10). However, the statistical test shows no significant differences between both algorithms.

On the other hand, for all tested instances, DNSGA-II-B, DPSO and MOEA/D-CER fail to show a good performance according to the IGD metric, as indicated by the large IGD values in Table 6.2. From Table 6.2, it can also be observed that all the algorithms improve its performance when increasing the frequency of change parameter value (τ_t). Such behavior could be attributed to the fact that a higher value of the τ_t parameter decreases the difficulty of the problem. Therefore, the algorithms obtain better results when the changes are less frequent, i.e., they have more time to converge to the new POF.

Figure 6.1 shows a summary of the number of test problems where DIGDE outperformed, was similar, or worst than the remaining algorithms according to the significance test applied.

From Fig. 6.1 it can be observed that for the parameter configurations (5,10), DIGDE has a similar behavior than Immune GDE3 in seven of seventeen test problems. On the other hand, by comparing DIGDE performance against SGEA, MOEA/D-CER, DPSO and DNSGA-II, Fig. 6.1 also shows that DIGDE obtains results with significant differences in most test problems when increasing the frequency of change parameter value. According to the results previously discussed, DIGDE, Immune GDE3 and SGEA are the three most competitive algorithms regarding IGD metric. However, DIGDE has a better performance in most test problems and parameter configurations. To analyze another kind of behavior of the three most competitive algorithms, the evolution curves plots of the average IGD values over 30 independent runs for some representative test problems are presented in Fig. 6.2. From such figure, it can be clearly observed that, compared with the other

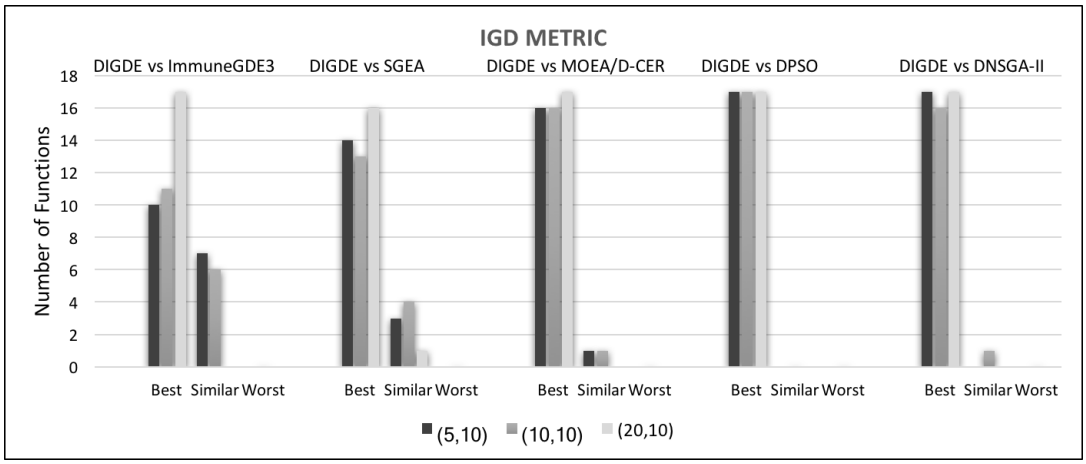

Figure 6.1: IGD metric resume of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test which indicates the number of functions in which DIGDE outperformed, was similar or worst than the remaining algorithms over different configurations of problem parameters

algorithms, DIGDE recovers faster to environmental changes for most of the test problems, thereby obtaining higher convergence performance. On the other hand, DIGDE and Immune GDE3 generate a good population diversity when a change occurs. Therefore, the IGD values obtained by both algorithms fluctuate widely on all test problems and have similar evolution curves on the majority of cases. In contrast, evolution curves of SGEA are more stable than Immune GDE3 and DIGDE evolution curves. However, such behavior is attributed to its poor population diversity, i.e., SGEA presents problems to increase diversity when changes occur.

Finally, based on the IGD metric, the good performance of DIGDE against the rest of the algorithms, can be attributed to the use of IGD metric to guide the search process. Therefore, the IGD indicator helps the algorithm to track the changes as quickly as possible and to obtain good approximations to the POF.

For the Hypervolume results, the behavior of all algorithms was very similar to that observed for the IGD results. From Table 6.3 it was observed that DIGDE is clearly more promising than the other algorithms to solve most of the test problems, but it is outperformed by SGEA and Immune GDE3 on FDA1 for the first two configurations of problem parameters where the changes are more frequent. However, the statistical test shows that it does not exist significant difference between DIGDE

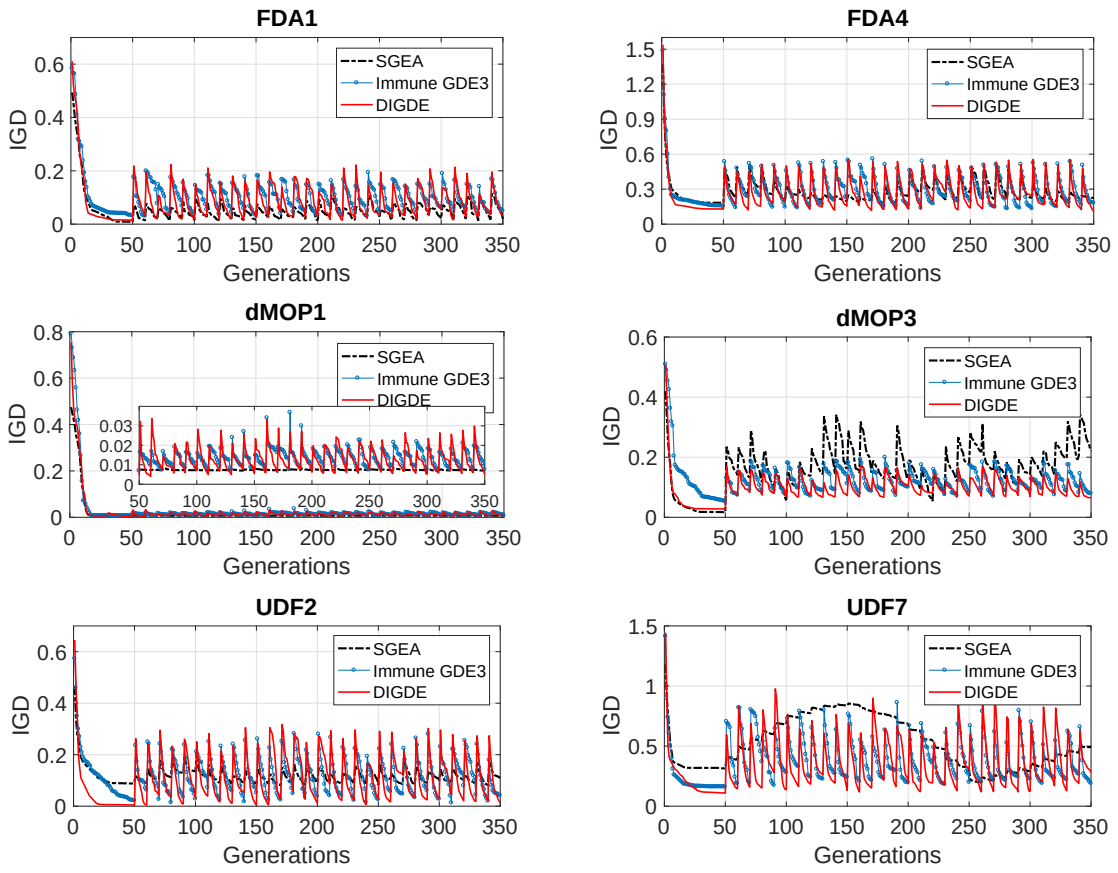

Figure 6.2: Evolution curves of IGD values obtained by the three most competitive algorithms for $\tau_t = 10$ and $n_t = 10$.

and those algorithms. Compared with DNSGA-II, DPSO-4 and MOEA/D-CER algorithms, it was observed that these algorithms have a worse performance than SGEA, Immune GDE3 and DIGDE. The results of the statistical test confirm such finding. On the other hand, from Table 6.3 and the summary of the statistical test presented in Fig. 6.3, it can also be observed that for the three three-objective problems, i.e., FDA4, FDA5, and UDF7, DNSGA-II, MOEA/D-CER and SGEA are most influenced by frequent changes and struggle to push their populations toward the POF, as indicated by their large HVD values. In contrast, Immune GDE3 and DIGDE seem less sensitive to the frequency of change, as it can be seen from their gradual improvement on HVD values. Besides, the HVD values obtained on UDF test problems are significantly higher to those obtained on FDA and dMOP problems,

which imply that the optimization difficulties are increased in UDF problems. Such behavior could be attributed to the fact that UDF problems have nonlinear linkages among decision variables.

Table 6.3: $\overline{HVD}$ mean and standard deviation values for all test problems on the empirical validation of DIGDE and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that DIGDE outperformed the algorithm in the corresponding column. "−" means that the algorithm in the corresponding column outperformed DIGDE. No significant differences between DIGDE and the algorithm in the corresponding column are indicated with "=". The best results are marked in **boldface**

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
FDA1	(5,10)	0.84501 ±(1.3491E-2) +	0.68167 ±(6.3444E-2) +	0.12850 ±(1.6339E-2) +	0.08166 ±(3.9833E-2) =	**0.08023 ±(1.1777E-3) =**	0.09542 ±(1.0545E-3)
	(10,10)	0.13519 ±(6.4432E-3) +	0.45499 ±(4.3669E-2) +	0.08107 ±(3.0662E-3) +	**0.03563 ±(2.0277E-2) =**	0.05471 ±(5.4097E-3) =	0.04874 ±(3.4105E-3)
	(20,10)	0.03655 ±(1.4629E-3) +	0.25473 ±(5.7566E-2) +	0.05716 ±(3.1004E-3) +	0.01970 ±(5.5412E-3) +	0.02623 ±(4.2305E-3) +	**0.01307 ±(1.7749E-4)**
FDA2	(5,10)	0.03985 ±(5.5513E-3) +	0.04430 ±(5.3404E-3) +	0.04026 ±(1.6480E-3) +	0.02283 ±(1.4483E-2) +	0.02140 ±(1.6725E-3) +	**0.01801 ±(1.0545E-3)**
	(10,10)	0.02414 ±(2.5881E-2) +	0.02297 ±(5.6239E-3) +	0.02817 ±(7.8075E-2) +	0.01957 ±(1.3017E-2) +	0.01770 ±(8.7710E-4) +	**0.01395 ±(3.0145E-4)**
	(20,10)	0.01425 ±(1.8521E-2) +	0.01931 ±(2.9070E-3) +	0.01358 ±(1.6095E-3) +	0.01268 ±(9.7391E-3) +	0.01563 ±(9.8963E-4) +	**0.01003 ±(3.5484E-4)**
FDA3	(5,10)	1.36097 ±(1.1758E-1) +	1.69140 ±(7.7064E-2) +	1.39939 ±(1.0070E-1) +	0.98681 ±(7.0134E-2) +	1.12962 ±(1.5212E-2) +	**0.93024 ±(1.0544E-3)**
	(10,10)	1.11535 ±(1.4867E-2) +	1.52929 ±(1.3689E-1) +	1.22007 ±(5.7267E-2) +	0.92823 ±(4.6482E-3) +	0.97615 ±(1.9430E-2) +	**0.86540 ±(2.0842E-2)**
	(20,10)	1.03329 ±(2.8567E-2) +	1.35199 ±(1.1217E-1) +	1.02693 ±(2.4223E-2) +	0.90996 ±(2.5413E-3) +	0.74957 ±(3.0478E-2) +	**0.65041 ±(4.5871E-3)**
FDA4	(5,10)	2.07003 ±(3.9712E-2) +	n/a	0.56069 ±(5.2668E-2) +	1.06390 ±(3.4992E-2) +	1.17116 ±(1.0470E-1) +	**0.98141 ±(2.4395E-3)**
	(10,10)	1.62427 ±(9.4611E-2) +	n/a	0.46432 ±(5.7790E-2) +	0.25303 ±(1.9474E-2) +	0.31336 ±(4.4003E-2) +	**0.21054 ±(2.8741E-3)**
	(20,10)	0.56792 ±(1.5667E-2) +	n/a	0.23121 ±(5.3474E-2) +	0.13812 ±(9.0830E-2) +	0.22190 ±(1.6849E-2) +	**0.10985 ±(3.9852E-3)**
FDA5	(5,10)	6.28445 ±(4.2495E-1) +	n/a	3.68672 ±(3.2655E-1) +	2.37557 ±(3.2905E-1) +	1.80394 ±(3.5505E-1) +	**1.08411 ±(2.7484E-3)**
	(10,10)	5.54654 ±(2.7897E-1) +	n/a	2.68656 ±(3.9240E-1) +	1.85607 ±(6.3785E-2) +	1.29750 ±(3.8586E-1) +	**0.99742 ±(4.8744E-3)**
	(20,10)	2.82941 ±(2.9092E-1) +	n/a	2.11155 ±(3.5352E-1) +	1.77163 ±(4.0812E-2) +	0.96029 ±(9.6431E-2) =	**0.96008 ±(3.9840E-2)**
dMOP1	(5,10)	0.03883 ±(6.1054E-3) +	0.08635 ±(6.2623E-3) +	0.03955 ±(3.6679E-3) +	0.03787 ±(5.1738E-2) +	0.01044 ±(3.5417E-4) +	**0.00958 ±(3.4584E-4)**
	(10,10)	0.02034 ±(1.7943E-2) +	0.07287 ±(7.3035E-3) +	0.02746 ±(6.2344E-3) +	0.01921 ±(1.0101E-2) +	0.00877 ±(3.2144E-4) +	**0.00565 ±(5.5498E-4)**
	(20,10)	0.01544 ±(1.8027E-3) +	0.05847 ±(5.8606E-3) +	0.02001 ±(2.2189E-3) +	0.01714 ±(1.0522E-2) +	0.00773 ±(3.8780E-4) +	**0.00499 ±(3.5899E-4)**
dMOP2	(5,10)	0.78959 ±(3.2258E-2) +	0.66115 ±(3.8445E-2) +	0.35203 ±(3.0077E-2) +	0.07990 ±(5.6383E-3) +	0.06150 ±(4.3416E-3) =	**0.05678 ±(4.5452E-3)**
	(10,10)	0.23510 ±(2.5342E-2) +	0.48250 ±(5.6692E-2) +	0.15120 ±(3.1997E-2) +	0.03576 ±(2.6861E-3) +	0.04781 ±(3.9442E-3) +	**0.02548 ±(2.4741E-4)**
	(20,10)	0.04317 ±(1.4165E-3) +	0.22509 ±(5.7399E-2) +	0.07354 ±(6.5957E-3) +	0.01619 ±(2.3079E-3) +	0.02346 ±(4.3497E-3) +	**0.00988 ±(2.0201E-4)**
dMOP3	(5,10)	0.91019 ±(6.4601E-2) +	0.56147 ±(5.3004E-2) +	0.40887 ±(1.0830E-2) +	0.40250 ±(6.9192E-3) +	0.19288 ±(2.0406E-2) +	**0.11545 ±(1.0548E-3)**
	(10,10)	0.59637 ±(5.6292E-2) +	0.11551 ±(1.3564E-2) +	0.26635 ±(2.1425E-2) +	0.31761 ±(1.1140E-2) +	0.05881 ±(4.5912E-3) +	**0.03878 ±(2.0158E-3)**
	(20,10)	0.24332 ±(3.6738E-2) +	0.08209 ±(2.8641E-3) +	0.22445 ±(2.2699E-2) +	0.23308 ±(1.4247E-2) +	0.02352 ±(6.4592E-3) +	**0.01904 ±(1.7874E-4)**

Continued on next page

Table 6.3 *Continued from previous page*

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
	(5,10)	0.84385 ±(5.2276E-2) +	0.92483 ±(4.0769E-2) +	0.77735 ±(3.7392E-2) +	0.74363 ±(6.3295E-2) +	0.55397 ±(5.1616E-2) +	**0.33054 ±(4.8478E-2)**
UDF1	(10,10)	0.45841 ±(5.6742E-2) +	0.66365 ±(4.2682E-2) +	0.54962 ±(4.0948E-2) +	0.51585 ±(1.2601E-2) +	0.26695 ±(5.5903E-2) +	**0.14254 ±(1.7476E-2)**
	(20,10)	0.09798 ±(7.8166E-3) +	0.40025 ±(5.3530E-2) +	0.29409 ±(4.5910E-2) +	0.22301 ±(2.3275E-2) +	0.21074 ±(3.6905E-3) +	**0.09861 ±(3.4844E-3)**
	(5,10)	0.71502 ±(3.9678E-2) +	0.91918 ±(2.9953E-2) +	0.78333 ±(3.2724E-2) +	0.75987 ±(4.3106E-2) +	0.62851 ±(6.5622E-2) +	**0.41005 ±(1.8711E-2)**
UDF2	(10,10)	0.53789 ±(2.8519E-2) +	0.68361 ±(4.4517E-2) +	0.49757 ±(1.7528E-2) +	0.51116 ±(1.5547E-2) +	0.21661 ±(2.9386E-2) +	**0.13074 ±(2.0875E-2)**
	(20,10)	0.13166 ±(2.0487E-2) +	0.40395 ±(6.1873E-2) +	0.21195 ±(2.7835E-2) +	0.50365 ±(1.8824E-3) +	0.11075 ±(1.4361E-3) +	**0.08118 ±(4.4847E-3)**
	(5,10)	2.16759 ±(2.6143E-1) +	3.24777 ±(2.2964E-1) +	2.21604 ±(3.0451E-1) +	1.32118 ±(1.8744E-2) +	1.01789 ±(6.7713E-2) +	**0.88941 ±(3.5405E-3)**
UDF3	(10,10)	1.16849 ±(9.4736E-2) +	1.50616 ±(2.0688E-1) +	1.60039 ±(1.1286E-1) +	1.22060 ±(4.4827E-3) +	0.86807 ±(7.2021E-3) +	**0.66044 ±(2.0022E-4)**
	(20,10)	0.91810 ±(3.5883E-2) +	1.00623 ±(8.2303E-2) +	1.17634 ±(9.9569E-2) +	0.97296 ±(5.5403E-3) +	0.58988 ±(5.4192E-3) +	**0.51076 ±(3.0108E-4)**
	(5,10)	0.56965 ±(4.5447E-2) +	0.79491 ±(4.7036E-2) +	0.57426 ±(6.4901E-2) +	0.48348 ±(4.4233E-2) +	0.39806 ±(9.7270E-3) =	**0.35251 ±(1.0577E-2)**
UDF4	(10,10)	0.26858 ±(4.3698E-2) +	0.57115 ±(4.9897E-2) +	0.36337 ±(5.4503E-2) +	0.33409 ±(4.8787E-3) +	0.20963 ±(3.1914E-2) =	**0.17820 ±(2.4486E-2)**
	(20,10)	0.12085 ±(1.9966E-2) +	0.28185 ±(5.6443E-2) +	0.19249 ±(4.7808E-2) +	0.10272 ±(7.7944E-3) +	0.12602 ±(1.6371E-3) +	**0.00981 ±(3.0574E-4)**
	(5,10)	0.46902 ±(3.6836E-2) +	0.87303 ±(3.1614E-2) +	0.40309 ±(1.0963E-2) =	0.38902 ±(5.8324E-2) =	0.34622 ±(3.7569E-2) =	**0.33587 ±(2.0587E-2)**
UDF5	(10,10)	0.27641 ±(5.7508E-3) +	0.57507 ±(4.3890E-2) +	0.10633 ±(9.3505E-2) =	0.27213 ±(5.5365E-3) +	0.22833 ±(1.8186E-2) +	**0.12058 ±(1.8548E-4)**
	(20,10)	0.13353 ±(2.0970E-2) +	0.10763 ±(4.6231E-1) +	0.12277 ±(8.5913E-3) +	0.10315 ±(7.2295E-3) +	0.11012 ±(1.3802E-2) +	**0.06764 ±(2.7740E-4)**
	(5,10)	0.89499 ±(4.6813E-2) +	1.35359 ±(1.7317E-1) +	1.25613 ±(1.3474E-1) +	1.11858 ±(5.5303E-1) +	0.73104 ±(1.8902E-2) =	**0.69871 ±(1.5188E-2)**
UDF6	(10,10)	0.83442 ±(6.6427E-2) +	1.02669 ±(5.2800E-2) +	0.94410 ±(3.1888E-2) +	0.97268 ±(4.6700E-2) +	0.56405 ±(5.8290E-2) +	**0.30546 ±(4.0548E-3)**
	(20,10)	0.58433 ±(5.3780E-2) +	0.84811 ±(4.6972E-2) +	0.79957 ±(5.1951E-2) +	0.61478 ±(8.5143E-2) +	0.31093 ±(5.2131E-2) +	**0.19784 ±(2.1849E-3)**
	(5,10)	4.81537 ±(2.5649E-1) +	n/a	3.00628 ±(8.8815E-2) +	3.14319 ±(8.5994E-2) +	2.07210 ±(2.8151E-1) +	**1.25484 ±(2.4841E-2)**
UDF7	(10,10)	2.12170 ±(2.8582E-1) +	n/a	1.59798 ±(2.4480E-1) +	1.97099 ±(1.6973E-1) +	1.04313 ±(6.6382E-2) +	**0.91058 ±(4.3087E-2)**
	(20,10)	1.00742 ±(6.5338E-2) +	n/a	1.19174 ±(9.7018E-2) +	1.35137 ±(2.0331E-1) +	0.78029 ±(2.9494E-2) +	**0.66877 ±(2.0178E-4)**
	(5,10)	2.36301 ±(6.8705E-1) +	4.15852 ±(5.7610E-1) +	1.90123 ±(7.2767E-2) +	2.48997 ±(2.4274E-1) +	1.19731 ±(7.3443E-2) =	**1.18411 ±(1.6595E-2)**
UDF8	(10,10)	1.47363 ±(2.3769E-1) +	3.55950 ±(4.5280E-1) +	1.17061 ±(1.1351E-1) +	1.12205 ±(7.5123E-2) +	0.91702 ±(4.3299E-2) =	**0.90788 ±(2.8474E-2)**
	(20,10)	0.91579 ±(4.0783E-2) +	2.41688 ±(2.7389E-1) +	0.90784 ±(3.9926E-2) +	0.96016 ±(3.1784E-2) +	0.59843 ±(5.4509E-2) +	**0.35078 ±(1.6068E-3)**
	(5,10)	2.17893 ±(3.2908E-1) +	5.46919 ±(3.6295E-1) +	4.13271 ±(8.7226E-2) +	1.95570 ±(6.6399E-2) +	1.21315 ±(1.4875E-1) +	**0.78311 ±(2.0087E-3)**
UDF9	(10,10)	1.74052 ±(1.4769E-1) +	4.67323 ±(7.1881E-1) +	2.82470 ±(4.4732E-1) +	1.41249 ±(1.9753E-1) +	0.93646 ±(3.3535E-2) +	**0.58054 ±(1.0584E-3)**
	(20,10)	1.25540 ±(1.6195E-1) +	2.46699 ±(3.8888E-1) +	1.96801 ±(2.3342E-1) +	1.00533 ±(6.6751E-3) +	0.63905 ±(5.9642E-2) +	**0.35891 ±(2.8978E-4)**

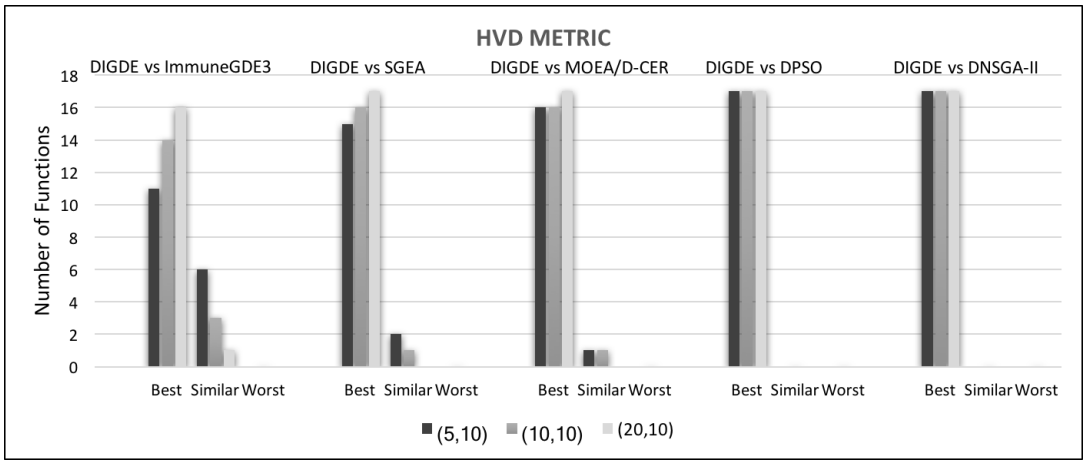

Figure 6.3: HVD metric resume of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test which indicates the number of functions in which DIGDE outperformed, was similar or worst than the remaining algorithms over different configurations of problem parameters

6.4.2 Distribution metrics discussion

To analyze how good the distribution of solutions is over the Pareto Front and to complement the results obtained by the proximity metrics, the results obtained by Spacing (S), Maximum Spread (MS) and Two-Set Coverage (C-metric) metrics are presented in this section.

As regards Spacing metric, it can be seen from Table 6.4 that DIGDE obtains the best results on most of the tested problems. Therefore, it maintains a better distribution of its approximations over environmental changes than the other compared algorithms. In addition, its performance slightly decreases in fast changes (τ_t= 5 and 10) when it is compared with Immune GDE3 for UDF1, UDF4, UDF7 and UDF8 test problems (Type III and Type II problems), and with SGEA for FDA1, FDA5 and UDF9 (Type I and Type II test problems). Compared against SGEA, the statistical test shows that there are not significant differences on such cases. Moreover, the statistical test also confirms that Immune GDE3 outperforms DIGDE on UDF4 and UDF8 test problems for τ_t= 5 (faster changes). Such behavior changes when increasing the value of τ_t parameter favoring the performance of DIGDE.

On the other hand, as regards to DPSO, DIGDE and Immune GDE3 always

obtain better results for all parameter configurations (see Table 6.4 and Fig. 6.4). One possible explanation to the poor performance of DPSO is that DPSO uses a diversity maintenance technique which consists in updating of the memory particles, but if there are a few solutions in the memory, this technique may not be effective to keep a set uniformly distributed solutions. From Table 6.4 it was also observed that the DNSGA-II algorithm presents similar behavior than DIGDE on FDA1 and FDA2 test problems for the first parameter configuration. Furthermore, MOEA/D-CER attempts to maintain competitive Spacing results. As the same way of proximity metrics, from Fig. 6.4 and Table 6.4, it is clearly observed that as the frequency of changes increases, DIGDE improves its performance.

Table 6.4: $\overline{S}$ mean and standard deviation values for all test problems on the empirical validation of DIGDE and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that DIGDE outperformed the algorithm in the corresponding column. "−" means that the algorithm in the corresponding column outperformed DIGDE. No significant differences between DIGDE and the algorithm in the corresponding column are indicated with "=". The best results are marked in **boldface**

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
FDA1	(5,10)	0.02755 ±(1.8963E-3) =	0.06739 ±(2.3935E-2) +	0.05739 ±(4.1316E-3) +	**0.01379 ±(3.1736E-3) =**	0.04883 ±(5.9498E-3) =	0.03650 ±(2.2485E-2)
	(10,10)	0.01247 ±(4.5614E-3) +	0.01970 ±(3.4256E-3) +	0.05712 ±(1.2079E-3) +	0.00698 ±(4.6456E-4) =	0.02979 ±(2.3621E-4) +	**0.00618 ±(3.5415E-3)**
	(20,10)	0.00597 ±(7.6355E-4) +	0.01062 ±(2.0743E-3) +	0.01219 ±(7.5346E-4) +	0.00345 ±(2.5352E-4) =	0.00576 ±(1.6356E-4) +	**0.00312 ±(1.1878E-4)**
FDA2	(5,10)	**0.00731 ±(3.8768E-4) =**	0.01716 ±(3.0737E-2) +	0.01615 ±(2.7805E-3) +	0.00989 ±(1.5454E-3) =	0.01247 ±(8.0401E-4) +	0.00999 ±(2.5441E-4)
	(10,10)	0.00567 ±(4.9892E-4) =	0.01107 ±(1.5072E-2) +	0.01014 ±(3.2100E-3) +	0.00620 ±(6.6254E-4) =	0.00868 ±(4.1849E-4) +	**0.00521 ±(1.3205E-4)**
	(20,10)	0.00477 ±(1.7634E-4) +	0.00998 ±(2.0743E-3) +	0.00967 ±(3.1540E-3) +	0.00417 ±(5.1767E-4) +	0.00601 ±(1.0782E-4) +	**0.00264 ±(1.2309E-4)**
FDA3	(5,10)	0.01670 ±(1.5368E-3) +	0.02835 ±(4.2981E-3) +	0.01901 ±(1.8785E-4) +	0.03416 ±(2.3310E-3) +	**0.01342 ±(9.3056E-4) =**	0.01369 ±(2.1810E-3)
	(10,10)	0.01140 ±(6.1381E-4) +	0.02602 ±(1.8735E-3) +	0.01451 ±(8.6254E-4) +	0.02309 ±(1.7408E-3) +	0.06998 ±(4.5364E-4) +	**0.00859 ±(2.5110E-4)**
	(20,10)	0.00857 ±(3.0101E-4) +	0.01363 ±(1.5336E-3) +	0.01163 ±(6.8985E-4) +	0.01828 ±(3.3515E-3) +	0.00819 ±(1.0016E-3) +	**0.00688 ±(1.1057E-4)**
FDA4	(5,10)	0.12213 ±(9.4296E-3) +	n/a	0.09354 ±(3.8416E-3) =	0.08558 ±(2.5098E-3) =	0.09062 ±(5.3846E-4) =	**0.07857 ±(1.0657E-3)**
	(10,10)	0.08767 ±(4.6642E-3) +	n/a	0.06702 ±(6.6295E-3) +	0.04265 ±(1.5280E-3) +	0.05317 ±(2.3723E-3) +	**0.07857 ±(1.0657E-3)**
	(20,10)	0.04952 ±(4.2408E-3) +	n/a	0.04340 ±(2.1806E-3) +	0.02510 ±(2.9285E-3) +	0.03390 ±(2.2090E-3) +	**0.07857 ±(1.0657E-3)**
FDA5	(5,10)	0.15706 ±(1.4403E-2) +	n/a	0.09333 ±(3.1799E-3) =	**0.08225 ±(1.7034E-3) =**	0.08516 ±(2.0292E-3) =	0.08774 ±(1.4784E-3)
	(10,10)	0.11545 ±(1.0182E-2) +	n/a	0.07075 ±(2.7836E-3) +	**0.04505 ±(3.2457E-3) =**	0.04960 ±(2.1821E-4) =	0.04654 ±(1.5088E-3)
	(20,10)	0.08242 ±(4.5718E-3) +	n/a	0.06052 ±(2.6429E-3) +	0.03469 ±(2.8599E-3) +	0.03075 ±(5.0421E-4) =	**0.03058 ±(1.0548E-4)**
dMOP1	(5,10)	0.00547 ±(2.4398E-4) +	0.00602 ±(2.2013E-4) +	0.00572 ±(4.7078E-4) +	0.00331 ±(1.3029E-4) +	0.00234 ±(1.5437E-4) =	**0.00229 ±(5.0548E-4)**
	(10,10)	0.00537 ±(2.0051E-4) +	0.00521 ±(2.8500E-4) +	0.00445 ±(3.2341E-4) +	0.00248 ±(2.8982E-4) +	0.00201 ±(2.1536E-4) +	**0.00132 ±(1.1085E-4)**
	(20,10)	0.00526 ±(1.6960E-4) +	0.00541 ±(3.1725E-4) +	0.00399 ±(1.6167E-4) +	0.00233 ±(2.2689E-4) +	**0.00100 ±(4.9485E-4) =**	0.00125 ±(1.0578E-3)
dMOP2	(5,10)	0.01545 ±(1.3509E-3) +	0.02352 ±(3.0280E-3) +	0.02887 ±(7.4130E-4) +	0.01293 ±(1.0805E-3) =	0.01410 ±(1.9161E-3) =	**0.01185 ±(1.1878E-3)**
	(10,10)	0.01108 ±(7.3946E-4) +	0.01401 ±(1.4027E-3) +	0.01454 ±(2.4230E-3) +	0.00677 ±(1.2095E-4) +	0.00686 ±(1.4948E-4) +	**0.00408 ±(1.0548E-4)**
	(20,10)	0.00643 ±(3.0659E-4) +	0.01106 ±(5.3917E-4) +	0.00527 ±(3.0305E-3) +	0.00390 ±(5.7782E-4) +	0.00409 ±(2.4693E-4) +	**0.00235 ±(1.2184E-4)**
dMOP3	(5,10)	0.01374 ±(9.7252E-4) +	0.01536 ±(2.7923E-3) +	0.01235 ±(7.8161E-4) +	0.00952 ±(7.4455E-4) +	0.00962 ±(1.5082E-4) +	**0.00745 ±(2.0541E-4)**
	(10,10)	0.00852 ±(2.5296E-4) +	0.01016 ±(5.3762E-4) +	0.00861 ±(2.3506E-4) +	0.00538 ±(1.1194E-4) +	0.00586 ±(9.3539E-5) +	**0.00288 ±(1.2208E-4)**
	(20,10)	0.00541 ±(2.5690E-4) +	0.00668 ±(4.6576E-4) +	0.00585 ±(4.4245E-4) +	0.00441 ±(6.9001E-4) +	0.00412 ±(9.8636E-5) +	**0.00152 ±(1.0548E-4)**

Continued on next page

Table 6.4 *Continued from previous page*

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
UDF1	(5,10)	0.04889 ±(8.9416E-3) +	0.06463 ±(2.9318E-3) +	0.06090 ±(9.8470E-4) +	0.05973 ±(6.0571E-2) +	0.03668 ±(3.3775E-3) =	**0.03397 ±(2.5447E-3)**
	(10,10)	0.02940 ±(9.7576E-3) =	0.04513 ±(3.3422E-3) +	0.04760 ±(4.5930E-3) +	0.05119 ±(6.3689E-2) +	**0.01998 ±(2.3479E-4) =**	0.02695 ±(2.1548E-3)
	(20,10)	0.01465 ±(1.8452E-3) +	0.03554 ±(3.6636E-3) +	0.03324 ±(1.9891E-3) +	0.01248 ±(8.6211E-2) +	0.01630 ±(1.8467E-3) +	**0.00999 ±(3.0574E-5)**
UDF2	(5,10)	0.04002 ±(6.7952E-2) +	0.05007 ±(4.2335E-2) +	0.03393 ±(6.9138E-3) +	0.03333 ±(2.4181E-3) +	0.03128 ±(4.4846E-3) +	**0.01199 ±(2.0548E-3)**
	(10,10)	0.02089 ±(6.6172E-3) +	0.03458 ±(3.2033E-2) +	0.02576 ±(7.3964E-3) +	0.02046 ±(2.9152E-3) +	0.00896 ±(6.9268E-4) +	**0.00587 ±(4.5008E-4)**
	(20,10)	0.01347 ±(1.8196E-3) +	0.02120 ±(4.1479E-2) +	0.01224 ±(1.7602E-3) +	0.01128 ±(6.2437E-4) +	0.00885 ±(3.8997E-4) +	**0.00503 ±(3.0155E-4)**
UDF3	(5,10)	0.00944 ±(2.1823E-4) +	0.01017 ±(1.1405E-3) +	0.01356 ±(1.2008E-3) +	0.00765 ±(2.1464E-4) +	0.00816 ±(4.1999E-4) +	**0.00654 ±(1.2659E-4)**
	(10,10)	0.00714 ±(8.1121E-4) +	0.00948 ±(8.2682E-3) +	0.01065 ±(9.1529E-4) +	0.00629 ±(9.6600E-4) +	0.00644 ±(6.3107E-4) +	**0.00518 ±(3.0548E-5)**
	(20,10)	0.00504 ±(2.5059E-4) +	0.00753 ±(3.5640E-2) +	0.00748 ±(3.5042E-4) +	0.00404 ±(9.5900E-4) +	0.00402 ±(1.1170E-4) +	**0.00265 ±(1.0548E-4)**
UDF4	(5,10)	0.01864 ±(2.2137E-3) +	0.03910 ±(4.7847E-3) +	0.14713 ±(1.0707E-2) +	0.02165 ±(1.0772E-3) +	**0.00683 ±(9.7193E-4) −**	0.00887 ±(1.5487E-3)
	(10,10)	0.01305 ±(9.4972E-2) +	0.02453 ±(2.1176E-3) +	0.00713 ±(6.7877E-4) +	0.01364 ±(3.0041E-4) +	**0.00406 ±(5.9216E-4) =**	0.00428 ±(3.4871E-4)
	(20,10)	0.00882 ±(4.7215E-3) +	0.00910 ±(5.1892E-4) +	0.00626 ±(3.5211E-4) +	0.00879 ±(5.6600E-5) +	0.00338 ±(2.8770E-4) +	**0.00194 ±(2.0548E-4)**
UDF5	(5,10)	0.03153 ±(4.2705E-3) +	0.02815 ±(2.2929E-3) +	0.15174 ±(1.2249E-2) +	0.03099 ±(6.0600E-3) +	0.03003 ±(1.2180E-2) +	**0.00915 ±(2.0254E-3)**
	(10,10)	0.02075 ±(5.5717E-4) +	0.02498 ±(2.5425E-3) +	0.10714 ±(1.0771E-2) +	0.01981 ±(1.7393E-3) +	0.02755 ±(8.4907E-3) +	**0.00765 ±(4.0548E-4)**
	(20,10)	0.01120 ±(1.6597E-3) +	0.02332 ±(2.9693E-3) +	0.09200 ±(2.5504E-3) +	0.01339 ±(6.7297E-3) +	0.02089 ±(1.2765E-3) +	**0.00483 ±(2.1540E-4)**
UDF6	(5,10)	0.08857 ±(8.9144E-2) +	0.07765 ±(3.2466E-3) +	0.11006 ±(1.0999E-2) +	0.09369 ±(6.2125E-3) +	0.05135 ±(1.0217E-2) =	**0.04541 ±(5.2540E-3)**
	(10,10)	0.07267 ±(2.9038E-2) +	0.06976 ±(3.1073E-3) +	0.08949 ±(1.0349E-2) +	0.08416 ±(3.8279E-2) +	0.04393 ±(7.8910E-3) =	**0.02848 ±(4.0547E-3)**
	(20,10)	0.05730 ±(2.7300E-3) +	0.05024 ±(4.8488E-3) +	0.08173 ±(3.5227E-3) +	0.07447 ±(4.9588E-3) +	0.03677 ±(3.0444E-3) =	**0.01465 ±(3.7710E-3)**
UDF7	(5,10)	0.29869 ±(5.4075E-2) +	n/a	0.12711 ±(1.0099E-2) +	0.27661 ±(5.3390E-2) +	**0.07534 ±(7.1539E-3) =**	0.07940 ±(6.1870E-4)
	(10,10)	0.15068 ±(1.8093E-2) +	n/a	0.11876 ±(1.3083E-2) +	0.13130 ±(1.8154E-2) +	0.05784 ±(6.1458E-3) =	**0.05505 ±(2.0187E-3)**
	(20,10)	0.07494 ±(7.4300E-3) +	n/a	0.08473 ±(5.8628E-3) +	0.06203 ±(6.1615E-3) +	0.03569 ±(5.6568E-3) +	**0.02647 ±(4.0484E-5)**
UDF8	(5,10)	0.24634 ±(1.0553E-2) +	0.17152 ±(8.3518E-3) +	0.13300 ±(1.1273E-2) +	0.11195 ±(1.3345E-2) +	**0.06593 ±(8.3796E-3) −**	0.09884 ±(1.0548E-2)
	(10,10)	0.10289 ±(5.8429E-3) +	0.14437 ±(1.1916E-2) +	0.10268 ±(1.2265E-2) +	0.04789 ±(8.5427E-3) +	**0.03603 ±(7.4343E-3) =**	0.04564 ±(4.8784E-3)
	(20,10)	0.09060 ±(9.2148E-3) +	0.14789 ±(1.7902E-2) +	0.07749 ±(1.1176E-2) +	0.04483 ±(4.8709E-3) +	**0.02481 ±(9.5433E-3) =**	0.02878 ±(1.0549E-3)
UDF9	(5,10)	0.15803 ±(6.2072E-3) +	0.16688 ±(5.3527E-3) +	0.09225 ±(1.0811E-2) +	0.07914 ±(1.0811E-2) +	0.05072 ±(1.0241E-2) +	**0.04110 ±(4.1844E-3)**
	(10,10)	0.13332 ±(5.6430E-3) +	0.12769 ±(5.0802E-3) +	0.07385 ±(1.1940E-2) +	**0.02089 ±(6.7396E-3) =**	0.02230 ±(5.4532E-3) =	0.02299 ±(4.0254E-3)
	(20,10)	0.10503 ±(4.3295E-3) +	0.13049 ±(3.7403E-3) +	0.04426 ±(1.2529E-2) +	0.03533 ±(6.6312E-3) +	0.03998 ±(8.5514E-3) +	**0.01608 ±(1.4088E-4)**

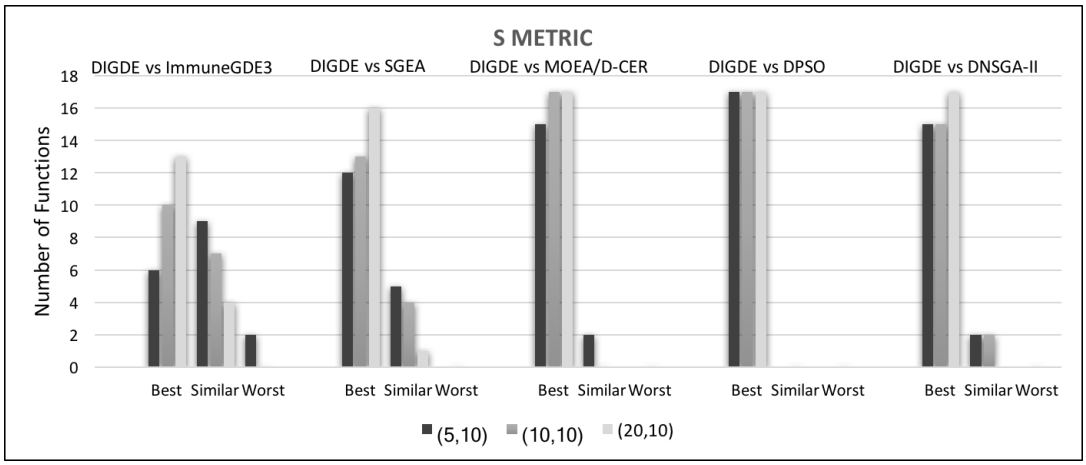

Figure 6.4: S metric resume of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test which indicates the number of functions in which DIGDE outperformed, was similar or worst than the remaining algorithms over different configurations of problem parameters

Regarding the Maximum Spread performance metric, those results of MS were slightly better to those obtained by Spacing metric. For MS performance metric, higher values of MS represent that the obtained algorithm can maintain a good coverage of the POF. From Table 6.5 it was observed that DIGDE obtains better statistical results than the rest of the algorithms in thirteen of seventeen test problems. Furthermore, similar to the other discussed performance metrics, DNSGA-II, DPSO and MOEA/D-CER are the less competitive algorithms in most of the parameters configurations. However, DNSGA-II covers the POF very well for three three-objective problems, i.e., FDA4, FDA5 and UDF7 when the change frequency is higher. This means that the change response mechanisms in DNSGA-II, MOEA/D-CER, and DPSO may face big challenges when dynamisms drastically aggravate population diversity.

In the case of FDA1, FDA4, dMOP2 and UDF7 test problems, the statistical results suggests that Immune GDE3 and SGEA algorithms are better than DIGDE for the first parameter configuration (n_t=10, τ_t=5). However, according to the statistical test there are not significant differences among these algorithms in such parameter configuration.

From Figure 6.5 it can be observed that DIGDE improves its performance when

the frequency of change is higher. That means that the algorithm was able to track the changes even though the presence of faster and slower changes in the environment. The statistical test also shows that DIGDE never presents poor performance with respect to the other algorithms.

Table 6.5: $\overline{MS}$ mean and standard deviation values for all test problems on the empirical validation of DIGDE and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that DIGDE outperformed the algorithm in the corresponding column. "−" means that the algorithm in the corresponding column outperformed DIGDE. No significant differences between DIGDE and the algorithm in the corresponding column are indicated with "=". The best results are marked in **boldface**

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
FDA1	(5,10)	0.62235 ±(8.1682E-3) +	0.40636 ±(2.0300E-2) +	0.77977 ±(4.2911E-2) +	**0.93265 ±(1.3182E-2) =**	0.81968 ±(5.4259E-4) +	0.90345 ±(2.0548E-4)
	(10,10)	0.88345 ±(3.0935E-2) +	0.56704 ±(7.8548E-2) +	0.91288 ±(3.3560E-2) +	**0.97412 ±(9.0135E-2) =**	0.90352 ±(8.3382E-3) +	0.93210 ±(4.6407E-3)
	(20,10)	0.92150 ±(4.8908E-2) +	0.49654 ±(5.8808E-2) +	0.92769 ±(2.2529E-2) +	**0.98634 ±(7.8570E-2) =**	0.95122 ±(1.9659E-3) +	0.97997 ±(3.1054E-3)
FDA2	(5,10)	0.98245 ±(7.8385E-3) +	0.86561 ±(2.6685E-2) +	0.88965 ±(2.5059E-2) +	0.99246 ±(6.6847E-3) =	0.98890 ±(4.8182E-3) +	**0.99301 ±(1.5400E-3)**
	(10,10)	0.98893 ±(5.3380E-3) +	0.87911 ±(2.4367E-2) +	0.93198 ±(2.0936E-2) +	0.99369 ±(6.6135E-3) +	0.99406 ±(3.2695E-3) =	**0.99453 ±(1.1875E-3)**
	(20,10)	0.98995 ±(4.2745E-3) +	0.92369 ±(1.4964E-2) +	0.94881 ±(1.8687E-2) +	0.99270 ±(2.2032E-3) +	0.99419 ±(2.2648E-3) =	**0.99603 ±(5.9812E-4)**
FDA3	(5,10)	0.52421 ±(7.3762E-2) +	0.52082 ±(5.4200E-2) +	0.75452 ±(2.5046E-2) +	0.88604 ±(2.9697E-2) +	0.89326 ±(1.7593E-2) +	**0.93132 ±(2.4084E-3)**
	(10,10)	0.69143 ±(3.7662E-2) +	0.65579 ±(4.1130E-2) +	0.82661 ±(1.2725E-2) +	0.93700 ±(6.0374E-2) +	0.90901 ±(1.0709E-2) +	**0.95942 ±(1.0548E-2)**
	(20,10)	0.74056 ±(2.7535E-2) +	0.69282 ±(4.8043E-2) +	0.85049 ±(2.2670E-2) +	0.94546 ±(4.8849E-2) +	0.93485 ±(1.0385E-2) +	**0.98151 ±(3.6798E-4)**
FDA4	(5,10)	0.99888 ±(6.0068E-4) +	n/a	0.99764 ±(8.0068E-4) +	**0.99813 ±(6.1061E-4) =**	0.99608 ±(2.0936E-3) =	0.99541 ±(2.2154E-3)
	(10,10)	0.99944 ±(3.2961E-4) +	n/a	0.99816 ±(5.9669E-4) +	0.99854 ±(2.7958E-5) +	**0.99955 ±(2.5616E-4) =**	0.99936 ±(1.5410E-5)
	(20,10)	0.99951 ±(2.5079E-4) +	n/a	0.99905 ±(5.6289E-4) +	0.99963 ±(2.6024E-5) +	1.00000 ±(3.0786E-6) +	**1.00000 ±(2.9412E-5)**
FDA5	(5,10)	0.99846 ±(2.9505E-5) +	n/a	0.99441 ±(1.8777E-3) +	0.99293 ±(1.2279E-4) +	0.99494 ±(1.2105E-5) +	**0.99865 ±(3.2913E-5)**
	(10,10)	0.99941 ±(3.3363E-5) +	n/a	0.99700 ±(1.6019E-3) +	0.99611 ±(1.7811E-5) +	0.99864 ±(7.5582E-4) +	**0.99969 ±(2.8213E-6)**
	(20,10)	1.00000 ±(2.4464E-6) =	n/a	0.99834 ±(9.9161E-4) +	0.99792 ±(9.9959E-6) +	0.99881 ±(6.4694E-4) +	**1.00000 ±(1.0586E-6)**
dMOP1	(5,10)	0.85727 ±(3.5494E-3) +	0.83040 ±(1.5738E-2) +	0.96681 ±(1.4024E-2) +	0.95912 ±(1.1620E-2) +	0.98518 ±(2.3625E-3) =	**0.98601 ±(2.0841E-4)**
	(10,10)	0.97019 ±(6.3208E-2) +	0.84936 ±(7.3456E-3) +	0.98364 ±(2.1093E-2) +	0.98606 ±(1.3118E-2) +	0.99143 ±(1.2321E-3) +	**0.99589 ±(2.0545E-5)**
	(20,10)	0.98821 ±(1.8769E-2) +	0.86366 ±(5.9869E-3) +	0.98797 ±(1.1974E-3) +	0.98474 ±(1.4520E-2) +	0.99400 ±(1.5841E-3) +	**0.99899 ±(1.1054E-5)**
dMOP2	(5,10)	0.74626 ±(5.5664E-3) +	0.75496 ±(8.3760E-3) +	0.82508 ±(2.4681E-2) +	0.94210 ±(7.6978E-3) =	**0.94578 ±(5.9814E-3) =**	0.93594 ±(1.0546E-2)
	(10,10)	0.81448 ±(3.5465E-3) +	0.81494 ±(1.0844E-2) +	0.90941 ±(2.1637E-2) +	0.97881 ±(5.4806E-3) +	0.97820 ±(8.2784E-3) +	**0.98108 ±(2.1054E-4)**
	(20,10)	0.89850 ±(6.9047E-3) +	0.81313 ±(5.1935E-2) +	0.93133 ±(2.4084E-2) +	0.99209 ±(2.7370E-3) +	0.98401 ±(6.5218E-3) +	**0.99699 ±(2.2894E-4)**
dMOP3	(5,10)	0.39685 ±(4.2295E-2) +	0.74892 ±(1.9919E-2) +	0.81465 ±(7.3111E-3) +	0.49983 ±(7.4898E-2) +	0.81566 ±(2.6049E-2) +	**0.87439 ±(2.1546E-2)**
	(10,10)	0.45257 ±(3.8602E-2) +	0.80155 ±(1.3986E-2) +	0.85553 ±(2.0715E-2) +	0.59693 ±(1.1787E-2) +	0.90152 ±(8.6113E-3) +	**0.93045 ±(2.0259E-4)**
	(20,10)	0.62611 ±(1.4121E-2) +	0.83104 ±(1.8449E-2) +	0.90897 ±(1.4192E-2) +	0.67818 ±(2.6869E-2) +	0.90474 ±(1.7626E-2) +	**0.93879 ±(1.1546E-4)**

Continued on next page

Table 6.5 *Continued from previous page*

Prob.	(τ_t, n_t)	DNSGA-II-B	DPSO	MOEA/D-CER	SGEA	Immune GDE3	DIGDE
	(5,10)	0.79393 ±(1.6337E-2) +	0.61536 ±(6.1274E-2) +	0.81704 ±(1.9016E-2) +	0.67023 ±(2.4722E-2) +	0.84462 ±(3.5106E-2) =	**0.86108 ±(1.0574E-2)**
UDF1	(10,10)	0.83590 ±(1.0792E-2) +	0.66937 ±(6.2007E-2) +	0.80711 ±(2.6383E-2) +	0.69806 ±(2.8652E-2) +	0.85869 ±(3.2111E-2) +	**0.90484 ±(2.1880E-3)**
	(20,10)	0.89881 ±(1.7422E-2) +	0.78216 ±(2.0173E-2) +	0.91670 ±(1.1187E-2) +	0.79540 ±(2.1829E-2) +	0.91849 ±(3.2497E-2) +	**0.95879 ±(2.1897E-4)**
	(5,10)	0.74683 ±(2.4632E-2) +	0.64485 ±(5.8477E-2) +	0.87781 ±(2.1991E-2) +	0.72738 ±(1.8260E-2) +	0.90291 ±(1.5890E-2) =	**0.91064 ±(1.7544E-2)**
UDF2	(10,10)	0.80924 ±(6.3146E-3) +	0.80539 ±(9.7187E-3) +	0.92649 ±(1.8821E-2) +	0.79293 ±(6.6900E-3) +	0.94084 ±(1.8479E-2) +	**0.95612 ±(1.1581E-4)**
	(20,10)	0.84124 ±(1.4661E-2) +	0.82190 ±(1.5449E-2) +	0.92794 ±(1.8408E-2) +	0.83688 ±(9.6925E-3) +	0.95643 ±(1.2451E-2) +	**0.98598 ±(1.0982E-4)**
	(5,10)	0.37291 ±(2.8626E-2) +	0.20730 ±(5.7842E-2) +	0.63416 ±(3.2051E-2) +	0.41021 ±(1.2273E-2) +	0.78378 ±(1.5806E-2) +	**0.85195 ±(2.2151E-2)**
UDF3	(10,10)	0.46584 ±(5.3240E-3) +	0.24651 ±(4.6478E-2) +	0.66980 ±(2.6487E-2) +	0.46593 ±(5.3254E-3) +	0.79704 ±(1.6639E-2) +	**0.87061 ±(2.5401E-2)**
	(20,10)	0.50506 ±(1.1218E-2) +	0.30948 ±(4.9309E-2) +	0.73899 ±(2.2551E-2) +	0.53159 ±(1.9756E-2) +	0.89658 ±(9.1981E-3) +	**0.93989 ±(1.0084E-3)**
	(5,10)	0.61658 ±(1.9981E-2) +	0.58827 ±(4.2304E-2) +	0.81900 ±(2.7474E-2) +	0.56886 ±(2.4692E-2) +	0.77221 ±(2.3800E-2) +	**0.84498 ±(2.0579E-2)**
UDF4	(10,10)	0.65200 ±(2.1770E-2) +	0.66296 ±(3.2181E-2) +	0.84336 ±(3.0054E-2) +	0.61857 ±(2.0534E-2) +	0.83986 ±(2.6720E-2) +	**0.87988 ±(1.9400E-3)**
	(20,10)	0.66091 ±(1.6987E-2) +	0.69133 ±(2.1379E-2) +	0.86515 ±(3.0523E-2) +	0.69385 ±(2.6381E-2) +	0.86471 ±(2.1077E-2) +	**0.94609 ±(1.1841E-3)**
	(5,10)	0.75138 ±(2.1298E-2) +	0.58685 ±(4.2034E-2) +	0.84246 ±(4.3053E-2) +	0.73118 ±(2.1270E-2) +	0.71618 ±(1.4008E-2) +	**0.78871 ±(1.0570E-2)**
UDF5	(10,10)	0.80994 ±(1.1358E-2) +	0.61713 ±(5.5014E-2) +	0.90545 ±(5.8746E-2) +	0.78490 ±(4.6717E-3) +	0.74628 ±(2.5512E-2) +	**0.87154 ±(1.1058E-2)**
	(20,10)	0.85061 ±(2.0196E-2) +	0.70194 ±(4.8757E-2) +	0.90353 ±(4.3193E-2) +	0.81704 ±(1.2012E-2) +	0.81785 ±(1.0189E-2) +	**0.90548 ±(1.5480E-3)**
	(5,10)	0.18763 ±(1.3201E-2) +	0.17849 ±(2.0224E-2) +	0.19743 ±(3.3622E-2) +	0.12780 ±(7.5062E-2) +	0.32685 ±(4.3271E-2) =	**0.35587 ±(3.9890E-2)**
UDF6	(10,10)	0.23211 ±(2.5401E-2) +	0.20881 ±(2.6391E-2) +	0.22034 ±(2.8674E-2) +	0.15079 ±(8.9601E-2) +	0.37103 ±(2.8265E-2) =	**0.38008 ±(2.1050E-2)**
	(20,10)	0.26837 ±(2.9459E-2) +	0.23772 ±(3.6794E-2) +	0.24566 ±(3.3403E-2) +	0.24016 ±(9.6588E-2) +	0.39802 ±(2.1508E-2) +	**0.48982 ±(1.0181E-3)**
	(5,10)	0.98153 ±(1.0933E-2) =	n/a	0.88904 ±(1.4822E-2) +	0.93293 ±(9.0506E-3) +	**0.98287 ±(2.7069E-3) =**	0.96489 ±(3.1089E-3)
UDF7	(10,10)	0.98889 ±(6.3668E-3) =	n/a	0.94867 ±(1.3823E-2) =	**0.99391 ±(2.1088E-3) =**	0.98872 ±(1.8552E-3) =	0.98915 ±(3.5411E-2)
	(20,10)	0.99134 ±(4.6053E-3) =	n/a	0.95696 ±(1.1728E-2) +	**0.99279 ±(4.4401E-3) =**	0.99092 ±(3.4084E-3) =	0.99261 ±(4.1041E-3)
	(5,10)	0.69442 ±(1.0131E-2) +	0.55508 ±(2.5810E-2) +	0.66389 ±(3.0374E-2) +	0.68295 ±(4.2134E-2) +	0.68783 ±(2.8814E-2) =	**0.70581 ±(2.1098E-2)**
UDF8	(10,10)	0.71819 ±(1.5416E-2) +	0.60013 ±(9.9695E-3) +	0.70820 ±(4.4613E-2) +	0.69564 ±(4.4435E-2) +	0.81433 ±(8.5830E-3) =	**0.81812 ±(3.0548E-2)**
	(20,10)	0.75810 ±(2.8309E-2) +	0.63830 ±(2.6590E-2) +	0.78344 ±(2.7719E-2) +	0.77150 ±(3.1049E-2) +	0.83620 ±(3.0974E-2) +	**0.88789 ±(1.1168E-3)**
	(5,10)	0.76996 ±(3.5852E-2) +	0.68097 ±(3.5161E-2) +	0.76608 ±(3.9985E-2) +	0.85283 ±(1.5126E-2) +	0.84875 ±(3.4017E-2) +	**0.89810 ±(1.0245E-3)**
UDF9	(10,10)	0.79268 ±(3.7433E-2) +	0.75056 ±(4.0319E-2) +	0.81523 ±(4.1850E-2) +	0.87967 ±(2.0748E-2) +	0.90252 ±(1.4780E-2) +	**0.93051 ±(1.0132E-4)**
	(20,10)	0.85210 ±(3.2930E-2) +	0.76542 ±(5.3925E-2) +	0.84384 ±(3.7560E-2) +	0.90619 ±(1.2214E-2) +	0.92455 ±(1.2259E-2) +	**0.97748 ±(2.0587E-3)**

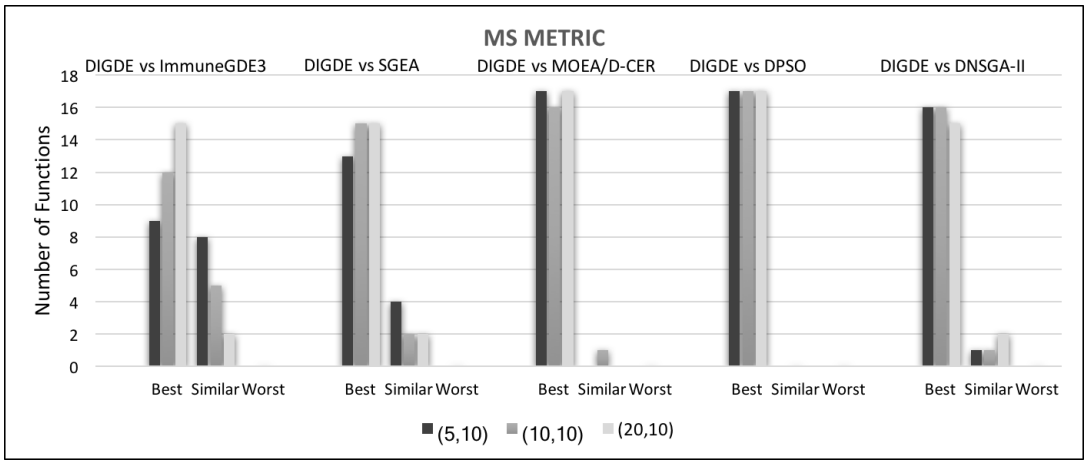

Figure 6.5: MS metric resume of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test which indicates the number of functions in which DIGDE outperformed, was similar or worst than the remaining algorithms over different configurations of problem parameters

Finally, Table 6.6 presents the summary of the Two-Set Coverage metric statistical results and the results of the Wilcoxon test applied to such results are shown in Figure 6.6. Since C-metric consist of all the pairwise combinations of the 30 independent runs executed by each algorithm, the experimental design regarding C-metric tends to grow significantly. For such reason, in order to apply the C-metric, the best three algorithms obtained from the previous experiments were selected for comparison purposes, i.e., the C-metric is computed over the SGEA, Immune GDE3 and DIGDE algorithms.

Table 6.6, shows the achieved results between each comparison in both directions in a specific test problem over three different configurations of problem parameters. The comparison of algorithms that obtains a higher value of non-dominated solutions is marked in boldface and means that the first algorithm in the comparison is considerably better than the second one.

From Table 6.6, is also observed that, when DIDGE is compared with SGEA, DIGDE always obtained better results independently of the configuration of parameters used. However, according to the Wilcoxon test, DIGDE was similar to SGEA in four and three test problems for parameter configurations τ_t=5, n_t=10 and τ_t=10, n_t=10, respectively, i.e., when changes in the environment are more frequent. On

the other hand, in the case of less frequent changes, DIGDE outperformed SGEA in all test problems. The results in Table 6.6 also suggest that DIGDE obtained better results than SGEA in three-objective problems, i.e., FDA4, FDA5 and UDF7. The worst performance of SGEA could be attributed to the fact that SGEA does not increase diversity when changes occur. Therefore, it is vulnerable to the loss diversity.

Regarding the second most competitive algorithm (Immune GDE3), the results of this binary metric showed that, when Immune GDE3 is compared against SGEA, both algorithms present a similar behavior in most test problems especially when the frequency of change is high. Furthermore, it was also observed that in FDA1, FDA3, dMOP2 and UDF4, SGEA was better than Immune GDE3 even though the frequency parameter changes. However, the statistical test showed that both algorithms have similar behavior in such test problems, with exception of FDA3 when the first parameter configuration is used. The statistical validation showed in Fig. 6.6 confirms such findings.

On the other hand, as regards to DIGDE and Immune GDE3 comparison results presented in Table 6.6 and Fig. 6.6 indicate that DIGDE was better than Immune GDE3 in most test problems, and its performance tends to improve when the changes are less frequent. Such behavior means that the proposal of the new selection mechanism implemented in DIGDE helps the algorithm to obtain a better percentage and distribution of non-dominated solutions when it is compared with its previous version (Immune GDE3).

Table 6.6: $\overline{C}$ mean and standard deviation values for all test problems and summary of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test. "+" means that DIGDE outperformed the algorithm in the corresponding row. "−" means that the algorithm in the corresponding row outperformed DIGDE. No significant differences between DIGDE and the algorithm in the corresponding row are indicated with "=". The best results are marked in **boldface**

Prob.	Algorithm comparison	(5,10)		(10,10)		(20,10)	
FDA1	SGEA vs Immune GDE3	**0.991961 ±(1.8405E-3)**		**0.966830 ±(4.3173E-3)**		**0.942406 ±(1.5498E-2)**	
	Immune GDE3 vs SGEA	0.963352 ±(5.6233E-3)	=	0.923064 ±(1.0587E-2)	=	0.909849 ±(2.0458E-2)	=
	SGEA vs DIGDE	0.962868 ±(6.0548E-3)		0.903529 ±(1.0584E-2)		0.839845 ±(1.0299E-2)	
	DIGDE vs SGEA	**0.995036 ±(1.1584E-3)**	=	**0.960084 ±(1.0889E-2)**	+	**0.950990 ±(1.0554E-3)**	+
	Immune GDE3 vs DIGDE	0.988916 ±(2.0584E-3)		0.936264 ±(3.0656E-3)		0.857941 ±(3.0555E-3)	
	DIGDE vs Immune GDE3	**0.995231 ±(1.0871E-3)**	=	**0.990595 ±(4.05929-4)**	+	**0.930481 ±(1.0491E-3)**	+

Continued on next page

Table 6.6 *Continued from previous page*

Prob.	Algorithm comparison	(5,10)		(10,10)		(20,10)	
FDA2	SGEA vs Immune GDE3	0.911139 ±(1.3562E-2)	=	**0.929164 ±(1.3321E-2)**	=	0.854041 ±(1.1724E-2)	=
	Immune GDE3 vs SGEA	**0.926063 ±(1.3546E-2)**		0.903684 ±(2.2085E-2)		**0.875404 ±(1.5865E-2)**	
	SGEA vs DIGDE	0.871180 ±(1.0596E-2)	+	0.852491 ±(1.0895E-2)	+	0.854881 ±(2.0952E-2)	+
	DIGDE vs SGEA	**0.940941 ±(1.0875E-3)**		**0.954847 ±(1.8748E-2)**		**0.988456 ±(1.9580E-2)**	
	Immune GDE3 vs DIGDE	0.868841 ±(2.5451E-2)	+	0.854810 ±(1.8515E-2)	+	0.858410 ±(1.0888E-2)	+
	DIGDE vs Immune GDE3	**0.940544 ±(1.0065E-2)**		**0.931441 ±(1.9781E-2)**		**0.941025 ±(1.5480E-2)**	
FDA3	SGEA vs Immune GDE3	**0.911128 ±(1.9692E-2)**	−	**0.899714 ±(1.6227E-2)**	=	0.785305 ±(1.5710E-2)	=
	Immune GDE3 vs SGEA	0.806247 ±(1.1330E-2)		0.838132 ±(1.9439E-2)		**0.804093 ±(1.6565E-2)**	
	SGEA vs DIGDE	0.905441 ±(1.0545E-2)	=	0.818781 ±(2.0654E-2)	+	0.781109 ±(1.05454-2)	+
	DIGDE vs SGEA	**0.920410 ±(1.5454E-2)**		**0.983012 ±(1.2265E-2)**		**0.920548 ±(1.1514E-2)**	
	Immune GDE3 vs DIGDE	0.851557 ±(1.0548E-2)	+	0.838113 ±(1.0545E-2)	+	0.810989 ±(1.0545E-2)	+
	DIGDE vs Immune GDE3	**0.942065 ±(1.8548E-2)**		**0.949980 ±(1.5460E-2)**		**0.920548 ±(1.9412E-2)**	
FDA4	SGEA vs Immune GDE3	0.780729 ±(1.2128E-2)	+	0.815766 ±(1.8865E-2)	=	0.814193 ±(1.3812E-2)	=
	Immune GDE3 vs SGEA	**0.868239 ±(1.6279E-2)**		**0.832852 ±(1.9172E-2)**		**0.824580 ±(1.5626E-2)**	
	SGEA vs DIGDE	0.770548 ±(2.0548E-2)	+	0.849471 ±(1.0655E-2)	=	0.770684 ±(1.0656E-2)	+
	DIGDE vs SGEA	**0.899811 ±(1.7054E-2)**		**0.854819 ±(2.4984E-2)**		**0.871848 ±(2.1562E-2)**	
	Immune GDE3 vs DIGDE	0.862206 ±(1.1584E-2)	=	0.840321 ±(1.1545E-2)	+	0.834584 ±(1.4806E-2)	+
	DIGDE vs Immune GDE3	**0.880541 ±(4.0574E-3)**		**0.870413 ±(1.2295E-4)**		**0.908970 ±(1.1878E-2)**	
FDA5	SGEA vs Immune GDE3	0.825903 ±(1.3543E-2)	+	0.859008 ±(1.0413E-2)	=	0.857492 ±(1.3103E-2)	=
	Immune GDE3 vs SGEA	**0.917355 ±(1.0916E-2)**		**0.866713 ±(1.0558E-2)**		**0.876694 ±(1.2084E-2)**	
	SGEA vs DIGDE	0.832051 ±(1.2165E-2)	+	0.870106 ±(1.0655E-2)	=	0.875656 ±(5.0656E-3)	+
	DIGDE vs SGEA	**0.906444 ±(1.6561E-2)**		**0.879889 ±(1.1656E-2)**		**0.910656 ±(1.0879E-4)**	
	Immune GDE3 vs DIGDE	**0.906561 ±(2.0065E-2)**	=	0.899899 ±(2.0965E-2)	=	0.905484 ±(1.5454E-2)	=
	DIGDE vs Immune GDE3	0.892015 ±(1.5465E-2)		**0.901008 ±(1.0266E-2)**		**0.930206 ±(1.0656E-3)**	
dMOP1	SGEA vs Immune GDE3	**0.833521 ±(8.1681E-3)**	=	0.835252 ±(1.0633E-2)	=	0.793066 ±(1.5755E-2)	+
	Immune GDE3 vs SGEA	0.814732 ±(1.4239E-2)		**0.852669 ±(1.2314E-2)**		**0.890202 ±(8.5290E-3)**	
	SGEA vs DIGDE	0.784307 ±(1.1878E-2)	+	0.766504 ±(1.0656E-2)	+	0.749261 ±(1.5487E-2)	+
	DIGDE vs SGEA	**0.930454 ±(1.0231E-2)**		**0.926891 ±(2.0545E-2)**		**0.945485 ±(1.3194E-2)**	
	Immune GDE3 vs DIGDE	0.767235 ±(1.5451E-2)	+	0.781478 ±(1.3551E-2)	+	0.794847 ±(1.5546E-3)	+
	DIGDE vs Immune GDE3	**0.945546 ±(2.0548E-3)**		**0.935498 ±(1.0553E-2)**		**0.950547 ±(1.0565E-2)**	
dMOP2	SGEA vs Immune GDE3	**0.854484 ±(1.3136E-2)**	=	**0.844481 ±(1.2954E-2)**	=	**0.842351 ±(1.8005E-2)**	=
	Immune GDE3 vs SGEA	0.811561 ±(1.9774E-2)		0.820661 ±(1.3311E-2)		0.836668 ±(1.8042E-2)	
	SGEA vs DIGDE	0.849854 ±(1.4844E-2)	=	0.804394 ±(1.0985E-2)	+	0.795229 ±(1.2546E-2)	+
	DIGDE vs SGEA	**0.864841 ±(1.2558E-2)**		**0.871556 ±(1.1598E-2)**		**0.877881 ±(1.1995E-2)**	
	Immune GDE3 vs DIGDE	0.828784 ±(1.1862E-2)	+	0.819809 ±(1.2659E-2)	+	0.820331 ±(1.2989E-2)	+
	DIGDE vs Immune GDE3	**0.898944 ±(4.3265E-3)**		**0.880655 ±(1.4056E-2)**		**0.910554 ±(1.0029E-3)**	
dMOP3	SGEA vs Immune GDE3	0.806025 ±(2.1232E-2)	=	0.785708 ±(1.6507E-2)	+	0.855593 ±(1.1188E-2)	=
	Immune GDE3 vs SGEA	**0.832884 ±(1.8329E-2)**		**0.897713 ±(1.6396E-2)**		**0.887740 ±(9.3129E-3)**	
	SGEA vs DIGDE	0.751668 ±(1.5456E-2)	+	0.785216 ±(1.0695E-2)	+	0.758519 ±(1.0669E-2)	+
	DIGDE vs SGEA	**0.940544 ±(1.6564E-2)**		**0.930664 ±(1.4066E-2)**		**0.954584 ±(1.0654E-2)**	
	Immune GDE3 vs DIGDE	0.840228 ±(1.0606E-2)	+	0.870023 ±(1.0955E-2)	+	0.882646 ±(9.0924E-3)	+
	DIGDE vs Immune GDE3	**0.926294 ±(1.5656E-2)**		**0.906565 ±(1.0655E-3)**		**0.932661 ±(1.5651E-2)**	
UDF1	SGEA vs Immune GDE3	0.820529 ±(1.7119E-2)	=	0.781230 ±(1.9107E-2)	+	0.803099 ±(1.8117E-2)	+
	Immune GDE3 vs SGEA	**0.884730 ±(2.1351E-2)**		**0.922978 ±(1.7289E-2)**		**0.893423 ±(2.2094E-2)**	
	SGEA vs DIGDE	0.756012 ±(1.3806E-2)	+	0.746988 ±(1.2164E-2)	+	0.770594 ±(1.4571E-2)	+
	DIGDE vs SGEA	**0.905456 ±(1.0126E-2)**		**0.934845 ±(3.0545E-3)**		**0.954065 ±(1.5060E-2)**	
	Immune GDE3 vs DIGDE	0.801197 ±(1.3206E-2)	+	0.824828 ±(1.7583E-2)	+	0.807849 ±(1.2656E-2)	+
	DIGDE vs Immune GDE3	**0.926561 ±(1.1167E-2)**		**0.903254 ±(6.0548E-3)**		**0.930622 ±(1.0548E-2)**	
UDF2	SGEA vs Immune GDE3	**0.821398 ±(1.4413E-2)**	=	0.769526 ±(1.3568E-2)	+	0.744203 ±(1.4945E-2)	+
	Immune GDE3 vs SGEA	0.773005 ±(2.4369E-2)		**0.934562 ±(1.3639E-2)**		**0.945866 ±(1.2857E-2)**	
	SGEA vs DIGDE	0.737822 ±(1.4442E-2)	+	0.703061 ±(1.8038E-2)	+	0.689897 ±(1.1805E-2)	+
	DIGDE vs SGEA	**0.970548 ±(1.1539E-2)**		**0.954941 ±(6.2025E-3)**		**0.960545 ±(1.0545E-2)**	
	Immune GDE3 vs DIGDE	0.737981 ±(1.0696E-2)	+	0.768770 ±(3.2274E-2)	+	0.871205 ±(1.2544E-2)	+
	DIGDE vs Immune GDE3	**0.978910 ±(1.0498E-2)**		**0.945101 ±(1.0545E-2)**		**0.957130 ±(4.0545E-3)**	

Continued on next page

Table 6.6 *Continued from previous page*

Prob.	Algorithm comparison	(5,10)		(10,10)		(20,10)	
UDF3	SGEA vs Immune GDE3	0.870535 ±(1.9071E-2)	=	0.805049 ±(1.7513E-2)	+	0.829534 ±(2.1164E-2)	+
	Immune GDE3 vs SGEA	**0.882029 ±(9.9941E-3)**		**0.893040 ±(1.2218E-2)**		**0.890002 ±(1.3104E-2)**	
	SGEA vs DIGDE	0.892234 ±(1.5115E-2)	+	0.850664 ±(1.8032E-2)	+	0.787511 ±(1.0546E-2)	+
	DIGDE vs SGEA	**0.933029 ±(2.0544E-3)**		**0.942341 ±(1.0326E-2)**		**0.954810 ±(1.0676E-2)**	
	Immune GDE3 vs DIGDE	0.900254 ±(1.4299E-2)	+	0.909142 ±(1.3624E-2)	+	0.830943 ±(1.0659E-2)	+
	DIGDE vs Immune GDE3	**0.960544 ±(5.0878E-3)**		**0.964841 ±(4.0659E-3)**		**0.950659 ±(2.0645E-2)**	
UDF4	SGEA vs Immune GDE3	**0.914163 ±(2.1304E-2)**	=	**0.935850 ±(1.8729E-2)**	=	**0.927725 ±(1.7138E-2)**	=
	Immune GDE3 vs SGEA	0.852454 ±(2.3445E-2)		0.902755 ±(2.3922E-2)		0.906425 ±(1.6879E-2)	
	SGEA vs DIGDE	**0.928748 ±(2.0545E-2)**	=	0.980446 ±(3.2390E-3)	=	0.855376 ±(1.0988E-2)	+
	DIGDE vs SGEA	0.920541 ±(1.5451E-2)		**0.983107 ±(5.0548E-3)**		**0.951048 ±(3.0089E-3)**	
	Immune GDE3 vs DIGDE	0.854784 ±(1.0659E-2)	+	0.956139 ±(1.1294E-2)	=	0.843758 ±(1.6796E-2)	+
	DIGDE vs Immune GDE3	**0.945414 ±(3.0554E-3)**		**0.970504 ±(6.0987E-3)**		**0.960870 ±(5.0659E-3)**	
UDF5	SGEA vs Immune GDE3	0.681967 ±(1.2640E-2)	+	0.792302 ±(1.9184E-2)	+	0.758903 ±(1.2331E-2)	+
	Immune GDE3 vs SGEA	**0.973408 ±(7.7363E-3)**		**0.890806 ±(1.2482E-2)**		**0.903854 ±(1.7168E-2)**	
	SGEA vs DIGDE	0.658041 ±(1.0544E-2)	+	0.798484 ±(1.4410E-2)	+	0.720499 ±(1.6480E-2)	+
	DIGDE vs SGEA	**0.930645 ±(5.1544E-3)**		**0.940087 ±(2.0545E-2)**		**0.918910 ±(3.1854E-2)**	
	Immune GDE3 vs DIGDE	0.899340 ±(1.3800E-2)	=	0.871759 ±(1.4862E-2)	=	0.867397 ±(1.7080E-2)	=
	DIGDE vs Immune GDE3	**0.906596 ±(2.0545E-3)**		**0.890548 ±(1.0548E-2)**		**0.927816 ±(2.0848E-2)**	
UDF6	SGEA vs Immune GDE3	0.760913 ±(1.3777E-2)	=	0.813123 ±(2.0093E-2)	+	0.824552 ±(1.7247E-2)	+
	Immune GDE3 vs SGEA	**0.827129 ±(1.7264E-2)**		**0.940342 ±(2.1557E-2)**		**0.904467 ±(1.8844E-2)**	
	SGEA vs DIGDE	0.741268 ±(2.1314E-2)	+	0.735918 ±(2.3001E-2)	+	0.612049 ±(2.1975E-2)	+
	DIGDE vs SGEA	**0.840178 ±(3.0548E-2)**		**0.930545 ±(2.0545E-2)**		**0.920584 ±(1.0878E-2)**	
	Immune GDE3 vs DIGDE	0.894439 ±(2.3577E-2)	=	0.884625 ±(1.2903E-2)	=	0.780223 ±(2.2124E-2)	+
	DIGDE vs Immune GDE3	**0.950205 ±(1.1221E-2)**		**0.920057 ±(1.0548E-2)**		**0.930548 ±(1.8440E-2)**	
UDF7	SGEA vs Immune GDE3	0.745583 ±(1.7843E-2)	=	0.720026 ±(2.0271E-2)	+	0.722268 ±(2.8751E-2)	+
	Immune GDE3 vs SGEA	**0.791355 ±(2.3062E-2)**		**0.826797 ±(1.6951E-2)**		**0.825642 ±(1.8525E-2)**	
	SGEA vs DIGDE	0.712375 ±(2.1054E-2)	+	0.697157 ±(1.9814E-2)	+	0.623317 ±(1.9839E-2)	+
	DIGDE vs SGEA	**0.890548 ±(2.1548E-2)**		**0.959015 ±(1.0148E-2)**		**0.938740 ±(6.1018E-3)**	
	Immune GDE3 vs DIGDE	0.879241 ±(1.1848E-2)	=	0.930548 ±(1.0548E-2)	=	0.781162 ±(1.7043E-2)	+
	DIGDE vs Immune GDE3	**0.900889 ±(1.0548E-2)**		**0.945878 ±(1.8487E-2)**		**0.949898 ±(4.2058E-3)**	
UDF8	SGEA vs Immune GDE3	0.896725 ±(1.3100E-2)	=	0.886523 ±(1.9222E-2)	=	0.838791 ±(1.4662E-2)	=
	Immune GDE3 vs SGEA	**0.913814 ±(1.4306E-2)**		**0.904824 ±(1.4333E-2)**		**0.867685 ±(9.6073E-3)**	
	SGEA vs DIGDE	0.855952 ±(1.4514E-2)	+	0.800927 ±(5.6444E-4)	+	0.688988 ±(9.5314E-2)	+
	DIGDE vs SGEA	**0.957889 ±(1.0659E-2)**		**0.970656 ±(1.0548E-2)**		**0.950848 ±(1.5361E-3)**	
	Immune GDE3 vs DIGDE	0.819285 ±(1.8841E-2)	+	0.835824 ±(2.6897E-2)	+	0.739925 ±(1.0548E-3)	+
	DIGDE vs Immune GDE3	**0.963615 ±(4.0874E-3)**		**0.950659 ±(1.4198E-3)**		**0.955490 ±(2.1780E-3)**	
UDF9	SGEA vs Immune GDE3	0.768925 ±(1.5395E-2)	+	0.693631 ±(1.9334E-2)	+	0.713509 ±(1.8556E-2)	+
	Immune GDE3 vs SGEA	**0.871057 ±(8.1720E-3)**		**0.923946 ±(1.8693E-2)**		**0.900308 ±(1.8127E-2)**	
	SGEA vs DIGDE	0.619072 ±(2.1448E-2)	+	0.644666 ±(2.3659E-2)	+	0.574661 ±(2.0209E-2)	+
	DIGDE vs SGEA	**0.970874 ±(1.5487E-2)**		**0.949898 ±(1.0484E-2)**		**0.960848 ±(1.5592E-2)**	
	Immune GDE3 vs DIGDE	0.734300 ±(3.7484E-2)	+	0.812655 ±(5.0878E-3)	+	0.754333 ±(1.0659E-2)	+
	DIGDE vs Immune GDE3	**0.920781 ±(1.0584E-2)**		**0.900891 ±(1.7595E-3)**		**0.923055 ±(1.0249E-3)**	

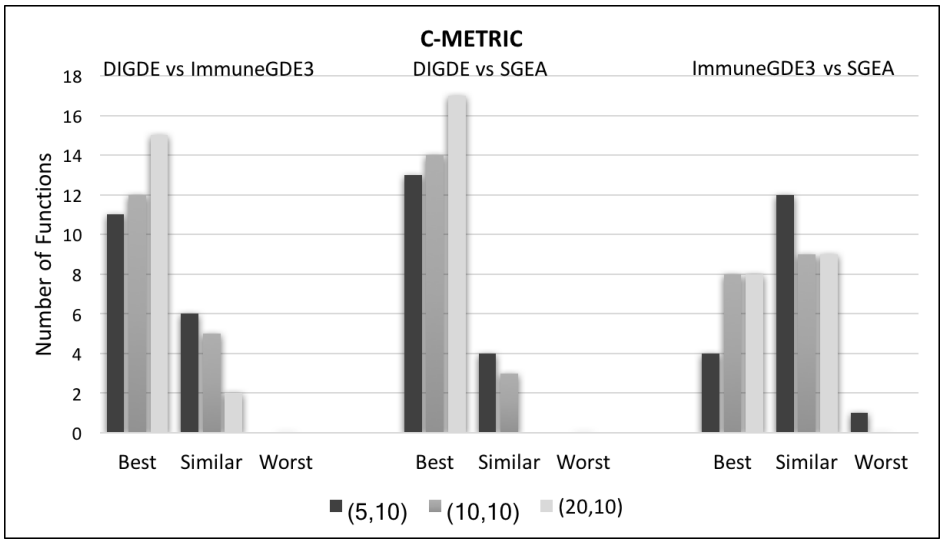

Figure 6.6: C-metric resume of Kruskal-Wallis test and the Bergmann-Hommels post-hoc test which indicates the number of functions in which DIGDE outperformed, was similar or worst than the remaining algorithms over different configurations of problem parameters

6.4.3 Further discussions

The previous experimental comparison and analysis have shown that DIGDE is capable of solving different types of DMOPs. Especially, DIGDE works very well on DMOPs without strong variable linkages, like most of the FDA and dMOP test problems. However, experimental comparisons on the UDF test problems, which have strong variable-linkage, show that DIGDE obtains better results than the rest of the algorithms. DIGDE also obtains competitive results even though the change frequency parameter changes. Moreover, the variation of the frequency parameter does not cause diversity loss in DIGDE. Therefore, DIGDE can track the moving POSs and POFs effectively. On the other hand, from the obtained results, it can be observed that DIGDE was better than its predecessor (Immune GDE3). That means that the incorporation of IGD indicator in the selection mechanism of DIGDE improves their performance significantly. The IGD metric helps the algorithm to obtain solutions closer to the POF, with a better distribution over the POF. The results obtained by each metric of the experimental design confirm such findings.

From the results previously presented, it can be seen that DIGDE, Immune GDE3

and SGEA were the best algorithms solving the DMOPs selected for the experimental design. For a graphical view of algorithms' tracking ability over changing environments, POFs obtained by the aforementioned algorithms over six representative test problems are presented in Fig. 6.8,

Each problem selected represents a different type of DMOP, i.e., type I, type II and type III test problems. Fig. 6.8 shows that DIGDE is very capable of tracking environmental changes, obtaining solutions with good distribution over the POF. Furthermore, from Fig. 6.8 it can be also observed that Immune GDE3 presents a competitive performance in the new selected test problems. The algorithms' tracking ability confirms the obtained results by proximity and distribution metrics.

6.5 Algorithm complexity

All the experiments of this thesis were executed on a PC with Intel Core i7-2635 (2.0GHz) CPU and 8 GB RAM using macOS Sierra 10.12.5 OS. The algorithm programs of DNSGA-II versions, DPSO-4, MOEA/D-BR, Immune GDE3 and DIGDE, were implemented and executed in MATLAB R2016b. The SGEA program was executed using C++ because the original program from the authors was used for the experiments. The computational complexity of the compared algorithms was calculated based on the method proposed in the technical report of the CEC 2017 competition [3]. Such method has been proposed to evaluate and compare the complexity of algorithms in recognized competitions since 2005, independently of the platform in which they had been implemented. More specifically, $T0$ is the execution time of running one million evaluations of a basic mathematical expression. $T1$ is the time to execute 200,000 evaluations of function UDF1. To compute $T1$, function UDF1 was selected because it represents the features of most of the test problems. Finally, $T2$ is the mean time to execute UDF1 function with each algorithm over five independent runs. A summary of the obtained results is shown in Table 6.7. From such table, it was observed that the computational time is high for most of the algorithms, including Immune GDE3 and DIGDE. The computational time of DIGDE is slightly higher than the computational time of Immune GDE3. This increment in the computational time is mainly attributed to the incorporation of the IGD metric

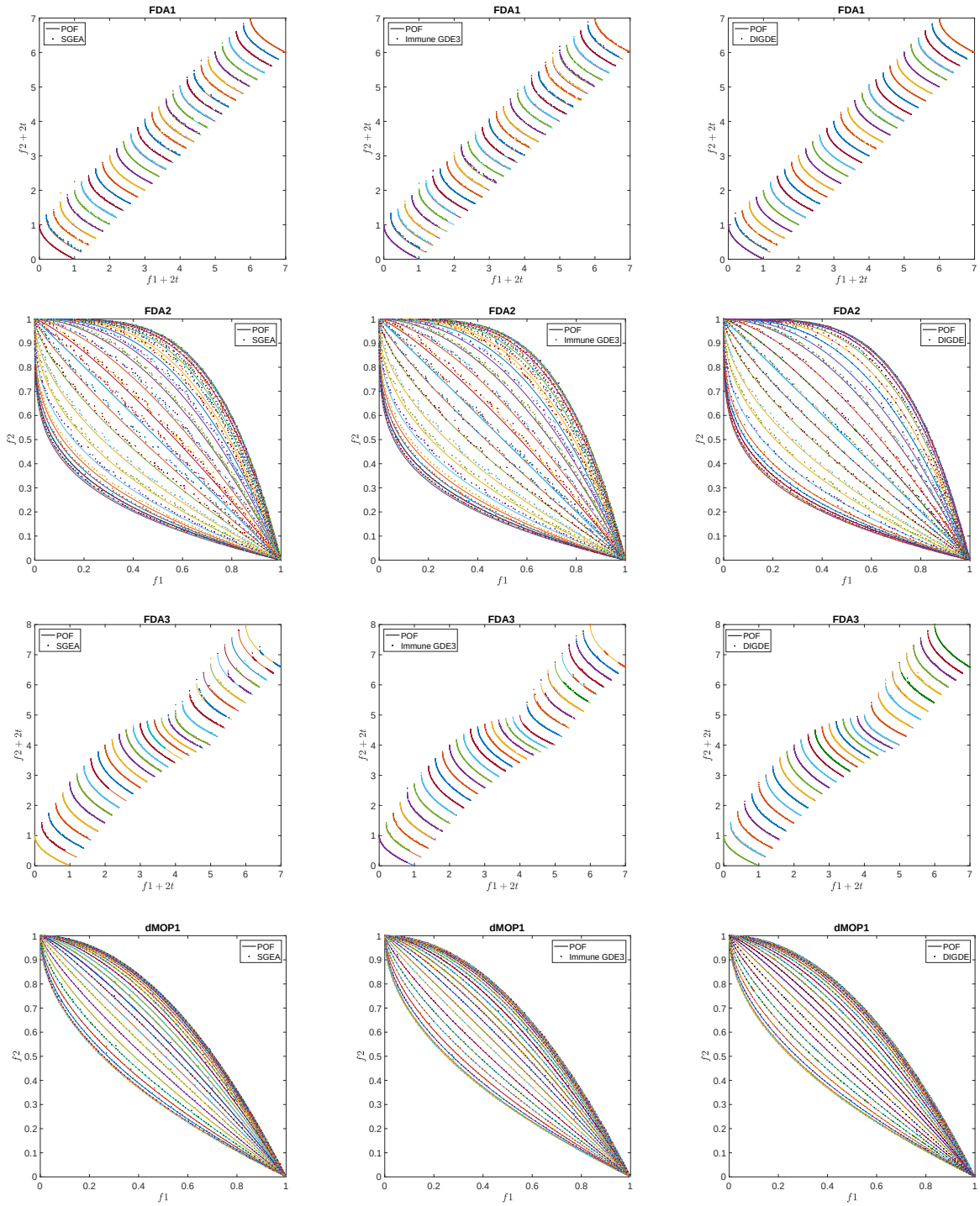

Figure 6.7: Obtained POFs by the three best algorithms on different representative test problems with $n_t = 10$ and $\tau_t = 10$.

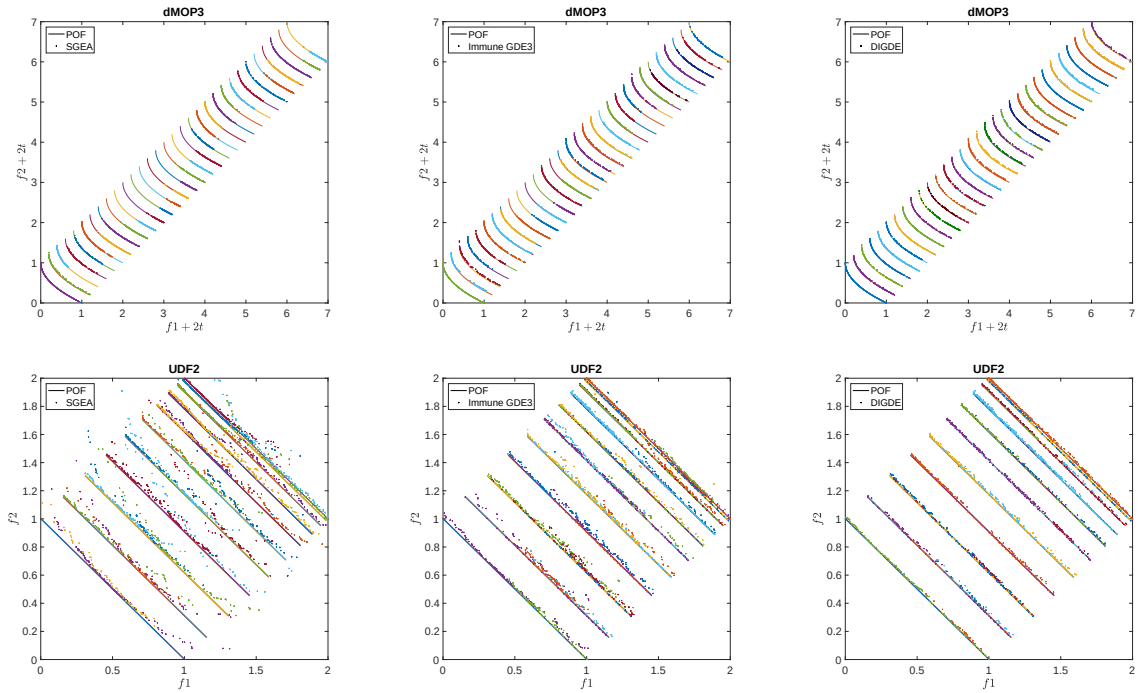

Figure 6.8: Obtained POFs by the three best algorithms on different representative test problems with $n_t = 10$ and $\tau_t = 10$.

in DIGDE selection mechanism. On the other hand, SGEA algorithm obtains the best computational time.

Table 6.7: Algorithm run-time complexity results

Algorithm	**T0**	**T1**	**T2**	**(T2-T1)/T0**
DNSGA-II-A			785.2401	9086.486111
DNSGA-II-B			803.8584	9301.975694
DPSO-4			568.2129	6574.597222
MOEA/D-BR	0.0864	0.1677	518.8979	6003.821759
MOEA/D-CER			602.4875	6971.293980
Immune GDE3			225.4639	2607.594907
DIGDE			368.1082	4258.570601
SGEA	0.1725	0.1832	115.7229	669.7953623

6.6 Summary

There has been a significant amount of research work in DMO. However, there is a lack of indicator-based algorithms in the field of DMO (see Section 4.3). Therefore, in this chapter, a new DMOEA based on IGD indicator has been proposed.

The resulting DMOEA called DIGDE (Distance-based Immune Generalized Differential Evolution) is an improved version of the previously proposed DMOEA called Immune GDE3 (see Chapter 5). The main feature of DIGDE is the use of the IGD indicator in their selection mechanism. The empirical validation of DIGDE is also presented in this chapter. To evaluate the performance of DIGDE, the results obtained by five performance metrics were compared against to those obtained by other popular DMOEAs, including Immune GDE3. Experimental results demonstrated that DIGDE significantly improved its performance in most of test benchmark problems. Therefore, DIGDE is very promising solving DMOPs. Furthermore, DIGDE presented better robustness and good performance in the presence of different change frequencies.

Since IGD metric measures not only the approximation to the POF but also the distribution over the POF, it is a promising metric to propose the new selection mechanism used by DIGDE. Therefore, the improved performance of such algorithm could be attributed to the selection based on IGD contributions.

The next chapter summarizes the research of this thesis and presents the obtained conclusions. Furthermore, some possible paths for future work are presented.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Evolutionary Multi-objective Optimization has been extensively studied in the last few years. However, there is a research gap to tackle multi-objective optimization in dynamic environments using nature-inspired algorithms. The main goal of this thesis was to contribute in this research area by proposing algorithms able to tackle DMOPs with competitive results. Therefore, in this thesis, the proposal of two different DMOEAs was presented. The first DMOEA, called Immune GDE3, consists of a hybrid approach which combines the advantages of Differential Evolution and the Artificial Immune System. In Immune GDE3, the Generalized Differential Evolution (GDE3) algorithm was adopted as MOP optimizer, while an immune response based on the clonal selection principle was used as change reaction mechanism (see Chapter 5). On the other hand, the proposal of a new DMOEA based on performance indicator was also presented. The indicator-based DMOEA was called DIGDE. This algorithm is an improved version of Immune GDE3. The main difference with respect to Immune GDE3 is based on the use of IGD contributions in its selection mechanism. Therefore, different from Immune GDE3 which uses only crowding distance, DIGDE uses the contributions to IGD to select the best individuals for next generations (see Chapter 6).

Both algorithms were empirically validated using a set of benchmark problems representative of the main features presented in real-world dynamic optimization

problems. Furthermore, the experiments were carried out using different performance metrics in order to evaluate the ability of the algorithms to track the changes in the Pareto front and the Pareto Set and its capacity to find a good distribution of solutions through the Pareto front obtained. The results obtained by these metrics were computed for each algorithm to carry out a direct comparison against other state-of-the-art algorithms. From the obtained results, statistically validated, it was found that Immune GDE3 and DIGDE were very promising for dealing with DMOPs.

Different issues of the proposed approaches were analyzed: (1) the general performance of both algorithms in comparison with other state-of-the-art algorithms, using a novel set of benchmark functions, (2) the sensitivity of both algorithms to different change severities and frequencies, (3) the role of their change reaction mechanism based on an immune response and (4) the improvement capability of Immune GDE3 using a performance metric in its selection mechanism. In this thesis, an adaptation of a binary metric to evaluate the performance of DMOEAs was also included.

The overall results of the experiments carried out lead to the following findings:

- Despite the different nature of the POF and the POS, Immune GDE3 and DIGDE were able to track the Pareto optimal front in most of the test problems, obtaining competitive results according to all the metrics (including the most challenging problems, Type III test instances with discrete and trigonometric nature).

- The specialized literature suggests that using binary metrics strengthen empirical evidence when MOEAs are tested and compared [109]. This research study provides interesting results, for example, Immune GDE3, DIGDE and SGEA achieve similar performance based on unary metrics. However, considering the binary metric (C-metric), Immune GDE3 and DIGDE outperform SGEA in most of the problems. Adapting binary metrics, such as two-set coverage, for dynamic optimization allows to confirm the achieved performance. Thus, using unary metrics provides a good evidence of algorithmic performance but does not necessarily provide a strong basis for comparison among MOEAs. Therefore, the usage and proposals of binary performance measures in DMOO are also promising research areas in order to improve the empirical validation of DMOEAs.

- For all metrics and test problems, it was clearly observed that in most cases, DIGDE and Immune GDE3 were in first and second rank, respectively. The good performance of these two algorithms could be explained as follows: both algorithms maintain a suitable population diversity, it could be attributed to the fact that those algorithms employ a memory to store the best members in the population and use the cloning process to generate clones of those members. Before a change occurs, the algorithms store the best solutions and maintain these solutions for next iterations. After a change occurs, both algorithms take advantage of the solutions in memory if the severity of changes does not change significantly, or use the cloning process, otherwise. Also, Immune GDE3 uses the non-dominance criteria and crowding distance to approximate and maintain a good distribution of solutions, respectively. On the other hand, DIGDE uses the IGD contributions to approximate the solutions to the POF, since IGD also measures the diversity of solutions over the POF. The incorporation of IGD in DIGDE selection mechanism also allows a good diversity of the obtained solutions. The statistical results confirm such findings.

- The poor performance of DNSGA-II versions can be justified as follows. DNSGA-II versions use a diversity introduction scheme. This kind of approaches depends on the ability of the optimization algorithm (NSGA-II in this case) to detect problem changes [77]. So that, it is possible that such algorithm may not be competitive when the problem changes are severe or very frequent. For the configuration ($n_t = 5$, $\tau_t = 5$), the frequency of changes was very frequent.

- DPSO was competitive, however, it could present problems to maintain the good distribution of solutions in some cases, especially in problems with a discrete nature of the POF, and can not be applied in problems with three or more objectives.

- SGEA showed a good performance regarding proximity metrics. However, its performance decreased considerably according to the distribution metrics. This behavior could be explained because SGEA does not have a mechanism to increase diversity when changes occur. Therefore, it seems to be vulnerable to loss of diversity.

- The variation of test problems parameters favored the good performance of the compared algorithms because the changes were less frequent and less severe. However, the obtained results showed that Immune GDE3 and DIGDE were not sensitive to different combinations of change severity and change frequency. In contrast, the performance of MOEA/D-BR, MOEA/D-CER and SGEA tends to decrease in the presence of frequent changes in the environment.

- Generally, the performance of the algorithms decrease for configuration ($n_t = 5$, $\tau_t = 5$), i.e., when the severity was considerable and the changes were very frequent. The effect of τ_t is more significant than n_t in the difficulty of the test problems.

- The change reaction mechanism based on immune response involved in Immune GDE3 and DIGDE plays an important role to react to the changes in the environment and to preserve the diversity of the population in the set of DMOPs solved in this study. The combination of immune response and GDE3 was a good alternative to solve this kind of problems. The third experiment in the empirical validation of Immune GDE3 confirms such finding (see Section 5.3.3).

- The thorough empirical assessment shows that Immune GDE3 and DIGDE are highly competitive optimization algorithms that can quickly converge to new POF positions. Furthermore, such assessment also provides significant insights on Differential Evolution as a very capable searching algorithmic technique when dealing with dynamic environments. DE has shown its ability to tackle static optimization problems and now looks like a competitive approach for DMOPs.

- In DMOO a faster convergence is needed in order to track environmental changes as quickly as possible. In experiment one of the empirical validation of Immune GDE3, Immune GDE3 is compared against other DMOEAs with very severe and frequent changes. Thus, algorithms with faster convergence are needed. GDE3 and MOEA/D are known as algorithms with faster convergence [101],[57]. The rest of the algorithms converge more slowly when

compared to GDE3 and MOEA/D. Such convergence behavior could also justify the good performance of Immune GDE3, DIGDE and MOEA/D versions against the rest of DMOEAs.

From the previous findings it was concluded that, in general, the DE-based DMOEAs presented in this thesis showed a highly competitive performance when they were compared against other algorithms of the state-of-the-art. The immune response included in Immune GDE3 and DIGDE is a good alternative to track the new position of solutions in the search space in spite of very frequent changes in the environment. A good distribution of solutions through the Pareto fronts in each time step is also one of the main advantages of both algorithms in comparison with the rest of the algorithms. DE is a promising optimization algorithm not only to tackle static optimization problems but also to deal with DMOPs. Therefore, at this point, it can be concluded that the hypothesis initially proposed in this thesis was confirmed. In addition, all the objectives set and the expected contributions were successfully reached.

7.2 Future work

Part of the future work includes solving other test problems with different characteristics, e.g., DMOPs with dynamic constraints and dynamic many-objective problems. Also, other DMOEAs based on different performance indicators could be proposed. On the other hand, since there is a lack regarding test instances and appropriate performance measures, the development of test instances with more than two objectives is an open research area. Furthermore, specific performance metrics for dynamic multi-objective optimization should be proposed. Adopting parameter control techniques for the proposed algorithms is also part of our research interest.

Bibliography

[1] Manuel Blanco Abello, Lam Thu Bui, and Zbignew Michalewicz. An adaptive approach for solving dynamic scheduling with time-varying number of tasks — part ii. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1711–1718, 2011. doi: 10.1109/CEC.2011.5949821.

[2] P. Amato and M. Farina. *An ALife-Inspired Evolutionary Algorithm for Dynamic Multiobjective Optimization Problems*, pages 113–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-32400-3. doi: 10.1007/3-540-32400-3_9. URL https://doi.org/10.1007/3-540-32400-3_9.

[3] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan. Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. Technical report, Nanyang Technological University, Singapore, November 2016.

[4] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy. *Soft Comput.*, pages 1–22, 2015. ISSN 1433-7479. doi: 10.1007/s00500-015-1820-4.

[5] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. *Dynamic Multi-objective Optimization Using Evolutionary Algorithms: A Survey*, pages 31–70. Springer International Publishing, Cham, 2017. ISBN 978-3-319-42978-6. doi: 10.1007/978-3-319-42978-6_2.

[6] Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-

based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, March 2011. ISSN 1063-6560. doi: 10.1162/EVCO_a_00009.

[7] Nicola Beume, Boris Naujoks, and Michael Emmerich. Sms-emoa: Multi-objective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653 – 1669, 2007. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2006.08.008. URL http://www.sciencedirect.com/science/article/pii/S0377221706005443.

[8] S. Biswas, S. Das, P.N. Suganthan, and C.A. Coello Coello. Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions. In *2014 IEEE Congress on Evolutionary Computation (CEC 2014)*, pages 3192–3199, July 2014. doi: 10.1109/CEC.2014.6900487.

[9] J T Buchanan. A naïve approach for solving mcdm problems: the guess method. *Journal of the Operational Research Society*, 48(2):202–206, Feb 1997. ISSN 1476-9360. doi: 10.1057/palgrave.jors.2600349. URL https://doi.org/10.1057/palgrave.jors.2600349.

[10] L. T. Bui, H. A. Abbass, and J. Branke. Multiobjective optimization for dynamic environments. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2349–2356 Vol. 3, Sept 2005. doi: 10.1109/CEC.2005.1554987.

[11] Lam Thu Bui and Zbigniew Michalewicz. An evolutionary multi-objective approach for dynamic mission planning. In *IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2010.

[12] Felipe Campelo, Frederico G. Guimarães, and Hajime Igarashi. *Overview of Artificial Immune Systems for Multi-objective Optimization*, pages 937–951. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-70928-2. doi: 10.1007/978-3-540-70928-2_69. URL https://doi.org/10.1007/978-3-540-70928-2_69.

[13] Mauro Campos and Renato A. Krohling. Entropy-based bare bones particle swarm for dynamic constrained optimization. *Knowl. Based Syst.*, 97:203 – 223, 2016. ISSN 0950-7051. doi: http://doi.org/10.1016/j.knosys.2015.12.017.

[14] Leandro R. de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002. ISBN 1852335947.

[15] A. Charnes and W. W. Cooper. Management models and industrial applications of linear programming. *Management Science*, 4(1):38–91, 1957. doi: 10.1287/mnsc.4.1.38.

[16] H. Chen, M. Li, and X. Chen. Using diversity as an additional-objective in dynamic multi-objective optimization algorithms. In *2009 Second International Symposium on Electronic Commerce and Security*, volume 1, pages 484–487, May 2009. doi: 10.1109/ISECS.2009.42.

[17] C.A.C. Coello and G.B. Lamont. *Applications of Multi-objective Evolutionary Algorithms*. Advances in natural computation. World Scientific, 2004. ISBN 9789812561060.

[18] Carlos A. Coello Coello and Nareli Cruz Cortes. An approach to solve multiobjective optimization problems based on an artificial immune system. In J. Timmis and P.J. Bentley, editors, *Proceedings of the 1st International Conference on Artificial Immune Systems*, pages 212–221, 2002.

[19] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, USA, 2002. ISBN 0-3064-6762-3.

[20] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387332545.

[21] Jared L. Cohon and David H. Marks. A review and evaluation of multiobjective programing techniques. *Water Resources Research*, 11(2):208–220, 1975. ISSN 1944-7973. doi: 10.1029/WR011i002p00208. URL http://dx.doi.org/10.1029/WR011i002p00208.

[22] NareliCruz Cortés and CarlosA.Coello Coello. Multiobjective optimization using ideas from the clonal selection principle. In Erick Cantú-Paz, JamesA. Foster, Kalyanmoy Deb, LawrenceDavid Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell Standish, Graham Kendall, Stewart Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, MitchA. Potter, AlanC. Schultz, KathrynA. Dowsland, Natasha Jonoska, and Julian Miller, editors, *Genetic and Evolutionary Computation — GECCO 2003*, volume 2723 of *Lecture Notes in Computer Science*, pages 158–170. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40602-0. doi: 10.1007/3-540-45105-6_22.

[23] K. Deb. *Optimization for Engineering Design: Algorithms and Examples*. Prentice-Hall of India, 2004. ISBN 9788120309432.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.*, 6(2):182–197, Apr 2002. ISSN 1089-778X. doi: 10.1109/4235.996017.

[25] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 047187339X.

[26] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter Scalable Test Problems for Evolutionary Multiobjective Optimization, pages 105–145. Springer London, London, 2005. ISBN 978-1-84628-137-2. doi: 10.1007/1-84628-137-7_6.

[27] Kalyanmoy Deb, UdayaBhaskara Rao N., and S. Karthik. Dynamic multiobjective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 803–817. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-70927-5. doi: 10.1007/978-3-540-70928-2_60.

[28] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.*, 1(1):3 – 18, 2011. ISSN 2210-6502. doi: http://dx.doi.org/10.1016/j.swevo.2011.02.002.

[29] Francis Ysidro Edgeworth. *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. C. Kegan Paul and Co., London, England, 1881.

[30] Matthias Ehrgott. *Multicriteria Optimization*. Springer-Verlag Berlin Heidelberg, Berlin, June 2005.

[31] Michael Emmerich, Nicola Beume, and Boris Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, EMO'05, pages 62–76, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-24983-4, 978-3-540-24983-2. doi: 10.1007/978-3-540-31880-4_5. URL http://dx.doi.org/10.1007/978-3-540-31880-4_5.

[32] Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006. ISBN 0470091916.

[33] Marco Farina, K. Deb, and P. Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Trans. Evol. Comput.*, 8(5):425–442, Oct 2004. ISSN 1089-778X. doi: 10.1109/TEVC.2004.831456.

[34] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1-55860-299-2. URL http://dl.acm.org/citation.cfm?id=645513.657757.

[35] Fabio Freschi, Carlos A. Coello Coello, and Maurizio Repetto. *Multiobjective Optimization and Artificial Immune Systems: A Review*, chapter 1, pages 1–21.

Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies. IGI Global, 2009.

[36] Saul Gass and Thomas Saaty. The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1-2):39–45, 1955. ISSN 1931-9193. doi: 10.1002/nav.3800020106. URL http://dx.doi.org/10.1002/nav.3800020106.

[37] C. K. Goh and K. C. Tan. An investigation on noisy environments in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 11(3):354–381, June 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2006.882428.

[38] Chi-Keong Goh and Kay Chen Tan. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Trans. Evol. Comput.*, 13(1):103–127, February 2009. doi: 10.1109/TEVC.2008.920671.

[39] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. ISBN 0201157675.

[40] David E. Goldberg and Robert E. Smith. Nonstationary function optimization using genetic algorithm with dominance and diploidy. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 59–68, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc. ISBN 0-8058-0158-8. URL http://dl.acm.org/citation.cfm?id=42512.42521.

[41] YV HAIMES YV, LS LASDON LS, and DA WISMER DA. On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-1(3):296–297, 7 1971. ISSN 0018-9472.

[42] Emma Hart and Jonathan Timmis. Application areas of ais: The past, the present and the future. In Christian Jacob, MarcinL. Pilat, PeterJ. Bentley, and JonathanI. Timmis, editors, *Artificial Immune Systems*, volume 3627 of

Lecture Notes in Computer Science, pages 483–497. Springer Berlin Heidelberg, 2005.

[43] Iason Hatzakis and David Wallace. Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 1201–1208, New York, NY, USA, 2006. ACM. ISBN 1-59593-186-4. doi: 10.1145/1143997.1144187.

[44] M. Helbig and A. P. Engelbrecht. Issues with performance measures for dynamic multi-objective optimisation. In *2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 17–24, April 2013. doi: 10.1109/CIDUE.2013.6595767.

[45] M. Helbig and A. P. Engelbrecht. Analysing the performance of dynamic multi-objective optimisation algorithms. In *2013 IEEE Congress on Evolutionary Computation*, pages 1531–1539, June 2013. doi: 10.1109/CEC.2013.6557744.

[46] M. Helbig and A.P. Engelbrecht. Analyses of guide update approaches for vector evaluated particle swarm optimisation on dynamic multi-objective optimisation problems. In *2012 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, June 2012. doi: 10.1109/CEC.2012.6252882.

[47] Mardé Helbig and Andries P. Engelbrecht. Benchmarks for dynamic multi-objective optimisation algorithms. *ACM Comput. Surv.*, 46(3):37:1–37:39, January 2014. ISSN 0360-0300. doi: 10.1145/2517649.

[48] MardÃ Â© Helbig and Andries P. Engelbrecht. Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm Evol. Comput.*, 14:31 – 47, 2014. ISSN 2210-6502. doi: http://dx.doi.org/10.1016/j.swevo.2013.08.004.

[49] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87 vol.1, Jun 1994. doi: 10.1109/ICEC.1994.350037.

[50] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Iterative approach to indicator-based multiobjective optimization. In *2007 IEEE Congress on Evolutionary Computation*, pages 3967–3974, Sept 2007. doi: 10.1109/CEC.2007.4424988.

[51] S. Jiang and S. Yang. Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons. *IEEE Trans. Cybern.*, 47(1):198–211, Jan 2017. ISSN 2168-2267. doi: 10.1109/TCYB.2015.2510698.

[52] S. Jiang and S. Yang. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(1):65–82, Feb 2017. ISSN 1089-778X. doi: 10.1109/TEVC.2016.2574621.

[53] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.

[54] Joshua Knowles and David Corne. *Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization*, pages 757–771. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-70928-2. doi: 10.1007/978-3-540-70928-2_57. URL https://doi.org/10.1007/978-3-540-70928-2_57.

[55] Joshua D. Knowles and David W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172, June 2000. ISSN 1063-6560. doi: 10.1162/106365600568167.

[56] S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *IEEE Congress on Evolutionary Computation, CEC 2006.*, pages 1179–1186, 2006. doi: 10.1109/CEC.2006.1688443.

[57] S. Kukkonen and J. Lampinen. Gde3: the third evolution step of generalized differential evolution. In *2005 IEEE Congress onEvolutionary Computation.*, volume 1, pages 443–450 Vol.1, Sept 2005. doi: 10.1109/CEC.2005.1554717.

[58] Saku Kukkonen and Jouni Lampinen. An extension of generalized differential evolution for multi-objective optimization with constraints. In Xin Yao, EdmundK. Burke, JoséA. Lozano, Jim Smith, JuanJulián Merelo-Guervós, JohnA. Bullinaria, JonathanE. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 752–761. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-23092-2. doi: 10.1007/978-3-540-30217-9_76.

[59] Saku Kukkonen and Jouni Lampinen. Comparison of generalized differential evolution algorithm to other multi-objective evolutionary algorithms. In *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS 2004*, 2004.

[60] M. Laumanns, E. Zitzler, and L. Thiele. A unified model for multi-objective evolutionary algorithms with elitism. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 46–53 vol.1, 2000. doi: 10.1109/CEC.2000.870274.

[61] X. Li, J. Branke, and M. Kirley. On performance metrics and particle swarm methods for dynamic multiobjective optimization problems. In *2007 IEEE Congress on Evolutionary Computation*, pages 576–583. IEEE, September 2007.

[62] C. Liu. New dynamic multiobjective evolutionary algorithm with core estimation of distribution. In *2010 International Conference on Electrical and Control Engineering (ICECE)*, pages 1345–1348. IEEE, June 2010.

[63] Chun-An Liu and Yuping Wang. *New Evolutionary Algorithm for Dynamic Multiobjective Optimization Problems*, volume 4221 of *Advances in Natural Computation. ICNC 2006. Lecture Notes in Computer Science*, pages 889–892. Springer, Berlin, Heidelberg, 2006.

[64] Ruochen Liu, Yangyang Chen, Wenping Ma, Caihong Mu, and Licheng Jiao. A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model. *Soft Comput.*, 18(10):1913–1929, 2014. ISSN 1433-7479. doi: 10.1007/s00500-013-1175-7.

[65] A. D. Manriquez, G. T. Pulido, and J. G. R. Torres. Handling dynamic multiobjective problems with particle swarm optimization. In *2nd International Conference on Agents and Artificial Intelligence (ICAART 2010)*, pages 337–342, 2010.

[66] M.G. Matínez-Peñaloza and E. Mezura-Montes. Immune generalized differential evolution for dynamic multiobjective optimization problems. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1918 – 1925. IEEE, May 2015.

[67] Adriana Menchaca-Mendez and Carlos A. Coello Coello. Gd-moea: A new multi-objective evolutionary algorithm based on the generational distance indicator. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization*, pages 156–170, Cham, 2015. Springer International Publishing. ISBN 978-3-319-15934-8.

[68] Efrén Mezura-Montes, Mariana Edith Miranda-Varela, and Rubí del Carmen Gómez-Ramón. Differential evolution in constrained numerical optimization: An empirical study. *Inf. Sci.*, 180(22):4223 – 4262, 2010. ISSN 0020-0255. doi: http://dx.doi.org/10.1016/j.ins.2010.07.023.

[69] Kaisa M. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[70] J.N. Morse. Reducing the size of the nondominated set: Pruning by clustering. *Computers & Operations Research*, 7(1):55–66, 1980. ISSN 0305-0548. doi: https://doi.org/10.1016/0305-0548(80)90014-3.

[71] A. Muruganantham, K. C. Tan, and P. Vadakkepat. Evolutionary dynamic multiobjective optimization via kalman filter prediction. *IEEE Trans. Cybern.*, 46(12):2862–2873, Dec 2016. ISSN 2168-2267. doi: 10.1109/TCYB.2015.2490738.

[72] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. doi: 10.1093/comjnl/7.4.308. URL +http://dx.doi.org/10.1093/comjnl/7.4.308.

[73] Shams K. Nseef, Salwani Abdullah, Ayad Turky, and Graham Kendall. An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems. *Knowl. Based Syst.*, 104:14 – 23, 2016. ISSN 0950-7051. doi: http://doi.org/10.1016/j.knosys.2016.04.005.

[74] Vilfredo Pareto. *Cours d'économie politique*. F. Rouge, Lausanne, 1896. ISBN 9782600040143.

[75] Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 3540209506.

[76] Shuqu Qian, Yongqiang Ye, Bin Jiang, and Guofeng Xu. A micro-cloning dynamic multiobjective algorithm with an adaptive change reaction strategy. *Soft Comput.*, 21(13):3781–3801, Jul 2017. ISSN 1433-7479. doi: 10.1007/s00500-016-2370-0.

[77] Carlo Raquel and Xin Yao. *Evolutionary Computation for Dynamic Optimization Problems*, chapter Dynamic Multi-objective Optimization: A Survey of the State-of-the-Art, pages 85–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-38416-5. doi: 10.1007/978-3-642-38416-5_4.

[78] David Rathbun, Sean Kragelund, Anawat Pongpunwattana, and Brian Capozzi. An evolution based path planning algorithm for autonomous motion of a uav through uncertain environments. In *Document Analysis Systems*, 2002. doi: 10.1109/DASC.2002.1052946.

[79] Margarita Reyes-sierra and Carlos A. Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE RESEARCH*, 2(3): 287–308, 2006.

[80] Hendrik Richter. *Dynamic Fitness Landscape Analysis*, pages 269–297. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-38416-5. doi: 10.1007/978-3-642-38416-5_11. URL https://doi.org/10.1007/978-3-642-38416-5_11.

[81] Richard S. Rosenberg. *Simulation of genetic populations with biochemical properties*. Phd thesis, Department of Communication Sciences, University of Michigan, Ann Arbor, Michigan, USA, June 1967.

[82] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Phd thesis, Vanderbilt University, Ann Arbor, Michigan, USA, June 1984.

[83] Jason R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Massachusetts Institute of Technology. Deparment of Aeronautics and Astronautics, 1995.

[84] Ronghua Shang, Licheng Jiao, Maoguo Gong, and Bin Lu. Clonal selection algorithm for dynamic multiobjective optimization. In *Proceedings of the 2005 International Conference on Computational Intelligence and Security - Volume Part I*, CIS'05, pages 846–851, Berlin, Heidelberg, 2005. Springer-Verlag.

[85] Ronghua Shang, Licheng Jiao, Yujing Ren, Lin Li, and Luping Wang. Quantum immune clonal coevolutionary algorithm for dynamic multiobjective optimization. *Soft Comput.*, 18(4):743–756, Apr 2014. ISSN 1433-7479. doi: 10.1007/s00500-013-1085-8.

[86] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.*, 2(3):221–248, September 1994. ISSN 1063-6560. doi: 10.1162/evco.1994.2.3.221. URL http://dx.doi.org/10.1162/evco.1994.2.3.221.

[87] Ralph E. Steuer. *The Tchebycheff Procedure of Interactive Multiple Objective Programming*, pages 235–249. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989. ISBN 978-3-642-74919-3. doi: 10.1007/978-3-642-74919-3_8.

[88] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. ISSN 1573-2916. doi: 10.1023/A:1008202821328. URL http://dx.doi.org/10.1023/A:1008202821328.

[89] Jiangjun Tang, Sameer Alam, Chris Lokan, and Hussein A. Abbass. A multi-objective evolutionary method for dynamic airspace re-sectorization using sectors clipping and similarities. In *IEEE Congress on Evolutionary Computation'12*, pages 1–8, 2012.

[90] Y. Tian, X. Zhang, R. Cheng, and Y. Jin. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 5222–5229, July 2016. doi: 10.1109/CEC.2016.7748352.

[91] Jon Timmis, Paul Andrews, and Emma Hart. On artificial immune systems and swarm intelligence. *Swarm Intell.*, 4(4):247–273, 2010. ISSN 1935-3812. doi: 10.1007/s11721-010-0045-5.

[92] Santosh Tiwari, Georges Fadel, and Kalyanmoy Deb. Amga2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. *Engineering Optimization*, 43(4):377–401, 2011. doi: 10.1080/0305215X.2010.491549. URL http://dx.doi.org/10.1080/0305215X.2010.491549.

[93] David A. Van Veldhuizen and David A. Van Veldhuizen. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. Technical report, Evolutionary Computation, 1999.

[94] D. w. Gong, J. Sun, and Z. Miao. A set-based genetic algorithm for interval many-objective optimization problems. *IEEE Trans. Evol. Comput.*, PP(99): 1–1, 2017. ISSN 1089-778X. doi: 10.1109/TEVC.2016.2634625.

[95] S. Wan and D. Wang. A novel differential evolution for dynamic multiobjective optimization with adaptive immigration scheme. In *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, pages 502–507, Oct 2013. doi: 10.1109/ICCSNT.2013.6967163.

[96] Yu Wang and Bin Li. Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. *Memetic Computing*, 2(1):3–24, Mar

2010. ISSN 1865-9292. doi: 10.1007/s12293-009-0012-0. URL https://doi.org/10.1007/s12293-009-0012-0.

[97] Yan Wu, Yaochu Jin, and Xiaoxiong Liu. A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Comput.*, 19(11):3221–3235, November 2015. ISSN 1432-7643. doi: 10.1007/s00500-014-1477-4.

[98] B. Xu, Y. Zhang, D. Gong, Y. Guo, and M. Rong. Environment sensitivity-based cooperative co-evolutionary algorithms for dynamic multi-objective optimization. *IEEE-ACM Trans. Comput. Biol. Bioinform.*, PP(99):1–1, 2017. ISSN 1545-5963. doi: 10.1109/TCBB.2017.2652453.

[99] Shengxiang Yang and Xin Yao, editors. *Evolutionary Computation for Dynamic Optimization Problems*, volume 490 of *Studies in Computational Intelligence*. Springer-Verlag Berlin Heidelberg, 2013.

[100] Sang Zeng, Guang Chen, Liang Zheng, Hui Shi, H. de Garis, Lixin Ding, and Lishan Kang. A dynamic multi-objective evolutionary algorithm based on an orthogonal design. In *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, pages 573–580, 2006.

[101] Q. Zhang and H. Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.*, 11(6):712–731, Dec 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2007.892759.

[102] Q. Zhang, A. Zhou, and Y. Jin. Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63, Feb 2008. ISSN 1089-778X. doi: 10.1109/TEVC.2007.894202.

[103] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. Evol. Comput.*, 12(1):41 – 63, February 2008. doi: 10.1109/TEVC.2007.894202.

[104] Zhuhong Zhang. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Appl. Soft Comput*, 8 (2):959–971, March 2008.

[105] Zhuhong Zhang and Shuqu Qian. Artificial immune system in dynamic environments solving time-varying non-linear constrained multi-objective problems. *Soft Comput.*, 15(7):1333–1349, 2011. ISSN 1432-7643. doi: 10.1007/s00500-010-0674-z.

[106] Aimin Zhou, Yaochu Jin, Qingfu Zhang, Bernhard Sendhoff, and Edward Tsang. *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings*, chapter Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization, pages 832–846. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-70928-2. doi: 10.1007/978-3-540-70928-2_62.

[107] Aimin Zhou, Yaochu Jin, and Qingfu Zhang. A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Trans. Cybern.*, 44(1):40–53, Jan 2014. ISSN 2168-2267. doi: 10.1109/TCYB.2013.2245892.

[108] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001. International Center for Numerical Methods in Engineering.

[109] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.*, 7(2):117–132, April 2003. ISSN 1089-778X. doi: 10.1109/TEVC.2003.810758.

[110] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *in Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII*, pages 832–842. Springer, 2004.

[111] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.*, 3(4):257–271, 1999. doi: 10.1109/4235.797969.

[112] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, June 2000. ISSN 1063-6560. doi: 10.1162/106365600568202.

[113] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2): 173–195, June 2000. ISSN 1063-6560. doi: 10.1162/106365600568202. URL http://dx.doi.org/10.1162/106365600568202.