GIAN MAURICIO FRITSCHE

HYPER-HEURISTIC BASED PARTICLE SWARM OPTIMIZATION FOR MANY-OBJECTIVE PROBLEMS

Master dissertation submitted as a partial requirement for the degree of Master in Informatics. Computer Science Department, Federal University of Paraná, Curitiba, Paraná, Brazil.
Advisor: Prof. Aurora Pozo, Ph.D.

CURITIBA

2016

GIAN MAURICIO FRITSCHE

HYPER-HEURISTIC BASED PARTICLE SWARM OPTIMIZATION FOR MANY-OBJECTIVE PROBLEMS

Master dissertation submitted as a partial requirement for the degree of Master in Informatics. Computer Science Department, Federal University of Paraná, Curitiba, Paraná, Brazil.
Advisor: Prof. Aurora Pozo, Ph.D.

CURITIBA

2016

F919h Fritsche, Gian Maurício
Hyper-heuristic based particle swarm optimization for many-objective problems/ Gian Maurício Fritsche. – Curitiba, 2016.
102 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2016.

Orientador: Aurora Pozo .
Bibliografia: p. 81-87.

1. Algoritmos genéticos. 2. Algoritmo - Otimização. 3. Metodologia multicritério. I. Universidade Federal do Paraná. II.Pozo, Aurora. III. Título.

CDD: 006.3823

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
Setor CIÊNCIAS EXATAS
Programa de Pós Graduação em INFORMÁTICA
Código CAPES: 40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **GIAN MAURICIO FRITSCHE**, intitulada: "**Hyper-heuristic Particle Swarm Optimization for Many-objective Problems**", após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua aprovação.

Curitiba, 29 de Fevereiro de 2016.

Prof AURORA TRINIDAD RAMIREZ POZO (UFPR)
(Presidente da Banca Examinadora)

Prof ROBERTO SANTANA HERMIDA (UPV)

Prof SILVIA REGINA VERGILIO (UFPR)

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

CONTENTS

LIST OF FIGURES

LIST OF TABLES

RESUMO

O algoritmo de Otimização por Enxame de Partículas (PSO) é uma meta-heurística inspirada no comportamento de bandos de aves a procura de alimento. Os bons resultados obtidos por esta técnica na otimização de problemas mono-objetivo incentivaram o estudo de variações para problemas multi- objetivo (MOPSO), que também alcançaram bons resultados. Para a adaptação do PSO para problemas multi-objetivo algumas modificações foram necessárias, tais como o uso de um operador para seleção de líder e a aplicação de um operador de arquivamento. Entretanto, a qualidade do algoritmo diminui conforme o aumento do número de objetivos. Encontrar, dentre os diferentes operadores de seleção de líder e de arquivamento, propostos na literatura, os mais apropriado para determinada instância de um problema permite amenizar esta perda de qualidade. Porém esta tarefa não é uma tarefa trivial. Em trabalhos anteriores o uso de hiper-heurística para a seleção de uma combinação apropriada destes operadores é proposta. Hiper-heurísticas são técnicas para a seleção, ou geração, de heurísticas para problemas de busca. Estas técnicas visam a seleção, ou geração, de uma heurística apropriada para determinada instância de um problema ou estágio da busca. Neste trabalho foi abordada a hipótese de que, o uso de métodos de seleção mais avançados poderiam melhorar desempenho do MOPSO baseado em Hiper-heurística (H-MOPSO). Para investigar esta hipótese quatro métodos de seleção foram avaliados e comparados a um algoritmo multi-objetivo estado da arte. Nos resultados apresentados o H-MOPSO obteve melhores resultados na maioria dos problemas.

ABSTRACT

Multi-objective Particle Swarm Optimization (MOPSO) is a promising meta-heuristic to solve Many-Objective Problems (MaOPs), however, its performance decreases as the number of objective functions increases. Selecting a good combination of leader and archiving methods helps the algorithm to deal with the challenges caused by this increase in the number of objectives, but finding the most appropriate combination for a given problem is a hard task. To deal with this issue, previous works proposed the use of a simple hyper-heuristic to select dynamically a good combination of leader and archiving methods and achieved promising results. In this work, we hypothesize that by using more advanced heuristic selection methods we could further improve the performance of the algorithm. To investigate this hypothesis we conducted experimental studies comparing four heuristic selection methods. After selecting the best performing variant from this study, we conducted a second empirical study to compare this variant to a state-of-the-art optimizer, where the resulting algorithm outperformed it in most of the problems investigated.

CHAPTER 1

INTRODUCTION

Several real world applications require the optimization of multi-objective problems (MOP). In those problems the objectives can be conflicting, that means that the optimization of one objective can degrade the optimization of others [3]. So, the simultaneous optimization of the objectives is required. In this way, two solutions may be not comparable as one can be better for one objective and a different one be better for another objective. The purpose of the Multi-Objective Optimization (MOO) is to find the set of Pareto optimal solutions. A solution is called Pareto optimal if one objective cannot be improved without degrading any other. However, an MOP is frequently NP-hard, because of the huge number of possible solutions, sometimes infinite. Usually, the aim is to find an approximation of the optimal set. The quality of this approximation is frequently evaluated by two measures, convergence, and diversity. The convergence evaluates the capability of generating solutions as close as possible to the optimal set. The diversity intent for assessing the ability to generate a set of solutions that well approximates the entire optimal set.

Nowadays, the optimization of multi-objective problems with two or three objectives can be successfully achieved using the state-of-art algorithms such as SMPSO, NSGA-II, and MOEA/D-DRA. However, problems with more than three objectives are present in many real word applications [4], such as Engineering design; Air traffic control; Nurse rostering; Car controlling optimization; and Water supply portfolio planning. Those problems are called Many-Objective Problems (MaOP) [4]. In this kind of problems, the state-of-art multi-objective algorithms fail to achieve convergence and diversity to the Pareto front, due to the increasing of the number of objectives [4]. As most solutions generated become non-dominated, it reduces the convergence pressure, and as a consequence, the selection operation can be a hard task [5]. Due to these challenges, many works have addressed Many-Objective Optimization [4, 6, 7, 8].

Another area that has drawn attention is Hyper-Heuristic (HH). HH are high-level approaches that aim at selecting (or generating), from a pool of low-level heuristics (or parts of heuristics), the most appropriate heuristic to apply given a problem instance and the search stage. Despite the HH characteristics, there are still few works on multi-objective hyper-heuristic [6, 9, 10, 11, 12].

MOPSO algorithms are Particle Swarm Optimization (PSO) algorithms adapted for multi-objective problems [13]. In PSO, the set of solutions is called *swarm*, and its solutions are called particles. Each particle has a velocity and a position, and for each iteration it updates them. The particle position is an n-dimensional point in the search space. The velocity is the size of the step, from one position to another. The particles have the ability to remember their best visited position *pBest* and also the best position visited by the swarm *gBest*. The velocity update is based on those previous positions, which guide the search towards the best solutions found so far.

Some changes are needed to adapt the PSO for multi-objective optimization problems. One of them is how to store the best solutions found so far since there is a set of incomparable solutions in the MOP (the Pareto approximation set). The non-dominated solutions are stored in an external repository to solve this problem. However, the number of solutions stored in the repository may increase quickly, increasing the computational cost to maintain the repository and select solutions from it. In this case, a limited size repository may be used to alleviate this problem. When the repository is full, an archiving method must be employed to decide if a new solution is accepted or not; and, which solution to remove in case of acceptance. There are some archiving methods, such as Multi-level Grid Archiving (MGA) [14, 15], Crowding Distance Archiving (CD) [16] and Ideal Archiving [17], most of them proposed for MOEAs and adapted for MOPSO. Some of these archiving methods aim at increasing the repository convergence, whereas others aim at increasing diversity. The *gBest* is selected from the repository using some leader selection method, such as Crowding Distance [18], NWSum [19], and Sigma [20]. Some leader selection methods aims at promote a search diversity, favoring the less covered areas. Other methods select solutions that are somehow similar to the current particle, to avoid the generation of erratic solutions. The empirical selection of methods for leader selection and archiving may be critical. A method may be better than other depending on the problem due to its different behavior [6, 21, 17].

In this work, it is proposed the use of Hyper-heuristic for automatic selecting archiving and leader selection methods in MOPSO. The objective is to reduce the performance deterioration of MOPSO in Many-Objective problems, using hyper-heuristic to automatic select leader selection and archiving methods. This work is based on related work [6, 22] that proposes a simple roulette-based hyper-heuristic for selecting archiving and leader selection methods in MOPSO (the H-MOPSO), and was able to obtain encouraging results. The proposal is to study other heuristic selection methods to replace the roulette in the H-MOPSO proposed by [22]. The purpose is to evaluate if more sophisticated methods are capable of increasing the H-MOPSO quality, regarding convergence and diversity, for MaOP. Two heuristic selection methods were selected due to its good results presented in literature: a Choice Function (CF) based [9, 23, 24, 1, 12, 25, 11] and a Multi-Armed Bandit (MAB) based [10, 26, 27]. Also, the heuristic selection methods are compared to a simple random-based heuristic selection. Initially, the work of [22] is partially replicated. Also, a deeper study was accomplished where different Choice Function based, and Multi-Armed Bandit based methods were empirically evaluated. After selecting a CF based method and an MAB-based method the parameters of them are configured. The rest of the studies are divided into three parts. First, the selection of the archiving method is evaluated, using as leader selection the method Crowding Distance. Then, for each heuristic selection method, three H-MOPSO strategies were evaluated: the version selecting just the archiving method, the previous version proposed by [22] that selects both methods as a single low-level heuristic and a new version that selects leader selection methods and archiving separately. For each heuristic selection method, an H-MOPSO strategy was chosen to be used. Then, the final four H-MOPSO (each one based on one heuristic selection method) are compared against each other and a state-of-art algorithm MOEA/D-DRA [28]. Finally, among the H-MOPSO variations evaluated, the one with the best performance is compared against the MOEA/D-DRA.

In this work, it was not possible to find strong evidence that a heuristic selection method evaluated is better than any other with a statistical difference in both IGD and HV. However, all H-MOPSO versions are capable of increasing MOPSO performance and reaching competitive results against the state-of-art algorithm MOEA/D-DRA. The H-MOPSO variation with the best results was the H-MOPSO-ACF. The H-MOPSO-ACF variation selects both archiving and leader selection methods together as a single low-level heuristic and employs the Adaptive Choice Function heuristic as the selection method. When compared to the MOEA/D-DRA, the H-MOPSO-ACF had the best IGD and Hypervolume values, with a statistical difference, in most cases. Moreover, it is possible to conclude that the use of hyper-heuristic is capable of reducing the convergence and diversity difficulty present in MaOP and achieving competitive results compared with a state-of-art algorithm.

1.1 Motivation

Several real word problems have many objectives [4]. However, the multi-objective optimization techniques suffer a quality deterioration when applied to this kind of problem [4]. Some works have focused on Many-objective optimization algorithms, such as MOEA/DD [7], MOMBI-II [8], and H-MOPSO [6]. Another motivation for this work is the use of Hyper-heuristic. Despite the small number of works on multi-objective literature [9], the use of Hyper-heuristic has shown good results when compared with the state-of-art multi-objective algorithms [9, 10, 6, 11]. In [6] a Hyper-heuristic strategy using a roulette-based heuristic selection is used for improving an MOPSO and compared to the state-of-art. The results in [6] were competitive against a state-of-art algorithm. The simplicity of the used heuristic selection suggests that a more sophisticated method, with another strategy of learning and selection, may increase the H-MOPSO quality. In Hyper-heuristic and Adaptive Operator Selection literature, including some works on multi-objective problems, two kinds of methods have shown competitive and efficient results: the Choice Function [9, 11, 12] based and the Multi-Armed Bandit based [10, 11, 26]. In this work, the evaluation of Choice Function based and MultiArmed Bandit based heuristic selection methods on H-MOPSO framework is proposed.

1.2 Objectives

The overall purpose of this work is to reduce the quality deterioration suffered by the Multi-Objective Particle Swarm Optimization algorithms when applied in Many-objective problems.

To accomplish the overall objective some specific objectives are proposed:

- The study and evaluation of other heuristic selection methods on the H-MOPSO framework for Many-objective problems.
- The study and evaluation of different Choice Function based methods.
- The study and evaluation of different Multi-Armed Bandit based methods.
- The study and evaluation of different heuristic selection methods.
- The evaluation of a new structure for H-MOPSO that selects the archiving and leader selection methods separately.
- The evaluation of the studied structures and methods compared to a state-of-art algorithm.

1.3 Contributions

In this work, the H-MOPSO algorithm proposed by [22] is used. In an initial study, that work is partially replicated, and the obtained results are favorable, supporting [22] conclusions. In another study, two Choice Function based heuristic selection methods are evaluated. There is no evidence that one outperforms the other with statistical difference. However, the Adaptive Choice Function had better results mainly for the IGD quality indicator. Also, three MAB-based methods were evaluated. Any of them showed statistical difference against the other two, but the FRRMAB-UCB1 method had the best average ranking in both IGD and HV quality indicators. Three H-MOPSO strategies were evaluated: one selecting just archiving; one selecting both methods together; one selecting archiving and leader selection methods separately. The H-MOPSO strategies that select both methods showed to be better when compared with the version selecting just archiving. Also, all H-MOPSO variations were able to get competitive results against the state-of-art algorithm MOEA/D-DRA. Moreover, the H-MOPSO-ACF was the variant with better results, with a statistical difference to MOEA/D-DRA, in almost all cases.

Throughout the master degree, three papers were published: On the first publication [29], a case study of automatic parameter configuration, a related area to hyper-heuristic, is presented. In the case study, two methods were evaluated to configure the Boids algorithm parameters: the Iterated Racing and a Differential Evolution. The second work [11], as co-author, published at the Genetic and Evolutionary Computation Conference (GECCO 2015). In that work a Choice Function and a Multi-Armed Bandit based heuristic selection methods are evaluated on a Hyper-heuristic for the Integration and Test Order Problem. Also, a publication in proceedings [30] focused on detection and representation of the interactions between the components of a multi-objective problem. The detection was made during the optimization process, and three statistical correlation methods were evaluated.

1.4 Organization

The remaining of this work is organized as follows:

First, in Chapter 2, related work is presented. Some recent proposals and comments for many-objective optimization and hyper-heuristic fields are described. Then the theoretical foundation for multi-objective optimization is presented in Chapter 3. Also, the many-objective optimization and the Multi-Objective Particle Swarm Optimization (Section 3.1) are described.

In Chapter 4, the hyper-heuristic concept is described. Two heuristic selection methods are detailed, the Choice Function based and the Multi-Armed Bandit based. The proposal of this work is presented in Chapter 5. First, a related work is presented, used as a base to this one. Then, the proposed studies are presented. Finally, the methodology used to evaluate the studies and hypotheses are described.

In Chapter 6, the experimental evaluation is presented. Nine studies are evaluated to assess different heuristic selection methods and different hyper-heuristic strategies. Finally, in Chapter 7, the conclusions are displayed and, as well as, some topics to be considered in future works. The work has four appendices: the Appendix A presents the extended results of a preliminary research. The detailed results for different parameter configurations of ACF are presented in Appendix B. The Appendix C presents the detailed results for different parameter configurations of UCB1. Finally, the detailed results for different H-MOPSO strategies are presented in Appendix D.

CHAPTER 2

RELATED WORK

In this chapter, related work to this proposal is presented. The primary basis of this work is presented in [6]. In that work, the use of Hyper-heuristic to increase MOPSO performance in Many-Objective Optimization is proposed (the H-MOPSO). The approach described in [6], a low-level heuristic is a pair of an archiving with a leader selection. A simple roulette-based hyper-heuristic is used to select the low-level heuristics. The strategy selects, once for each iteration, a low-level heuristic to be applied. Due to the multi-objective scenario, the hyper-heuristic is guided by a quality indicator. The R2 indicator is used, because of its desirable characteristics and low computational cost. The experimental studies compare the H-MOPSO against a state-of-art multi-objective algorithm MOEA/D-DRA. The results have shown that the H-MOPSO is competitive against the state-of-art algorithm and that the use of hyper-heuristic is capable of increasing the MOPSO performance for Many-Objective Optimization. No other work uses Hyper-heuristic for selecting MOPSO operators for Many-objective optimization.

In [26], a new Adaptive Operator Selection (AOS) for MOEA/D has been proposed: the Fitness-Rate-Rank-based Multi-Armed Bandit (FRRMAB). The objective is to select the operators based on their recent performances. The results indicate that the AOS with FRRMAB can significantly increase the performance of an MOEA/D. In [10], two MOEA/D-FRRMAB variations are proposed based on Multi-Armed Bandit literature: MOEA/D-UCB-Tuned and MOEA/D-UCB-V. Those proposed methods use the variance of the operators' reward to get a better evaluation of its performance. In that work, the MOEA/D-UCB-Tuned gets favorable results when compared with the other two MAB-based methods and other two state-of-art Adaptive Operator Selection MOEA/D.

Another heuristic selection method is the Choice Function (CF) [24]. In [12], a CF variation is proposed. The main difference is the lack of the pair component. The objective was to combine three multi-objective evolutionary algorithms to solve multi-objective problems. The high-level approach demonstrated effectiveness when compared with each low-level heuristic applied individually and an adaptive algorithm. In [11] a CF variation that also does not use the pair component is proposed, based on [12]. Also, the elapsed time is not counted in seconds but how many heuristic selections have passed since the last time that the heuristic has been applied. In that way, a heuristic that takes more time but improves the quality of the solution has higher probability of being selected than a heuristic that takes less time but does not increase much the quality of the solution. Another contribution of that work is a new measure to evaluate the quality of generated solutions in multi-objective problems. The measure uses the relation of dominance between the generated solutions (or solution) with their parents to compute the quality. The best value is reached when all offspring dominate all parents, and the worst value occurs when all parents dominate all offspring. In this work, the CF proposed by [11] is referred as Simplified Choice Function (SCF). It is used in a Hyper-heuristic framework for a Search-Based Software Engineering problem. That framework also uses a Multi-Armed Bandit heuristic selection method.

In [9], an Adaptive Choice Function (ACF) is proposed, to select mutation methods in an MOEA/D framework. The ACF is proposed based on another CF variation called Modified Choice Function (MCF). In MCF, the intensification and exploration parameters are variable along the search. The intensification parameter is set to the maximum value every time that a heuristic improves a solution and the value gradually decrease if the heuristic does not improves the solution. The exploration parameter has an opposite behavior, being set to a minimum value when the heuristic improves and increasing otherwise. In ACF, a scale factor is added, to balance the measure of the heuristics reward and the elapsed time, measured in seconds. Another difference is that the quality of a heuristic is the mean of the previous rewards instead of the accumulated value as in MCF.

The proposal for this work is to evaluate the performance of other heuristic selection methods on the framework proposed by [6, 22]. The objective is to assess if the use of different methods increases the performance obtained by H-MOPSO. Two types of methods were elected to be studied: one based on Multi-Armed Bandit (MAB) and other based on Choice Function. From MAB literature, the method FRRMAB [26] and the variations proposed in [10] were selected. From CF literature, the Simplified Choice Function proposed by [11] and the Adaptive Choice Function proposed by [9] were chosen.

CHAPTER 3

MULTI-OBJECTIVE OPTIMIZATION

In this chapter, some theoretical foundations needed for a proper understanding of the remainder of this work are presented. First, the Multi-Objective Optimization (MOO) is described along with relevant definitions and difficulties. Then, the Many-Objective Optimization concept is explained. Also, the Multi-Objective Particle Swarm Optimization is described.

Multi-Objective Problems (MOP) exist in many real-world applications. MOP, are problems with more than one objective function, and its optimization is called Multi-Objective Optimization (MOO). In an MOP the solution may not be unique, due to objective conflicts, one solution may be better in one objective and other solution be better in another one, being then not comparable [4]. A problem presents conflicted objectives if the optimization of one objective degrades the optimization of the other.

Usually, in MOO, the concept of Pareto dominance is used. According to this concept if a solution j is better than i in one objective and not worse in any other, the solution i is dominated by j, and j dominates i ($j \prec i$) [4]. A solution i is optimal if there is not any other solutions that dominate it. The set of all optimal solutions is called Pareto set, and its image in the objective space is called Pareto front [4].

The purpose of the MOO is to find an approximation set to the Pareto Front (PF). The goals are to find an approximation (I) as close as possible to the PF, and (II) as diverse as possible in the objective space. Based on these goals the quality of an approximation set can be evaluated regarding two aspects: convergence, that measures how close the approximation set is to the PF in the objective space, and diversity, that measures how well distributed the approximation set is over the PF. Moreover, the decision about the best solution is left to a decision maker, which decides based on his/her preferences.

In Figure 3.1 a representation for a Multi-Objective Problem is illustrated. On the left side, the decision space or the space of the solutions of the problem is represented. The feasible decision space is represented by the gray area. A feasible solution is represented by $\bullet$. The objective space (right side) is the image of the decision space and represents the objective values of the solutions. The dominated solutions are represented by $\circ$, and the non-dominated solutions are represented by $\square$.

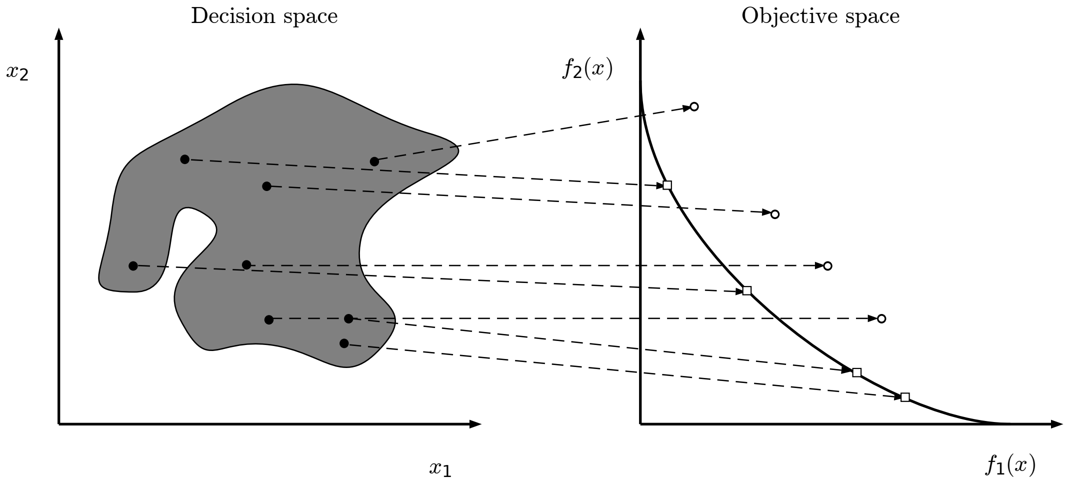


Figure 3.1: Multi-objective problem representation

Population-based meta-heuristics are widely used for solving multi-objective problems, they allow to output a Pareto approximation in a single run. Besides, they are less sensitive to different MOP characteristics than traditional Operational Research techniques [3]. Nowadays several problems with two or three objectives can be successfully solved by the state-of-art algorithms such as Speed-constrained Multi-objective PSO (SMPSO) [18] and Non-dominated Sorting Genetic Algorithm II (NSGA-II) [16]. The Many-Objective Optimization (MaOO) is a sub-area that focuses on solving problems with more than three objectives, which occur in several real world problems. The multi-objective techniques have their search abilities degraded as the number of objectives increases [5]. It happens because of [4]:

- Difficulty of convergence, as the result of increasing the portion of non-dominated solutions, known as dominance resistance (DR) phenomenon. The convergence of Pareto-based algorithms is affected since most solutions became not comparable, as the result the search became random, losing convergence pressure. Another problem is related to the decision of which solutions keep and which discard because the newly generated solutions tend to be non-dominated too.

- Difficulty of diversity, it is the difficulty of finding a solution set that represents the characteristics of the true front. Due to the increase of the Pareto Front, a higher solution set size is needed to describe such front.

- Difficulty of visualization of the solutions, since special techniques are required to represent the solution set in the objective space.

Because of these challenges, there is an increasing number of works in past few years focusing MaOO [21, 31, 7, 8]. In [4] the multi-objective approaches are categorized into seven classes. The relaxed dominance based: that relaxes the dominance concept to enhance the convergence pressure, however, these techniques can make difficult the diversity maintenance. The diversity-based that uses customized diversity maintenance strategies to improve the performance. The aggregation-based that uses scalarizing functions and a set of weighting vectors to rank the solutions. In that approach, the selection of the scalarizing function and the weighting vectors could affect the algorithm performance. The indicator-based guides the search using the indicator value of the solution set. The Hypervolume Indicator has been widely used due to its consistency with the concept of Pareto dominance. However, the high computational cost to compute the hypervolume has encouraged the study of other indicators, such as the R2 [4]. The reference set techniques uses a set of reference solutions to evaluate the solutions and guide the search. In this approach, the choice of how to manage and how to use the reference set to assess the solutions quality could affect the algorithm performance [4]. The preference-based approaches tries to reduce the complexity of the problem, focusing on a subregion of the Pareto front. That subregion is selected based on the decision maker preferences. The dimensionality reduction approaches reduces the number of objectives to change the problem to another one with fewer objectives, easier to be optimized. However, that approach can lose some information, as a consequence of the reduction.

Despite the increasing number of works, there are still many open problems and improvements to be made. Such as the design of diversity maintaining methods for aggregation approaches; or the design of hybrid algorithms, combining different approaches [4].

3.1 Multi-objective Particle Swarm Optimization

This section presents the Particle Swarm Optimization (PSO) algorithm and its adaptation for multi-objective optimization, the Multi-Objective Particle Swarm Optimization (MOPSO). The Particle Swarm Optimization is an algorithm inspired by the behavior of flocks of birds searching for food [32]. The PSO shares with the MOEA's (Multi-objective Evolutionary Algorithms) the concept of population, where each individual (in PSO is called a particle) is a complete solution. The particles (solutions) move around the search space using a simple velocity operator, composed of three components:

- **Individual :** the individual, or local, component represents the best solution found by the particle. It computes the difference between the current position x and the best position visited by the particle $pBest$; This component is responsible for exploring the region of the best solutions discovered so far.

- **Social:** The social, or global, component represents the best solutions found by any particle. It computes the difference between the current position x and the best position discovered by any solution $gBest$. This component is responsible for guiding the swarm towards the best solutions discovered so far;

- **Inertia :** the inertia component is a scale factor for how much from the previous velocity will be considered to compute the new one. It is responsible for keeping the particle moving towards the same direction that previously. That component controls the swarm behavior, higher ω values will increase the search space exploration; on the other hand, lower values will increase exploitation. To guarantee an initial high exploration and, as the search goes on, increasing exploitation a variable ω value can be used, starting with a high value and then decreasing it;

Each particle is composed of its position and velocity. The position of the particle in the search space represents a solution, in other words, a vector of decision variables.

The velocity v represents the size of particles step in the search space. v is computed as the sum of the previous velocity with the social and individual components; then it is scaled by an inertia factor ω. Both individual and social components are scaled by a fixed factor (c_1 for individual and c_2 for social) and a random factor (φ_1 for individual and φ_2 for social components). Each particle computes its new velocity and then updates its position as the sum of its previous position and its velocity, according to Equations 3.1 and 3.2.

$$v = \overbrace{\omega v}^{\text{inertia}} + \overbrace{c_1\varphi_1(pBest - x)}^{\text{individual component}} + \overbrace{c_2\varphi_2(gBest - x)}^{\text{social component}} \tag{3.1}$$

The new position x is computed as the sum of the previous position and the new velocity. Then, the particle moves to the new position $x + v$.

$$x = x + v \tag{3.2}$$

Initially, the position and velocity of the particles are initialized. Then, until the stop criteria is reached the algorithm does the following steps: first the particles fitness is evaluated, and the individual and social components are updated. Then, the new velocity is computed and the position updated. This iterative and cooperative swarm behavior moves the search towards the best solutions of the search space.

Based on the PSO, a multi-objective version was proposed, the Multi-objective Particle Swarm Optimization (MOPSO) [33]. The population-based characteristic allows the algorithm to output in a single run an approximation of the Pareto front. To work with multi-objectives, some issues need to be considered: the selection of the global best $gBest$ is not trivial as for one objective since, in MOO, the algorithm may generate not-comparable solutions. Another change is related to the storage of the best solutions, and also the diversity maintenance. Some changes on original mono-objective PSO were proposed to deal with those issues. The global best $gBest$ solution is selected from a set of non-dominated solutions (called leaders), using some leader selection rule. The non-dominated solutions are stored in a repository (also called archive, usually based on

MOEA's archive) to be selected further; they are also the algorithm output. As the size of this repository may be limited, some rule needs to be applied to select which solution should be kept and which should be discarded when the number of non-dominated solutions is higher than the repository capacity; The convergence of each particle depends on the leader selection rule since the particles may select different leaders and converge to different regions of the search space.

As shown in Algorithm 1, the main structure of the PSO does not change, but two steps are included: the leader selection and the archiving (update repository) methods. The MOPSO initially sets the position and velocity of the particles (same way as PSO), evaluates the fitness of the particles for each objective, and initializes the repository with the non-dominated solutions. Then, until the stop criterion is reached, the algorithm updates the velocity and position of each particle, using a leader selection rule to select the global best component for each particle. After the update of the position, some implementations may apply a "fly turbulence", similar to a mutation method, to increase diversity. Finally, the repository is updated with the non-dominated solutions, using some archiving rule to truncate the repository when the number of non-dominated solutions exceeds the repository capacity. When the algorithm finishes the repository of non-dominated solutions (leaders) is the algorithm output.

Algorithm 3.1: MOPSO

```
initialize the swarm;
evaluate the particles;
initialize the repository;
while not reached the stop criteria do
    foreach particle ∈ swarm do
        leader selection; // Section 3.1.2
        update velocity; // Equation 3.1
        update position; // Equation 3.2
        turbulence;
        fitness evaluation;
        update pBest;
    end
    update repository (swarm); // Section 3.1.1
end
return repository;
```

The use of leader selection and archiving methods for adaptation of the PSO was able to produce good results for multi-objective problems. Moreover, it was the focus of several works [21, 22, 18, 33]. However, the different leader selection and archiving rules show different behavior (and quality) depending on the problem and the stage of the search.

3.1.1 Archiving methods

Usually, MOEA and MOPSO implementations use a repository to keep the non-dominated solutions. Moreover, in general, the archiving methods employed in MOPSO were initially proposed to MOEAs. A repository with limited capacity can be used to deal with that problem, using a truncation strategy to filter the solutions. Usually, a precise archiving is used, where each newly generated solution is compared against all solutions inside the repository. If the new solution is dominated by some other, it is discarded. If the new solution dominates one or more solutions in the repository, the new solution is inserted and the solutions dominated are removed. If the new solution and those from the repository are all non-dominated, and the repository is not full, the new solution is inserted. Finally, if all solutions are non-dominated, and the repository is full then it is applied a method to filter the repository and to decide if the new solution will be included and, if included, which one will be removed. Some archiving methods are described below:

- **Crowding Distance (CD)**:

  The Crowding Distance archive, proposed by [16] for the NSGA-II algorithm, aims for Diversity Preservation. The *crowding distance* value measures the perimeter of the hypercube surrounding a solution using the nearest neighbors as vertices. A solution in a region with a higher *crowding distance* value (low density of solutions) is preferable. Then, when the repository capability exceeds the solution with the lowest *crowding distance* value is removed.

  To compute the *crowding distance* value, the sorting of the population is required. Also, the objectives are normalized before computation. The boundary solutions have an infinity value assigned to them. To the others, the CD value is the sum of the distance between two adjacent solutions for each objective. As shown in Figure 3.2, the CD value of a solution (□) is the sum of the distance between two adjacent solutions for each objective (dashed lines).

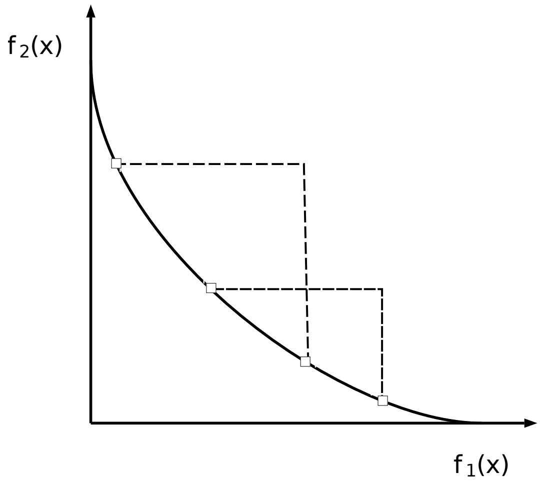


Figure 3.2: Example of Crowding distance computation for four particles

- **Ideal Archiving**: This method aims to improve the convergence of the algorithm. If the archive is full, an ideal point is computed. The ideal point is composed of the best objective value discovered so far, for each objective. Then, the distance for each solution to the ideal point is computed. Finally, the farthest solution is removed [17]. The Figure 3.3 shows an example of it.

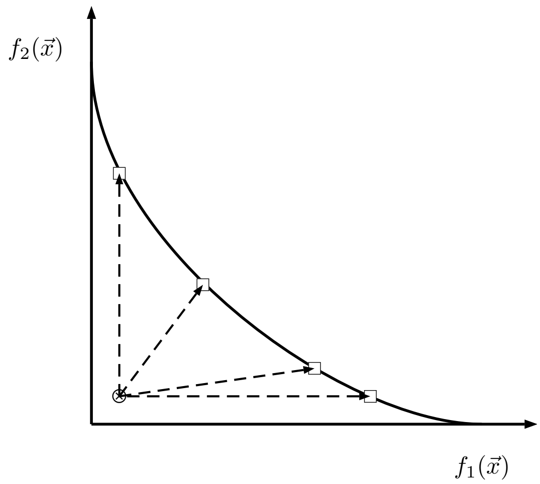


Figure 3.3: Example of distance to Ideal point computation for four particles

- **Multilevel Grid Archiving (MGA)**: When the repository is full, the objective space is divided into a grid, and each solution has a box index vector associated. Then the dominance relation of the box index vectors is evaluated. If the box index vector of the new solution is weakly dominated, the new solution is not accepted. Otherwise, the new solution is included, and a solution from a weakly dominated box index vector is removed randomly. If there are no weakly dominated box index vectors, the objective space is divided into smaller boxes until a dominated box index vector is found [14, 15]. In Figure 3.4 an example is illustrated. Initially, the objective space is divided into boxes. Since no box index vector is weakly dominated by any other, the objective space is divided into smaller boxes. Finally, a set of weakly dominated box index vectors is found, and an arbitrary solution from this set is removed.

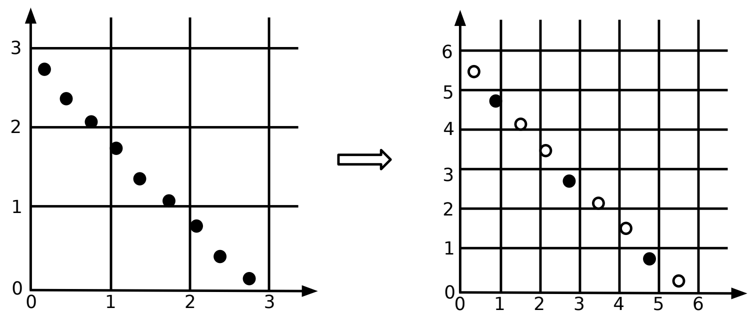


Figure 3.4: Example of Multi-Level Grid archiving

Different archiving methods have different behaviors and objectives. For example, Ideal Archiving aims at convergence, while CD and MGA aims at diversity. Due to it, different methods have different qualities depending on the problem, and the stage of the search. Meanwhile, the number of non-dominated solutions may increase quickly, specially for many-objective problems [4]. The high number of non-dominated solutions may increase the computational cost to keep and maintain the repository.

3.1.2 Leader selection methods

The leader selection step is an important part in an MOPSO. It is responsible for guiding the search and providing diversity. However, it is not a trivial task, since all solutions inside the repository are non-dominated and, from a dominance relationship point of view, all equally good. Some leader selection techniques are presented below [22]:

- **Crowding Distance (CD)** [18] : The CD method aims at providing diversity, trying to guide the search towards the less crowded areas. The crowding distance of each leader candidate is computed as the sum of the distance between two adjacent solutions for each objective. In this method, two solutions are randomly selected from the repository, and then the one with the highest crowding distance value is selected.

- **NWSum** [19]: The NWSum is a Weighted Sum (WSum) variant, but in NWSum, the particle with the highest weighted sum value is selected instead of the smallest value used in WSum. This method aims at providing a good spread of solutions. The NWSum is found to be more efficient than the original WSum.

- **Sigma** [20]: In this method, for each possible candidate leader, a Sigma vector (a vector from the origin to the leader candidate) is computed. Then the selected leader is that one that generates a Sigma vector closer to the Sigma vector of the particle that will be updated. In this method, the particle selects a leader that is similar to it.

As the archiving methods, the leader selection methods may vary their performance depending on the problem and the stage of the search. Previous works [6, 22] have shown that the choice of leader selection and archiving methods may increase the convergence and diversity of the algorithm considerably.

3.1.3 Speed-constrained Multi-objective PSO

Due to its good results in the literature the Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO) is the MOPSO used in [6] and in this work. In [18], the SMPSO was compared with five multi-objective meta-heuristics and was able to obtain remarkable results regarding the quality of the approximation set and the convergence speed. In that work, the SMPSO was the best or second best algorithm in all cases for the Epsilon indicator, in ten of twelve for Spread and in nine of twelve for Hypervolume. The SMPSO is based on the OMOPSO algorithm. The main difference between SMPSO and OMOPSO is the strategy to limit the velocity of the particles. It avoids the velocity to become too high and allows producing new effective particle positions inside the feasible search space. In addition to the velocity constriction mechanism, the original SMPSO uses polynomial mutation, Crowding Distance Archiver and Crowding Distance Leader Selection. In the velocity constriction mechanism, first, the velocity is computed, as it is in the original OMOPSO; then the velocity is multiplied by a constriction factor, and finally it is constrained to a maximum velocity value *delta*.

Based on the SMPSO algorithm, Castro and Pozo [22] proposed the H-MOPSO. A Hyper-heuristic Multi-Objective Particle Swarm Optimization that automatically selects leader and archiving methods for an SMPSO algorithm. In this work, the study and evaluation of other heuristic selection methods is proposed and compared to the roulette-based used in [22]. The concept of Hyper-heuristic is detailed in the next chapter.

CHAPTER 4

HYPER-HEURISTIC

In many cases, there is a set of algorithms that can be used to optimize a given problem. In that situation there is a challenge: the selection of the best algorithm or combination of algorithms, and the algorithm configuration to be used. That challenge becomes difficult with the increase of possibilities and the lack of guidance on how to select an appropriate algorithm. Also, there is a theorem, called No Free Lunch Theorem [34], that says: "If all optimization algorithms were evaluated in all optimization problems, in average, they all would have the same performance". It means that the algorithms have different performances depending on the problem, for instance: if an algorithm has excellent performance in a given problem, there is another problem where it will perform poorly.

The Hyper-Heuristic (HH) techniques emerge as high level approaches intended to select (or generate) algorithms, based on a pool of algorithms or parts of it. In other words, HH is high-level techniques that, for a given problem instance, and a set of heuristics (or parts of heuristics), produces a proper combination for the problem [1]. In that way, hyper-heuristic operate over the heuristic space, rather than the solution space. The term was first defined in 2000 [24]. However, it is possible to find related work on the topic since the sixties, for instance, the work described in [35] that combines a set of rules for a scheduling problem. In [35], it is concluded that a random combination of rules was best than any rule applied individually, and it is commented that the use of learning could be used. That work [35] and many other ones related with hyper-heuristic were applied for scheduling problems [1].

It is possible to classify the hyper-heuristic approaches in two classes: generation and selection techniques, that can be used together [1]. In Figure 4.1 a classification scheme is shown, it is possible to classify the hyper-heuristic by the feedback source: *Online*, the learning occurs during the optimization process; *Offline*, the learning is made in a training set of problem instances, the output is an algorithm trained to optimize instances of the problem that was trained; the no-learning techniques do not use any information about the heuristic performance, for example, a simple random heuristic selection. Hyper-heuristic approaches can also be classified by the nature of the search space: the selection or generation of construction heuristics, that incrementally generate a solution; Alternatively, perturbation heuristics, that operate over a complete solution.

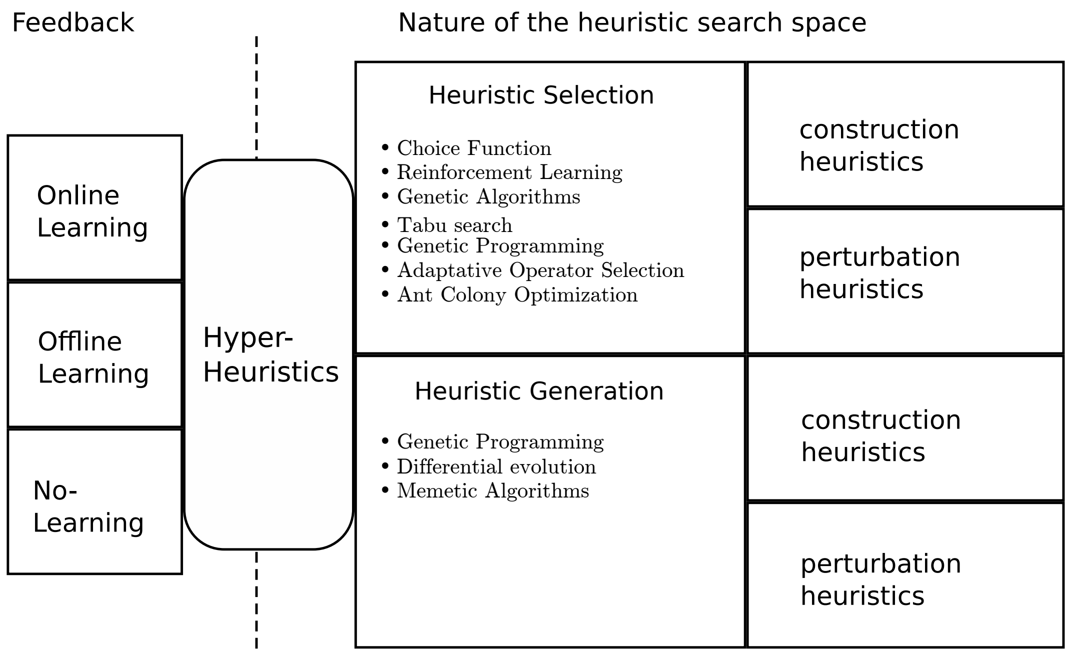


Figure 4.1: Hyper-heuristic classification scheme and examples (Adapted from [1] and [2])

According to [1] the heuristic generation objective is to build complete heuristics from a set of components. The heuristic generation can have a high computational cost. However, the generation of heuristics returns a new heuristic, specialized for the problem instance which was designed for and other instances of the problem. Despite the computational cost, a heuristic generation is usually less time consuming than a manual heuristic design. Additionally, a specialized heuristic is usually better than a general one regarding the solution optimality. In heuristic generation literature, most works use Genetic Programming [1]. The heuristic generation is not the focus of this work. The focus of this work is the heuristic selection.

The heuristic selection methods are techniques that, for a given problem instance and a pool of heuristics, or heuristic components, is responsible for selecting an appropriated heuristic to be applied. The objective is to get the best heuristic to be applied depending on the search stage and the search space properties. Those techniques aims at handling the Exploration *vs* Exploitation Dilemma (EvE). According to EvE, just as it is important to apply more often operators (or heuristics) with better performance, it is also important to employ those that have not been employed recently, to evaluate its current performance. In heuristic selection hyper-heuristic, a domain barrier may be considered, to keep separated the heuristic selection from the problem properties. The objective of this barrier is to allow the hyper-heuristic to be problem independent. The low-level heuristics are responsible for dealing with the problem. This domain independence allows the heuristic selection to be used in other domains, by changing the low-level heuristic pool. In 2011, a competition was created with the objective of encouraging the research on the hyper-heuristic field, mainly the Cross-domain: the Cross-Domain Heuristic Search Challenge(CHeSH) [36]. In the competition, many hyper-heuristic were evaluated in different combinatorial optimization problems.

According to [1] the selection of construction heuristics has been used for the following problem domains: production scheduling; educational timetabling; 1D packing; 2D cutting and packing; workforce scheduling; constraint satisfaction; vehicle routing. Using in most the following selection methods: Variable Neighborhood Search; Tabu Search; Artificial Neural Networks; Evolutionary Algorithms; Genetic Algorithms and variations; Accuracy-based classifier system; Machine Learning and others. The selection of perturbation heuristics have been used for the following problems [1]: personal scheduling; educational timetabling; space allocation; cutting and packing; vehicle routing; and sports scheduling. There are also studies involving Cross-domain, it means, the application of a hyper-heuristic method independent of the problem domain. Most works have focused on discrete problems, and a few have used population-based low-level heuristics. In Figure 4.2 a generic heuristic selection framework is presented.

The Hyper-heuristic for perturbation heuristic selection usually works in two phases [1]:

1. **Heuristic selection**: In this phase, a low-level heuristic is chosen and applied.

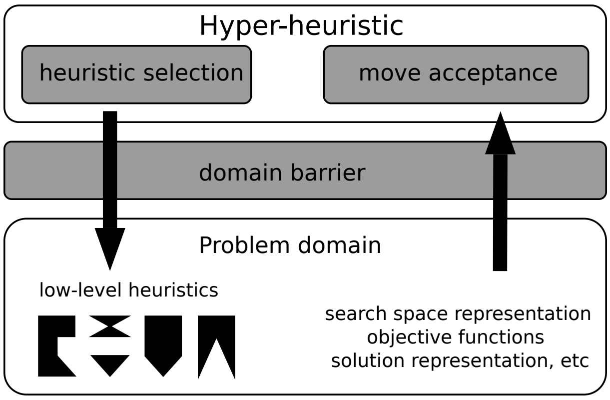


Figure 4.2: Generic scheme for heuristic selection

There are different heuristic selection methods, for instance: the Choice Function [12, 24]; Reinforcement learning; and Adaptive Operator Selection. Moreover, even simple methods are usually better than apply any low-level heuristic individually. Also, the quality of the optimization depends directly on the quality of the pool of low-level heuristics. The simplest are those with no learning, e.g. a random selection. There is also rank-based heuristic selection methods where the best-ranked heuristics are more likely to be selected. Those methods use an update rule to rank the low-level heuristics, based on its performance. Some of them may use a memory update that indicates how much the oldest performances will affect the selection.

2. **Move acceptance**: In this phase, the solution generated by the applied heuristic is evaluated, using some quality information received from the problem domain, usually the fitness value of the solution. Then the solution is accepted or rejected depending on its performance and the move acceptance method rules. Normally, if a solution is better than a previous one, it is accepted; otherwise, an acceptance criterion is used. The use of an appropriate move acceptance method may increase the optimization performance substantially. Some move acceptance methods are presented in Section 4.1.

Due to their good results in the literature, two heuristic selection methods were selected to be used: Choice Function [24] and Multi-armed Bandit [27]. They are described in Sections 4.2 and 4.3.

4.1 Move acceptance

According to Burke *et. al.*, the decision of the move acceptance seems to be more important than the heuristic selection. Some move acceptance methods are [1]:

- **All Moves**: The new solution always replaces the old one;
- **Only Improvements**: If the new solution is better than the old one, it replaces the old one.
- **Improving or Equal**: If the new solution is better or equal to the old one, it replaces the old one.
- **Monte Carlo**: Always accepts improving or equal solutions, otherwise an acceptance probability is applied;
- **Simulated Annealing**: The probability of acceptance is based on the search stage. In that way, the initial probability of accepting bad solutions is higher and then decreases as the search goes on. Improving or equal solutions are always accepted.
- **Threshold acceptance**: All improving or equal solutions are accepted; otherwise the new solution is compared with a threshold solution. The threshold solution starts with the best solution, if the threshold is not reached after a while, it is updated to the next best solution (second best, third best, and so on) until a new solution is accepted.
- **Great deluge**: Initially, all solutions are accepted, and then become more selective. In the end, only improving or equal solutions are accepted.
- **Late acceptance**: The method keeps a limited size list of solutions. Each new solution is compared against the oldest solution in the list. If it is better than the oldest one it is accepted, and inserted on the list. In case the list is full, the oldest solution is removed.

4.2 Choice Function

The Choice Function method is an *online* learning heuristic selection based on ranking, proposed by [24]. The ranking is made based on a function with three components [37]:

$$cf(h_i) = \alpha f_1(h_i) + \beta f_2(h_j, h_i) + \delta f_3(h_i) \tag{4.1}$$

1. The first component (f_1) is the fitness of the low-level heuristic h_i. It is responsible for increasing exploitation, increasing the probability of the heuristics with the best performance.

2. The second (f_2) is the fitness of the pair of low-level heuristics (h_i, h_j) when applied together. The objective is to find a cooperative behavior between heuristics that, when applied together, have a good performance.

3. The function also has an exploration component (f_3). The elapsed time since the last time that the low-level heuristic h_i was applied is computed. It increases the probability of the low-level heuristics not recently applied, to evaluate its current performance.

The function value $cf(h)$ of the heuristic h_i is the sum of the three components (f_1, f_2 and f_3), weighted by some scale factor parameters (α, β and δ). After updating the ranking, the choice can be made using different ways, such as: to get the best-ranked heuristic, or to use some simple roulette rule.

There is a Simplified Choice Function (SCF) version [12], that does not consider the pair component (Equation 4.2). In [11] this simplified version is used, and the exploration function ($f_3(h_i)$) is adapted to return how many operator applications have passed since the last time when the operator (h_i) was applied, instead of using the elapsed time in seconds as the original choice function.

$$cf(h_i) = \alpha f_1(h_i) + \delta f_3(h_i) \tag{4.2}$$

With the objective of avoiding parameter configuration of α, β and δ, in [23] an Adaptive Choice Function (ACF) is proposed:

$$cf(h_i) = \phi f_1(h_i) + \phi f_2(h_j, h_i) + \delta f_3(h_i) \tag{4.3}$$

Where ϕ is an exploitation factor for the best-performed heuristics, and δ is an exploration factor, responsible for increasing the probability of the heuristics that have not been applied recently. The parameter ϕ is set to 0.99 each time that the heuristic improved the solution quality and decreased by 0.01, otherwise. The δ parameter is set to $(1-\phi)$. The objective is to increase exploitation when the solution quality is improving, and to increase exploration when the current best operators cannot increase the solution quality. In [9] two modifications are proposed: the use of an scale factor (SF) parameter, since the measures used in f_1 and f_2 may be in different scales when compared with f_3; and the use of the mean values of f_1 and f_2, instead of the accumulated values.

In this work, the study of Simplified and Adaptive Choice Functions to select leader selection and archiving methods in MOPSO is proposed. For this, the H-MOPSO framework proposed in [6] is used.

4.3 Multi-armed Bandit

The Multi-armed Bandit is a problem that considers a set of K independent arms, with unknown probability of being rewarded. The objective is, along the time, selecting the arms that maximize the accumulated reward. The Multi-armed Bandit problem fits in the Exploitation vs. Exploration (EvE) dilemma. According to EvE, it is important to apply often the arm with the highest performance, but, it is also important to employ the other arms once a while to evaluate its current performance since its quality may change. Many algorithms have been proposed to tackle the MAB problem, one of them is the Upper Confidence Bound (UCB), which provides asymptotic optimality guarantees. In UCB, the selected arm is the one that maximizes the UCB function (Equation 4.4) [10, 26].

$$\hat{q}_i + C \times \sqrt{\frac{2 \times \ln \sum_K n_{K,it}}{n_{i,it}}} \tag{4.4}$$

Where each arm has an empirical quality estimated ($\hat{q}_i$). Same way as the f_1 component from Choice Function; the $\hat{q}_i$ is responsible for increasing exploitation, i.e., enhance the probability of the heuristics with the best performance. Moreover, a confidence interval computed based on the number of times that the arm has been tried before. The C parameter is a scale factor between exploration (the term right, that measures how frequently the arm has been tried) and exploitation (the term left, that measures the arm quality) [26, 10].

Some UCB-based algorithms have been proposed for Adaptive Operator Selection (AOS), as it aims at handling the EvE dilemma. The objective is to select the operators that increases the quality of the optimization solution output. One of the UCB-based algorithms is the Sliding Multi-Armed Bandit (SlMAB) [38]. In the SlMAB the empirical quality estimate (Equation 4.5) and the confidence interval (Equation 4.6) are computed based on a sliding time window, to evaluate the operators based on the current search stage.

$$\hat{q}_{i,it+1} = \hat{q}_{i,it} \times \frac{W}{W + (t)} + r_{i,it} \times \frac{1}{n_{i,it} + 1} \tag{4.5}$$

$$n_{i,it+1} = n_{i,it} \times \left(\frac{W}{W + (t)} + \frac{1}{n_{i,it} + 1} \right) \tag{4.6}$$

Where it is the current iteration. W is the sliding window size; it means that only the last W fitness improvements will be considered. t is the elapsed number of iterations since the previous time when the operator was applied; Moreover, r is the operator reward (it can be computed using some criteria, such as the fitness obtained in the last time that the operator was applied; or the accumulated fitness in the last W iterations). In a case of an operator be often applied the n value increases quickly. Otherwise, its n value increases slowly, increasing the probability of being selected.

In [38], it is also proposed a Rank-based Multi-Armed Bandit (RMAB). It also uses a sliding time window, but the $\hat{q}_i$ is computed based on a ranking of the operators rewards. Moreover, the n is the number of times that the operator has been applied in the last W iterations.

In [26], a Fitness-Rate-Rank-Based Multi-Armed Bandit (FRRMAB) is proposed. Usually, the gross value of the fitness improvements is used as the reward. However, the range of these values may vary depending on the problem and the search stage. To deal with that, FRRMAB uses a fitness improvement rate (FIR), defined as (Equation 4.7):

$$FIR = \frac{pf - cf}{pf} \tag{4.7}$$

Where the FIR of the operator is the difference between the fitness of the solution before (pf) and after (cf) applying the operator, divided by the old fitness (pf). The FIR values of the last W applications are stored in a sliding time window. Moreover, the $Reward$ of an operator (i) is the sum of all FIR of that operator in the sliding window. Then, all operators are ranked by reward, and a decaying factor (D) is applied, (Equation 4.9) to increase the probability of the best-ranked operators:

$$Reward_i = \sum_{k \leftarrow 0}^{W} \begin{cases} FIR_k^{op} & \text{if } op = i \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

$$Decay_i = D^{Rank_i} \times Reward_i \tag{4.9}$$

Finally, the credit value of the operator i is computed as (Equation 4.10):

$$FRR_i = \frac{Decay_i}{\sum_{j=1}^{K} Decay_j} \tag{4.10}$$

To select the operators, the empirical reward $\hat{q}_i$ is replaced by the FRR_i value in the UCB function. Moreover, n_i indicates the number of times that the operator (i) has been applied in the last W iterations. The FRRMAB uses the pure UCB function. The use of two other functions from UCB literature, the UCB-V, and UCB-Tuned is investigated in [10]. The classical UCB function and the UCB-V provide asymptotic optimality guarantees. Moreover, the UCB-V and UCB-Tuned use the reward's variance to obtain a better EvE trade off. In UCB-Tuned, the selected operator is the one that maximizes the following function: (Equation 4.11)

$$FRR_{op} + C \times \sqrt{\frac{\ln \sum_i^K n_i}{n_{op}} \times \min(\frac{1}{4}, V_{op})} \tag{4.11}$$

Where V_{op} is:

$$V_{op} = \sigma_{op}^2 + \sqrt{\frac{2 \times \ln \sum_i^K n_i}{n_{op}}} \tag{4.12}$$

In the UCB-V function, the selected operator is the one that maximizes the following function (Equation 4.13):

$$FRR_{op} + C \times \sqrt{\frac{2 \times \ln \sum_i^K n_i \times \sigma_{op}^2}{n_{op}}} + 3 \times \frac{\sum_i^K n_i}{n_{op}} \tag{4.13}$$

In this work, the study of the original FRRMAB(-UCB1) and its variants FRRMAB-UCB-V and FRRMAB-UCB-Tuned to select leader selection and archiving methods in MOPSO is proposed. For this, the H-MOPSO framework proposed in [6] is used.

CHAPTER 5

HYPER-HEURISTIC GUIDED MANY-OBJECTIVE PARTICLE SWARM OPTIMIZATION

In this chapter, the proposal for this work is presented. Initially, the motivation and objectives for this proposal are described. Then, the base framework for this work is introduced: the Hyper-heuristic Multi-Objective Particle Swarm Optimization (H-MOPSO), proposed by [6]. Also, the studies and improvements proposed for H-MOPSO and the methodology used are presented.

The motivation of this work is based on the fact that several real world problems are many-objective. Besides, the algorithms designed for multi-objective optimization have a deterioration of convergence and diversity when applied to problems with more than three objectives [4]. Some many-objective algorithms have been proposed to solve this kind of problems. One of those algorithms is the Hyper-heuristic Multi-objective Particle Swarm Optimization (H-MOPSO) [6]. That work is based on studies that show that different archiving and leader selection methods have different performance depending on the problem and the search stage [21, 17]. Those characteristics indicated that the use of hyper-heuristic to select automatically archiving and leader selection methods may increase the MOPSO performance in different circumstances, including many-objective optimization. The H-MOPSO proposed by [6] uses a simple roulette-based heuristic selection, which obtained encouraging results. The results obtained by [6] shows that the use of hyper-heuristic can improve MOPSO performance. Also, it is possible that the use of more accurate methods may improve the selection and consequently the obtained results. The aim of this work is to reduce the deterioration of convergence and diversity present on many-objective optimization. To do so hyper-heuristic can be employed, according to [6].

In this work it is proposed the study of other heuristic selection methods on the H-

MOPSO framework. It is also proposed the evaluation of a new H-MOPSO structure.

5.1 Hyper-heuristic Multi-objective Particle Swarm Optimization

The Hyper-heuristic Multi-objective Particle Swarm Optimization (H-MOPSO) is based on the Speed-constrained Multi-objective PSO (SMPSO). The H-MOPSO selects, for each iteration, a pair of archiving and leader selection methods to be applied. Then, an iteration of the SMPSO using the methods previously selected is executed. When the iteration finishes, the swarm is evaluated for acceptance for the next iteration. The quality of the generated swarm is used to assess the performance of the pair of leader selection and archiving methods selected.

H-MOPSO is illustrated in Algorithm 5.1. The highlighted steps are the hyper-heuristic steps introduced in SMPSO. First the swarm is initialized, and then the particles are evaluated and the repository is initialized with the non-dominated particles. Finally, before the main loop starts, the hyper-heuristic strategy is initialized. Until the stop criterion is reached the algorithm repeats the following steps: First, a low-level heuristic is selected (a combination of archiving and leader selection methods) using some heuristic selection method. Then, for each particle, the SMPSO steps are executed: first, the leader selection method is applied; then the particle velocity and position are updated; after updating the position, a turbulence method is applied (mutation); Finally, the fitness of the particle is evaluated, and the local best component ($pBest$) is updated. After computing those steps for each particle the algorithm updates the repository, using the selected archiving method; then the reward of the applied low-level heuristic is computed. Finally, the move acceptance method is applied, to verify if the new repository is going to be accepted (to replace the old one) or discarded (to keep using the old one). When the stop criterion is reached, the algorithm outputs the approximate Pareto front, *i.e.* the repository.

In [6], the low-level heuristic selection was made using a simple roulette based method,

Algorithm 5.1: H-MOPSO

```
initialize the swarm;
evaluate the particles;
initialize the repository;
initialize hyper-heuristic;
while not reached the stop criteria do
    heuristic selection;
    foreach particle ∈ swarm do
        leader selection; // Section 3.1.2
        update velocity; // Equation 3.1
        update position; // Equation 3.2
        turbulence; // mutation
        fitness evaluation;
        update pBest;
    end
    update repository (swarm); // Section 3.1.1
    credit assignment;
    move acceptance; // Section 4.1
end
return repository;
```

initially with same probability for all low-level heuristics. The selection was guided by the quality indicator R2 [39]. The R2 quality indicator is based on the weighted Tchebycheff function and allows evaluating both convergence and diversity with low computational cost. A weight vector set is needed to compute the R2 value, a set of uniform distributed weight vectors is frequently used. It also uses an ideal point; it means, a point that no feasible solution dominates. The use of the R2 quality indicator instead of Hypervolume is preferred. Since it has low computational cost, and prioritizes well-distributed fronts, the Hypervolume biases to the knee of the front. According to [39], the R2 indicator evaluates all desired aspects of a Pareto front approximation. The R2 indicator is weakly monotonic and less time consuming when compared with Hypervolume indicator. Besides, the R2 indicator is assumed to produce a more uniform distribution.

For each low-level heuristic applied, if the R2 value of the repository decreases, the roulette portion of the applied low-level heuristic increases. On the other hand, if the R2 value increases, the roulette portion decreases. The improving or equal acceptance method was used, it means, if the old repository has a worse or equal R2 value than the new one, it is replaced, and the new one begins to be used. Another decision made by [6] is related to performance estimation of the low-level heuristics. Unless the repository size limit is exceeded, every archiving method will have the same behavior: to accept all non-dominated solutions. So, at the initial stages, the quality of different archiving methods cannot be estimated. To avoid this noise on learning the default SMPSO is used until the repository size limit is exceeded (using Crowding Distance Archiver and Leader Selection); just then, the selection of the methods and the learning process begins.

In [6], nine low-level heuristics are used. Each low-level heuristic is a combination of three leader selection methods, with three archiving methods:

- **Archiving methods**: Crowding Distance Archiving (CD); Ideal Archiving; Multi-level Grid Archiving (MGA).
- **Leader selection methods**: Crowding Distance Leader Selection (CD); NWSum; Sigma.

The leader selection and archiving concepts have been described previously in Sections 3.1.2 and 3.1.1.

The turbulence method used was the polynomial mutation with probability $p_m = 1.0/L$, where L is the number of decision variables, as proposed in [18]. Initially, the probability of each low-level heuristic is the same. The credit assignment was fixed: if the R2 value of the new repository is better than the value obtained by the previous one, the probability of the low-level heuristic applied was incremented by 0.1 of the initial probability. Otherwise, it was decreased by the same value. A minimum probability of 0.5% was applied to guarantee that a low-level heuristic will not be removed of the roulette (it means, 0% of probability).

5.2 Proposal

In this work different heuristic selection methods are investigated in the hyper-heuristic strategy proposed by [22] with the focus on Many-objective Problems (MaOP). Also, the study of a new structure for the H-MOPSO framework is proposed. The objective is to evaluate the performance of the H-MOPSO using other heuristic selection methods and to compare to the roulette-based proposed by [6] and a state-of-art multi-objective algorithm. The study of two methods based on Choice Function is proposed: the Adaptive Choice Function [9] and the Simplified Choice Function [11]. Also, three Multi-armed Bandit based methods, the FFRMAB [26], the FFRMAB-UCB-V [10] and the FFRMAB-UCB-Tuned [10] are proposed. On the experiments, those heuristic selection methods replace the roulette-based proposed by [6] on the H-MOPSO framework. Those methods were selected based on Choice Function [22, 6, 11, 9, 12, 23] and Multi-Armed Bandit [26, 10, 22, 6, 11] literature.

Also, the study of a new H-MOPSO strategy is proposed. In this new strategy, the selection of the archiving and the leader selection methods are made separately. The archiving method still is selected for each iteration. However, the leader selection method is selected for each particle. The objective is to compare the selection made separately to the selection made together, as a single low-level heuristic. The aim of this new strategy is to have a heuristic selection method specifically for archiving and another for leader selection. In that way, the learning and the selection will be based only on the estimated quality of the specific method. Also, the selection of the leader selection method is made for each particle, which allows the algorithm to use different methods in a single iteration. The proposed new strategy is illustrated by the Algorithm 5.2

On that new structure, the archiving method remains being selected iteratively. Consequently, the credit assignment can be based on the quality of the repository. Therefore, as in [6] the R2 indicator can be used. However, the leader selection method is selected for each particle, so, the credit assignment should evaluate the quality of the updated particle. In this work, a credit assignment rule based on [11] is used, that proposes an equation based on the dominance relation between parents and offspring to evaluate the reward of crossover operators. In this work, it is used to compare the dominance of the new position of the particle with its previous position and with the leader selected from the archive (Equation 5.1). However, other credit assignment rule could be used.

Algorithm 5.2: H-MOPSO-II

```
initialize the swarm;
evaluate the particles;
initialize the repository;
initialize hyper-heuristic for leader selection;
initialize hyper-heuristic for archiving methods;
while not reached the stop criteria do
    foreach particle ∈ swarm do
        selection of the leader selection method;
        leader selection;
        update velocity;
        update position;
        turbulence;
        fitness evaluation;
        update pBest;
        credit assignment for leader selection;
        move acceptance for leader selection;
    end
    selection of the archiving method;
    update repository (swarm);
    credit assignment for archiving;
    move acceptance for archiving;
end
return repository;
```

$$r = \left(\begin{cases} 1.0 & \text{if } x_{new} \prec x_{old} \\ 0.0 & \text{if } x_{old} \prec x_{new} \\ 0.5 & \text{otherwise} \end{cases} + \begin{cases} 1.0 & \text{if } x_{new} \prec gBest \\ 0.0 & \text{if } gBest \prec x_{new} \\ 0.5 & \text{otherwise} \end{cases} \right) - 1 \tag{5.1}$$

The reward r is the sum of two components, the first evaluates the dominance relation between the previous position x_{old} with the new position x_{new}. If the new position dominates the previous position then the first component value is 1.0, if the previous dominates the new position, the value is 0.0, and 0.5 if both are non-dominated. The second component evaluates the dominance between the new position and the selected leader $gBest$. The value of the second component is 1.0 if the new position dominates the leader $gBest$, it is 0.0 if $gBest$ dominates the new position and it is 0.5 if they are both non-dominated. The sum of the two components are in the range $[0, 2]$, as it is desirable to represent good performances with positive values and negative, otherwise, it is subtracted one to set the range $[-1, 1]$. The reward r is used to compute the quality of a low-level heuristic, i.e., used to calculate the f_1 component for Choice Function and to calculate the FRR for FRRMAB. In other words, it replaces the FIR for the credit assignment of leader selection methods.

5.3 Methodology

In this section, the methodology used and the proposed experiments are presented. In this work several experiments are proposed. Initially, a preliminary study based on [6] was made. In that experiment, the original H-MOPSO algorithm was evaluated. Then, different Choice Function based heuristic selection methods were studied: the Adaptive Choice Function and the Simplified Choice Function. After that, the parameters of the selected CF are empirically configured. Then, the study of Multi-Armed Bandit (MAB) based heuristic selection methods is performed. Moreover, the parameters of the selected MAB-based are empirically set. Finally, the best MAB and CF based methods are compared to each other, to the roulette-based method proposed by [6], to a simple random heuristic selection, and to a state-of-art algorithm, MOEA/D-DRA. In this step, the selection was made only for the archiving method.

Then, three H-MOPSO strategies were evaluated: the selection of only the archiving method; the selection of both archiving and leader selection methods as a single low-level heuristic; and the selection of both methods separately. Those strategies were evaluated for each heuristic selection method. Moreover, for each heuristic selection method an H-MOPSO strategy was chosen. Finally, each heuristic selection method with the selected strategy is compared to each other and the MOEA/D-DRA algorithm. To conclude the experiments, the best H-MOPSO variant is compared with the MOEA/D-DRA.

The H-MOPSO and the low-level heuristics are implemented using the jMetal framework [40]. Except by the first study (Section 6.1), that replicates the related work [6], all other studies follow the setup illustrated in Table 5.1. The experimental setup is based on [31] and [7].

The benchmark problems used were the DTLZ1 to DTLZ4 functions from DTLZ test

Table 5.1: Experimental studies configuration

Parameter	**value**	**description**
m	3, 5, 8, 10 and 15	objective number
problem	DTLZ1 to 4 and WFG6 and 7	benchmark problems
popsize	91, 210, 156, 275 and 135	population size
repsize	91, 210, 156, 275 and 135	repository size
runs	20	independent runs

suite and WFG6 and WFG7 functions from WFG suite. Those benchmark problems allows the configuration of the number of objectives and the number of decision variables. For each objective number, the decision variables setup was used as suggested in [41] and [42]. For the DTLZ suite the number of decision variables is suggested by [41] as $n = m + r - 1$, where m is the number of objectives, r is set to 5 in DTLZ1, and 10 in DTLZ2 to DTLZ4. For the WFG suite the suggestion for the number of decision variables is $n = l + k$, where l is set to 20 and k is set to 4 if the number of objectives is 2, and k is set to $2 \times m - 1$ if the number of objectives m is higher than 2 [42].

The experiments were evaluated with five values for the objective number $m = \{3, 5, 8, 10, 15\}$. The population size, *popsize*, and repository size, *repsize*, use the same value and were set according to the number of objectives, 91, 210, 156, 275, 135, respectively to 3, 5, 8, 10, and 15 objectives. The number of independent runs were set to 20. The number of iterations was set for each test instance, according to Table 5.2.

Table 5.2: Number of iterations for different test instances

	Number of objectives (m)				
Problem	**3**	**5**	**8**	**10**	**15**
DTLZ1	400	600	750	1000	1500
DTLZ2	250	350	500	750	1000
DTLZ3	1000	1000	1000	1500	2000
DTLZ4	600	1000	1250	2000	3000
WFG6	3000	3000	3000	3000	3000
WFG7	3000	3000	3000	3000	3000

For each experiment and quality indicator, the average indicator value of the 20 independent runs is presented, for each algorithm, benchmark problem, and objective number. On the presented results, for each row (test instance), the best indicator value is highlighted with a bold face. It is presented with the gray background if it is the best value and has a statistical difference according to the Kruskal-Wallis statistical test [43], for each quality indicator. The average ranking of the algorithms is also computed. The best-ranked algorithm is highlighted with a bold face. It is presented with the gray background if it is the best ranked with a statistical difference according to the Friedman statistical test [44].

To evaluate the proposed studies the following quality indicators were used:

- **Hypervolume [45]**: The Hypervolume quality indicator computes the multi-dimensional volume of the objective space that is dominated by the Pareto front approximation. It is used to measure both, convergence and diversity and its computation does not require the knowledge of the true Pareto front. The Hypervolume is strictly monotonic; it means if a Pareto front A dominates another front B, the Hypervolume value of the front A will be better (higher) than B. Although, it is sensitive to the number of objectives and extreme values. Its main weakness is its high computational cost; that increases with the number of objectives. Due to this, in this work the Hypervolume quality indicator was not executed for $m = 15$.

- **Inverted Generational Distance (IGD) [46]**: The IGD quality indicator evaluates both convergence and diversity. It computes the distance between the true Pareto front and the Pareto front approximation. However, some Pareto fronts have an enormous number of solutions, sometimes infinity. To compute the IGD of such front, a representative set of reference points can be used. Then, the IGD can be computed as the average of the distance between each reference point to the closest solution to the Pareto front approximation. In this work, a set of reference points was used to represent the true Pareto front.

According to [47], the Hypervolume and IGD may disagree when the Pareto front is concave, which is the case of most of the problems evaluated. It may occur because the Hypervolume is biased to the knee of the front while IGD may favor a more uniform distribution. In Figures 5.1 and 5.2 are shown two examples of Pareto front approximations. The first front (Figure 5.1) has a better Hypervolume value, due to having a better convergence towards the knee of the front. The second front (Figure 5.2) has a better distribution over the true Pareto front, and due to it, has a better IGD value.

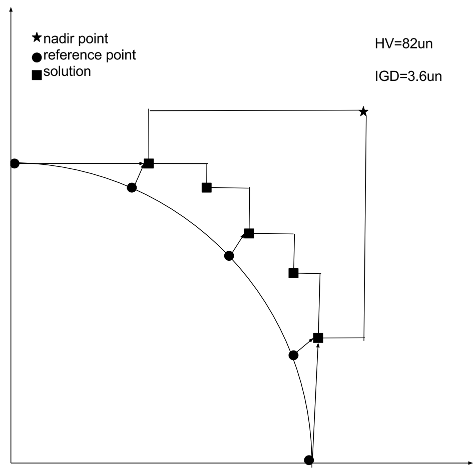


Figure 5.1: Example of Pareto front approximation with better Hypervolume and worse IGD than the front from Figure 5.2

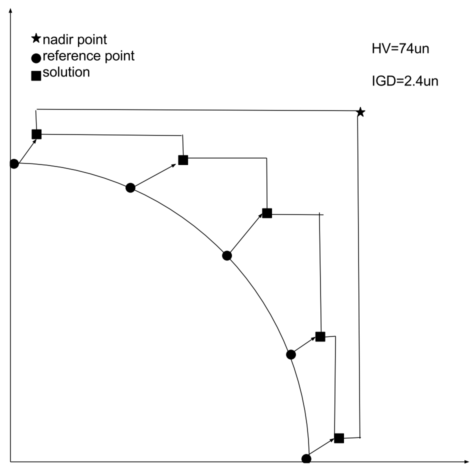


Figure 5.2: Example of Pareto front approximation with better IGD and worse Hypervolume than the front from Figure 5.1

Besides the decision of how to perform the experiments and analysis, some decisions about the H-MOPSO framework were made, based on literature. In this work the credit assignment used on H-MOPSO is based on the fitness improvement rate (FIR) proposed in [26]. The FIR is used as the reward to compute the f_1 component from Choice Function and the FRR component from FRRMAB. Usually, the credit assignment is based on the raw fitness value, but it can vary for each problem, and depends on the search stage. To alleviate this issue, the FIR computes the credit assignment of a low-level heuristic as the difference between the previous fitness value and the current one, divided by the previous value (Equation 4.7 of Section 4.3). In this work the FIR is computed as the difference between the R2 value of the previous repository and the R2 value of the current repository, divided by the previous R2 value:

$$FIR = \frac{R2_{old} - R2_{new}}{R2_{old}} \tag{5.2}$$

In this work, the same move acceptance method as [6] is used, the Improving or Equal (IE) move acceptance. If the R2 value of the current repository is better or equal than the R2 value of the previous repository then, the previous repository is discarded, and the new repository is used in the next iteration. Otherwise, the new repository is discarded, and the previous repository is used again. In some studies performed [6], the H-MOPSO is compared to a state-of-art algorithm: the MOEA/D-DRA. The MOEA/D-DRA is a multi-objective evolutionary algorithm based on decomposition with dynamic resource allocation and winner of the CEC 2009 MOEA contest [28].

CHAPTER 6

EXPERIMENTAL EVALUATION

In this section the experimental analysis is presented. First, a preliminary research was made. That initial review partially reproduces the main related work. Then, a study of different Choice Function based methods was performed. In that study, the Adaptive Choice Function and the Simplified Choice Function were compared. After that, the Adaptive Choice Function was selected and then its parameters were empirically configured. Also, three Multi-Armed Bandit based methods were evaluated: the FRRMAB-UCB1, the FRRMAB-UCBV and the FRRMAB-UCB-Tuned. The FRRMAB-UCB1 was selected, and its parameter configured empirically. The selected CF and MAB-based methods were evaluated in the selection of MOPSO archiving methods, and compared to a random selection, a roulette-based selection and a state-of-art multi-objective metaheuristic.

Also, a new H-MOPSO strategy was investigated. The new approach was evaluated using the selected CF and MAB-based methods in addition to a random selection and a roulette-based selection. Finally, the best version of each different heuristic selection hyperheuristic was applied to the selection of archiving and leader selection methods and compared to a state-of-art algorithm. To conclude the experiments, the hyperheuristic with the best results was compared to a state-of-art algorithm.

6.1 Preliminary research

In this section, a preliminary analysis of related work [6] is conducted, where the use of hyper-heuristic for solving many objective problems is proposed. This section is intended to reproduce that work, and to evaluate the implementation of the roulette based selection method. The goal is to investigate if the implementation is capable of reaching the same results as in [6] and if a simple selection of low-level heuristics has a better overall performance against each low-level heuristic individually.

In [6], nine low-level heuristics were used, as the combination of three archiving methods (Crowding Distance, Ideal and Multilevel Grid Archiving (MGA)) with three leader selection methods (Crowding Distance, NWSum, and Sigma). The hyper-heuristic is then compared with the nine low-level heuristics individually for the R2 indicator. In that paper, the hyper-heuristic (called H-MOPSO) using a roulette-based selection method had a better performance and the best overall ranking.

6.1.1 Experimental analysis

In this experiment, the same configurations than in [6] are used. Such configurations are shown in Table 6.1. All algorithms were executed with m objectives, where $m = \{2, 3, 5, 10, 15, 20\}$. The used benchmark problems were the DTLZ class (1 to 7). Both population and repository size were set to 100; the algorithms were executed in 30 independent runs with 100 iterations each. The ROULETTE was used as alias to describe the implementation of the hyper-heuristic using a roulette-based selection method; for each low-level heuristic, the concatenation of the archiving method acronym with the leader selection method acronym was used as an alias. For instance: the CDNWSUM alias means Crowding Distance Archiving with NWSum leader selection. Also, the implementation used is based on the Java framework jMetal [40], the related work was originally implemented in C++ using CUDA library. Besides some language and framework differences, all the algorithm structures, parameters and methods were used as in [6].

Table 6.1: Preliminary experiments configuration

Parameter	**value**	**description**
m	2, 3, 5, 10, 15 e 20	objective number
problem	DTLZ1 to 7	benchmark problems
popsize	100	population size
repsize	100	repository size
runs	30	independent runs

An overall analysis of the results is shown in Table 6.2. For each algorithm, it is shown the mean ranking (and standard deviation): in that analysis the ROULETTE algorithm had the best mean ranking, 1.667, with a standard deviation of 0.891, which means that the ROULETTE usually was the first or second ranked algorithm. In comparison, the second best-ranked algorithm was the MGACD with 3.095 mean ranking, 1.428 higher than the best ranked. The standard deviation of the second best ranked was 1.9, which means that the algorithm usually was the first to the fifth best algorithm. A more detailed analysis is presented in Appendix A.

Table 6.2: Mean ranking (and standard deviation) for each low-level heuristic and the hyper-heuristic based on roulette for the R2 indicator

Algorithm	**Mean ranking (and standard deviation)**
IDEALNWSUM	7.952(2.278)
MGACD	3.095(1.900)
MGANWSUM	6.119(1.854)
IDEALSIGMA	8.238(2.202)
CDSIGMA	5.095(1.810)
ROULETTE	**1.667(0.891)**
CDCD	3.976(3.043)
IDEALCD	6.976(2.464)
MGASIGMA	5.786(1.767)
CDNWSUM	6.095(1.849)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 3.355387e - 34$

It is possible to evaluate the difference between the algorithms based on the Table 6.3, where each algorithm is compared to the one with best mean ranking (ROULETTE). The ROULETTE was statistically different in almost all cases, with critical difference of 90.48263, just MGACD had no statistical difference, with an observed difference between the accumulated ranking of 60. The higher differences were against CDNWSUM (186), IDEALCD (223), IDEALNWSUM (264), IDEALSIGMA (276) and MGANWSUM (187).

6.1.2 Discussion

In this experiment, it was reproduced a related work from [6]. The objective was to evaluate the implementation of the method proposed in [6], and then, to reach the same conclusions. It was compared a pool of nine low-level heuristics against a simple roulette high-level that uses this pool. In that work, the author concludes that the proposed hyper-heuristic (called H-MOPSO) did not achieve the best performances in all cases, but, in

Table 6.3: Multiple comparisons between groups after Friedman test for the R2 indicator

Comparisons		**observed dif.**	**critical dif.**	**difference**
ROULETTE	CDCD	97	90.48263	✓
ROULETTE	CDNWSUM	186	90.48263	✓
ROULETTE	CDSIGMA	144	90.48263	✓
ROULETTE	IDEALCD	223	90.48263	✓
ROULETTE	IDEALNWSUM	264	90.48263	✓
ROULETTE	IDEALSIGMA	276	90.48263	✓
ROULETTE	MGACD	60	90.48263	
ROULETTE	MGANWSUM	187	90.48263	✓
ROULETTE	MGASIGMA	173	90.48263	✓

Comparisons: comparison between the best ranked algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

general, the proposed hyper-heuristic can properly select low-level heuristics increasing performance. It is possible to conclude that, the hyper-heuristic (ROULETTE) in general, had a better performance according to the R2 indicator in comparison with the low-level heuristics and a better overall ranking for different problems and objective numbers; supporting the conclusions of [6].

6.2 Study of Choice Function based methods

In this section it is shown the comparison between two Choice Function based methods: the Simplified Choice Function (SCF) [12, 11] and the Adaptive Choice Function (ACF) [23, 9]. The methods were used to replace the roulette selection method on H-MOPSO algorithm. The objective of this section is to evaluate the selection methods and to select one to be employed in the next studies.

Initially, it was adapted the H-MOPSO algorithm to select just archiving methods, the leader selection method selection will be introduced back in further sections. It is used the alias ACFFIXED to reference the H-MOPSO using ACF to select the archiving method with fixed leader selection (the default method: Crowding Distance). Moreover, the alias SCFFIXED to describe the H-MOPSO using SCF with default leader selection method.

The ACF proposed by [9] has two parameters an Scale Factor SF and a boolean parameter *mean*. In this study, the same parameter values as [9] are used: $SF = 0.5$ and $mean = true$. The Simplified Choice Function used in [11] has two scale factor parameters, α and β. And the values used in [11] and also in this experiment are $\alpha = 1.0$ and $\beta = 0.00005$.

In Table 6.4 is shown the average IGD value of the 20 independent runs, for each algorithm, benchmark problem and objective number. According to the IGD quality indicator, the H-MOPSO using ACF has a better performance than it using SCF for most test instances, especially from eight or more objectives. In Table 6.5 is shown the mean ranking of the algorithms. Where H-MOPSO-ACF had the best mean ranking (1.3 against 1.7 from H-MOPSO using SCF), with a statistical difference. Those values show that H-MOPSO-ACF was able to reach better results against H-MOPSO using SCF, according to the IGD quality indicator.

According to the Hypervolume quality indicator average value (Table 6.6) and mean ranking (Table 6.7), the H-MOPSO using SCF shows better average HV values than H-MOPSO using ACF, mainly for 3 and 5 objectives. Consequently, it reached a better mean ranking with a statistical difference according to Friedman statistical test (1.29 against 1.70 from ACF). It means that, in this study, the Hypervolume and IGD conclusions disagree.

Then, it was investigated the average contribution of each reference point to the IGD computation to evaluate the indicators disagreement. The more similar average contribution means that the fronts are more uniformly distributed. Moreover, the small average contributions mean higher convergence. In Figures 6.1(a) and 6.1(b) it is shown the average distance from the reference points to the Pareto front approximation of two instances (DTLZ2 with ten objectives and DTLZ2 with 15 objectives), which can be seen as the contribution of each reference point to the IGD computation. It is possible to observe that the H-MOPSO using ACF achieves more uniformly distributed fronts, with similar average contribution from the reference points. Also, the SCF version obtained a

Table 6.4: Mean (and standard deviation) for ACFFIXED and SCFFIXED for the IGD indicator

Obj.	problem	ACFFIXED	SCFFIXED
3	DTLZ1	7.22E-3(1.48E-3)	**6.56E-3(4.8E-4)**
	DTLZ2	9.47E-3(5.63E-4)	**9.45E-3(8.43E-4)**
	DTLZ3	1.5E-2(1.26E-2)	**9.01E-3(8.96E-4)**
	DTLZ4	**1.33E-2(4.81E-3)**	1.88E-2(4.9E-3)
	WFG6	**1.06E-2(1.66E-3)**	1.07E-2(7.09E-4)
	WFG7	1.4E-2(2.63E-3)	**1.2E-2(9.43E-4)**
5	DTLZ1	**1.17E-2(8.84E-4)**	1.21E-2(1.36E-3)
	DTLZ2	1.88E-2(1.33E-3)	**1.67E-2(4.83E-4)**
	DTLZ3	2.81E-2(3.12E-3)	**2.77E-2(3.6E-3)**
	DTLZ4	**2.25E-2(2.54E-3)**	2.38E-2(2.48E-3)
	WFG6	**1.52E-2(5.96E-4)**	1.53E-2(6.32E-4)
	WFG7	1.95E-2(2.01E-3)	**1.82E-2(1.46E-3)**
8	DTLZ1	**2.99E-2(2.85E-3)**	3.01E-2(2.63E-3)
	DTLZ2	**5.26E-2(3.33E-3)**	5.33E-2(3.85E-3)
	DTLZ3	**6.28E-2(4.29E-3)**	6.88E-2(4.64E-3)
	DTLZ4	**4.05E-2(1.78E-3)**	4.78E-2(2.35E-3)
	WFG6	4E-2(4.08E-3)	**3.95E-2(1.82E-3)**
	WFG7	**4.41E-2(4.42E-3)**	5.11E-2(5.44E-3)
10	DTLZ1	**2.43E-2(1.66E-3)**	2.55E-2(1.33E-3)
	DTLZ2	**4.49E-2(2.46E-3)**	4.72E-2(1.98E-3)
	DTLZ3	**5.19E-2(1.98E-3)**	5.34E-2(3.28E-3)
	DTLZ4	**3.25E-2(1.48E-3)**	3.56E-2(2.46E-3)
	WFG6	**3.61E-2(2.78E-3)**	3.61E-2(1.63E-3)
	WFG7	**5.41E-2(3.9E-3)**	6.75E-2(3.34E-3)
15	DTLZ1	**5.53E-2(2.16E-3)**	5.7E-2(2.41E-3)
	DTLZ2	**9.06E-2(2.85E-3)**	9.21E-2(3E-3)
	DTLZ3	**9.68E-2(3.17E-3)**	1.01E-1(4.84E-3)
	DTLZ4	**5.36E-2(2.37E-3)**	6.49E-2(6.87E-3)
	WFG6	1.11E-1(1.05E-2)	**1.06E-1(1.53E-2)**
	WFG7	**1.64E-1(1.36E-2)**	2.27E-1(2.1E-2)

for each problem the best mean performance is shown with **bold face**
the best value with statistical difference is shown with bold face and gray background

Table 6.5: Mean ranking for ACFFIXED and SCFFIXED for the IGD indicator

Algorithm	**Mean ranking**
SCFFIXED	1.7
ACFFIXED	**1.3**

the best value with statistical difference is shown with **bold face** and gray background
Friedman test p-value: $p = 0.02845974$

better convergence in some parts of the front, but with a higher difference between the contribution of the reference points.

Table 6.6: Mean (and standard deviation) for ACFFIXED and SCFFIXED for the HV indicator

Obj.	problem	ACFFIXED	SCFFIXED
3	DTLZ1	7.71E-1(6.88E-3)	**7.82E-1(4.35E-3)**
	DTLZ2	6.09E-1(7.49E-3)	**6.21E-1(3.24E-3)**
	DTLZ3	9.99E-1(5.01E-3)	**1E0(3.93E-7)**
	DTLZ4	**4.7E-1(1.03E-2)**	4.52E-1(1.83E-2)
	WFG6	3.78E-1(1.04E-2)	**3.8E-1(1.1E-2)**
	WFG7	4.1E-1(1.91E-2)	**4.42E-1(1.32E-2)**
5	DTLZ1	9.57E-1(8.97E-3)	**9.69E-1(5.3E-3)**
	DTLZ2	9.89E-1(2.08E-3)	**9.93E-1(4.89E-4)**
	DTLZ3	1E0(1.67E-8)	**1E0(1.31E-8)**
	DTLZ4	**9.98E-1(1.06E-4)**	9.98E-1(1.42E-4)
	WFG6	5.75E-1(2.21E-2)	**6.13E-1(1.1E-2)**
	WFG7	5.65E-1(2.21E-2)	**5.97E-1(2.13E-2)**
8	DTLZ1	1E0(7.26E-8)	**1E0(7.37E-10)**
	DTLZ2	9.93E-1(5.51E-3)	**9.98E-1(1.15E-3)**
	DTLZ3	1E0(1.37E-8)	**1E0(4.54E-9)**
	DTLZ4	**1E0(6.96E-6)**	1E0(3.24E-5)
	WFG6	6.27E-1(3.97E-2)	**7.01E-1(1.69E-2)**
	WFG7	**4.44E-1(2.63E-2)**	4.13E-1(3.07E-2)
10	DTLZ1	1E0(1.71E-9)	**1E0(0E0)**
	DTLZ2	9.96E-1(1.91E-3)	**9.98E-1(1.02E-3)**
	DTLZ3	**1E0(9.89E-10)**	1E0(2.12E-9)
	DTLZ4	**1E0(3.28E-7)**	1E0(7.29E-7)
	WFG6	7.47E-1(3.79E-2)	**7.84E-1(1.9E-2)**
	WFG7	**4.57E-1(2.76E-2)**	4.04E-1(2.17E-2)

for each problem the best mean performance is shown with **bold face**

the best value with statistical difference is shown with bold face and gray background

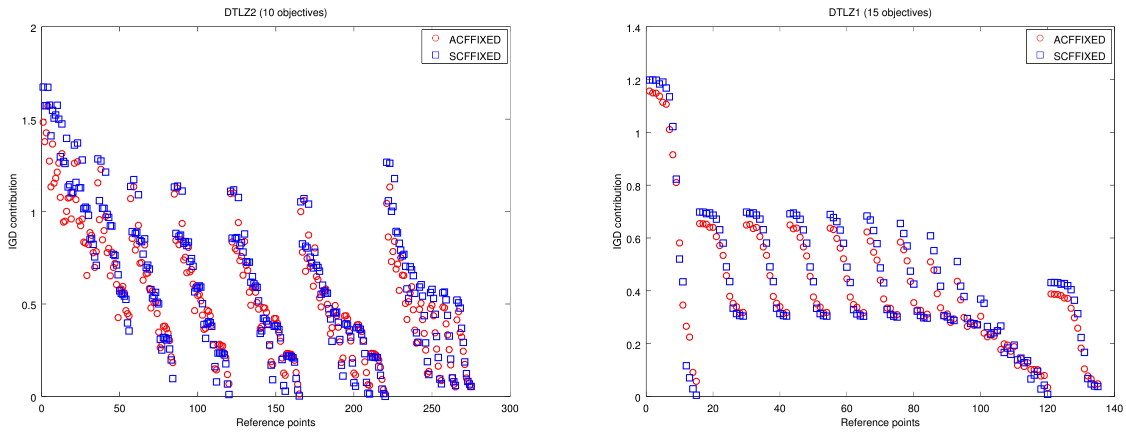
Table 6.7: Mean ranking for ACFFIXED and SCFFIXED for the HV indicator

Algorithm	**Mean ranking**
SCFFIXED	**1.291667**
ACFFIXED	1.708333

the best value with statistical difference is shown with **bold face** and gray background

Friedman test p-value: $p = 0.04122683$

In Figure 6.2, it is shown the boxplot of the average distance from the reference points to the front approximations of some problem instances. H-MOPSO with SCF (SCFFIXED) usually has a higher deviation of average distances, with some reference points with a smaller distance and others with a higher distance. On the other hand, ACFFIXED usually has a smaller deviation of the average distances.



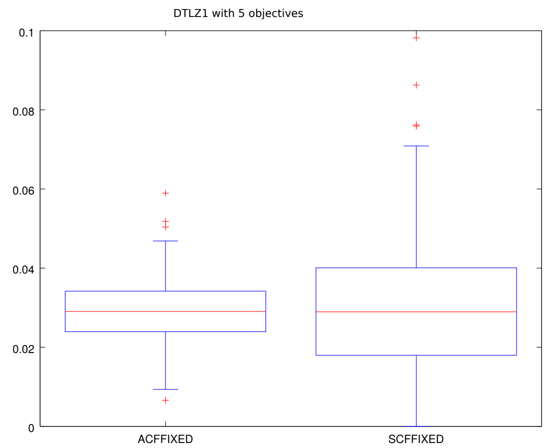
(a) DTLZ2 with 10 objectives

(b) DTLZ1 with 15 objectives

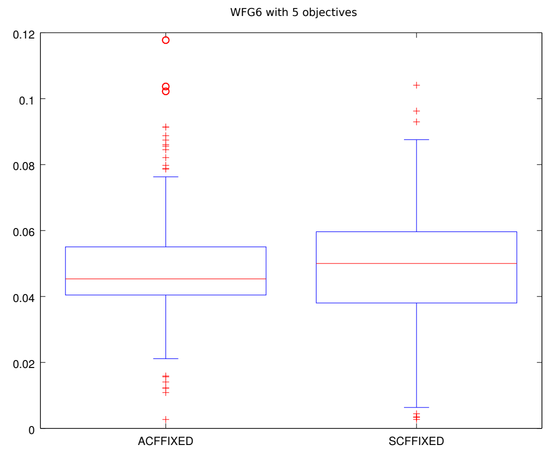
Figure 6.1: Average distance from the reference points to the Pareto front approximation in different problem instances

6.2.1 Discussion

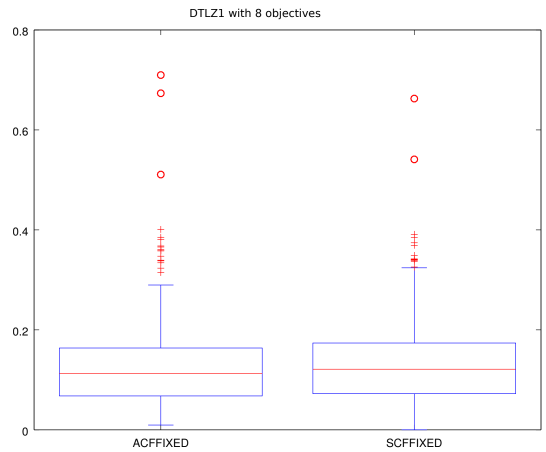
In this section, the H-MOPSO algorithm was evaluated using two different Choice Function based methods for archiving method selection. In this study, the leader selection method used was the default SMPSO method: Crowding Distance. The results show that none of the methods evaluated outperforms the other in both IGD and Hypervolume. In a detailed evaluation of the results, it was possible to observe that the ACF results were better distributed than the obtained by SCF, and with good convergence, this behavior was able to get a better IGD value than SCF. On the other hand, H-MOPSO-SCF was able to get a better convergence in some points of the front, in an expense of other points. This behavior was able to get better Hypervolume values than ACF. The main difference between both algorithms is the lack of the pair component in the SCF and the adaptive parameter strategy used in ACF. Finally, it was decided to select H-MOPSO using ACF to be the choice function based method used in subsequent studies due to a good convergence and diversity and, consequently, better results in IGD indicator mainly when the number of objectives increases.



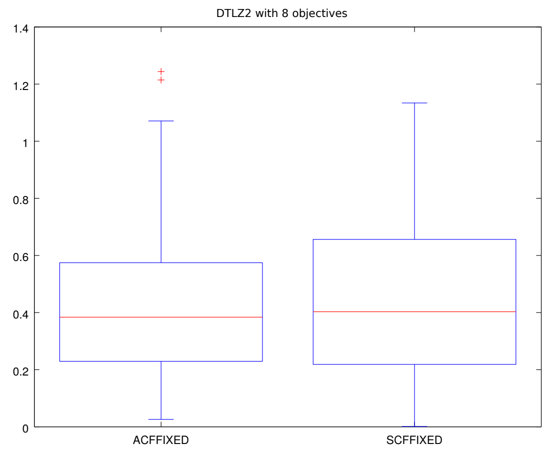
(a) DTLZ1 - 5 objectives



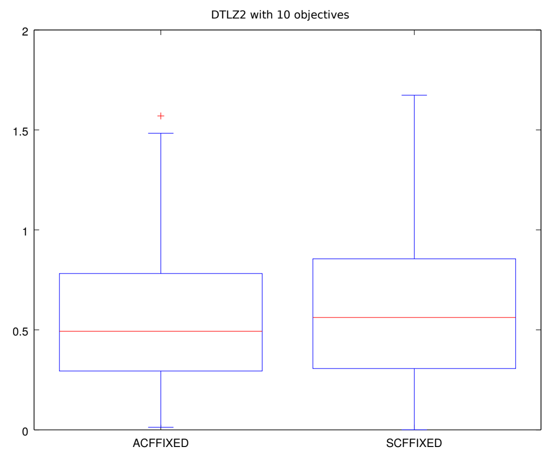
(b) WFG6 - 5 objectives



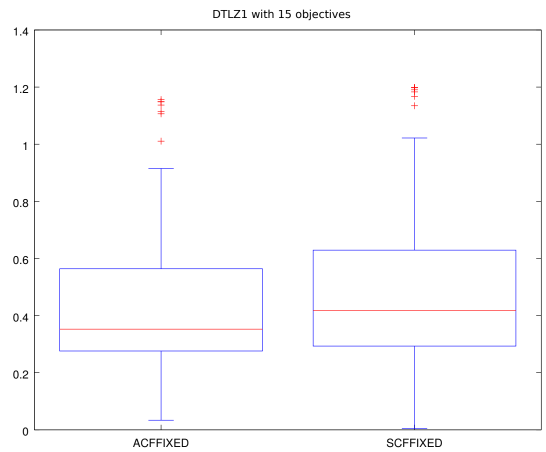
(c) DTLZ1 - 8 objectives



(d) DTLZ2 - 8 objectives



(e) DTLZ2 - 10 objectives



(f) DTLZ1 - 15 objectives

Figure 6.2: Boxplot of the average distance from the reference points to the Pareto front approximation

6.3 Adaptive Choice Function parameter configuration

In previous Section 6.2 two Choice Function based methods were evaluated: Adaptive Choice Function and Simplified Choice Function. Moreover, the Adaptive Choice Function was selected to be used in further studies. In this section, the parameters of the ACF are configured. For the Scale Factor parameter SF it was tried six values and a version based on normalization. For the boolean parameter *mean* it was tried both true and false. Those sets of values were combined into ten configuration instances, illustrated in Table 6.8.

Table 6.8: Configuration instances for Adaptive Choice Function

configuration	SF	*mean*
ACFFIXED001	0.01	true
ACFFIXED005	0.05	true
ACFFIXED01	0.1	true
ACFFIXED05	0.5	true
ACFFIXED1	1.0	true
ACFFIXED5	5.0	true
ACFFIXEDM01	0.1	false
ACFFIXEDM05	0.5	false
ACFFIXEDN	based on normalization	true
ACFFIXEDMN	based on normalization	false

The detailed results for each problem and the objective number is illustrated in Appendix B. In Table 6.9 is shown the mean ranking of the configuration instances for IGD, no configuration is better than all others with a statistical difference, in fact, most configuration instances had a similar mean ranking. The configuration with $SF = 5.0$ and $mean =$ true had the best mean ranking for IGD. Although, it had statistical difference just against one configuration, the ACFFIXED with $mean = false$ and using normalization, according to Table 6.10. The configuration with $SF = 0.1$ and $mean =$ false, had the second best mean ranking for IGD.

When evaluated with the Hypervolume indicator the configuration instances had very similar results, with no statistical difference against each other (according to Tables 6.11 and 6.12). The configuration with $SF = 0.1$ and $mean =$ false, had the best mean ranking for HV.

In this section, it was configured the parameters of the Adaptive Choice Function to select a configuration to be used in further studies. The selected configuration is the ACFFIXEDM01, it means, $SF = 0.1$ and $mean =$ false. It was selected because it gets

Table 6.9: Mean ranking for each configuration instance for the IGD indicator

Algorithm	Mean ranking
ACFFIXED5	**4.4**
ACFFIXEDM05	5.3
ACFFIXED1	5.933333
ACFFIXED005	4.933333
ACFFIXED05	5.5
ACFFIXEDN	5.6
ACFFIXED001	5.5
ACFFIXED01	6.033333
ACFFIXEDM01	4.633333
ACFFIXEDMN	7.166667

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.03217689$

Table 6.10: Multiple comparisons between groups after Friedman test for the IGD indicator

Comparisons		observed dif.	critical dif.	difference
ACFFIXED5	ACFFIXED001	33	76.47178	
ACFFIXED5	ACFFIXED005	16	76.47178	
ACFFIXED5	ACFFIXED01	49	76.47178	
ACFFIXED5	ACFFIXED05	33	76.47178	
ACFFIXED5	ACFFIXED1	46	76.47178	
ACFFIXED5	ACFFIXEDM01	7	76.47178	
ACFFIXED5	ACFFIXEDM05	27	76.47178	
ACFFIXED5	ACFFIXEDMN	83	76.47178	✓
ACFFIXED5	ACFFIXEDN	36	76.47178	

Comparisons: comparison between the best ranked configuration instance and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

the best mean ranking for the HV indicator and the second best for IGD indicator, also, getting the best accumulated mean ranking (IGD mean ranking plus HV mean ranking).

Table 6.11: Mean ranking for each configuration instance for the HV indicator

Algorithm	Mean ranking
ACFFIXED5	5.666667
ACFFIXEDM05	4.875
ACFFIXED1	6.375
ACFFIXED005	5.333333
ACFFIXED05	5.916667
ACFFIXEDN	5.166667
ACFFIXED001	6.041667
ACFFIXED01	6.083333
ACFFIXEDM01	**4.375**
ACFFIXEDMN	5.166667

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.420663$

Table 6.12: Multiple comparisons between groups after Friedman test for the HV indicator

Comparisons		observed dif.	critical dif.	difference
ACFFIXEDM01	ACFFIXED001	40	68.39844	
ACFFIXEDM01	ACFFIXED005	23	68.39844	
ACFFIXEDM01	ACFFIXED01	41	68.39844	
ACFFIXEDM01	ACFFIXED05	37	68.39844	
ACFFIXEDM01	ACFFIXED1	48	68.39844	
ACFFIXEDM01	ACFFIXED5	31	68.39844	
ACFFIXEDM01	ACFFIXEDM05	12	68.39844	
ACFFIXEDM01	ACFFIXEDMN	19	68.39844	
ACFFIXEDM01	ACFFIXEDN	19	68.39844	

Comparisons: comparison between the best ranked configuration instance and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

6.4 Study of Multi-Armed Bandit based methods

In this section, three Multi-Armed Bandit based selection methods are evaluated. The studied methods are: the method proposed in [26] UCB1, and two methods proposed in [9], UCBV and UCBTuned. The objective is to evaluate the UCB methods on the H-MOPSO algorithm proposed by [22]. On the experiments of this section, same as in

Section 6.2, the selection method will be used just for selecting archiving methods, and the leader selection method will be the default: Crowding Distance.

The selection methods parameters are used based on values proposed by [26, 9]. The scale factor C is set to 0.5. The decay factor D is set to 1.0 and the sliding window size W is set to 20. The W value proposed in the literature is half of population size, and the selection is made for each individual in the population, although the selection is made once for each iteration, in this section. Although, the literature value is not coherent to this scenario. All three selection methods have the same parameters and use the same values.

In Table 6.13 is shown the average IGD values of the three selection methods for each problem and objective value. The UCB1 has the best average value in most cases, and it has the best value with statistical difference against the two other methods in one instance (DTLZ1 with ten objectives). The UCB1 method has also the best mean ranking (1.63) with a statistical difference to UCBV (Table 6.14).

When evaluated with the HV indicator the methods had more similar results. The UCB1 had the best mean value in more cases, on the other hand, UCBV had three best values with statistical difference against the two other methods. When evaluated the mean ranking, the UCB1 had the best value again, but with no statistical difference in the other methods.

In this section, three selection methods based on UCB were evaluated. In general, the methods had similar results, but the UCB1 method had the best average quality indicator value in most cases, and it also had the best mean ranking for the two quality indicators. Due to its good results, the UCB1 method is the Multi-Armed Bandit based method selected to be used in further studies.

6.5 FRRMAB-UCB1 parameter configuration

The parameter configuration of the UCB1 selection method is presented in this Section. It was selected after comparison against UCBV and UCBTuned, all three using default parameter values. Three values were tried for the sliding window size parameter W: 10,

Table 6.13: Mean (and standard deviation) for UCBTunedFIXED, UCB1FIXED and UCBVFIXED for the IGD indicator

Obj.	problem	UCBTunedFIXED	UCB1FIXED	UCBVFIXED
	DTLZ1	7.76E-3(1.29E-3)	**7.5E-3(1.21E-3)**	8.79E-3(6.07E-3)
3	DTLZ2	1.08E-2(1.12E-3)	1.05E-2(1.83E-3)	**9.78E-3(8.17E-4)**
	DTLZ3	1.38E-2(4.84E-3)	1.52E-2(1.26E-2)	**1.19E-2(6.51E-3)**
	DTLZ4	2.11E-2(8.83E-3)	1.94E-2(1.04E-2)	**1.6E-2(8.2E-3)**
	WFG6	1.11E-2(1.5E-3)	1.11E-2(1.56E-3)	**1.03E-2(1.14E-3)**
	WFG7	1.61E-2(3.04E-3)	1.75E-2(3.32E-3)	**1.55E-2(2.8E-3)**
	DTLZ1	**1.15E-2(1.26E-3)**	1.18E-2(9.26E-4)	1.61E-2(1.61E-3)
5	DTLZ2	1.96E-2(1.08E-3)	**1.89E-2(1.27E-3)**	2.28E-2(3.93E-3)
	DTLZ3	2.88E-2(2.43E-3)	**2.8E-2(2.9E-3)**	2.86E-2(3.2E-3)
	DTLZ4	2.37E-2(2.12E-3)	2.31E-2(2.25E-3)	**2.22E-2(2.12E-3)**
	WFG6	1.56E-2(8.54E-4)	**1.52E-2(5.52E-4)**	1.54E-2(6.85E-4)
	WFG7	1.99E-2(1.73E-3)	**1.93E-2(1.16E-3)**	2.03E-2(1.84E-3)
	DTLZ1	3.18E-2(2.7E-3)	**3.11E-2(1.92E-3)**	1.35E-1(2.71E-1)
8	DTLZ2	5.17E-2(3.06E-3)	5.09E-2(3.11E-3)	**5.05E-2(3.1E-3)**
	DTLZ3	6.52E-2(4.69E-3)	**6.21E-2(4.38E-3)**	2.83E0(2.76E0)
	DTLZ4	4.26E-2(2.04E-3)	**4.23E-2(1.82E-3)**	4.29E-2(2.07E-3)
	WFG6	**3.91E-2(3.03E-3)**	3.95E-2(3.33E-3)	4.04E-2(3.13E-3)
	WFG7	4.42E-2(4.11E-3)	4.45E-2(4.28E-3)	**4.3E-2(3.06E-3)**
	DTLZ1	2.67E-2(1.62E-3)	**2.45E-2(1.55E-3)**	1.82E-1(2.67E-1)
10	DTLZ2	**4.47E-2(2.25E-3)**	4.51E-2(3.05E-3)	4.48E-2(2.95E-3)
	DTLZ3	5.24E-2(1.5E-3)	**5.19E-2(2.03E-3)**	2.05E0(1.1E0)
	DTLZ4	3.33E-2(1.09E-3)	**3.24E-2(1.05E-3)**	3.28E-2(1.15E-3)
	WFG6	**3.44E-2(2.33E-3)**	3.51E-2(2.08E-3)	3.54E-2(2.72E-3)
	WFG7	5.46E-2(4.18E-3)	**5.28E-2(5.74E-3)**	5.54E-2(4.91E-3)
	DTLZ1	5.68E-2(1.59E-3)	**5.66E-2(2.27E-3)**	6.12E-1(9.05E-1)
15	DTLZ2	9.15E-2(2.56E-3)	**9.09E-2(2.72E-3)**	9.24E-2(2.68E-3)
	DTLZ3	9.8E-2(3E-3)	**9.66E-2(2.93E-3)**	5.19E0(3.89E0)
	DTLZ4	6.01E-2(5.75E-3)	5.54E-2(3.74E-3)	**5.46E-2(2.04E-3)**
	WFG6	1.09E-1(1.26E-2)	**1.08E-1(1.3E-2)**	1.1E-1(1.41E-2)
	WFG7	**1.62E-1(1.75E-2)**	1.67E-1(1.44E-2)	1.66E-1(1.04E-2)

for each problem the best mean performance is shown with **bold face**
the best value with statistical difference is shown with bold face and gray background

Table 6.14: Mean ranking for UCBTunedFIXED, UCB1FIXED and UCBVFIXED for the IGD indicator

Algorithm	Mean ranking
UCB1FIXED	**1.633333**
UCBTunedFIXED	2.133333
UCBVFIXED	2.233333

the best value is shown with **bold face**
Friedman test p-value: $p = 0.0450492$

20 and 50. For the scale factor C parameter it was tried four values: 10, 5, 1 and 0.5. Those two sets of values were combined into 12 configuration instances. In this section, the UCB method is used to select just the archiving method, the leader selection method was the default: Crowding Distance.

Table 6.15: Mean (and standard deviation) for UCBTunedFIXED, UCB1FIXED and UCBVFIXED for the HV indicator

Obj.	problem	UCBTunedFIXED	UCB1FIXED	UCBVFIXED
	DTLZ1	**9.99E-1(5.98E-4)**	9.99E-1(4.32E-3)	9.99E-1(1.31E-3)
3	DTLZ2	6.77E-1(1.67E-2)	6.8E-1(1.54E-2)	**6.83E-1(7.25E-3)**
	DTLZ3	1E0(9.14E-5)	1E0(1.52E-3)	**1E0(8.11E-5)**
	DTLZ4	8.25E-1(8.69E-3)	8.28E-1(9.79E-3)	**8.29E-1(1E-2)**
	WFG6	3.71E-1(6.67E-3)	3.73E-1(9.76E-3)	**3.82E-1(1.08E-2)**
	WFG7	4.02E-1(2.18E-2)	4.04E-1(2.46E-2)	**4.15E-1(2.03E-2)**
	DTLZ1	**9.81E-1(5.71E-3)**	9.79E-1(5.12E-3)	9.61E-1(1.11E-2)
5	DTLZ2	**9.95E-1(1.58E-3)**	9.94E-1(1.94E-3)	9.89E-1(5.88E-3)
	DTLZ3	1E0(8.5E-9)	**1E0(7.82E-9)**	1E0(3.96E-8)
	DTLZ4	9.98E-1(8.27E-5)	9.98E-1(9.06E-5)	**9.98E-1(9.07E-5)**
	WFG6	5.54E-1(2.4E-2)	**5.62E-1(2E-2)**	5.51E-1(2.77E-2)
	WFG7	5.53E-1(2.69E-2)	**5.64E-1(2.26E-2)**	5.55E-1(2.93E-2)
	DTLZ1	**1E0(3.08E-11)**	1E0(1.74E-8)	1E0(1.77E-8)
8	DTLZ2	9.93E-1(3.58E-3)	9.93E-1(3.71E-3)	**9.96E-1(1.31E-3)**
	DTLZ3	1E0(6.21E-8)	**1E0(5.95E-9)**	1E0(2.04E-6)
	DTLZ4	1E0(1E-5)	**1E0(5.3E-6)**	1E0(1.09E-5)
	WFG6	6.67E-1(3.41E-2)	**6.82E-1(4.38E-2)**	6.72E-1(3.66E-2)
	WFG7	4.32E-1(2.59E-2)	4.38E-1(2.25E-2)	**4.41E-1(1.99E-2)**
	DTLZ1	**1E0(2.24E-11)**	1E0(1.03E-9)	1E0(2.44E-7)
10	DTLZ2	9.96E-1(2.29E-3)	**9.96E-1(2.02E-3)**	9.95E-1(2.93E-3)
	DTLZ3	1E0(2.61E-9)	**1E0(3.1E-9)**	1E0(3E-7)
	DTLZ4	1E0(7.72E-7)	**1E0(1.64E-7)**	1E0(3.19E-7)
	WFG6	**7.65E-1(3.92E-2)**	7.62E-1(3.39E-2)	7.58E-1(4.38E-2)
	WFG7	4.52E-1(3E-2)	**4.54E-1(3.27E-2)**	4.47E-1(2.93E-2)

Table 6.16: Mean ranking for UCBTunedFIXED, UCB1FIXED and UCBVFIXED for the HV indicator

Algorithm	**Mean ranking**
UCB1FIXED	**1.708333**
UCBTunedFIXED	2.166667
UCBVFIXED	2.125

the best value is shown with **bold face**
Friedman test p-value: $p = 0.2140241$

In Table 6.17 and 6.18 the results of the configurations for the IGD quality indicator are shown. The configuration with $C = 10$ and $W = 10$ had the best mean ranking, with statistical difference against all three configurations using $C = 0.5$ and two configurations using $C = 1.0$. When evaluated the Hypervolume quality indicator the configuration with the best mean ranking was the configuration with $C = 10$ and $W = 20$. The configuration with $C = 10$ and $W = 10$ had the second best mean ranking. The best mean ranked configuration also had statistical difference against all three configurations using $C = 0.5$ and one configuration using $C = 1.0$.

Table 6.17: Mean ranking (and standard deviation) for each configuration instance for the IGD indicator

Algorithm	**Mean ranking (and standard deviation)**
UCBC1W20	7.633(1.516)
UCBC0.5W50	9.767(2.940)
UCBC1W10	5.667(1.758)
UCBC0.5W20	9.900(3.026)
UCBC10W20	4.333(3.399)
UCBC0.5W10	7.900(2.797)
UCBC10W10	**4.100(2.663)**
UCBC5W50	4.767(2.789)
UCBC10W50	4.933(2.250)
UCBC5W10	5.100(3.300)
UCBC1W50	9.000(2.898)
UCBC5W20	4.900(2.891)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 7.278681e - 21$

Table 6.18: Multiple comparisons between groups after Friedman test for the IGD indicator

Comparisons		**observed dif.**	**critical dif.**	**difference**
UCBC10W10	UCBC0.5W10	114	94.05885	✓
UCBC10W10	UCBC0.5W20	174	94.05885	✓
UCBC10W10	UCBC0.5W50	170	94.05885	✓
UCBC10W10	UCBC10W20	7	94.05885	
UCBC10W10	UCBC10W50	25	94.05885	
UCBC10W10	UCBC1W10	47	94.05885	
UCBC10W10	UCBC1W20	106	94.05885	✓
UCBC10W10	UCBC1W50	147	94.05885	✓
UCBC10W10	UCBC5W10	30	94.05885	
UCBC10W10	UCBC5W20	24	94.05885	
UCBC10W10	UCBC5W50	20	94.05885	

Comparisons: comparison between the best ranked configuration instance and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

The evaluation of the parameters of the UCB shows that some parameters are better than others. The value of $C = 10$ was present in the best-ranked configuration in both IGD and HV. Also, in three of the first fifth-best ranked configurations in both IGD and HV. Due to that, and also due to be the first best-ranked configuration in IGD, the

Table 6.19: Mean ranking (and standard deviation) for each configuration instance for the HV indicator

Algorithm	**Mean ranking (and standard deviation)**
UCBC1W20	7.542(1.707)
UCBC0.5W50	9.917(2.943)
UCBC1W10	6.125(2.006)
UCBC0.5W20	10.208(2.958)
UCBC10W20	**4.125(2.368)**
UCBC0.5W10	8.000(2.887)
UCBC10W10	4.292(2.700)
UCBC5W50	5.417(2.798)
UCBC10W50	4.375(2.463)
UCBC5W10	4.292(2.525)
UCBC1W50	9.083(2.660)
UCBC5W20	4.625(3.133)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 1.077046e - 18$

Table 6.20: Multiple comparisons between groups after Friedman test for the HV indicator

Comparisons		**observed dif.**	**critical dif.**	**difference**
UCBC10W20	UCBC0.5W10	93	84.1288	✓
UCBC10W20	UCBC0.5W20	146	84.1288	✓
UCBC10W20	UCBC0.5W50	139	84.1288	✓
UCBC10W20	UCBC10W10	4	84.1288	
UCBC10W20	UCBC10W50	6	84.1288	
UCBC10W20	UCBC1W10	48	84.1288	
UCBC10W20	UCBC1W20	82	84.1288	
UCBC10W20	UCBC1W50	119	84.1288	✓
UCBC10W20	UCBC5W10	4	84.1288	
UCBC10W20	UCBC5W20	12	84.1288	
UCBC10W20	UCBC5W50	31	84.1288	

Comparisons: comparison between the best ranked configuration instance and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

second best in HV and the best accumulated ranking (IGD ranking plus HV ranking) the selected configuration is the $C = 10$, $W = 10$, with statistical difference against 5 of 11 other configurations.

6.6 Study of archiving method selection in MOPSO

In this section four selection methods are evaluated in the H-MOPSO algorithm: i) the ACF selected in Section 6.2 and configured in Section 6.3; ii) the FRRMAB-UCB selected in Section 6.4 and configured in Section 6.5; iii) the Roulette based selection method proposed in [6] (the original H-MOPSO selection method); and iv) a simple random based selection, that does not use the information of the previous low-level heuristics application to select the next low-level heuristic. The four different H-MOPSO algorithms (ACF, UCB, ROULETTE and RANDOM) are compared with a state-of-art algorithm MOEA/D-DRA.

The ACF parameters configured in Section 6.3 are $SF = 0.1$ and $mean =$ false. The UCB parameters configured in Section 6.5 are $C = 10$, $D = 1.0$ and $W = 10$. In this section the selection methods will select just the archiving method. The leader selection method used is the default Crowding Distance.

In Table 6.23 the average IGD value of the algorithms for each problem and objective number is presented. The H-MOPSO using ACF had the best result in most cases, on the other hand, the H-MOPSO using roulette was able to get the best result with statistical difference against all other algorithms in DTLZ2 with five objectives. Besides, the MOEA/D-DRA had the best average IGD with a statistical difference in DTLZ3 with ten objectives, DTLZ2 with 15 objectives and WFG6 with 15 objectives.

The ACF get the best average ranking (2.067), with a standard deviation of 1.123. That means that the H-MOPSO-ACF usually was the first to the third-ranked algorithm. The UCB was the second best ranked with 2.733. The RANDOM method was the third one (2.933). The fourth was the ROULETTE (3.033) and the MOEA/D-DRA the fifth (4.233) according to Table 6.21. The H-MOPSO was able to get best average ranking with statistical difference against the MOEA/D-DRA with all selection methods (Table 6.22).

Table 6.21: Mean ranking (and standard deviation) for UCB, ACF, ROULETTE, RANDOM and MOEADDRA for the IGD indicator

Algorithm	Mean ranking (and standard deviation)
ROULETTE	3.033(1.329)
MOEADDRA	4.233(1.499)
UCB	2.733(1.093)
ACF	**2.067(1.123)**
RANDOM	2.933(1.031)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 5.83008e-06$

Table 6.22: Multiple comparisons between groups after Friedman test for IGD indicator

Comparisons		observed dif.	critical dif.	difference
ACF	UCB	20	34.379	
ACF	MOEADDRA	65	34.379	✓
ACF	RANDOM	26	34.379	
ACF	ROULETTE	29	34.379	
UCB	MOEADDRA	45	34.379	✓
UCB	RANDOM	6	34.379	
UCB	ROULETTE	9	34.379	
MOEADDRA	RANDOM	39	34.379	✓
MOEADDRA	ROULETTE	36	34.379	✓
RANDOM	ROULETTE	3	34.379	

Comparisons: comparison between each algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

Table 6.23: Mean (and standard deviation) for UCB, ACF, ROULETTE, RANDOM and MOEADDRA for the IGD indicator

Obj.	problem	UCB	ACF	ROULETTE	RANDOM	MOEADDRA
3	DTLZ1	7.33E-3(1.42E-3)	7.83E-3(4.28E-3)	6.98E-3(7.16E-4)	**6.85E-3(8.65E-4)**	4.89E-2(8.72E-2)
	DTLZ2	1.03E-2(9.51E-4)	9.81E-3(7.36E-4)	**9.29E-3(7.06E-4)**	1.02E-2(1.51E-3)	1.04E-2(5.71E-4)
	DTLZ3	1.84E-2(1.36E-2)	**1.1E-2(8.1E-3)**	1.6E-2(1.48E-2)	1.24E-2(5.12E-3)	7.77E-1(1.64E0)
	DTLZ4	1.56E-2(5.14E-3)	**1.19E-2(6.24E-3)**	2.38E-2(5.25E-3)	1.42E-2(4.89E-3)	2.53E-2(9.71E-3)
	WFG6	1.19E-2(1.96E-3)	**1E-2(1.07E-3)**	1.02E-2(1.03E-3)	1.07E-2(1.44E-3)	1.21E-2(5.51E-4)
	WFG7	1.58E-2(2.91E-3)	1.35E-2(1.9E-3)	**1.23E-2(7.38E-4)**	1.58E-2(2.5E-3)	1.48E-2(9.16E-4)
5	DTLZ1	1.17E-2(7.95E-4)	1.16E-2(6.41E-4)	**1.09E-2(8.37E-4)**	1.19E-2(9.33E-4)	2.63E-2(1.16E-2)
	DTLZ2	1.94E-2(1.38E-3)	1.85E-2(1.28E-3)	**1.64E-2(3.88E-4)**	1.91E-2(1.32E-3)	2.17E-2(2.3E-4)
	DTLZ3	2.82E-2(2.54E-3)	2.88E-2(3.28E-3)	2.91E-2(2.96E-3)	**2.81E-2(1.9E-3)**	5.88E-2(4.45E-2)
	DTLZ4	**2.16E-2(2.67E-3)**	2.19E-2(1.94E-3)	2.65E-2(4.59E-3)	2.21E-2(2.54E-3)	2.4E-2(1.65E-3)
	WFG6	1.53E-2(4.57E-4)	**1.51E-2(5.3E-4)**	1.53E-2(4.47E-4)	1.53E-2(8.66E-4)	2.37E-2(2.28E-4)
	WFG7	1.95E-2(1.64E-3)	**1.75E-2(8.91E-4)**	1.75E-2(7.06E-4)	2E-2(1.94E-3)	3.27E-2(2.69E-3)
8	DTLZ1	3.06E-2(2E-3)	**2.87E-2(2.8E-3)**	2.9E-2(3.11E-3)	3.06E-2(2.53E-3)	3.12E-2(1.35E-3)
	DTLZ2	5.1E-2(2.96E-3)	**5.08E-2(3.38E-3)**	5.19E-2(3.84E-3)	5.17E-2(3.93E-3)	5.22E-2(1.59E-3)
	DTLZ3	6.29E-2(3.93E-3)	**6.13E-2(3.75E-3)**	6.68E-2(4.67E-3)	6.29E-2(3.45E-3)	8.2E-2(8.05E-2)
	DTLZ4	4.25E-2(1.75E-3)	4.21E-2(2.38E-3)	4.74E-2(2.24E-3)	**4.19E-2(2.5E-3)**	5.81E-2(3.53E-3)
	WFG6	**3.89E-2(2.71E-3)**	3.97E-2(3.76E-3)	4.01E-2(2.84E-3)	4.05E-2(2.75E-3)	4.51E-2(3.06E-3)
	WFG7	4.46E-2(4.8E-3)	**4.36E-2(4.62E-3)**	5.03E-2(6.09E-3)	4.53E-2(4.85E-3)	6.03E-2(7.18E-3)
10	DTLZ1	2.45E-2(2.26E-3)	**2.31E-2(1.19E-3)**	2.39E-2(1.68E-3)	2.49E-2(1.3E-3)	2.6E-2(3.47E-4)
	DTLZ2	4.55E-2(3.11E-3)	4.51E-2(2.8E-3)	4.65E-2(2.08E-3)	4.47E-2(2.48E-3)	**4.44E-2(5.72E-4)**
	DTLZ3	5.07E-2(2.26E-3)	5.03E-2(2.42E-3)	5.47E-2(3.69E-3)	5.11E-2(2.64E-3)	**4.57E-2(1.48E-3)**
	DTLZ4	3.23E-2(1.28E-3)	**3.2E-2(1.27E-3)**	3.7E-2(1.05E-3)	3.23E-2(1.51E-3)	4.91E-2(2.6E-3)
	WFG6	3.48E-2(2.72E-3)	**3.44E-2(2.18E-3)**	3.58E-2(2.35E-3)	3.55E-2(2.04E-3)	4.47E-2(2.2E-3)
	WFG7	**5.5E-2(4.46E-3)**	5.7E-2(3.4E-3)	6.46E-2(3.47E-3)	5.57E-2(3.97E-3)	6.62E-2(6.96E-3)
15	DTLZ1	5.55E-2(2.05E-3)	5.45E-2(1.76E-3)	5.43E-2(2.49E-3)	5.67E-2(1.87E-3)	**5.37E-2(1.37E-3)**
	DTLZ2	9.17E-2(2.95E-3)	9.21E-2(1.32E-3)	9.17E-2(2.39E-3)	9.18E-2(2.52E-3)	**8.6E-2(2.14E-3)**
	DTLZ3	**9.56E-2(2.18E-3)**	1.32E-1(1.54E-1)	9.78E-2(3.48E-3)	9.59E-2(2.69E-3)	1.67E-1(2.53E-1)
	DTLZ4	**5.27E-2(1.09E-3)**	5.38E-2(2.4E-3)	5.66E-2(4.04E-3)	5.37E-2(2.18E-3)	8.99E-2(2.35E-3)
	WFG6	1.08E-1(8.7E-3)	1.04E-1(1.7E-2)	**9.79E-2(7.75E-3)**	1.11E-1(1.29E-2)	1.22E-1(1.72E-2)
	WFG7	1.65E-1(9.92E-3)	1.74E-1(1.29E-2)	2.18E-1(1.97E-2)	1.62E-1(1.55E-2)	**1.18E-1(1.08E-2)**

Table 6.24: Mean ranking (and standard deviation) for UCB, ACF, ROULETTE, RANDOM and MOEADDRA for the HV indicator

Algorithm	Mean ranking (and standard deviation)
ROULETTE	2.625(1.679)
MOEADDRA	3.667(1.818)
UCB	3.292(1.020)
ACF	**2.333(0.986)**
RANDOM	3.083(0.862)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.02931538$

According to the Hypervolume quality indicator, the H-MOPSO using roulette got the best average in most cases (Table 6.26), and the best average with a statistical difference in DTLZ2 with three objectives, DTLZ1 with five objectives and WFG6 with 5, 8 and ten objectives. The MOEA/D-DRA had the best average Hypervolume with a statistical difference in DTLZ3 with eight objectives, WFG7 with 8, DTLZ3 with ten objectives and WFG7 with ten objectives. In Table 6.24 the mean ranking for the HV indicator is shown. The H-MOPSO using ACF had the best mean ranking (2.333), with a standard deviation of 0.986, what means that it usually was the first to third best algorithm; the H-MOPSO-ROULETTE was the second best ranked (2.625), with 1.679 of standard deviation, what means that it usually was the first to the fourth best algorithm; The RANDOM was the third in the rank and UCB the forth; The MOEA/D-DRA was the fifth, with a standard deviation of 1.818. For the Hypervolume quality indicator, only the H-MOPSO using ACF was better with statistical difference against MOEA/D-DRA.

In both IGD and HV, the RANDOM method got the lowest standard deviation on ranking, and the MOEA/D-DRA got the highest standard deviation. It means that the MOEA/D-DRA was the algorithm with more irregular ranking, being the best in some problem instances and the worst in others. It also shows that, the use of hyper-heuristic in MOPSO, make possible to reach good results in almost all problem instances. That behavior refers to the hyper-heuristic ability to deal with the No Free Lunch theorem. According to No Free Lunch theorem, no algorithm outperforms others in all problems. The hyper-heuristic ability is to select a proper algorithm to be used for each problem instance and even a random selection usually is better than any low-level heuristic applied

Table 6.25: Multiple comparisons between groups after Friedman test for HV indicator

Comparisons		**observed dif.**	**critical dif.**	**difference**
ACF	UCB	23	30.74951	
ACF	MOEADDRA	32	30.74951	✓
ACF	RANDOM	18	30.74951	
ACF	ROULETTE	7	30.74951	
UCB	MOEADDRA	9	30.74951	
UCB	RANDOM	5	30.74951	
UCB	ROULETTE	16	30.74951	
MOEADDRA	RANDOM	14	30.74951	
MOEADDRA	ROULETTE	25	30.74951	
RANDOM	ROULETTE	11	30.74951	

Comparisons: comparison between each algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

individually.

In this section, the H-MOPSO algorithm using four different heuristic selection methods is evaluated: ACF, UCB, ROULETTE and RANDOM. The H-MOPSO variations were compared with a state-of-art algorithm MOEA/D-DRA. The H-MOPSO using ACF had the best average ranking in both IGD and Hypervolume, although, all four selection methods were statistically equivalent in both IGD and Hypervolume. When compared with MOEA/D-DRA all H-MOPSO variations were able to get a better average ranking with a statistical difference in IGD. In Hypervolume, all methods got better average ranking than MOEA/D-DRA but just the H-MOPSO-ACF with statistical difference.

In general, it is possible to conclude that the H-MOPSO can get good results, being able to outperforms a state-of-art algorithm regarding average ranking. It was not found any strong evidence that one selection method is better than any other. Also, even the H-MOPSO using a random selection was able to get better results when compared with the MOEA/D-DRA (regarding average ranking).

Table 6.26: Mean (and standard deviation) for UCB, ACF, ROULETTE, RANDOM and MOEADDRA for the HV indicator

Obj.	problem	UCB	ACF	ROULETTE	RANDOM	MOEADDRA
3	DTLZ1	1E0(6.83E-5)	1E0(6.24E-6)	**1E0(6.32E-9)**	1E0(5.77E-6)	1E0(4.21E-4)
	DTLZ2	8.08E-1(1.3E-2)	8.16E-1(3.16E-3)	**8.22E-1(1.94E-3)**	8.08E-1(1.62E-2)	7.99E-1(3.86E-3)
	DTLZ3	1E0(1.04E-5)	**1E0(2.02E-6)**	1E0(9.2E-6)	1E0(2.24E-6)	9.98E-1(5.93E-3)
	DTLZ4	8.44E-1(3.76E-3)	**8.46E-1(4.96E-3)**	8.34E-1(7.76E-3)	8.44E-1(4.44E-3)	8.29E-1(8.37E-3)
	WFG6	3.7E-1(1.1E-2)	3.85E-1(1.2E-2)	**3.9E-1(1.29E-2)**	3.78E-1(8.52E-3)	3.17E-1(9.59E-3)
	WFG7	4.09E-1(2.32E-2)	4.32E-1(1.76E-2)	**4.55E-1(1.61E-2)**	4.08E-1(1.59E-2)	3.79E-1(1.68E-2)
5	DTLZ1	1E0(2.44E-8)	1E0(3.32E-8)	**1E0(0E0)**	1E0(4.18E-8)	1E0(4.79E-6)
	DTLZ2	9.9E-1(2.03E-3)	9.92E-1(2.39E-3)	**9.94E-1(2.68E-4)**	9.91E-1(2.16E-3)	9.94E-1(1.78E-4)
	DTLZ3	**1E0(1.53E-8)**	1E0(3.43E-8)	1E0(1.18E-7)	1E0(5.3E-8)	1E0(4.52E-5)
	DTLZ4	**9.99E-1(7.13E-5)**	9.99E-1(4.79E-5)	9.99E-1(1.29E-4)	9.99E-1(7.12E-5)	9.99E-1(6.9E-5)
	WFG6	5.56E-1(2.57E-2)	5.72E-1(1.98E-2)	**5.97E-1(1.1E-2)**	5.66E-1(3.06E-2)	3.53E-1(1.98E-2)
	WFG7	5.08E-1(2.03E-2)	5.53E-1(1.34E-2)	5.57E-1(1.42E-2)	5.05E-1(2.28E-2)	**6.15E-1(4.5E-2)**
8	DTLZ1	1E0(2.86E-8)	1E0(1.7E-9)	1E0(1.34E-10)	1E0(1.17E-8)	**1E0(0E0)**
	DTLZ2	9.95E-1(1.76E-3)	9.96E-1(1.79E-3)	**9.97E-1(1.75E-3)**	9.95E-1(5.5E-3)	9.91E-1(1.35E-3)
	DTLZ3	1E0(1.82E-8)	1E0(2.44E-9)	1E0(9.97E-8)	1E0(6.77E-9)	**1E0(1.85E-9)**
	DTLZ4	1E0(6.97E-6)	**1E0(1.23E-5)**	1E0(4.78E-6)	1E0(8.37E-6)	9.93E-1(1.08E-3)
	WFG6	6.26E-1(2.55E-2)	6.42E-1(4.06E-2)	**7.01E-1(3.06E-2)**	6.49E-1(3.22E-2)	4.38E-1(5.4E-2)
	WFG7	4.18E-1(2.02E-2)	4.25E-1(3.15E-2)	3.97E-1(2.82E-2)	4.14E-1(1.77E-2)	**4.95E-1(6.8E-2)**
10	DTLZ1	1E0(1.09E-10)	1E0(1.75E-8)	1E0(5.58E-9)	1E0(1.76E-10)	**1E0(0E0)**
	DTLZ2	9.95E-1(2.57E-3)	9.97E-1(2.05E-3)	**9.98E-1(7.77E-4)**	9.96E-1(1.7E-3)	9.86E-1(1.99E-3)
	DTLZ3	1E0(1.58E-9)	1E0(6.67E-10)	1E0(1.23E-8)	1E0(2.99E-9)	**1E0(0E0)**
	DTLZ4	1E0(3.32E-7)	**1E0(2.37E-7)**	1E0(3.01E-7)	1E0(2.78E-7)	9.9E-1(8.42E-4)
	WFG6	7.29E-1(3.4E-2)	7.17E-1(3.49E-2)	**7.76E-1(2.1E-2)**	7.36E-1(2.64E-2)	4.59E-1(8.15E-2)
	WFG7	4.35E-1(2.21E-2)	4.42E-1(2.03E-2)	4.08E-1(1.89E-2)	4.26E-1(1.69E-2)	**4.99E-1(4.9E-2)**

6.7 Study of different H-MOPSO strategies

In this section, three different H-MOPSO strategies are evaluated.

1. One is the H-MOPSO used in previous sections, for select just the archiving method and using the leader selection method default Crowding Distance. For this strategy, it is used as alias the name of the selection method plus a suffix “FIXED”, for instance, the H-MOPSO using this first strategy and the random heuristic selection method will be called RANDOMFIXED.

2. In a second strategy, differently from previous sections, selected both archiving and leader selection methods are selected, as proposed in [22]. Each pair (archiving and leader selection) is seen as a low-level heuristic and is selected once for each iteration. That strategy is illustrated in Section 5.2, Algorithm 5.1. From now on, it is used the capitalized selection method name (e.g. RANDOM) as the acronym for this strategy using that method.

3. The third strategy is illustrated in Section 5.2, Algorithm 5.2. Where there are two instances of the selection method. One of them is responsible for selecting the archive method, once for each iteration. Moreover, the other is responsible for selecting the leader selection method, for each particle of the swarm, every iteration. To describe that strategy it is used as an alias the name of the method used to select archiving methods plus the name of the method used to select leader selection methods, for instance: RANDOMRANDOM. It is possible to use different heuristic selection methods (one for archiving and other for leader selection), but in this section it was used the same method in both.

In this section, each strategy for all four heuristic selection methods are compared and the best strategy for each selection method is chosen. The detailed results, for each problem instance and the objective number are presented in Appendix D. For the random heuristic selection method the strategy proposed by [22] (of selecting both methods together, once for each iteration) had the best IGD average ranking (Table 6.27), with statistical difference against the strategy selecting just the archiving method. The second strategy also had the best HV average ranking (Table 6.28). Finally, the H-MOPSO-RANDOM is the selected strategy for random heuristic selection, due to having the best IGD and HV average ranking.

Table 6.27: Mean ranking (and standard deviation) for RANDOMRANDOM, RANDOM and RANDOMFIXED for the IGD indicator

Algorithm	**Mean ranking (and standard deviation)**
RANDOM	**1.633(0.657)**
RANDOMRANDOM	1.933(0.727)
RANDOMFIXED	2.433(0.844)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.007446583$

Table 6.28: Mean ranking (and standard deviation) for RANDOMRANDOM, RANDOM and RANDOMFIXED for the HV indicator

Algorithm	**Mean ranking (and standard deviation)**
RANDOM	**1.854(0.770)**
RANDOMRANDOM	1.896(0.692)
RANDOMFIXED	2.250(0.878)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.3097335$

The second strategy (ACF) also had the best IGD average ranking for the Adaptive Choice Function heuristic selection method (Table 6.29), with statistical difference against the strategy selecting both methods separately (ACFACF). It also had the best HV average ranking (Table 6.30). Due to it, the second strategy is the selected strategy for ACF heuristic selection method.

Table 6.29: Mean ranking (and standard deviation) for ACFACF, ACF and ACFFIXED for the IGD indicator

Algorithm	**Mean ranking (and standard deviation)**
ACF	**1.633(0.706)**
ACFACF	2.567(0.667)
ACFFIXED	1.800(0.748)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.0005912135$

When evaluated for the UCB heuristic selection the third strategy (selecting archiving and leader selection separately — UCBUCB) had best average ranking with statistical

Table 6.30: Mean ranking (and standard deviation) for ACFACF, ACF and ACFFIXED for the HV indicator

Algorithm	Mean ranking (and standard deviation)
ACF	**1.708(0.676)**
ACFACF	2.333(0.745)
ACFFIXED	1.958(0.889)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.09301449$

difference against the strategy selecting just the archiving method for both IGD and HV (Tables 6.31 and 6.32). Moreover, it is the selected strategy for UCB heuristic selection.

Table 6.31: Mean ranking (and standard deviation) for UCBUCB, UCB and UCBFIXED for the IGD indicator

Algorithm	Mean ranking (and standard deviation)
UCBUCB	**1.700(0.690)**
UCBFIXED	2.400(0.879)
UCB	1.900(0.700)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.02024191$

Table 6.32: Mean ranking (and standard deviation) for UCBUCB, UCB and UCBFIXED for the HV indicator

Algorithm	Mean ranking (and standard deviation)
UCBUCB	**1.708(0.720)**
UCBFIXED	2.417(0.909)
UCB	1.875(0.582)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.03467619$

For the roulette-based heuristic selection method the selected strategy is the second, due to having the best average ranking with statistical difference against the third strategy in both IGD and HV (Tables 6.33 and 6.34).

In this section, three different H-MOPSO strategies were evaluated. The objective was to identify if there is some strategy that outperforms the others and to select the best strategy for each heuristic selection method. For the random-based, ACF and roulette-based heuristic selection methods, the strategy proposed in [22] has the best IGD and HV average ranking, and it is the selected strategy: RANDOM, ACF, and ROULETTE. Only for UCB method, the third strategy had the best IGD and HV average ranking, and in

Table 6.33: Mean ranking (and standard deviation) for ROULETTEROULETTE, ROULETTE and ROULETTEFIXED for the IGD indicator

Algorithm	Mean ranking (and standard deviation)
ROULETTEROULETTE	2.833(0.373)
ROULETTEFIXED	1.867(0.562)
ROULETTE	**1.300(0.586)**

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 1.473068e - 08$

Table 6.34: Mean ranking (and standard deviation) for ROULETTEROULETTE, ROULETTE and ROULETTEFIXED for the HV indicator

Algorithm	Mean ranking (and standard deviation)
ROULETTEROULETTE	2.542(0.644)
ROULETTEFIXED	1.875(0.781)
ROULETTE	**1.583(0.702)**

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.00305289$

none method, the fourth strategy was the best. It is possible to conclude that no strategy outperforms the other two in all heuristic selection methods, but it was found indications that selecting both archiving and leader selection methods (together or separately) may be better than selecting just archiving methods.

6.8 Selection of archiving and leader selection methods in MOPSO

In this section, four different H-MOPSO algorithms are compared to a state-of-art algorithm MOEA/D-DRA. In the evaluated methods, one selects leader selection and archiving methods separately using the UCB; The other three versions select both leader selection and archiving methods together, once for each iteration. They are H-MOPSO-ACF, H-MOPSO-ROULETTE and H-MOPSO-RANDOM.

In Table 6.39 the average IGD value of each algorithm for each problem instance and objective value are detailed. The best average value is well distributed among the algorithms, mainly for ACF, ROULETTE, and RANDOM. Usually, all five algorithms were statistically equivalent according to Kruskal-Wallis test, except by WFG7 with 3 and 15 objectives.

The H-MOPSO using ACF got the best average ranking for IGD (Table 6.35). The

Table 6.35: Mean ranking (and standard deviation) for UCBUCB, ACF, ROULETTE, RANDOM and MOEADDRA for the IGD indicator

Algorithm	Mean ranking (and standard deviation)
ACF	**2.233(1.086)**
MOEADDRA	4.567(1.116)
RANDOM	2.500(1.118)
UCBUCB	2.700(0.936)
ROULETTE	3.000(1.438)

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 3.273113e-08$

Table 6.36: Multiple comparisons between groups after Friedman test for IGD indicator

Comparisons		observed dif.	critical dif.	difference
ACF	UCBUCB	14	34.379	
ACF	MOEADDRA	70	34.379	✓
ACF	RANDOM	8	34.379	
ACF	ROULETTE	23	34.379	
UCBUCB	MOEADDRA	56	34.379	✓
UCBUCB	RANDOM	6	34.379	
UCBUCB	ROULETTE	9	34.379	
MOEADDRA	RANDOM	62	34.379	✓
MOEADDRA	ROULETTE	47	34.379	✓
RANDOM	ROULETTE	15	34.379	

Comparisons: comparison between each algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

RANDOM was the second, and the UCBUCB the third. The ROULETTE algorithm got the fourth average ranking for IGD, with the highest standard deviation. All H-MOPSO variants got better average ranking than MOEA/D-DRA with a statistical difference for IGD (Table 6.36).

For the Hypervolume quality indicator, the H-MOPSO-ROULETTE got the best average value, with a statistical difference in five problem instances: DTLZ1, WFG6, and WFG7 with three objectives and WFG6 and WFG7 with five objectives. The MOEA/D-DRA got best average value with a statistical difference in WFG7 with ten objectives (Table 6.40). When evaluated the average ranking, the ROULETTE was the best algorithm, the ACF was the second, and the H-MOPSO using a random based heuristic selection was the third. The fourth was the UCBUCB, and the MOEA/D-DRA got the worst results. Although, just the H-MOPSO using ROULETTE got the best ranking with statistical difference against MOEA/D-DRA. In this experiment, the H-MOPSO using ROULETTE was better than using UCB with statistical difference (Table 6.38), for the Hypervolume quality indicator.

Table 6.37: Mean ranking (and standard deviation) for UCBUCB, ACF, ROULETTE, RANDOM and MOEADDRA for the HV indicator

s

Algorithm	**Mean ranking (and standard deviation)**
ACF	2.687(1.019)
MOEADDRA	3.896(1.744)
RANDOM	3.021(0.743)
UCBUCB	3.438(1.093)
ROULETTE	**1.958(1.274)**

the best mean ranking is shown with **bold face**
Friedman test p-value: $p = 0.0002284456$

Table 6.38: Multiple comparisons between groups after Friedman test for HV indicator

Comparisons		**observed dif.**	**critical dif.**	**difference**
ACF	UCBUCB	18.0	30.74951	
ACF	MOEADDRA	29.0	30.74951	
ACF	RANDOM	8.0	30.74951	
ACF	ROULETTE	17.5	30.74951	
UCBUCB	MOEADDRA	11.0	30.74951	
UCBUCB	RANDOM	10.0	30.74951	
UCBUCB	ROULETTE	35.5	30.74951	✓
MOEADDRA	RANDOM	21.0	30.74951	
MOEADDRA	ROULETTE	46.5	30.74951	✓
RANDOM	ROULETTE	25.5	30.74951	

Comparisons: comparison between each algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

Table 6.39: Mean (and standard deviation) for UCBUCB, ACF, ROULETTE, RANDOM and MOEADDRA for the IGD indicator

Obj.	problem	RANDOM	ACF	UCBUCB	ROULETTE	MOEADDRA
3	DTLZ1	7.45E-3(2.48E-3)	6.83E-3(3.75E-4)	6.94E-3(4.56E-4)	**6.69E-3(5.11E-4)**	4.89E-2(8.72E-2)
	DTLZ2	9.57E-3(7.15E-4)	**9.56E-3(5.87E-4)**	9.67E-3(7.07E-4)	9.66E-3(1.01E-3)	1.04E-2(5.71E-4)
	DTLZ3	1.41E-2(1.14E-2)	**9.83E-3(2.19E-3)**	1.19E-2(8.5E-3)	1.35E-2(1.1E-2)	7.77E-1(1.64E0)
	DTLZ4	9.4E-3(9.8E-4)	**9.34E-3(1.2E-3)**	9.42E-3(1.26E-3)	1.14E-2(3.23E-3)	2.53E-2(9.71E-3)
	WFG6	1.14E-2(2.19E-3)	1.06E-2(1.41E-3)	1.15E-2(1.38E-3)	**1E-2(8.04E-4)**	1.21E-2(5.51E-4)
	WFG7	1.55E-2(2.29E-3)	1.51E-2(2.06E-3)	1.6E-2(2.6E-3)	**1.3E-2(4.86E-4)**	1.48E-2(9.16E-4)
5	DTLZ1	1.15E-2(5.01E-4)	1.19E-2(6.24E-4)	1.19E-2(8.19E-4)	**1.08E-2(5.93E-4)**	2.63E-2(1.16E-2)
	DTLZ2	1.76E-2(6.85E-4)	1.77E-2(7.89E-4)	1.77E-2(8.56E-4)	**1.71E-2(9.74E-4)**	2.17E-2(2.3E-4)
	DTLZ3	**1.79E-2(8.37E-4)**	1.85E-2(1.02E-3)	1.8E-2(7.2E-4)	1.98E-2(3.53E-3)	5.88E-2(4.45E-2)
	DTLZ4	1.55E-2(5.61E-4)	**1.55E-2(4.73E-4)**	1.56E-2(5.14E-4)	1.68E-2(1.6E-3)	2.4E-2(1.65E-3)
	WFG6	1.59E-2(5.25E-4)	1.57E-2(8.16E-4)	1.57E-2(4.22E-4)	**1.55E-2(8.82E-4)**	2.37E-2(2.28E-4)
	WFG7	1.93E-2(1.14E-3)	1.86E-2(6.25E-4)	1.9E-2(1E-3)	**1.83E-2(8.68E-4)**	3.27E-2(2.69E-3)
8	DTLZ1	2.61E-2(1.91E-3)	2.52E-2(1.7E-3)	2.52E-2(1.36E-3)	**2.47E-2(1.44E-3)**	3.12E-2(1.35E-3)
	DTLZ2	**4.71E-2(2.69E-3)**	4.78E-2(2.44E-3)	4.74E-2(1.9E-3)	5.08E-2(3.72E-3)	5.22E-2(1.59E-3)
	DTLZ3	5.41E-2(5.43E-3)	**5.14E-2(3.87E-3)**	5.22E-2(3.91E-3)	5.35E-2(5.13E-3)	8.2E-2(8.05E-2)
	DTLZ4	**3.57E-2(3.13E-3)**	3.6E-2(2.86E-3)	3.57E-2(3.11E-3)	3.84E-2(2.53E-3)	5.81E-2(3.53E-3)
	WFG6	4.16E-2(3.71E-3)	**4.14E-2(2.81E-3)**	4.25E-2(3.96E-3)	4.36E-2(3.67E-3)	4.51E-2(3.06E-3)
	WFG7	**4.51E-2(3.15E-3)**	4.52E-2(2.95E-3)	4.62E-2(4.63E-3)	4.75E-2(3.98E-3)	6.03E-2(7.18E-3)
10	DTLZ1	2.03E-2(1.03E-3)	2.03E-2(8.68E-4)	2.05E-2(8.85E-4)	**1.93E-2(9.84E-4)**	2.6E-2(3.47E-4)
	DTLZ2	**4.07E-2(2.37E-3)**	4.07E-2(2.74E-3)	4.12E-2(2.44E-3)	4.46E-2(1.8E-3)	4.44E-2(5.72E-4)
	DTLZ3	4.3E-2(2.54E-3)	**4.18E-2(2E-3)**	4.34E-2(3.09E-3)	4.45E-2(4.29E-3)	4.57E-2(1.48E-3)
	DTLZ4	2.73E-2(1.34E-3)	2.72E-2(1.22E-3)	**2.7E-2(9.31E-4)**	2.97E-2(2.2E-3)	4.91E-2(2.6E-3)
	WFG6	3.76E-2(3.46E-3)	**3.68E-2(3.29E-3)**	3.69E-2(2.78E-3)	3.82E-2(2.73E-3)	4.47E-2(2.2E-3)
	WFG7	5.98E-2(3.07E-3)	5.93E-2(4.77E-3)	**5.91E-2(3.39E-3)**	6.46E-2(4.88E-3)	6.62E-2(6.96E-3)
15	DTLZ1	**4.93E-2(1.88E-3)**	5.02E-2(1.85E-3)	5.04E-2(2.57E-3)	5.17E-2(2.16E-3)	5.37E-2(1.37E-3)
	DTLZ2	8.81E-2(2.77E-3)	8.71E-2(3.54E-3)	8.82E-2(3.19E-3)	9.03E-2(2.94E-3)	**8.6E-2(2.14E-3)**
	DTLZ3	8.95E-2(3.42E-3)	4.56E-1(6.95E-1)	**8.9E-2(2.15E-3)**	9.27E-2(3.99E-3)	1.67E-1(2.53E-1)
	DTLZ4	5.27E-2(2.6E-3)	**5.2E-2(2.81E-3)**	5.31E-2(3.5E-3)	5.47E-2(3.75E-3)	8.99E-2(2.35E-3)
	WFG6	**9.42E-2(1.04E-2)**	1.07E-1(1.87E-2)	9.72E-2(1.08E-2)	9.52E-2(8.29E-3)	1.22E-1(1.72E-2)
	WFG7	1.89E-1(1.07E-2)	1.87E-1(1.53E-2)	1.84E-1(1.38E-2)	2.05E-1(1.63E-2)	**1.18E-1(1.08E-2)**

Table 6.40: Mean (and standard deviation) for UCBUCB, ACF, ROULETTE, RANDOM and MOEADDRA for the HV indicator

Obj.	problem	RANDOM	ACF	UCBUCB	ROULETTE	MOEADDRA
3	DTLZ1	1E0(3.82E-5)	1E0(6.2E-5)	1E0(4.3E-5)	**1E0(7.39E-9)**	1E0(4.21E-4)
	DTLZ2	8.06E-1(2.21E-3)	8.03E-1(7.64E-3)	8.06E-1(2.24E-3)	**8.08E-1(2.35E-3)**	7.87E-1(4.09E-3)
	DTLZ3	1E0(5.2E-6)	1E0(4.61E-6)	**1E0(3.86E-6)**	1E0(6.92E-6)	9.92E-1(2.29E-2)
	DTLZ4	4.75E-1(5.64E-3)	**4.77E-1(3.94E-3)**	4.76E-1(5.4E-3)	4.76E-1(6.71E-3)	4.13E-1(2.88E-2)
	WFG6	3.78E-1(1.37E-2)	3.8E-1(9.46E-3)	3.74E-1(6.12E-3)	**3.93E-1(7.87E-3)**	3.17E-1(9.59E-3)
	WFG7	3.96E-1(1.68E-2)	3.95E-1(1.6E-2)	3.86E-1(1.66E-2)	**4.25E-1(1.05E-2)**	3.66E-1(1.69E-2)
5	DTLZ1	1E0(7E-8)	1E0(8.11E-8)	1E0(3.65E-8)	**1E0(3.08E-11)**	1E0(4.79E-6)
	DTLZ2	9.9E-1(7.14E-4)	9.9E-1(5.7E-4)	9.88E-1(2.38E-3)	9.9E-1(3.02E-3)	**9.9E-1(2.98E-4)**
	DTLZ3	1E0(0E0)	**1E0(0E0)**	1E0(0E0)	1E0(1.79E-10)	1E0(2.77E-6)
	DTLZ4	9.9E-1(7.44E-4)	9.9E-1(3.17E-4)	9.9E-1(5.8E-4)	**9.91E-1(7.12E-4)**	9.89E-1(6.88E-4)
	WFG6	5.33E-1(2.24E-2)	5.45E-1(3.27E-2)	5.43E-1(2.08E-2)	**5.77E-1(1.43E-2)**	3.49E-1(1.96E-2)
	WFG7	5.09E-1(1.36E-2)	5.2E-1(1.38E-2)	5.09E-1(1.49E-2)	5.43E-1(1.48E-2)	**5.9E-1(4.62E-2)**
8	DTLZ1	1E0(7.48E-9)	1E0(3E-7)	1E0(1.3E-7)	1E0(4.51E-9)	**1E0(1.18E-10)**
	DTLZ2	9.97E-1(2.02E-3)	9.97E-1(1.64E-3)	9.96E-1(2.3E-3)	**9.97E-1(1.1E-3)**	9.91E-1(1.32E-3)
	DTLZ3	1E0(2.24E-11)	**1E0(0E0)**	1E0(0E0)	1E0(0E0)	1E0(1.61E-9)
	DTLZ4	1E0(2.78E-5)	1E0(1.58E-5)	1E0(3.91E-5)	**1E0(1.4E-5)**	9.92E-1(1.16E-3)
	WFG6	5.74E-1(4.3E-2)	5.71E-1(3.54E-2)	5.6E-1(3.75E-2)	**5.8E-1(5E-2)**	4.17E-1(4.91E-2)
	WFG7	4.27E-1(4.25E-2)	4.16E-1(3.26E-2)	4.14E-1(3.14E-2)	4.14E-1(2.82E-2)	**4.48E-1(6.58E-2)**
10	DTLZ1	1E0(8.47E-10)	1E0(1.23E-9)	1E0(2.83E-9)	1E0(3.27E-10)	**1E0(0E0)**
	DTLZ2	9.98E-1(9.83E-4)	**9.99E-1(7.56E-4)**	9.98E-1(8.78E-4)	9.98E-1(1.22E-3)	9.88E-1(1.59E-3)
	DTLZ3	1E0(0E0)	**1E0(0E0)**	1E0(0E0)	1E0(3.08E-11)	1E0(0E0)
	DTLZ4	1E0(6.76E-6)	1E0(4.26E-6)	1E0(5.16E-6)	**1E0(2.84E-6)**	9.76E-1(1.25E-3)
	WFG6	6.5E-1(4.54E-2)	6.48E-1(4.73E-2)	6.5E-1(6.07E-2)	**6.83E-1(5.61E-2)**	4.51E-1(8.16E-2)
	WFG7	4.48E-1(3.43E-2)	4.45E-1(3.11E-2)	4.43E-1(1.9E-2)	4.48E-1(3.79E-2)	**5.25E-1(5.67E-2)**

In this section, four H-MOPSO variants were evaluated. Those variants are based on four heuristic selection methods, selected and configured in previous sections. The H-MOPSO algorithms are compared against a state-of-art algorithm MOEA/D-DRA. It is possible to conclude that no heuristic selection method outperforms any other in both IGD and HV. Also, no method outperforms all others in any indicator. When compared against a state-of-art algorithm the H-MOPSO variants got competing results. For the IGD indicator, all methods got best average ranking than MOEA/D-DRA, with statistical difference. All methods got best average ranking than MOEA/D-DRA, but just ROULETTE with statistical difference.

6.9 Comparison between the H-MOPSO variant with the best results and the state-of-art

In this section, a comparison between H-MOPSO variation that obtained the best results in previous studies and a state-of-art algorithm, MOEA/D-DRA, is presented. To do so, the H-MOPSO-ACF was selected. It is an H-MOPSO variant that selects both leader selection and archiving methods as a single low-level heuristic using the Adaptive Choice Function heuristic selection. It was chosen due to on a previous study when compared with other three H-MOPSO variants and a state-of-art algorithm. In that study, H-MOPSO-ACF had the best IGD mean ranking and the second best for the Hypervolume quality indicator. Moreover, H-MOPSO-ACF had the best mean of IGD and Hypervolume ranking. Despite its good results, it was statistically equivalent to the other H-MOPSO variants in both IGD and Hypervolume mean ranking. This analysis is made to get a detailed review of the best variant and highlight its results.

When evaluated the IGD indicator of H-MOPSO-ACF *vs* MOEA/D-DRA, illustrated in Table 6.46, H-MOPSO-ACF had the best value, with a statistical difference in almost all problem instances. Besides, MOEA/D-DRA had the best or equivalent result in four problem instances, with a statistical difference in just one. When evaluated the mean ranking for the IGD indicator H-MOPSO-ACF had a better value, with a statistical

Table 6.41: Mean (and standard deviation) for H-MOPSO-ACF and MOEADDRA for the IGD indicator

Obj.	problem	H-MOPSO-ACF	MOEADDRA
3	DTLZ1	**6.83E-3(3.75E-4)**	4.89E-2(8.72E-2)
	DTLZ2	**9.56E-3(5.87E-4)**	1.04E-2(5.71E-4)
	DTLZ3	**9.83E-3(2.19E-3)**	7.77E-1(1.64E0)
	DTLZ4	**9.34E-3(1.2E-3)**	2.53E-2(9.71E-3)
	WFG6	**1.06E-2(1.41E-3)**	1.21E-2(5.51E-4)
	WFG7	1.51E-2(2.06E-3)	**1.48E-2(9.16E-4)**
5	DTLZ1	**1.19E-2(6.24E-4)**	2.63E-2(1.16E-2)
	DTLZ2	**1.77E-2(7.89E-4)**	2.17E-2(2.3E-4)
	DTLZ3	**1.85E-2(1.02E-3)**	5.88E-2(4.45E-2)
	DTLZ4	**1.55E-2(4.73E-4)**	2.4E-2(1.65E-3)
	WFG6	**1.57E-2(8.16E-4)**	2.37E-2(2.28E-4)
	WFG7	**1.86E-2(6.25E-4)**	3.27E-2(2.69E-3)
8	DTLZ1	**2.52E-2(1.7E-3)**	3.12E-2(1.35E-3)
	DTLZ2	**4.78E-2(2.44E-3)**	5.22E-2(1.59E-3)
	DTLZ3	**5.14E-2(3.87E-3)**	8.2E-2(8.05E-2)
	DTLZ4	**3.6E-2(2.86E-3)**	5.81E-2(3.53E-3)
	WFG6	**4.14E-2(2.81E-3)**	4.51E-2(3.06E-3)
	WFG7	**4.52E-2(2.95E-3)**	6.03E-2(7.18E-3)
10	DTLZ1	**2.03E-2(8.68E-4)**	2.6E-2(3.47E-4)
	DTLZ2	**4.07E-2(2.74E-3)**	4.44E-2(5.72E-4)
	DTLZ3	**4.18E-2(2E-3)**	4.57E-2(1.48E-3)
	DTLZ4	**2.72E-2(1.22E-3)**	4.91E-2(2.6E-3)
	WFG6	**3.68E-2(3.29E-3)**	4.47E-2(2.2E-3)
	WFG7	**5.93E-2(4.77E-3)**	6.62E-2(6.96E-3)
15	DTLZ1	**5.02E-2(1.85E-3)**	5.37E-2(1.37E-3)
	DTLZ2	8.71E-2(3.54E-3)	**8.6E-2(2.14E-3)**
	DTLZ3	4.56E-1(6.95E-1)	**1.67E-1(2.53E-1)**
	DTLZ4	**5.2E-2(2.81E-3)**	8.99E-2(2.35E-3)
	WFG6	**1.07E-1(1.87E-2)**	1.22E-1(1.72E-2)
	WFG7	1.87E-1(1.53E-2)	**1.18E-1(1.08E-2)**

difference. H-MOPSO mean ranking was 1.133 as MOEA/D-DRA had 1.867, according to Table 6.44.

Table 6.42: Mean ranking (and standard deviation) for H-MOPSO-ACF and MOEAD-DRA for the IGD indicator

Algorithm	**Mean ranking (and standard deviation)**
H-MOPSO-ACF	**1.133(0.340)**
MOEADDRA	1.867(0.340)

the best mean ranking with statistical difference is shown with **bold face** and gray background
Friedman test p-value: $p = 5.903578e - 05$

Table 6.43: Multiple comparisons between groups after Friedman test for IGD indicator

Comparisons		observed dif.	critical dif.	difference
H-MOPSO-ACF	MOEADDRA	22	10.73516	✓

Comparisons: comparison between each algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

When evaluated the Hypervolume quality indicator, H-MOPSO-ACF had the best value, with a statistical difference, in most cases. However, MOEA/D-DRA had more best results in Hypervolume than in IGD, with six values better than or equivalent to H-MOPSO-ACF, of which four with a statistical difference. As in IGD, H-MOPSO had the best mean ranking in Hypervolume too, with a statistical difference. H-MOPSO ranking was 1.25 while for MOEA/D-DRA it was 1.75.

Table 6.44: Mean ranking (and standard deviation) for H-MOPSO-ACF and MOEAD-DRA for the HV indicator

Algorithm	Mean ranking (and standard deviation)
H-MOPSO-ACF	**1.250(0.433)**
MOEADDRA	1.750(0.433)

the best mean ranking with statistical difference is shown with **bold face** and gray background
Friedman test p-value: $p = 0.01430588$

Table 6.45: Multiple comparisons between groups after Friedman test for HV indicator

Comparisons		observed dif.	critical dif.	difference
H-MOPSO-ACF	MOEADDRA	12	9.601823	✓

Comparisons: comparison between each algorithm and the others
observed dif.: difference between the accumulated ranking of the compared algorithms
critical dif.: critical difference to consider the samples statistically different
difference: ✓ if there is statistical difference

In this section, it was presented a comparison between H-MOPSO variant with the best results so far and a state-of-art multi-objective algorithm. On the results, H-MOPSO had the best value, with a statistical difference in most problem instances, for both IGD and Hypervolume. When evaluated the mean ranking H-MOPSO also had the best value, with a statistical difference. With values close to one, 1.133 for IGD and 1.25 to Hypervolume.

Table 6.46: Mean (and standard deviation) for H-MOPSO-ACF and MOEADDRA for the HV indicator

Obj.	problem	H-MOPSO-ACF	MOEADDRA
3	DTLZ1	**1E0(6.2E-5)**	1E0(4.21E-4)
	DTLZ2	**8.03E-1(7.64E-3)**	7.87E-1(4.09E-3)
	DTLZ3	**1E0(4.61E-6)**	9.92E-1(2.29E-2)
	DTLZ4	**4.59E-1(4.07E-3)**	3.93E-1(2.98E-2)
	WFG6	**3.8E-1(9.47E-3)**	3.17E-1(9.6E-3)
	WFG7	**3.9E-1(1.63E-2)**	3.61E-1(1.71E-2)
5	DTLZ1	**1E0(8.11E-8)**	1E0(4.79E-6)
	DTLZ2	9.89E-1(5.93E-4)	**9.9E-1(3.12E-4)**
	DTLZ3	**1E0(0E0)**	1E0(7.34E-6)
	DTLZ4	**9.86E-1(4.5E-4)**	9.84E-1(9.76E-4)
	WFG6	**5.63E-1(3.38E-2)**	3.61E-1(2.03E-2)
	WFG7	5.19E-1(1.39E-2)	**5.88E-1(4.64E-2)**
8	DTLZ1	1E0(6.01E-6)	**1E0(2.56E-9)**
	DTLZ2	**9.97E-1(1.71E-3)**	9.9E-1(1.44E-3)
	DTLZ3	**1E0(0E0)**	1E0(3.81E-8)
	DTLZ4	**9.99E-1(4.72E-4)**	9.32E-1(1.76E-2)
	WFG6	**5.58E-1(3.65E-2)**	3.99E-1(5.08E-2)
	WFG7	3.98E-1(2.95E-2)	**4.2E-1(6.67E-2)**
10	DTLZ1	1E0(3.17E-9)	**1E0(3.66E-11)**
	DTLZ2	**9.98E-1(7.84E-4)**	9.85E-1(1.96E-3)
	DTLZ3	**1E0(0E0)**	1E0(1.16E-10)
	DTLZ4	**1E0(4.65E-6)**	9.75E-1(1.28E-3)
	WFG6	**6.34E-1(4.7E-2)**	4.35E-1(8.35E-2)
	WFG7	4.32E-1(2.92E-2)	**4.8E-1(5.07E-2)**

In other words, H-MOPSO obtained competitive results when compared with a state-of-art algorithm.

CHAPTER 7

CONCLUSIONS

In this work, the incorporation of hyper-heuristic on the MOPSO algorithm is evaluated. The objective is to improve the MOPSO convergence and diversity for Many-Objective problems and reduce the quality deterioration. The hypothesis of improving MOPSO using hyper-heuristic is proposed in [22], and the results obtained are encouraging. Based on [22] different hyper-heuristic approaches are studied, with focus on the Multi-Armed Bandit and Choice Function based heuristic selection methods. Those two methods were selected based on its good results presented in the literature.

In the first study, an experiment made in [22] is reproduced. That experiment compares original H-MOPSO with each low-level heuristic. As [22], it is concluded that H-MOPSO, in general, have a better performance than any low-level heuristic applied individually. Also, H-MOPSO have a better average ranking than any low-level heuristic, supporting the conclusions of [22]. In a second study, two different Choice-Function based heuristic selection were compared: the Adaptive Choice Function (ACF) [9] and the Simplified Choice Function (SCF) [11]. ACF had better IGD values, mainly for problems with 8 or more objectives, on the other hand, SCF had better Hypervolume values, primarily for three and five objectives. Also, the average distance between the reference points and the Pareto front was compared, to have a better evaluation of the algorithms. It was evaluated mainly the instances where IGD and Hypervolume disagree. ACF usually had a smaller deviation; it means that the Pareto front approximations generated by ACF were more distributed. Also, both methods had a good convergence towards the true Pareto front. However, SCF had a better convergence than ACF in some points of the front, which is responsible for increasing its Hypervolume value. Finally, the ACF was selected as the CF-based method to be used, due to its good convergence and diversity.

Also, three Multi-Armed Bandit based methods were evaluated: the FRRMAB that uses UCB1 function [26] and two variations, the FRRMAB-UCBV and FRRMAB-UCBTuned [10]. In general, all methods had similar performance. Although, the version using UCB1 had a better average ranking in both IGD and Hypervolume and was selected to be used in other studies. Then, the ACF and FRRMAB-UCB1 parameters were configured. When evaluated in an H-MOPSO selecting just archiving the heuristic selection methods had similar performance, with a slight advantage for H-MOPSO-ACF. Besides, all four heuristic selection methods had better IGD and HV average ranking than the state-of-art algorithm MOEA/D-DRA, with a statistical difference for IGD.

Also, different H-MOPSO strategies were studied: H-MOPSO selecting just the archiving method; H-MOPSO selecting both archiving and leader selection methods together, once for each iteration; and the version that selects both methods separately. Moreover, for each heuristic selection method, the best strategy was selected. Except by UCB, that had better results with the third strategy, all other methods had better results with the second strategy. It is possible to conclude then, that selection of archiving and leader selection methods (independent of structure) is better than selecting just archiving.

Finally, the four H-MOPSO final versions were compared to each other and the MOEA/D-DRA. It is possible to conclude that, no heuristic selection method outperforms any other in both IGD and Hypervolume. Also, no method outperforms all others in IGD or Hypervolume. H-MOPSO (using any heuristic method) had better results than MOEA/D-DRA, with a statistical difference for IGD. It is possible to conclude that the use of hyper-heuristic is capable of increasing the MOPSO performance, achieving competitive results against the state-of-art algorithm MOEA/D-DRA.

Also, it is possible that a better tuning of the algorithms, mainly the Choice Function and Multi-Armed Bandit based (including a previous configuration for the studies that select the CF and MAB-based methods to be used) may increase the algorithm's performance. However, usually on the studies, any evaluated method, with any configuration, are statistically equivalent. This behavior supports that: a more refined tuning would have high computational cost and potentially would not increase the algorithm's performance significantly. Another comment is about the roulette-based H-MOPSO. Usually, it had the highest standard deviation of the mean ranking among H-MOPSO variants. It shows that H-MOPSO-ROULETTE had less uniform performance when compared with other H-MOPSO, with excellent results in some problem instances, and performing poorly in others. Also, H-MOPSO-ACF, that despite the statistical analysis, had the best IGD average ranking and second best for Hypervolume.

Also, some topics to be considered in future works are proposed. The first of them is to use some automatic tuning of the algorithms parameters, such as racing techniques, to eliminate candidate configurations as soon as possible. The use of automatic tuning can improve the configuration accuracy and reduce the effort spent on that task. Another proposal is to evaluate the behavior of the heuristic selection methods along the search. The objective is to use this information to increase H-MOPSO performance. Also, it is possible to evaluate the selection of archiving and leader selection methods separately using different heuristic selection methods. The objective is to assess if the use of different heuristic selection methods, properly chosen and configured to select a specific method, may increase H-MOPSO performance.

BIBLIOGRAPHY

[1] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu, "Hyper-heuristic," *J Oper Res Soc*, vol. 64, no. 12, pp. 1695–1724, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1057/jors.2013.71

[2] M. Misir, "Intelligent hyper-heuristic: A tool for solving generic optimisation problems," Ph.D. dissertation, Department of Computer Science, KU Leuven, 2012.

[3] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation).* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[4] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 13:1–13:35, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2792984

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, JuneJun. 2008, pp. 2419–2426.

[6] Castro Jr., Olacir R. and Pozo, Aurora, "Using hyper-heuristic to select leader and archiving methods for many-objective problems," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Springer International Publishing, 2015, vol. 9018, pp. 109–123. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15934-8_8

[7] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[8] R. Hernández Gómez and C. A. Coello Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '15. New York, NY, USA: ACM, 2015, pp. 679–686. [Online]. Available: http://doi.acm.org/10.1145/2739480.2754776

[9] R. A. Gonçalves, J. l. Kuk, C. P. Almeida, and S. M. Venske, "MOEA/D-HH: A hyper-heuristic for multi-objective problems," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Springer International Publishing, 2015, vol. 9018, pp. 94–108. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15934-8_7

[10] R. A. Gonçalves, C. P. Almeida, and A. Pozo, "Upper confidence bound (UCB) algorithms for adaptive operator selection in MOEA/D," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Springer International Publishing, 2015, vol. 9018, pp. 411–425. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15934-8_28

[11] G. Guizzo, G. M. Fritsche, S. R. Vergilio, and A. T. R. Pozo, "A hyper-heuristic for the multi-objective integration and test order problem," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, ser. GECCO '15. New York, NY, USA: ACM, 2015, pp. 1343–1350. [Online]. Available: http://doi.acm.org/10.1145/2739480.2754725

[12] M. Maashi, E. Özcan, and G. Kendall, "A multi-objective hyper-heuristic based on choice function," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4475–4493, 2014.

[13]

[14] M. Laumanns and R. Zenklusen, "Stochastic convergence of random search

methods to fixed size Pareto front approximations," *European Journal of Operational Research*, vol. 213, no. 2, pp. 414–421, sep 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221711002761

[15] M. Lòpez-Ibáñez, J. Knowles, and M. Laumanns, "On sequential online archiving of objective vectors," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds. Springer Berlin Heidelberg, 2011, vol. 6576, pp. 46–60. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19893-9_4

[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature PPSN VI*, ser. Lecture Notes in Computer Science, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 2000, vol. 1917, pp. 849–858. [Online]. Available: http://dx.doi.org/10.1007/3-540-45356-3_83

[17] A. Britto and A. Pozo, "Using archiving methods to control convergence and diversity for many-objective problems in particle swarm optimization," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, Jun. 2012, pp. 1–8.

[18] A. Nebro, J. Durillo, J. Garcia-Nieto, C. Coello Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Computational intelligence in multi-criteria decision-making, 2009. mcdm '09. ieee symposium on*, Mar. 2009, pp. 66–73.

[19] N. Padhye, J. Branke, and S. Mostaghim, "Empirical comparison of MOPSO methods - guide selection and diversity preservation -," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 2516–2523.

[20] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, April 2003, pp. 26–33.

[21] O. Castro Jr., A. Britto, and A. Pozo, "A comparison of methods for leader selection in many-objective problems," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*, Jun. 2012, pp. 1–8.

[22] O. R. Castro Jr. and A. Pozo, "A MOPSO based on hyper-heuristic to optimize many-objective problems," in *Swarm Intelligence (SIS), 2014 IEEE Symposium on*, Dec. 2014, pp. 1–8.

[23] J. Drake, E. Ozcan, and E. Burke, "An improved choice function heuristic selection for cross domain heuristic search," in *Parallel Problem Solving from Nature - PPSN XII*, ser. Lecture Notes in Computer Science, C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, Eds. Springer Berlin Heidelberg, 2012, vol. 7492, pp. 307–316. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32964-7_31

[24] P. Cowling, G. Kendall, and E. Soubeiga, "A Hyperheuristic Approach to Scheduling a Sales Summit," in *Practice and Theory of Automated Timetabling III*, ser. Lecture Notes in Computer Science, E. Burke and W. Erben, Eds. Springer Berlin Heidelberg, 2001, vol. 2079, pp. 176–190.

[25] J. Blazewicz, E. Burke, G. Kendall, W. Mruczkiewicz, C. Oguz, and A. Swiercz, "A hyper-heuristic approach to sequencing by hybridization of dna sequences," *Annals of Operations Research*, vol. 207, no. 1, pp. 27–41, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10479-011-0927-y

[26] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 1, pp. 114–130, Feb 2014.

[27] A. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag, "Analyzing bandit-based adaptive operator selection mechanisms," *Annals of Mathematics and Artificial Intelligence*, vol. 60, no. 1-2, pp. 25–64, 2010.

[28] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained mop test instances," in *Proceedings of the Eleventh Conference on Congress on Evolutionary Computation*, ser. CEC'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 203–208. [Online]. Available: http://dl.acm.org/citation.cfm?id=1689599.1689626

[29] G. M. Fritsche and A. T. R. Pozo, "Configuração automática de parâmetros: um estudo de caso," *Encontro Nacional de Inteligência Artificial e Computacional*, Oct. 2014.

[30] G. Fritsche, A. Strickler, A. Pozo, and R. Santana, "Capturing Relationships in Multi-Objective Optimization," in *Brazilian Conference on Intelligent Systems (BRACIS 2015), Natal, Brazil*, 2015.

[31] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[32] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[33] C. A. C. Coello and M. Reyes-Sierra, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol. 2, 2006.

[34] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[35] H. Fisher and G. L. Thompson, "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules," pp. 225–251, 1963.

[36] E. Burke, M. Gendreau, M. Hyde, G. Kendall, B. McCollum, G. Ochoa, A. Parkes, and S. Petrovic, "The cross-domain heuristic search challenge - an international

research competition," in *Learning and Intelligent Optimization*, ser. Lecture Notes in Computer Science, C. Coello, Ed. Springer Berlin Heidelberg, 2011, vol. 6683, pp. 631–634. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25566-3_49

[37] E. Ozcan, Y. Bykov, M. Birben, and E. Burke, "Examination timetabling using late acceptance hyper-heuristic," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 997–1004.

[38] A. Fialho, "Adaptive operator selection for optimization," Ph.D. dissertation, Université Paris-Sud XI, Orsay, France, Dec. 2010.

[39] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the R2 indicator," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '12. New York, NY, USA: ACM, 2012, pp. 465–472. [Online]. Available: http://doi.acm.org/10.1145/2330163.2330230

[40] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, pp. 760–771, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997811001219

[41] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, ser. Advanced Information and Knowledge Processing, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer London, 2005, pp. 105–145. [Online]. Available: http://dx.doi.org/10.1007/1-84628-137-7_6

[42] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *Trans. Evol. Comp*, vol. 10, no. 5, pp. 477–506, Oct. 2006. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2005.861417

[43] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical As-*

sociation, vol. 47, no. 260, pp. 583–621, 1952. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441

[44] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01621459.1937.10503522

[45] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 1, pp. 86–95, Feb 2012.

[46] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," in *Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 2014 IEEE Symposium on*, Dec 2014, pp. 170–177.

[47] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *Cybernetics, IEEE Transactions on*, vol. 44, no. 12, pp. 2391–2404, Dec 2014.

APPENDIX A

R2 RESULTS OF THE PRELIMINARY RESEARCH

In Tables A.1 and A.2 are shown the mean (and standard deviation) for the R2 indicator for each low-level heuristic and the hyper-heuristic method based on roulette. The ROULETTE algorithm had the best mean performance based on the R2 indicator in most cases (22 from 42 experiments): mainly in DTLZ1 and DTLZ4. It is used the Kruskal-Wallis statistical test to evaluate if the best performance is statistically different from all others algorithms performances: in all cases the best result had no difference to, at least, some other algorithm.

Table A.1: Mean (and standard deviation) for each low-level heuristic and the hyper-heuristic based on roulette for the R2 indicator (Part I)

Obj.	problem	ROULETTE	CDCD	CDNWSUM	CDSIGMA	IDEALCD
2	DTLZ1	**1.23E-3(3.33E-4)**	2.21E-3(5.85E-3)	4.04E-3(1.4E-2)	3.15E-3(9.34E-3)	2.96E-3(2.34E-3)
	DTLZ2	1.5E-1(6.14E-5)	**1.5E-1(6.71E-5)**	1.53E-1(3.1E-3)	1.52E-1(2.1E-3)	2.47E-1(4.26E-2)
	DTLZ3	**2.31E-3(1.17E-3)**	3.46E-3(6.39E-3)	4.46E-3(1.56E-2)	3.28E-3(9.33E-3)	3.44E-3(4.3E-3)
	DTLZ4	1.34E-1(7.54E-5)	**1.33E-1(5.59E-5)**	1.4E-1(5.95E-3)	1.4E-1(6.54E-3)	2.03E-1(4.5E-2)
	DTLZ5	1.47E-1(7.49E-5)	**1.46E-1(4.4E-5)**	1.49E-1(3.39E-3)	1.48E-1(2.3E-3)	2.49E-1(6.11E-2)
	DTLZ6	**2.68E-2(1.08E-2)**	2.8E-2(2.27E-2)	3.57E-2(3.22E-2)	3.8E-2(2.86E-2)	5.14E-2(4.71E-2)
	DTLZ7	**5.72E-2(1.1E-4)**	6.72E-2(5.47E-2)	7.11E-2(6.14E-2)	8.25E-2(5.96E-2)	1.12E-1(5.89E-2)
3	DTLZ1	**5.28E-4(5.41E-5)**	8.1E-4(1.47E-3)	3.81E-3(1.57E-2)	1.86E-3(6.13E-3)	1.71E-3(3.07E-3)
	DTLZ2	2.16E-2(1.46E-4)	2.18E-2(1.17E-4)	2.29E-2(1.05E-3)	2.28E-2(1.58E-3)	6.97E-2(2.28E-2)
	DTLZ3	**2.88E-4(8.06E-5)**	3.88E-4(4.26E-4)	1.44E-3(6.15E-3)	5.37E-4(1.6E-3)	3.9E-4(3.35E-4)
	DTLZ4	**2.84E-2(5.65E-4)**	2.86E-2(6.01E-4)	3.13E-2(5.71E-3)	2.96E-2(6.02E-3)	8.11E-2(5.19E-2)
	DTLZ5	1.25E-1(9.52E-5)	**1.25E-1(4.45E-5)**	1.28E-1(1.54E-3)	1.26E-1(1.11E-3)	2.73E-1(4.38E-2)
	DTLZ6	2.03E-2(5.12E-3)	2.06E-2(1.05E-2)	3.5E-2(1.08E-2)	2.84E-2(1.21E-2)	2.9E-2(2.64E-2)
	DTLZ7	**4.37E-2(6.37E-3)**	4.86E-2(3.67E-2)	8.94E-2(3.4E-2)	6.36E-2(3.69E-2)	5.98E-2(4.22E-2)
5	DTLZ1	**5.47E-6(1.02E-6)**	3.41E-5(8.26E-5)	1.41E-4(6.99E-4)	5.68E-5(2.61E-4)	3.33E-5(4.33E-5)
	DTLZ2	**7.59E-4(2.96E-5)**	9.38E-4(2.93E-4)	2.15E-3(1.6E-3)	4.35E-3(2.17E-3)	4.86E-2(1.13E-2)
	DTLZ3	**2.56E-6(1.7E-6)**	4.38E-5(3.71E-5)	1.63E-5(6.02E-5)	3.02E-5(6.5E-5)	3.1E-5(1.58E-5)
	DTLZ4	**8.63E-4(4.63E-5)**	9E-4(4.07E-5)	1.51E-3(1.99E-3)	1.08E-3(8.85E-4)	2.79E-2(2.44E-2)
	DTLZ5	**2.07E-2(2.69E-4)**	2.14E-2(7.3E-4)	2.4E-2(1.03E-3)	2.34E-2(1.1E-3)	5.6E-2(1.53E-3)
	DTLZ6	7.82E-3(1.35E-3)	**6.66E-3(1.9E-3)**	1.01E-2(4.82E-3)	1.31E-2(3.98E-3)	1.62E-2(2.63E-2)
	DTLZ7	4.27E-2(1.04E-2)	5.27E-2(1.55E-2)	7.17E-2(1.47E-2)	**2.76E-2(2.03E-2)**	3.22E-2(2.52E-2)
10	DTLZ1	1.25E-5(4.3E-6)	5.32E-4(2.76E-4)	1.85E-5(6.17E-6)	1.36E-5(2.97E-6)	1.58E-5(2.72E-5)
	DTLZ2	2.39E-3(1.38E-4)	4.1E-3(5.07E-4)	3.81E-3(3.71E-4)	3.07E-3(2.13E-4)	2.12E-2(7.16E-3)
	DTLZ3	5.24E-5(6.79E-5)	8.95E-4(3.06E-4)	1.09E-3(4.58E-4)	7.62E-4(2.27E-4)	3.1E-5(1.51E-5)
	DTLZ4	**2.08E-3(1.17E-4)**	3.15E-3(3.83E-4)	3.72E-3(4.35E-4)	3.19E-3(4.07E-4)	2.54E-2(8.54E-3)
	DTLZ5	6.84E-3(5.18E-4)	7.04E-3(4.8E-4)	7.94E-3(5.56E-4)	7.63E-3(6.53E-4)	2.85E-2(4.35E-4)
	DTLZ6	3.06E-3(4.72E-4)	**2.62E-3(1.16E-3)**	4.02E-3(1.73E-3)	5.27E-3(1.52E-3)	1.35E-2(2.15E-2)
	DTLZ7	2.73E-2(4.46E-3)	3.6E-2(3.21E-2)	4.05E-2(3.04E-2)	4.84E-2(2.22E-2)	2.85E-2(2.72E-2)
15	DTLZ1	**5.12E-6(1.94E-6)**	2.68E-4(1.38E-4)	3.64E-5(7.89E-5)	1.28E-5(2.05E-5)	1.95E-5(7.2E-5)
	DTLZ2	**7.76E-4(4.68E-5)**	1.41E-3(1.72E-4)	1.17E-3(1.14E-4)	9.98E-4(1.12E-4)	1.27E-2(6.47E-3)
	DTLZ3	1.12E-5(1.78E-5)	4.12E-4(9.98E-5)	3.96E-4(1.1E-4)	2.57E-4(8.66E-5)	1.95E-5(1.14E-5)
	DTLZ4	**7.4E-4(4.6E-5)**	1.22E-3(1.51E-4)	1.38E-3(1.75E-4)	1.23E-3(1.59E-4)	1.52E-2(7.49E-3)
	DTLZ5	2.15E-3(1.45E-4)	2.29E-3(2.56E-4)	2.47E-3(2.46E-4)	2.4E-3(2.77E-4)	1.9E-2(7.47E-15)
	DTLZ6	9.03E-4(1.57E-4)	**8.33E-4(4.52E-4)**	1.25E-3(7.45E-4)	1.62E-3(7.09E-4)	7.61E-3(1.14E-2)
	DTLZ7	**1.74E-2(2.39E-3)**	2.28E-2(4.3E-2)	2.89E-2(4.21E-2)	3.76E-2(3.57E-2)	2.78E-2(4.08E-2)
20	DTLZ1	**2.89E-6(1.6E-6)**	1.65E-4(8.69E-5)	7.53E-6(6.23E-6)	9.17E-6(1.41E-5)	2.43E-5(8.1E-5)
	DTLZ2	4.49E-4(3.26E-5)	7.94E-4(9.46E-5)	6.65E-4(6.61E-5)	5.64E-4(7.66E-5)	1.06E-2(4.85E-3)
	DTLZ3	1.48E-5(2.03E-5)	2.68E-4(6.19E-5)	2.38E-4(7.03E-5)	1.49E-4(5.54E-5)	1.61E-5(8.78E-6)
	DTLZ4	**4.41E-4(2.59E-5)**	7.31E-4(8.6E-5)	8.2E-4(1.25E-4)	7.04E-4(1.01E-4)	1.14E-2(5.83E-3)
	DTLZ5	**1.24E-3(7.31E-5)**	1.36E-3(1.5E-4)	1.43E-3(1.5E-4)	1.39E-3(1.82E-4)	1.43E-2(2.75E-17)
	DTLZ6	5.39E-4(8.44E-5)	**5.13E-4(2.71E-4)**	8.56E-4(4.73E-4)	1.04E-3(4.34E-4)	6.45E-3(8.51E-3)
	DTLZ7	**1.31E-2(1.83E-3)**	1.88E-2(4.61E-2)	2.36E-2(4.41E-2)	3.16E-2(3.97E-2)	2.49E-2(4.29E-2)

for each problem the best mean performance is shown with **bold face**

Table A.2: Mean (and standard deviation) for each low-level heuristic and the hyper-heuristic based on roulette for the R2 indicator (Part II)

Obj.	problem	IDEALNWSUM	IDEALSIGMA	MGACD	MGANWSUM	MGASIGMA
2	DTLZ1	4.05E-3(1.4E-2)	3.62E-3(9.27E-3)	1.84E-3(3.48E-3)	4.07E-3(1.4E-2)	3.13E-3(9.34E-3)
	DTLZ2	1.78E-1(4.03E-2)	3.18E-1(3.95E-2)	1.5E-1(7.51E-5)	1.53E-1(3.13E-3)	1.53E-1(2.01E-3)
	DTLZ3	4.52E-3(1.56E-2)	3.3E-3(9.32E-3)	2.46E-3(1.57E-3)	4.54E-3(1.56E-2)	3.28E-3(9.33E-3)
	DTLZ4	1.41E-1(6.6E-3)	2.36E-1(1.14E-1)	1.34E-1(4.57E-4)	1.4E-1(6.11E-3)	1.41E-1(6.02E-3)
	DTLZ5	1.86E-1(5.56E-2)	3.14E-1(6.04E-2)	1.47E-1(8.37E-5)	1.49E-1(3.35E-3)	1.49E-1(2.21E-3)
	DTLZ6	4.06E-2(3.23E-2)	4.69E-2(4.95E-2)	3.04E-2(2.28E-2)	3.87E-2(3.27E-2)	3.35E-2(2.81E-2)
	DTLZ7	7.78E-2(6.05E-2)	8.49E-2(6.48E-2)	6.8E-2(5.73E-2)	7.18E-2(6.11E-2)	8.16E-2(5.98E-2)
3	DTLZ1	3.8E-3(1.57E-2)	3.03E-3(5.87E-3)	9.72E-4(2.33E-3)	3.78E-3(1.57E-2)	1.84E-3(6.14E-3)
	DTLZ2	4.37E-2(1.98E-2)	9.77E-2(6.84E-3)	**2.16E-2(8.4E-5)**	2.31E-2(1.02E-3)	2.34E-2(1.65E-3)
	DTLZ3	1.65E-3(6.21E-3)	5.47E-4(1.6E-3)	3.33E-4(8.42E-5)	1.4E-3(6.15E-3)	5.24E-4(1.6E-3)
	DTLZ4	3.67E-2(3.25E-2)	8.94E-2(3.9E-2)	2.92E-2(5.71E-4)	3.11E-2(5.73E-3)	3.05E-2(5.85E-3)
	DTLZ5	1.54E-1(5.51E-2)	2.82E-1(3.51E-2)	1.25E-1(6.08E-5)	1.27E-1(1.55E-3)	1.27E-1(1.01E-3)
	DTLZ6	4.44E-2(4.88E-2)	3.74E-2(4.61E-2)	**2E-2(1.01E-2)**	3.36E-2(1.1E-2)	2.75E-2(1.22E-2)
	DTLZ7	1.01E-1(6.45E-2)	7E-2(3.93E-2)	5.18E-2(3.62E-2)	8.99E-2(3.58E-2)	6.53E-2(3.66E-2)
5	DTLZ1	1.48E-4(7.07E-4)	1.01E-3(1.13E-3)	1.25E-5(3.82E-5)	1.44E-4(6.99E-4)	6.53E-5(2.61E-4)
	DTLZ2	4.19E-2(1.54E-2)	5.59E-2(4.21E-3)	9.94E-4(8.64E-4)	3.09E-3(2.16E-3)	5.92E-3(2.93E-3)
	DTLZ3	1.99E-5(5.8E-5)	3.49E-5(5.22E-5)	3.58E-6(7.31E-6)	1.86E-5(6.19E-5)	9.99E-6(2.04E-5)
	DTLZ4	3.06E-2(2.17E-2)	4.27E-2(1.93E-2)	1.06E-3(6.51E-5)	1.53E-3(1.99E-3)	1.2E-3(8.62E-4)
	DTLZ5	5.61E-2(5.61E-3)	5.48E-2(6.1E-3)	2.12E-2(8.41E-4)	2.44E-2(9.86E-4)	2.45E-2(7.21E-4)
	DTLZ6	3.25E-2(1.69E-2)	4.07E-2(2.99E-2)	8.31E-3(3.65E-3)	1.26E-2(4.11E-3)	1.22E-2(4.15E-3)
	DTLZ7	7.88E-2(1.68E-2)	2.98E-2(2.36E-2)	4.65E-2(1.59E-2)	7.87E-2(1.37E-2)	2.85E-2(2.02E-2)
10	DTLZ1	1.81E-5(1.03E-5)	1.26E-4(6.22E-5)	**1.11E-5(3.38E-6)**	1.71E-5(4.32E-6)	1.5E-5(3.17E-6)
	DTLZ2	2.28E-2(7.9E-3)	2.76E-2(4.74E-3)	**2.22E-3(2.41E-4)**	3.29E-3(5.16E-4)	3.22E-3(7.83E-4)
	DTLZ3	**2.48E-5(6.06E-5)**	2.84E-4(4.97E-4)	3.32E-4(3.57E-4)	4.46E-5(6.82E-5)	3.28E-4(1.97E-4)
	DTLZ4	2.77E-2(4.74E-3)	2.4E-2(9.4E-3)	3.22E-3(2.38E-4)	4.01E-3(1.07E-3)	4.9E-3(2.41E-3)
	DTLZ5	2.8E-2(3.17E-3)	2.78E-2(3.52E-3)	**6.69E-3(6.37E-4)**	8.18E-3(6.94E-4)	8.9E-3(6.09E-4)
	DTLZ6	2.08E-2(8.09E-3)	1.95E-2(6.61E-3)	3.25E-3(1.77E-3)	4.2E-3(1.68E-3)	4.75E-3(1.48E-3)
	DTLZ7	3.95E-2(2.5E-2)	**2E-2(3.88E-2)**	3.77E-2(2.28E-2)	4.36E-2(2.23E-2)	6.71E-2(1.88E-2)
15	DTLZ1	2.53E-5(8.84E-5)	2.85E-4(1.88E-4)	5.98E-6(3.96E-6)	1.01E-5(7.36E-6)	1.03E-5(3.41E-6)
	DTLZ2	1.57E-2(5.43E-3)	1.72E-2(4.24E-3)	7.78E-4(1.28E-4)	1.11E-3(2.41E-4)	9.58E-4(1.65E-4)
	DTLZ3	**1.04E-5(1.28E-5)**	1.76E-4(3.33E-4)	1.58E-4(1.24E-4)	5.39E-5(8E-5)	1.68E-4(9.4E-5)
	DTLZ4	1.81E-2(3.62E-3)	1.45E-2(7.65E-3)	2.63E-3(2.86E-3)	2.4E-3(1.74E-3)	4.83E-3(3.29E-3)
	DTLZ5	1.86E-2(2.26E-3)	1.8E-2(3.91E-3)	**2.14E-3(2.82E-4)**	2.65E-3(4.02E-4)	3.15E-3(4.19E-4)
	DTLZ6	1.47E-2(5.52E-3)	1.48E-2(3.94E-3)	1.12E-3(7.76E-4)	1.3E-3(7.45E-4)	1.57E-3(6.97E-4)
	DTLZ7	3.36E-2(3.86E-2)	2.35E-2(4.77E-2)	4.23E-2(3.75E-2)	4.26E-2(3.65E-2)	6.07E-2(3.15E-2)
20	DTLZ1	1.65E-5(5.41E-5)	2.07E-4(1.49E-4)	3.86E-6(2.42E-6)	5.5E-6(5.07E-6)	5.91E-6(3.34E-6)
	DTLZ2	1.22E-2(2.98E-3)	1.41E-2(2.83E-3)	**4.45E-4(9.38E-5)**	6.4E-4(1.24E-4)	7.04E-4(2.56E-4)
	DTLZ3	**8.97E-6(9.43E-6)**	2.36E-4(4.54E-4)	1.48E-4(7.44E-5)	4.52E-5(4.62E-5)	1.23E-4(4.78E-5)
	DTLZ4	1.38E-2(2.52E-3)	1.25E-2(4.52E-3)	1.72E-3(1.9E-3)	1.08E-3(7.61E-4)	2.05E-3(1.45E-3)
	DTLZ5	1.4E-2(1.73E-3)	1.39E-2(2.22E-3)	1.27E-3(1.85E-4)	1.61E-3(2.31E-4)	1.91E-3(2.8E-4)
	DTLZ6	1.04E-2(5.03E-3)	1.08E-2(2.61E-3)	6.55E-4(5.1E-4)	7.75E-4(4.79E-4)	9.24E-4(4.8E-4)
	DTLZ7	3.39E-2(4.14E-2)	2.52E-2(4.74E-2)	5.64E-2(3.57E-2)	5.69E-2(3.52E-2)	5.94E-2(3.44E-2)

for each problem the best mean performance is shown with **bold face**

APPENDIX B

DETAILED RESULTS FOR DIFFERENT PARAMETER CONFIGURATIONS OF ACF

In Tables B.1 and B.2 are shown the detailed results for different parameter configurations evaluated for the Adaptive Choice Function selection method. In general, most configuration had similar results, and no one had the best value with a statistical difference in any test instance.

Table B.1: Mean (and standard deviation) for each configuration instance for ACF for the IGD indicator

Obj.	problem	ACFFIXED001	ACFFIXED005	ACFFIXED01	ACFFIXED05	ACFFIXED1
3	DTLZ1	7.28E-3(1.48E-3)	7.01E-3(1.28E-3)	6.8E-3(9.85E-4)	6.89E-3(1.16E-3)	**6.53E-3(6.01E-4)**
	DTLZ2	9.96E-3(1E-3)	9.83E-3(7.48E-4)	1.01E-2(1.27E-3)	9.81E-3(1.31E-3)	9.54E-3(6.88E-4)
	DTLZ3	1.64E-2(1.26E-2)	1.52E-2(1.13E-2)	1.74E-2(1.55E-2)	2.1E-2(1.72E-2)	1.97E-2(1.74E-2)
	DTLZ4	1.4E-2(6.68E-3)	1.43E-2(6.74E-3)	1.58E-2(8.56E-3)	1.2E-2(5.48E-3)	1.32E-2(6.39E-3)
	WFG6	1.16E-2(1.64E-3)	1.04E-2(1.54E-3)	1.07E-2(1.59E-3)	1.08E-2(1.68E-3)	1.04E-2(1.23E-3)
	WFG7	1.36E-2(2.14E-3)	1.41E-2(2.34E-3)	1.41E-2(2.43E-3)	1.39E-2(2.44E-3)	1.41E-2(2.38E-3)
5	DTLZ1	1.19E-2(8.23E-4)	1.15E-2(9.3E-4)	1.14E-2(9.3E-4)	1.15E-2(8.01E-4)	1.15E-2(6.89E-4)
	DTLZ2	1.88E-2(1.06E-3)	1.9E-2(9.54E-4)	1.9E-2(1.52E-3)	1.86E-2(9.74E-4)	1.9E-2(1.53E-3)
	DTLZ3	2.78E-2(3.6E-3)	2.8E-2(2.21E-3)	2.8E-2(2.61E-3)	2.84E-2(2.8E-3)	2.85E-2(3.73E-3)
	DTLZ4	2.22E-2(1.94E-3)	2.18E-2(1.98E-3)	**2.13E-2(2.07E-3)**	2.18E-2(1.96E-3)	2.18E-2(2.07E-3)
	WFG6	1.51E-2(5.96E-4)	1.53E-2(6.58E-4)	1.5E-2(5.33E-4)	1.55E-2(3.96E-4)	1.52E-2(7.8E-4)
	WFG7	1.92E-2(1.54E-3)	1.88E-2(1.31E-3)	1.96E-2(1.28E-3)	1.91E-2(1.64E-3)	1.9E-2(1.57E-3)
8	DTLZ1	3.08E-2(2.39E-3)	2.99E-2(2.8E-3)	3.09E-2(2.12E-3)	2.99E-2(2.78E-3)	3.07E-2(2.35E-3)
	DTLZ2	5.19E-2(3.71E-3)	**5.07E-2(2.55E-3)**	5.16E-2(2.54E-3)	5.22E-2(3.58E-3)	5.1E-2(2.79E-3)
	DTLZ3	6.25E-2(3.41E-3)	6.14E-2(3.13E-3)	6.24E-2(3.25E-3)	6.31E-2(3.75E-3)	**6.11E-2(2.64E-3)**
	DTLZ4	4.11E-2(2.36E-3)	4.06E-2(1.89E-3)	4.09E-2(1.89E-3)	4.14E-2(2.48E-3)	4.15E-2(2.47E-3)
	WFG6	3.89E-2(3.73E-3)	3.87E-2(3.08E-3)	3.97E-2(2.79E-3)	**3.87E-2(3.07E-3)**	3.95E-2(3.08E-3)
	WFG7	**4.33E-2(4.65E-3)**	4.38E-2(5.55E-3)	4.35E-2(3.43E-3)	4.44E-2(5.54E-3)	4.44E-2(4.62E-3)
10	DTLZ1	2.46E-2(1.34E-3)	2.44E-2(1.75E-3)	2.47E-2(1.56E-3)	2.39E-2(1.76E-3)	2.42E-2(2.14E-3)
	DTLZ2	**4.37E-2(2.46E-3)**	4.52E-2(2.66E-3)	4.42E-2(2.63E-3)	4.49E-2(1.91E-3)	4.52E-2(2.63E-3)
	DTLZ3	5.09E-2(2.21E-3)	5.05E-2(2.51E-3)	5.14E-2(2.46E-3)	5.08E-2(2.44E-3)	5.16E-2(1.65E-3)
	DTLZ4	3.23E-2(1.42E-3)	3.24E-2(1.3E-3)	3.21E-2(1.18E-3)	3.21E-2(1.28E-3)	**3.2E-2(1.24E-3)**
	WFG6	3.44E-2(2.38E-3)	3.67E-2(3.75E-3)	3.53E-2(2.66E-3)	3.62E-2(2.27E-3)	3.62E-2(2.07E-3)
	WFG7	5.28E-2(3.9E-3)	5.23E-2(4.72E-3)	5.39E-2(4.5E-3)	5.15E-2(4.32E-3)	5.33E-2(4.57E-3)
15	DTLZ1	5.59E-2(2.56E-3)	5.59E-2(1.8E-3)	5.58E-2(1.86E-3)	5.52E-2(2.12E-3)	5.46E-2(1.63E-3)
	DTLZ2	9.08E-2(2.84E-3)	9.05E-2(2.75E-3)	9.22E-2(2.52E-3)	9.03E-2(2.42E-3)	9.19E-2(2.21E-3)
	DTLZ3	9.57E-2(1.65E-3)	9.6E-2(2.32E-3)	9.63E-2(2.16E-3)	**9.52E-2(2.56E-3)**	9.72E-2(1.8E-3)
	DTLZ4	5.28E-2(9.17E-4)	5.28E-2(1.31E-3)	5.28E-2(1.12E-3)	5.35E-2(1.57E-3)	5.31E-2(1.13E-3)
	WFG6	1.05E-1(1.29E-2)	1.04E-1(1.44E-2)	1.1E-1(1.52E-2)	1.06E-1(1.41E-2)	1.14E-1(1.82E-2)
	WFG7	**1.56E-1(1.44E-2)**	1.63E-1(1.51E-2)	1.63E-1(1.25E-2)	1.63E-1(1.43E-2)	1.6E-1(1.43E-2)

Obj.	problem	ACFFIXED5	ACFFIXEDM01	ACFFIXEDM05	ACFFIXEDN	ACFFIXEDMN
3	DTLZ1	7.07E-3(1.13E-3)	7.83E-3(4.28E-3)	8.03E-3(6E-3)	6.65E-3(1.16E-3)	8.9E-3(6.71E-3)
	DTLZ2	9.45E-3(6.71E-4)	9.81E-3(7.36E-4)	**9.25E-3(5.6E-4)**	9.52E-3(9.75E-4)	9.43E-3(8.98E-4)
	DTLZ3	1.16E-2(9.99E-3)	**1.1E-2(8.1E-3)**	1.22E-2(1.01E-2)	1.46E-2(1.16E-2)	1.93E-2(1.49E-2)
	DTLZ4	1.45E-2(5.56E-3)	1.19E-2(6.24E-3)	1.31E-2(4.63E-3)	**1.19E-2(4.5E-3)**	2.3E-2(7.11E-3)
	WFG6	1.02E-2(1.52E-3)	1E-2(1.07E-3)	1.07E-2(1.92E-3)	**9.96E-3(1.04E-3)**	1.01E-2(8.4E-4)
	WFG7	1.35E-2(1.43E-3)	1.35E-2(1.9E-3)	**1.27E-2(1.08E-3)**	1.41E-2(2.46E-3)	1.3E-2(1.21E-3)
5	DTLZ1	1.17E-2(8.2E-4)	1.16E-2(6.41E-4)	1.1E-2(8.32E-4)	1.19E-2(7.33E-4)	**1.09E-2(9.12E-4)**
	DTLZ2	1.84E-2(7.36E-4)	1.85E-2(1.28E-3)	1.73E-2(8.28E-4)	1.88E-2(1.35E-3)	**1.7E-2(8.38E-4)**
	DTLZ3	2.82E-2(4.05E-3)	2.88E-2(3.28E-3)	2.79E-2(3.16E-3)	**2.72E-2(2.04E-3)**	2.98E-2(2.6E-3)
	DTLZ4	2.28E-2(1.55E-3)	2.19E-2(1.94E-3)	2.21E-2(2.16E-3)	2.28E-2(2.49E-3)	2.37E-2(3.64E-3)
	WFG6	**1.49E-2(7.26E-4)**	1.51E-2(5.3E-4)	1.5E-2(5.67E-4)	1.54E-2(7.15E-4)	1.54E-2(7.15E-4)
	WFG7	1.86E-2(1.08E-3)	**1.75E-2(8.91E-4)**	1.81E-2(1.17E-3)	1.91E-2(1.56E-3)	1.85E-2(2.59E-3)
8	DTLZ1	2.97E-2(2.42E-3)	2.87E-2(2.8E-3)	2.85E-2(3.03E-3)	2.94E-2(2.22E-3)	**2.76E-2(3.11E-3)**
	DTLZ2	5.11E-2(2.74E-3)	5.08E-2(3.38E-3)	5.16E-2(2.47E-3)	5.1E-2(2.75E-3)	5.23E-2(4.64E-3)
	DTLZ3	6.3E-2(3.44E-3)	6.13E-2(3.75E-3)	6.4E-2(5E-3)	6.31E-2(3.21E-3)	1.29E-1(2.51E-1)
	DTLZ4	**4.04E-2(1.81E-3)**	4.21E-2(2.38E-3)	4.16E-2(1.69E-3)	4.12E-2(1.55E-3)	4.29E-2(3.97E-3)
	WFG6	4E-2(3.04E-3)	3.97E-2(3.76E-3)	3.91E-2(2.63E-3)	4.03E-2(3.16E-3)	4.09E-2(4.57E-3)
	WFG7	4.4E-2(3.21E-3)	4.36E-2(4.62E-3)	4.4E-2(4.18E-3)	4.41E-2(3.86E-3)	4.77E-2(4.38E-3)
10	DTLZ1	**2.27E-2(1.51E-3)**	2.31E-2(1.19E-3)	2.34E-2(1.6E-3)	2.41E-2(1.47E-3)	2.27E-2(1.98E-3)
	DTLZ2	4.44E-2(2.8E-3)	4.51E-2(2.8E-3)	4.52E-2(3.14E-3)	4.65E-2(2.48E-3)	4.56E-2(2.5E-3)
	DTLZ3	5.08E-2(3.49E-3)	**5.03E-2(2.42E-3)**	5.06E-2(2.69E-3)	5.11E-2(2.47E-3)	5.6E-2(6.06E-3)
	DTLZ4	3.24E-2(1.29E-3)	3.2E-2(1.27E-3)	3.22E-2(9.38E-4)	3.2E-2(1.11E-3)	3.45E-2(4.82E-3)
	WFG6	3.51E-2(2.14E-3)	**3.44E-2(2.18E-3)**	3.69E-2(2.99E-3)	3.49E-2(2.24E-3)	3.63E-2(3.1E-3)
	WFG7	5.24E-2(4.89E-3)	5.7E-2(3.4E-3)	5.61E-2(3.94E-3)	**5.14E-2(4.4E-3)**	6.62E-2(3.71E-3)
15	DTLZ1	5.58E-2(1.91E-3)	**5.45E-2(1.76E-3)**	5.63E-2(3.45E-3)	5.61E-2(2.12E-3)	5.62E-2(3.72E-3)
	DTLZ2	9.13E-2(2.38E-3)	9.21E-2(1.32E-3)	9.15E-2(2.3E-3)	**9.02E-2(2.81E-3)**	9.16E-2(1.93E-3)
	DTLZ3	9.63E-2(2.62E-3)	1.32E-1(1.54E-1)	9.9E-2(2.68E-3)	9.61E-2(1.5E-3)	4.18E-1(1.23E0)
	DTLZ4	**5.26E-2(9.8E-4)**	5.38E-2(2.4E-3)	5.34E-2(2.61E-3)	5.37E-2(2.01E-3)	6.72E-2(1.84E-2)
	WFG6	1.03E-1(1.35E-2)	1.04E-1(1.7E-2)	1.05E-1(1.41E-2)	1.1E-1(1.24E-2)	**1.01E-1(1.46E-2)**
	WFG7	1.62E-1(1.47E-2)	1.74E-1(1.29E-2)	1.79E-1(1.05E-2)	1.67E-1(1.48E-2)	1.87E-1(1.39E-2)

Table B.2: Mean (and standard deviation) for each configuration instance for ACF for the HV indicator

Obj.	problem	ACFFIXED001	ACFFIXED005	ACFFIXED01	ACFFIXED05	ACFFIXED1
3	DTLZ1	9.99E-1(7.96E-4)	1E0(7.18E-4)	1E0(5.4E-4)	1E0(4.75E-4)	**1E0(2.6E-4)**
	DTLZ2	8.81E-1(4.32E-3)	8.78E-1(7.65E-3)	8.76E-1(1.31E-2)	8.78E-1(9.04E-3)	8.82E-1(2.02E-3)
	DTLZ3	1E0(8.5E-5)	1E0(3.45E-5)	1E0(9.08E-5)	1E0(2.94E-4)	1E0(5.64E-4)
	DTLZ4	7.86E-1(7.86E-3)	7.86E-1(7.69E-3)	7.83E-1(1.05E-2)	7.87E-1(6.92E-3)	7.87E-1(7.18E-3)
	WFG6	3.74E-1(8.51E-3)	3.82E-1(1.3E-2)	3.79E-1(8.03E-3)	3.81E-1(9.61E-3)	3.79E-1(7.67E-3)
	WFG7	4.18E-1(1.81E-2)	4.19E-1(1.93E-2)	4.15E-1(2E-2)	4.22E-1(1.68E-2)	4.17E-1(1.69E-2)
5	DTLZ1	9.85E-1(3.14E-3)	9.85E-1(4.12E-3)	9.85E-1(4.05E-3)	9.85E-1(2.91E-3)	9.85E-1(3.76E-3)
	DTLZ2	9.92E-1(1.34E-3)	9.91E-1(3.74E-3)	9.91E-1(2.35E-3)	9.92E-1(1.44E-3)	9.92E-1(2E-3)
	DTLZ3	1E0(8.2E-9)	1E0(6.93E-9)	1E0(1.41E-8)	1E0(5.97E-9)	1E0(1.31E-8)
	DTLZ4	9.99E-1(4.55E-5)	9.99E-1(4.85E-5)	**9.99E-1(5.78E-5)**	9.99E-1(5.4E-5)	9.99E-1(4.52E-5)
	WFG6	5.46E-1(2.34E-2)	5.59E-1(1.88E-2)	5.62E-1(2.08E-2)	5.52E-1(2.72E-2)	5.59E-1(2.22E-2)
	WFG7	5.81E-1(1.69E-2)	5.89E-1(2.19E-2)	5.79E-1(2.33E-2)	5.86E-1(2.12E-2)	5.81E-1(2.35E-2)
8	DTLZ1	1E0(5.05E-8)	1E0(1.19E-8)	1E0(7.34E-10)	1E0(1.61E-9)	1E0(2.03E-8)
	DTLZ2	9.96E-1(7.92E-4)	9.96E-1(1.56E-3)	9.94E-1(4.93E-3)	9.95E-1(2.87E-3)	9.96E-1(1.5E-3)
	DTLZ3	1E0(1.66E-8)	1E0(1.94E-9)	1E0(5.62E-9)	1E0(3.73E-8)	1E0(1.89E-9)
	DTLZ4	1E0(4.79E-6)	1E0(6.25E-6)	**1E0(2.09E-6)**	1E0(4.44E-6)	1E0(3.02E-5)
	WFG6	6.71E-1(4.04E-2)	6.76E-1(3.77E-2)	6.64E-1(2.98E-2)	6.67E-1(3.63E-2)	6.64E-1(3.76E-2)
	WFG7	4.55E-1(1.97E-2)	4.53E-1(3.44E-2)	4.63E-1(3.18E-2)	4.6E-1(3.14E-2)	4.6E-1(3.24E-2)
10	DTLZ1	1E0(1.54E-9)	1E0(3.29E-9)	1E0(1.68E-9)	1E0(1.55E-9)	1E0(5.24E-10)
	DTLZ2	9.98E-1(1.21E-3)	9.97E-1(1.37E-3)	9.97E-1(1.75E-3)	9.96E-1(1.91E-3)	9.97E-1(2.2E-3)
	DTLZ3	1E0(1.63E-9)	1E0(3.91E-10)	1E0(3.53E-9)	1E0(4.12E-10)	1E0(4.05E-10)
	DTLZ4	1E0(1.31E-7)	**1E0(1.8E-7)**	1E0(6.15E-7)	1E0(2.95E-7)	1E0(2.43E-7)
	WFG6	7.67E-1(3.89E-2)	7.57E-1(4.57E-2)	7.69E-1(3.24E-2)	7.63E-1(2.83E-2)	7.68E-1(3.25E-2)
	WFG7	4.91E-1(2.61E-2)	4.84E-1(3.07E-2)	4.87E-1(2.13E-2)	4.95E-1(2.08E-2)	4.83E-1(2.16E-2)

Obj.	problem	ACFFIXED5	ACFFIXEDM01	ACFFIXEDM05	ACFFIXEDN	ACFFIXEDMN
3	DTLZ1	1E0(5.96E-4)	1E0(6.24E-4)	9.99E-1(1.64E-3)	1E0(5.77E-4)	1E0(1.56E-3)
	DTLZ2	8.83E-1(1.36E-3)	8.82E-1(2.35E-3)	8.83E-1(1.74E-3)	8.81E-1(3.27E-3)	**8.86E-1(1.38E-3)**
	DTLZ3	1E0(2.8E-4)	**1E0(1.42E-5)**	1E0(1.64E-4)	1E0(1.42E-5)	1E0(2.28E-5)
	DTLZ4	7.85E-1(6.8E-3)	7.88E-1(6.86E-3)	7.87E-1(5.65E-3)	**7.88E-1(4.7E-3)**	7.71E-1(1.48E-2)
	WFG6	3.8E-1(1.06E-2)	3.85E-1(1.2E-2)	3.81E-1(1.06E-2)	3.8E-1(7.15E-3)	**3.89E-1(9.51E-3)**
	WFG7	4.14E-1(1.14E-2)	4.19E-1(1.72E-2)	4.26E-1(1.94E-2)	4.19E-1(1.91E-2)	**4.27E-1(1.7E-2)**
5	DTLZ1	9.84E-1(3.74E-3)	9.84E-1(3.79E-3)	9.84E-1(5.21E-3)	9.84E-1(3.68E-3)	**9.87E-1(5.28E-3)**
	DTLZ2	9.92E-1(9.53E-4)	9.92E-1(2.38E-3)	9.94E-1(5.99E-4)	9.91E-1(2.34E-3)	**9.94E-1(8.17E-4)**
	DTLZ3	1E0(8.57E-9)	1E0(8.59E-9)	1E0(2.81E-8)	**1E0(2.47E-9)**	1E0(5.18E-9)
	DTLZ4	9.99E-1(4.93E-5)	9.99E-1(3.87E-5)	9.99E-1(5.53E-5)	9.99E-1(5.56E-5)	9.99E-1(7.45E-5)
	WFG6	5.54E-1(1.89E-2)	5.65E-1(1.95E-2)	**5.65E-1(1.46E-2)**	5.54E-1(2.54E-2)	5.6E-1(1.91E-2)
	WFG7	5.92E-1(2.01E-2)	**6.21E-1(1.49E-2)**	6.06E-1(2.35E-2)	5.84E-1(2.73E-2)	6.03E-1(4.8E-2)
8	DTLZ1	1E0(1.54E-9)	1E0(6.71E-10)	1E0(2.15E-10)	1E0(2.05E-10)	**1E0(0E0)**
	DTLZ2	9.96E-1(2.7E-3)	9.96E-1(1.8E-3)	9.96E-1(3.2E-3)	9.95E-1(2.86E-3)	**9.97E-1(2.26E-3)**
	DTLZ3	1E0(3.95E-9)	**1E0(1.94E-9)**	1E0(6.27E-8)	1E0(2.02E-8)	1E0(9.58E-7)
	DTLZ4	1E0(5.87E-6)	1E0(1.2E-5)	1E0(4.31E-6)	1E0(3.58E-6)	1E0(7.28E-5)
	WFG6	**6.77E-1(3.2E-2)**	6.68E-1(3.78E-2)	6.62E-1(2.93E-2)	6.75E-1(3.8E-2)	6.45E-1(4.2E-2)
	WFG7	4.7E-1(2.27E-2)	4.62E-1(3.79E-2)	**4.76E-1(2.43E-2)**	4.53E-1(2.01E-2)	4.66E-1(4.35E-2)
10	DTLZ1	1E0(9.04E-10)	1E0(5.8E-9)	1E0(4.07E-9)	1E0(5.07E-9)	**1E0(0E0)**
	DTLZ2	9.97E-1(1.45E-3)	9.97E-1(2.04E-3)	9.97E-1(1.51E-3)	9.97E-1(2.04E-3)	**9.98E-1(1.16E-3)**
	DTLZ3	1E0(7.11E-9)	1E0(3.23E-10)	**1E0(1.98E-10)**	1E0(5.04E-10)	1E0(6.74E-9)
	DTLZ4	1E0(1.99E-7)	1E0(2.37E-7)	1E0(1.59E-7)	1E0(1.83E-7)	1E0(7.09E-5)
	WFG6	**7.76E-1(3.35E-2)**	7.57E-1(3.15E-2)	7.57E-1(3.01E-2)	7.69E-1(3.47E-2)	7.51E-1(4.86E-2)
	WFG7	4.88E-1(2.02E-2)	4.94E-1(2.2E-2)	4.91E-1(2.25E-2)	**4.98E-1(2.78E-2)**	4.79E-1(4.11E-2)

APPENDIX C

DETAILED RESULTS FOR DIFFERENT PARAMETER CONFIGURATIONS OF UCB1

In Tables C.1 and C.2 are shown the detailed results for different parameter configurations evaluated for the UCB1 selection method. In general, most configuration had similar results, and no one had the best value with a statistical difference in any test instance.

Table C.1: Mean (and standard deviation) for each configuration instance for UCB1 for the IGD indicator

Obj.	problem	UCBC10W10	UCBC10W20	UCBC10W50	UCBC5W10	UCBC5W20	UCBC5W50
3	DTLZ1	7.33E-3(1.42E-3)	8.72E-3(6.02E-3)	8.05E-3(2E-3)	**7.25E-3(1.12E-3)**	7.6E-3(1.3E-3)	7.37E-3(1.17E-3)
	DTLZ2	1.03E-2(9.51E-4)	1.03E-2(1.09E-3)	1.01E-2(9E-4)	1.04E-2(1.45E-3)	**9.91E-3(8.2E-4)**	1.07E-2(1.98E-3)
	DTLZ3	1.84E-2(1.36E-2)	1.51E-2(1.22E-2)	1.53E-2(9.03E-3)	1.95E-2(1.73E-2)	**1.31E-2(8.83E-3)**	1.64E-2(1.55E-2)
	DTLZ4	1.56E-2(5.14E-3)	**1.55E-2(6.34E-3)**	1.8E-2(8.27E-3)	1.69E-2(8.87E-3)	2.03E-2(8.19E-3)	1.63E-2(8.13E-3)
	WFG6	1.19E-2(1.96E-3)	**1.06E-2(1.44E-3)**	1.16E-2(2.08E-3)	1.2E-2(1.94E-3)	1.1E-2(1.46E-3)	1.1E-2(1.55E-3)
	WFG7	1.58E-2(2.91E-3)	**1.56E-2(2.69E-3)**	1.67E-2(3.09E-3)	1.62E-2(3.3E-3)	1.81E-2(2.53E-3)	1.65E-2(3.1E-3)
5	DTLZ1	1.17E-2(7.95E-4)	1.16E-2(6.8E-4)	**1.16E-2(7.69E-4)**	1.19E-2(6.07E-4)	1.17E-2(9.3E-4)	1.17E-2(7.82E-4)
	DTLZ2	1.94E-2(1.38E-3)	1.95E-2(1.22E-3)	1.94E-2(1.08E-3)	1.93E-2(1.13E-3)	**1.93E-2(1.17E-3)**	1.97E-2(1E-3)
	DTLZ3	2.82E-2(2.54E-3)	2.87E-2(2.59E-3)	2.8E-2(2.2E-3)	2.86E-2(2.47E-3)	2.88E-2(2.5E-3)	2.8E-2(2.14E-3)
	DTLZ4	**2.16E-2(2.67E-3)**	2.27E-2(2.84E-3)	2.34E-2(1.76E-3)	2.27E-2(1.78E-3)	2.28E-2(1.49E-3)	2.5E-2(1.98E-3)
	WFG6	1.53E-2(4.57E-4)	1.54E-2(7.28E-4)	1.53E-2(5.25E-4)	**1.52E-2(5.43E-4)**	1.53E-2(6.36E-4)	1.57E-2(9.09E-4)
	WFG7	1.95E-2(1.64E-3)	2.06E-2(2.53E-3)	1.94E-2(1.72E-3)	1.93E-2(1.69E-3)	1.97E-2(1.62E-3)	**1.91E-2(1.07E-3)**
8	DTLZ1	**3.06E-2(2E-3)**	3.08E-2(2.82E-3)	3.17E-2(2.16E-3)	3.14E-2(2.37E-3)	3.09E-2(2.53E-3)	3.18E-2(2.18E-3)
	DTLZ2	5.1E-2(2.96E-3)	**5.05E-2(3.2E-3)**	5.3E-2(2.97E-3)	5.32E-2(2.24E-3)	5.19E-2(3.14E-3)	5.19E-2(2.76E-3)
	DTLZ3	**6.29E-2(3.93E-3)**	6.33E-2(3.73E-3)	6.81E-2(2.29E-2)	6.46E-2(5.09E-3)	6.39E-2(3.51E-3)	6.42E-2(3.6E-3)
	DTLZ4	4.25E-2(1.75E-3)	4.19E-2(1.74E-3)	4.25E-2(2.55E-3)	4.17E-2(1.67E-3)	4.17E-2(2.54E-3)	**4.15E-2(1.99E-3)**
	WFG6	3.89E-2(2.71E-3)	3.93E-2(3.67E-3)	3.76E-2(2.72E-3)	4.08E-2(2.84E-3)	3.9E-2(3.31E-3)	3.83E-2(3.03E-3)
	WFG7	4.46E-2(4.8E-3)	4.68E-2(4.94E-3)	4.48E-2(5.03E-3)	4.48E-2(3.8E-3)	**4.39E-2(4.31E-3)**	4.48E-2(3.38E-3)
10	DTLZ1	2.45E-2(2.26E-3)	**2.44E-2(1.24E-3)**	2.51E-2(1.62E-3)	2.52E-2(1.68E-3)	2.48E-2(1.86E-3)	2.52E-2(1.09E-3)
	DTLZ2	4.55E-2(3.11E-3)	4.45E-2(2.77E-3)	**4.42E-2(2.96E-3)**	4.42E-2(3.27E-3)	4.57E-2(2.43E-3)	4.43E-2(2.5E-3)
	DTLZ3	**5.07E-2(2.26E-3)**	5.08E-2(2.32E-3)	5.11E-2(2.27E-3)	5.19E-2(1.74E-3)	5.15E-2(1.95E-3)	5.12E-2(1.87E-3)
	DTLZ4	3.23E-2(1.28E-3)	**3.21E-2(1.14E-3)**	3.32E-2(1.52E-3)	3.29E-2(1.13E-3)	3.28E-2(1.74E-3)	3.32E-2(1.27E-3)
	WFG6	3.48E-2(2.72E-3)	3.52E-2(1.97E-3)	3.38E-2(1.67E-3)	3.49E-2(2.38E-3)	3.54E-2(1.75E-3)	3.5E-2(2.25E-3)
	WFG7	5.5E-2(4.46E-3)	5.3E-2(5.35E-3)	5.37E-2(4.36E-3)	**5.24E-2(3.84E-3)**	5.55E-2(4.56E-3)	5.33E-2(4.59E-3)
15	DTLZ1	5.55E-2(2.05E-3)	5.59E-2(2.46E-3)	5.57E-2(2.16E-3)	5.62E-2(2.42E-3)	5.62E-2(1.71E-3)	**5.55E-2(1.89E-3)**
	DTLZ2	9.17E-2(2.95E-3)	9.15E-2(2.26E-3)	9.2E-2(2.03E-3)	9.02E-2(3.2E-3)	9.16E-2(2.76E-3)	**8.98E-2(3.78E-3)**
	DTLZ3	9.56E-2(2.18E-3)	9.57E-2(2.22E-3)	9.61E-2(3.16E-3)	9.74E-2(6.27E-3)	9.6E-2(2.14E-3)	**9.5E-2(3.43E-3)**
	DTLZ4	**5.27E-2(1.09E-3)**	5.46E-2(3.18E-3)	5.62E-2(3.08E-3)	5.46E-2(1.56E-3)	5.44E-2(2.22E-3)	5.83E-2(6.17E-3)
	WFG6	1.08E-1(8.7E-3)	1.09E-1(1.48E-2)	1.09E-1(1E-2)	1.14E-1(1.62E-2)	1.06E-1(1.2E-2)	1.1E-1(1.14E-2)
	WFG7	1.65E-1(9.92E-3)	**1.58E-1(9.25E-3)**	1.66E-1(1.6E-2)	1.61E-1(1.31E-2)	1.65E-1(1.35E-2)	1.64E-1(1.15E-2)

Obj.	problem	UCBC1W10	UCBC1W20	UCBC1W50	UCBC0.5W10	UCBC0.5W20	UCBC0.5W50
3	DTLZ1	7.87E-3(1.32E-3)	8.38E-3(1.1E-3)	9.5E-3(5.72E-3)	8.26E-3(1.25E-3)	1.03E-2(5.67E-3)	9.47E-3(1.01E-3)
	DTLZ2	1.03E-2(6.53E-4)	1.13E-2(1.28E-3)	1.15E-2(7.61E-4)	1.14E-2(1.26E-3)	1.22E-2(1.64E-3)	1.21E-2(1.76E-3)
	DTLZ3	1.68E-2(1.19E-2)	1.69E-2(1.29E-2)	1.89E-2(1.2E-2)	1.85E-2(1.29E-2)	1.94E-2(1.27E-2)	1.61E-2(7.94E-3)
	DTLZ4	2.05E-2(9.46E-3)	1.76E-2(9.01E-3)	2.27E-2(8.32E-3)	2.26E-2(8.09E-3)	2.72E-2(8.8E-3)	2.18E-2(9.22E-3)
	WFG6	1.24E-2(1.96E-3)	1.25E-2(1.43E-3)	1.28E-2(1.25E-3)	1.26E-2(1.6E-3)	1.34E-2(1.02E-3)	1.38E-2(6.61E-4)
	WFG7	1.86E-2(2.94E-3)	2.08E-2(2.57E-3)	2.12E-2(2.41E-3)	1.86E-2(3.02E-3)	2.21E-2(2.06E-3)	2.09E-2(1.35E-3)
5	DTLZ1	1.17E-2(8.9E-4)	1.17E-2(9.98E-4)	1.31E-2(1.98E-3)	1.18E-2(1.12E-3)	1.25E-2(1.68E-3)	1.28E-2(1.24E-3)
	DTLZ2	2E-2(9.87E-4)	2.03E-2(1.76E-3)	2.19E-2(1.1E-3)	2.12E-2(1.54E-3)	2.32E-2(1.57E-3)	2.36E-2(1.4E-3)
	DTLZ3	**2.78E-2(3.77E-3)**	2.83E-2(2.95E-3)	2.83E-2(3.59E-3)	2.8E-2(3.23E-3)	2.84E-2(2.06E-3)	2.84E-2(2.68E-3)
	DTLZ4	2.34E-2(2.32E-3)	2.43E-2(2.1E-3)	2.4E-2(2.04E-3)	2.57E-2(2.39E-3)	2.53E-2(2.23E-3)	2.39E-2(2.8E-3)
	WFG6	1.52E-2(5.98E-4)	1.55E-2(6E-4)	1.55E-2(5.13E-4)	1.52E-2(8.13E-4)	1.57E-2(8.52E-4)	1.59E-2(1.12E-3)
	WFG7	2.01E-2(1.64E-3)	2.05E-2(1.27E-3)	2.1E-2(1.73E-3)	2.09E-2(1.3E-3)	2.17E-2(1.61E-3)	2.16E-2(1.37E-3)
8	DTLZ1	3.34E-2(3.27E-3)	3.23E-2(2.98E-3)	3.7E-2(6.01E-3)	3.46E-2(3.35E-3)	3.8E-2(4.9E-3)	3.86E-2(5.23E-3)
	DTLZ2	5.22E-2(3.39E-3)	5.31E-2(2.12E-3)	5.37E-2(3.88E-3)	5.28E-2(2.8E-3)	5.5E-2(3.2E-3)	5.54E-2(2.4E-3)
	DTLZ3	6.71E-2(5.23E-3)	6.67E-2(5.75E-3)	7.01E-2(6.62E-3)	1.46E-1(3.48E-1)	7.55E-2(8.25E-3)	1.44E-1(2.28E-1)
	DTLZ4	4.2E-2(2.12E-3)	4.45E-2(2.98E-3)	4.64E-2(3.96E-3)	4.39E-2(3.62E-3)	4.77E-2(3.59E-3)	4.96E-2(4.15E-3)
	WFG6	3.84E-2(3.28E-3)	3.82E-2(2.49E-3)	3.72E-2(2.2E-3)	**3.72E-2(2.27E-3)**	3.81E-2(2.89E-3)	3.78E-2(1.8E-3)
	WFG7	4.48E-2(4.62E-3)	4.61E-2(5.31E-3)	4.46E-2(4.07E-3)	4.46E-2(3.92E-3)	4.74E-2(5.22E-3)	4.57E-2(5.25E-3)
10	DTLZ1	2.58E-2(2.17E-3)	2.7E-2(1.74E-3)	2.76E-2(2.67E-3)	2.72E-2(2.32E-3)	2.97E-2(2.68E-3)	6.92E-2(1.64E-1)
	DTLZ2	4.49E-2(2.8E-3)	4.52E-2(2.1E-3)	4.62E-2(2.52E-3)	4.53E-2(2.45E-3)	4.53E-2(2.15E-3)	4.77E-2(1.79E-3)
	DTLZ3	5.18E-2(2.02E-3)	5.35E-2(2.69E-3)	5.52E-2(2.32E-3)	5.51E-2(3.04E-3)	9.64E-2(1.76E-1)	5.78E-2(4.54E-3)
	DTLZ4	3.32E-2(1.09E-3)	3.4E-2(1.96E-3)	3.68E-2(3.05E-3)	3.6E-2(3.04E-3)	4E-2(3.17E-3)	3.84E-2(3.42E-3)
	WFG6	3.48E-2(2.01E-3)	3.49E-2(2.5E-3)	3.56E-2(2.11E-3)	3.55E-2(2.1E-3)	**3.27E-2(1.63E-3)**	3.36E-2(2.01E-3)
	WFG7	5.52E-2(4.03E-3)	5.67E-2(6.14E-3)	5.76E-2(4.29E-3)	5.65E-2(5.97E-3)	5.48E-2(6.2E-3)	6.18E-2(6.24E-3)
15	DTLZ1	5.77E-2(2.79E-3)	5.81E-2(2.14E-3)	5.79E-2(3.43E-3)	5.88E-2(3.46E-3)	8.22E-2(1.01E-1)	7.5E-2(4.05E-2)
	DTLZ2	9.13E-2(2.33E-3)	9.28E-2(2.2E-3)	9.33E-2(2.41E-3)	9.15E-2(2.76E-3)	9.27E-2(2.8E-3)	9.2E-2(3.5E-3)
	DTLZ3	1.23E-1(8.47E-2)	1.04E-1(2.67E-2)	1.51E-1(1.59E-1)	2.93E-1(3.24E-1)	2.73E-1(5.41E-1)	1.64E-1(2.03E-1)
	DTLZ4	5.61E-2(3.04E-3)	5.93E-2(7.15E-3)	7.49E-2(1.16E-2)	6.07E-2(9.63E-3)	6.93E-2(1.25E-2)	7.73E-2(1.44E-2)
	WFG6	1.05E-1(1.16E-2)	1.07E-1(1.12E-2)	1.01E-1(1.23E-2)	1.06E-1(1.14E-2)	**9.69E-2(8.62E-3)**	9.84E-2(1.35E-2)
	WFG7	1.64E-1(1.59E-2)	1.62E-1(1.49E-2)	1.61E-1(1.91E-2)	1.64E-1(1.54E-2)	1.67E-1(1.6E-2)	1.67E-1(1.73E-2)

Table C.2: Mean (and standard deviation) for each configuration instance for UCB1 for the HV indicator

Obj.	problem	UCBC10W10	UCBC10W20	UCBC10W50	UCBC5W10	UCBC5W20	UCBC5W50
3	DTLZ1	1E0(2.55E-4)	1E0(3.61E-4)	1E0(2.24E-4)	**1E0(1.53E-4)**	1E0(2.3E-4)	1E0(2.95E-4)
	DTLZ2	8.2E-1(1.31E-2)	8.2E-1(1.91E-2)	8.2E-1(1.35E-2)	8.22E-1(6.4E-3)	**8.25E-1(3.53E-3)**	8.15E-1(2.94E-2)
	DTLZ3	1E0(8.89E-5)	1E0(2.46E-4)	1E0(2.15E-5)	1E0(3.61E-4)	**1E0(1.98E-5)**	1E0(5.76E-4)
	DTLZ4	**9.47E-1(1.29E-3)**	9.46E-1(1.94E-3)	9.45E-1(2.32E-3)	9.46E-1(2.73E-3)	9.45E-1(2.47E-3)	9.46E-1(2.34E-3)
	WFG6	3.7E-1(1.1E-2)	**3.78E-1(1.12E-2)**	3.75E-1(1.1E-2)	3.75E-1(1.34E-2)	3.73E-1(8.85E-3)	3.72E-1(9.09E-3)
	WFG7	**4.86E-1(2.04E-2)**	4.85E-1(1.62E-2)	4.82E-1(1.73E-2)	4.78E-1(2.53E-2)	4.77E-1(2E-2)	4.77E-1(1.97E-2)
5	DTLZ1	9.96E-1(1.86E-3)	9.96E-1(1.61E-3)	**9.96E-1(2.08E-3)**	9.96E-1(1.68E-3)	9.96E-1(1.75E-3)	9.96E-1(1.82E-3)
	DTLZ2	9.95E-1(1.11E-3)	9.95E-1(1.37E-3)	9.95E-1(1.24E-3)	**9.95E-1(7.49E-4)**	9.94E-1(1.62E-3)	9.93E-1(2.09E-3)
	DTLZ3	1E0(2.82E-9)	1E0(4.68E-9)	1E0(1.94E-9)	1E0(7.49E-9)	1E0(1.17E-8)	1E0(1.59E-9)
	DTLZ4	**9.99E-1(5.95E-5)**	9.99E-1(7.09E-5)	9.99E-1(5.62E-5)	9.99E-1(6.85E-5)	9.99E-1(4.29E-5)	9.99E-1(6.75E-5)
	WFG6	5.5E-1(2.54E-2)	5.56E-1(2.23E-2)	5.5E-1(3.02E-2)	5.55E-1(2.08E-2)	5.57E-1(2.41E-2)	5.46E-1(2.49E-2)
	WFG7	5.64E-1(2.32E-2)	5.51E-1(2.54E-2)	**5.69E-1(1.79E-2)**	5.67E-1(2.84E-2)	5.65E-1(1.97E-2)	5.66E-1(1.33E-2)
8	DTLZ1	1E0(1.23E-8)	1E0(2.06E-9)	1E0(5.18E-10)	1E0(4.92E-10)	1E0(2.1E-8)	**1E0(5.23E-11)**
	DTLZ2	9.96E-1(1.73E-3)	**9.96E-1(2.32E-3)**	9.95E-1(1.83E-3)	9.95E-1(1.91E-3)	9.93E-1(3.64E-3)	9.95E-1(1.8E-3)
	DTLZ3	1E0(7.53E-9)	1E0(7.97E-9)	1E0(1.02E-8)	1E0(1.29E-8)	1E0(5.83E-9)	**1E0(6.69E-9)**
	DTLZ4	1E0(6.72E-6)	1E0(3.68E-6)	1E0(1.28E-5)	1E0(4.36E-6)	**1E0(3.61E-6)**	1E0(3.45E-6)
	WFG6	6.62E-1(2.21E-2)	**6.83E-1(2.79E-2)**	6.79E-1(2.76E-2)	6.82E-1(4.09E-2)	6.71E-1(3.06E-2)	6.64E-1(2.46E-2)
	WFG7	**4.8E-1(2.48E-2)**	4.76E-1(1.71E-2)	4.71E-1(2.72E-2)	4.73E-1(2.39E-2)	4.8E-1(3.02E-2)	4.78E-1(2.78E-2)
10	DTLZ1	1E0(3.08E-11)	1E0(7.16E-11)	1E0(5.23E-11)	1E0(4.25E-10)	1E0(4.85E-9)	1E0(3.04E-9)
	DTLZ2	9.96E-1(2.56E-3)	9.96E-1(2.29E-3)	9.95E-1(2.8E-3)	9.96E-1(2.64E-3)	**9.97E-1(1.48E-3)**	9.96E-1(2.53E-3)
	DTLZ3	1E0(7.79E-10)	1E0(1.79E-9)	**1E0(3.01E-10)**	1E0(6.59E-10)	1E0(5.52E-10)	1E0(1.69E-9)
	DTLZ4	1E0(3.27E-7)	**1E0(1.31E-7)**	1E0(3.8E-7)	1E0(2.49E-7)	1E0(4.24E-7)	1E0(4.16E-7)
	WFG6	7.82E-1(2.83E-2)	7.82E-1(2.44E-2)	7.63E-1(2.17E-2)	7.76E-1(3.06E-2)	7.83E-1(2.8E-2)	7.88E-1(2.14E-2)
	WFG7	4.86E-1(2.24E-2)	4.87E-1(3.11E-2)	4.85E-1(2.27E-2)	4.87E-1(2.12E-2)	4.94E-1(2.32E-2)	4.85E-1(1.87E-2)

Obj.	problem	UCBC1W10	UCBC1W20	UCBC1W50	UCBC0.5W10	UCBC0.5W20	UCBC0.5W50
3	DTLZ1	1E0(2.15E-4)	1E0(2.38E-4)	1E0(3.27E-4)	1E0(1.55E-4)	1E0(3.96E-4)	1E0(2.72E-4)
	DTLZ2	8.16E-1(1.5E-2)	8.09E-1(3.15E-2)	8.01E-1(2.8E-2)	8.05E-1(2.43E-2)	7.89E-1(3.47E-2)	7.92E-1(3.35E-2)
	DTLZ3	1E0(2.27E-4)	1E0(9.92E-4)	1E0(2.6E-5)	1E0(4.18E-4)	1E0(9.92E-5)	1E0(2.68E-5)
	DTLZ4	9.44E-1(3.12E-3)	9.45E-1(2.86E-3)	9.44E-1(2.96E-3)	9.44E-1(2.82E-3)	9.41E-1(4.42E-3)	9.43E-1(3.48E-3)
	WFG6	3.67E-1(1.24E-2)	3.67E-1(9.29E-3)	3.61E-1(7.94E-3)	3.6E-1(9.31E-3)	3.57E-1(7.99E-3)	3.56E-1(7.15E-3)
	WFG7	4.6E-1(1.58E-2)	4.44E-1(1.14E-2)	4.35E-1(1.41E-2)	4.5E-1(1.54E-2)	4.24E-1(1.12E-2)	4.3E-1(1.17E-2)
5	DTLZ1	9.95E-1(2.31E-3)	9.94E-1(5.98E-3)	9.9E-1(6.01E-3)	9.93E-1(2.47E-3)	9.9E-1(7.35E-3)	9.91E-1(3.76E-3)
	DTLZ2	9.93E-1(1.58E-3)	9.94E-1(1.29E-3)	9.92E-1(1.95E-3)	9.91E-1(2.3E-3)	9.9E-1(1.66E-3)	9.9E-1(2.87E-3)
	DTLZ3	1E0(1.05E-8)	1E0(5.06E-9)	1E0(3.35E-8)	1E0(1.38E-8)	**1E0(1.38E-9)**	1E0(4.68E-9)
	DTLZ4	9.99E-1(6.94E-5)	9.99E-1(5.34E-5)	9.99E-1(8.47E-5)	9.99E-1(7.61E-5)	9.99E-1(6.42E-5)	9.99E-1(9.21E-5)
	WFG6	**5.57E-1(2.15E-2)**	5.55E-1(1.75E-2)	5.51E-1(1.97E-2)	5.55E-1(2.23E-2)	5.35E-1(3.8E-2)	5.28E-1(4.63E-2)
	WFG7	5.46E-1(2.06E-2)	5.32E-1(2.15E-2)	5.15E-1(2.62E-2)	5.19E-1(2.76E-2)	5.02E-1(2.45E-2)	5.08E-1(2.02E-2)
8	DTLZ1	1E0(6.01E-10)	1E0(8.26E-10)	1E0(7.25E-9)	1E0(5.5E-11)	1E0(5.78E-9)	1E0(1.43E-9)
	DTLZ2	9.92E-1(4.93E-3)	9.92E-1(4.5E-3)	9.89E-1(5.55E-3)	9.91E-1(4.09E-3)	9.87E-1(5.64E-3)	9.86E-1(5.2E-3)
	DTLZ3	1E0(5.63E-9)	1E0(2.97E-8)	1E0(5.15E-8)	1E0(4.46E-8)	1E0(2.31E-8)	1E0(4.09E-8)
	DTLZ4	1E0(1.4E-5)	1E0(2.32E-5)	1E0(9.34E-5)	1E0(3.76E-5)	1E0(7.36E-5)	1E0(8.76E-5)
	WFG6	6.74E-1(2.96E-2)	6.69E-1(3.72E-2)	6.51E-1(1.92E-2)	6.65E-1(3.51E-2)	6.58E-1(3.39E-2)	6.33E-1(1.45E-2)
	WFG7	4.77E-1(2.12E-2)	4.67E-1(2.82E-2)	4.57E-1(3.25E-2)	4.49E-1(3.06E-2)	4.48E-1(2.82E-2)	4.66E-1(4.28E-2)
10	DTLZ1	**1E0(2.24E-11)**	1E0(6.71E-11)	1E0(2.24E-11)	1E0(4.47E-11)	1E0(2.24E-11)	1E0(7.96E-8)
	DTLZ2	9.96E-1(2.88E-3)	9.95E-1(2.49E-3)	9.93E-1(3.67E-3)	9.94E-1(3.32E-3)	9.92E-1(3.01E-3)	9.91E-1(4.6E-3)
	DTLZ3	1E0(2.07E-9)	1E0(2.09E-9)	1E0(1.68E-9)	1E0(4.18E-9)	1E0(7.61E-9)	1E0(2.03E-8)
	DTLZ4	1E0(7.29E-7)	1E0(1.71E-6)	1E0(6.51E-6)	1E0(8.45E-6)	1E0(1.09E-5)	1E0(1.63E-5)
	WFG6	7.84E-1(2.73E-2)	7.88E-1(3.62E-2)	7.92E-1(3E-2)	**7.92E-1(2.64E-2)**	7.62E-1(2.17E-2)	7.63E-1(2.7E-2)
	WFG7	4.83E-1(1.68E-2)	4.85E-1(2.06E-2)	4.82E-1(2.5E-2)	4.88E-1(2.15E-2)	4.75E-1(3.69E-2)	**4.98E-1(5.47E-2)**

APPENDIX D

DETAILED RESULTS FOR DIFFERENT H-MOPSO STRATEGIES

In this section it is illustrated the detailed results for different H-MOPSO strategy for each problem instance and objective number (Tables 6.27 to D.8).

Table D.1: Mean (and standard deviation) for RANDOMRANDOM, RANDOM and RANDOMFIXED for the IGD indicator

Obj.	problem	RANDOMRANDOM	RANDOM	RANDOMFIXED
3	DTLZ1	7.04E-3(5.42E-4)	**6.76E-3(4.47E-4)**	6.85E-3(8.65E-4)
	DTLZ2	**9.46E-3(5.64E-4)**	9.89E-3(1.17E-3)	1.02E-2(1.51E-3)
	DTLZ3	1.41E-2(1.26E-2)	**1.21E-2(8.53E-3)**	1.24E-2(5.12E-3)
	DTLZ4	9.03E-3(5.95E-4)	**8.99E-3(8.66E-4)**	1.42E-2(4.89E-3)
	WFG6	1.15E-2(1.72E-3)	1.11E-2(1.52E-3)	**1.07E-2(1.44E-3)**
	WFG7	1.58E-2(2.45E-3)	1.62E-2(2.63E-3)	**1.58E-2(2.5E-3)**
5	DTLZ1	1.19E-2(5.96E-4)	**1.16E-2(4.21E-4)**	1.19E-2(9.33E-4)
	DTLZ2	1.79E-2(5.3E-4)	**1.77E-2(7.23E-4)**	1.91E-2(1.32E-3)
	DTLZ3	1.84E-2(1.52E-3)	**1.8E-2(7.68E-4)**	2.81E-2(1.9E-3)
	DTLZ4	**1.55E-2(7.75E-4)**	1.55E-2(5.29E-4)	2.21E-2(2.54E-3)
	WFG6	1.58E-2(6.93E-4)	1.58E-2(6E-4)	**1.53E-2(8.66E-4)**
	WFG7	**1.89E-2(9.9E-4)**	1.92E-2(9.95E-4)	2E-2(1.94E-3)
8	DTLZ1	**2.62E-2(1.47E-3)**	2.64E-2(1.78E-3)	3.06E-2(2.53E-3)
	DTLZ2	4.88E-2(2.9E-3)	**4.79E-2(3.4E-3)**	5.17E-2(3.93E-3)
	DTLZ3	**5.32E-2(3.93E-3)**	5.47E-2(4.78E-3)	6.29E-2(3.45E-3)
	DTLZ4	3.61E-2(3.47E-3)	**3.56E-2(4.2E-3)**	4.19E-2(2.5E-3)
	WFG6	4.29E-2(2.95E-3)	4.28E-2(3.98E-3)	**4.05E-2(2.75E-3)**
	WFG7	4.37E-2(3.95E-3)	**4.34E-2(3.63E-3)**	4.53E-2(4.85E-3)
10	DTLZ1	**2.03E-2(1.12E-3)**	2.05E-2(1.03E-3)	2.49E-2(1.3E-3)
	DTLZ2	4.21E-2(2.51E-3)	**4.17E-2(2.45E-3)**	4.47E-2(2.48E-3)
	DTLZ3	**4.19E-2(2.51E-3)**	4.39E-2(3.12E-3)	5.11E-2(2.64E-3)
	DTLZ4	2.75E-2(1.2E-3)	**2.74E-2(1.15E-3)**	3.23E-2(1.51E-3)
	WFG6	3.65E-2(2.29E-3)	3.64E-2(3.17E-3)	**3.55E-2(2.04E-3)**
	WFG7	5.95E-2(3.75E-3)	6.05E-2(3.57E-3)	**5.57E-2(3.97E-3)**
15	DTLZ1	5.04E-2(2.25E-3)	**4.98E-2(2.22E-3)**	5.67E-2(1.87E-3)
	DTLZ2	**8.8E-2(3.08E-3)**	8.86E-2(2.83E-3)	9.18E-2(2.52E-3)
	DTLZ3	**8.93E-2(4.04E-3)**	8.95E-2(2.53E-3)	9.59E-2(2.69E-3)
	DTLZ4	5.34E-2(2.7E-3)	**5.3E-2(2.71E-3)**	5.37E-2(2.18E-3)
	WFG6	1E-1(1.5E-2)	**9.87E-2(1.01E-2)**	1.11E-1(1.29E-2)
	WFG7	1.84E-1(1.74E-2)	1.8E-1(8.49E-3)	**1.62E-1(1.55E-2)**

Table D.2: Mean (and standard deviation) for RANDOMRANDOM, RANDOM and RANDOMFIXED for the HV indicator

Obj.	problem	RANDOMRANDOM	RANDOM	RANDOMFIXED
3	DTLZ1	7.64E-1(7.03E-3)	7.68E-1(3.85E-3)	**7.69E-1(8.29E-3)**
	DTLZ2	**7.93E-1(2.74E-3)**	7.92E-1(4.96E-3)	7.84E-1(1.81E-2)
	DTLZ3	1E0(5.51E-5)	**1E0(9.1E-6)**	1E0(5.61E-6)
	DTLZ4	8.65E-1(1.17E-3)	**8.65E-1(1.16E-3)**	8.62E-1(3.94E-3)
	WFG6	3.77E-1(9.79E-3)	**3.8E-1(8.88E-3)**	3.78E-1(8.52E-3)
	WFG7	3.9E-1(1.81E-2)	3.9E-1(1.86E-2)	**3.95E-1(1.56E-2)**
5	DTLZ1	9.71E-1(4.12E-3)	**9.73E-1(2.71E-3)**	9.68E-1(9.94E-3)
	DTLZ2	**9.94E-1(4.65E-4)**	9.94E-1(7.4E-4)	9.92E-1(2.08E-3)
	DTLZ3	1E0(0E0)	**1E0(0E0)**	1E0(4.91E-9)
	DTLZ4	**9.99E-1(6.18E-5)**	9.99E-1(8.45E-5)	9.99E-1(9.44E-5)
	WFG6	5.38E-1(2.56E-2)	5.35E-1(1.58E-2)	**5.59E-1(3.03E-2)**
	WFG7	**5.52E-1(1.77E-2)**	5.48E-1(1.56E-2)	5.25E-1(2.58E-2)
8	DTLZ1	**1E0(4.07E-10)**	1E0(1.74E-8)	1E0(1.44E-8)
	DTLZ2	9.96E-1(1.52E-3)	**9.97E-1(1.21E-3)**	9.95E-1(5.51E-3)
	DTLZ3	1E0(0E0)	**1E0(0E0)**	1E0(3.91E-9)
	DTLZ4	1E0(1.59E-5)	1E0(8.49E-5)	**1E0(8.43E-6)**
	WFG6	6.54E-1(3.12E-2)	6.55E-1(3.1E-2)	**7.01E-1(2.74E-2)**
	WFG7	4.53E-1(3.85E-2)	**4.78E-1(2.84E-2)**	4.25E-1(1.85E-2)
10	DTLZ1	**1E0(7.81E-10)**	1E0(5.55E-8)	1E0(1.1E-9)
	DTLZ2	9.98E-1(1.1E-3)	**9.98E-1(1.39E-3)**	9.96E-1(1.7E-3)
	DTLZ3	1E0(0E0)	**1E0(0E0)**	1E0(1.39E-9)
	DTLZ4	1E0(6.1E-7)	1E0(2.92E-6)	**1E0(3.29E-7)**
	WFG6	6.8E-1(2.81E-2)	6.71E-1(4.4E-2)	**7.3E-1(2.83E-2)**
	WFG7	4.51E-1(2.55E-2)	**4.57E-1(2.71E-2)**	4.38E-1(1.83E-2)

Table D.3: Mean (and standard deviation) for ACFACF, ACF and ACFFIXED for the IGD indicator

Obj.	problem	ACFACF	ACF	ACFFIXED
3	DTLZ1	9.98E-3(3.3E-3)	6.83E-3(3.75E-4)	**6.63E-3(6.86E-4)**
	DTLZ2	1.11E-2(8.93E-4)	**9.56E-3(5.87E-4)**	9.57E-3(1.04E-3)
	DTLZ3	1.75E-2(9.95E-3)	**9.83E-3(2.19E-3)**	1.32E-2(1.06E-2)
	DTLZ4	1.5E-2(2.35E-3)	**9.34E-3(1.2E-3)**	1.41E-2(5.69E-3)
	WFG6	1.12E-2(1.22E-3)	1.06E-2(1.41E-3)	**1.02E-2(1.69E-3)**
	WFG7	1.74E-2(2.05E-3)	1.51E-2(2.06E-3)	**1.34E-2(1.9E-3)**
5	DTLZ1	1.31E-2(1.69E-3)	1.19E-2(6.24E-4)	**1.13E-2(6.31E-4)**
	DTLZ2	2.04E-2(5.63E-4)	**1.77E-2(7.89E-4)**	1.81E-2(1.08E-3)
	DTLZ3	2.98E-2(8.01E-3)	**1.85E-2(1.02E-3)**	2.84E-2(2.8E-3)
	DTLZ4	1.98E-2(3.93E-3)	**1.55E-2(4.73E-4)**	2.22E-2(1.45E-3)
	WFG6	1.65E-2(1.72E-3)	1.57E-2(8.16E-4)	**1.49E-2(6.8E-4)**
	WFG7	2.02E-2(7.72E-4)	1.86E-2(6.25E-4)	**1.78E-2(6.89E-4)**
8	DTLZ1	3.59E-2(7.6E-3)	**2.52E-2(1.7E-3)**	2.94E-2(2.5E-3)
	DTLZ2	**4.75E-2(2.92E-3)**	4.78E-2(2.44E-3)	5.08E-2(2.81E-3)
	DTLZ3	1.68E-1(4.44E-1)	**5.14E-2(3.87E-3)**	6.19E-2(3.8E-3)
	DTLZ4	4.03E-2(2.74E-3)	**3.6E-2(2.86E-3)**	4.19E-2(1.66E-3)
	WFG6	4.08E-2(4.53E-3)	4.14E-2(2.81E-3)	**3.92E-2(3.09E-3)**
	WFG7	4.66E-2(4.21E-3)	4.52E-2(2.95E-3)	**4.3E-2(4.94E-3)**
10	DTLZ1	2.51E-2(4.38E-3)	**2.03E-2(8.68E-4)**	2.33E-2(1.78E-3)
	DTLZ2	**4.02E-2(2.25E-3)**	4.07E-2(2.74E-3)	4.52E-2(3.07E-3)
	DTLZ3	5.22E-2(5.5E-3)	**4.18E-2(2E-3)**	5.16E-2(2.22E-3)
	DTLZ4	3.06E-2(1.19E-3)	**2.72E-2(1.22E-3)**	3.23E-2(1.4E-3)
	WFG6	3.68E-2(2.72E-3)	3.68E-2(3.29E-3)	**3.55E-2(2.94E-3)**
	WFG7	6E-2(3.07E-3)	5.93E-2(4.77E-3)	**5.43E-2(6.13E-3)**
15	DTLZ1	5.76E-2(5.65E-3)	**5.02E-2(1.85E-3)**	5.56E-2(2.12E-3)
	DTLZ2	8.76E-2(3.75E-3)	**8.71E-2(3.54E-3)**	9.25E-2(2.5E-3)
	DTLZ3	**1.03E-1(4.53E-3)**	4.56E-1(6.95E-1)	3.22E-1(5.77E-1)
	DTLZ4	5.42E-2(4.19E-3)	**5.2E-2(2.81E-3)**	5.31E-2(2.26E-3)
	WFG6	1.04E-1(1.68E-2)	1.07E-1(1.87E-2)	**1.03E-1(1.23E-2)**
	WFG7	1.79E-1(1.65E-2)	1.87E-1(1.53E-2)	**1.78E-1(1.48E-2)**

Table D.4: Mean (and standard deviation) for ACFACF, ACF and ACFFIXED for the HV indicator

Obj.	problem	ACFACF	ACF	ACFFIXED
3	DTLZ1	9.87E-1(4.09E-2)	9.9E-1(1.42E-2)	**9.95E-1(1.02E-2)**
	DTLZ2	8.05E-1(3.49E-3)	8.07E-1(7.47E-3)	**8.08E-1(9.3E-3)**
	DTLZ3	1E0(2.96E-5)	**1E0(5.47E-6)**	1E0(9.16E-4)
	DTLZ4	6.46E-1(1.52E-2)	**6.66E-1(2.51E-3)**	6.6E-1(9.41E-3)
	WFG6	3.83E-1(1.73E-2)	4.02E-1(9.28E-3)	**4.05E-1(1.22E-2)**
	WFG7	4.63E-1(1.8E-2)	4.95E-1(1.35E-2)	**5.18E-1(1.21E-2)**
5	DTLZ1	9.96E-1(9.18E-3)	**9.99E-1(1.83E-3)**	9.97E-1(3.39E-3)
	DTLZ2	9.97E-1(3.21E-4)	**9.97E-1(1.5E-4)**	9.96E-1(6.03E-4)
	DTLZ3	1E0(4.08E-10)	**1E0(0E0)**	1E0(8.62E-10)
	DTLZ4	9.98E-1(3.29E-4)	**9.98E-1(6.96E-5)**	9.98E-1(8.72E-5)
	WFG6	5.78E-1(4.4E-2)	5.94E-1(2.9E-2)	**6.13E-1(1.65E-2)**
	WFG7	5.32E-1(2.6E-2)	5.51E-1(1.53E-2)	**5.51E-1(1.51E-2)**
8	DTLZ1	**1E0(1.7E-9)**	1E0(3.51E-7)	1E0(9.87E-9)
	DTLZ2	**9.97E-1(3.1E-3)**	9.97E-1(1.66E-3)	9.97E-1(1.73E-3)
	DTLZ3	1E0(7.88E-8)	**1E0(0E0)**	1E0(1.98E-9)
	DTLZ4	1E0(4.3E-5)	1E0(1.53E-5)	**1E0(7.26E-6)**
	WFG6	6.02E-1(6.55E-2)	5.79E-1(3.79E-2)	**6.42E-1(3.5E-2)**
	WFG7	**4.91E-1(4.35E-2)**	4.37E-1(3.32E-2)	4.19E-1(2.54E-2)
10	DTLZ1	1E0(2.61E-8)	**1E0(2.61E-9)**	1E0(1.27E-8)
	DTLZ2	9.98E-1(2.28E-3)	**9.99E-1(7.65E-4)**	9.97E-1(1.75E-3)
	DTLZ3	1E0(5.09E-10)	**1E0(0E0)**	1E0(8.74E-10)
	DTLZ4	1E0(5.76E-6)	1E0(1.12E-6)	**1E0(3.03E-7)**
	WFG6	7.09E-1(5.14E-2)	6.83E-1(4.3E-2)	**7.46E-1(4.22E-2)**
	WFG7	**5.06E-1(3.5E-2)**	4.46E-1(3.06E-2)	4.36E-1(2.35E-2)

Table D.5: Mean (and standard deviation) for UCBUCB, UCB and UCBFIXED for the IGD indicator

Obj.	problem	UCBUCB	UCB	UCBFIXED
3	DTLZ1	**6.94E-3(4.56E-4)**	7.17E-3(8.31E-4)	7.33E-3(1.42E-3)
	DTLZ2	**9.67E-3(7.07E-4)**	9.74E-3(9.24E-4)	1.03E-2(9.51E-4)
	DTLZ3	1.19E-2(8.5E-3)	**1.06E-2(2.37E-3)**	1.84E-2(1.36E-2)
	DTLZ4	**9.42E-3(1.26E-3)**	1E-2(2.35E-3)	1.56E-2(5.14E-3)
	WFG6	**1.15E-2(1.38E-3)**	1.2E-2(1.75E-3)	1.19E-2(1.96E-3)
	WFG7	1.6E-2(2.6E-3)	1.67E-2(2.76E-3)	**1.58E-2(2.91E-3)**
5	DTLZ1	1.19E-2(8.19E-4)	**1.16E-2(5.27E-4)**	1.17E-2(7.95E-4)
	DTLZ2	**1.77E-2(8.56E-4)**	1.79E-2(6.57E-4)	1.94E-2(1.38E-3)
	DTLZ3	**1.8E-2(7.2E-4)**	1.81E-2(9.4E-4)	2.82E-2(2.54E-3)
	DTLZ4	1.56E-2(5.14E-4)	**1.55E-2(6.95E-4)**	2.16E-2(2.67E-3)
	WFG6	1.57E-2(4.22E-4)	1.57E-2(7.61E-4)	**1.53E-2(4.57E-4)**
	WFG7	**1.9E-2(1E-3)**	1.95E-2(1.24E-3)	1.95E-2(1.64E-3)
8	DTLZ1	**2.52E-2(1.36E-3)**	2.54E-2(2.26E-3)	3.06E-2(2E-3)
	DTLZ2	**4.74E-2(1.9E-3)**	4.81E-2(2.97E-3)	5.1E-2(2.96E-3)
	DTLZ3	**5.22E-2(3.91E-3)**	5.83E-2(8.88E-3)	6.29E-2(3.93E-3)
	DTLZ4	3.57E-2(3.11E-3)	**3.46E-2(2.89E-3)**	4.25E-2(1.75E-3)
	WFG6	4.25E-2(3.96E-3)	4.16E-2(3.1E-3)	**3.89E-2(2.71E-3)**
	WFG7	4.62E-2(4.63E-3)	4.53E-2(3.63E-3)	**4.46E-2(4.8E-3)**
10	DTLZ1	**2.05E-2(8.85E-4)**	2.09E-2(1.43E-3)	2.45E-2(2.26E-3)
	DTLZ2	4.12E-2(2.44E-3)	**4.06E-2(1.65E-3)**	4.55E-2(3.11E-3)
	DTLZ3	**4.34E-2(3.09E-3)**	4.56E-2(5.94E-3)	5.07E-2(2.26E-3)
	DTLZ4	2.7E-2(9.31E-4)	**2.69E-2(1.05E-3)**	3.23E-2(1.28E-3)
	WFG6	3.69E-2(2.78E-3)	3.59E-2(2.51E-3)	**3.48E-2(2.72E-3)**
	WFG7	5.91E-2(3.39E-3)	5.92E-2(3.84E-3)	**5.5E-2(4.46E-3)**
15	DTLZ1	5.04E-2(2.57E-3)	**5.02E-2(1.47E-3)**	5.55E-2(2.05E-3)
	DTLZ2	8.82E-2(3.19E-3)	**8.72E-2(2.62E-3)**	9.17E-2(2.95E-3)
	DTLZ3	**8.9E-2(2.15E-3)**	8.92E-2(3.84E-3)	9.56E-2(2.18E-3)
	DTLZ4	5.31E-2(3.5E-3)	5.34E-2(3.31E-3)	**5.27E-2(1.09E-3)**
	WFG6	9.72E-2(1.08E-2)	**9.6E-2(8.81E-3)**	1.08E-1(8.7E-3)
	WFG7	1.84E-1(1.38E-2)	1.84E-1(1.21E-2)	**1.65E-1(9.92E-3)**

Table D.6: Mean (and standard deviation) for UCBUCB, UCB and UCBFIXED for the HV indicator

Obj.	problem	UCBUCB	UCB	UCBFIXED
3	DTLZ1	7.75E-1(5.72E-3)	**7.76E-1(5.38E-3)**	7.74E-1(1.1E-2)
	DTLZ2	**6.95E-1(3.52E-3)**	6.94E-1(5.03E-3)	6.84E-1(1.2E-2)
	DTLZ3	**1E0(8.66E-6)**	1E0(1.02E-5)	1E0(8.89E-5)
	DTLZ4	**5.38E-1(4.75E-3)**	5.35E-1(7.88E-3)	5.27E-1(1.14E-2)
	WFG6	**3.75E-1(6.14E-3)**	3.73E-1(1.14E-2)	3.71E-1(1.1E-2)
	WFG7	3.9E-1(1.56E-2)	3.87E-1(1.71E-2)	**4.01E-1(2.25E-2)**
5	DTLZ1	9.71E-1(3.79E-3)	**9.71E-1(4.39E-3)**	9.7E-1(5.91E-3)
	DTLZ2	**9.94E-1(1.25E-3)**	9.94E-1(2.44E-3)	9.91E-1(1.85E-3)
	DTLZ3	**1E0(0E0)**	1E0(0E0)	1E0(3.11E-9)
	DTLZ4	9.99E-1(5.41E-5)	**9.99E-1(5.09E-5)**	9.99E-1(6.62E-5)
	WFG6	5.43E-1(2.08E-2)	5.42E-1(1.83E-2)	**5.49E-1(2.54E-2)**
	WFG7	**5.34E-1(1.67E-2)**	5.26E-1(1.77E-2)	5.14E-1(2.33E-2)
8	DTLZ1	1E0(2.43E-8)	**1E0(4.2E-9)**	1E0(4.01E-8)
	DTLZ2	9.96E-1(2.32E-3)	**9.97E-1(1.22E-3)**	9.95E-1(1.76E-3)
	DTLZ3	**1E0(0E0)**	1E0(1.13E-10)	1E0(1.9E-8)
	DTLZ4	1E0(4.02E-5)	1E0(1.74E-5)	**1E0(8.94E-6)**
	WFG6	5.91E-1(3.88E-2)	6.07E-1(4.33E-2)	**6.3E-1(2.65E-2)**
	WFG7	**4.81E-1(3.38E-2)**	4.71E-1(3.3E-2)	4.52E-1(2.27E-2)
10	DTLZ1	1E0(3.71E-9)	1E0(7.63E-8)	**1E0(1.4E-10)**
	DTLZ2	**9.98E-1(8.8E-4)**	9.98E-1(1.34E-3)	9.96E-1(2.57E-3)
	DTLZ3	**1E0(0E0)**	1E0(0E0)	1E0(8.18E-10)
	DTLZ4	1E0(1.51E-6)	1E0(1.17E-6)	**1E0(3.81E-7)**
	WFG6	6.51E-1(6.19E-2)	6.69E-1(4.13E-2)	**7.24E-1(3.49E-2)**
	WFG7	**4.67E-1(2.26E-2)**	4.59E-1(1.9E-2)	4.56E-1(2.45E-2)

Table D.7: Mean (and standard deviation) for ROULETTEROULETTE, ROULETTE and ROULETTEFIXED for the IGD indicator

Obj.	problem	ROULETTEROULETTE	ROULETTE	ROULETTEFIXED
3	DTLZ1	9.67E-3(1.59E-3)	**6.69E-3(5.11E-4)**	6.98E-3(7.16E-4)
	DTLZ2	1.11E-2(1.43E-3)	9.66E-3(1.01E-3)	**9.29E-3(7.06E-4)**
	DTLZ3	1.79E-2(1.25E-2)	**1.35E-2(1.1E-2)**	1.6E-2(1.48E-2)
	DTLZ4	2.11E-2(4.47E-3)	**1.14E-2(3.23E-3)**	2.38E-2(5.25E-3)
	WFG6	1.18E-2(1.52E-3)	**1E-2(8.04E-4)**	1.02E-2(1.03E-3)
	WFG7	1.63E-2(3.1E-3)	1.3E-2(4.86E-4)	**1.23E-2(7.38E-4)**
5	DTLZ1	1.26E-2(1.96E-3)	**1.08E-2(5.93E-4)**	1.09E-2(8.37E-4)
	DTLZ2	1.91E-2(3.07E-3)	1.71E-2(9.74E-4)	**1.64E-2(3.88E-4)**
	DTLZ3	2.85E-2(4.68E-3)	**1.98E-2(3.53E-3)**	2.91E-2(2.96E-3)
	DTLZ4	2.46E-2(4.84E-3)	**1.68E-2(1.6E-3)**	2.65E-2(4.59E-3)
	WFG6	1.67E-2(1.6E-3)	1.55E-2(8.82E-4)	**1.53E-2(4.47E-4)**
	WFG7	2.26E-2(3.62E-3)	1.83E-2(8.68E-4)	**1.75E-2(7.06E-4)**
8	DTLZ1	3.45E-2(4.57E-3)	**2.47E-2(1.44E-3)**	2.9E-2(3.11E-3)
	DTLZ2	5.35E-2(4.5E-3)	**5.08E-2(3.72E-3)**	5.19E-2(3.84E-3)
	DTLZ3	6.87E-2(8.4E-3)	**5.35E-2(5.13E-3)**	6.68E-2(4.67E-3)
	DTLZ4	4.79E-2(3.17E-3)	**3.84E-2(2.53E-3)**	4.74E-2(2.24E-3)
	WFG6	4.26E-2(4.2E-3)	4.36E-2(3.67E-3)	**4.01E-2(2.84E-3)**
	WFG7	5.64E-2(5.36E-3)	**4.75E-2(3.98E-3)**	5.03E-2(6.09E-3)
10	DTLZ1	2.45E-2(4.05E-3)	**1.93E-2(9.84E-4)**	2.39E-2(1.68E-3)
	DTLZ2	4.71E-2(2.11E-3)	**4.46E-2(1.8E-3)**	4.65E-2(2.08E-3)
	DTLZ3	5.71E-2(4.81E-3)	**4.45E-2(4.29E-3)**	5.47E-2(3.69E-3)
	DTLZ4	3.89E-2(2.93E-3)	**2.97E-2(2.2E-3)**	3.7E-2(1.05E-3)
	WFG6	3.66E-2(2.54E-3)	3.82E-2(2.73E-3)	**3.58E-2(2.35E-3)**
	WFG7	7.15E-2(9.05E-3)	**6.46E-2(4.88E-3)**	6.46E-2(3.47E-3)
15	DTLZ1	5.83E-2(4.4E-3)	**5.17E-2(2.16E-3)**	5.43E-2(2.49E-3)
	DTLZ2	9.26E-2(2.46E-3)	**9.03E-2(2.94E-3)**	9.17E-2(2.39E-3)
	DTLZ3	2.11E-1(4.75E-1)	**9.27E-2(3.99E-3)**	9.78E-2(3.48E-3)
	DTLZ4	6.03E-2(7.1E-3)	**5.47E-2(3.75E-3)**	5.66E-2(4.04E-3)
	WFG6	1.04E-1(8.87E-3)	**9.52E-2(8.29E-3)**	9.79E-2(7.75E-3)
	WFG7	2.37E-1(7.12E-2)	**2.05E-1(1.63E-2)**	2.18E-1(1.97E-2)

Table D.8: Mean (and standard deviation) for ROULETTEROULETTE, ROULETTE and ROULETTEFIXED for the HV indicator

Obj.	problem	ROULETTEROULETTE	ROULETTE	ROULETTEFIXED
3	DTLZ1	7.53E-1(1.87E-2)	7.81E-1(6.55E-3)	**7.81E-1(5.6E-3)**
	DTLZ2	4.39E-1(1.12E-2)	4.51E-1(6.73E-3)	**4.58E-1(5.88E-3)**
	DTLZ3	1E0(9.08E-4)	**1E0(1.31E-4)**	1E0(6.74E-4)
	DTLZ4	3.9E-1(2.17E-2)	**4.33E-1(7.26E-3)**	3.83E-1(2.89E-2)
	WFG6	3.73E-1(1.48E-2)	**3.93E-1(7.87E-3)**	3.9E-1(1.29E-2)
	WFG7	4.35E-1(4.08E-2)	4.66E-1(1.02E-2)	**4.83E-1(1.59E-2)**
5	DTLZ1	9.98E-1(6.3E-4)	9.98E-1(3.47E-4)	**9.98E-1(2.23E-4)**
	DTLZ2	9.53E-1(6.25E-3)	**9.58E-1(3.79E-3)**	9.58E-1(2.06E-3)
	DTLZ3	1E0(3.84E-9)	**1E0(1.14E-9)**	1E0(5.07E-8)
	DTLZ4	9.92E-1(1.27E-3)	**9.94E-1(4.73E-4)**	9.92E-1(8.53E-4)
	WFG6	5.9E-1(3.06E-2)	6.09E-1(1.32E-2)	**6.21E-1(9.96E-3)**
	WFG7	5.41E-1(7.09E-2)	**6.39E-1(1.61E-2)**	6.34E-1(1.56E-2)
8	DTLZ1	1E0(1.23E-8)	1E0(1.47E-8)	**1E0(3.74E-9)**
	DTLZ2	9.93E-1(3.25E-3)	9.94E-1(1.71E-3)	**9.94E-1(2.55E-3)**
	DTLZ3	1E0(5.51E-8)	**1E0(2.24E-11)**	1E0(3.08E-7)
	DTLZ4	1E0(8.26E-5)	**1E0(2.1E-5)**	1E0(8.28E-6)
	WFG6	5.85E-1(9.04E-2)	5.66E-1(4.9E-2)	**6.63E-1(3E-2)**
	WFG7	3.56E-1(4.08E-2)	**4.34E-1(2.62E-2)**	3.94E-1(2.44E-2)
10	DTLZ1	**1E0(2.17E-10)**	1E0(7.35E-9)	1E0(3.8E-8)
	DTLZ2	**9.98E-1(1.29E-3)**	9.97E-1(1.82E-3)	9.98E-1(9.1E-4)
	DTLZ3	1E0(1.81E-9)	**1E0(4.89E-11)**	1E0(1.43E-8)
	DTLZ4	1E0(1.67E-5)	**1E0(1.49E-6)**	1E0(6.5E-7)
	WFG6	6.17E-1(8.29E-2)	6.7E-1(5.63E-2)	**7.64E-1(2.25E-2)**
	WFG7	4.08E-1(5.29E-2)	**4.96E-1(3.86E-2)**	4.52E-1(1.95E-2)