Available online at www.sciencedirect.com

ScienceDirect

Swarm and Evolutionary Computation 00 (2021) 1–29

Swarm and E. C.

A Decomposition-Based Many-Objective Evolutionary Algorithm Updating Weights When Required

Lucas R. C. de Farias[a], Aluizio F. R. Araújo[a]

[a]*Universidade Federal de Pernambuco (UFPE), Centro de Informática (CIn), Av. Jornalista Anibal Fernandes s/n, Recife, Brazil*

Abstract

Multi-objective evolutionary algorithms based on decomposition (MOEA/D) usually work effectively when they have an appropriate set of weight vectors. A uniformly distributed set of unchanging weight vectors may lead to well-distributed solutions over a smooth, continuous, and well-spread Pareto front. However, fixed-value weight vectors may lead to solutions that fail, depending on the geometry of the problem. Several studies have used a predefined lapse of time to adapt weight vectors. This suggests that adaptation may not be being performed at the most appropriate moments of the evolutionary process. This paper presents the MOEA/D with updating when required (MOEA/D-UR) that uses a metric that detects improvements so as to determine when to adjust weights and a procedure for dividing the objective space in order to increase diversity. The results of experimental tests, which used real-world problems and the problem classes WFG1-WFG9, DTLZ1-DTLZ7, IDTLZ1-2, and MaOP1-6 with 3, 5, 6, 8, 10, 12, and 15 objectives, suggest that MOEA/D-UR is more effective, when compared with ten state-of-the-art algorithms.

Keywords: Computational Intelligence, Optimization, Evolutionary Computation, Decomposition.

1. Introduction

A multi-objective optimization problem (MOP) is characterized by there being two or more potentially conflicting objectives that should be simultaneously optimized. An MOP can be formulated as follows [1]:

$$\begin{aligned} &\text{minimize } \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}) \ldots f_m(\boldsymbol{x}))^T, \\ &\text{subject to } \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

where $\boldsymbol{x}=(x_1 \ldots x_n)^T$ is an n-dimensional decision vector in the space $\mathbb{R}^n$ and $\boldsymbol{F}(\boldsymbol{x})$ is an m-dimensional objective vector in the objective space $\mathbb{R}^m$. $\Omega \subset \mathbb{R}^n$ determines the feasible region of the decision variables. An optimization problem with more than three objectives is called a many-objective optimization problem (MaOP) [1]. MaOPs have attracted growing research interest from the evolutionary computing community due to their wide range of applicability and relevance [1–3].

The conflicting nature of the different objectives sees to it that all optimal solutions cannot simultaneously reach the very best values for all objectives. Instead, solutions reaching optimal trade-offs between different objectives form a Pareto optimal front (POF). Multi-objective evolutionary algorithms (MOEAs) have been widely used to approximate the POF of MOPs. However, an MOEA may present substantial difficulties if used to solve MaOPs [4, 5].

In the context of Pareto-based MOEAs [6, 7], the selection criteria may lose their effectiveness with regard to guiding the population towards the POF. This occurs because the percentage of non-dominated solutions for a particular population increases significantly when the number of objectives is increased [8]. For indicator-based selection criteria [9–11], MOEAs use one or more indicators to evaluate the fitness values of individuals [12] and thus, they do not suffer from the difficulties arising from the loss of selection pressure. However, the literature suggests that a critical aspect in practice is the complexity of their runtime [13]. Additionally, we have briefly described a promising-region based evolutionary multi-objective optimization algorithm (PREA) [14], and included it in the group of algorithms that we compare with each other. Decomposition-based methods use weight vectors to select solutions in the objective space [15]. The final set of solutions depends on how these weight vectors are distributed. Thus, they present the challenge of dealing with the complexity of the adequately assigning the weight vectors to the high-dimensional objective space [16, 17]. Recently, decomposition-based MOEAs have received much attention from researchers. [1, 8] In the literature, the methods for adapting these weight vectors have been considered a promising approach that can be used in decomposition-based MOEAs to deal with MaOPs [18–23].

Using a multi-objective evolutionary algorithm based on decomposition (MOEA/D) [15] is one of the most relevant options among the decomposition-based MOEAs. It breaks an MOP down into a number of single-objective optimization subproblems using a set of weight vectors and then uses a search heuristic to optimize these subproblems cooperatively. The diversity of the evolutionary population is controlled explicitly by weight vectors. Each weight corresponds to a subproblem, ideally associated with a single solution in the population. There are also cases where each weight corresponds to a set of multi-objective subproblems, such weights being associated with multiple solutions in the population [24]. Thus, diverse weights may give rise to different Pareto optimal sets [25].

Recent studies have pointed out that the performance of an MOEA/D can strongly depend on the shape of the Pareto front (PF) of the problem [17, 26]. In other words, the use of unchanging weight vectors [15, 27] may be appropriate for regular PFs i.e., those that are smooth, continuous, and well spread. However, when dealing with irregular PFs i.e., those that are degenerate, disconnected, inverted, or sharp-tailed, the final solution set can present results that fall below the initial expectations. Thus, the best results obtained by MOEA/D are most likely to occur only if the distribution of weight vectors is consistent with the PF shape of the problem to be solved [3, 17, 26]. A potential solution to this problem consists of approaches that can progressively modify the weights during the evolutionary process. Several interesting attempts have been made to use this strategy [1, 23, 25, 28–34]. Even though several approaches in the literature adapt weights, the interval between the two adaptations is predefined. This suggests that adaptation may not have been performed at the most appropriate time during the entire evolutionary process.

For the decomposition-based MOEAs, the evolutionary replacement of individuals in the current population may not be uniformly distributed; i.e., there may be regions with different concentrations of individuals. Also, the decomposition-based approach does not prohibit the presence of repeated individuals. In the literature, there are algorithms that split the space of the objectives to strengthen the convergence of the population while maintaining its diversity. Liu et al. [24] proposed the MOEA/D-M2M that combines selection approaches based on decomposition and dominance. Each weight vector divides the objective space into sub-regions, thus allowing more than one solution to be allocated to each weight vector. Cheng et al. [35] proposed the OPE-MOEA that divides the space with uniformly distributed weights and prioritizes fitness evaluations for promising regions. The hpaEA [36] uses reference vectors with an environmental selection to identify prominent solutions and strengthen the selection pressure. Even though several approaches divide the objective space, the use of weight/reference vectors that are uniformly distributed constrains the number of sub-regions or allocated solutions. It is also interesting to note the potential use of maintaining an external population to increase diversity in each sub-region.

To tackle the points raised, we propose an algorithm called MOEA/D with Update when Required (MOEA/D-UR). The main contributions of this approach are as follows:

1. A new scheme to adapt the weight vectors is triggered depending on the status of the evolutionary process, i.e., adaptation only occurs whenever convergence is detected;
2. A new method is proposed to divide the objective space into groups to increase the spread of individuals in the population;
3. A new methodology is proposed for determining the parameter values of the periodicity that are used for adapting the weight vectors. We define a threshold equation to estimate the level of regularity of the POF shape and to determine the suitable parameters.
4. The proposed methodology can be conveniently plugged into any MOEA that adapts weight vectors.

Validation of the proposed method is conducted by testing it on a number of benchmark functions: WFG1-9 [37], DTLZ1-7 [38], IDTLZ1-2 [29], and MaOP1-6 [39]. Furthermore, we test the algorithm on real-world problems: a multi-objective traveling salesman problem (MOTSP) [40], a multi-objective knapsack problem (MOKP) [41], and a water resource planning problem (WRP) [42]. All experiments are carried out for three, five, six, eight, ten, twelve, and fifteen objectives. The MOEA/D-UR is compared with 10 other MOEAs or MaOEAs: A-NSGA-III [29], MOEA/D-AWA [30], KnEA [43], RVEA [31], NMPSO [44], MOEA/D-URAW [34], AR-MOEA [3], PREA [14], FDEA-I [45], and FDEA-II [45]. The overall results suggest that MOEA/D-UR is more effective in comparison with the other algorithms, especially for irregular problems, including real-world problems.

The remainder of this paper is organized as follows: Section 2 introduces and discusses the decomposition approaches and the methods for generating weights for decomposition-based MOEAs already presented in the literature Section 3 describes the proposed MOEA/D-UR in detail. Thereafter, Section 4 seeks to validate the proposed algorithm, based on simulated and real problems found in the literature. Section 5 draws some conclusions paper and presents some suggestions for future lines of research.

2. Relevant Previous Knowledge

This section introduces the decomposition approaches and the methods for generating weights for decomposition-based MOEAs already presented in the literature, which this paper draws on. Then, we discuss the motivation for constructing an approach that can be adapted to the moment at which weight or reference vectors are adjusted.

2.1. Decomposition Approach

Three widely used decomposition methods for multi-objective optimization are weighted sum, Tchebycheff (TCH), and penalty-based boundary intersection [32]. The model to be proposed takes the Tchebycheff decomposition approach:

$$\begin{aligned} &\text{minimize } g^{TCH}(\boldsymbol{x}|\boldsymbol{\lambda}, \boldsymbol{z}^*) = \max\nolimits_{1\leq j\leq m}(\lambda_j|f_j(\boldsymbol{x}) - z_j^*|), \\ &\text{subject to } \boldsymbol{x} \in \Omega \end{aligned} \tag{2}$$

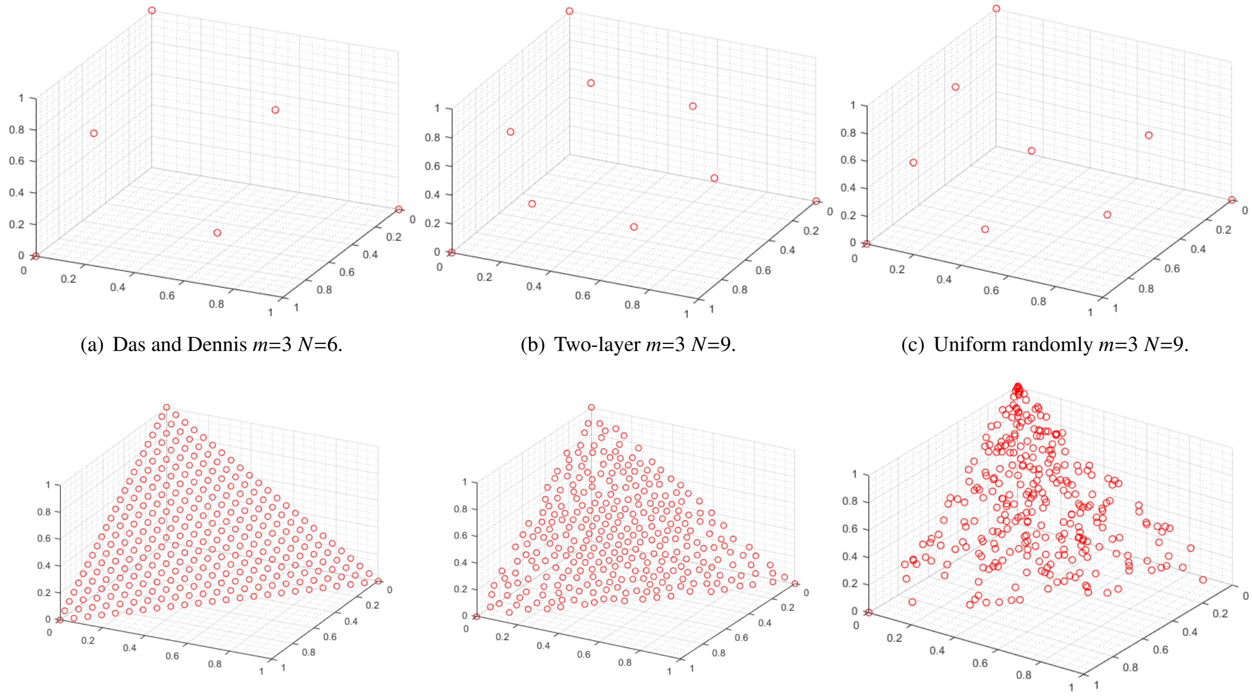
where m is the number of objectives and $\boldsymbol{z}^*$ is the reference point (utopian objective vector), i.e. $z_j^* = \min\{f_j(\boldsymbol{x})|\boldsymbol{x} \in \Omega\}$, for every $j = 1, ..., m$. The m-dimensional weight vector is defined as $\boldsymbol{\lambda} = (\lambda_1 \ ... \ \lambda_m)^T$, $\sum_i^m \lambda_i = 1$ and $\lambda_i \geq 0$, for all $i \in 1, ..., m$ [15]. By altering weight vectors, different Pareto-optimal solutions can be obtained by using the TCH approach [32].

2.2. Methods for Generating Weights

The original MOEA/D [15] uses the Das and Dennis (DD) approach [46], the simplex-lattice design method, to generate evenly distributed weight vectors. In this method, the population size grows significantly as the number of objectives increases. A different method of generating weights that lessens this limitation is the two-layer one [47]. In this approach, the DD is used in a boundary and internal layer. Thus, more weight vectors can be generated, especially for solving MaOPs. Another weight vector generation method widely used in MOEA/D variants [27, 34] is based on a uniform random sampling paradigm [48]. This paradigm differs from the previous ones since it can be adapted to set the population size [8]. Recently, Li et al. proposed a method for weight vector generation based on the Tchebycheff scalarization function [25], which is not based on the uniform paradigm but allows the size of the population to be freely adjusted. In this method, the weight vectors depend on how the population is distributed in the objective space at initialization. These four methods yield N weight vectors $\boldsymbol{\lambda} \in \mathbb{R}^m$, where m is the number of objectives, as described below.

2.2.1. Das and Dennis

Most variants of MOEA/D use the method proposed by Das and Dennis [46] to systematically generate a set of fixed weight vectors uniformly distributed over a unit simplex. Let H be established as the number of divisions of each axis, and then, a set of $N = \binom{H+m-1}{m-1}$ weight vectors can be generated. Since H should not be smaller than m to prevent creating intermediate points [49], the number of generated weight vectors that are generated may become very high for more than three objectives.



(a) Das and Dennis m=3 N=6. (b) Two-layer m=3 N=9. (c) Uniform randomly m=3 N=9.

(d) DD-based m=3 N=300. (e) Uniform randomly m=3 N=301. (f) Tchebycheff Function m=3 N=301.

Figure 1. Examples of the sets of weight vectors generated by the DD, two-layer, and UR approaches. N is the population size and m is the total number of the objectives.

2.2.2. *Two-layer*

This approach generates the weight vectors in three steps. Firstly, two sets with $N(N_1 + N_2)$ weight vectors, in the boundary layer ($\boldsymbol{B} = \{\boldsymbol{b}^1 \ldots \boldsymbol{b}^{N_1}\}$) and in the inside layer ($\boldsymbol{I} = \{\boldsymbol{i}^1 \ldots \boldsymbol{i}^{N_2}\}$), are initialized according to the DD method, with different H settings. Then, a coordinate transformation compresses the coordinates of weight vectors in the inside layer where each vector $\boldsymbol{i}^k = (\boldsymbol{i}_1^k \ldots \boldsymbol{i}_m^k)^T$, $k \in \{1 \ldots N_2\}$. Its j-th component is defined as

$$i_j^k = \frac{1-\tau}{m} + \tau \times i_j^k \tag{3}$$

where $j \in \{1 \ldots m\}$ and $\tau \in [0, 1]$ is a contraction factor (Li et al. [47] default τ to 0.5). In the third step, $\boldsymbol{B}$ and $\boldsymbol{I}$ are combined to form the final set of weight vectors $\boldsymbol{W}$ [47].

2.2.3. *Uniform Random (UR) Sampling*

The steps below describe the process as a set of N weight vectors (λ) is being generated:

1. From a uniform distribution, generate 5000 weight vectors, thereby constructing the set $\boldsymbol{\lambda}_1$. Also, initialize $\boldsymbol{\lambda}$ as the set containing all the weight vectors presented in the m-dimensional identity matrix.
2. Find the weight vector at $\boldsymbol{\lambda}_1$ with the highest of the shortest quadratic Euclidean distances for one of the weight vectors at $\boldsymbol{\lambda}$ [7], and then transfer it from $\boldsymbol{\lambda}_1$ to $\boldsymbol{\lambda}$.
3. Stop and return $\boldsymbol{\lambda}$ if the size of $\boldsymbol{\lambda}$ is N, otherwise, go to the second item and repeat the process.

Note that in this approach, the population size does not depend on the total number of objectives.

2.2.4. *Tchebycheff Function (TF)*

Li et al. [25] proposed a method for generating the weight vector based on the TCH function. The method is often used to adapt weight vectors [25, 30, 50]. Formally, let $\boldsymbol{z}^*$ be the reference point and $\boldsymbol{\lambda}$ be the optimal weight vector for a solution $\boldsymbol{x}$. Then Equation (4) holds:

$$\frac{f_1(\boldsymbol{x}) - z_1^*}{\lambda_1} = \frac{f_2(\boldsymbol{x}) - z_2^*}{\lambda_2} = \cdots = \frac{f_m(\boldsymbol{x}) - z_m^*}{\lambda_m} \tag{4}$$

Since $\lambda_1 + \lambda_2 + ... + \lambda_m = 1$, we have Equation (5).

$$\boldsymbol{\lambda} = (\lambda_1 \cdots \lambda_m)^T = (\frac{f_1(\boldsymbol{x}) - z_1^*}{\sum_{i=1}^m f_i(\boldsymbol{x}) - z_i^*} \cdots \frac{f_m(\boldsymbol{x}) - z_m^*}{\sum_{i=1}^m f_i(\boldsymbol{x}) - z_i^*})^T \tag{5}$$

Figure 1 presents the sets of weight vectors generated by the DD, two-layer, UR, and TF approaches. Such methods assign any value to N for the case of 2 objectives. However, for the 3-objective cases, the DD approach permits N values that cause gaps, N=10 is the only option allowed above 6. The two-layer decreases the number of gaps, thus allowing N=9. Figure 1.(d) illustrates the case where N=300 for 3 objectives. Both DD-based approaches allow 325 as the size of next population. Instead, the UR approach can reach other N values in those gaps, for instance N=301 (Figure 1.(e)). The TF approach also can make N=301 (Figure 1.(f)); however, the weight vectors depend on the population, which produces a non-uniform distribution.

2.3. MOEA/D

As a typical instance of using decomposition-based multi-objective evolutionary algorithms, MOEA/D [15] decomposes an MOP into a number of scalar optimization subproblems, and these subproblems associated with different weight vectors are solved simultaneously. The main task of MOEA/D concerns how to approximate the optimal solutions of these different single-objective optimization subproblems [20].

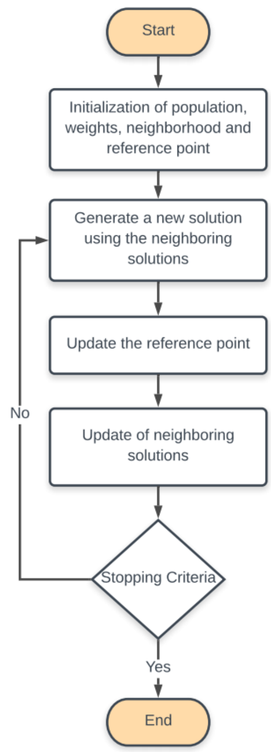


Figure 2. Flowchart of execution of the MOEA/D [15]

Figure 2 presents a flowchart of the components of the original version of MOEA/D [15]: initialization, reproduction, replacement, and stopping operators. Alg. 1 presents this MOEA/D version in more detail. This algorithm is characterized by a set of subproblems that are solved collaboratively. The collaboration is manifested as the local mating (line 6) and the local updating (line 9).

Algorithm 1: General Framework of MOEA/D

Initialize the N weight vectors $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^1 \ldots \boldsymbol{\lambda}^N)$ and each neighborhood $\boldsymbol{B}(i)$
Initialize the population $\boldsymbol{Pop} = (\boldsymbol{x}^1 \ldots \boldsymbol{x}^N)$ and calculate all fitness $\boldsymbol{F}(\boldsymbol{x}^i)$
Initialize the reference point $\boldsymbol{z}^*$ according to $\boldsymbol{F}(\boldsymbol{Pop})$;
while *an end condition is not met* **do**
  for *each subproblem i = 1 to N* **do**
    $\boldsymbol{y} \leftarrow$ Reproduction $(\boldsymbol{Pop}, \boldsymbol{B}(i))$
    Calculate $\boldsymbol{F}(\boldsymbol{y})$
    Update the reference point $(\boldsymbol{z}^*, \boldsymbol{F}(\boldsymbol{y}))$
    Replacement $(\boldsymbol{Pop}, \boldsymbol{\lambda}, \boldsymbol{B}(i), \boldsymbol{z}^*, \boldsymbol{y})$
Return $\boldsymbol{Pop}$

2.4. Adaptation Timing

In recent years, a large number of decomposition-based approaches that adapt weight or reference vectors have been proposed in the literature. The moment adopted for adapting each of these is summarized below:

- paλ-MOEA/D [28]: Only once in the whole evolutionary process, when the external population size is greater than or equal to twice the size of the evolutionary population.
- TPEA-PBA [51]: Only once in the whole evolutionary process, when the final 50 generations have been reached.
- A-NSGA-III [29], MOEA/D-RW [52], A-IM-MOEA [53], PICEA-w [54], GP-A-NSGA-III [55], DBEA-DS [56], AR-MOEA [3], EARPEA [57], DDEA [58], DEA-GNG [59], and iRVEA [60]: Once in each generation.
- MOEA/D-SOM and M2M-SOM [61]: Starting when the external population size is greater than or equal to five times the size of the evolutionary population, once in each generation.
- MOEA/D-LTD [62]: Starting at 30% of the evolutionary process, every 20 generations.
- W-MOEA/D [63]: Starting at 33.3% of the evolutionary process, every 20 generations.
- MOEA/D-AWG [32]: At between 30% and 70% of the evolutionary process, every 20 generations.
- MOEA/D-AWA [30]: Starts after 80% of the evolutionary process has occurred, according to the *wag* parameter (PlatEMO [64] defaults to 100), every 20 generations.
- A-GWASF-GA [23]: Starting at 80% of the evolutionary process, every 3.3%.
- AdaW [25] and MOEA/D-URAW [34]: Every 5%, up to 90% of the evolutionary process.
- RVEA [31]: Every 10% of the evolutionary process.
- g-DBEA [33]: Every 10 generations.
- AWD-MOEA/D [65] and MOEA/D-AM2M [1]: Every 100 generations.
- FV-MOEA/D [66]: Every 300 generations.
- tw-MOEA/D [67]: Starting at 50,000 fitness evaluations of the evolutionary process, every 5,000 fitness evaluations.

The number of adaptations ranges from once in the whole evolutionary process up to once in every generation. Thus, there is not a consensus on the appropriate moment to adapt. The proposed MOEA/D-UR is presented in Section 3 below.

3. MOEA/D with Update when Required (MOEA/D-UR)

In this Section, the details of the MOEA/D-UR are set out. Initially, a high-level scheme is shown in Figure 3. We should emphasize that two stages are inherited from the original MOEA/D [15, 27]:

- Reproduction: Identically to the original MOEA/D [15, 27], we used a δ parameter that assigns a probability to the selection of parents which is carried out from one of two origins: the neighborhood of the weight vector or the entire population.
- Replacement: The original MOEA/D [15] allowed a descendant to replace all solutions in the neighborhood if it improved the result of the applied decomposition function applied, thus causing a significant drop in the diversity levels of the original model. A subsequent version allows a new solution to replace a maximum of two existing solutions [27]. This is used in MOEA/D-UR.

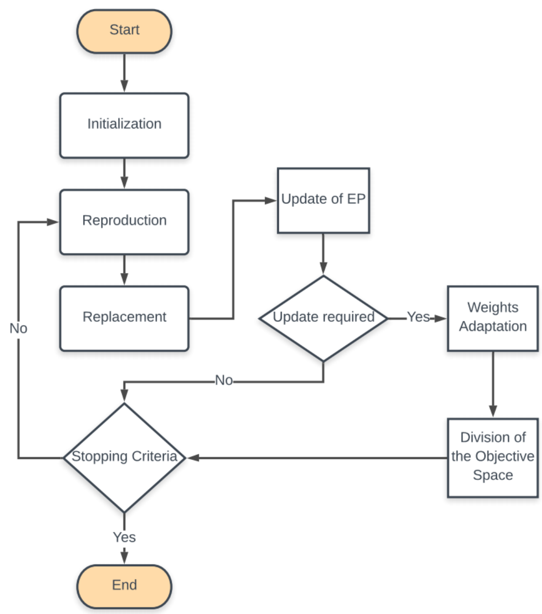


Figure 3. Flowchart of execution of the MOEA/D-UR

In addition to these elements, the remaining items that make up the MOEA/D-UR are the following:

- Initialization: Like most MOEA/Ds, in this stage, we define the individuals of the population, P, weight vectors, λ, and the reference point, z*. In particular, the weight vectors are initially generated using the UR sampling; and the WS-transformation is applied to these weight vectors (Subsection 3.1).
- Update of EP: The external population containing only non-dominated individuals is unlimited (Subsection 3.3).
- Weights adaptation: The sparsity level of individuals influences the insertion and removal of the weight vectors (Subsection 3.4).
- Division of the objective space: This operation aims to improve the spread of individuals in the POF of the main population. It is triggered after the weights are adapted (Subsection 3.5).
- Update required: MOEA/D-UR verifies the convergence of the individuals of the population to decide whether or not the weight vectors have been adapted (Subsection 3.6). Selecting dynamic parameters for periodicity is carried out to determine when the adaptation occurs. (Subsection 3.7).

- Stopping Criteria: the algorithm stops when the maximum number of generations or fitness evaluations is reached.

In the following paragraphs, we provide details on the components of the MOEA/D-UR, and finally, we analyze its computational complexity (Section 3.8).

3.1. Generating Weights

First, the MOEA/D-UR uses the UR method (Section 2.2.3) to generate weight vectors. Then, the WS-transformation maps the weight vector of a scalar subproblem, $\boldsymbol{\lambda} \in \mathbb{R}^m$, into its solution mapping vector, $\boldsymbol{\lambda'}$ [30].

Miettinen [68] proved that under mild conditions, for each Pareto optimal solution $\boldsymbol{x}$, there is a weight vector $\boldsymbol{\lambda}$ such that $\boldsymbol{x}$ is the optimal solution of Equation (2). On the other hand, each optimal solution of Equation (2) is a Pareto optimal solution to Equation (1). This property allows different Pareto optimal solutions to be obtained by varying the weight vectors. Qi et al. [30] proposed the WS-Transformation for the weight vectors, $\boldsymbol{\lambda}$, to map the solution vectors, $\boldsymbol{\lambda'}$, by:

$$\boldsymbol{\lambda'} = WS(\lambda) = \left(\frac{\frac{1}{\lambda_1}}{\sum_{i=1}^{m}\frac{1}{\lambda_i}} \cdots \frac{\frac{1}{\lambda_m}}{\sum_{i=1}^{m}\frac{1}{\lambda_i}}\right)^T \tag{6}$$

where $\boldsymbol{\lambda} = 0$ is replaced by 10^{-6}.

3.2. Sparsity Level

The sparsity level between individuals indicates the subproblem to be removed and the fittest one to be added. The sparsity level is based on a neighborhood distance [69]

$$SL(\boldsymbol{ind^j}, \boldsymbol{pop}) = \prod_{i=1}^{m} L_2^{NN_i^j} \tag{7}$$

where $L_2^{NN_i^j}$ is the Euclidean distance between the j-th individual, $\boldsymbol{ind^j}$, and its i-th nearest neighbor in the population, $\boldsymbol{pop}$. The parameter m (number of objectives) defines the total number of closest Euclidean distances to be taken into account [30, 34].

3.3. Update of the External Population (EP)

We use the EP to store the visited non-dominated solutions and to provide guidance for adding and removing subproblems in the current population under evolution with a view to augmenting diversity. In MOEA/D-UR, the size of the EP is unlimited. As the EP is designed to adjust the weight vectors, the larger the size of the EP, the more suitable it should be to adjust the weight vectors to capture different PFs. To avoid high computational endeavor, the EP is reset to zero after adaptation. The update of the EP is shown in Alg. 2. If an offspring $\boldsymbol{o}$ is non-dominated concerning the $\boldsymbol{EP}$, then it receives the offspring $\boldsymbol{o}$ at the expense of the removal of any $\boldsymbol{q}$ solution that is dominated by offspring $\boldsymbol{o}$ (lines 1-5).

Algorithm 2: Update External Population

1 Define: $\boldsymbol{EP}$ (external population) and $\boldsymbol{O}$ (offsprings)
2 **for** *each* $\boldsymbol{o} \in \boldsymbol{O}$ **do**
3   **if** $\nexists \boldsymbol{q} \in \boldsymbol{EP}, \boldsymbol{q} \prec \boldsymbol{o}$ **then**
4         $\boldsymbol{EP} \leftarrow \boldsymbol{EP} \cup \boldsymbol{o}$;
5         $\boldsymbol{EP} \leftarrow \boldsymbol{EP} \setminus \{\boldsymbol{q} \in \boldsymbol{EP} \mid \boldsymbol{o} \prec \boldsymbol{q}\}$;
6 **return** $\boldsymbol{EP}$;

3.4. Adapting Weights

The number of updated subproblems (*nus*) in this model is a percentage of the population size, i.e. a number of the subproblems are removed from the population and the same number of new subproblems are created and then added to the population. Alg. 3 describes the process of updating the set of weight vectors.

Initially, the sparsity level of each individual in the population $\boldsymbol{P}$ is calculated using Equation (7). The *nus* individuals with the lowest sparsity level or overcrowding are removed [34] (lines 1-6).

The procedure to create new *nus* subproblems consists of calculating the sparsity level of each individual of the $\boldsymbol{EP}$ with respect to the current population $\boldsymbol{P}$, using Equation (7). Then, each new subproblem $\boldsymbol{\lambda^{sp}}$ is yielded by using the individual with the current highest sparsity level from $\boldsymbol{EP}$, $\boldsymbol{x^{sp}}$. $\boldsymbol{\lambda^{sp}}$ can be formulated as follows:

$$\boldsymbol{\lambda^{sp}} = \left(\frac{\frac{1}{f_1^{sp}-z_1^*}}{\sum_{k=1}^{m}\frac{1}{f_k^{sp}-z_k^*}} \cdots \frac{\frac{1}{f_m^{sp}-z_m^*}}{\sum_{k=1}^{m}\frac{1}{f_k^{sp}-z_k^*}}\right)^T \tag{8}$$

considering $\boldsymbol{F(x^{sp})} = (f_1^{sp} \ \ldots \ f_m^{sp})^T$ and the current reference point z^*. Finally, the solution $\boldsymbol{x^{sp}}$ is associated with the new subproblem $\boldsymbol{\lambda^{sp}}$ and added to the population (lines 7-11).

Last of all, the change in positioning implies an update in the neighborhood, $\boldsymbol{B}(i)$, of each weight vector $\boldsymbol{\lambda^i} \in \boldsymbol{\lambda}$ (line 13).

Algorithm 3: Update Weights

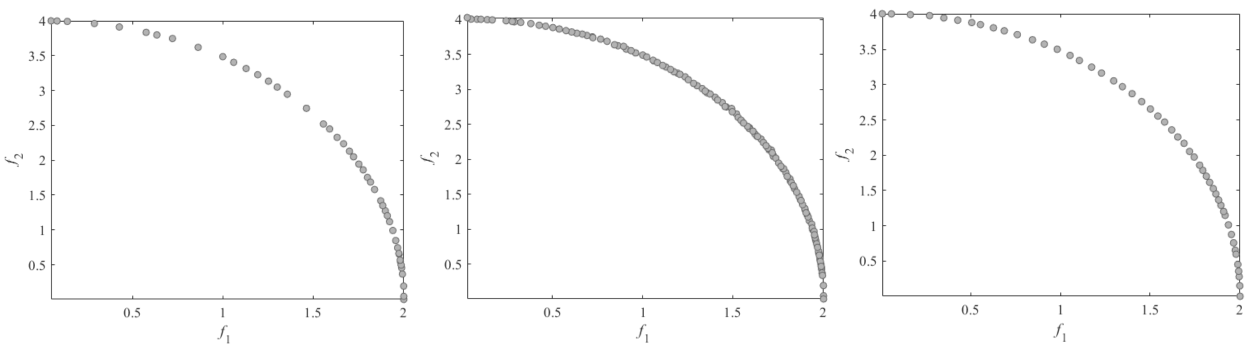
1 Define: $\boldsymbol{P}$ (population), $\boldsymbol{\lambda}$ (weights), $\boldsymbol{EP}$ (external population), and *nus* (number of updated subproblems)
2 $adjust \leftarrow 0$
3 **while** $adjust < nus$ **do**
4   Calculate the sparsity level of each individual in population $\boldsymbol{P}$ among $\boldsymbol{P}$ using Equation (7);
5   Remove the smallest sparsity level individual in $\boldsymbol{P}$ and the associated weight vector;
6     $adjust \leftarrow adjust + 1$;
7 **while** $adjust > 0$ **do**
8   Calculate the sparsity level of each individual in $\boldsymbol{EP}$ among the population $\boldsymbol{P}$ using Equation (7);
9   Generate a new subproblem $\boldsymbol{\lambda^{sp}}$ by Equation (8) using the individual with the largest sparsity level $\boldsymbol{x^{sp}}$;
10   Add the newly constructed subproblem $\boldsymbol{\lambda^{sp}}$ to $\boldsymbol{\lambda}$ associated to the individual $\boldsymbol{x^{sp}}$ to current population $\boldsymbol{P}$;
11     $adjust \leftarrow adjust - 1$;
12 Update the neighborhood $\boldsymbol{B}(i)$ of each $\boldsymbol{\lambda^i} \in \boldsymbol{\lambda}$;
13 **return** $\boldsymbol{P}$, $\boldsymbol{\lambda}$, $\boldsymbol{B}$;

3.5. Division of the Objective Space

The objective space is divided to include new non-dominated individuals in the current population, $\boldsymbol{P}$. These individuals then increase the spread of the Pareto Optimal Front (POF). The evolutionary replacement of individuals in the current population may not be uniformly distributed i.e., there may be regions in which there is a higher concentration of individuals than in others. Also, the decomposition-based approach does not prohibit the presence of repeated individuals, thus impairing diversity. To mitigate this scenario, we propose to divide the objective space into a k number of sets. Such a division allows the addition of new individuals to the current population and the external population. Therefore, the individuals that contribute to improving the spread in the current POF are preserved. Figure 4 shows the population before and during the addition of new points, and selecting the N points that should remain in the evolutionary process.

The pseudo-code for this procedure is described in Alg. 4. Initially, the k-means algorithm [70] is used to divide the objective space into k groups. For each one of them, N (population size) weight vectors are generated by the UR method. Also, the algorithm constructs a Union set, $\boldsymbol{U}$, formed by individuals in the $\boldsymbol{EP}$ and in the current population, $\boldsymbol{P}$ (lines 1-4).

Then, for each weight vector of each group, the algorithm detects which individual in $\boldsymbol{U}$ has the best result from the TCH decomposition function in order to associate the individual to the weight vector (lines 5-8). After assigning



(a) (b) (c)

Figure 4. The behavior of the population in the division of objective space in the WFG4 problem of two objectives. (a) Current population before the division approach. (b) Individuals during the division approach. (c) Current population after the division approach

---

Algorithm 4: Division of Objective Space

Define: $\boldsymbol{P}$ (population), $\boldsymbol{\lambda}$ (weights), $\boldsymbol{EP}$ (external population), N (population size), and z^* (reference point)
Initialize a set $\boldsymbol{\lambda^n}$ of N weight vectors using the UR approach;
$\boldsymbol{G} \leftarrow$ three groups by using k-means to divide the objective space of population $\boldsymbol{P}$;
$\boldsymbol{U} \leftarrow \boldsymbol{P} \cup \boldsymbol{EP}$;
for *each group* $\boldsymbol{g} \in \boldsymbol{G}$ **do**
  $original\_group\_size \leftarrow \boldsymbol{group\_size}(g)$
  $\boldsymbol{\lambda^g}$ receives $\boldsymbol{\lambda^n}$ normalized with the max-min values of the weight vectors $\boldsymbol{\lambda}$ from individuals in group $\boldsymbol{g}$;
  Assign to each weights vector in $\boldsymbol{\lambda^g}$ the individual of $\boldsymbol{U}$ with the best g^{TCH};
  Randomly select mating solutions from the group $\boldsymbol{g}$ to generate *original_group_size* offsprings $\boldsymbol{O}$;
  $\boldsymbol{z^*} \leftarrow min(\boldsymbol{z^*}, \boldsymbol{F}(\boldsymbol{O}))$;
  $\boldsymbol{U} \leftarrow \boldsymbol{U} \cup \boldsymbol{O}$;
for *each solution* $s \in \boldsymbol{U}$ **do**
  $\boldsymbol{\lambda^j} \leftarrow arg_{\boldsymbol{\lambda^k} \in \boldsymbol{\lambda}} min\{g^{TCH}(\boldsymbol{x}^s|\boldsymbol{\lambda}^k, \boldsymbol{z}^*)\}$;
  if $\nexists \boldsymbol{x}^s \in \boldsymbol{P}$ ***and*** $g^{TCH}(\boldsymbol{x}^s|\boldsymbol{\lambda}^j, \boldsymbol{z}^*) \leq g^{TCH}(\boldsymbol{x}^j|\boldsymbol{\lambda}^j, \boldsymbol{z}^*)$ **then**
    $\boldsymbol{x}^j \leftarrow \boldsymbol{x}^s$;
return $\boldsymbol{P}$, $\boldsymbol{EP}$, $\boldsymbol{z^*}$;

---

solutions from U to the weight vectors, a new offspring per group is generated with as many individuals as the number of solutions in each group. The variation operators are those already applied to the current population: simulated binary crossover (SBX) and polynomial mutation. The new individuals are added to the $\boldsymbol{U}$ set (lines 9-11).

Finally, for each weight vector of the current population, $\boldsymbol{P}$, the algorithm determines the individual in $\boldsymbol{U}$ with the best TCH decomposition result. Such an individual from $\boldsymbol{U}$ replaces the individual in $\boldsymbol{P}$ if this individual is not already present in $\boldsymbol{P}$ and has a better TCH decomposition result. Consequently, the spread in the current population can be improved by inserting such distinct and non-dominated individuals (lines 12-16).

K-means is useful because it automatically assigns solutions to a group. However, before starting, the number of groups has to be chosen. An experimental design study [71] is conducted so as to make a sensitivity analysis of the parameter k. The experiments show the normalized HV outcome for the parameter k, ranging from 2 to 15, for regular (DTLZ1-4 and WFG4-9) and irregular (DTLZ5-7, IDTLZ1-2, and WFG1-3) problems with 3, 5, 8, 10, and 15 objectives (Figure 5). Besides, the situation without division was also tested and this showed a performance that was inferior to that of the divided objective space. The best overall performance was reached with k=10.

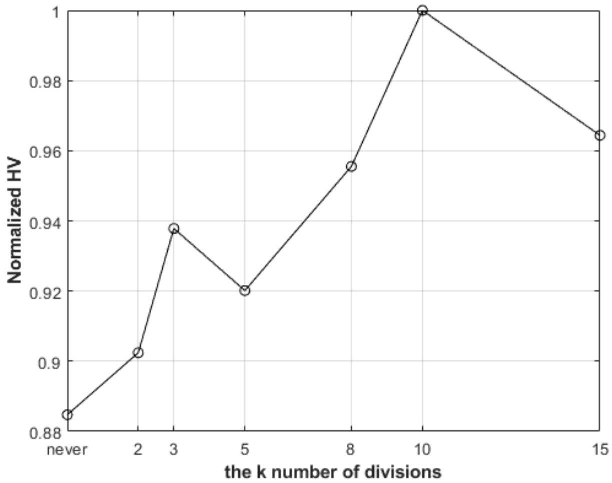


Figure 5. Performance of MOEA/D-UR (measured by a normalized hypervolume metric, the larger, the better) with different k specification values. It includes the scenario in which there is not a division of the objective space.

3.6. Update When Required

The use of the improvement metric (IM), Equation (9), aims at detecting a trend of convergence of the population solutions to the weight vectors, evidenced by little variation in the mean values of the Tchebycheff function.

$$IM = 1 - \frac{mean(\boldsymbol{I}_{new})}{mean(\boldsymbol{I}_{old})} \tag{9}$$

where the variables ($\boldsymbol{I_{new}}$) and ($\boldsymbol{I_{old}}$) are vectors with the values obtained from the Tchebycheff decomposition function (Equation (2)), from each population solution in considering the current and the previous instant of time.

As presented in Subsection 2.4, regarding periodicity, four parameters are usually defined in the literature: *start* and *end* define the interval throughout the evolutionary process to start and stop the verification of the weight vector adaptation; *period* determines the gap between verifications established in generations, and *nus* defines the number of updated subproblems. The MOEA/D-UR proposes the use of a new parameter for periodicity, ρ. Its role takes into consideration that the best moment to adapt the weight vector is when the individuals of the population converge. Weights are updated when the absolute value of the IM is less than or equal to a threshold, the parameter ρ. The next Subsection details the methodology for the dynamic selection of these five parameters that is used in the MOEA/D-UR.

Alg. 5 describes the steps to perform the update when required. Initially, the updating of the weight vectors when required is started when the evolutionary process reaches a given percentage of the maximum number of fitness evaluations, defined by the *start* parameter (line 2). Verification of the POF shape is made by examining the fitness of the population, thus defining the other periodicity parameters: *period*, *nus*, and ρ (lines 3-11). Finally, for every *period* of generations, between the *start* and *end* of the evolutionary process, the MOEA/D-UR uses the IM to check whether or not the weight vectors of the population must be updated (lines 12-17). When the convergence of the solutions is determined, i.e., IM has a value lower than that of the ρ parameter, so the procedure for removing and creating weight vectors is triggered (line 18). Thus, the objective space is divided, where new and old solutions are used to improve the spread of the current population. (line 19).

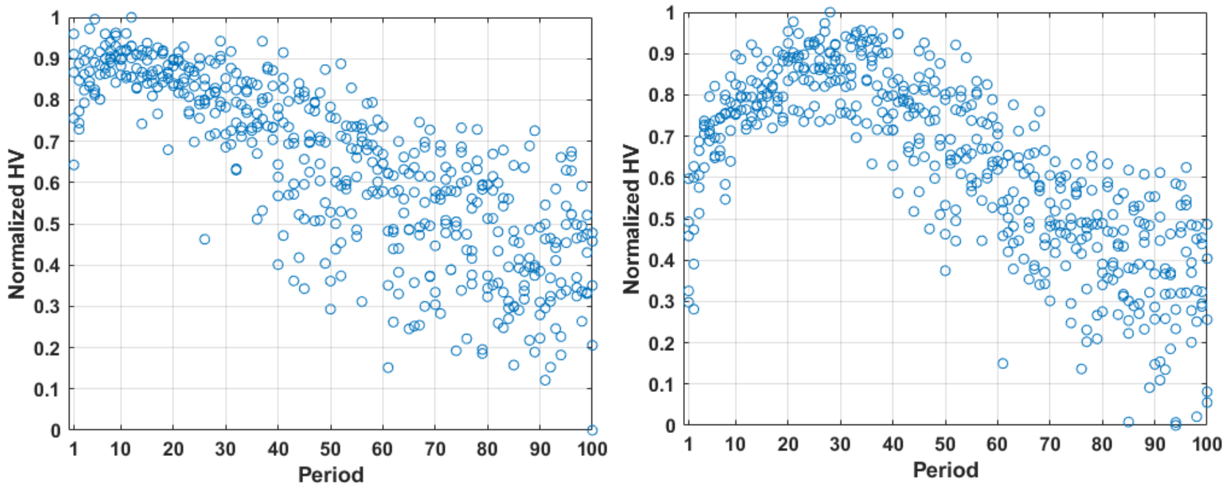
Algorithm 5: Update When Required

1 Define: $\boldsymbol{P}$ (population), $\boldsymbol{\lambda}$ (weights), $\boldsymbol{EP}$ (external population), N (population size), and z^* (reference point)
2 **if** Gen/Gen_{max} == *start* **then**
3   Calculate the *spread index* using Eq. 11;
4   Calculate the *threshold* using Eq. 12;
5   **if** *spreading index* $\leq$ *threshold* **then**
6   *period* $\leftarrow$ *period_regular*;
7   *nus* $\leftarrow$ *nus_regular*;
8   **else**
9   *period* $\leftarrow$ *period_irregular*;
10   *nus* $\leftarrow$ *nus_irregular*;
11   Calculate the ρ value using Eq. 13;
12     $\boldsymbol{I_{old}} \leftarrow g^{TCH}(\boldsymbol{P}|\boldsymbol{\lambda}, z^*)$;
13 **if** *mod(Gen,period) == 0* ***and*** $Gen/Gen_{max} \leq$ *end* **then**
14     $\boldsymbol{I_{new}} \leftarrow g^{TCH}(\boldsymbol{P}|\boldsymbol{\lambda}, z^*)$;
15   Calculate IM using Eq. 9;
16     $\boldsymbol{I_{old}} \leftarrow \boldsymbol{I_{new}}$;
17   **if** $abs(IM) \leq \rho$ **then**
18         $[\boldsymbol{P}, \boldsymbol{\lambda}, \boldsymbol{B}] \leftarrow$ Perform weights adaptation using Alg. 3;
19         $[\boldsymbol{P}, z^*] \leftarrow$ Divide objective space using Alg. 4;
20         $\boldsymbol{EP} \leftarrow \emptyset$;
21 **return** $\boldsymbol{P}, \boldsymbol{\lambda}, \boldsymbol{B}, \boldsymbol{EP}, z^*$;

3.7. Determination of the Dynamic Parameters

In Subsection 2.4, various fixed values of periodicities used in the literature are shown. Using a single parametric configuration for adapting the periodicity for any problem, regardless of the number of objectives and (apagar: the) POF shapes, may not be suitable. Such a unique parametric setup implies that the set of weight vectors must be modified according to the same periodicity for all POF shapes [17]. Thus, using a methodology to select the parametric configuration for adapting periodicity, while taking into account problem features such as the POF shape and the number of objectives, can improve the performance of MOEAs that adapt the set of weight vectors.

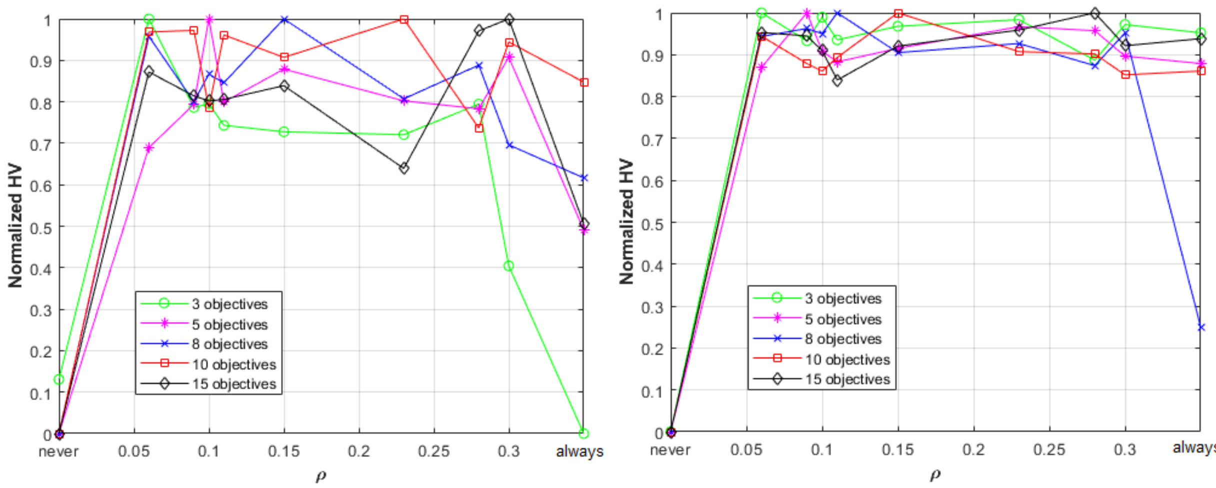
An experimental design technique [71] was used to determine the MOEA/D-UR parametric set up for *start*, *end*, *period*, *nus*, and ρ. The Latin Hypercube Sample (LHS) [72] was used to generate 500 parameter tuples, at the min-max ranges: 0-0.3; 0.7-1; 1-100; 0.01-0.3; and 0.01-0.1 for regular (DTLZ1-4 and WFG4-9) and irregular (DTLZ5-7, IDTLZ1-2, and WFG1-3) PFs with 3, 5, 8, 10, and 15 objectives. The results of determining the parameters are presented in Figure 6.(a) for regular PF shapes and their respective highest normalized HV value is reached when *start*, *end*, *period*, *nus*, and ρ are 0.2, 0.9, 12, 0.25, and 0.09, respectively. Figure 6.(b) shows the results for irregular PFs in which the highest value for HV is obtained when *start*, *end*, *period*, *nus*, and ρ are 0.2, 0.9, 28, 0.075, and 0.1, respectively. In the resulting 5-dimensional tuples, *start* and *end* assume the same values for both scenarios. However, other parameters, *period*, *nus*, and ρ, depend on the category of the PF shape.



(a) Regular PF shapes. (b) Irregular PF shapes

Figure 6. Result obtained by the LHS of 500-tuple configurations for regular and irregular problems with 3, 5, 8, 10, and 15 objectives.

A second experimental design study is conducted to verify the adequacy of using the parameter ρ as a threshold for the adaptation of weight vectors. This study uses the best tuple found by the LHS method for regular and irregular problems. Figure 7 shows the performance of the parameter ρ for a broader range, 0.01 to 0.3, depending on the number of objectives and the PF shape. Besides, a comparison is made with the scenarios of always adapting and never adapting the weight vectors to the best tuple periodicity. We notice that not adapting the weight vectors, in general, reaches a performance that is worse than adapting them continuously according to the period determined by the LHS method, except for regular problems with three objectives. Moreover, using the ρ parameter together with IM to identify convergence presents a performance that is better than always adapting it in that period. However, to choose an appropriate ρ value, one must be aware of the PF shape and the number of objectives of the problem.



(a) Regular PF shapes. (b) Irregular PF shapes

Figure 7. Performance of MOEA/D-UR (measured by a normalized hypervolume metric, the larger, the better) with different ρ specification values. It includes the scenarios in which adapting weight vectors is always performed and never performed.

As seen in Figure 7, regular and irregular PFs have peak performances under different parametric tuples. Therefore, we defined two equations for determining the parametric tuple depending on the features of a given MOP or

MaOP. The first expression calculates the spread index (SI) to assess the degree of regularity of a considered POF. Following that evaluation, the second equation calculates ρ, the most sensitive parameter, independently of the number of objectives or the dimension of the decision space. Such estimation can be used for determining the value of the parameter for any particular MaOP.

The first equation consists of the Frobenius norm [73] applied to the population fitness regularized by 2-norm defined as:

$$\boldsymbol{F'}(\boldsymbol{x}) = (\frac{f_1}{\sqrt{\sum_{k=1}^{m} f_k^2}} \cdots \frac{f_m}{\sqrt{\sum_{k=1}^{m} f_k^2}})^T \tag{10}$$

where the fitness of the individual $\boldsymbol{x}$, $\boldsymbol{F}(\boldsymbol{x}) = (f_1 \ldots f_m)^T$. Hence, SI can be formulated as follows:

$$SI(\boldsymbol{F'}(\boldsymbol{pop}), h) = \frac{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{m} f_{ij}^2}}{h} \tag{11}$$

where f_{ij} is the normalized fitness of the i-th individual of a population, $\boldsymbol{pop}$, of size N in the j-th objective. The parameter h is used for normalization purposes. In this paper, the value four is adopted without loss of generality. This metric is used at 20% of the evolutionary process (*start* parameter). Figure 8 presents the median of 31 runs for each regular and irregular problem in this paper for objectives 3, 5, 8, 10, and 15. Regular POFs are easy to distinguish from this metric because they very often present values below an average value of the medians. However, there are irregular PFs with values greater than, similar to, or smaller than those of regular POFs. This can be understood by using the 2-norm to position the fitness of the solutions. Particularly in irregular POFs, these proximities can be very high or low while, for regular POFs, the values can be clearly grouped. The threshold equation consists of the interpolation of an average value for each objective defined as:

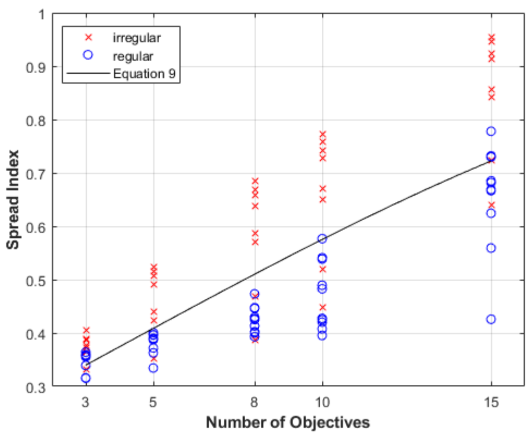


Figure 8. Median of 31 runs of the Spread Index on each regular and irregular problem with 3, 5, 8, 10, and 15 objectives. A threshold function marks the boundary between problems.

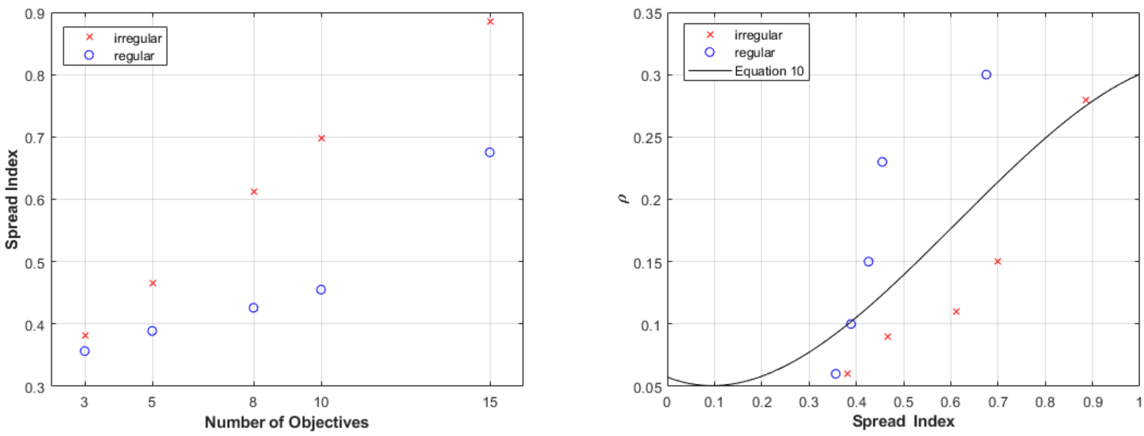
$$threshold(m) = -0.00001989 * m^3 + 0.0002034 * m^2 + 0.03376 * m - 0.2373 \tag{12}$$

where m is the number of MOP objectives. If the function SI gives an output lower than the threshold line defined by Equation (12), then the problem is deemed to be regular; otherwise, it is treated as irregular. It is worth highlighting that the use of these equations in a validation set, i.e., another 31 executions of the 90 problems (50 regular and 40 irregular ones), presented a success rate in predicting the type of PF, of 76% for regular problems and of 81% for irregular problems.

The second equation aims to determine the value of the parameter ρ, regardless of the number of objectives of an MOP or an MaOP. Three steps were taken to achieve this goal. First, for each number of objectives considered (3, 5, 8, 10, 15), we determine two medians of the Spread Index (SI) concerning the POFs of all test functions. We run each test function 31 times. They were divided into two groups, namely regular and irregular POFs, and each point in Figure 9.(a) represents an SI median for a particular number of objectives. Then, each SI median is associated with the best value of ρ (Figure 7) found by experimental design as shown in Figure 9.(b). Finally, we construct the equation used to determine the ρ parameter for both regular and irregular estimated POF problems by interpolating the values of the selected representatives as follows:

$$\rho(SI) = -0.4707 * (SI)^3 + 0.8644 * (SI)^2 - 0.1508 * (SI) + 0.05745 \tag{13}$$

where SI is the Spread Index. Figure 9.(b) presents the values of the parameter ρ within the interval [0,1]. In general, we can observe a growing trend of ρ associated with the growth of the SI of the MOPS.



(a) Median of 31 runs of the Spread Index on regular and irregular problems.

(b) Values of the ρ parameter according to the Spread Index.

Figure 9. Specification of the parameter value ρ regardless of the number of objectives of the MaOP.

We expect that the estimation of the parameter ρ can be used for different classes of problems such as real-valued functions, real-world and combinatorial problems. The validity of making this estimation is supported by the results presented in Subsection 4.4.

3.8. *Framework of the Algorithm*

The MOEA/D-UR pseudo code is presented in Alg. 6. Initially, the algorithm initializes the population $\boldsymbol{P}$, the set of weight vectors $\boldsymbol{\lambda}$, the neighborhood structure $\boldsymbol{B}(i)$ of each weight vector λ^i, and the ideal reference point z^* (lines 1-7). In each generation, a descendant is generated for each weight vector, using neighbor individuals, using the SBX crossover and polynomial mutation (lines 8-17). Within the neighborhood of the descendant, the current solutions of the weight vectors in which the TCH decomposition function has been minimized are updated (lines 18-23). After each generation, the external population (EP) (lines 24 and 25) and weight vectors (lines 26 and 27) may be updated. The MOEA/D-UR stops when the maximum number of generations is reached.

The maximum computational complexity of the MOEA/D-UR is defined as the sum of the time cost of the MOEA/D [74] (Alg. 6, lines 1-23), plus the maintenance of the $\boldsymbol{EP}$ (Alg. 6, line 25), and plus the weights adaptation (Alg. 6, line 27): $O(mTN) + O(mN^2) + O(mTN^2)$, where m is the number of objectives, T is the neighborhood size of the subproblems, and N is the population size. Therefore, the computational complexity is expressed as $O(mTN^2)$. The runtime comparison of MOEA/D-UR with the other MaOEAs can be found in Table 1 of the supplementary document. Moreover, the performance of HV versus the number of function evaluations on some problems can be found in Figure 1 of the supplementary document.

Algorithm 6: MOEA/D-UR

Initialize the population $\boldsymbol{P}$ and a weight vectors set $\boldsymbol{\lambda}$;
Apply the WS-transformation on the weight vectors $\boldsymbol{\lambda}$;
Determine the neighbors $\boldsymbol{B}(i)$ of each $\boldsymbol{\lambda}^i \in \boldsymbol{\lambda}$;
Calculate the reference point z^* according to $\boldsymbol{P}$;
$\boldsymbol{EP} \leftarrow \emptyset$;
$start \leftarrow 0.2$; $end \leftarrow 0.9$;
$Gen \leftarrow 0$;
while $Gen < Gen_{max}$ **do**
    $\boldsymbol{O} \leftarrow \emptyset$;
    for *each* $i \in \{1 \ldots N\}$ **do**
        if $uniform(0, 1) < \delta$ **then**
            $\boldsymbol{E} \leftarrow \boldsymbol{B}(i)$;
        else
            $\boldsymbol{E} \leftarrow \{1 \ldots N\}$;
        Randomly select mating solutions from $\boldsymbol{E}$ to generate an offspring $\bar{\boldsymbol{x}}$, Evaluate $\boldsymbol{F}(\bar{\boldsymbol{x}})$;
        $z^* \leftarrow min(z^*, \boldsymbol{F}(\bar{\boldsymbol{x}}))$;
        $\boldsymbol{O} \leftarrow \boldsymbol{O} \cup \bar{\mathbf{x}}$;
        $update \leftarrow 0$;
        while $update < nr$ ***and*** $\boldsymbol{E} \neq \emptyset$ **do**
            $j \leftarrow$ Randomly select an index from $\boldsymbol{E}$;
            $\boldsymbol{E} \leftarrow \boldsymbol{E} \setminus j$;
            if $g^{TCH}(\bar{\boldsymbol{x}}|\boldsymbol{\lambda}^j, z^*) \leq g^{TCH}(\boldsymbol{x}^j|\boldsymbol{\lambda}^j, z^*)$ **then**
                $\boldsymbol{x}^j \leftarrow \bar{\boldsymbol{x}}$; $update++$;
    if $Gen/Gen_{max} \leq end$ **then**
        $\boldsymbol{EP} \leftarrow$ Update $\boldsymbol{EP}$ using Alg. 2;
    if $start \leq Gen/Gen_{max} \leq end$ **then**
        Perform update when required using Alg. 5;
    $Gen \leftarrow Gen + 1$;
return $\boldsymbol{P}$;

4. Validation of the Algorithm

A series of experiments to evaluate the performance of the proposed algorithm against peer competitors in solving MaOPs is performed. The experimental results assessed by chosen performance metrics are then analyzed. A more detailed summary of how this section is organized is as follows: First, the MaOEAs selected for comparison are presented; a description of real-world problems follows this. Then the experimental setup for the proposed algorithm and the competitors is given. Finally, the experimental results are reported and analyzed.

4.1. The Algorithms Chosen for Comparisons

The algorithms considered for comparison are presented in chronological order.

- **A-NSGA-III** [29] extends NSGA-III by adapting the reference points. This method consists of two operations: 1) deletion of each added reference point associated with an empty niche, and 2) the random addition of new reference points inside each reference point with a high niche count [3].
- **MOEA/D-AWA** [30] has an adaptive scheme to dynamically change the weights at the late stage of the optimization. Starting at 80% of the evolutionary process, weight vectors in the dense regions are periodically

removed and new weight vectors are generated in the sparse regions. To detect the dense regions and sparse regions an external population is maintained.

- **KnEA** [43] is a knee point-driven evolutionary algorithm for solving many-objective problems. In KnEA, solutions for the next generation are first chosen based on the non-dominance selection criterion, and then knee points are used as the secondary selection criteria. The preference over knee points can approximate a bias towards a larger HV (hypervolume), and thus this accounts for convergence and diversity [75].

- **RVEA** [31] applies a framework similar to that of the NSGA-II algorithm, from which RVEA adopts an elitism strategy, where the offspring population is generated using traditional genetic operators. Then, the offspring and parent populations are combined to undergo an elitism selection. RVEA uses the reference vector guided selection and the reference vector adaptation. The former is responsible for using the reference vectors to partition the objective space into a number of subspaces in which the selection is performed separately while the latter deals with objective functions that are not well normalized. Furthermore, RVEA applies a decomposition approach called the angle penalized distance (APD) to dynamically balance the convergence and diversity of solutions in high-dimensional objective space [75].

- **NMPSO** [44] is a multi-objective particle swarm optimizer (MOPSO) that is suitable for solving MaOPs. NMPSO uses balanceable fitness estimation (BFE) and two more operators. This BFE method combines the convergence and diversity distances to decrease the effect of the curse of dimensionality in MaOPs and to guide all the particles to approach the POF. The first operator, an evolutionary search on the external archive, can provide another search pattern and overcome the ineffectiveness of a particle swarm optimizer (PSO)-based search on certain types of MaOPs. The second operator updates the velocity equation to change the search direction to the PSO-based search and to induce more diversity.

- **MOEA/D-URAW** [34] is a multi-objective evolutionary algorithm based on decomposition that uses the UR initialization method combined with weights adaptation in order to obtain both flexible population size and better adapted final solution sets. The sparsity level determines the solutions that are removed and added.

- **AR-MOEA** [3] is an IGD-NS indicator-based evolutionary algorithm with reference point adaptation which is used to solve both MOPs and MaOPs considering various types of Pareto fronts. In order to improve the versatility of the proposed AR-MOEA, a parameterless reference point adaptation method adjusts the reference points at each generation so as to calculate the indicator. In the adaptation scheme, the reference points are altered using an initialized reference set point together with candidate solutions stored in an external archive based on their contributions to the IGD-NS indicator.

- **PREA** [14] is a region-based many-objective evolutionary algorithm with a diversity maintenance mechanism using parallel distance to handle MOPs and MaOPs. In such an algorithm, a ratio-based indicator I_{∞}^{r} is used to form promising regions in the objective space to discard individuals with poor fitness and to select fit candidate solutions. Moreover, a diversity maintenance mechanism, based on the parallel distance, is used to further enhance the diversity of the population. The combination of these two strategies effectively ensures the diversity of the results obtained.

- **FDEA-I** and **FDEA-II** [45] are MaOEAs proposed on the basis of the fractional dominance relation and the improved objective space decomposition strategy. The two algorithms first use the fractional dominance relation to retain some solutions with a promising performance, and then the improved strategy for decomposing the objective is used to maintain diversity for the population obtained. The difference between FDEA-I and FDEA-II lies in the way solutions with sound convergence performance are selected. In FDEA-I, the fractional dominance factor and the fractional dominance score are applied to keep the convergence performance of the new population. In FDEA-II, the fractional dominance set is used to select solutions with effective convergence.

4.2. Real-world Applications

This Subsection introduces the problems solved by the algorithms that are compared: a multi-objective traveling salesman, a multi-objective knapsack, and one for water resources planning.

The mathematical formulation of the multi-objective traveling salesman problem is as follows: [40]:

$$\text{minimize} \sum_{i=1}^{n-1} c^k_{R(i),R(i+1)} + c^k_{R(n),R(1)} \tag{14}$$

where, given a set of n cities and a cost c^k_{ij}, where $k = 1, 2, ..., p$ (trips from city i to j). The main purpose of this problem is to find a city route, i.e., a cyclic permutation R of n cities. In this paper, the number of objectives, p, is defined as 3, 5, 8, 10, and 15; and the number of decision variables, n, is 30 for all cases.

The multi-objective knapsack problem 0/1 is a combinatorial optimization problem. In the 0/1 multi-objective knapsack problem, there are q knapsacks and a set of n items. The goal of this problem is to find the vector $\boldsymbol{z} = [z_1\ z_2\ ...\ z_n]^T \in \{0, 1\}^n$, where $z_j = 1$ or $z_j = 0$ denote that the item j is or is not in the knapsack, respectively. The mathematical formulation of the multi-objective knapsack problem 0/1 is defined as follows [40]:

$$\begin{aligned} \forall k \in \{1, 2, ..., q\} : \text{maximize } f_k(\boldsymbol{z}) = \sum_{j=1}^{n} p_{kj} \times z_j \\ \text{subject to} \sum_{j=1}^{n} w_{kj} \times z_j \leq c_k \end{aligned} \tag{15}$$

where p_{kj} is the profit of item j placed in the knapsack k, w_{kj} is the weight of item j placed in the knapsack k, and c_k is the capacity of the knapsack k. z_j is the j item selected in the knapsack. According to [40], the values of p_{kj} and w_{kj} are within the range [10, 100].

The problem of water resources planning [42] is a relevant issue in the domain of environmental engineering which consists of planning for stormwater drainage systems in urban areas. Any particular sub-basin within a river basin, taken as hydrologically independent of other sub-basins, has its own treatment plant, an on-site detention storage facility, a drainage network, and a range of water sources. The goal of the problem is to examine the sub-basin in terms of its stormwater drainage needs. There are three project variables x_1, x_2, and x_3; where x_1 represents the local holding storage capacity (basin × inches), x_2 represents the maximum treatment rate (basin × inches / hour) and x_3 represents the maximum tolerable overflow rate (basin × inches / hour).

The optimization problem is to find the values of the decision variables so that five objectives are simultaneously achieved, namely the following costs are minimized: Drainage network cost, $f_1(\boldsymbol{x})$; storage installation cost, $f_2(\boldsymbol{x})$; treatment facility cost, $f_3(\boldsymbol{x})$; expected cost of flood damage, $f_4(\boldsymbol{x})$; expected economic loss due to flooding, $f_5(\boldsymbol{x})$. Seven constraints ($g_1(\boldsymbol{x})$ to $g_7(\boldsymbol{x})$) are also considered:

$$\begin{aligned} &\text{minimize} \\ &\quad f_1(\boldsymbol{x}) = 106780.37(x_2 + x_3) + 61704.67 \\ &\quad f_2(\boldsymbol{x}) = 3000x_1 \\ &\quad f_3(\boldsymbol{x}) = (305700)2289x_2/(0.06 \times 2289)^{0.65} \\ &\quad f_4(\boldsymbol{x}) = (250)2289exp(39.75x_2 + 9.9x_3 + 2.74) \\ &\quad f_5(\boldsymbol{x}) = 25[1.39/(x_1x_2) + 4940x_3 - 80] \end{aligned} \tag{16}$$

$$\begin{aligned} &\text{subject to:} \\ &\quad g_1(\boldsymbol{x}) = 0.00139/(x_1x_2) + 4.94x_3 - 0.08 - 1 \leq 0 \\ &\quad g_2(\boldsymbol{x}) = 0.000306/(x_1x_2) + 1.082x_3 - 0.0986 - 1 \leq 0 \\ &\quad g_3(\boldsymbol{x}) = 12.307/(x_1x_2) + 49408.24x_3 + 4051.02 - 50000 \leq 0 \\ &\quad g_4(\boldsymbol{x}) = 2.098/(x_1x_2) + 8046.33x_3 - 696.71 - 16000 \leq 0 \\ &\quad g_5(\boldsymbol{x}) = 2.138/(x_1x_2) + 7883.39x_3 - 705.04 - 10000 \leq 0 \\ &\quad g_6(\boldsymbol{x}) = 0.417/(x_1x_2) + 1721.26x_3 - 136.54 - 2000 \leq 0 \\ &\quad g_7(\boldsymbol{x}) = 0.164/(x_1x_2) + 631.13x_3 - 54.48 - 550 \leq 0 \end{aligned} \tag{17}$$

where, $0.01 \leq x_1 \leq 0.45$, $0.01 \leq x_2 \leq 0.10$, and $0.01 \leq x_3 \leq 0.10$.

Table 1. Settings of the number of the objectives and decision variables for each test problem.

Benchmark Problem	*Objectives (m)*	*Variables (d)*	Pareto front
Regular Pareto front			
DTLZ1	3,5,6,8,10,12,15	m-1+5	Linear
DTLZ2-4	3,5,6,8,10,12,15	m-1+10	Concave
WFG4-9	3,5,6,8,10,12,15	m-1+10	Concave
Irregular Pareto front			
DTLZ5-6	3,5,6,8,10,12,15	m-1+10	Degenerate
DTLZ7	3,5,6,8,10,12,15	m-1+20	Disconnected
IDTLZ1	3,5,6,8,10,12,15	m-1+5	Inverted
IDTLZ2	3,5,6,8,10,12,15	m-1+10	Inverted
WFG1	3,5,6,8,10,12,15	m-1+10	Sharp tails
WFG2	3,5,6,8,10,12,15	m-1+10	Disconnected
WFG3	3,5,6,8,10,12,15	m-1+10	Degenerate
Real-World Problem			
MOTSP	3,5,8,10,15	30	Real-world
MOKP	3,5,8,10,15	250	Real-world
WRP	5	3	Real-world
Many-objective Optimization Problems Benchmark			
MaOP1	3,5	10	Inverted
MaOP2	3,5	10	Extreme convexity
MaOP3	3,5	10	Bias
MaOP4	3,5	10	Bias
MaOP5	3,5	10	Disconnected
MaOP6	3,5	10	Disconnected

4.3. Experimental Setup

We have chosen 27 test problems, namely WFG1-WFG9 [37], DTLZ1-7 [38], IDTLZ1-2 [29], MaOP1-6 [39] and the real-world problems MOTSP [40], MOKP [41], and WRP [42] to run the comparative tests. Table 1 presents the number of objectives, the number of decision variables, and the PF shape for each test problem. The maximum number of function evaluations before terminating every run is set to 60,000. The population is formed by 120 individuals for all test functions, except for the MaOEAs that use the Das and Dennis method for generating weights (A-NSGA-III, MOEA/D-AWA, RVEA, FDEA-I, and FDEA-II) in problems with 5, 6, 10, and 12 objectives. For such numbers of objectives, the population size is defined as 105, 112, 110, and 90 individuals, respectively.

The proposed algorithm was implemented using the MATLAB-based PlatEMO platform [64] in which the codes for the ten MaOEAs which were used for comparison were already available [1]. A description of each of the MaOEAs is given in Subsection 4.1. The specific reference and parameter settings of each algorithm are shown in Table 2. The values of parameters of the variation operators are shown in Table 3. All experiments were run in a computer with 16 gigabytes of RAM and a 3.60GHz 6-core Intel Core i5-8400 processor.

The experimental design technique LHS [71], Subsection 3.7, was chosen to determine the four MOEA/D-UR parameters that are used for adaptation: *start* and *end* define the interval for which the evolutionary process starts and stops verifying the condition for weight vector adaptation; *period* determines the number of generations between verifications, and *nus* defines the number of subproblems for updating. A fifth parameter introduced in MOEA/D-UR is used as a threshold to perform the adaptation process, ρ. It is adptatively determined whenever MOEA/D-UR runs.

[1]The source code of the FDEA-I and FDEA-II are available on https://github.com/P-N-Suganthan.

The LHS was employed to generate 500 tuples for tuning the four parameters above. The minimum and maximum values informed to the LHS method were 0 to 0.3, 0.7 to 1, 1 to 100, and 0.01 to 0.3 for the parameters *start*, *end*, *period*, and *nus*. The functions used to run the LHS were DTLZ1-4 and WFG4-9 for the regular PFs and DTLZ5-7, IDTLZ1-2, and WFG1-3 for the irregular PFs having 3, 5, 8, 10 and 15 objectives. For the regular PF shapes, the best tuple was 0.2, 0.9, 12, and 0.25 whereas for the irregular PFs, the best tuple was 0.2, 0.9, 28, and 0.075. it is worth mentioning that the parameter ρ is determined by two equations depending on the type of the PF, regular or irregular. The paper introduces a way for estimating on the fly the level of regularity of a PF (details in Subsection 3.7). Finally, the remaining parameters, neighborhood size (T), δ, and nr, had their values extracted from the common values used in the literature, thus, they were not determined by any experimental design method in our work.

Table 2. The ten selected algorithms. N is the population size and m is the number of objectives.

Reference	*Algorithm*	*Parameters of the PlatEMO*
[29]	A-NSGA-III	*start*: 0.13, *end*: 1.00, *period*: 34
[30]	MOEA/D-AWA	*start*: 0.01, *end*: 0.87, *period*: 37, *nus*: 0.07, T: 0.1, δ: 0.9, *nr*: 2
[43]	KnEA	rate: 0.5
[31]	RVEA	*start*: 0.18, *end*: 0.86, *period*: 26, α: 2
[44]	NMPSO	ω: [0.1, 0.5], c_1, c_2, c_3: [1.5, 2.5]
[34]	MOEA/D-URAW	*start*: 0.05, *end*: 0.87, *period*: 30, *nus*: 0.05, T: 0.1, δ: 0.9, *nr*: 2
[3]	AR-MOEA	*start*: 0.06, *end*: 0.83, *period*: 2
[14]	PREA	P_s: 0.7
[45]	FDEA-I	*start*: 0.3, *end*: 0.72, *period*: 94, $\alpha(a, b)$: 0.1, 0.4
[45]	FDEA-II	*start*: 0.25, *end*: 0.75, *period*: 48, $\alpha(a, b)$: 0.1, 0.4, θ: 0.6
	MOEA/D-UR	T: 0.1, δ: 0.9, *nr*: 2, k: 10

For a fair comparison, the same procedure was employed for tuning the parameters of all algorithms that perform an adaptation process. We have used the LHS method to determining the best parameter tuple for A-NSGA-III, MOEA/D-AWA, RVEA, MOEA/D-URAW, AR-MOEA, FDEA-I, and FDEA-II. Thus, 500 tuples of parameters were generated for *start*, *end*, *period*, and *nus*; in the same ranges informed above, remembering that only MOEA/D-AWA and MOEA/D-URAW need the *nus* parameter, whereas for the other models this parameter is not applied. Selected parameters can be viewed in Table 2. The values for the remaining parameters were collected from the literature as we did for MOEA/D-UR.

The performance of each algorithm is evaluated through two widely-used metrics, namely the inverted generational distance (IGD) [76] and the hypervolume (HV) [41]. We normalize each objective according to the ideal point and the nadir point of the PF in order to exclude the impact of different objective scales on how these two metrics are calculated.

IGD: Let $\boldsymbol{P^*}$ be a set of uniformly distributed reference points on the PF and $\boldsymbol{P}$ be the set of solutions. The IGD value of $\boldsymbol{P}$ can be defined as follows:

$$IGD(\boldsymbol{P}, \boldsymbol{P^*}) = \frac{\sum_{\boldsymbol{v} \in \boldsymbol{P^*}} d(\boldsymbol{v}, \boldsymbol{P})}{|\boldsymbol{P^*}|} \tag{18}$$

where $d(\boldsymbol{v}, \boldsymbol{P})$ is the Euclidean distance from point $\boldsymbol{v} \in \boldsymbol{P^*}$ to its nearest point in $\boldsymbol{P}$. $|\boldsymbol{P^*}|$ is the cardinality of $\boldsymbol{P^*}$. The computational experiments use 10,000 reference points [64]; only for the water resource planning problem (WRP) the reference set contains 2429 solutions [77]. The smaller the IGD, the better the quality of $\boldsymbol{P}$ for approximating the whole PF.

HV: Given a reference point $\boldsymbol{z}^r = (z_1^r \dots z_n^r)^T$ dominated by all Pareto-optimal solutions, the HV of a set of solutions $\boldsymbol{P}$ is defined as the volume of the objective space dominated by all solutions in $\boldsymbol{P}$, bounded by z^r:

$$HV(\boldsymbol{P}, \boldsymbol{z}^r) = Vol(\bigcup_{\boldsymbol{p} \in P} [f_1^{\boldsymbol{p}}, z_1^r] \times \dots \times [f_m^{\boldsymbol{p}}, z_m^r]), \tag{19}$$

where $Vol(\cdot)$ denotes the Lebesgue measure. The larger HV, the better the approximation quality of $\boldsymbol{P}$. We use a reference point 10% higher than the upper bound of the PF for our experiments. To reduce the computational complexity of determining HV if $m > 4$, we use the Monte Carlo method with 1,000,000 sampling points to approximate its value.

For each problem, 30 independent tests were run for each algorithm, and the mean and standard deviation of each metric value is recorded. The Wilcoxon rank-sum test with a significance level of 0.05 is adopted to perform statistical analysis on the experimental results, where the symbols "+", "-", and "≈" indicate that the result by another MaOEA is significantly better, significantly worse, and statistically similar to that obtained by MOEA/D-UR [64].

Table 3. Settings of the reproduction operators, d is the number of decision variables)

Parameters	Values
Multi-Objective Knapsack Problem	
Crossover	One Point
Mutation	Bitwise
Multi-Objective Traveling Salesman Problem	
Crossover	Order
Mutation	Slight
Continuous Problems	
Crossover	Simulated Binary
P_c	1.00
η_c	20
Mutation	Polynomial
P_m	1/d
η_m	20

4.4. Results

Tables 4 to 11 present the experimental results reached by the MOEA/D-UR and ten state-of-the-art MaOEAs (Section 4.1) used for solving the problems DTLZ1-DTLZ7, IDTLZ1-IDTLZ2, WFG1-WFG9, MaOP, MOTSP, MOKP, and WRP with 3, 5, 8, 10, and 15 objectives (Tables 4 to 8, 10, and 11) and 6 and 12 objectives (Table 8 and Table 9). The standard deviations are presented in Tables 2-9 of the supplementary document.

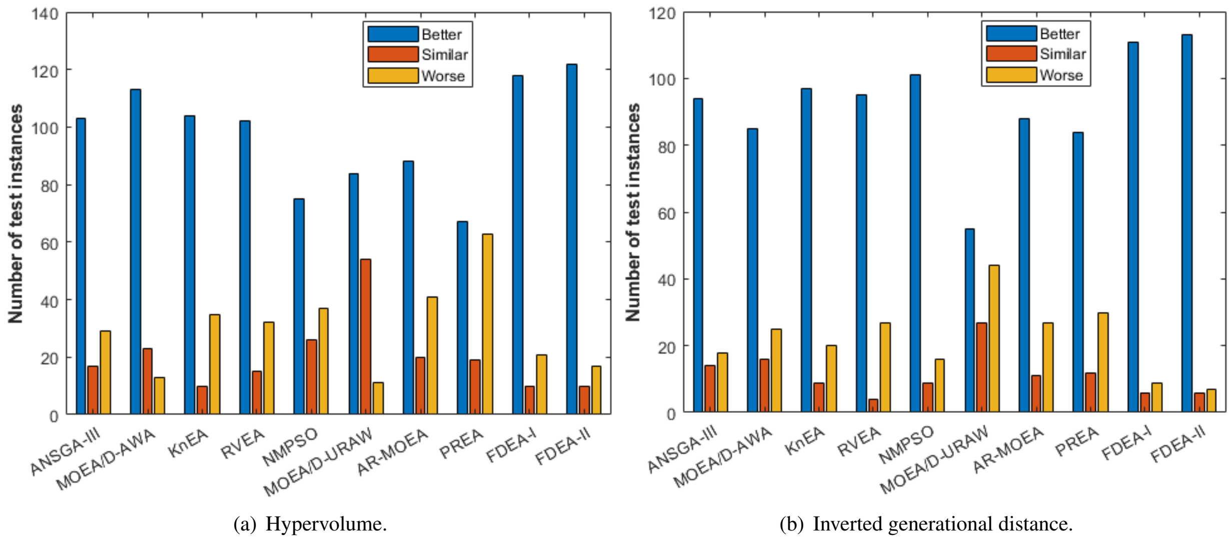


Figure 10. Wilcoxon signed-rank test at a 5% significance with respect to HV and IGD metrics between MOEA/D-UR and the state-of-the-art MaOEAs.

The resultant HV and IGD values are derived from two groups of experiments, considering the number of objectives. The first one mainly aims to validate the algorithm while the second set of experiments aims mostly to assess the generalization capacity of the algorithm. First, the tests involve regular (Table 4 and Table 5) and irregular (Table 6 and Table 7) problems with 3, 5, 8, 10, and 15 objectives. Such experiments were also used to construct the equations that define the parameters of the variables (Section 3.7). Secondly, the results concern regular and irregular problems with 6 and 12 objectives (Table 8 and Table 9), PF shapes not previously estimated by the model with 3 and 5 objectives (Table 10), and real-world problems with 3, 5, 8, 10, and 15 objectives (Table 11).

Table 4 shows results for MaOPs considering benchmark functions with regular POFs. For such tests, the MOEA/D-UR reached the best overall performance in 14 out of 50 instances. The POFs of these problems are characterized by having linear and concave shapes. The results suggest that the proposed model deals well with the DTLZ problem class, especially with the DTLZ1 and DTLZ3 when POFs share linear and concave features, respectively. Also, the AR-MOEA and the PREA performed better than the other algorithms for the WFG problem class. Particularly, the MOEA/D-UR presented difficulties in the WFG5 and the WFG9; although both have a concave POF shape, both problems have deceptive properties in PS. The IGD results (Table 5) indicate that the MOEA/D-UR performed better than all competitors but the MOEA/D-URAW reached the best overall results in this class of problems. Figure 11 illustrates the approximation of the POFs of the algorithms considered for the three-objective DTLZ1 problem.

Table 6 shows results for MaOPs considering benchmark functions with irregular POFs. For these tests, the MOEA/D-UR reached the best overall performance in 11 out of 40 instances. The shapes of the POFS in such problems are degenerate, disconnected, inverted, and sharp-tailed. The performance was favorable, in particular, for the DTLZ5-6 problems both of which feature degenerate POF shapes. Table 7 shows the IGD results. Note that the MOEA/D-UR did well in the IDTLZ2 problem under the IGD metric. This indicates that the model determined an appropriate distribution throughout the POF. Figures 12-14 exemplify the approximation of the POFs of the algorithms compared for the three-objective IDTLZ1, DTLZ7, and WFG3 problems.

Table 8 and Table 9 present the results of the MOEA/D-UR in regular and irregular problems with 6 and 12 objectives. In this scenario, we aimed to assess the generalization capacity of the model to determine the parameter ρ for problems with numbers of objectives that we had not previously used (Equation (13)). The MOEA/D-UR achieved the best overall performance in 6 out of 36 instances under the HV metric. It did better than all other MaOEAs except the PREA. This algorithm keeps on achieving the best performance in problems of the WFGs class whereas the MOEA/D-UR continues to present difficulties in tackling the problems with deceptive PS and with the IDTLZ2 problem with a POF which has an inverted concave shape.

Table 10 presents the results of the MOEA/D-UR for MaOP benchmark problems with 3 and 5 objectives. In this scenario, the expressions for determining the degree of regularity of the POF, spread index, and threshold, are validated for MaOPs with PF shapes not previously estimated. The MOEA/D-UR achieved the best overall performance in 4 out of 12 instances. The results suggest that the MOEA/D-UR reached a performance that was statistically superior to all other MOEAs (two-by-two comparisons) even though NMPSO did better than all others for MaOP3-4 problems; both have a biased POF shape. An important point is that the performance of the MOEA/D-UR was better on the MaOP5-6, an instance of disconnected shape problems with a high correlation among objectives, because the mechanism for detecting the degree of regularity of the MOP triggers suitable adaptation.

Table 11 shows the MOEA/D-UR results for real-world problems. In this scenario, the use of the MOEA/D-UR molded-in regular and irregular continuous optimization problems is extended to real-world problems. The MOEA/D-UR achieved the best overall performance in 9 out of 11 instances. The results suggest that the performance of the MOEA/D-UR surpasses that of all other MaOEAs. The results suggest that the MOEA/D-UR is promising for combinatorial problems, i.e., MOKP and MOTSP. The results in combinatorial problems suggest the performance of the MOEAs based on decomposition is superior. It is worth noting that all problems were categorized as irregular by the MOEA/D-UR. Also, the MOEA/D-UR does not adapt at all moments in combinatorial problems, which explains why it has better results than other MOEAs that have adapted in an unrestricted mode. Regarding the continuous problem of the real world, WRP, the MOEA/D-UR was one of the MaOEAs that achieved the best performance, especially for the metric of IGD presented in Table 10 of the supplementary document. The sparsity level contributed to the population being well distributed throughout the POF.

Figure 10 presents a summary of the results of the overall test instances for both the HV and IGD metrics. The terms better, similar, and worse stand for the number of cases in which the MOEA/D-UR performs better, similar, and worse with respect to the algorithms that were compared. From Figure 10.(a), we can observe that the MOEA/D-UR outperforms all MaOEAs numerically. The PREA results are the closest to those of the MOEA/D-UR. The PREA performed well on regular problems, while the MOEA/D-UR did well in irregular and real-world problems. We can also observe from Figure 10.(b) that the number of performances that the MOEA/D-UR had that went better than those of the state-of-the-art algorithms compared. The MOEA/D-URAW performance was the closest to that one of MOEA/D-UR. This is mainly because the MOEA/D-UR is an improved version from the HV perspective of the MOEA/D-URAW. Thus, the optimal periodicity for the HV is not necessarily optimum for the IGD metric. In sum, the MOEA/D-UR was shown to be more effective for the types of problems considered because it has a sensor that can estimate the features based on the dynamics of the evolutionary processing. Thus, the algorithm can adapt itself to respond to the features detected.

5. Conclusion

In this paper, we have proposed the MOEA/D with Update when Required (MOEA/D-UR) for MaOPs. This model uses a new metric, a spread index (SI), to estimate the POF level of regularity. Also, when the weight vectors show signs of convergence, assessed by the improvement metric (IM), the adaptation process is triggered. Finally, a space division approach of the objectives to increase diversity is run. Despite the use of uniform random weights, sparsity-based weights adaptation - timely applied – enables a competitive performance to be achieved when compared with ten state-of-the-art MaOEAs in most of the 149 MaOPs and MOPs with different PF shapes.

The MOEA/D-UR is an improved version of the MOEA/D-URAW, from which it inherits favorable features such as allowing a flexible population size without losing performance on irregular POFs. However, in the MOEA/D-UR, the frequency with which weights are updated is not predetermined, but rather can be adapted to the convergence of individuals to weight vectors.

Despite the achievements, we can observe limitations in the MOEA/D-UR which raise possibilities for future lines of research:

1. The spatial division approach aims to insert non-dominated solutions, thus increasing the quality of the individuals in the population. During its execution, there is a stage where genetic operators are used. Investigating other ways of using these operators can improve the diversity of this approach.
2. The use of a spread index and an improvement metric was tested for some tens of problems; therefore, extending the SI and IM indices to other MOEAs and MaOPs can shed more light on this type of adaptation. For example, in the recently proposed A-GWASF-GA [23], the weights vectors are updated by projecting from the ideal and nadir points. Knowing the degree of regularity of the POF can indicate the parameters appropriate to the MOP. Also, proving the convergence in the population by using the improvement metric can trigger adapting the population associated to a specific reference point.

References

[1] H.-L. Liu, L. Chen, Q. Zhang, K. Deb, Adaptively allocating search effort in challenging many-objective optimization problems, IEEE Transactions on Evolutionary Computation 22 (3) (2018) 433–448.

[2] S. Mane, M. N. Rao, Many-objective optimization: Problems and evolutionary algorithms - a short review, International Journal of Applied Engineering Research 12 (20) (2017) 9774–9793.

[3] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Transactions on Evolutionary Computation 22 (4) (2018) 609–622.

[4] H. Ishibuchi, N. Akedo, H. Ohyanagi, Y. Nojima, Behavior of EMO algorithms on many-objective optimization problems with correlated objectives, in: 2011 IEEE Congress of Evolutionary Computation (CEC), IEEE, 2011, pp. 1465–1472.

[5] H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems, IEEE Transactions on Evolutionary Computation 19 (2) (2014) 264–283.

[6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.

[7] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, TIK-report 103 (2001).

[8] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Transactions on Evolutionary Computation 21 (3) (2017) 440–462.

[9] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842.

[10] J. Bader, E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization, Evolutionary Computation 19 (1) (2011) 45–76.

[11] K. Shang, H. Ishibuchi, M.-L. Zhang, Y. Liu, A new R2 indicator for better hypervolume approximation, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2018, pp. 745–752.

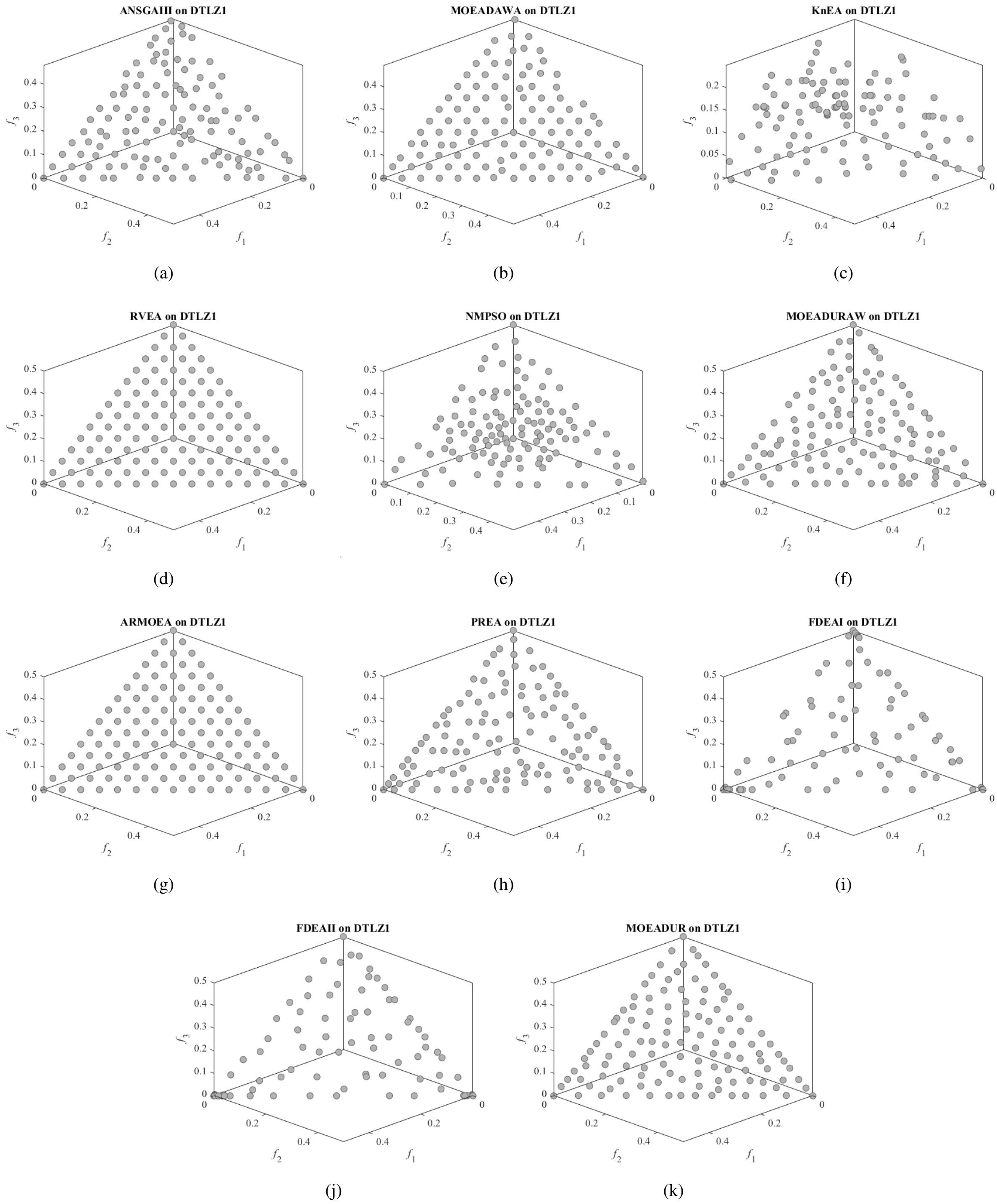


Figure 11. The set of final non-dominated solutions using the median HV metric values obtained by the algorithms for the three-objective DTLZ1 problem.

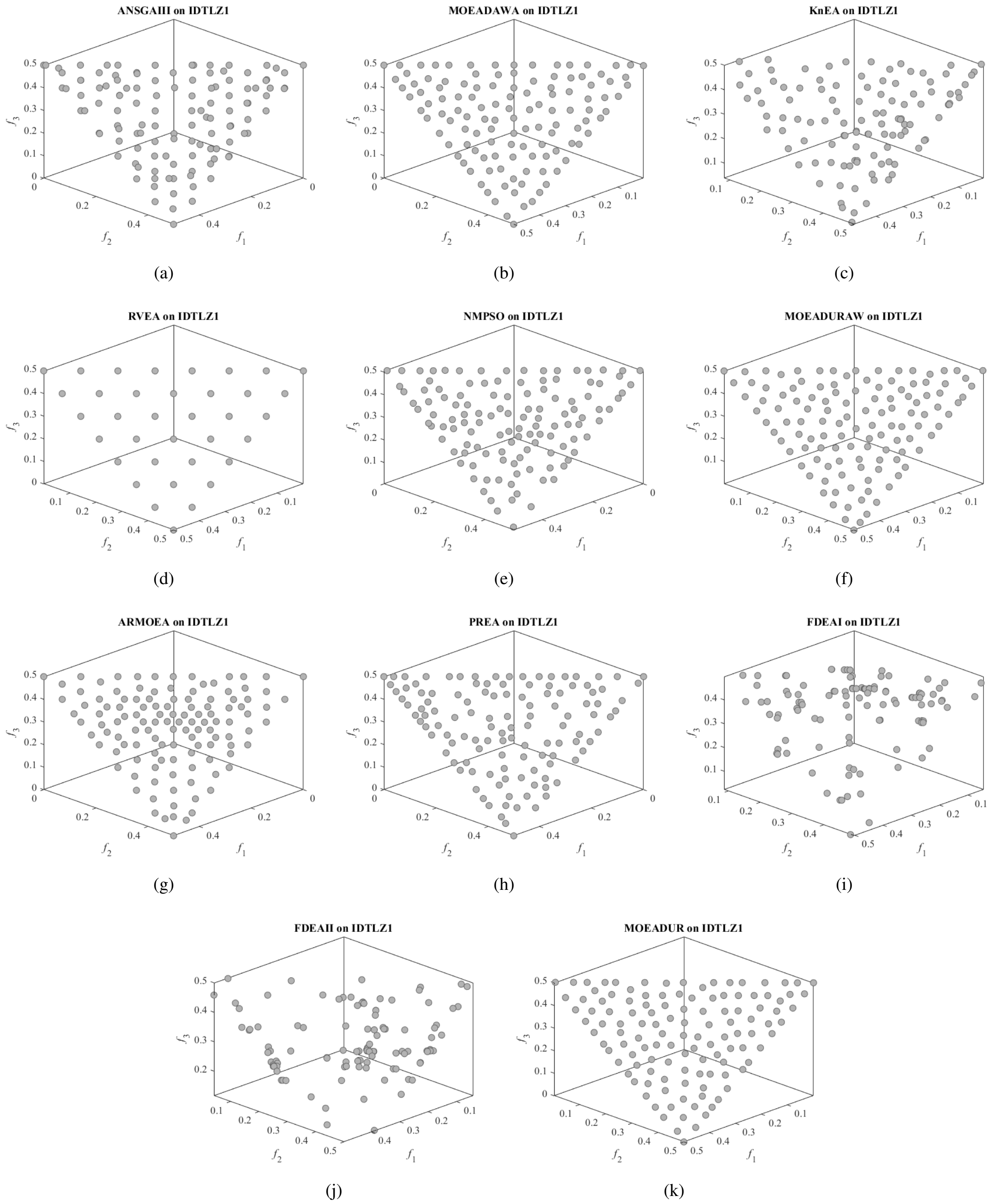


Figure 12. The set of final non-dominated solutions using the median HV metric values obtained by the algorithms for the three-objective IDTLZ1 problem.

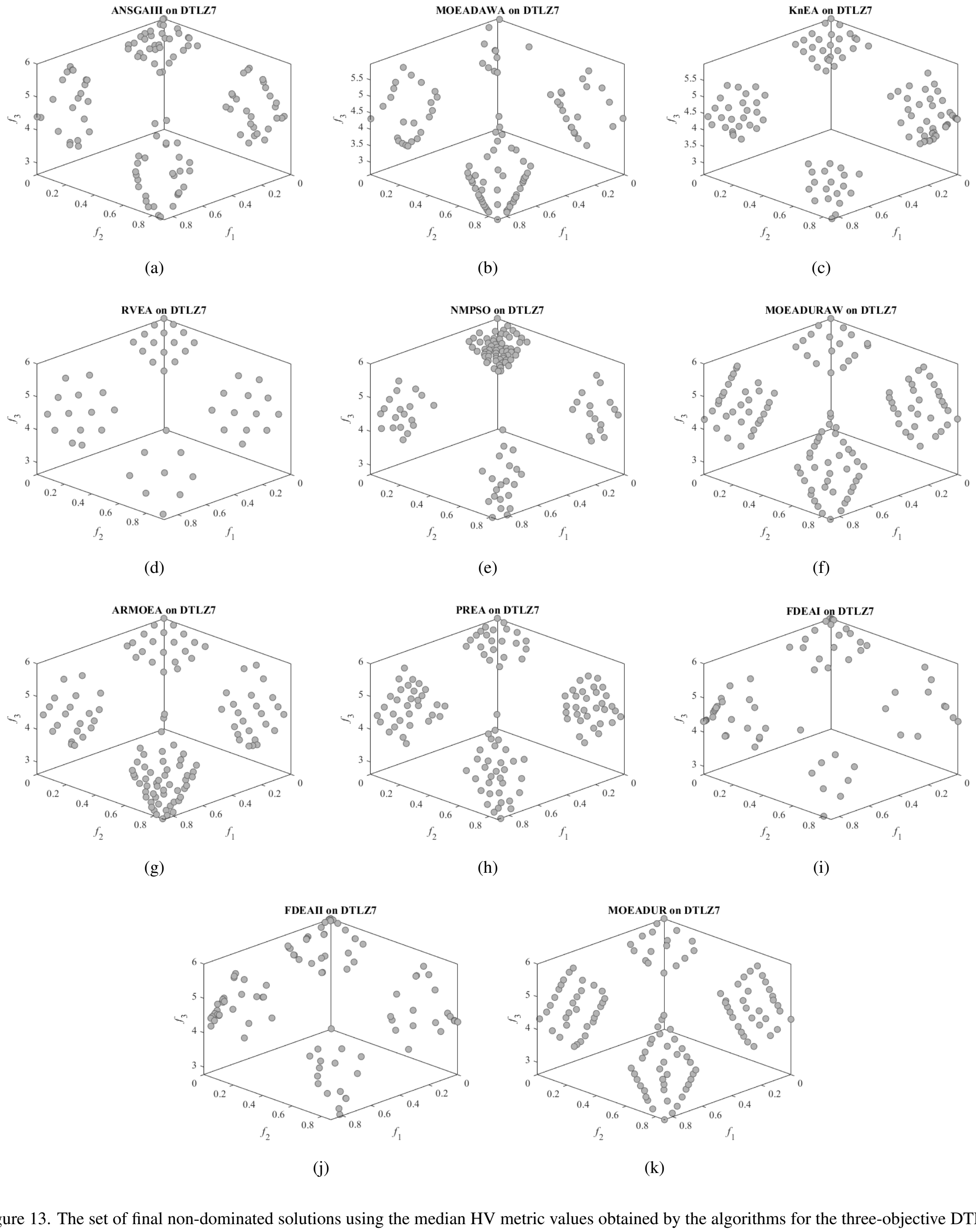


Figure 13. The set of final non-dominated solutions using the median HV metric values obtained by the algorithms for the three-objective DTLZ7 problem.

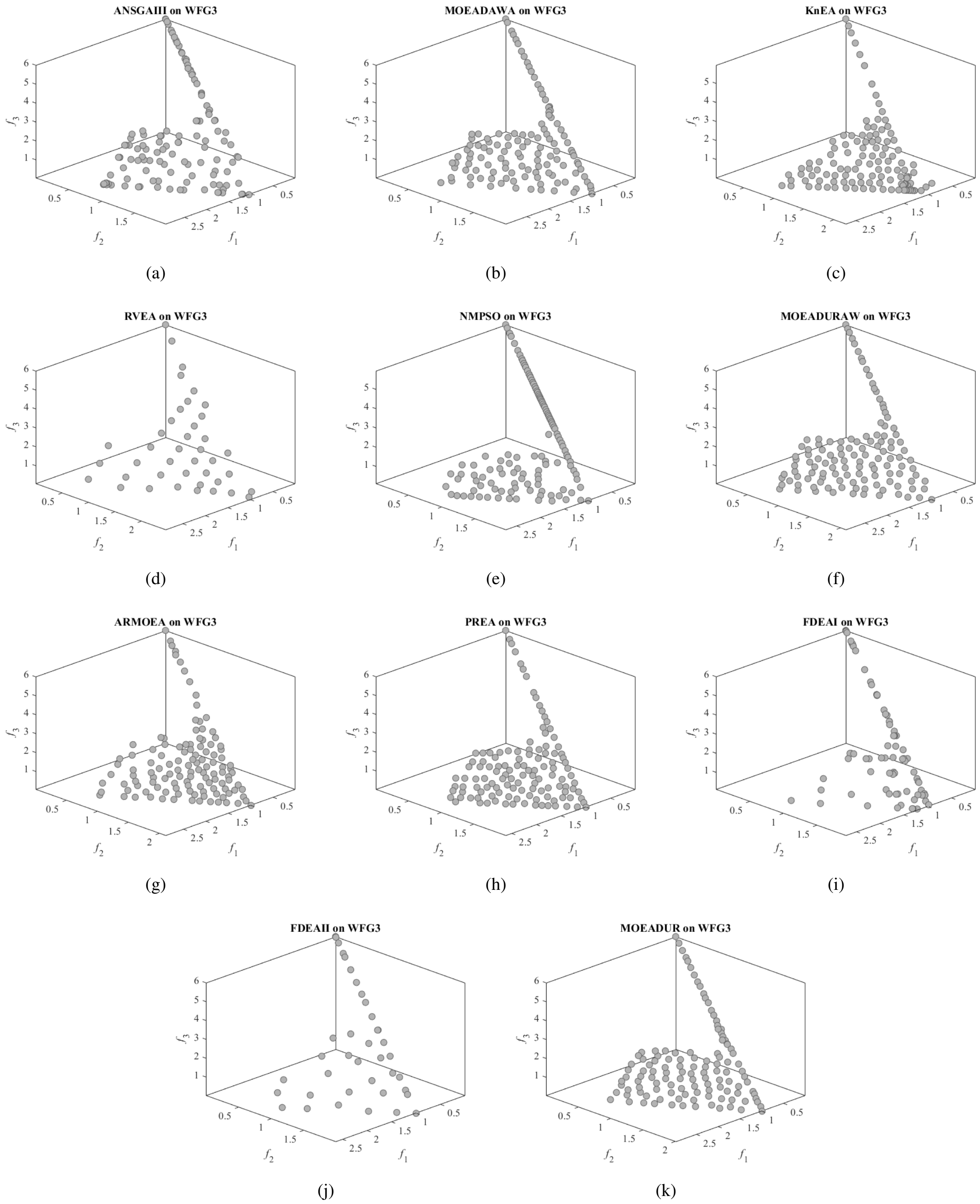


Figure 14. The set of final non-dominated solutions using the median HV metric values obtained by the algorithms for the three-objective WFG3 problem.

[12] B. Li, K. Tang, J. Li, X. Yao, Stochastic ranking algorithm for many-objective optimization based on multiple indicators, IEEE Transactions on Evolutionary Computation 20 (6) (2016) 924–938.
[13] J. G. Falcón-Cardona, C. A. C. Coello, Indicator-based multi-objective evolutionary algorithms: A comprehensive survey, ACM Computing Surveys (CSUR) 53 (2) (2020) 1–35.
[14] J. Yuan, H.-L. Liu, F. Gu, Q. Zhang, Z. He, Investigating the properties of indicators and an evolutionary many-objective algorithm based on a promising region, IEEE Transactions on Evolutionary Computation.
[15] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
[16] H. Wang, L. Jiao, X. Yao, Two_Arch2: An improved two-archive algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (4) (2014) 524–541.
[17] H. Ishibuchi, Y. Setoguchi, H. Masuda, Y. Nojima, Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes, IEEE Transactions on Evolutionary Computation 21 (2) (2017) 169–190.
[18] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, L. Jiao, On the use of two reference points in decomposition based multiobjective evolutionary algorithms, Swarm and Evolutionary Computation 34 (2017) 89–102.
[19] X. Cai, M. Hu, D. Gong, Y.-n. Guo, Y. Zhang, Z. Fan, Y. Huang, A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjective optimization, Swarm and Evolutionary Computation 49 (2019) 178–193.
[20] Z. Chen, Y. Zhou, X. Zhao, Y. Xiang, J. Wang, A historical solutions based evolution operator for decomposition-based many-objective optimization, Swarm and Evolutionary Computation 41 (2018) 167–189.
[21] Y. Qi, X. Li, J. Yu, Q. Miao, User-preference based decomposition in MOEA/D without using an ideal point, Swarm and Evolutionary Computation 44 (2019) 597–611.
[22] J. Luo, Y. Yang, X. Li, Q. Liu, M. Chen, K. Gao, A decomposition-based multi-objective evolutionary algorithm with quality indicator, Swarm and Evolutionary Computation 39 (2018) 339–355.
[23] M. Luque, S. Gonzalez-Gallardo, R. Saborido, A. B. Ruiz, Adaptive global $wasf - ga$ to handle many-objective optimization problems, Swarm and Evolutionary Computation 54 (2020) 100644.
[24] H.-L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE Transactions on Evolutionary Computation 18 (3) (2013) 450–455.
[25] M. Li, X. Yao, What weights work for you? Adapting weights for any pareto front shape in decomposition-based evolutionary multiobjective optimisation, Evolutionary Computation 28 (2) (2020) 227–253.
[26] M. Li, S. Yang, X. Liu, Pareto or non-pareto: Bi-criterion evolution in multiobjective optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2015) 645–665.
[27] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, 2009, pp. 203–208.
[28] J. Siwei, C. Zhihua, Z. Jie, O. Yew-Soon, Multiobjective optimization by decomposition with pareto-adaptive weight vectors, in: 2011 Seventh International Conference on Natural Computation, Vol. 3, IEEE, 2011, pp. 1260–1264.
[29] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach, IEEE Transactions on Evolutionary Computation 18 (4) (2013) 602–622.
[30] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evolutionary Computation 22 (2) (2014) 231–264.
[31] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2016) 773–791.
[32] M. Wu, S. Kwong, Y. Jia, K. Li, Q. Zhang, Adaptive weights generation for decomposition-based multi-objective optimization using gaussian process regression, in: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, 2017, pp. 641–648.
[33] M. Asafuddoula, H. K. Singh, T. Ray, An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors, IEEE Transactions on Cybernetics 48 (8) (2017) 2321–2334.
[34] L. R. de Farias, P. H. Braga, H. F. Bassani, A. F. Araújo, MOEA/D with uniformly randomly adaptive weights, in: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, 2018, pp. 641–648.
[35] H. Chen, G. Wu, W. Pedrycz, P. N. Suganthan, L. Xing, X. Zhu, An adaptive resource allocation strategy for objective space partition-based multiobjective optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems.
[36] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, L. Wang, Hyperplane assisted evolutionary algorithm for many-objective optimization problems, IEEE Transactions on Cybernetics 50 (7) (2019) 3367–3380.
[37] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506.
[38] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: Evolutionary Multiobjective Optimization, Springer, 2005, pp. 105–145.
[39] H. Li, K. Deb, Q. Zhang, P. N. Suganthan, L. Chen, Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties, Swarm and Evolutionary Computation 46 (2019) 104–117.
[40] D. W. Corne, J. D. Knowles, Techniques for highly multiobjective optimisation: Some nondominated points are better than others, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM, 2007, pp. 773–780.
[41] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.
[42] K. Musselman, J. Talavage, A tradeoff cut approach to multiple objective optimization, Operations Research 28 (6) (1980) 1424–1435.
[43] X. Zhang, Y. Tian, Y. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (6) (2015) 761–776.
[44] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C. A. C. Coello, K.-C. Wong, J. Zhang, Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems, IEEE Transactions on Evolutionary Computation 22 (1) (2016) 32–46.
[45] W. Qiu, J. Zhu, G. Wu, M. Fan, P. N. Suganthan, Evolutionary many-objective algorithm based on fractional dominance relation and improved

objective space decomposition strategy, Swarm and Evolutionary Computation 60 (2021) 100776.

[46] I. Das, J. E. Dennis, Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM Journal on Optimization 8 (3) (1998) 631–657.

[47] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Transactions on Evolutionary Computation 19 (5) (2015) 694–716.

[48] A. Jaszkiewicz, On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment, IEEE Transactions on Evolutionary Computation 6 (4) (2002) 402–412.

[49] K. Deb, H. Jain, 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) (2014) 577–601.

[50] F. Gu, H.-L. Liu, K. C. Tan, A multiobjective evolutionary algorithm using dynamic weight design method, International Journal of Innovative Computing, Information and Control 8 (5 (B)) (2012) 3677–3688.

[51] C. Zhu, X. Cai, Z. Fan, M. Sulaman, A two-phase many-objective evolutionary algorithm with penalty based adjustment for reference lines, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 2161–2168.

[52] H. Li, M. Ding, J. Deng, Q. Zhang, On the use of random weights in MOEA/D, in: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2015, pp. 978–985.

[53] R. Cheng, Y. Jin, K. Narukawa, Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2015, pp. 127–140.

[54] R. Wang, R. C. Purshouse, P. J. Fleming, Preference-inspired co-evolutionary algorithms using weight vectors, European Journal of Operational Research 243 (2) (2015) 423–441.

[55] A. Masood, Y. Mei, G. Chen, M. Zhang, A PSO-based reference point adaption method for genetic programming hyper-heuristic in many-objective job shop scheduling, in: Australasian Conference on Artificial Life and Computational Intelligence, Springer, 2017, pp. 326–338.

[56] K. S. Bhattacharjee, H. K. Singh, T. Ray, Q. Zhang, Decomposition based evolutionary algorithm with a dual set of reference vectors, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2017, pp. 105–112.

[57] C. Zhou, G. Dai, C. Zhang, X. Li, K. Ma, Entropy based evolutionary algorithm with adaptive reference points for many-objective optimization problems, Information Sciences 465 (2018) 232–247.

[58] X. He, Y. Zhou, Z. Chen, Q. Zhang, Evolutionary many-objective optimization based on dynamical decomposition, IEEE Transactions on Evolutionary Computation 23 (3) (2018) 361–375.

[59] Y. Liu, H. Ishibuchi, N. Masuyama, Y. Nojima, Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular pareto fronts, IEEE Transactions on Evolutionary Computation 24 (3) (2019) 439–453.

[60] Q. Liu, Y. Jin, M. Heiderich, T. Rodemann, Adaptation of reference vectors for evolutionary many-objective optimization of problems with irregular pareto fronts, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 1726–1733.

[61] F. Gu, Y.-M. Cheung, Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm, IEEE Transactions on Evolutionary Computation 22 (2) (2017) 211–225.

[62] M. Wu, K. Li, S. Kwong, Q. Zhang, J. Zhang, Learning to decompose: A paradigm for decomposition-based multiobjective optimization, IEEE Transactions on Evolutionary Computation 23 (3) (2018) 376–390.

[63] F.-Q. Gu, H.-L. Liu, A novel weight design in multi-objective evolutionary algorithm, in: 2010 International Conference on Computational Intelligence and Security, IEEE, 2010, pp. 137–141.

[64] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum], IEEE Computational Intelligence Magazine 12 (4) (2017) 73–87.

[65] X. Guo, X. Wang, Z. Wei, MOEA/D with adaptive weight vector design, in: 2015 11th International Conference on Computational Intelligence and Security (CIS), IEEE, 2015, pp. 291–294.

[66] S. Jiang, L. Feng, D. Yang, C. K. Heng, Y.-S. Ong, A. N. Zhang, P. S. Tan, Z. Cai, Towards adaptive weight vectors for multiobjective evolutionary algorithm based on decomposition, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 500–507.

[67] M. Pilát, R. Neruda, General tuning of weights in MOEA/D, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 965–972.

[68] K. Miettinen, Nonlinear multiobjective optimization kluwer academic publishers, Boston, Massachusetts.

[69] S. Kukkonen, K. Deb, A fast and effective method for pruning of non-dominated solutions in many-objective problems, in: Parallel Problem Solving from Nature - PPSN IX, Vol. 4193, 2006, pp. 553–562.

[70] S. Lloyd, Least squares quantization in pcm, IEEE Transactions on Information Theory 28 (2) (1982) 129–137.

[71] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, John Wiley & Sons, 2008.

[72] J. C. Helton, F. Davis, J. D. Johnson, A comparison of uncertainty and sensitivity analysis results obtained with random and latin hypercube sampling, Reliability Engineering & System Safety 89 (3) (2005) 305–330.

[73] N. J. Higham, Accuracy and Stability of Numerical Algorithms, Vol. 80, Siam, 2002.

[74] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 284–302.

[75] K. Li, R. Wang, T. Zhang, H. Ishibuchi, Evolutionary many-objective optimization: A comparative study of the state-of-the-art, IEEE Access 6 (2018) 26194–26214.

[76] P. A. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 174–188.

[77] J. J. Durillo, A. J. Nebro, jMetal: A java framework for multi-objective optimization, Advances in Engineering Software 42 (10) (2011) 760–771.

Table 4. HV mean values obtained by MaOEAs for regular PF problems with 3, 5, 8, 10, and 15 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
DTLZ1	3	8.4054e-1 −	8.4144e-1 −	7.6502e-1 −	8.4430e-1 −	8.3693e-1 −	8.4430e-1 ≈	8.4554e-1 +	8.4355e-1 −	8.0254e-1 −	8.0545e-1 −	8.4473e-1
	5	9.5789e-1 −	9.3651e-1 −	5.7170e-1 −	9.6767e-1 −	9.5481e-1 −	9.6765e-1 −	9.7148e-1 −	9.7276e-1 −	9.2527e-1 −	9.2867e-1 −	9.7360e-1
	8	9.5775e-1 −	9.6874e-1 −	1.0758e-1 −	9.9558e-1 ≈	7.9756e-1 −	9.8825e-1 −	9.9638e-1 ≈	9.9646e-1 ≈	9.8120e-1 −	9.6868e-1 −	9.9664e-1
	10	9.3223e-1 −	9.2417e-1 −	0.0000e+0 −	9.8925e-1 ≈	6.3734e-1 −	9.9010e-1 −	9.9538e-1 ≈	8.7771e-1 −	9.8763e-1 −	8.0235e-1 −	9.9709e-1
	15	8.0261e-1 −	9.2899e-1 −	0.0000e+0 −	9.9897e-1 ≈	0.0000e+0 −	9.8804e-1 −	9.9902e-1 ≈	6.7994e-2 −	9.5475e-1 ≈	4.3484e-1 −	9.9905e-1
DTLZ2	3	5.6546e-1 ≈	5.6501e-1 −	5.4907e-1 −	5.6590e-1 +	5.6441e-1 −	5.6531e-1 −	5.6592e-1 +	5.6367e-1 −	5.4046e-1 −	5.4278e-1 −	5.6565e-1
	5	7.7982e-1 −	7.3897e-1 −	7.6384e-1 −	7.7993e-1 −	7.9210e-1 ≈	7.8311e-1 −	7.7970e-1 −	7.8790e-1 −	7.1785e-1 −	7.0652e-1 −	7.9269e-1
	8	9.0467e-1 −	8.7740e-1 −	8.6761e-1 −	9.1875e-1 +	9.1764e-1 +	8.8984e-1 −	9.1892e-1 +	9.1326e-1 +	8.6654e-1 −	8.5763e-1 −	9.1041e-1
	10	9.0088e-1 −	8.0456e-1 −	9.3203e-1 −	9.4328e-1 +	9.3787e-1 −	8.8575e-1 −	9.3971e-1 ≈	9.4697e-1 +	8.9233e-1 −	8.4811e-1 −	9.4086e-1
	15	9.7583e-1 +	8.0892e-1 +	9.7775e-1 +	9.8786e-1 +	8.3774e-1 +	7.7567e-1 +	9.8904e-1 +	9.6553e-1 +	9.1680e-1 +	8.0111e-1 +	7.6247e-1
DTLZ3	3	5.4085e-1 −	5.5088e-1 −	4.9951e-1 −	5.4314e-1 −	9.7029e-2 −	5.5600e-1 ≈	5.4042e-1 −	5.4001e-1 −	5.0034e-1 −	5.2571e-1 −	5.5769e-1
	5	5.8791e-1 −	4.9644e-1 −	3.4265e-1 −	7.2769e-1 −	1.3055e-1 −	7.3068e-1 −	7.4624e-1 −	7.7301e-1 ≈	5.0859e-1 −	4.1045e-1 −	7.7430e-1
	8	1.0678e-1 −	6.2462e-1 −	0.0000e+0 −	7.9316e-1 ≈	1.2141e-1 −	7.5439e-1 −	2.7998e-1 −	4.2640e-1 −	1.0414e-2 −	1.3616e-3 −	7.9906e-1
	10	6.6035e-2 −	2.2213e-1 −	0.0000e+0 −	7.6226e-1 −	0.0000e+0 −	7.2601e-1 −	5.8941e-1 ≈	0.0000e+0 −	1.4938e-2 −	0.0000e+0 −	7.8539e-1
	15	2.6939e-2 −	4.0877e-1 −	0.0000e+0 −	6.9495e-1 −	0.0000e+0 −	6.8626e-1 ≈	3.7109e-1 ≈	0.0000e+0 −	0.0000e+0 −	1.6342e-2 −	6.9781e-1
DTLZ4	3	5.1317e-1 −	5.0992e-1 ≈	5.5052e-1 +	5.6589e-1 ≈	5.6518e-1 +	5.5120e-1 ≈	4.6285e-1 ≈	3.8995e-1 −	5.4414e-1 −	5.4401e-1 −	5.5025e-1
	5	7.5931e-1 −	6.9633e-1 −	7.4898e-1 −	7.7363e-1 +	7.8605e-1 ≈	7.7192e-1 +	6.4189e-1 −	6.9403e-1 −	7.0294e-1 −	7.0395e-1 −	7.6468e-1
	8	8.8503e-1 ≈	8.6043e-1 −	8.8889e-1 −	9.1827e-1 +	8.1480e-1 ≈	7.5399e-1 −	8.9132e-1 +	8.6655e-1 ≈	8.5204e-1 −	8.6065e-1 −	8.9103e-1
	10	9.2172e-1 ≈	8.7939e-1 −	9.3479e-1 ≈	9.3786e-1 +	8.1177e-1 −	5.0876e-1 −	9.5212e-1 +	9.3840e-1 ≈	8.8056e-1 −	8.2955e-1 −	9.3235e-1
	15	9.7990e-1 +	9.3207e-1 −	9.7895e-1 +	9.8935e-1 +	4.3930e-1 −	3.4625e-2 −	9.9021e-1 +	9.8514e-1 +	9.3429e-1 −	5.6922e-1 −	9.7371e-1
WFG4	3	5.3571e-1 −	5.6397e-1 ≈	5.4494e-1 −	5.5470e-1 −	5.6178e-1 −	5.6340e-1 −	5.6256e-1 −	5.6348e-1 −	5.4082e-1 −	5.4157e-1 −	5.6402e-1
	5	6.9406e-1 −	7.7323e-1 −	7.6360e-1 −	7.6550e-1 −	7.7665e-1 −	7.7514e-1 −	7.6489e-1 −	7.8688e-1 +	7.2358e-1 −	6.9807e-1 −	7.7873e-1
	8	8.9795e-1 +	8.7707e-1 −	8.9052e-1 ≈	8.9658e-1 +	8.8951e-1 ≈	8.8485e-1 −	9.0004e-1 +	9.0687e-1 +	8.4702e-1 −	8.4514e-1 −	8.9054e-1
	10	8.8076e-1 −	8.8500e-1 −	9.4524e-1 +	9.0410e-1 −	9.0722e-1 ≈	9.1182e-1 ≈	9.1135e-1 ≈	9.4205e-1 +	7.7219e-1 −	8.2575e-1 −	9.1108e-1
	15	9.6862e-1 +	9.3312e-1 +	9.7697e-1 +	9.5722e-1 +	8.9489e-1 −	9.2970e-1 ≈	9.5817e-1 +	9.7455e-1 +	8.1379e-1 −	7.7459e-1 −	9.2807e-1
WFG5	3	5.0811e-1 −	5.1876e-1 −	5.0985e-1 −	5.2211e-1 −	5.2456e-1 +	5.2155e-1 −	5.2417e-1 +	5.2403e-1 +	5.0008e-1 −	5.0071e-1 −	5.2305e-1
	5	6.7784e-1 −	7.1455e-1 −	7.2077e-1 −	7.2665e-1 ≈	7.2799e-1 ≈	7.2339e-1 −	7.2523e-1 −	7.3827e-1 +	6.7177e-1 −	6.6512e-1 −	7.2698e-1
	8	8.5474e-1 +	8.1429e-1 −	8.2451e-1 −	8.5455e-1 +	8.3061e-1 +	8.2044e-1 −	8.4890e-1 +	8.5044e-1 +	8.0345e-1 −	8.0267e-1 −	8.2917e-1
	10	8.3957e-1 ≈	7.9836e-1 −	8.8557e-1 +	8.7331e-1 +	8.5711e-1 +	8.4205e-1 ≈	8.6000e-1 +	8.8004e-1 +	8.0102e-1 −	7.9905e-1 −	8.4445e-1
	15	9.0990e-1 +	8.6543e-1 +	9.0593e-1 +	9.1324e-1 +	8.5813e-1 ≈	8.5780e-1 ≈	9.0312e-1 +	9.0672e-1 +	7.9489e-1 −	8.0839e-1 −	8.5624e-1
WFG6	3	4.9317e-1 −	5.1424e-1 ≈	4.9314e-1 −	5.0555e-1 ≈	4.2489e-1 −	5.1835e-1 ≈	5.1329e-1 ≈	5.1727e-1 ≈	4.9239e-1 −	4.8852e-1 −	5.1283e-1
	5	6.4914e-1 −	7.0377e-1 ≈	6.9187e-1 −	7.0024e-1 −	6.1627e-1 −	7.1251e-1 ≈	6.9761e-1 −	7.2570e-1 +	6.5971e-1 −	6.5284e-1 −	7.1272e-1
	8	8.2864e-1 +	7.8697e-1 −	7.8276e-1 −	8.2358e-1 +	7.1605e-1 −	8.1264e-1 ≈	8.2299e-1 +	8.3347e-1 +	7.9265e-1 −	8.0289e-1 ≈	8.0761e-1
	10	8.1115e-1 −	7.9684e-1 −	8.4225e-1 ≈	6.8793e-1 −	7.3203e-1 −	8.2922e-1 ≈	8.4087e-1 ≈	8.6123e-1 +	7.5523e-1 −	7.6600e-1 −	8.3605e-1
	15	8.7972e-1 +	8.2834e-1 ≈	8.6473e-1 +	5.7621e-1 −	7.3715e-1 −	8.5011e-1 +	8.7282e-1 +	8.8123e-1 +	8.1284e-1 −	7.9687e-1 −	8.3970e-1
WFG7	3	5.3909e-1 −	5.6414e-1 ≈	5.5036e-1 −	5.5707e-1 −	5.6305e-1 −	5.6396e-1 −	5.6204e-1 −	5.6412e-1 −	5.3977e-1 −	5.3969e-1 −	5.6473e-1
	5	7.0968e-1 −	7.6762e-1 −	7.7232e-1 −	7.6600e-1 −	7.8484e-1 +	7.7758e-1 ≈	7.6261e-1 −	7.8928e-1 +	7.2316e-1 −	7.1923e-1 −	7.7585e-1
	8	9.0602e-1 +	8.7248e-1 −	8.7735e-1 −	8.8931e-1 +	9.0429e-1 +	8.8344e-1 ≈	8.9864e-1 +	9.1060e-1 +	8.7432e-1 −	8.8011e-1 ≈	8.8332e-1
	10	9.0301e-1 ≈	8.6138e-1 −	9.4214e-1 +	8.9281e-1 −	9.3511e-1 +	9.1022e-1 +	9.1371e-1 +	9.4557e-1 +	8.4320e-1 −	8.6037e-1 −	9.0426e-1
	15	9.7744e-1 +	8.8030e-1 −	9.7899e-1 +	7.3078e-1 ≈	9.2616e-1 +	9.1867e-1 ≈	9.6014e-1 +	9.7209e-1 +	8.9572e-1 ≈	8.7321e-1 −	9.1010e-1
WFG8	3	4.5742e-1 −	4.8233e-1 −	4.5802e-1 −	4.7125e-1 −	4.8425e-1 ≈	4.8404e-1 ≈	4.7934e-1 −	4.8483e-1 ≈	4.5254e-1 −	4.5114e-1 −	4.8553e-1
	5	5.7913e-1 −	6.6074e-1 −	6.3038e-1 −	6.5309e-1 −	6.7555e-1 +	6.6473e-1 −	6.6354e-1 −	6.7924e-1 +	6.2449e-1 −	6.0506e-1 −	6.6864e-1
	8	7.8683e-1 −	7.6944e-1 −	7.3101e-1 −	6.1353e-1 −	7.8919e-1 −	7.8221e-1 −	7.9401e-1 ≈	7.8276e-1 −	7.6145e-1 −	7.3608e-1 −	7.9701e-1
	10	7.9687e-1 −	8.0573e-1 −	7.7196e-1 −	6.3103e-1 −	8.6204e-1 +	8.2726e-1 −	7.8295e-1 −	8.3120e-1 −	7.6120e-1 −	7.3908e-1 −	8.4008e-1
	15	9.0723e-1 +	8.7409e-1 −	7.5481e-1 −	4.5541e-1 −	9.0219e-1 ≈	8.9405e-1 ≈	9.0559e-1 +	8.5850e-1 −	7.7920e-1 −	6.2745e-1 −	8.9830e-1
WFG9	3	5.1774e-1 −	5.3086e-1 −	5.3357e-1 −	5.3865e-1 −	5.0800e-1 ≈	5.4062e-1 −	5.3662e-1 −	5.4980e-1 +	5.2691e-1 −	5.2900e-1 −	5.4238e-1
	5	6.5003e-1 −	6.7550e-1 ≈	7.3803e-1 +	7.1662e-1 +	6.9680e-1 +	7.0432e-1 +	6.8650e-1 −	7.6033e-1 +	6.8643e-1 −	6.5355e-1 −	6.9216e-1
	8	8.1287e-1 +	6.8905e-1 ≈	8.3863e-1 +	8.1008e-1 +	7.5428e-1 +	7.4944e-1 +	7.7991e-1 +	8.4126e-1 +	7.7120e-1 +	7.5012e-1 +	7.0609e-1
	10	7.5737e-1 +	6.6235e-1 ≈	8.7941e-1 +	8.0401e-1 +	7.6969e-1 +	7.3194e-1 +	7.3834e-1 +	8.5453e-1 +	7.5348e-1 +	7.3328e-1 +	6.9029e-1
	15	8.5873e-1 +	6.3776e-1 ≈	8.5442e-1 +	7.7034e-1 +	7.6639e-1 +	6.9851e-1 +	7.7665e-1 +	8.4254e-1 +	7.8783e-1 +	7.6421e-1 +	6.5577e-1
+/ − / ≈		14/31/5	3/37/10	14/33/3	19/23/8	16/24/10	8/25/17	22/17/11	27/17/6	4/44/2	4/44/2	

Table 5. IGD mean values obtained by MaOEAs for regular PF problems with 3, 5, 8, 10, and 15 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
DTLZ1	3	2.0455e-2 –	1.8999e-2 –	4.3342e-2 –	1.9071e-2 –	2.0913e-2 –	1.8760e-2 –	1.7758e-2 +	1.9830e-2 –	4.1946e-2 –	4.1454e-2 –	1.8262e-2
	5	7.1089e-2 ≈	7.0313e-2 –	2.2125e-1 –	6.8980e-2 –	6.6538e-2 ≈	6.7269e-2 –	6.3039e-2 +	6.8156e-2 –	1.1630e-1 –	1.1026e-1 –	6.6248e-2
	8	1.4663e-1 ≈	1.5967e-1 –	1.1275e+0 –	1.2601e-1 +	2.4392e-1 –	1.2007e-1 +	1.2210e-1 +	1.1815e-1 +	1.6863e-1 –	1.7349e-1 –	1.2842e-1
	10	1.7518e-1 ≈	1.5208e-1 +	6.6509e+0 –	1.2089e-1 +	1.6272e+0 –	1.4624e-1 +	1.1808e-1 +	2.1658e-1 ≈	1.9734e-1 –	2.5883e-1 –	1.5947e-1
	15	2.8942e-1 –	1.8508e-1 +	1.0420e+1 –	1.8583e-1 +	3.4573e+1 –	1.9653e-1 ≈	1.6616e-1 +	1.9323e+0 –	2.5942e-1 –	6.3588e-1 –	2.0382e-1
DTLZ2	3	4.7023e-2 ≈	4.8768e-2 –	6.0885e-2 –	4.6737e-2 +	7.1848e-2 –	4.7731e-2 –	4.6763e-2 +	5.2146e-2 –	7.6394e-2 –	7.3543e-2 –	4.6925e-2
	5	1.9561e-1 –	2.0416e-1 –	2.2091e-1 –	1.9563e-1 –	2.1820e-1 –	1.9503e-1 ≈	1.9560e-1 –	2.0464e-1 –	2.7625e-1 –	2.7317e-1 –	1.9487e-1
	8	3.9296e-1 –	4.2244e-1 –	4.0705e-1 –	3.6912e-1 +	3.8347e-1 –	3.7844e-1 ≈	3.6883e-1 +	3.7900e-1 +	4.4458e-1 –	4.4259e-1 –	3.8011e-1
	10	5.3915e-1 –	4.4994e-1 +	5.0561e-1 –	4.3722e-1 +	5.0351e-1 ≈	4.8680e-1 ≈	4.3760e-1 +	4.7433e-1 +	5.6046e-1 –	5.6963e-1 –	4.7759e-1
	15	6.4739e-1 +	7.2175e-1 +	6.2390e-1 +	6.3814e-1 +	8.2417e-1 –	7.2858e-1 +	6.2714e-1 +	5.9536e-1 +	7.0187e-1 +	7.6885e-1 ≈	7.7747e-1
DTLZ3	3	5.8557e-2 –	6.0282e-2 –	9.9128e-2 –	5.6857e-2 –	1.8630e+0 –	5.0335e-2 ≈	5.8658e-2 –	6.4122e-2 –	1.2304e-1 –	8.9252e-2 –	4.9783e-2
	5	3.6555e-1 –	4.8179e-1 –	6.2601e-1 –	2.3899e-1 –	1.9482e+0 –	2.3530e-1 –	2.0876e-1 –	2.0839e-1 –	5.1698e-1 –	9.9648e-1 –	2.0285e-1
	8	3.5092e+0 –	6.9378e-1 –	8.6520e+1 –	4.8487e-1 +	1.2239e+1 –	5.3489e-1 ≈	1.9087e+0 –	1.1262e+0 ≈	3.8466e+0 –	6.3170e+0 –	4.9967e-1
	10	4.0172e+0 –	1.1835e+0 –	3.0014e+2 –	6.2395e-1 +	2.6168e+1 –	7.0085e-1 ≈	9.0035e-1 ≈	5.0952e+0 –	6.9190e+0 –	1.0188e+1 –	6.4449e-1
	15	5.5185e+0 –	9.7909e-1 –	3.8813e+2 –	9.6957e-1 –	7.6128e+1 –	8.9320e-1 ≈	2.1283e+0 ≈	3.7938e+1 –	1.6291e+1 –	1.7039e+1 –	9.0258e-1
DTLZ4	3	1.6249e-1 ≈	1.6567e-1 –	5.9659e-2 +	4.6737e-2 +	7.1183e-2 +	8.0446e-2 +	2.7435e-1 ≈	4.1024e-1 –	7.3307e-2 +	7.3218e-2 +	8.1794e-2
	5	2.3541e-1 ≈	3.8939e-1 –	2.4800e-1 +	2.1055e-1 ≈	2.3339e-1 +	2.3249e-1 ≈	4.7456e-1 –	3.9652e-1 –	3.1606e-1 –	2.9619e-1 –	2.6103e-1
	8	4.3313e-1 +	5.1497e-1 –	3.9643e-1 ≈	3.7217e-1 +	4.9533e-1 ≈	4.9525e-1 –	4.0746e-1 +	4.9595e-1 –	4.8516e-1 –	4.5826e-1 ≈	4.4041e-1
	10	4.9800e-1 ≈	6.0809e-1 –	5.0177e-1 ≈	4.5071e-1 +	5.9820e-1 –	7.1914e-1 –	4.6209e-1 +	5.1999e-1 ≈	5.8078e-1 –	6.0261e-1 –	5.0845e-1
	15	6.5121e-1 –	7.2085e-1 –	6.2534e-1 ≈	6.3650e-1 ≈	9.3626e-1 –	1.2710e+0 –	6.3157e-1 ≈	6.1682e-1 +	6.8710e-1 –	7.8689e-1 –	6.4159e-1
WFG4	3	2.2214e-1 –	1.9035e-1 –	2.3089e-1 –	2.0481e-1 –	2.7745e-1 –	1.8997e-1 –	1.8958e-1 –	2.0538e-1 –	3.0142e-1 –	2.9930e-1 –	1.8906e-1
	5	1.2003e+0 –	1.1298e+0 –	1.2506e+0 –	1.1359e+0 –	1.2806e+0 –	1.0860e+0 +	1.1310e+0 –	1.1627e+0 –	1.5606e+0 –	1.6905e+0 –	1.0943e+0
	8	3.4873e+0 –	3.2180e+0 –	3.6282e+0 –	3.4468e+0 –	3.3017e+0 –	3.0606e+0 +	3.4645e+0 –	3.2176e+0 –	3.7793e+0 –	3.7325e+0 –	3.0752e+0
	10	5.4511e+0 –	4.7688e+0 +	5.4975e+0 –	4.9782e+0 –	4.8808e+0 –	4.7549e+0 +	5.0935e+0 –	4.8907e+0 –	6.6092e+0 –	6.0558e+0 –	4.7806e+0
	15	9.4747e+0 –	9.5059e+0 –	9.7117e+0 –	9.6690e+0 –	8.5390e+0 +	9.1429e+0 ≈	9.3545e+0 –	8.6102e+0 +	1.2602e+1 –	1.2954e+1 –	9.2018e+0
WFG5	3	2.3132e-1 –	2.0071e-1 –	2.4073e-1 –	2.0648e-1 –	2.7209e-1 –	1.9940e-1 –	2.0141e-1 –	2.1623e-1 –	3.0790e-1 –	3.0503e-1 –	1.9861e-1
	5	1.1834e+0 –	1.1121e+0 –	1.2302e+0 –	1.1314e+0 –	1.2095e+0 –	1.0662e+0 +	1.1252e+0 –	1.1615e+0 –	1.5536e+0 –	1.5613e+0 –	1.0729e+0
	8	3.4259e+0 –	3.1532e+0 –	3.5286e+0 –	3.4314e+0 –	3.1866e+0 –	3.0330e+0 +	3.4312e+0 –	3.2817e+0 –	3.9460e+0 –	3.9273e+0 –	3.0481e+0
	10	5.2281e+0 –	4.6181e+0 +	5.4573e+0 –	4.8868e+0 –	4.7329e+0 ≈	4.6785e+0 +	5.0688e+0 –	4.9870e+0 –	6.0304e+0 –	6.0219e+0 –	4.7138e+0
	15	9.3581e+0 –	8.9562e+0 –	9.3440e+0 –	9.4817e+0 –	9.3098e+0 –	8.8399e+0 ≈	9.4281e+0 –	8.7811e+0 +	1.1816e+1 –	1.1533e+1 –	8.8625e+0
WFG6	3	2.4957e-1 –	2.1144e-1 ≈	2.6755e-1 –	2.2962e-1 –	3.8709e-1 –	2.0786e-1 ≈	2.1116e-1 ≈	2.2617e-1 –	3.0517e-1 –	3.0906e-1 –	2.1085e-1
	5	1.2294e+0 –	1.1263e+0 –	1.2721e+0 –	1.1412e+0 –	1.3117e+0 –	1.0793e+0 +	1.1321e+0 –	1.1784e+0 –	1.5013e+0 –	1.5264e+0 –	1.0876e+0
	8	3.4251e+0 –	3.2290e+0 –	3.7038e+0 –	3.3730e+0 –	3.3868e+0 –	3.0634e+0 ≈	3.4196e+0 –	3.4175e+0 –	3.7922e+0 –	3.7100e+0 –	3.0690e+0
	10	6.2409e+0 –	4.6876e+0 +	5.8234e+0 –	5.4639e+0 –	5.2372e+0 –	4.7227e+0 +	5.1142e+0 –	5.0647e+0 –	6.0287e+0 –	6.0128e+0 –	4.7605e+0
	15	9.7595e+0 –	9.3688e+0 –	1.0149e+1 –	1.0473e+1 –	9.1443e+0 ≈	8.9572e+0 +	9.4515e+0 –	8.5213e+0 +	1.0699e+1 –	1.0974e+1 –	9.0400e+0
WFG7	3	2.2229e-1 –	1.9087e-1 –	2.2467e-1 –	1.9878e-1 –	2.8485e-1 –	1.9041e-1 –	1.9034e-1 –	2.0715e-1 –	3.0694e-1 –	3.0258e-1 –	1.8945e-1
	5	1.2150e+0 –	1.1341e+0 –	1.2470e+0 –	1.1431e+0 –	1.3061e+0 –	1.0917e+0 +	1.1377e+0 –	1.1814e+0 –	1.5487e+0 –	1.5624e+0 –	1.1088e+0
	8	3.4623e+0 –	3.2501e+0 –	3.5351e+0 –	3.4231e+0 –	3.3017e+0 –	3.0949e+0 +	3.4512e+0 –	3.3309e+0 –	3.7698e+0 –	3.7258e+0 –	3.1087e+0
	10	5.2230e+0 –	4.8060e+0 ≈	5.3403e+0 –	4.9250e+0 –	5.0364e+0 –	4.7403e+0 +	5.1031e+0 –	5.0040e+0 –	6.0324e+0 –	5.9175e+0 –	4.7878e+0
	15	9.3893e+0 –	9.2381e+0 ≈	9.2691e+0 –	9.3984e+0 –	9.5825e+0 –	9.0831e+0 ≈	9.3959e+0 –	8.4617e+0 +	1.0735e+1 –	1.1019e+1 –	9.1736e+0
WFG8	3	2.9278e-1 –	2.5427e-1 –	3.1326e-1 –	2.8264e-1 –	3.3047e-1 –	2.4795e-1 ≈	2.5719e-1 –	2.6748e-1 –	3.4381e-1 –	3.4481e-1 –	2.4796e-1
	5	1.2965e+0 –	1.1464e+0 –	1.3051e+0 –	1.1776e+0 –	1.2676e+0 –	1.1167e+0 –	1.1520e+0 –	1.1876e+0 –	1.5133e+0 –	1.5099e+0 –	1.1122e+0
	8	3.4066e+0 –	3.3756e+0 –	3.6223e+0 –	3.3055e+0 –	3.3938e+0 –	3.1872e+0 –	3.3167e+0 –	3.3541e+0 –	3.8184e+0 –	3.9014e+0 –	3.1537e+0
	10	5.2986e+0 –	5.0019e+0 ≈	5.8360e+0 –	5.4738e+0 –	5.2310e+0 –	4.8637e+0 ≈	5.1167e+0 –	5.0980e+0 –	6.2083e+0 –	6.3825e+0 –	4.8601e+0
	15	9.4500e+0 ≈	9.6919e+0 ≈	1.0080e+1 –	1.0218e+1 –	9.9442e+0 –	9.4825e+0 ≈	9.2449e+0 ≈	8.7438e+0 +	1.1548e+1 –	1.2707e+1 –	9.4914e+0
WFG9	3	2.1627e-1 –	1.9456e-1 –	2.0780e-1 –	1.9715e-1 –	2.9812e-1 –	1.8811e-1 ≈	1.9183e-1 –	2.0456e-1 –	3.0577e-1 –	3.0024e-1 –	1.8902e-1
	5	1.1822e+0 –	1.1158e+0 –	1.1863e+0 –	1.1203e+0 –	1.2039e+0 –	1.0696e+0 +	1.1224e+0 –	1.1404e+0 –	1.5752e+0 –	1.6499e+0 –	1.0785e+0
	8	3.2641e+0 –	3.2801e+0 –	3.4183e+0 –	3.3168e+0 –	3.1830e+0 –	3.1137e+0 +	3.2948e+0 –	3.1853e+0 –	3.8286e+0 –	3.8695e+0 –	3.1557e+0
	10	5.0455e+0 –	4.7340e+0 +	5.3305e+0 –	4.8411e+0 –	4.6663e+0 +	4.7164e+0 +	5.0037e+0 –	4.8010e+0 ≈	6.0443e+0 –	6.0269e+0 –	4.7802e+0
	15	8.8082e+0 +	8.9583e+0 +	9.0206e+0 ≈	9.5035e+0 –	8.7721e+0 +	8.8581e+0 +	9.2434e+0 –	8.2560e+0 +	1.1056e+1 –	1.1090e+1 –	9.0561e+0
+/ − / ≈		3/39/8	9/36/5	3/43/4	12/36/2	5/40/5	20/12/18	11/33/6	11/35/4	2/48/0	1/47/2	

Table 6. HV mean values obtained by MaOEAs for irregular PF problems with 3, 5, 8, 10, and 15 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
DTLZ5	3	1.9719e-1 −	2.0039e-1 ≈	1.9562e-1 −	1.5606e-1 −	1.9650e-1 −	2.0033e-1 ≈	2.0023e-1 −	2.0023e-1 −	1.8627e-1 −	1.8483e-1 −	2.0033e-1
	5	1.0414e-1 −	7.4799e-2 −	5.4968e-2 −	9.1691e-2 −	1.1577e-1 −	1.0726e-1 −	1.0765e-1 −	1.0211e-1 −	1.0961e-1 −	9.6973e-2 −	1.2029e-1
	8	8.7861e-2 −	9.4556e-2 −	3.9071e-2 −	9.2750e-2 −	5.9080e-2 −	9.2719e-2 −	8.2013e-2 −	8.9237e-2 −	1.0482e-3 −	0.0000e+0 −	9.9724e-2
	10	8.7540e-2 −	4.0481e-2 −	3.3072e-2 −	9.2471e-2 −	4.3445e-2 −	9.0003e-2 −	8.5638e-2 −	8.8941e-2 −	1.2422e-4 −	0.0000e+0 −	9.6376e-2
	15	8.3935e-2 −	9.1468e-2 −	1.8313e-2 −	8.8855e-2 −	8.4590e-2 −	8.9870e-2 −	7.9040e-2 −	8.5149e-2 −	2.7681e-4 −	0.0000e+0 −	9.3086e-2
DTLZ6	3	1.9535e-1 −	2.0017e-1 −	1.9382e-1 −	1.5717e-1 −	1.9670e-1 −	2.0038e-1 −	2.0029e-1 −	2.0032e-1 −	1.8225e-1 −	1.8068e-1 −	2.0043e-1
	5	9.3180e-2 −	6.5166e-2 −	6.8344e-2 −	9.6296e-2 −	1.1626e-1 −	9.6169e-2 −	1.0202e-1 −	9.1791e-2 −	1.0143e-1 −	1.0242e-1 −	1.2432e-1
	8	5.8347e-2 −	9.3034e-2 −	2.8903e-3 −	5.9333e-2 −	9.1022e-2 −	8.6532e-2 −	9.1337e-2 −	7.7750e-2 −	1.4748e-5 −	0.0000e+0 −	1.0016e-1
	10	3.0294e-2 −	3.8035e-2 −	0.0000e+0 −	9.1548e-2 −	9.0907e-2 −	8.3528e-2 −	9.0436e-2 −	2.7707e-2 −	0.0000e+0 −	0.0000e+0 −	9.6499e-2
	15	5.1434e-2 −	9.2075e-2 −	0.0000e+0 −	3.5334e-2 −	9.0909e-2 −	8.9810e-2 −	9.0832e-2 −	0.0000e+0 −	0.0000e+0 −	0.0000e+0 −	9.3387e-2
DTLZ7	3	2.7524e-1 −	2.6942e-1 −	2.7928e-1 −	2.6747e-1 −	2.7740e-1 −	2.8050e-1 −	2.2390e-1 −	2.6713e-1 −	2.6856e-1 −	2.6905e-1 −	2.8102e-1
	5	2.3870e-1 −	1.9616e-1 −	2.5001e-1 +	2.1231e-1 −	2.6204e-1 +	2.4323e-1 −	2.1688e-1 −	2.5051e-1 +	2.2405e-1 −	2.2163e-1 −	2.4812e-1
	8	1.7809e-1 −	1.5752e-1 −	1.1269e-1 −	1.3958e-1 −	2.0725e-1 +	1.2149e-1 −	1.8034e-1 −	1.8231e-1 ≈	1.7212e-1 −	1.6315e-1 −	1.8209e-1
	10	1.2244e-1 −	1.7610e-1 +	2.4035e-2 −	1.3166e-1 −	1.8630e-1 +	2.9266e-2 −	1.0743e-1 −	1.2056e-1 −	1.5403e-1 −	1.3723e-1 −	1.6073e-1
	15	1.3232e-1 ≈	1.2863e-1 −	5.9501e-4 −	1.1758e-1 −	1.4988e-1 +	9.5650e-3 −	4.9748e-3 −	2.8803e-3 −	1.2984e-1 −	1.0721e-1 −	1.3692e-1
IDTLZ1	3	2.2220e-1 −	2.2520e-1 −	1.5108e-1 −	1.8836e-1 −	2.2209e-1 −	2.2589e-1 ≈	2.2399e-1 −	2.2378e-1 −	1.9408e-1 −	1.9061e-1 −	2.2586e-1
	5	6.8859e-3 −	8.9721e-3 −	9.9489e-3 +	1.6183e-3 −	8.4594e-3 −	9.8933e-3 ≈	9.4462e-3 −	9.6338e-3 −	6.5681e-3 −	5.0209e-3 −	9.9232e-3
	8	2.5450e-5 +	1.1569e-5 −	2.3028e-5 +	2.7824e-6 −	9.3526e-6 −	1.2116e-5 −	1.8310e-5 +	1.5633e-5 ≈	2.0235e-5 +	1.8159e-5 +	1.4314e-5
	10	3.2539e-7 +	1.3484e-7 −	2.3197e-7 +	5.8231e-9 −	6.4903e-8 ≈	1.8648e-7 ≈	1.1006e-7 −	3.3333e-8 −	2.2234e-7 +	9.3080e-8 −	1.6246e-7
	15	3.7881e-12 +	8.2942e-13 +	6.1862e-13 +	1.6231e-13 +	4.7624e-15 ≈	0.0000e+0 ≈	4.4621e-13 +	0.0000e+0 ≈	2.7445e-12 +	7.1626e-13 +	0.0000e+0
IDTLZ2	3	5.2856e-1 −	5.4280e-1 −	5.3574e-1 −	5.2200e-1 −	5.2585e-1 −	5.4386e-1 −	5.3419e-1 −	5.4117e-1 −	5.1302e-1 −	5.1510e-1 −	5.4435e-1
	5	7.7508e-2 −	8.9694e-2 −	1.0789e-1 +	7.6533e-2 −	1.1454e-1 +	9.4537e-2 −	9.8372e-2 +	1.0778e-1 +	8.7461e-2 −	8.0692e-2 −	9.5884e-2
	8	1.7839e-3 +	1.3907e-3 +	2.0287e-3 +	1.3865e-3 +	2.2498e-3 +	8.2452e-4 −	1.2664e-3 +	2.3318e-3 +	3.4009e-3 +	3.2324e-3 +	9.4255e-4
	10	1.7545e-4 +	1.0508e-4 +	1.0410e-4 +	1.0655e-4 +	1.5452e-4 +	2.2125e-5 −	1.3796e-4 +	8.9464e-5 +	2.1763e-4 +	7.1306e-5 +	3.6668e-5
	15	1.3864e-7 +	2.0717e-8 +	1.3253e-9 −	6.6175e-8 +	7.3103e-8 +	0.0000e+0 −	1.2687e-7 +	0.0000e+0 −	1.8731e-7 +	1.7206e-8 ≈	9.0629e-9
WFG1	3	8.9773e-1 −	9.3844e-1 −	9.2795e-1 −	8.5558e-1 −	7.5571e-1 −	9.3851e-1 −	9.3226e-1 −	9.4544e-1 +	9.3940e-1 −	9.3758e-1 −	9.4157e-1
	5	7.7756e-1 −	9.9176e-1 −	9.2662e-1 −	8.8262e-1 −	7.9408e-1 −	9.8543e-1 −	8.8982e-1 −	9.9719e-1 +	9.9295e-1 −	9.9059e-1 −	9.9544e-1
	8	8.4824e-1 −	9.9780e-1 −	9.0890e-1 −	8.9368e-1 −	9.0030e-1 −	9.8916e-1 ≈	9.1830e-1 −	9.9987e-1 +	9.9558e-1 ≈	9.9835e-1 ≈	9.9923e-1
	10	8.8010e-1 −	9.9219e-1 −	9.0445e-1 −	8.7714e-1 −	9.2288e-1 −	9.9956e-1 ≈	9.0208e-1 −	9.9996e-1 +	9.9261e-1 −	9.8788e-1 −	9.9951e-1
	15	9.4480e-1 −	9.9571e-1 −	6.9179e-1 −	9.5388e-1 −	9.3792e-1 −	9.9579e-1 ≈	9.6274e-1 −	9.9998e-1 +	9.8862e-1 −	9.9902e-1 −	9.9980e-1
WFG2	3	9.3019e-1 −	9.3063e-1 −	9.3115e-1 −	9.2691e-1 −	8.6276e-1 −	9.3483e-1 ≈	9.3432e-1 −	9.3774e-1 +	9.2632e-1 −	9.2543e-1 −	9.3492e-1
	5	9.8123e-1 −	9.9160e-1 −	9.8795e-1 −	9.7839e-1 −	9.5581e-1 −	9.9279e-1 ≈	9.9268e-1 ≈	9.9680e-1 +	9.9135e-1 −	9.9162e-1 −	9.9326e-1
	8	9.9493e-1 −	9.9519e-1 ≈	9.8927e-1 −	9.7682e-1 −	9.8190e-1 −	9.9628e-1 ≈	9.9468e-1 −	9.9898e-1 +	9.9773e-1 +	9.9810e-1 +	9.9600e-1
	10	9.9256e-1 −	9.8985e-1 −	9.8753e-1 −	9.5837e-1 −	9.8142e-1 −	9.9543e-1 ≈	9.8779e-1 −	9.9877e-1 +	9.9614e-1 +	9.9594e-1 +	9.9552e-1
	15	9.9105e-1 ≈	9.8642e-1 ≈	9.6991e-1 −	9.5968e-1 −	9.9012e-1 ≈	9.8895e-1 ≈	9.9055e-1 ≈	9.9846e-1 +	9.9478e-1 +	9.8949e-1 −	9.8963e-1
WFG3	3	3.9062e-1 −	4.0233e-1 ≈	3.8013e-1 −	3.4395e-1 −	4.0830e-1 +	4.0196e-1 −	3.8394e-1 −	3.9817e-1 −	3.9034e-1 −	3.9224e-1 −	4.0320e-1
	5	1.7924e-1 −	2.0783e-1 ≈	6.0286e-2 −	1.3003e-1 −	1.7772e-1 −	2.0731e-1 ≈	1.0508e-1 −	1.9746e-1 −	2.2348e-1 ≈	2.3015e-1 +	2.1218e-1
	8	4.6116e-2 +	6.6120e-2 +	0.0000e+0 −	0.0000e+0 −	5.9076e-4 −	1.1426e-2 −	7.6343e-3 −	4.7261e-2 +	1.1946e-1 +	1.5519e-1 +	2.1166e-2
	10	7.5326e-4 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	1.3274e-3 +	9.3781e-2 +	7.7121e-2 +	0.0000e+0
	15	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0
+/ − / ≈		7/29/4	6/27/7	8/30/2	4/34/2	9/26/5	0/24/16	6/30/4	16/20/4	11/26/3	9/28/3	

Table 7. IGD mean values obtained by MaOEAs for irregular PF problems with 3, 5, 8, 10, and 15 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
DTLZ5	3	8.2405e-3 –	3.5501e-3 –	7.6569e-3 –	6.5918e-2 –	1.4439e-2 –	3.5546e-3 –	3.7503e-3 –	3.8817e-3 –	2.7143e-2 –	4.2321e-2 –	3.5111e-3
	5	1.3463e-1 –	8.5019e-2 –	2.5331e-1 –	3.5429e-1 –	3.9562e-2 –	9.1391e-2 –	7.4502e-2 –	1.2379e-1 –	7.7968e-2 –	5.8817e-1 –	3.5593e-2
	8	2.5693e-1 –	1.0715e-1 –	3.0568e-1 –	1.9656e-1 –	6.7132e-1 –	1.2564e-1 –	1.4044e-1 –	3.2098e-1 –	4.0626e-1 –	7.7609e-1 –	4.9610e-2
	10	3.5589e-1 –	8.1567e-2 –	3.6788e-1 –	4.6866e-1 –	7.5919e-1 –	1.4983e-1 –	1.6180e-1 –	3.5964e-1 –	6.6140e-1 –	2.0986e+0 –	5.3683e-2
	15	3.5063e-1 –	8.2761e-2 –	6.9656e-1 –	2.7212e-1 –	7.4558e-1 –	1.4938e-1 –	1.6338e-1 –	3.9145e-1 –	2.0296e+0 –	1.9190e+0 –	4.3734e-2
DTLZ6	3	1.1431e-2 –	3.6015e-3 –	1.0735e-2 –	6.8234e-2 –	1.4188e-2 –	3.5272e-3 ≈	3.5847e-3 –	3.8820e-3 –	3.5147e-2 –	3.8156e-2 –	3.5189e-3
	5	2.5600e-1 –	7.5915e-2 –	3.5499e-1 –	2.4329e-1 –	4.6142e-2 –	2.1606e-1 –	9.8306e-2 –	2.5447e-1 –	1.2542e-1 –	1.1298e-1 –	2.3722e-2
	8	6.0614e-1 –	1.3700e-1 –	1.0204e+0 –	1.7037e-1 –	6.7998e-1 –	3.7834e-1 –	1.6462e-1 –	4.6213e-1 –	5.7648e-1 –	1.1166e+0 –	5.1889e-2
	10	8.3160e-1 –	1.3410e-1 –	1.7156e+0 –	2.9638e-1 –	7.2122e-1 –	4.2571e-1 –	1.7854e-1 –	9.5535e-1 –	2.7358e+0 –	4.3003e+0 –	5.3651e-2
	15	5.8277e-1 –	1.3728e-1 –	2.0234e+0 –	3.1817e-1 –	7.4209e-1 –	4.1769e-1 –	1.4355e-1 –	1.8051e+0 –	7.8689e+0 –	8.6521e+0 –	4.6620e-2
DTLZ7	3	6.3210e-2 –	9.2659e-2 –	5.9940e-2 –	9.2717e-2 –	6.2068e-2 –	5.3433e-2 –	5.8508e-1 –	1.5549e-1 ≈	8.7638e-2 –	8.8178e-2 –	5.1818e-2
	5	3.5300e-1 –	4.7689e-1 –	3.4627e-1 –	4.5467e-1 –	2.8355e-1 –	2.8063e-1 –	1.4143e+0 –	3.0695e-1 –	4.8448e-1 –	4.6731e-1 –	2.6486e-1
	8	8.5084e-1 –	8.9638e-1 –	7.1096e-1 +	2.7394e+0 –	8.7741e-1 –	7.5354e-1 –	1.4158e+0 –	6.6977e-1 +	1.6839e+0 –	1.6631e+0 –	7.2827e-1
	10	1.3737e+0 –	1.1782e+0 –	1.0802e+0 ≈	1.8496e+0 –	1.1486e+0 –	1.1382e+0 –	1.8587e+0 –	9.7282e-1 +	2.3631e+0 –	1.9439e+0 –	1.0850e+0
	15	8.1894e+0 –	2.3626e+0 ≈	2.9477e+0 –	5.2901e+0 –	2.6567e+0 ≈	1.9230e+0 +	3.9786e+0 –	2.3693e+0 ≈	7.8514e+0 –	7.3539e+0 –	2.5203e+0
IDTLZ1	3	1.9833e-2 –	1.8791e-2 –	9.3207e-2 –	4.1213e-2 –	1.9837e-2 –	1.8526e-2 –	1.8492e-2 –	1.9869e-2 –	3.5202e-2 –	3.8802e-2 –	1.8349e-2
	5	8.5856e-2 –	7.0874e-2 –	7.1372e-2 –	1.8501e-1 –	7.3101e-2 –	6.5755e-2 –	6.7103e-2 –	6.9258e-2 –	1.0963e-1 –	1.2952e-1 –	6.5559e-2
	8	1.3674e-1 –	1.5239e-1 –	1.1972e-1 –	2.2582e-1 –	1.3495e-1 –	1.1878e-1 –	1.2690e-1 ≈	1.2266e-1 –	1.5980e-1 –	1.6338e-1 –	1.1754e-1
	10	1.5483e-1 –	1.6622e-1 –	1.4179e-1 +	2.9322e-1 –	1.5854e-1 –	1.4603e-1 –	1.5932e-1 –	1.5006e-1 –	1.6961e-1 –	2.1010e-1 –	1.4239e-1
	15	1.7038e-1 +	2.2362e-1 –	2.0227e-1 –	2.5818e-1 –	3.1798e-1 –	1.8459e-1 –	2.0915e-1 –	1.8854e-1 –	1.8515e-1 –	2.2787e-1 –	1.7923e-1
IDTLZ2	3	6.4781e-2 –	4.8218e-2 –	6.5900e-2 –	7.2866e-2 –	8.6520e-2 –	4.8402e-2 –	6.7055e-2 –	5.0890e-2 –	1.0797e-1 –	1.0579e-1 –	4.7846e-2
	5	2.6435e-1 –	2.0123e-1 –	2.2982e-1 –	3.0578e-1 –	2.3148e-1 –	1.9347e-1 –	2.0909e-1 –	1.9711e-1 –	4.0428e-1 –	4.1334e-1 –	1.9189e-1
	8	5.1994e-1 –	3.9333e-1 –	4.3391e-1 –	5.9339e-1 –	4.1556e-1 –	3.6921e-1 –	4.3756e-1 –	3.8401e-1 –	6.3531e-1 –	6.4094e-1 –	3.6265e-1
	10	6.7452e-1 –	5.0584e-1 –	5.5884e-1 –	7.3467e-1 –	5.0457e-1 –	4.6426e-1 –	5.2859e-1 –	4.5746e-1 –	6.9033e-1 –	7.7697e-1 –	4.4537e-1
	15	8.4630e-1 –	7.0813e-1 ≈	6.3137e-1 +	8.5946e-1 –	6.4911e-1 +	6.0320e-1 +	6.5643e-1 +	5.8806e-1 +	7.9924e-1 –	9.0879e-1 –	7.0707e-1
WFG1	3	1.9467e-1 –	1.6663e-1 –	1.9987e-1 –	2.8700e-1 –	5.7264e-1 –	1.6711e-1 –	1.9294e-1 –	1.5745e-1 ≈	1.9966e-1 –	2.0122e-1 –	1.5746e-1
	5	7.0761e-1 –	5.3001e-1 ≈	5.1369e-1 ≈	5.3971e-1 ≈	9.5301e-1 –	5.1218e-1 ≈	5.3755e-1 –	5.6551e-1 –	6.4418e-1 –	6.0429e-1 –	5.2375e-1
	8	1.1744e+0 +	1.4065e+0 –	1.0459e+0 +	1.1117e+0 +	1.7488e+0 –	1.0836e+0 +	1.1653e+0 +	1.2381e+0 +	1.4715e+0 –	1.4307e+0 –	1.3067e+0
	10	1.4368e+0 +	1.4530e+0 +	1.2818e+0 +	1.2170e+0 +	2.3256e+0 –	1.3602e+0 +	1.5616e+0 +	1.6620e+0 +	1.8261e+0 ≈	1.8378e+0 ≈	1.7783e+0
	15	2.1534e+0 +	2.1844e+0 +	2.3490e+0 +	1.8925e+0 +	3.8437e+0 ≈	2.0077e+0 +	2.2568e+0 +	2.3748e+0 +	2.5623e+0 +	2.2130e+0 +	3.5954e+0
WFG2	3	1.4896e-1 ≈	1.4514e-1 +	1.7354e-1 –	1.6822e-1 –	4.5268e-1 –	1.5061e-1 –	1.4007e-1 +	1.5934e-1 –	2.3571e-1 –	2.3307e-1 –	1.4794e-1
	5	7.1130e-1 ≈	5.5255e-1 +	5.4137e-1 +	4.5289e-1 +	1.0842e+0 –	6.3488e-1 +	4.5843e-1 +	5.9798e-1 +	6.6191e-1 ≈	6.5889e-1 ≈	6.5385e-1
	8	1.2091e+0 +	1.5375e+0 +	1.1241e+0 +	1.0583e+0 +	1.8161e+0 –	1.5803e+0 +	1.1041e+0 +	1.3285e+0 +	1.3695e+0 +	1.3543e+0 +	1.7022e+0
	10	1.3748e+0 +	1.3648e+0 +	1.3286e+0 +	1.3601e+0 +	2.1140e+0 +	2.1054e+0 +	1.1856e+0 +	1.8188e+0 +	1.8809e+0 +	1.9382e+0 +	2.3110e+0
	15	1.8463e+0 +	2.1086e+0 +	1.9743e+0 +	1.8216e+0 +	3.5149e+0 –	3.6914e+0 –	1.9384e+0 +	2.5367e+0 ≈	3.5555e+0 –	2.6908e+0 ≈	2.9736e+0
WFG3	3	9.5915e-2 –	5.9845e-2 +	9.9136e-2 –	1.9360e-1 –	3.9046e-2 +	6.7184e-2 –	9.9494e-2 –	7.7293e-2 –	1.0688e-1 –	1.0414e-1 –	6.1386e-2
	5	4.9220e-1 –	3.9827e-1 ≈	5.8997e-1 –	7.2346e-1 –	2.9930e-1 +	4.1726e-1 –	6.7608e-1 –	4.7465e-1 –	4.6935e-1 ≈	6.6080e-1 –	3.9641e-1
	8	6.6214e-1 +	1.3659e+0 –	1.5666e+0 –	2.8467e+0 –	7.3186e-1 +	1.1990e+0 ≈	2.1086e+0 –	1.4187e+0 –	1.6363e+0 ≈	4.2209e+0 –	1.1421e+0
	10	1.5075e+0 +	2.0813e+0 –	2.3883e+0 –	4.3403e+0 –	9.6639e-1 +	2.0102e+0 –	2.8906e+0 –	2.0657e+0 –	1.0401e+0 +	1.1362e+0 +	1.7694e+0
	15	2.4858e+0 +	3.7081e+0 ≈	4.5191e+0 –	9.1009e+0 –	1.1570e+0 +	5.3388e+0 –	6.1692e+0 –	4.5365e+0 –	3.7963e+0 –	9.1592e+0 –	3.4450e+0
+/ − / ≈		10/28/2	8/27/5	10/28/2	7/32/1	7/31/2	8/29/3	9/30/1	9/27/4	4/32/4	4/33/3	

Table 8. HV mean values obtained by MaOEAs for regular and irregular problems with 6 and 12 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
DTLZ1	6	9.6607e-1 –	9.3334e-1 –	6.0216e-1 –	9.8264e-1 –	9.6431e-1 –	9.8009e-1 –	9.8411e-1 –	9.8716e-1 ≈	9.5339e-1 –	9.6177e-1 –	9.8709e-1
	12	8.4941e-1 ≈	9.1263e-1 –	0.0000e+0 –	9.9889e-1 +	2.8707e-1 –	9.7994e-1 –	9.9767e-1 +	1.8947e-1 –	9.7585e-1 –	5.4388e-1 –	9.9324e-1
DTLZ2	6	8.2635e-1 –	7.3588e-1 –	8.1797e-1 –	8.2685e-1 –	8.5012e-1 ≈	8.3605e-1 –	8.2681e-1 –	8.4561e-1 –	7.7416e-1 –	7.6120e-1 –	8.5040e-1
	12	9.5006e-1 ≈	8.0916e-1 –	9.6032e-1 +	9.6428e-1 +	9.1601e-1 –	8.4178e-1 –	9.7144e-1 +	9.5794e-1 +	8.5221e-1 –	7.8615e-1 –	9.4215e-1
DTLZ3	6	2.3527e-1 –	4.5130e-1 –	0.0000e+0 –	7.4376e-1 –	1.6556e-1 –	7.5014e-1 –	6.9941e-1 –	7.4508e-1 ≈	3.0607e-1 –	1.2330e-1 –	8.1751e-1
	12	0.0000e+0 –	1.0782e-1 –	0.0000e+0 –	8.9469e-1 +	0.0000e+0 –	7.1268e-1 –	3.9816e-1 ≈	0.0000e+0 –	0.0000e+0 –	0.0000e+0 –	7.4154e-1
DTLZ4	6	8.1266e-1 ≈	7.4358e-1 –	8.2922e-1 ≈	8.2053e-1 ≈	7.9953e-1 ≈	8.1240e-1 ≈	7.5645e-1 –	7.6915e-1 –	7.6728e-1 –	7.6210e-1 –	8.1156e-1
	12	9.3978e-1 –	9.2682e-1 –	9.6724e-1 +	9.6400e-1 ≈	7.5886e-1 –	1.5159e-1 –	9.7171e-1 +	9.6686e-1 +	8.7690e-1 –	6.8689e-1 –	9.6190e-1
WFG4	6	7.3343e-1 –	8.1961e-1 –	8.0464e-1 –	8.0181e-1 –	8.2917e-1 –	8.2751e-1 –	8.0653e-1 –	8.4278e-1 +	7.7272e-1 –	7.7584e-1 –	8.3181e-1
	12	9.5207e-1 +	9.1234e-1 –	9.5926e-1 +	9.4554e-1 +	9.2012e-1 ≈	9.2469e-1 ≈	9.5647e-1 +	9.6173e-1 +	7.6836e-1 –	7.4272e-1 –	9.2242e-1
WFG5	6	7.1596e-1 –	7.5526e-1 –	7.7062e-1 –	7.6741e-1 –	7.7608e-1 ≈	7.7043e-1 –	7.6522e-1 –	7.9113e-1 +	7.1994e-1 –	7.2194e-1 –	7.7643e-1
	12	8.9812e-1 +	8.3085e-1 –	8.9923e-1 +	9.0076e-1 +	8.6230e-1 ≈	8.5396e-1 ≈	8.9584e-1 +	8.9698e-1 +	7.3708e-1 –	7.6217e-1 –	8.5405e-1
WFG6	6	6.9427e-1 –	7.4867e-1 ≈	7.3563e-1 –	7.3612e-1 –	6.6425e-1 –	7.6388e-1 ≈	7.3564e-1 –	7.8112e-1 +	7.1328e-1 –	7.2025e-1 –	7.5715e-1
	12	8.6666e-1 +	8.1850e-1 –	8.5753e-1 +	6.5672e-1 –	7.3835e-1 –	8.4537e-1 ≈	8.7041e-1 +	8.7014e-1 +	7.2666e-1 –	7.3115e-1 –	8.4037e-1
WFG7	6	7.3717e-1 –	8.1148e-1 –	8.2487e-1 ≈	7.9587e-1 –	8.3922e-1 +	8.2667e-1 ≈	8.0588e-1 –	8.4509e-1 +	7.8462e-1 –	7.9009e-1 –	8.2506e-1
	12	9.4712e-1 +	9.0715e-1 ≈	9.6414e-1 +	9.1954e-1 +	9.3453e-1 +	9.2077e-1 +	9.6330e-1 +	9.6133e-1 +	8.3723e-1 –	7.9258e-1 –	9.1352e-1
WFG8	6	6.2020e-1 –	7.0059e-1 –	6.5339e-1 –	6.3874e-1 –	7.1925e-1 ≈	7.0814e-1 –	6.7638e-1 –	7.2020e-1 ≈	6.8268e-1 –	6.7427e-1 –	7.1886e-1
	12	8.4631e-1 –	8.5529e-1 –	8.5402e-1 –	4.6383e-1 –	8.8155e-1 +	8.6106e-1 –	8.6544e-1 –	8.5656e-1 –	7.2687e-1 –	6.5942e-1 –	8.7120e-1
WFG9	6	6.5952e-1 –	7.0302e-1 –	7.8161e-1 +	7.3247e-1 ≈	7.1599e-1 ≈	7.2139e-1 ≈	7.0448e-1 –	8.0230e-1 +	6.9909e-1 ≈	7.1166e-1 ≈	7.2733e-1
	12	8.2959e-1 +	6.3682e-1 –	8.4215e-1 +	7.6731e-1 +	7.6797e-1 +	7.3569e-1 +	7.7994e-1 +	8.5677e-1 +	7.4522e-1 +	6.9874e-1 ≈	6.7629e-1
DTLZ5	6	9.5743e-2 –	7.9617e-2 –	7.9093e-2 –	9.3477e-2 –	9.7934e-2 –	9.8736e-2 –	9.1302e-2 –	9.1221e-2 –	1.0194e-1 –	1.0459e-1 –	1.0985e-1
	12	9.0642e-2 –	2.0349e-2 –	2.8022e-2 –	9.1001e-2 –	5.2294e-2 –	9.0701e-2 –	9.1098e-2 –	8.6420e-2 –	1.5053e-3 –	3.8933e-3 –	9.4635e-2
DTLZ6	6	8.4768e-2 –	5.2385e-2 –	4.1086e-2 –	9.5758e-2 –	1.0250e-1 –	8.7084e-2 –	9.2530e-2 –	9.0923e-2 –	9.6510e-2 –	9.5110e-2 –	1.1136e-1
	12	2.4433e-2 –	6.2422e-2 –	0.0000e+0 –	8.7903e-2 –	9.0909e-2 –	8.7883e-2 –	9.0959e-2 –	1.5142e-2 –	0.0000e+0 –	0.0000e+0 –	9.4760e-2
DTLZ7	6	2.2016e-1 ≈	1.9792e-1 –	2.0778e-1 –	1.7572e-1 –	2.4585e-1 +	2.0570e-1 –	2.1654e-1 –	2.3210e-1 +	2.0756e-1 –	2.0547e-1 –	2.1940e-1
	12	1.4614e-1 –	1.6147e-1 +	5.0732e-3 –	1.3062e-1 –	1.6931e-1 +	2.2636e-2 –	5.5651e-2 –	8.2909e-2 –	1.3537e-1 –	1.2599e-1 –	1.5286e-1
IDTLZ1	6	9.6765e-4 –	1.0949e-3 –	1.5873e-3 +	1.5499e-4 –	1.0323e-3 –	1.2561e-3 ≈	1.3153e-3 ≈	1.2602e-3 ≈	9.3365e-4 –	8.3825e-4 –	1.2896e-3
	12	2.5657e-9 +	8.6290e-10 +	1.0423e-9 +	4.2275e-11 –	0.0000e+0 ≈	0.0000e+0 ≈	1.3220e-9 +	0.0000e+0 ≈	1.7039e-9 +	4.7599e-10 –	6.9231e-10
IDTLZ2	6	1.6832e-2 –	2.3340e-2 ≈	3.3904e-2 +	1.9151e-2 –	3.4955e-2 +	2.2954e-2 ≈	2.7146e-2 +	3.5128e-2 +	3.0075e-2 +	2.9007e-2 +	2.2753e-2
	12	1.2958e-5 +	1.7217e-6 ≈	1.1280e-6 –	8.5614e-6 +	7.9074e-6 +	4.4944e-7 –	9.6243e-6 +	2.4663e-6 ≈	5.4051e-6 +	8.1889e-7 –	1.6878e-6
WFG1	6	7.4714e-1 –	9.8264e-1 –	9.7724e-1 –	7.9910e-1 –	8.8551e-1 –	9.9173e-1 –	8.7366e-1 –	9.9917e-1 +	9.9476e-1 –	9.9456e-1 –	9.9673e-1
	12	9.8462e-1 –	9.9906e-1 –	8.2754e-1 –	9.3737e-1 –	9.2146e-1 –	9.9974e-1 ≈	9.8279e-1 –	9.9995e-1 +	9.7953e-1 –	9.8858e-1 –	9.9971e-1
WFG2	6	9.8960e-1 –	9.9245e-1 –	9.9142e-1 –	9.7062e-1 –	9.6793e-1 –	9.9513e-1 ≈	9.9134e-1 –	9.9858e-1 +	9.9569e-1 ≈	9.9590e-1 ≈	9.9544e-1
	12	9.9421e-1 ≈	9.8919e-1 –	9.8891e-1 –	9.5588e-1 –	9.8686e-1 –	9.9509e-1 ≈	9.9211e-1 –	9.9868e-1 +	9.9317e-1 ≈	9.9357e-1 ≈	9.9434e-1
WFG3	6	7.8516e-2 –	9.5916e-2 –	8.2482e-2 –	2.8162e-2 –	8.5468e-2 –	1.3857e-1 ≈	2.1952e-2 –	1.3566e-1 –	1.9544e-1 +	2.1897e-1 +	1.4748e-1
	12	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	1.0848e-4 ≈	1.1551e-2 +	1.0592e-2 +	0.0000e+0
+/ – / ≈		7/23/6	2/29/5	11/22/3	8/24/4	8/19/9	2/19/15	11/22/3	18/11/7	6/27/3	3/29/4	

Table 9. IGD mean values obtained by MaOEAs for regular and irregular problems with 6 and 12 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
DTLZ1	6	9.2021e-2 ≈	8.4843e-2 +	2.3419e-1 −	7.8071e-2 +	9.4702e-2 −	8.6282e-2 +	7.6800e-2 +	8.7706e-2 +	1.3544e-1 −	1.2063e-1 −	8.8648e-2
	12	2.3672e-1 ≈	1.6572e-1 +	9.8367e+0 −	1.6063e-1 +	1.6796e+1 −	1.7064e-1 +	1.5219e-1 +	1.0610e+0 −	2.4301e-1 −	5.6493e-1 −	1.8180e-1
DTLZ2	6	2.5289e-1 +	2.5655e-1 +	2.8941e-1 −	2.5294e-1 +	2.8070e-1 −	2.6232e-1 ≈	2.5287e-1 +	2.7195e-1 −	3.4770e-1 −	3.4517e-1 −	2.6196e-1
	12	6.2062e-1 −	5.8812e-1 −	5.7445e-1 −	5.9407e-1 −	6.7343e-1 −	5.8049e-1 −	5.7678e-1 −	5.3981e-1 +	6.9346e-1 −	7.2721e-1 −	5.6116e-1
DTLZ3	6	1.3544e+0 −	5.6344e-1 −	4.4314e+0 −	3.1174e-1 −	2.3332e+0 −	3.3263e-1 −	3.5694e-1 ≈	3.6036e-1 ≈	9.1239e-1 −	1.8997e+0 −	2.8505e-1
	12	6.8132e+0 −	2.2656e+0 −	5.0597e+2 −	6.9365e-1 +	4.8728e+1 −	7.9626e-1 ≈	1.5367e+0 ≈	1.8647e+1 −	9.9587e+0 −	1.5048e+1 −	7.6922e-1
DTLZ4	6	2.8283e-1 +	4.5510e-1 −	2.8568e-1 ≈	2.7482e-1 +	3.7482e-1 −	3.0168e-1 ≈	4.0065e-1 ≈	4.3322e-1 −	3.7248e-1 ≈	3.7755e-1 −	3.5461e-1
	12	6.4828e-1 −	6.6535e-1 −	5.6536e-1 +	6.0666e-1 −	7.0867e-1 −	9.9062e-1 −	5.8054e-1 ≈	5.7481e-1 +	6.8513e-1 −	7.3647e-1 −	5.8113e-1
WFG4	6	1.8399e+0 −	1.7079e+0 ≈	1.9497e+0 −	1.7639e+0 −	1.9009e+0 −	1.6924e+0 +	1.7653e+0 −	1.7877e+0 −	2.2624e+0 −	2.2663e+0 −	1.7062e+0
	12	7.9884e+0 −	7.0066e+0 −	6.9943e+0 −	7.8924e+0 −	6.6770e+0 ≈	6.6117e+0 +	8.0284e+0 −	6.6188e+0 +	9.4491e+0 −	9.8214e+0 −	6.6606e+0
WFG5	6	1.7881e+0 −	1.6716e+0 +	1.9302e+0 −	1.7580e+0 −	1.8082e+0 −	1.6628e+0 +	1.7560e+0 −	1.7880e+0 −	2.2754e+0 −	2.2738e+0 −	1.6780e+0
	12	7.8887e+0 −	6.7922e+0 −	7.0617e+0 −	7.8032e+0 −	6.6203e+0 −	6.4836e+0 +	7.8794e+0 −	6.7327e+0 −	9.4940e+0 −	9.2767e+0 −	6.5456e+0
WFG6	6	1.9082e+0 −	1.6943e+0 ≈	2.0174e+0 −	1.7869e+0 −	1.9331e+0 −	1.6804e+0 +	1.7681e+0 −	1.8208e+0 −	2.2206e+0 −	2.2256e+0 −	1.6964e+0
	12	8.1533e+0 −	6.9469e+0 −	8.0253e+0 −	8.1261e+0 −	7.0729e+0 −	6.5642e+0 +	8.1471e+0 −	6.6077e+0 ≈	8.9867e+0 −	9.0052e+0 −	6.6174e+0
WFG7	6	1.8326e+0 −	1.7047e+0 +	1.9445e+0 −	1.7710e+0 −	1.9100e+0 −	1.7033e+0 +	1.7727e+0 −	1.8330e+0 −	2.2325e+0 −	2.2174e+0 −	1.7225e+0
	12	8.0532e+0 −	6.9570e+0 −	7.0161e+0 −	7.9845e+0 −	6.9542e+0 −	6.5693e+0 +	8.1287e+0 −	6.6057e+0 +	8.7854e+0 −	9.2649e+0 −	6.6544e+0
WFG8	6	1.9072e+0 −	1.7289e+0 −	1.9766e+0 −	1.8964e+0 −	1.8968e+0 −	1.7296e+0 −	1.8092e+0 −	1.8235e+0 −	2.1936e+0 −	2.2225e+0 −	1.7174e+0
	12	7.6683e+0 −	7.1018e+0 −	7.6305e+0 −	7.8434e+0 −	7.0176e+0 −	6.6553e+0 ≈	7.9115e+0 −	6.8194e+0 −	9.2350e+0 −	9.5330e+0 −	6.7182e+0
WFG9	6	1.8074e+0 −	1.6895e+0 ≈	1.8257e+0 −	1.7583e+0 −	1.7948e+0 −	1.6712e+0 +	1.7589e+0 −	1.7557e+0 −	2.2628e+0 −	2.2694e+0 −	1.6874e+0
	12	7.3640e+0 −	6.7216e+0 −	6.5870e+0 −	7.6007e+0 −	6.3223e+0 +	6.3852e+0 +	7.6361e+0 −	6.2890e+0 +	8.9940e+0 −	9.1965e+0 −	6.4823e+0
DTLZ5	6	1.7826e-1 −	5.9886e-2 −	2.0957e-1 −	4.0610e-1 −	8.2439e-2 −	1.1762e-1 −	1.0056e-1 −	2.1850e-1 −	9.6182e-2 −	1.3076e-1 −	4.3175e-2
	12	2.2505e-1 −	1.6384e-1 −	4.1226e-1 −	3.5934e-1 −	7.6087e-1 −	1.5095e-1 −	1.3619e-1 −	4.0732e-1 −	2.2337e+0 −	1.3104e+0 −	4.6761e-2
DTLZ6	6	3.5435e-1 −	7.4762e-2 −	4.5288e-1 −	2.3973e-1 −	7.3773e-2 −	3.1688e-1 −	1.2910e-1 −	3.2025e-1 −	1.4865e-1 −	1.6176e-1 −	3.3337e-2
	12	1.0761e+0 −	2.2732e-1 −	1.6426e+0 −	2.2311e-1 −	7.4209e-1 −	4.3975e-1 −	1.3166e-1 −	1.1368e+0 −	7.0929e+0 −	9.6895e+0 −	5.1577e-2
DTLZ7	6	4.7204e-1 −	5.7405e-1 −	3.9649e-1 −	7.0864e-1 −	4.2372e-1 −	4.0084e-1 −	1.6609e+0 −	3.9754e-1 −	8.0952e-1 −	8.0555e-1 −	3.9208e-1
	12	3.8883e+0 −	1.6409e+0 ≈	1.5280e+0 ≈	2.3501e+0 −	1.4036e+0 +	1.4656e+0 +	3.3993e+0 −	1.2797e+0 +	4.0472e+0 −	3.8123e+0 −	1.6615e+0
IDTLZ1	6	1.0457e-1 −	9.7868e-2 −	8.2865e-2 +	2.2577e-1 −	9.7110e-2 −	8.6432e-2 −	8.6478e-2 −	9.0199e-2 −	1.3823e-1 −	1.4429e-1 −	8.5793e-2
	12	1.5144e-1 +	1.9271e-1 −	1.9099e-1 −	3.3170e-1 −	2.2536e-1 −	1.6751e-1 −	1.8593e-1 −	1.7016e-1 −	1.7839e-1 −	2.3345e-1 −	1.5663e-1
IDTLZ2	6	3.5943e-1 −	2.6271e-1 −	2.8496e-1 −	5.0005e-1 −	2.8311e-1 −	2.5951e-1 −	2.8277e-1 −	2.6730e-1 −	5.0992e-1 −	5.0562e-1 −	2.5696e-1
	12	7.0775e-1 −	5.6737e-1 −	5.3821e-1 −	7.5405e-1 −	5.5485e-1 −	5.3455e-1 −	5.3861e-1 −	5.1301e-1 −	7.7025e-1 −	8.8819e-1 −	5.0609e-1
WFG1	6	9.0031e-1 −	7.5272e-1 ≈	6.2559e-1 +	7.5454e-1 ≈	1.0746e+0 −	7.1186e-1 +	7.0906e-1 +	7.6343e-1 ≈	8.8146e-1 −	9.8154e-1 −	7.5934e-1
	12	1.5537e+0 +	1.4833e+0 +	1.5087e+0 +	1.3594e+0 +	3.2617e+0 −	1.3666e+0 +	1.5746e+0 +	1.8936e+0 +	2.1307e+0 +	1.8835e+0 +	2.3568e+0
WFG2	6	9.5801e-1 ≈	7.1505e-1 +	7.2809e-1 +	6.5041e-1 +	1.3250e+0 −	9.1899e-1 +	6.1402e-1 +	8.4208e-1 +	8.3409e-1 +	8.5580e-1 +	9.6741e-1
	12	1.5277e+0 +	1.5221e+0 +	1.4673e+0 +	1.4038e+0 +	2.4762e+0 ≈	2.5624e+0 ≈	1.3427e+0 +	1.9932e+0 +	2.6860e+0 ≈	2.5708e+0 ≈	2.6834e+0
WFG3	6	9.1208e-1 −	6.6106e-1 ≈	6.4881e-1 ≈	1.1953e+0 −	4.6910e-1 +	6.5129e-1 ≈	1.0812e+0 −	7.5831e-1 −	4.7057e-1 +	1.0221e+0 −	6.1212e-1
	12	2.9872e+0 ≈	3.2577e+0 −	2.0377e+0 +	6.4486e+0 −	1.1608e+0 +	3.1609e+0 −	4.1870e+0 −	2.6452e+0 ≈	6.3299e+0 −	7.3897e+0 −	2.6391e+0
+/ − / ≈		5/27/4	8/22/6	7/26/3	8/27/1	4/30/2	16/14/6	7/25/4	10/22/4	3/31/2	2/33/1	

Table 10. HV mean values obtained by MaOEAs for MaOP problems with 3 and 5 objectives.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	NMPSO	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
MaOP1	3	7.5790e-1 –	7.8099e-1 +	7.3849e-1 –	7.6798e-1 –	7.6805e-1 –	7.7927e-1 ≈	7.7923e-1 ≈	7.7380e-1 –	7.0003e-1 –	6.9778e-1 –	7.7917e-1
	5	3.0820e-1 –	3.9108e-1 –	3.1612e-1 –	3.0763e-1 –	2.7263e-1 –	3.9426e-1 ≈	3.8927e-1 –	3.6173e-1 –	1.7856e-1 –	1.7605e-1 –	3.9497e-1
MaOP2	3	9.5078e-1 –	9.5050e-1 –	8.5933e-1 –	9.4905e-1 –	8.5013e-1 –	9.5754e-1 ≈	9.3941e-1 –	9.4407e-1 –	9.1469e-1 –	8.9550e-1 –	9.5452e-1
	5	9.9474e-1 –	9.8966e-1 –	9.9240e-1 –	9.9436e-1 –	9.7693e-1 –	9.9615e-1 –	9.9712e-1 –	9.9775e-1 +	9.9304e-1 –	9.7906e-1 –	9.9755e-1
MaOP3	3	4.7811e-1 –	4.8469e-1 –	4.3004e-1 –	4.6513e-1 –	4.8151e-1 ≈	4.8383e-1 –	4.9182e-1 –	4.9160e-1 –	4.5692e-1 –	4.4047e-1 –	4.9204e-1
	5	7.8029e-1 –	7.2988e-1 –	7.5255e-1 –	7.5791e-1 –	7.9788e-1 +	7.6461e-1 –	7.8320e-1 –	7.9140e-1 –	7.0169e-1 –	6.6104e-1 –	7.9265e-1
MaOP4	3	4.0478e-1 –	4.0441e-1 –	3.4368e-1 –	4.7829e-1 +	4.8928e-1 +	4.1639e-1 +	4.0716e-1 –	4.2010e-1 +	3.8946e-1 –	3.8880e-1 –	4.1295e-1
	5	7.1489e-1 –	6.9510e-1 –	7.2493e-1 –	7.3009e-1 –	7.7867e-1 +	7.5733e-1 ≈	7.2011e-1 –	7.2885e-1 –	6.9567e-1 –	6.7605e-1 –	7.5936e-1
MaOP5	3	4.6641e-1 –	4.7349e-1 –	4.3631e-1 –	4.6530e-1 –	4.7252e-1 –	4.7990e-1 –	4.6035e-1 –	4.5699e-1 –	4.3157e-1 –	4.1595e-1 –	4.8226e-1
	5	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0
MaOP6	3	1.0128e-1 –	1.5675e-1 –	1.3535e-1 –	2.0076e-1 –	1.9764e-1 –	2.1245e-1 ≈	1.4630e-1 –	1.7592e-1 –	2.0815e-1 –	1.8066e-1 –	2.1575e-1
	5	2.7884e-3 ≈	5.3749e-3 +	3.8229e-3 ≈	1.1798e-3 –	4.9251e-3 +	1.7069e-3 –	6.3655e-3 +	2.2539e-3 –	2.5593e-3 ≈	4.0955e-3 +	3.2113e-3
+/ − / ≈		0/10/2	2/9/1	0/10/2	1/10/1	4/6/2	1/5/6	1/9/2	2/9/1	0/10/2	1/10/1	

Table 11. HV mean values obtained by MaOEAs on real-world problems.

Problem	M	ANSGA-III	MOEA/D-AWA	KnEA	RVEA	MOEA/D-URAW	AR-MOEA	PREA	FDEA-I	FDEA-II	MOEA/D-UR
MOTSP	3	4.7082e-1 –	4.9867e-1 –	5.5460e-1 –	4.7754e-1 –	5.1136e-1 –	4.7518e-1 –	5.1186e-1 –	4.7768e-1 –	4.7927e-1 –	5.9406e-1
	5	1.9298e-1 –	2.3590e-1 –	3.0290e-1 –	2.0350e-1 –	2.5914e-1 –	2.1198e-1 –	2.6334e-1 –	2.2602e-1 –	2.2137e-1 –	3.5614e-1
	8	6.2001e-2 –	6.3973e-2 –	6.2886e-2 –	7.0826e-2 –	7.8707e-2 –	6.7378e-2 –	7.9808e-2 –	7.4844e-2 –	7.5435e-2 –	1.0789e-1
	10	1.9293e-2 –	2.2005e-2 –	1.7809e-2 –	1.9984e-2 –	2.7107e-2 –	2.2563e-2 –	2.6891e-2 –	2.7254e-2 –	2.5211e-2 –	3.7151e-2
	15	2.6552e-3 –	2.0438e-3 –	1.1752e-3 –	2.6650e-3 –	2.6601e-3 –	2.3330e-3 –	2.7191e-3 –	3.2030e-3 –	3.0789e-3 –	3.5380e-3
MOKP	3	3.3789e-1 –	3.1871e-1 –	3.1912e-1 –	3.1299e-1 –	3.2194e-1 –	3.3315e-1 –	3.0642e-1 –	3.3582e-1 –	3.3562e-1 –	3.4930e-1
	5	1.1894e-1 –	1.0692e-1 –	1.2118e-1 –	1.0724e-1 –	1.1569e-1 –	1.2141e-1 –	1.1003e-1 –	1.2083e-1 –	1.1947e-1 –	1.3460e-1
	8	2.1332e-2 –	1.7040e-2 –	2.1569e-2 –	1.8280e-2 –	2.0522e-2 –	2.1968e-2 –	2.1176e-2 –	2.1387e-2 –	2.1266e-2 –	2.4990e-2
	10	6.7444e-3 –	5.1663e-3 –	8.4908e-3 +	5.5203e-3 –	6.3497e-3 –	7.1617e-3 –	6.7657e-3 –	7.1792e-3 –	6.9593e-3 –	8.1975e-3
	15	4.2696e-4 –	2.7198e-4 –	3.6056e-4 –	3.2097e-4 –	3.5157e-4 –	4.2401e-4 –	4.0106e-4 –	4.3091e-4 –	4.1740e-4 –	4.8525e-4
WRP	5	9.6427e-2 +	7.9784e-2 –	9.8096e-2 +	5.3258e-2 –	8.1591e-2 –	9.5304e-2 +	9.3731e-2 ≈	6.9736e-2 –	5.7315e-2 –	9.3000e-2
+/ – / ≈		1/10/0	0/11/0	2/9/0	0/11/0	0/11/0	1/10/0	0/10/1	0/11/0	0/11/0	