Noname manuscript No.
(will be inserted by the editor)

Evolutionary Many-objective Optimization based on Linear Assignment Problem Transformations

Luis Miguel Antonio · José A. Molinet Berenguer · Carlos A. Coello Coello

Received: date / Accepted: date

Abstract The selection mechanisms that are most commonly adopted by Multi-Objective Evolutionary Algorithms (MOEAs) are based on Pareto optimality. However, recent studies have provided theoretical and experimental evidence regarding the unsuitability of Pareto-based selection mechanisms when dealing with problems having four or more objectives. In this paper, we propose a novel MOEA designed for solving many-objective optimization problems. The selection mechanism of our approach is based on the transformation of a multi-objective optimization problem into a linear assignment problem (LAP), which is solved by the Kuhn-Munkres' (Hungarian) algorithm. Our proposed approach is compared with respect to three state-of-the-art MOEAs, designed for solving many-objective optimization problems (i.e., problems having four or more objectives), adopting standard test problems and performance indicators taken from the specialized literature. Since one of our main aims was to analyze the scalability of our proposed approach, its validation was performed adopting test problems having from two to nine objective functions. Our preliminary experimental results indicate that our proposal is very competitive with respect to all the other MOEAs compared, obtaining the best results in several of the test problems adopted, but at a significantly lower computational cost.

Keywords Multi-objective Optimization, Many-objective Optimization · Evolutionary Computation

The third author acknowledges support from CONACyT project no. 221551.

Computer Science Department, CINVESTAV-IPN
Av. Instituto Politécnico Nacional 2508, Gustavo A. Madero, San Pedro Zacatenco, 07360 Mexico City, Mexico
Tel.: +52-1-5531992612
E-mail: lmiguel@computacion.cs.cinvestav.mx, ccoello@cs.cinvestav.mx

1 Introduction

A large number of problems arise in academic and industrial areas, where there is a need to optimize several conflicting objectives simultaneously [11]. They are known as multi-objective optimization problems (MOPs). In general, MOPs do not have a single solution, but a set of them, representing the best possible trade-offs among all the objective functions of the problem.

The notion of optimality normally adopted in multi-objective optimization is *Pareto optimality*, which refers to the process of finding the best possible trade-offs among the objective functions of an MOP. These solutions constitute the *Pareto Optimal Set*. The hyper-surface formed by the Pareto optimal solutions when plotted in objective function space, is called the *Pareto Optimal Front*. Among the different techniques available to solve MOPs, multi-objective evolutionary algorithms (MOEAs) have become very popular, mainly due to their flexibility, their capability of approximating the Pareto Optimal Set in a single run and their effectiveness in a wide variety of problems [11]. When solving an MOP, we normally aim to minimize the distance between the approximation found and the Pareto Optimal Front, while obtaining a distribution of solutions as uniform as possible along the Pareto Optimal Front. It is important to note, however, that although the distance between the approximation set generated by a MOEA and the Pareto Optimal Front is normally available in benchmark problems, in real-world applications, this value is typically unknown.

For several years, the selection mechanisms that were most commonly adopted by MOEAs were those based on Pareto optimality. However, recent studies have provided theoretical and experimental evidence regarding the unsuitability of Pareto-based selection mechanisms when dealing with problems having four or more objectives (the so-called *many-objective optimization problems*) [20,32]. In such cases, the number of nondominated solutions significantly increases, which makes harder to generate selection pressure, giving rise to a phenomenon known as *dominance resistance* [24,32]. Dominance resistance can influence the performance of MOEAs in several ways. For example, this phenomenon may limit the ability of the dominance relation to distinguish between good-quality and poor-quality solutions.

A strategy to improve the scalability of MOEAs to deal with many-objective problems is to increase the selection pressure towards the Pareto Optimal Front. For this sake, many researchers have created alternative dominance relations to provide more strict dominance criteria (e.g., preferability, preferred, ϵ-preferred, k-optimality, and preference order ranking [20]). Nevertheless, using a different dominance relation for selection may guide the search towards a specific subspace and it could consequently fail to produce well-spread solutions along the entire Pareto Optimal Front [11]. Besides, as the proportion of locally Pareto non-dominated solutions increases and the generated solutions are likely to be also non-dominated, diversity operators become the primary mechanism for determining survival. However, the constant application of diversity maintenance mechanisms can cause deterioration of solutions and can prevent the convergence of a MOEA [32].

In this work, we propose to use an alternative selection mechanism which is not based on Pareto optimality or on any performance indicator and which we argue that is part of a different family of selection mechanisms for MOEAs. Our proposed approach performs a transformation of the original MOP into a linear assignment problem which is then solved using the Kuhn-Munkres (Hungarian) algorithm [26]. As will be seen below, this paper extends and improves the proposal that we originally introduced in [4]. We will show how our proposed approach represents a novel choice for solving many-objective optimization problems in an effective and efficient manner.

The remainder of this paper is organized as follows. Section 2 states the problem of our interest. Thereafter, in Section 3, we present a brief discussion of the most relevant previous related work. In Section 4, we describe in detail our proposed approach. The experiments performed and the results obtained are described and discussed in Section 5. Finally, our conclusions and some possible paths for future work are briefly discussed in Section 6.

2 The Multi-objective Optimization Problem

Without loss of generality, we will assume only minimization problems. We are interested in solving problems of the type:

$$\text{minimize } \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})] \tag{1}$$

subject to:

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

$$h_i(\mathbf{x}) = 0 \quad i = 1, 2, \ldots, p \tag{3}$$

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

To describe the concept of optimality in which we are interested, we will introduce next a few definitions [11].

Definition 1. Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^k$, we say that $\mathbf{u} \leq \mathbf{v}$ if $u_i \leq v_i$ for $i = 1, ..., k$, and that $\mathbf{u} < \mathbf{v}$ if $\mathbf{u} \leq \mathbf{v}$ and $\mathbf{u} \neq \mathbf{v}$.

Definition 2. Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^k$, we say that **u dominates v** (denoted by $\mathbf{u} \prec \mathbf{v}$) iff $\mathbf{u} < \mathbf{v}$.

Definition 3. We say that a vector of decision variables $\mathbf{x}^* \in \mathcal{F}$ ($\mathcal{F}$ is the feasible region) is **Pareto optimum** if there does not exist another $\mathbf{x} \in \mathcal{F}$

such that $\mathbf{f}(\mathbf{x}) \prec \mathbf{f}(\mathbf{x}^*)$.

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

$$\mathcal{P}^* = \{\mathbf{x} \in \mathcal{F} | \mathbf{x} \text{ is Pareto optimum}\}$$

Thus, the Pareto Optimal Set contains solutions in which it is not possible to improve one objective without worsening another. The vectors $\mathbf{x}^*$ corresponding to the solutions included in the Pareto Optimal Set are called *nondominated*.

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

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

When plotted in objective function space, the nondominated vectors contained in the Pareto Optimal Set are collectivelly known as the Pareto Optimal Front.

Evidently, our aim is to determine the Pareto Optimal Set from the set $\mathcal{F}$ of all the decision variable vectors that satisfy (2) and (3).

3 Previous Related work

In this section, we present a brief review of the main strategies that have been proposed in the specialized literature to deal with many-objective optimization problems using MOEAs. For a comprehensive and more detailed review on many-objective optimization, the interested reader is referred to other papers (see for example [29,37,27,21]).

The Nondominated Sorting Genetic Algorithm-III (NSGA-III) [13] is one of the most popular MOEAs adopted for many-objective optimization. NSGA-III modifies the selection mechanism of NSGA-II by a clustering operator (aided by a set of well-distributed reference points). NSGA-III performs an analysis of the distances of the individuals in the population with respect to the supplied reference points, preferring population members that are non-dominated and close to such reference points.

An approach called θ-Dominance-based Evolutionary Algorithm (θ-DEA) was proposed in [40] with the aim of improving the convergence of NSGA-III while preserving its good diversity properties. θ-DEA performs the computation of reference vectors proposed by Das and Dennis [12] combined with a non-dominated sorting procedure, based on the θ-dominance relation, for the selection of new individuals. θ-dominance produces solutions that are associated to particular (well-distributed) reference points and, therefore, favors both convergence and diversity.

Decomposition is another well-established technique for dealing with MOPs, which has been found to be efficient and effective when coupled to evolutionary algorithms even when dealing with many objectives. MOEA/D [41] is

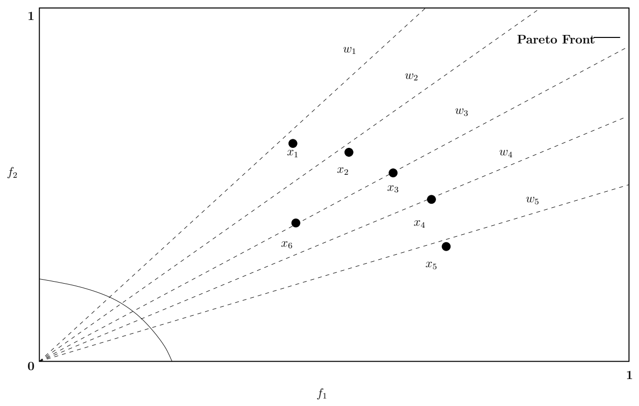


Fig. 1: Behavior of MOEA/D when solutions are replaced. For each weight vector w_i, $i = 1, 2, 3, 4, 5$, the solution x_6 has the highest utility value. Therefore, x_6 is the best solution of the four subproblems.

perhaps the most popular MOEA based on decomposition. This algorithm decomposes the MOP into a set of single-objective subproblems (by means of well-defined scalarizing functions) and solves these subproblems simultaneously using an evolutionary algorithm. It adopts a set of weights each of which corresponds to a single subproblem. Each weight vector is adopted as a search direction to define a scalar function. MOEA/D has shown to be a very good alternative to solve MOPs with low or high dimensionality (regarding objetive function space). However, MOEA/D has some important drawbacks. For instance, it creates a new solution from a fixed neighborhood and, therefore, the new solution cannot be created from solutions which belong to different neighborhoods. Also, a new solution with a high fitness value replaces several solutions and, therefore, the population can lose diversity, as depicted in Figure 1. A variant of MOEA/D proposed by Li and Zhang in [28], which is called "MOEA/D-DE", allows a new solution to be generated from solutions from different neighborhoods. This approach limits the number of solutions that can be replaced by the same new solution. However, both MOEAs assign the best individual to each subproblem in an independent way, without considering the best assignment in a global way. This is shown in Figure 2, where we depict the assignment made by MOEA/D and MOEA/D-DE.

Since Pareto-based MOEAs are unable to provide appropriate selection pressure when dealing with many-objective optimization problems [29, 22, 20] (because as the number of objectives increases, all solutions quickly become non-dominated), the use of performance indicators as selection mechanisms has become relatively popular. From the different performance indicators available in the literature, the hypervolume [42] has become the most popular choice for implementing indicator-based MOEAs. This is due to its nice theoretical properties [10]. In fact, the hypervolume is the only unary indicator that is known to be Pareto compliant and it has been proved that its maximization

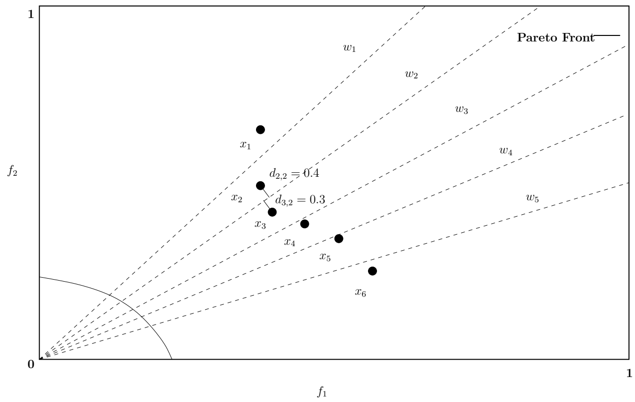


Fig. 2: Behavior of MOEA/D when solutions are selected for each subproblem. Solution x_3 is assigned to the subproblem w_2 and then solution x_2 is replaced by x_3. It is relevant to notice that solution x_2 is better than solution x_1 for the subproblem w_1. Nevertheless, both algorithms (MOEA/D and MOEA/D-DE) eliminate it.

is equivalent to finding the Pareto optimal set [16]. This has also been empirically corroborated by some researchers (see for example [5]). Additionally, maximizing the hypervolume also leads to sets of solutions whose spread along the Pareto Optimal Front is maximized (although this does not necessarily guarantee a uniform distribution along the Pareto Optimal Front).

An algorithm based on the Covariance Matrix Adaptation Pareto Archived Evolution Strategy (CMA-PAES) which makes use of the hypervolume indicator is presented in [33]. The so-called Covariance Matrix Adaptation Pareto Archived Evolution Strategy with Hypervolume-sorted Adaptive Grid Algorithm (CMA-PAES-HAGA) makes use of an adaptive grid and the archive of the Pareto Archived Evolution Strategy (PAES) [23] to perform a local approximation of the hypervolume indicator, which is then used as a selection criterion. However, the main disadvantage of adopting the hypervolume to select solutions is that the best algorithms known to compute it have a computational cost which grows exponentially on the number of objective functions of the MOP [8]. Hypervolume computation has been proven to be $\#P$-hard (analogous to NP-hard for counting problems) in the number of objective functions [7]. Although some proposals have attempted to approximate the hypervolume contributions required for using it in the selection mechanism of a MOEA (see for example [3, 9]), the currently available MOEAs that approximate the hypervolume contributions quickly degrade their performance as the number of objectives increases whereas those using exact hypervolume contributions normally become unaffordable for problems having more than five objectives.

To the best of the authors' knowledge, the only preliminary work in which some sort of transformation of an MOP is adopted is the one proposed in [39], which formulates the fitness assignment of a MOEA as a Portfolio selection problem, which takes solutions as assets whose returns are random variables. In that proposal, fitness is the investment in such assets (solutions). This gives rise to a bi-objective problem in which the aim is to maximize the expected return while minimizing the return variance (related to risk). It is worth indicating, however, that the approach proposed in [39] was validated only with 0/1 knapsack problems having up to four objectives. In contrast, our proposed approach is validated here using continuous problems with up to 9 objectives.

4 Our proposed approach

As indicated before, our proposed approach transforms the selection process of a MOEA into a linear assignment problem (LAP) and then such LAP is solved using the *Munkres assignment algorithm* [26]. As we will see later on, the intriguing aspect about our proposal is that the solution of this LAP makes a MOEA to produce a set of solutions that converge to the Pareto Optimal Front and, have, at the same time, a good distribution along it. Our proposed approach does not fall into any of the classes of MOEAs that have been traditionally used (i.e., Pareto-based, decomposition-based or indicator-based) and is, therefore, a novel mechanism to solve MOPs which we show here to be effective, particularly when dealing with many-objective optimization problems.

4.1 Linear assignment problem transformation

The assignment problem is an elementary type of combinatorial optimization problem [17, 30]. In its most general form, an assignment problem can be defined as the problem of creating an optimal assignment of n agents to m tasks, taking for granted that some numerical assortment are incurred for each agent performing each of the tasks [26]. The optimal solution to this kind of problem is an assigment which makes the sum of the agents' ratings for their tasks a maximum (or minimum, according to the problem). The *Linear Assignment Problem* (LAP) is the most basic type of assignment problem. In the canonical LAP, we have the same number of agents and tasks, and any agent can be assigned to perform any task. An LAP can be defined as follows:

Let $A = \{a_1, ..., a_n\}$ and $T = \{t_1, ..., t_m\}$ be a set of agents and tasks with the same cardinality, given a cost function $C : A \times T \to \mathbb{R}$ and having $\Phi : A \to T$ as the set of all possible bijections between A and T, the LAP can be stated as follows:

$$\underset{\phi \in \Phi}{\text{minimize}} \sum_{i=1}^{n} C(a_i, \phi(a_i)) \tag{4}$$

In most cases, the cost function can also be represented by a squared real-valued matrix C with elements $C_{ij} = C(a_i, t_j)$, and the set Φ of all possible bijections between A and T as a set of assignment matrices $\mathcal{X}$. Therefore, the LAP can be also expressed as an integer linear program as follows:

$$\begin{aligned} \underset{x \in \mathcal{X}}{\text{minimize}} \quad & \sum_{i,j=1}^{n} C_{ij} x_{ij} \\ \text{subject to:} \quad & \sum_{i=1}^{n} x_{ij} = 1, \; \forall j \in \{1,..,n\}, \\ & \sum_{j=1}^{n} x_{ij} \leq 1, \; \forall i \in \{1,...,n\}, \\ & x_{ij} \in \{0,1\}, \; \forall i,j \in \{1,...,n\} \end{aligned} \tag{5}$$

Harold W. Kuhn [26] proposed an algorithm for computing a maximum weight perfect assignment in a bipartite graph. This approach is able to solve the assignment problem in polynomial time. After this work, James Munkres [30] improved Kuhn's approach and provided several important contributions to the theoretical aspects of the algorithm. Munkres showed that Kuhn's approach has a polynomial complexity and proposed an improved version which has a complexity of $O(n^3)$. The contribution of Munkres made posible the creation of the *Hungarian algorithm* also referred to as the *Kuhn-Munkres* or *Munkres assignment algorithm.* Later on, Bourgeois and Lassalle [6] developed an extension for rectangular matrices which allows the algorithm to operate in assignment problems where the number of agents and the number of tasks are unequal. Such extension can be stated as follows:

Given an $n \times m$ matrix (c_{ij}) of real numbers, we want to find a set of k independent elements $[k = min(n, m)]$ such that the sum of these elements is minimized.

A compact description of the steps of the algorithm proposed by Bourgeois and Lassalle [6] is given next (a manual execution of the algorithm is presented in Figure 3):

1. For each row of the matrix, find the smallest element and subtract it from each element in its row.
2. Find a zero in the resulting matrix. If there is no starred zero in its row nor its column, mark that zero with a star (*). Repeat for each zero in the matrix.
3. Cover each column containing a starred zero. If n columns are covered, the starred zeros describe a complete set of unique assignments. In this case, stop; otherwise, continue with step 4.
4. Find an uncovered zero and prime it. If there is no starred zero in the row containing this primed zero, go to Step 5. Otherwise, cover this row and uncover the column containing the starred zero. Repeat this process until there are no uncovered zeros left. After saving the smallest uncovered value go to Step 6.
5. Construct a path of alternating primed and starred zeros as follows. Let Z_0 represent the uncovered primed zero found in Step 4. Let Z_1 denote the

starred zero in the column of Z_0 (if any). Let Z_2 denote the primed zero in the row of Z_1 (there will always be one). Continue until the sequence terminates at a primed zero that has no starred zero in its column. Un-star each starred zero of the sequence; star each primed zero of the sequence. Erase all primes and uncover every line in the matrix; return to Step 3.

6. Add the value found in Step 4 to every element of each covered row, and subtract it from every element of each uncovered column. Return to Step 4 without altering any stars, primes, or covered lines.

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 1 & 2 & 3 \\ a_2 & 2 & 4 & 6 \\ a_3 & 3 & 6 & 9 \end{array}$$
Cost matrix

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0 & 1 & 2 \\ a_2 & 0 & 2 & 4 \\ a_3 & 0 & 3 & 6 \end{array}$$
Step 1

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0^* & 1 & 2 \\ a_2 & 0 & 2 & 4 \\ a_3 & 0 & 3 & 6 \end{array}$$
Step 2

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0^* & 1 & 2 \\ a_2 & 0 & 2 & 4 \\ a_3 & 0 & 3 & 6 \end{array}$$
Step 3

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0^* & \boxed{1} & 2 \\ a_2 & 0 & 2 & 4 \\ a_3 & 0 & 3 & 6 \end{array}$$
Step 4

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0^* & 0 & 1 \\ a_2 & 0 & 1 & 3 \\ a_3 & 0 & 2 & 5 \end{array}$$
Step 6

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0^* & 0' & 1 \\ a_2 & 0' & 1 & 3 \\ a_3 & 0 & 2 & 5 \end{array}$$
Step 4

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & \mathbf{0} & \mathbf{0}^* & 1 \\ a_2 & \mathbf{0}^* & 1 & 3 \\ a_3 & 0 & 2 & 5 \end{array}$$
Step 5

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0 & 0^* & 1 \\ a_2 & 0^* & 1 & 3 \\ a_3 & 0 & 2 & 5 \end{array}$$
Step 3

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0 & 0^* & \boxed{1} \\ a_2 & 0^* & 1 & 3 \\ a_3 & 0 & 2 & 5 \end{array}$$
Step 4

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0 & 0^* & 0 \\ a_2 & 0^* & 1 & 2 \\ a_3 & 0 & 2 & 4 \end{array}$$
Step 6

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 0 & 0^* & 0' \\ a_2 & 0^* & \boxed{1} & 2 \\ a_3 & 0 & 2 & 4 \end{array}$$
Step 4

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 1 & 0^* & 0' \\ a_2 & 0^* & 0 & 1 \\ a_3 & 0 & 1 & 3 \end{array}$$
Step 6

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 1 & 0^* & 0' \\ a_2 & 0^* & 0' & 1 \\ a_3 & 0' & 1 & 3 \end{array}$$
Step 4

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 1 & \mathbf{0} & \mathbf{0}^* \\ a_2 & \mathbf{0} & \mathbf{0}^* & 1 \\ a_3 & \mathbf{0}^* & 1 & 3 \end{array}$$
Step 5

$$\begin{array}{cccc} & t_1 & t_2 & t_3 \\ a_1 & 1 & 0 & 0^* \\ a_2 & 0 & 0^* & 1 \\ a_3 & 0^* & 1 & 3 \end{array}$$
Step 3

Fig. 3: A manual execution of the Kuhn-Munkres' algorithm.

Here, we explain how an MOP can be transformed into an LAP using the k-dimensional objective vectors from the individuals (solutions) in a MOEA. Since we can compute uniformly spread weight vectors in objective function space, one can reformulate an MOP in the following manner: having n individuals and m vectors well-distributed in a $(k-1)$-dimensional unit simplex of the objective functions space, a cost can be incurred for each individual representing some vector in the Pareto Optimal Front approximation. Therefore, the task is to describe all regions covered by the n vectors using only m individuals in such a way that the total cost of the assignment needs to be minimized. A cost matrix can then be created so that it minimizes the total cost implied in retaining the solutions which are a good approximation of the Pareto Optimal Front. This procedure is described next.

First, the n vectors of objective function values are normalized so that the current objective function space is reduced to a unit hypercube. This allows us to deal with non-commensurable objective functions. The maximum $\mathbf{z}^{max}$

and minimal $\mathbf{z}^{min}$ vectors are computed to perform the normalization in the following way.

$$\begin{aligned} \mathbf{z}^{max} &= [z_1^{max}, ..., z_k^{max}]^T, z_i^{max} = \max_{j=1,...,n} f_i(\mathbf{x}_j), i = 1, ..., k, \\ \mathbf{z}^{min} &= [z_1^{min}, ..., z_k^{min}]^T, z_i^{min} = \min_{j=1,...,n} f_i(\mathbf{x}_j), i = 1, ..., k, \end{aligned} \tag{6}$$

where $f_i(\mathbf{x}_j)$ is the i^{th} function value of the j^{th} solution, and its normalized value $\tilde{f}_i(\mathbf{x}_j)$ is computed as:

$$\tilde{f}_i(\mathbf{x}_j) = \frac{f_i(\mathbf{x}_j) - z_i^{min}}{z_i^{max} - z_i^{min}}, \quad j = 1, ..., n, \ i = 1, ..., k. \tag{7}$$

Thereafter, let W be a set of m weight vectors uniformly scattered in objective function space.

$$W \subset \mathbb{W} = \{\mathbf{w} \mid \mathbf{w} \in [0,1]^k, \sum_{i=1}^{k} w_i = 1\}, \ |W| = m, \tag{8}$$

The cost C_{rj} of assigning the solution $\mathbf{x}_j$ to the weight vector $\mathbf{w}_r$ is given by:

$$C_{rj} = \max_{i=1,...,k} w_{ri} \times \tilde{f}_i(\mathbf{x}_j), \ r = 1, ..., m, \ j = 1, ..., n. \tag{9}$$

So, the matrix C indicates how each solution is suitable to represent each region of the Pareto Optimal Front approximation. The solution to our assignment problem adequation is obtained by finding the combination of values in C which produces the smallest sum, subject to the following conditions:

- Exactly one value must be chosen in each row, so that only one solution is assigned to each position on the Pareto Optimal Front.
- At most, one value can be selected in each column. This ensures that no solution is assigned to more than one position.

The matrix C and the above conditions are formally represented by (5) as an LAP. The solution to this problem is then computed by means of the version of the Kuhn-Munkres algorithm for rectangular matrices [6]. This approach was adopted for being able to work with LAPs where the number of agents and the number of tasks are unequal, which is the case of our problems. The matrix that solves (5) represents the solutions assigned to each weight vector such that it minimizes the total cost of the assignment, allowing to retain the best solutions to approximate the Pareto Optimal Front.

4.2 Generation of weight vectors using Uniform Design

In this section, we describe an alternative way to generate weight vectors in a more flexible and efficient manner. In the state of the art, there exist several MOEAs [41,31,13] that require a set of weight vectors uniformly scattered on a $(k-1)$-unit simplex to perform the search of solutions along the entire Pareto Optimal Front in a k-objectives MOP.

In the specialized literature we can find several methods to obtain an evenly distributed subset of weights in a simplex [15]. Among the existing approaches, the simplex-lattice design method [34] has been the most commonly used within MOEAs. Nevertheless, this approach has some drawbacks which can be easily identified [15]. One of them is that the weight vectors are not very uniformly distributed. Also, this method generates too many vectors at the boundaries of the domain. Another problem is that the number of vectors generated by this method increases nonlinearly with respect to the number of objective functions of an MOP. Hence, if H divisions are considered along each objective, the total number of weight vectors in a k-objectives problem is given by: $\binom{H+k-1}{k-1}$, which is also the population size. Therefore, when adopting this method, the size of the population is defined by this restriction and not according to the user's needs.

Some MOEAs have adopted other techniques to compute a certain (desired) number of scattered weight vectors, such as the approach reported in [31], where a hypervolume-based weight vector generator is proposed. This approach creates well-distributed vectors by maximizing the hypervolume covered by them in objective function space. Another approach of this sort can be found in [36], where the uniform design (UD) [15] and the good lattice point (*glp*) [25] methods were combined to set the weight vectors. However, both the hypervolume and the *glp* method have a high computational cost when the number of objectives grows, which makes them unaffordable when dealing with MOPs having many objectives.

Uniform design is a space-filling design method that seeks experimental points to be uniformly distributed in the domain [15]. In this method, a set of points is considered uniformly spread throughout the entire domain if it has a small *discrepancy*, where discrepancy is a numerical measure of scattering. Fang and Wang [15] proposed several methods to generate points that can be applied to the computation of a set of space-filling design points. The good lattice point (*glp*) method and Hammersley's method [18] excel among the different proposals, both of which are efficient quasi Monte Carlo methods.

Here, we generate weight vectors using uniform design combined with Hammersley's method. This approach allows a more uniform distribution of the weight vectors over the objective space of an MOP (in fact, it is even better distributed than when using the simplex-lattice method). Additionally, in this case the population size does not increase nonlinearly with the number of objectives and no special considerations need to be taken into account. Furthermore, Hammersley's method computes a set of design points with low discrepancy, very similar to the *glp* method, but at a much lower computational cost [15].

Algorithm 1 Generation of weight vectors

Require: number of objectives (k), number of weights (n)
Ensure: W (set of weight vectors with low-discrepancy)

$p \leftarrow$ array with the first $k-2$ prime numbers
$U \leftarrow \emptyset$
for $i = 1$ **to** n **do**
  $u_{i1} \leftarrow (2i-1)/2n$
  for $j = 2$ **to** $k-1$ **do**
    $u_{ij} \leftarrow 0$
    $f \leftarrow 1/p_{j-1}$
    $d \leftarrow i$
    while $d > 0$ **do**
      $u_{ij} \leftarrow u_{ij} + f \times (d \mod p_{j-1})$
      $d \leftarrow \lfloor d/p_{j-1} \rfloor$
      $f \leftarrow f/p_{j-1}$
    end while
  end for
  $U \leftarrow U \cup \{u\}$
end for
$W \leftarrow$ Apply the transformation of (13) to U

Hammersley's method is based on the p-adic representation of natural numbers. Any positive integer m can be uniquely expressed using a prime base $p \geq 2$ as:

$$m = \sum_{i=0}^{r} b_i \times p^i, \quad 0 \leq b_i \leq p-1, \quad i = 0, \ldots, r, \tag{10}$$

where $p^r \leq m < p^{r+1}$. Then, for any integer $m \geq 1$ with a representation given by (10), let

$$y_p(m) = \sum_{i=0}^{r} b_i \times p^{-(i+1)}, \tag{11}$$

where $y_p(m) \in (0,1)$ and is known as the radical inverse of m base p. Let $k \geq 2$ and $p_1, \ldots, p_{k-1}$ be $k-1$ distinct prime numbers. Then, the Hammersley set consisting of n points uniformly scattered on $[0,1]^k$ is given by

$$\mathbf{x}_i = \left[\frac{2i-1}{2n}, y_{p_1}(i), \ldots, y_{p_{k-1}}(i) \right]^T, \; i = 1, \ldots, n. \tag{12}$$

Previous work presented in [38], uses uniform design for experiments with mixture that compute points which are uniformly scattered in the domain $\mathbb{W}$, as defined by (8). They adopted the transformation method for the computation of such uniform design. This approach requires a set of vectors $U = \{\mathbf{u}_i = [u_{i1}, ..., u_{i(k-1)}]^T, i = 1, ..., n\} \subset [0,1]^{k-1}$ with small discrepancy.

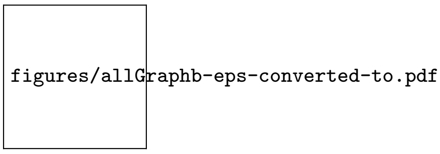


Fig. 4: Computed weight vectors by three different methods, for three and four dimensions. On the left hand side, a set computed by the simplex-lattice method is shown. In the middle, we show the set generated by the combination of Hammersley's method and uniform design. Finally, on the right hand side, we show a set of weight vectors obtained by Monte Carlo sampling.

Here, we propose to make use of Hammersley's method to obtain U and then we apply the following transformation:

$$\begin{aligned} w_{ti} &= (1 - u_{ti}^{\frac{1}{k-i}}) \prod_{j=1}^{i-1} u_{tj}^{\frac{1}{k-j}}, \quad i = 1, ..., k-1, \\ w_{tk} &= \prod_{j=1}^{k-1} u_{tj}^{\frac{1}{k-j}}, \quad t = 1, ..., n. \end{aligned} \tag{13}$$

Then, $\{\mathbf{w}_t = [w_{ti}, \ldots, w_{tk}]^T,\ t = 1, \ldots, n\}$ is a uniform design on $\mathbb{W}$. The pseudocode of the approach used to generate weight vectors is presented in Algorithm 1.

In Figure 4, we show different sets of weight vectors generated by three different methods: simplex-lattice, our pproposed approach based on Uniform Design (described in Algorithm 1) and Monte Carlo sampling. As can be seen from the figure, the method based on uniform design produces less points at the boundary of the domain than the simplex-lattice method while maintaining a good distribution. Also, the repetition rate of each component of the weight vectors is much lower. When using Monte Carlo sampling method (we adopted the approach described in [35]), the generated vectors tend to concentrate in the center of the domain and show a bad distribution. From these figures, it should be clear that our proposed approach allows to compute a very well distributed set of weight vectors and at the same time avoids the problems (described before) of the simplex-lattice method.

4.3 Description of our proposed approach

Making use of the transformation of an MOP into an LAP, we propose a new kind of MOEA that uses an evolutionary model to select solutions based on an LAP selection method. The description of our proposed *MOEA based on linear assignment problem selection* (MOEA-LAPS) is given next.

MOEA-LAPS starts with an initial population P_0 of n individuals and performs a random initialization of each solution within the allowable ranges of their decision variables and then evaluates all individuals on each of the objective functions of the MOP. Thereafter, at each generation g, we create a new offspring population P'_g of n individuals from the actual population P_g. Having the parents and offspring populations, we form a set $Q_g = P_g \cup P'_g$ of $2n$ posible solutions to the MOP. Then, an LAP is created from the MOP using the k-dimensional objective vectors from Q_g (as explained in Section 4.1) and n weight vectors uniformly spread in objective function space are computed as described in Algorithm 1. With this, a cost can be incurred for each individual representing some vector in the Pareto Optimal Front approximation. As shown in Section 4.1, the solution to an LAP can be seen as a square matrix. In our case, rows are the weight vectors and columns represent the individuals in our population, such that a selection i, j means individual j is assigned to the weight vector i.

Therefore, a selection procedure based on the LAP, obtained from the MOP transformation, can be performed in order to select n individuals from Q_g which are closer to the weight vectors and with this, we can obtain a solution to the created LAP. These individuals will then form the population P_{g+1} that constitute the next generation. In other words, the goal will be to describe all regions covered by the $2n$ vectors using only n individuals in such a way that the total cost of the assignment is minimized. This process is repeated until a stopping criterion is fulfilled. The whole procedure is summarized in Algorithm 2.[1]

Algorithm 2 MOEA-LAPS

Require: MOP, population size (n), maximum number of generations (g_{max}), parameters C_r and F for DE/$rand$/1/bin
Ensure: $P_{g_{max}}$ (approximation of the $\mathcal{P}^*$ and $\mathcal{PF}^*$)
1: Generate initial population P_1 randomly
2: Evaluate each individual in P_1
3: $W \leftarrow$ Generate n weight vectors using Algorithm 1
4: **for** $g = 1$ **to** g_{max} **do**
5: $P^*_g \leftarrow$ Generate offspring P'_g from P_g using the operators of the MOEA
6: Evaluate each individual in P'_g
7: $Q_g \leftarrow P_g \cup P'_g$
8: Calculate $\mathbf{z}^{max}$ and $\mathbf{z}^{min}$ by (6)
9: Normalize objectives of each individual in Q_g by (7)
10: Generate the cost matrix C by (9) using Q_g and W
11: $\mathcal{I} \leftarrow$ Obtain the best assignment in C using the Hungarian Method
12: $P_{g+1} \leftarrow \{\mathbf{x}_i \mid i \in \mathcal{I}, \mathbf{x}_i \in Q_g\}$
13: **end for**

For the construction of the cost matrix C, described in (9), we use here a modified version of the Tchebycheff decomposition approach introduced in [13]. This approach is different from the one adopted in the preliminary version of this approach, which was presented in [4], in which the original Tchebycheff

[1] The source code of our proposed approach is available for download at: `https://www.cs.cinvestav.mx/~EVOCINV/software/LAP/LAP.html`

decomposition approach [41] was adopted. The reason for this change is that we realized that the weight vectors of a subproblem and the direction of its optimal solution coincide with this modified version, whereas in the case in which the original Tchebycheff decomposition approach is adopted this relation is non-linear. Therefore, when adopting the modified Tchebycheff decomposition it is posible to control in a direct manner the distribution of solutions in objective function space. Next, we present the re-formulation for the computation of the cost matrix, used for the LAP transformation applied in this work:

The modified cost C_{rj} of assigning the individual $\mathbf{x}_j$ to the weight vector $\mathbf{w}_r$ is given by:

$$C_{rj} = \max_{i=1,\dots,k} \frac{\tilde{f}_i(\mathbf{x}_j)}{w_{ri}},\ r = 1, ..., m,\ j = 1, ..., n. \tag{14}$$

The values of the cost matrix C then denote how suitable a solution is for representing each region of the Pareto Optimal Front.

5 Experimental Results

We validated our proposed MOEA-LAPS using the two versions of the Tchebycheff decomposition: the original version (this version is called HDE, and it was previously introduced in [4]) and the modified version introduced in [13]. Additionally, we compare the performance of our proposed approach with respect to that of three MOEAs that are representative of the state-of-the-art in many-objective optimization: the CMA-PAES-HAGA,[2] the θ-DEA[3] [40] and the NSGA-III[4] [13].

5.1 Parameterization

In our experiments, we adopted 16 MOPs: seven problems having from two to nine objective functions taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [14] and nine problems taken from the Walking-Fish-Group (WFG) test suite [19]. In the DTLZ test problems, the total number of variables is equal to $n = m+k$, where $m = 2, ..., 9$ is the number of objectives and k was set to 10 for DTLZ1-6 and 20 for DTLZ7. For the case of the WFG test problems the total number of variables is given by $n = k + l$, where $k = 2(m - 1)$ (for $m = 3, ..., 9$) and $l = 20$. In instances of 2 objective functions k was set to 4.

In order to assess the performance of each MOEA, we adopted the hypervolume indicator. The hypervolume is defined as the volume of the hypercube

[2] The source code of CMA-PAES-HAGA was provided to us by Shahin Rostamin in Python, but it is also available at: `https://github.com/shahinrostami/CMA-HAGA-release`

[3] The source code of θ-DEA was provided to us by Yuan Yuan in Java and is available at: `http://www.cs.bham.ac.uk/~xin/papers/TEVC2016FebManyEAs.zip`

[4] We used the version of NSGA-III that was included in the source code provided to us by Yuan Yuan, and which is also available at: `http://www.cs.bham.ac.uk/~xin/papers/TEVC2016FebManyEAs.zip`

formed by the space covered by a set of possible solutions, capturing both convergence and maximum spread in a single unary value [42]. In order to compute the hypervolume indicator, we used the following set of reference points: $\mathbf{y}_{ref} = [y_1, \cdots, y_m]$ such that: $y_i = 1.1$ for DTLZ1, DTLZ2 and DTLZ4; $y_i = 3$ for DTLZ3, DTLZ5 and DTLZ6; and $y_i = 7$ for DTLZ7. Finally, for WFG1-9 $y_i = (i+1) * 2 + 0.1$. In our experiments, each MOEA was executed 30 times for each problem instance and we measured the average hypervolume values over these independent runs.

Both of our proposed algorithms (HDE and MOEA-LAPS) were tested using differential evolution as their search engine. The parameters adopted in our experiments were: $F = 1.0$ and $Cr = 0.4$, for both cases. The recombination operators for θ-DEA and NSGA-III were simulated binary crossover and polynomial-based mutation as proposed by their authors. Their corresponding parameters were also set as indicated by their authors: the crossover probability was set to $p_c = 1$, and for the mutation probability we adopted a value of $p_m = 1$. The distribution indexes for all cases were set in the following manner: $\eta_c = 30$ and $\eta_m = 20$. CMA-PAES-HAGA was run with an archive capacity equal to the population size and the number of grid divisions was set to 2, as suggested by its author. The algorithms MOEA-LAPS, HDE and CMA-PAES-HAGA can use an arbitrary population size, but in NSGA-III and θ-DEA the population size increases nonlinearly with the number of objectives. Hence, we used different population sizes for each number of objectives. For the DTLZ problems with 2, 3, 4 and 8 objectives, the population size was set to 120. For the DTLZ problems with 5 and 6 objectives, the population size was set to 126. Finally, for problems having 7 and 9 objectives, the population size was set to 210 and 165, respectively.[5] The maximum number of generations used for each MOEA when solving each of the adopted test problems was set to 300.

5.2 Discussion of Results

In Tables 1, 2 and 3, we show the average hypervolume values over the DTLZ and WFG test problems, as well as the results of the statistical analysis of results that we performed (we adopted Wilcoxon's rank sum). The cells containing the best hypervolume value for each problem have the darkest grey colored background and the second best values have a lighter grey color.

In the case of the DTLZ test problems, the two versions of our proposed approach had the best overall results in most cases. For the case of DTLZ1, MOEA-LAPS outperformed the other MOEAs in the 5 problem instances with more objectives (i.e., those having 5, 6, 7, 8 and 9 objectives), HDE outperformed the other MOEAs in the 4 objectives instance, while θ-DEA and

[5] This apparent inconsistency in the population sizes for 7 and 9 objectives arises due to the procedure adopted to compute the number of weight vectors when using the simplex-lattice method.

MOP	MOEA-LAPS	HDE	MOEA-LAPS VS HDE	CMA-PAES-HAGA	MOEA-LAPS VS CMA-P-H	θ-DEA	MOEA-LAPS VS θ-DEA	NSGA-III	MOEA-LAPS VS NSGA-III
	HV	HV	P(H)	HV	P(H)	HV	P(H)	HV	P(H)
DTLZ1 (2)	1.01159	1.04502	0.882464 (0)	0.00000	0.000000 (1)	1.03486	0.673495 (0)	1.06305	0.157976 (0)
DTLZ1 (3)	1.26674	1.21314	0.662735 (0)	0.00000	0.000000 (1)	1.27909	0.673495 (0)	1.19283	0.641424 (0)
DTLZ1 (4)	1.41894	1.44326	0.853382 (0)	0.00000	0.000000 (1)	1.39996	0.491783 (0)	1.06033	0.559231 (0)
DTLZ1 (5)	1.59060	1.58800	0.529782 (0)	0.00000	0.000000 (1)	1.58838	0.043584 (1)	0.94777	0.176128 (0)
DTLZ1 (6)	1.75922	1.74372	0.318304 (0)	0.00000	0.000000 (1)	1.70507	0.057460 (0)	0.96371	0.067869 (0)
DTLZ1 (7)	1.94859	1.94858	0.428886 (0)	0.00000	0.000000 (1)	1.94711	0.717185 (0)	1.64819	0.761825 (0)
DTLZ1 (8)	2.09028	1.98784	0.876628 (0)	0.00000	0.000000 (1)	1.98406	0.549321 (0)	0.77253	0.946955 (0)
DTLZ1 (9)	2.35607	2.33828	0.609953 (0)	0.00000	0.000000 (1)	2.32767	0.958730 (0)	1.33927	0.589441 (0)
DTLZ2 (2)	0.42079	0.42058	0.180887 (0)	0.40461	0.784457 (0)	0.42087	0.569209 (0)	0.42069	0.496440 (0)
DTLZ2 (3)	0.74769	0.73614	0.455297 (0)	0.68579	0.610008 (0)	0.75275	0.387100 (0)	0.75216	0.641424 (0)
DTLZ2 (4)	1.01577	0.98404	0.569220 (0)	0.75131	0.057460 (0)	1.03056	0.652044 (0)	1.02880	0.888303 (0)
DTLZ2 (5)	1.25739	1.22274	0.899995 (0)	0.63271	0.630876 (0)	1.27706	0.510598 (0)	1.27139	0.982307 (0)
DTLZ2 (6)	1.47814	1.44551	0.510598 (0)	0.33952	0.958731 (0)	1.50921	0.549327 (0)	1.49983	0.935192 (0)
DTLZ2 (7)	1.74685	1.72095	0.428963 (0)	0.41808	0.589451 (0)	1.76975	0.379036 (0)	1.75740	0.222573 (0)
DTLZ2 (8)	1.92810	1.90181	0.599689 (0)	0.03981	0.055546 (0)	1.96316	0.876635 (0)	1.94581	0.911709 (0)
DTLZ2 (9)	2.19753	2.17122	0.739399 (0)	0.03519	0.970516 (0)	2.23017	0.673495 (0)	2.21346	0.728265 (0)
DTLZ3 (2)	8.03130	7.52212	0.673495 (0)	0.00000	0.000000 (1)	8.00869	0.559231 (0)	7.78823	0.403538 (0)
DTLZ3 (3)	24.72025	24.49285	0.923442 (0)	0.00000	0.000000 (1)	25.88856	0.297272 (0)	21.34563	0.673495 (0)
DTLZ3 (4)	78.70207	77.85389	0.970516 (0)	0.00000	0.000000 (1)	77.48017	0.982307 (0)	28.91196	0.579289 (0)
DTLZ3 (5)	239.97094	237.13918	0.641424 (0)	0.00000	0.000000 (1)	231.89920	0.784460 (0)	92.54490	0.970516 (0)
DTLZ3 (6)	721.63279	715.44073	0.673495 (0)	0.00000	0.000000 (1)	671.48191	0.149449 (0)	248.90446	0.807275 (0)
DTLZ3 (7)	2186.78203	2185.52393	0.589436 (0)	0.00000	0.000000 (1)	2185.65366	0.689749 (0)	1645.51603	0.579284 (0)
DTLZ3 (8)	6211.65972	6041.12319	0.206205 (0)	0.00000	0.000000 (1)	4635.91566	0.673495 (0)	2270.41537	0.473347 (0)
DTLZ3 (9)	19682.81309	19666.14228	0.684310 (0)	0.00000	0.000000 (1)	19015.89337	0.899992 (0)	10593.79222	0.958730 (0)
DTLZ4 (2)	0.40652	0.41606	0.830255 (0)	0.37113	0.318304 (0)	0.30690	0.347828 (0)	0.37938	0.750587 (0)
DTLZ4 (3)	0.73952	0.73301	0.491783 (0)	0.63423	0.549327 (0)	0.75271	0.876635 (0)	0.74252	0.982307 (0)
DTLZ4 (4)	1.01649	0.98559	0.420386 (0)	0.84323	0.589451 (0)	1.03137	0.157970 (0)	1.02976	0.297272 (0)
DTLZ4 (5)	1.26512	1.22955	0.610008 (0)	1.02312	0.539510 (0)	1.27876	0.437641 (0)	1.27588	0.549327 (0)
DTLZ4 (6)	1.49039	1.45193	0.283778 (0)	1.16056	0.157976 (0)	1.50728	0.304177 (0)	1.50734	0.200949 (0)
DTLZ4 (7)	1.76265	1.72997	0.166866 (0)	1.32700	0.652044 (0)	1.77273	0.311188 (0)	1.76794	0.028129 (1)
DTLZ4 (8)	1.94583	1.90657	0.318304 (0)	1.41296	0.818746 (0)	1.96768	0.520145 (0)	1.96198	0.876635 (0)
DTLZ4 (9)	2.21702	2.17928	0.773120 (0)	1.59093	0.970516 (0)	2.23470	0.387100 (0)	2.23101	0.784460 (0)
DTLZ5 (2)	8.21079	8.21058	0.818746 (0)	8.19477	0.037782 (1)	8.21088	0.347828 (0)	8.21076	0.946956 (0)
DTLZ5 (3)	23.90173	23.97910	0.630872 (0)	23.89903	0.482517 (0)	23.89727	0.200949 (0)	23.95619	0.970516 (0)
DTLZ5 (4)	71.59319	71.52693	0.264326 (0)	71.06347	0.599689 (0)	70.88288	0.033874 (1)	71.29680	0.717189 (0)
DTLZ5 (5)	214.17456	213.83002	0.807275 (0)	211.20073	0.750587 (0)	208.34231	0.529782 (0)	207.05607	0.411911 (0)
DTLZ5 (6)	640.55372	640.30482	0.923442 (0)	622.40414	0.841801 (0)	627.32861	0.706171 (0)	606.17420	0.684323 (0)
DTLZ5 (7)	1927.24180	1926.74890	0.411911 (0)	1876.05123	0.549327 (0)	1890.75305	0.501144 (0)	1816.64305	0.684323 (0)
DTLZ5 (8)	5733.34679	5700.48482	0.728265 (0)	5337.63782	0.070127 (0)	5560.72154	0.347828 (0)	5338.03114	0.355472 (0)
DTLZ5 (9)	17234.49677	17197.45493	0.994102 (0)	16261.37885	0.935192 (0)	16714.46090	0.935192 (0)	16003.08973	0.233989 (0)

Table 1: Average of the hypervolume indicator (HV) values over 30 independent runs of the results obtained for DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ5. The cells containing the best hypervolume value for each problem have the darkest grey colored background and those having the second best values have a lighter grey colored background. The P(H) columns show the results of the Wilcoxon's rank sum test. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level.

MOP	MOEA-LAPS	HDE	MOEA-LAPS VS HDE	CMA-PAES-HAGA	MOEA-LAPS VS CMA-P-H	θ-DEA	MOEA-LAPS VS θ-DEA	NSGA-III	MOEA-LAPS VS NSGA-III
	HV	HV	P(H)	HV	P(H)	HV	P(H)	HV	P(H)
DTLZ6 (2)	8.21084	8.21084	0.795834 (0)	8.15458	0.347814 (0)	8.19875	0.491765 (0)	8.19954	0.684314 (0)
DTLZ6 (3)	23.89275	23.98192	0.067861 (0)	23.71328	0.641419 (0)	23.00868	0.157970 (0)	22.60842	0.387093 (0)
DTLZ6 (4)	71.39857	71.42395	0.935192 (0)	69.73852	0.529782 (0)	61.84732	0.428963 (0)	37.71466	0.620404 (0)
DTLZ6 (5)	213.22287	213.57169	0.233989 (0)	202.51484	0.739399 (0)	164.01720	0.630876 (0)	0.74634	0.395267 (0)
DTLZ6 (6)	636.84244	638.55173	0.211561 (0)	587.52549	0.717189 (0)	396.11959	0.673495 (0)	0.00455	0.000018 (1)
DTLZ6 (7)	1920.93461	1919.85744	0.750587 (0)	1784.30300	0.379036 (0)	1227.34390	0.290472 (0)	0.00000	0.000000 (1)
DTLZ6 (8)	5704.01827	5653.68394	0.464273 (0)	4876.59477	0.129670 (0)	2451.64389	0.569220 (0)	0.00000	0.000000 (1)
DTLZ6 (9)	17176.14316	17082.02787	0.728265 (0)	15123.65205	0.190730 (0)	8040.41026	0.630876 (0)	0.00000	0.000000 (1)
DTLZ7 (2)	31.87442	31.88078	0.122353 (0)	31.75919	0.001680 (1)	31.88310	0.029205 (1)	31.61827	0.015014 (1)
DTLZ7 (3)	199.98791	200.51900	0.662735 (0)	198.47767	0.105470 (0)	197.97888	0.706171 (0)	198.00384	0.876635 (0)
DTLZ7 (4)	1227.60858	1226.52659	0.620404 (0)	1162.67003	0.684323 (0)	1106.33134	0.501144 (0)	1127.26332	0.355472 (0)
DTLZ7 (5)	7295.27013	7249.32546	0.176128 (0)	5203.58768	0.589451 (0)	5698.00680	0.970516 (0)	5737.81358	0.784460 (0)
DTLZ7 (6)	38082.13494	39329.46289	0.641424 (0)	5130.42842	0.491783 (0)	31505.32971	0.830255 (0)	30452.35714	0.491783 (0)
DTLZ7 (7)	222059.90560	202192.03310	0.211561 (0)	4215.22368	0.076927 (0)	112393.14100	0.750587 (0)	56780.82658	0.283778 (0)
DTLZ7 (8)	838454.34270	698779.80150	0.520145 (0)	0.00000	0.000000 (1)	84226.46624	0.023083 (1)	12345.94719	0.000108 (1)
DTLZ7 (9)	3668251.79400	2936863.31200	0.520145 (0)	0.00000	0.000000 (1)	43577.00192	0.000000 (1)	0.00000	0.000000 (1)
WFG1 (2)	18.35	18.38	0.332855 (0)	13.29	0.695215 (0)	14.80	0.899995 (0)	14.92	0.773120 (0)
WFG1 (3)	125.97	127.17	0.153667 (0)	111.24	0.728265 (0)	129.57	0.023243 (1)	122.18	0.006972 (1)
WFG1 (4)	1132.14	1156.80	0.491783 (0)	1055.56	0.428963 (0)	1162.23	0.946956 (0)	1021.10	0.437641 (0)
WFG1 (5)	12625.66	12761.71	0.133454 (0)	11843.69	0.185767 (0)	14217.05	0.970516 (0)	11888.60	0.067869 (0)
WFG1 (6)	183039.47	174176.50	0.464273 (0)	155910.28	0.195791 (0)	205992.10	0.946956 (0)	174684.03	0.589451 (0)
WFG1 (7)	2821385.05	2688076.15	0.379036 (0)	2371710.22	0.876635 (0)	3744642.17	0.970516 (0)	3044182.67	0.325527 (0)
WFG1 (8)	61733524.77	49607048.07	0.673495 (0)	40518423.49	0.970516 (0)	66168464.63	0.728265 (0)	56516803.75	0.888303 (0)
WFG1 (9)	1220027849.00	1037691188.00	0.206205 (0)	776637900.80	0.853382 (0)	1457182486.00	0.864994 (0)	1208287911.00	0.630876 (0)
WFG2 (2)	19.55	19.52	0.185767 (0)	20.00	0.510598 (0)	19.66	0.072446 (0)	19.65	0.180900 (0)
WFG2 (3)	185.84	189.10	0.008684 (1)	185.00	0.133454 (0)	187.60	0.096263 (0)	185.75	0.102326 (0)
WFG2 (4)	1915.92	1932.30	0.841801 (0)	1869.82	0.864994 (0)	1874.56	0.311188 (0)	1885.36	0.520145 (0)
WFG2 (5)	23345.59	23255.50	0.040595 (1)	22482.43	0.437641 (0)	23012.51	0.491783 (0)	22605.44	0.200949 (0)
WFG2 (6)	324548.64	327186.99	0.807275 (0)	312713.45	0.994102 (0)	315086.23	0.818746 (0)	315826.02	0.864994 (0)
WFG2 (7)	5355656.85	5469402.79	0.795846 (0)	5054351.83	0.251881 (0)	5261422.97	0.371077 (0)	5282287.91	0.750587 (0)
WFG2 (8)	84927905.18	82238217.15	0.379036 (0)	87635827.07	0.129670 (0)	85802943.32	0.125970 (0)	91515158.83	0.245814 (0)
WFG2 (9)	1852591909.00	1752196504.00	0.935192 (0)	1774375047.00	0.589451 (0)	1750335798.00	0.428963 (0)	1798756609.00	0.501144 (0)
WFG3 (2)	20.71	20.67	0.569220 (0)	19.68	0.033874 (1)	20.80	0.347828 (0)	20.80	0.063533 (0)
WFG3 (3)	162.31	162.48	0.016955 (1)	154.73	0.029205 (1)	158.96	0.264326 (0)	158.86	0.035137 (1)
WFG3 (4)	1587.41	1582.30	0.641424 (0)	1475.04	0.446419 (0)	1509.33	0.673495 (0)	1495.88	0.510598 (0)
WFG3 (5)	18486.29	18401.63	0.673495 (0)	16775.29	0.137323 (0)	17531.89	0.501144 (0)	17392.56	0.420386 (0)
WFG3 (6)	251343.64	246296.73	0.190730 (0)	221388.02	0.807275 (0)	177089.94	0.118817 (0)	206066.22	0.761828 (0)
WFG3 (7)	4044693.78	3961521.01	0.994102 (0)	3530282.23	0.340288 (0)	3153200.44	0.166866 (0)	3635567.49	0.750587 (0)
WFG3 (8)	69476315.15	68491749.65	0.180900 (0)	55628293.12	0.153667 (0)	59317666.90	0.162375 (0)	63845403.50	0.057460 (0)
WFG3 (9)	1410422276.00	1361401462.00	0.270705 (0)	1136577695.00	0.549327 (0)	1253569462.00	0.145319 (0)	1288134959.00	0.108690 (0)

Table 2: Average of the hypervolume indicator (HV) values over 30 independent runs of the results obtained for DTLZ6 and DTLZ7 as well as for WFG1, WFG2 and WFG3. The cells containing the best hypervolume value for each problem have the darkest grey colored background and those having the second best values have a lighter grey colored background. The P(H) columns show the results of the Wilcoxon's rank sum test. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level.

MOP	MOEA-LAPS	HDE	MOEA-LAPS VS HDE	CMA-PAES-HAGA	MOEA-LAPS VS CMA-P-H	θ-DEA	MOEA-LAPS VS θ-DEA	NSGA-III	MOEA-LAPS VS NSGA-III
	HV	HV	P(H)	HV	P(H)	HV	P(H)	HV	P(H)
WFG4 (2)	18.61	18.61	0.270705 (0)	17.53	0.079782 (0)	18.50	0.026077 (1)	18.46	0.036439 (1)
WFG4 (3)	172.15	171.56	0.008684 (1)	162.71	0.024157 (1)	171.48	0.501144 (0)	171.16	0.096263 (0)
WFG4 (4)	1812.55	1810.88	0.277189 (0)	1663.82	0.332855 (0)	1798.54	0.739399 (0)	1792.84	0.082357 (0)
WFG4 (5)	22304.32	22372.22	0.251881 (0)	19678.19	0.180900 (0)	22002.72	0.157976 (0)	21845.96	0.070127 (0)
WFG4 (6)	311409.89	312114.64	0.818746 (0)	261298.79	0.539510 (0)	310682.55	0.379036 (0)	307753.47	0.853382 (0)
WFG4 (7)	4849220.06	5036595.10	0.332855 (0)	4176822.05	0.122353 (0)	5028716.36	0.539510 (0)	4989699.04	0.070127 (0)
WFG4 (8)	67827861.03	74571614.52	0.970516 (0)	57269619.97	0.899995 (0)	89365610.42	0.641424 (0)	88070487.89	0.728265 (0)
WFG4 (9)	1490059240.00	1460381516.00	0.017649 (1)	888274329.00	0.030317 (1)	1789454139.00	0.051877 (0)	1772121350.00	0.093341 (0)
WFG5 (2)	17.81	17.83	0.411911 (0)	16.32	0.761828 (0)	17.82	0.559231 (0)	17.82	0.630876 (0)
WFG5 (3)	165.63	165.85	0.129670 (0)	149.25	0.641424 (0)	165.98	0.082357 (0)	165.91	0.200949 (0)
WFG5 (4)	1744.10	1746.86	0.363222 (0)	1434.47	0.970516 (0)	1760.49	0.387100 (0)	1754.52	0.529782 (0)
WFG5 (5)	21364.47	21629.60	0.420386 (0)	15968.99	0.899995 (0)	21701.04	0.395267 (0)	21607.39	0.340288 (0)
WFG5 (6)	300926.78	306229.31	0.024157 (1)	200851.04	0.297272 (0)	307390.08	0.304177 (0)	305457.90	0.245814 (0)
WFG5 (7)	4957029.76	4992761.00	0.795846 (0)	3180936.39	0.706171 (0)	4980823.10	0.739399 (0)	4951696.05	0.297272 (0)
WFG5 (8)	76330933.17	77062841.19	0.099258 (0)	38848297.14	0.228230 (0)	89036040.54	0.166866 (0)	88425576.77	0.059428 (0)
WFG5 (9)	1354197018.00	1360070224.00	0.099258 (0)	707338447.30	0.036439 (1)	1792180683.00	0.118817 (0)	1780032315.00	0.042067 (1)
WFG6 (2)	18.21	18.20	0.923442 (0)	17.13	0.706171 (0)	18.09	0.233989 (0)	18.04	0.217017 (0)
WFG6 (3)	168.37	167.95	0.325527 (0)	156.84	0.630876 (0)	167.55	0.579294 (0)	167.27	0.411911 (0)
WFG6 (4)	1778.51	1775.42	0.784460 (0)	1546.16	0.599689 (0)	1772.91	0.579294 (0)	1767.64	0.569220 (0)
WFG6 (5)	21894.75	21976.52	0.006377 (1)	17899.58	0.016285 (1)	21825.03	0.008684 (1)	21779.96	0.004427 (1)
WFG6 (6)	310912.63	312374.76	0.888303 (0)	230278.01	0.662735 (0)	311338.18	0.923442 (0)	309092.44	0.923442 (0)
WFG6 (7)	5058122.27	5108677.48	0.473347 (0)	3599779.16	0.982307 (0)	5048177.76	0.739399 (0)	5028125.40	0.245814 (0)
WFG6 (8)	88978531.59	89975711.95	0.171450 (0)	48481852.57	0.118817 (0)	90948339.61	0.437641 (0)	90372629.74	0.133454 (0)
WFG6 (9)	1778639482.00	1752029738.00	0.180900 (0)	787399964.00	0.325527 (0)	1827359690.00	0.446419 (0)	1822150812.00	0.395267 (0)
WFG7 (2)	18.65	18.64	0.053685 (0)	17.61	0.958731 (0)	18.58	0.017649 (1)	18.57	0.911709 (0)
WFG7 (3)	173.59	172.55	0.176128 (0)	161.73	0.684323 (0)	172.79	0.403538 (0)	172.49	0.245814 (0)
WFG7 (4)	1840.78	1830.46	0.283778 (0)	1608.45	0.222573 (0)	1828.92	0.290472 (0)	1817.15	0.311188 (0)
WFG7 (5)	22729.68	22774.11	0.039167 (1)	18689.33	0.264326 (0)	22480.21	0.251881 (0)	22299.21	0.195791 (0)
WFG7 (6)	321894.84	325114.62	0.784460 (0)	239774.30	0.750587 (0)	320401.61	0.347828 (0)	317866.83	0.464273 (0)
WFG7 (7)	5306175.67	5341618.32	0.830255 (0)	3779872.91	0.761828 (0)	5225332.72	0.559231 (0)	5171966.65	0.371077 (0)
WFG7 (8)	81306355.78	85253409.17	0.807275 (0)	49687980.12	0.728265 (0)	92655657.14	0.630876 (0)	91432208.31	0.133454 (0)
WFG7 (9)	1507705098.00	1666071515.00	0.087710 (0)	759688927.80	0.589451 (0)	1863429164.00	0.048413 (1)	1843212523.00	0.030317 (1)
WFG8 (2)	17.08	17.07	0.923442 (0)	16.15	0.185767 (0)	16.80	0.027086 (1)	16.84	0.355472 (0)
WFG8 (3)	163.22	162.27	0.133454 (0)	151.15	0.251881 (0)	162.07	0.318304 (0)	161.50	0.662735 (0)
WFG8 (4)	1692.88	1685.99	0.420386 (0)	1467.87	0.072446 (0)	1683.09	0.411911 (0)	1676.51	0.510598 (0)
WFG8 (5)	20508.18	20489.23	0.200949 (0)	16437.22	0.055546 (0)	20331.70	0.096263 (0)	20173.57	0.021506 (1)
WFG8 (6)	278331.25	276675.08	0.935192 (0)	206235.47	0.994102 (0)	283047.07	0.773120 (0)	278837.63	0.559231 (0)
WFG8 (7)	4624457.40	4609669.43	0.958731 (0)	3250013.38	0.599689 (0)	4541494.21	0.347828 (0)	4497319.16	0.784460 (0)
WFG8 (8)	76545545.83	76950487.02	0.077272 (0)	37290332.18	0.853382 (0)	79383636.13	0.529782 (0)	77823587.80	0.141278 (0)
WFG8 (9)	1577098100.00	1542265099.00	0.437641 (0)	485458368.40	0.162375 (0)	1607435528.00	0.190730 (0)	1573161821.00	0.085000 (0)
WFG9 (2)	17.55	17.43	0.082357 (0)	17.20	0.028129 (1)	17.54	0.005322 (1)	17.61	0.001302 (1)
WFG9 (3)	159.17	158.67	0.501144 (0)	158.28	0.529782 (0)	160.05	0.464273 (0)	161.93	0.864994 (0)
WFG9 (4)	1650.34	1648.98	0.818746 (0)	1596.56	0.153667 (0)	1666.62	0.239850 (0)	1650.02	0.074827 (0)
WFG9 (5)	20081.14	20205.60	0.695215 (0)	18425.93	0.180900 (0)	20078.75	0.129670 (0)	19827.45	0.129670 (0)
WFG9 (6)	280269.94	282510.61	0.264326 (0)	239065.13	0.145319 (0)	281380.33	0.222573 (0)	277758.49	0.129670 (0)
WFG9 (7)	4560024.58	4565898.74	0.630876 (0)	3795296.27	0.355472 (0)	4549093.29	0.795846 (0)	4525723.68	0.864994 (0)
WFG9 (8)	77402706.19	75445058.55	0.970516 (0)	51441271.88	0.970516 (0)	78901918.77	0.695215 (0)	77825756.11	0.684323 (0)
WFG9 (9)	1502995280.00	1515904110.00	0.096263 (0)	833958409.70	0.007617 (1)	1603598908.00	0.340288 (0)	1570044489.00	0.318304 (0)

Table 3: Average of the hypervolume indicator (HV) values over 30 independent runs of the results obtained for WFG4, WFG5, WFG6, WFG7, WFG8 and WFG9. The cells containing the best hypervolume value for each problem have the darkest grey colored background and those having the second best values have a lighter grey colored background. The P(H) columns show the results of the Wilcoxon's rank sum test. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level.

NSGA-III outperformed the others in the 3 and 2 objectives instances, respectively. CMA-PAES-HAGA could not obtain any solution which dominated the reference point adopted for the hypervolume in this problem. When dealing with DTLZ2, θ-DEA had the best results in all instances. The second place was for NSGA-III, which obtained the second best place for all instances, except for the one with 2 objectives, where MEA-LAPS obtained the second place. HDE obtained very competitive results, but not better than those produced by MOEA-LAPS. Conversely, CMA-PAES-HAGA produced poor values for the hypervolume indicator in all the problem instances that had more than 3 objectives. For the case of DTLZ3, MOEA-LAPS outperformed all the other MOEAs, except for the one with 3 objectives, where θ-DEA outperformed the others. In this case, HDE was the second best approach in almost all instances and θ-DEA obtained very competitive results, but not better than those produced by MOEA-LAPS. NSGA-III did not produce any outstanding results but had a better performance than CMA-PAES-HAGA, which was unable to produce any solution that dominated the reference points that we adopted for computing the hypervolume. Very similar results to those obtained for DTLZ2 were produced in DTLZ4. In this case, θ-DEA outperformed again to all the other MOEAs in most of the problem instances, and obtained the second place in the 6 objectives instance as well as the last place in the 2 objectives instance. It was followed by NSGA-III, which obtained the second place in almost all instances of this problem, with the exception of the 6 objectives instance, where it obtained the fist place and in the 2 objectives instance where this approach obtained the third best results. In the case of DTLZ5, MOEA-LAPS was again able to outperform all the other MOEAs in the instances having more than 3 objectives, followed by HDE, which obtained the second best results in this case. All the other approaches obtained very competitive results in this problem, being θ-DEA one of the most outstanding approaches. For DTLZ6, both MOEA-LAPS and HDE obtained the best results. In this case, HDE was able to outperform MOEA-LAPS in the instances from 2 up to 6 objectives and MOEA-LAPS obtained the best overall results in the instances of more than 6 objectives. NSGA-III was not able to produce any solution which dominated the reference points that we adopted for computing the hypervolume in this problem, in instances of more than 6 objectives. Finally, for DTLZ7, MOEA-LAPS obtained the best overall results, outperforming all the other MOEAs in most of the instances (from 4 to 9 objectives), followed by HDE, which was the second best performer in all the instances, except for the one with 3 objectives where it obtained the first place. θ-DEA was able to outperform all the other MOEAs only in the instance having 2 objectives. CMA-PAES-HAGA was not able to obtain solutions which dominated the reference points that we adopted for instances of more than 7 objectives and NSGA-III had this same behavior in the 9 objectives instance.

For the case of the WFG test suite, when dealing with WFG1, θ-DEA outperformed the other approaches in all the instances of more than 2 objectives. It was followed by HDE in the instances from 3 to 5 objectives and by MOEA-LAPS in the instances of more than 5 objectives. For WFG2, HDE obtained the best results in the instances of 3, 4, 6 and 7 objectives, MOEA-LAPS obtained the best results in the instances of 5 and 9 objectives, NSGA-III obtained the best results in the instance with 8 objectives and CMA-PAES-HAGA obtained the best results in the instance with 2 objectives. In general, all the algorithms showed a very competitive performance in this problem. When dealing with WFG3, MOEA-LAPS had the best performance in instances of more than 3 objectives, followed by HDE, which obtained the second best results, outperforming MOEA-LAPS only in the 3 objectives instances. The best results for the instance of 2 objectives were obtained by θ-DEA, and NSGA-III obtained the second best results in this case. However, in WFG4, MOEA-LAPS had the best performance in the smaller instances (from 2 to 4 objectives), and HDE outperformed the other approaches in the instances with 5, 6 and 7 objectives. θ-DEA obtained the best results for the instances with more objectives (8 and 9), followed by NSGA-III which had very competitive results compared to θ-DEA. In WFG5, θ-DEA was the best performer in almost all the instances, being outperformed by HDE only in the instances with 2 and 7 objectives. NSGA-III obtained very competitive results being the second best in most of the instances of this problem. In WFG6, MOEA-LAPS obtained the best results for the first three instances of the problem, i.e., for 3,4 and 5 objectives. HDE outperformed the others in the instances of 5, 6 and 7 objectives, while θ-DEA obtained the best results in the instances of 8 and 9 objectives. In WFG7, the results were the same as in WFG6. In WFG8, MOEA-LAPS obtained the best results for the instances of 2, 3, 4, 5 and 7 objectives and θ-DEA obtained the best results for the rest of the instances. HDE was the third best performer, obtaining the second best results for all the cases in which MOEA-LAPS won, followed by NSGA-III which obtained competitive results being the second best in the instances of 6 and 8 objectives. Finally, for WFG9, the best performer was θ-DEA, obtaining the best results for instances of 4, 8 and 9 objectives, and it obtained the second place in the 3 objectives instances. The second best results were obtained by HDE, which obtained the best results for the instances of 5, 6 and 7 objectives, followed by NSGA-III, which obtained the first place in the instances of 2 and 3 objectives and the second place in the instances of 8 and 9 objectives.

In general, MOEA-LAPS was the approach that produced the best overall results, obtaining the first place in 49 of the instances, followed by θ-DEA, which obtained the first place in 44 instances. HDE obtained the first place in 29 instances, NSGA-III was the best in 5 instances and CMA-PAES-HAGA was the best in only one instance. Based on the results obtained with the Wilcoxon's rank sum we can say that MOEA-LAPS and HDE had a similar performance, although MOEA-LAPS outperformed HDE. The same can be said when comparing θ-DEA and NSGA-III, since, even though θ-DEA clearly outperformed NSGA-III, they had similar results and both were better than the others in almost all the same instances. For the case of CMA-PAES-HAGA, it could produce acceptable results in most cases, but when dealing with problems of more than 3 objectives it provided the poorest performance

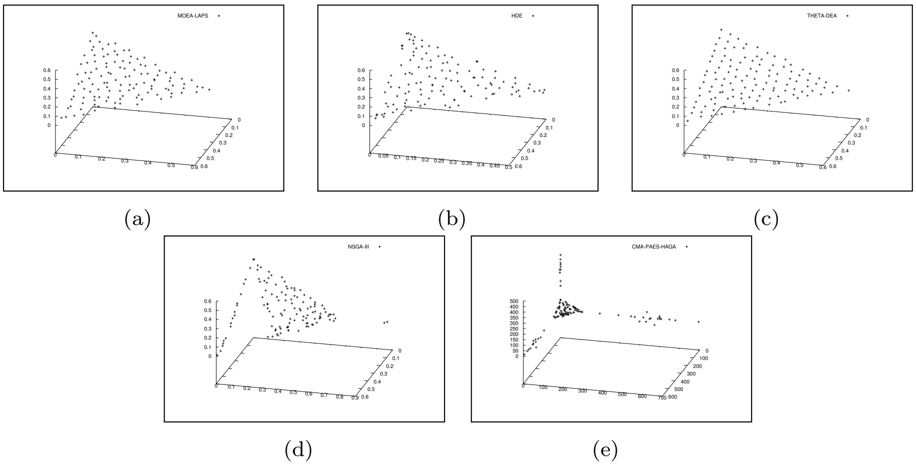


Fig. 5: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for DTLZ1 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.

from all the approaches used in our comparative study. Also, it could not produce any results for DTLZ1 and DTLZ3.

To illustrate the results obtained by all the algorithms adopted in our comparative study, in Figures 5, 6, 7, 8, 9, 10, 11 and 12, we plotted the results obtained for DTLZ1, DTLZ2, DTLZ5, DTLZ7, WFG1, WFG2, WFG3 and WFG5, respectively. These results correspond to the median of the hypervolume value of the 30 independent runs performed for the 3 objectives instances.

As can be seen in the figures, our proposed approach based on uniform design, allowed us to compute a very well-distributed set of weight vectors, which led to a well-spread set of final solutions to the problems. Also, we can observe how the change in the cost matrix computation improves the distribution of the solutions, since in all cases, results from MOEA-LAPS are more evenly distributed than those obtained by HDE. Here, we also can appreciate the tendency of CMA-PAES-HAGA to concentrate the solutions in the central part of the Pareto Optimal Front. Also, it can be observed from the results produced by both θ-DEA and NSGA-III how when adopting the simplex-lattice approach, algorithms tend to generate more solutions at the boundary of the domain, or in the case of DTLZ5 and WFG3 (which are problems with degenerate Pareto Optimal Fronts with a linear shape), the distribution of the solutions is not as uniform and complete as the one obtained by our approach.

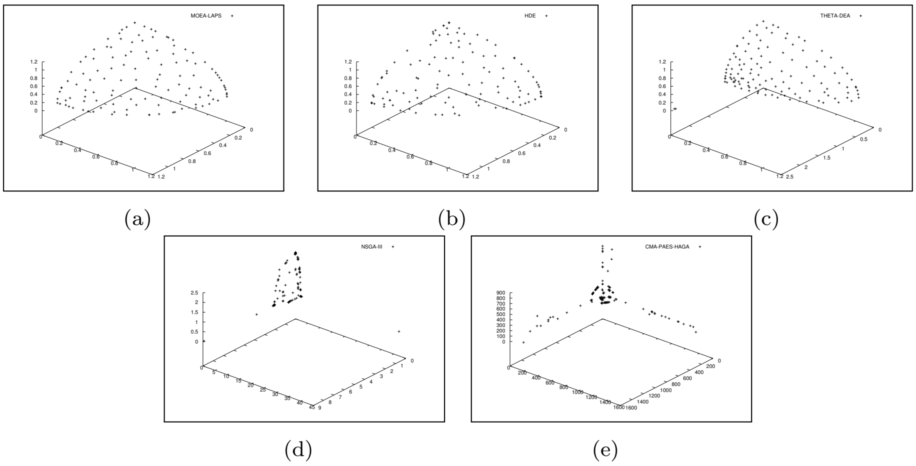


Fig. 6: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for DTLZ3 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.

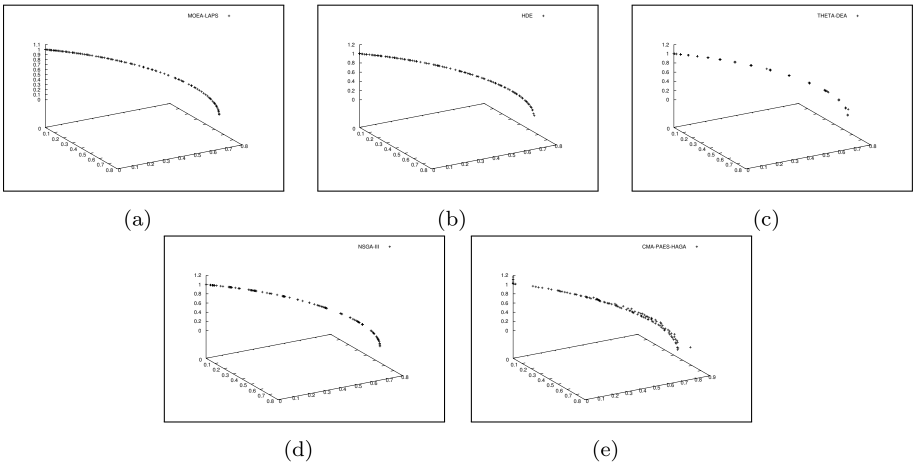


Fig. 7: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for DTLZ5 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.

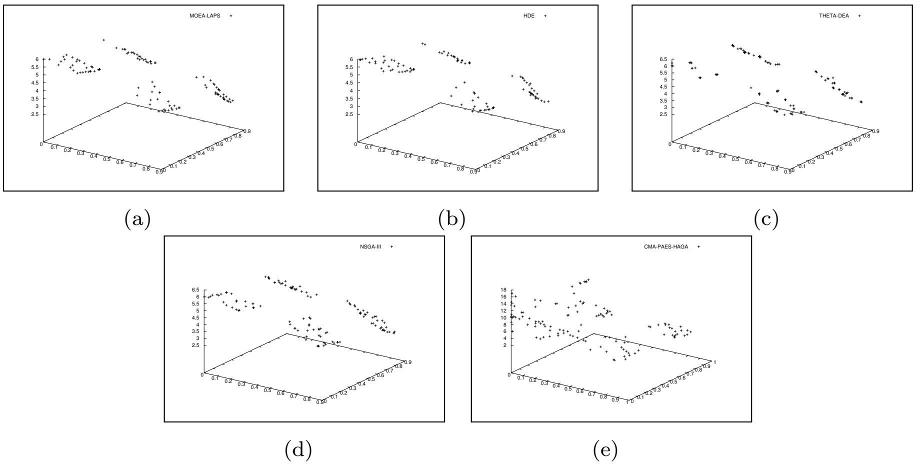


Fig. 8: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for DTLZ7 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.

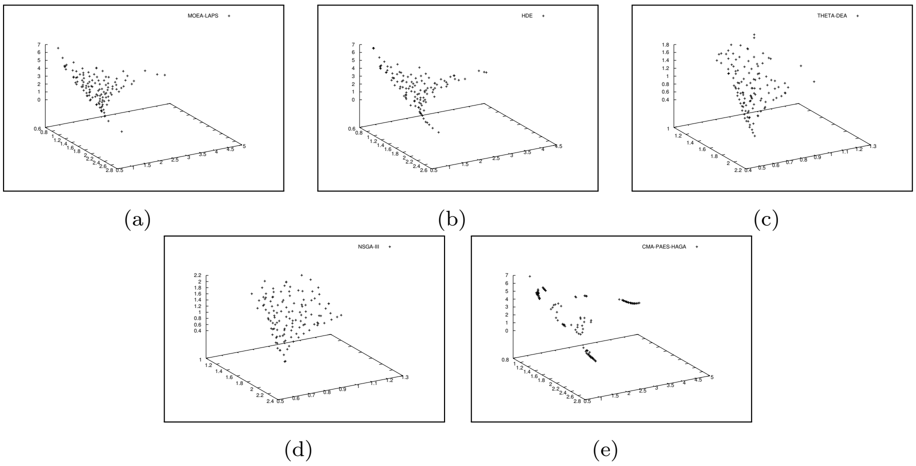
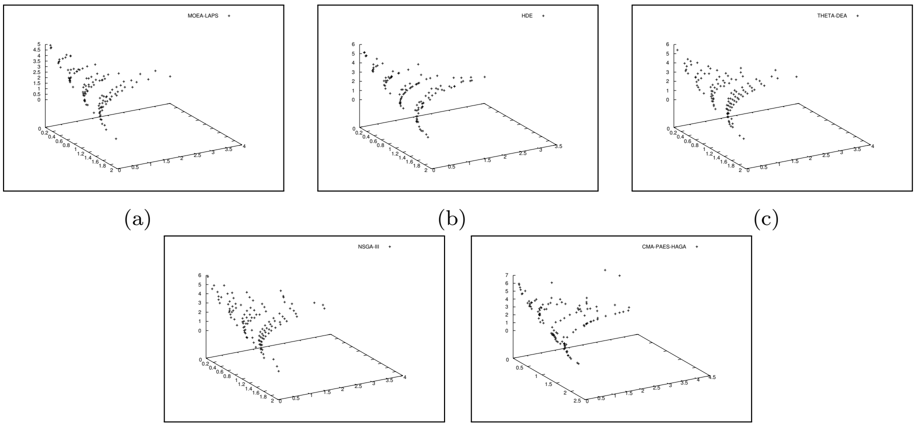
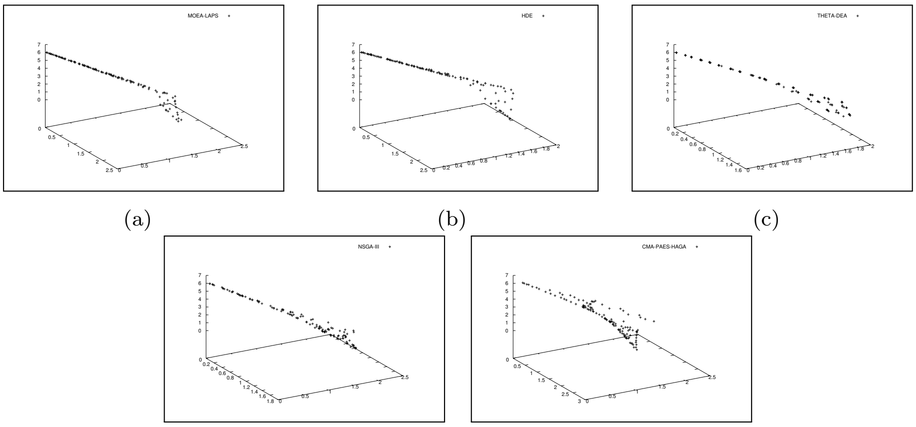


Fig. 9: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for WFG1 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.



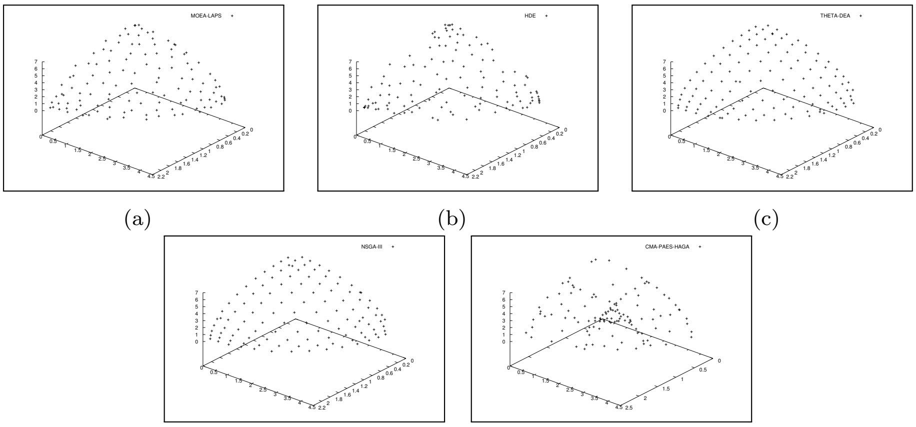
(a) (b) (c) (d) (e)

Fig. 10: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for WFG2 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.



(a) (b) (c) (d) (e)

Fig. 11: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for WFG3 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.



(a) (b) (c)

(d) (e)

Fig. 12: Plots of the approximations obtained by (a) MOEA-LAPS, (b) HDE, (c) θ-DEA, (d) NSGA-III and (e) CMA-PAES-HAGA for WFG5 with 3 objectives. These plots correspond to the median hypervolume value from the 30 independent runs performed.

6 Conclusions and Future Work

This paper has proposed a novel selection mechanism for MOEAs which does not belong to any of the selection techniques that have been used so far in evolutionary multi-objective optimization. Our proposed approach performs a transformation of a multi-objective problem into an LAP using a set of well-distributed points. The resulting LAP is then solved using the Kuhn-Munkres algorithm. Additionally, we have also proposed an algorithm based on the uniform design method to generate a set of weight vectors more uniformly scattered than those obtained by the simplex-lattice method.

Our proposed approach was validated with respect to state-of-the-art MOEAs, being able to outperform them in 49 out of 128 problem instances. In fact, our results showed that our proposed approach was able to provide good results in terms of convergence and distribution of solutions both in problems with a few (two or three) objectives as well as in problems with more than four objectives.

Another important contribution of this work is that we were able to show that the way in which the Tchebycheff decomposition is applied to create a cost matrix for the LAP that we need to solve, is a very relevant factor that contributed to improve the performance of our proposed approach in some problems. When adopting the modified version of the Tchebycheff decomposition, our proposed approach showed the best overall results, which indicates that with this approach it is possible to control in a direct manner, the distribution of solutions in objective function space.

As part of our future work, we intend to study other methods to generate sets of points that are uniformly distributed and which are computationally inexpensive. We also plan to analyze other methods for solving LAPs at a lower computational cost, since the Kuhn-Munkres algorithm has a cubic algorithmic complexity. Furthermore, we would like to improve the performance that our approach has on unimodal problems with concave geometry (e.g., DTLZ4), as well as in those problems with a highly deceptive behavior (e.g., WFG5). Additionally, we would like to apply our approach to domains such as feature selection [1,2].

7 Compliance with Ethical Standards

We hereby submit the paper entitled "Evolutionary Many-objective Optimization based on Linear Assignment Problem Transformations", which is submitted for possible publication in this journal. This is an original contribution and is not being considered for possible publication in any other journal.

References

1. Abualigah, L.M., Khader, A.T.: Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. The Journal of Supercomputing **73**(11), 4773–4795 (2017)
2. Abualigah, L.M., Khader, A.T., Hanandeh, E.S.: A new feature selection method to improve the document clustering using particle swarm optimization algorithm. Journal of Computational Science (2018). (in press)
3. Bader, J., Zitzler, E.: HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation **19**(1), 45–76 (Spring, 2011)
4. Berenguer, J.A.M., Coello Coello, C.A.: Evolutionary Many-Objective Optimization Based on Kuhn-Munkres' Algorithm. In: A. Gaspar-Cunha, C.H. Antunes, C. Coello Coello (eds.) Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015, pp. 3–17. Springer. Lecture Notes in Computer Science Vol. 9019, Guimarães, Portugal (2015)
5. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research **181**(3), 1653–1669 (2007)
6. Bourgeois, F., Lassalle, J.C.: An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. Communications of the ACM **14**(12), 802–804 (1971)
7. Bringmann, K., Friedrich, T.: Tight Bounds for the Approximation Ratio of the Hypervolume Indicator. In: R. Schaefer, C. Cotta, J. Kołodziej, G. Rudolph (eds.) Parallel Problem Solving from Nature–PPSN XI, 11th International Conference, Proceedings, Part I, pp. 607–616. Springer, Lecture Notes in Computer Science Vol. 6238, Kraków, Poland (2010)
8. Bringmann, K., Friedrich, T.: Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. Theoretical Computer Science **425**, 104–116 (2012)
9. Bringmann, K., Friedrich, T.: Convergence of Hypervolume-Based Archiving Algorithms II: Competitiveness. In: 2012 Genetic and Evolutionary Computation Conference (GECCO'2012), pp. 457–464. ACM Press, Philadelphia, USA (2012)
10. Brockhoff, D., Friedrich, T., Neumann, F.: Analyzing Hypervolume Indicator Based Algorithms. In: G. Rudolph, T. Jansen, S. Lucas, C. Poloni, N. Beume (eds.) Parallel Problem Solving from Nature PPSN X, *Lecture Notes in Computer Science*, vol. 5199, pp. 651–660. Springer Berlin Heidelberg (2008)

11. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, second edn. Springer, New York (2007)
12. Das, I., Dennis, J.E.: Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. SIAM Journal on Optimization **8**(3), 631–657 (1998)
13. Deb, K., Jain, H.: An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
14. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: A. Abraham, L. Jain, R. Goldberg (eds.) Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, pp. 105–145. Springer Berlin Heidelberg, USA (2005)
15. Fang, K.T., Wang, Y.: Number-Theoretic Methods in Statistics. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis (1994)
16. Fleischer, M.: The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics. In: C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, L. Thiele (eds.) Evolutionary Multi-Criterion Optimization (EMO 2003), *Lecture Notes in Computer Science*, vol. 2632, pp. 519–533. Springer Berlin Heidelberg, Faro, Portugal (2003)
17. Gale, D., Shapley, L.: College Admissions and the Stability of Marriage. The American Mathematical Monthly **69**(1), 9–15 (1962)
18. Hammersley, J.M.: Monte-Carlo methods for solving multivariable problems. Annals of the New York Academy of Sciences **86**(3), 844–874 (1960)
19. Huband, S., Hingston, P., Barone, L., While, L.: A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (2006)
20. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: 2008 IEEE Congress on Evolutionary Computation CEC'2008 (IEEE World Congress on Computational Intelligence), pp. 2424–2431. Hong Kong (2008)
21. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: 2008 IEEE Congress on Evolutionary Computation (CEC'2008), pp. 2419–2426. IEEE Press, Hong Kong (2008)
22. Knowles, J., Corne, D.: Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In: S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, T. Murata (eds.) Evolutionary Multi-Criterion Optimization EMO'2007, *Lecture Notes in Computer Science*, vol. 4403, pp. 757–771. Springer Berlin Heidelberg (2007)
23. Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation **8**(2), 149–172 (2000)
24. Kokolo, I., Hajime, K., Shigenobu, K.: Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to Optimal? In: Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001), vol. 2, pp. 957–962. IEEE Service Center, Piscataway, New Jersey (2001)
25. Korobov, N.M.: The approximate computation of multiple integrals. Doklady Akademii Nauk SSSR **124**, 1207–1210 (1959)
26. Kuhn, H.W.: The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly **2**(1–2), 83–97 (1955)
27. Li, B., Li, J., Tang, K., Yao, X.: Many-objective evolutionary algorithms: A survey. ACM Computing Surveys **48**(1), 13:1–13:35 (2015)
28. Li, H., Zhang, Q.: Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation **13**(2), 284–302 (2009)
29. Li, M., Yang, S., Liu, X., Shen, R.: A Comparative Study on Evolutionary Algorithms for Many-Objective Optimization. In: R.C. Purshouse, P.J. Fleming, C.M. Fonseca, S. Greco, J. Shaw (eds.) Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013, pp. 261–275. Springer. Lecture Notes in Computer Science Vol. 7811, Sheffield, UK (2013)
30. Munkres, J.: Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (1957)

31. Phan, D.H., Suzuki, J.: R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization. In: IEEE Congress on Evolutionary Computation (CEC'2013), pp. 1836–1845 (2013)
32. Purshouse, R.C., Fleming, P.J.: On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Transactions on Evolutionary Algorithms **11**(6), 770–784 (2007)
33. Rostami, S., Neri, F.: Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm. Integrated Computer-Aided Engineering **23**(4), 313–329 (2016)
34. Scheffé, H.: Experiments with mixtures. Journal of the Royal Statistical Society. Series B (Methodological) **20**(2), 344–360 (1958)
35. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation and Application. John Wiley, New York (1986)
36. Tan, Y.Y., Jiao, Y.C., Li, H., Wang, X.K.: MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives. Computers & Operations Research **40**(6), 1648–1660 (2013)
37. von Lücken, C., Baran, B., Brizuela, C.: A survey on multi-objective evolutionary algorithms for many-objective problems. Computational Optimization and Applications **58**(3), 707–756 (2014)
38. Wang, Y., Fang, K.T.: Number-Theoretic Method in Applied statistics (II). Chinese Annals of Mathematics. Serie B **11**, 859–914 (1990)
39. Yevseyeva, I., Guerreiro, A.P., Emmerich, M.T., Fonseca, C.M.: A Portfolio Optimization Approach to Selection in Multiobjective Evolutionary Algorithms. In: T. Bartz-Beielstein, J. Branke, B. Filipič, J. Smith (eds.) Parallel Problem Solving from Nature - PPSN XIII, 13th International Conference, pp. 672–681. Springer. Lecture Notes in Computer Science Vol. 8672, Ljubljana, Slovenia (2014)
40. Yuan, Y., Xu, H., Wang, B., Yao, X.: A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization. IEEE Transactions on Evolutionary Computation **20**(1), 16–37 (2016)
41. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007)
42. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (1999)