axioms

MDPI

Article

An Improved MOEA/D with an Auction-Based Matching Mechanism

Guangjian Li [1,†], Mingfa Zheng [2,*], Guangjun He [1,†], Yu Mei [2], Gaoji Sun [3] and Haitao Zhong [2]

[1] Air Defense and Anti-Missile School, Air Force Engineering University, Xi'an 710051, China
[2] Fundamentals Department, Air Force Engineering University, Xi'an 710051, China; meiyu414@stu.xjtu.edu.cn (Y.M.)
[3] College of Economics and Management, Zhejiang Normal University, Jinhua 321004, China
* Correspondence: mingfazheng@stu.xjtu.edu.cn; Tel.: +86-13201855000
† These authors contributed equally to this work.

Abstract: Multi-objective optimization problems (MOPs) constitute a vital component in the field of mathematical optimization and operations research. The multi-objective evolutionary algorithm based on decomposition (MOEA/D) decomposes a MOP into a set of single-objective subproblems and approximates the true Pareto front (PF) by optimizing these subproblems in a collaborative manner. However, most existing MOEA/Ds maintain population diversity by limiting the replacement region or scale, which come at the cost of decreasing convergence. To better balance convergence and diversity, we introduce auction theory into algorithm design and propose an auction-based matching (ABM) mechanism to coordinate the replacement procedure in MOEA/D. In the ABM mechanism, each subproblem can be associated with its preferred individual in a competitive manner by simulating the auction process in economic activities. The integration of ABM into MOEA/D forms the proposed MOEA/D-ABM. Furthermore, to make the appropriate distribution of weight vectors, a modified adjustment strategy is utilized to adaptively adjust the weight vectors during the evolution process, where the trigger timing is determined by the convergence activity of the population. Finally, MOEA/D-ABM is compared with six state-of-the-art multi-objective evolutionary algorithms (MOEAs) on some benchmark problems with two to ten objectives. The experimental results show the competitiveness of MOEA/D-ABM in the performance of diversity and convergence. They also demonstrate that the use of the ABM mechanism can greatly improve the convergence rate of the algorithm.

Keywords: MOEA/D; multi-objective optimization problem; auction theory; weight vectors adjustment

MSC: 68 Computer Science; 90 Operations research, mathematical programming

check for updates

Citation: Li, G.; Zheng, M.; He, G.; Mei, Y.; Sun, G.; Zhong, H. An Improved MOEA/D with an Auction-Based Matching Mechanism. *Axioms* **2024**, *13*, 644. https://doi.org/10.3390/axioms13090644

Academic Editor: Behzad Djafari-Rouhani

Received: 13 August 2024
Revised: 11 September 2024
Accepted: 13 September 2024
Published: 20 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Multi-objective optimization problems (MOPs) constitute an important branch of mathematical optimization and operations research, which are characterized by the existence of multiple conflicting objectives. Due to its pervasive presence in practical applications, MOP has been of keen interest to researchers in recent years. Multi-objective evolutionary algorithms (MOEAs) are a class of intelligence algorithms based on population evolution for dealing with MOPs. Owing to a flexible framework, MOEAs have strong applicability for solving MOPs with various characteristics and have been widely used in many fields, such as trajectory planning [1], flood control scheduling [2], rail vehicle systems [3], transmission gear design [4], and job shop scheduling [5].

According to the selection criteria of population, existing MOEAs can be classified into three categories: dominance-based MOEAs [6–8], indicator-based MOEAs [9–12], and decomposition-based MOEAs [13–15]. In the context of dominance-based MOEAs, the non-dominance sorting rank and crowded degree are used to evaluate the fitness

of individuals, determining the reserved population in the evolution process. However, when the number of objectives is large, the proportion of non-dominated solutions in the population could be significant, which leads to the loss of selection pressure. As for the indicator-based MOEAs, the performance indicators, such as hypervolume [16] and inverted generational distance [17], are used to evaluate the fitness of individuals. These algorithms can overcome the challenges brought by the loss of selection pressure, but their high computational complexity is a crucial issue, resulting in excessive runtime [12].

As a new class of MOEAs, the decomposition-based MOEAs (MOEA/Ds) have attracted significant attention from many scholars due to their good generality and low computational complexity. MOEA/Ds utilize a set of weight vectors to decompose a MOP into a series of simpler subproblems, and then, the population approximates the Pareto front (PF) by optimizing these subproblems in parallel. Since the primal MOEA/D was proposed in [13], researchers have developed numerous variant versions, which made improvements on various aspects such as the decomposition method [18–23], reproduction strategy [24–26], replacement procedure [27–31], computational resource allocation [32,33], and weight vector adjustment [34–38].

In the original MOEA/D [13], the replacement procedure solely driven by the scalarizing function value is the primary force to encourage convergence, enabling the population to approximate the PF. However, this approach has a detrimental effect on the diversity of the population. To better balance convergence and diversity, some efforts have been made on the replacement mechanism, such as adaptive neighborhood [39], stable-state neighborhood [31], adaptive replacement strategy [40], distance-based updating [28], improved global replacement [41], stable matching model [27], and incomplete matching list [42]. Specifically, some other categories, including MOEA/DD [43] and MOEA/DDS [44], exploit the advantages of different selection criteria. Despite their contributions to enhancing algorithm performance, these approaches also exhibit some drawbacks. Most of the existing methods limit the region or scale of replacement to preserve the diversity of the population, which may exclude more reasonable association patterns between subproblems and individuals, resulting in insufficient convergence efficiency. As illustrated in Figure 1, the weight vectors λ^1, λ^2, λ^3, and λ^4 are associated with the individuals x^1, x^2, x^3, and x^4, respectively. Comparing x^3 with the new individual y, it is evident that y exhibits better convergence for λ^3, yet y falls outside the replacement region (marked in gray) constrained by diversity estimation, such as crowding distance, vector angle, or niche technique. This phenomenon occurs frequently during evolution. In contrast, the previously proposed stable matching model [27] and incomplete matching list [42] enable one-to-one global matching between subproblems and individuals, achieving a better equilibrium between convergence and diversity. However, these matching-based replacement mechanisms, which lack competition, have a high risk of matching an individual with an unpreferred subproblem, ultimately leading to imbalanced association outcomes. In fact, if we consider the subproblems and individuals as two agent sets, the replacement procedure in MOEA/D can be regarded as a bilateral selection process: each subproblem prefers an individual that possesses a better scalarizing function value to enhance convergence, while each individual aims to avoid repetitive associations to maintain diversity. From this perspective, the auction concept, originating from the economic field, can be adopted as a natural framework to model this selection process. By treating the subproblems as bidders and the individuals as goods to be allocated, the auction mechanism introduces an efficient and rational way to coordinate the replacement procedure. Distinguished from existing approaches, the competitive nature of auctions encourages subproblems to compete for their preferred individuals among the whole population, leading to a more balanced and effective association. This paper presents a first attempt along this line.

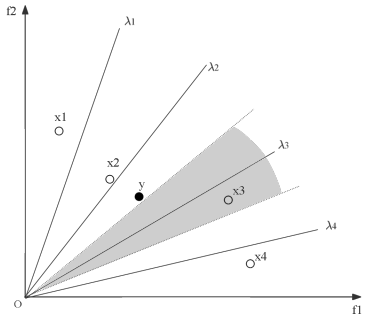


Figure 1. The illustration of the limited replacement region.

Aside from research on the replacement procedure, the approach of adjusting weight vectors is another aspect of improving the performance of MOEA/D. Relevant studies showed that the final non-dominated solutions obtained by MOEA/D significantly depend on the distribution of weight vectors, and the best results can only be achieved when the distribution of weight vectors is consistent with the shape of PF [45–47]. Generally, utilizing a series of fixed uniform weight vectors is inefficient, especially when dealing with MOPs with irregular PFs that are degenerate, convex, discontinuous, or sharp-tailed. To achieve the appropriate distribution of weight vectors, adaptively adjusting them during evolution is a promising method that has been adopted by many scholars [14,34–38,48–51]. Although many effective strategies were designed to adjust the weight vectors, their trigger timing was predefined and devoid of an instructor during evolution, potentially coming at an inappropriate moment. To address this issue, Dong [52] utilized a measure called population activity to determine the trigger timing of weight vector adjustment. Similarly, Lucas [45] presented MOEA/D-UR in which an improvement metric is defined to determine when to adjust the weight vectors. These research results indicated that the adjustment strategy with measure-dependent trigger timing is more effective than that with predefined trigger timing. Therefore, we drew inspiration from these methods to design an improved weight vector adjustment strategy in MOEA/D, whose trigger timing is determined by the convergence activity of the population.

Based on the above crucial discussion, an improved MOEA/D with an auction-based matching mechanism (ABM), called MOEA/D-ABM, is proposed in this paper. As an application of auction theory [53], the ABM mechanism is the main innovation of this paper. Meanwhile, several advanced techniques were incorporated into MOEAD/-ABM, including the dynamic normalization method and the weight vector adjustment strategy, to further enhance the performance of the algorithm. The main contribution of this paper can be summarized as follows:

(1) A novel auction-based matching mechanism is proposed to match the weight vectors (subproblems) with current solutions (individuals) in each generation. Specifically, the weight vectors and solutions are regarded as the bidders and the auctioned commodities, respectively. By simulating the auction process in economic activities, each weight vector can be associated with its preferred solution that possesses a better scalarizing function value in a competitive manner. This mechanism leverages the strengths of free competition to encourage convergence in a fair and efficient way. Meanwhile, the one-to-one matching model avoids the presence of repeated solutions in the population, maintaining diversity. We demonstrate the rationality of this mechanism from a mathematical analysis perspective.

(2) A dynamic normalization method is employed to normalize the objectives. Commonly, the ideal and nadir points are estimated to normalize the objectives, lessening the influence of distinct scale of objectives on the evolution, especially when solving many-objective problems. However, in early evolution, the estimated nadir point significantly differs from the true one, which may disrupt the appropriate search direction [54]. In the dynamic normalization method, we decouple the estimated nadir point from the normalization procedure at the initial stage evolution, and

gradually increase its intensity of the estimated nadir as the evolution progresses. This approach can effectively avoid the above disruption.

(3) A weight vector adjustment mechanism based on the sparsity level of the population is adopted, and a measure based on the convergence activity of the population is defined to periodically identify the trigger timing of adjustment. Low-level population convergence activity indicates difficulty in further optimizing the current subproblems, necessitating adjusting the weight vectors. During the adjustment process, crowded individuals and their associated weight vectors are deleted, and, subsequently, sparse non-dominated solutions and the newly generated weight vectors are added. This strategy helps to estimate the shape of PF comprehensively and improve the diversity of the final output solutions.

(4) Finally, we apply MOEA/D-ABM compared with six state-of-the-art MOEAs to solve some benchmark problems including the DTLZ test suit [55], WFG test suit [56], and BT test suit [57]. The experimental results show the competitiveness of our algorithm in both diversity and convergence. Especially when dealing with the BT test suit which poses significant convergence challenges, MOEA/D-ABM significantly outperforms the other compared MOEAs in terms of convergence rate. The results demonstrate that integrating ABM into MOEA/D can significantly enhance the convergence efficiency of the algorithm while maintaining the diversity.

The remaining structure of this paper is arranged as follows. In Section 2, some basic knowledge about multi-objective optimization problem and MOEA/D algorithm is introduced. Section 3 describes our proposed MOEA/D-ABM in detail. In Section 4, we implement MOEA/D-ABW and six comparison MOEAs to solve some MOP test suits. The experimental results are analyzed to verify the effectiveness of our proposed algorithm. Finally, the work of this paper is concluded in Section 5.

2. Background

In this section, some basic knowledge about multi-objective optimization problems (MOPs) and the MOEA/D is introduced.

2.1. Multi-Objective Optimization Problem

An MOP is characterized by an optimization problem with two or more objectives. Without loss of generality, a minimized MOP with m objectives can be formulated as follows:

$$\begin{array}{ll} \min & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x}))^\top, \\ s.t. & \boldsymbol{x} \in \Omega, \end{array} \tag{1}$$

where Ω is the feasible region of decision variable $\boldsymbol{x}$. Normally, $\boldsymbol{x} = (x_1, x_2, ..., x_n)^\top$ is an n-dimensional vector in the space $\mathbb{R}^n$.

Generally, there is no absolutely optimal solution that optimizes all objectives in problem (1) simultaneously due to the conflicts among objectives. Therefore, a set of Pareto-optimal solutions representing the tradeoffs among the objectives are considered, and the relevant definitions given as follows.

Definition 1 ([58] (Pareto dominance)). *Let $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \Omega$; we say that $\boldsymbol{x}^1$ Pareto dominates (or dominates) $\boldsymbol{x}^2$ if and only if*

$$\forall i \in \{1, 2, ..., m\}, f_i(\boldsymbol{x}^1) \leqslant f_i(\boldsymbol{x}^2),$$

and

$$\exists j \in \{1, 2, ..., m\}, f_j(\boldsymbol{x}^1) < f_j(\boldsymbol{x}^2),$$

which is denoted by $\boldsymbol{x}^1 \prec \boldsymbol{x}^2$.

Definition 2 ([58] (Pareto-optimal solution)). *The decision variable $\boldsymbol{x}^* \in \Omega$ is called a Pareto-optimal solution if there is no $\boldsymbol{x} \in \Omega$ such that $\boldsymbol{x} \prec \boldsymbol{x}^*$.*

Definition 3 ([58] (Pareto-optimal set)). *The set of all Pareto-optimal solutions is called the Pareto-optimal set, which can be formulated as follows:*

$$PS = \{\boldsymbol{x} \in \Omega | \neg \exists \boldsymbol{x}' \in \Omega, \boldsymbol{x}' \prec \boldsymbol{x}\}.$$

Definition 4 ([58] (Pareto front)). *The images of Pareto solutions are called the Pareto front, which can be formulated as follows:*

$$PF = \{\boldsymbol{f}(\boldsymbol{x}) | \boldsymbol{x} \in PS\}.$$

Definition 5 ([30] (Ideal point and nadir point)). *The ideal point $\boldsymbol{z}^* = (z_1^*, z_2^*, ..., z_m^*)^\top$ and nadir point $\boldsymbol{z}^{nad} = (z_1^{nad}, z_2^{nad}, ..., z_m^{nad})^\top$ are two objective vectors defined as follows:*

$$z_i^* = \min_{\boldsymbol{x} \in \Omega} f_i(\boldsymbol{x}), \quad z_i^{nad} = \max_{\boldsymbol{x} \in PS} f_i(\boldsymbol{x}).$$

2.2. Multi-Objective Evolutionary Algorithm Based on Decomposition

The original MOEA/D was proposed by Zhang [13]. It decomposes a MOP into a series of single-objective optimization subproblems through a scalarizing function and a set of weight vectors, and then optimizes these subproblems collaboratively such that the population converges to the real PF. The basic framework of MOEA/D is shown in Algorithm 1, which is composed of population initialization, weight vector generation, offspring reproduction, and elite selection. The decomposition method and weight vector generation play the crucial roles in MOEA/D, which provide technical support for facilitating the convergence and maintaining the diversity of the population. They are briefly introduced below.

Algorithm 1: MOEA/D

Input: Population size N.
Output: Population $\boldsymbol{P}$.
1 Initialize N feasible solutions randomly to form the population $\boldsymbol{P} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^N\}$, and evaluate their objectives $\boldsymbol{f}(\boldsymbol{x}^i)$ $(i = 1, ..., N)$. Initialize the ideal point $\boldsymbol{z}^*$, where $z_i^* = \min_{x \in \boldsymbol{P}} f_i(\boldsymbol{x})$;
2 Generate N weight vectors $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^N)$ to be associated with the solutions in $\boldsymbol{P}$, and initialize the neighborhood $\boldsymbol{B}(i)$ for each individual $(i = 1, ..., N)$.
3 **while** *the termination condition is not met* **do**
4 **for** *i=1:N* **do**
5 $\boldsymbol{o} \leftarrow$Reproduce an offspring for $\boldsymbol{x}^i$;
6 Evaluate $\boldsymbol{f}(\boldsymbol{o})$; Replace the solutions in $\boldsymbol{B}(i)$ with $\boldsymbol{o}$ according to the scalarizing function; Update the ideal point $\boldsymbol{z}^*$.
7 **end**
8 **end**

2.2.1. Decomposition Method

There are four decomposition methods widely used in MOEA/Ds: penalty-based boundary intersection [13], weighted sum, L_p scalarizing and Tchebycheff [18]. The main distinctions among them are reflected in the scalarizing function, which is to evaluate the fitness of different solutions for the subproblems in the evolution process. Pescador [59] provided an overview of these scalarizing functions and gave some suggestions for their usage. The algorithm to be proposed takes the Tchebycheff decomposition method, since its optimal solution is not limited to the shape of PF.

Generally, the objectives in a MOP have distinct measures, which may compromise the uniformity of the final output. To address this issue, the normalization procedure based on the ideal and nadir points should be applied to each objective as follows:

$$\bar{f}_i = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*}, \forall i \in \{1, 2, ..., m\}, \tag{2}$$

where z_i^* and z_i^{nad} are the elements of the ideal point and nadir point, respectively; f_i is the original objective and $\bar{f}_i$ is the corresponding normalized objective. Next, the scalarizing function of the Tchebycheff method can be expressed as follows [18]:

$$\min_{\boldsymbol{x} \in \Omega} g^{TCH}(\boldsymbol{x} \big| \boldsymbol{\lambda}, \boldsymbol{z}^*, \boldsymbol{z}^{nad}) = \max_{1 \le i \le m} \lambda_i \big| \bar{f}_i(\boldsymbol{x}) \big|, \tag{3}$$

where $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_m)^\top$, $\lambda_i \ge 0$ for $\forall i \in \{1, ..., m\}$ and $\sum_{i=1}^m \lambda_i = 1$. It has been proved that the Tchebycheff method allows any Pareto-optimal solution to be achieved by altering weight vectors [60].

2.2.2. Weight Vector Generation

In most MOEA/Ds, each weight vector corresponds to a unique subproblem, determining a convergence tendency of the population. There are, mainly, four methods to generate the weight vectors, which are introduced as follows:

(1) Das and Dennis method
The Das and Dennis method [61] is widely used in MOEA/Ds to generate evenly distributed weight vectors. The number of its generated weight vectors is $N = C_{H+m-1}^{m-1}$, where m is the number of objectives and H is the number of divided points of each axis. However, its drawback is that the value of N cannot be flexibly set.

(2) Jaszkiewicz method
The Jaszkiewicz method [62] can generate a series of weight vectors in a random manner, as shown in the following formula:

$$\begin{aligned} &\lambda_1 = 1 - (\mathcal{U}(0,1))^{1-m}, ... \\ &\lambda_i = \left(1 - \sum_{j=1}^{i-1} \lambda_j\right)\left(1 - (\mathcal{U}(0,1))^{j+1-m}\right), ... \\ &\lambda_m = 1 - \sum_{j=1}^{m-1} \lambda_j \,, \end{aligned} \tag{4}$$

where $\mathcal{U}(0,1)$ is a uniform sampling function in the range of $[0,1]$. It satisfies $\sum_{i=1}^m \lambda_i = 1$. The number of generated weight vectors is flexibly set.

(3) Uniformly Random method
The Uniformly Random (UR) method [63] is another method to generate weight vectors in a random manner, which can be described by the following steps. Firstly, 5000 weight vectors are randomly generated to form set $\boldsymbol{\Lambda}_1$, and m boundary vectors are initialized to form set $\boldsymbol{\Lambda}$. Secondly, the element in $\boldsymbol{\Lambda}_1$ with the largest distance to $\boldsymbol{\Lambda}$ is moved to $\boldsymbol{\Lambda}$. The second step is repeated until the number of elements in $\boldsymbol{\Lambda}$ reaches N (the number of generated weight vectors). Finally, $\boldsymbol{\Lambda}$ is returned.

(4) Tchebycheff method
The Tchebycheff method [18] generates a new weight vector based on the objectives of the added solution, which is widely adopted in weight vector adjustment strategies. The generation formula of the weight vector is shown as follows:

$$\lambda_i = \frac{\frac{1}{\bar{f}_i(\boldsymbol{x})}}{\sum_{j=1}^m \frac{1}{\bar{f}_j(\boldsymbol{x})}}, \forall i = 1, ..., m, \tag{5}$$

where $\bar{f}_j(\boldsymbol{x})$ is the normalized j-th dimension objective of the added solution $\boldsymbol{x}$, as shown in (2).

In the MOEA/D-ABM proposed in this paper, the UR method was utilized to initialize weight vectors due to its good uniformity and simplicity, and the Tchebycheff method was used in the adjustment of weight vectors.

3. Proposed Algorithm

In this section, we first present the overall framework of MOEA/D-ABM. After that, we give the detailed instruction of several crucial components. The major innovation of our proposed algorithm can be embodied in the auction-based matching (ABM) mechanism. Moreover, the dynamic normalization method and weight vector adjustment strategy are employed to further enhance the performance of the algorithm.

3.1. The Framework of MOEA/D-ABM

The main framework of MOEA/D-ABM is depicted in Algorithm 2.

Initially, N weight vectors are generated by the UR method, and then, WS-transformation [34] is applied to them. The neighborhood of each weight vector $\boldsymbol{B}(i)$ is composed of the T weight vectors closest to it. N feasible solutions are randomly generated to form the population $\boldsymbol{P}$. The ideal point $\boldsymbol{z}^*$ and nadir point $\boldsymbol{z}^{nad}$ are estimated by the minimum and maximum value of each objective among $\boldsymbol{P}$, respectively. An external population, denoted as $\boldsymbol{EP}$, is initialized as an empty set to store the non-dominated solutions during the evolution process.

As the iteration proceeds, for each individual $\boldsymbol{x}^i \in \boldsymbol{P}$, the mating pool $\boldsymbol{E}$ is chosen from its neighborhood $\boldsymbol{B}(i)$ or the whole population $\boldsymbol{P}$ with a selection probability of δ for $\boldsymbol{B}(i)$, and a new offspring $\boldsymbol{o}$ is generated by simulating binary crossover (SBX) [64] and polynomial mutation operators on $\boldsymbol{E}$ (Algorithm 2, lines 6–11). After that, $\boldsymbol{o}$ is evaluated and the ideal point is updated, and, at most, nr solutions in $\boldsymbol{E}$ are replaced by $\boldsymbol{o}$ if their associated scalarizing function values can be further optimized (Algorithm 2, lines 12–14). It should be noted that, in the calculation of scalarizing function value, a dynamic normalization procedure is employed to control the intensity of the nadir point. After the offspring reproduction and replacement, the non-dominated solutions stored in $\boldsymbol{EP}$ is updated based on the offspring set $\boldsymbol{O}$, and the nadir point $\boldsymbol{z}^{nad}$ is re-estimated by the maximum value of each objective among $\boldsymbol{EP}$ (Algorithm 2, lines 16–17). Furthermore, in every generation, the ABM mechanism is incorporated to rematch the weight vectors (subproblems) with all current solutions in the union set $\boldsymbol{U}$, aiming to enhance the convergence of the population (Algorithm 2, lines 18–20). Additionally, for every 5% during 10–90% of the whole evolution process, a measure of convergence activity, denoted as Ca, is calculated to determine whether or not to adjust the weight vectors (Algorithm 2, lines 22–27). If $Ca < \rho$, nus weight vectors are adjusted based on the distribution of $\boldsymbol{EP}$. Finally, the solutions in $\boldsymbol{P}$ are output.

The following subsections provide detailed instruction for several components of MOEA/D-ABM.

Algorithm 2: MOEA/D-ABM

Input: Population size N; maximum number of function evaluations $maxFE$; neighborhood size T; selection probability δ; replacement size nr; adjustment size nus; convergence activity coefficient ρ
Output: Population $\boldsymbol{P}$

Initialize the weight vector set $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^N\}$ by UR method; apply the WS-transformation on $\boldsymbol{\Lambda}$; determine the neighborhood $\boldsymbol{B}(i)$ for each $\boldsymbol{\lambda}^i \in \boldsymbol{\Lambda}$; initialize the population $\boldsymbol{P} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^N\}$ randomly; estimate the ideal point $\boldsymbol{z}^*$ and nadir point $\boldsymbol{z}^{nad}$ according to $\boldsymbol{P}$;
$FE \leftarrow N$; $\boldsymbol{EP} \leftarrow \emptyset$;
$Gen_{max} = \text{ceil}(maxFE/N)$;
while $FE < maxFE$ **do**
  for $i = 1$ *to* N **do**
    if $rand < \delta$ **then**
      $\boldsymbol{E} \leftarrow \boldsymbol{B}(i)$;
    else
      $\boldsymbol{E} \leftarrow \{1, ..., N\}$;
    end
    $\boldsymbol{o} \leftarrow$Randomly select mating solutions from $\boldsymbol{E}$ to generate an offspring by SBX and polynomial mutation;
    Evaluate $\boldsymbol{f}(\boldsymbol{o})$; $FE \leftarrow FE + 1$; $\boldsymbol{O} \leftarrow \boldsymbol{O} \cup \{\boldsymbol{o}\}$;
    $\boldsymbol{z}^* \leftarrow \min(\boldsymbol{z}^*, \boldsymbol{f}(\boldsymbol{o}))$;
    $[\boldsymbol{P}] \leftarrow$Replacement$(\boldsymbol{o}, \boldsymbol{P}, \boldsymbol{E}, \boldsymbol{\Lambda}, \boldsymbol{z}^*, \boldsymbol{z}^{nad}, nr)$ (Algorithm 3);
  end
  $[\boldsymbol{EP}] \leftarrow$UpdateEP$(\boldsymbol{O}, \boldsymbol{EP})$ (Algorithm 4);
  $\boldsymbol{z}^{nad} \leftarrow \max_{\boldsymbol{x} \in \boldsymbol{EP}}(\boldsymbol{f}(\boldsymbol{x}))$;
  $\boldsymbol{U} \leftarrow \boldsymbol{P} \cup \boldsymbol{EP}$;
  Delete the repeated solutions in $\boldsymbol{U}$;
  $[\boldsymbol{P}] \leftarrow$AuctionBasedMatch$(\boldsymbol{U}, \boldsymbol{\Lambda})$ (Algorithm 5);
  $t = ceil(FE/N)$;
  if $t \geqslant 10\% \times Gen_{max}$ **and** $t \leqslant 90\% \times Gen_{max}$ **and** $mod(t/Gen_{max}, 5\%) == 0$ **then**
    Calculate Ca by (23);
    if $Ca < \rho$ **then**
      $[\boldsymbol{P}, \boldsymbol{\Lambda}, \boldsymbol{B}] =$AdjustWeights($\boldsymbol{P}$,$\boldsymbol{EP}$,$\boldsymbol{\Lambda}$, nus) (Algorithm 6);
    end
  end
end

Algorithm 3: Replacement$(\boldsymbol{o}, \boldsymbol{P}, \boldsymbol{E}, \boldsymbol{\Lambda}, \boldsymbol{z}^*, \boldsymbol{z}^{nad}, nr)$

Input: Offspring $\boldsymbol{o}$; population $\boldsymbol{P}$; mating pool $\boldsymbol{E}$; weight vectors $\boldsymbol{\Lambda}$; ideal point $\boldsymbol{z}^*$; nadir point $\boldsymbol{z}^{nad}$; replacement size nr.
Output: Updated population $\boldsymbol{P}$.

$update \leftarrow 0$;
while $update < nr$ *and* $\boldsymbol{E} \neq \emptyset$ **do**
  Randomly select $j \in \boldsymbol{E}$; $\boldsymbol{E} \leftarrow \boldsymbol{E} \backslash j$;
  if $g^{TCH}(\boldsymbol{o}|\boldsymbol{\lambda}^j, \boldsymbol{z}^*, \boldsymbol{z}^{nad}) < g^{TCH}(\boldsymbol{x}^j|\boldsymbol{\lambda}^j, \boldsymbol{z}^*, \boldsymbol{z}^{nad})$ **then**
    $\boldsymbol{x}^j \leftarrow \boldsymbol{o}$; $update \leftarrow update + 1$;
  end
end

3.2. Weight Vector Initialization

We used the UR method introduced in Section 2.2.2 to the initial N uniformly distributed weight vectors, denoted as $\boldsymbol{\Lambda}$=$\{\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^N\}$, due to its simplicity and good performance in uniformity. The optimal solution of Tchebycheff scalarizing function is the intersection point of vector $(1/\lambda_1, ..., 1/\lambda_m)^\top$ and normalized PF; thus, the WS-transformation is applied on the generated weight vectors. In this method, the weight vector, $(\lambda_1, ..., \lambda_m)^\top$, is transformed as follows:

$$\boldsymbol{\lambda}' = WS(\boldsymbol{\lambda}) = \left(\frac{\frac{1}{\lambda_1}}{\sum_{j=1}^{m} \frac{1}{\lambda_j}}, ..., \frac{\frac{1}{\lambda_m}}{\sum_{j=1}^{m} \frac{1}{\lambda_j}} \right)^\top. \tag{6}$$

Due to the existence of boundary weight vectors, λ_j is replaced by 10^{-6} when λ_j=0.

3.3. External Population

EP is defined to store the non-dominated solutions during the evolution process. The distribution of ***EP*** resembles that of the true PF to some extent; thus, it is used to provide guidance for weight vector adjustment. The more solutions ***EP*** includes, the more suitable it is for weight vector adjustment, but this leads to a high computational endeavor. As a compromise, we set the size of ***EP*** to twice that of the population, i.e., $|\boldsymbol{EP}| = 2N$. See Algorithm 4 for the update of ***EP***. If an offspring is non-dominated among ***EP***, we add it to ***EP*** and delete any one dominated by it (Algorithm 4, lines 1–5). If the number of solutions in ***EP*** exceeds $2N$, we delete the most crowded one in ***EP*** until its size is equal to $2N$. The sparsity level is utilized to measure the crowded degree of a solution among a population by the following formulation [65]:

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

where $L_2^{NN_i^j}$ denotes the Euclidean distance in the normalized objective space between the solution, ind^j, and its i-th nearest solutions in the population, pop; m is the number of objectives.

Algorithm 4: UpdateEP($\boldsymbol{O}, \boldsymbol{EP}$)

Input: External population ***EP***; offsprings set ***O***.
Output: Updated external population ***EP***.

```
1 for each x ∈ O do
2     if ¬∃q ∈ EP, q ≺ x then
3         EP ← EP ∪ {x}; EP ← EP\{q ∈ EP | x ≺ q}
4     end
5 end
6 while |EP| > 2N do
7     Calculate the sparsity level of each individual in EP among EP by (7);
8     Delete the individual with the lowest sparsity level in EP.
9 end
```

3.4. Dynamic Normalization Procedure

In the calculation of the scalarizing function value given in (2), the ideal point and nadir point need to be estimated. The estimation of the ideal point could be the minimum value of each objective found in the evolution process. On the other hand, the estimation of the nadir point could be the maximum value of each objective in the non-dominated solutions, ***EP***. However, ***EP*** has little reliable information about the true PF in the early generations; thus, the estimated nadir point may be significantly different from the true one, resulting in degradation during evolution [54]. To address this issue, we employed a novel dynamic normalization method proposed in [66] to control the intensity of the nadir point in the normalization procedure. In this method, the calculation of normalized objectives presented in (2) is modified as follows:

$$\bar{f}_i = \frac{f_i - z^*}{L_i} \tag{8}$$

$$L_i = \frac{z^{nad} - z^*}{(1-\alpha)(z^{nad} - z^* - 1) + 1} \tag{9}$$

Following the original paper, we used the sigmoid function to control the parameter α as follows:

$$\alpha(t) = \frac{1}{1 + \exp\left[-8\left(\frac{t-1}{Gen_{\max}-1} - 0.5\right)\right]} \tag{10}$$

where t and $Gen_{\max}$ are the numbers of current generation and maximum generation, respectively. Initially, there is little reliable information about the true PF; thus, $\alpha(t) \approx 0$ and, further, $L_i \approx 1$, indicating that the nadir point has no impact on the normalization. As the algorithm executes, the value of $\alpha(t)$ increases, allowing a greater impact of the nadir point on the normalization. At the end of the algorithm, $\alpha(t) = 1$, making (8) and (2) equivalent.

3.5. Auction-Based Matching Mechanism

In the traditional replacement strategy of MOEA/Ds, as shown in Algorithm 3, nr is usually set to be a small value to maintain diversity, but this leads to the insufficient optimization of some subproblems, decreasing the convergence efficiency. To address this issue, we proposed a novel matching mechanism based on the auction algorithm to rematch the subproblems with the solutions for better association between them, promoting convergence efficiency.

Algorithm 5: AuctionBasedMatch($\boldsymbol{U}$, $\boldsymbol{\Lambda}$)

Input: Union set of solutions $\boldsymbol{U}$; weight vector set (subproblems) $\boldsymbol{\Lambda}$
Output: Re-matched population $\boldsymbol{P}$.

Calculate a_{ij} by (11) for $i = 1, ...N$, $j = 1, ..., |\boldsymbol{U}|$;
$p_j \leftarrow 0$; $b_{ij} \leftarrow 0$ ($i = 1, ...N$, $j = 1, ..., |\boldsymbol{U}|$);
$v_{ij} \leftarrow 0$ ($i = 1, ...N$, $j = 1, ..., |\boldsymbol{U}|$);
while *any weight vector is not assigned a solution, i.e.,* $\exists i \in \{1, ..., N\}$, $\sum_{j=1}^{|\boldsymbol{U}|} v_{ij} = 0$ **do**
  $\boldsymbol{I} \leftarrow$ The indexes of the weight vectors not assigned a solution;
  $\boldsymbol{J} \leftarrow \emptyset$;
  $\boldsymbol{A}(j) \leftarrow \emptyset$, for $j = 1, ..., |\boldsymbol{U}|$;
  for *each* $i \in \boldsymbol{I}$ **do**
    $\pi_i \leftarrow \max_j\{a_{ij} - p_j\}$; $j_i^* \leftarrow \arg\max_j\{a_{ij} - p_j\}$;
    $\omega_i \leftarrow \max_{j \neq j_i^*}\{a_{ij} - p_j\}$;
    $b_{ij_i^*} \leftarrow p_{j_i^*} + \pi_i - \omega_i + \varepsilon$;
    $\boldsymbol{J} \leftarrow \boldsymbol{J} \cup \{j_i^*\}$; *The indexes of bidded solutions*
    $\boldsymbol{A}(j_i^*) \leftarrow \boldsymbol{A}(j_i^*) \cup \{i\}$; *The indexes of the weight vectors that bid on solution* j_i^*
  end
  for *each* $j \in \boldsymbol{J}$ **do**
    $i^p \leftarrow \arg\max_{i \in \boldsymbol{A}(j)}\{b_{ij}\}$;
    $p_j \leftarrow b_{i^p j}$;
    $v_{:j} \leftarrow 0$; $v_{i^p j} \leftarrow 1$;
  end
end
for $i = 1 : N$ **do**
  $j \leftarrow \text{find}(v_{i:} \neq 0)$; $\boldsymbol{x}^j \leftarrow$the jth solitions in $\boldsymbol{U}$;
end

The auction algorithm originally proposed by Bertsekas [53] is a distributed algorithm that simulates the process of human auction activities. The main idea of this method is to introduce the concept of bidding: (1) Each bidder bids on their preferred item, which brings the maximum earning value to it. (2) The auctioneer assigns the item to the bidder that offers the highest bidding price, and then updates the item's price. In the proposed ABM, we regard the weight vectors (subproblems) and the solutions as bidders and auction items, respectively, and the relevant symbols are given as follows:

- $\boldsymbol{\Lambda}$: the collection of weight vectors (bidders), i.e., $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}^1, ..., \boldsymbol{\lambda}^N\}$.
- $\boldsymbol{U}$: the collection of solutions (auction items), i.e., $\boldsymbol{U} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^{|\boldsymbol{U}|}\}$.
- a_{ij}: the value of solution $\boldsymbol{x}^j$ to the weight vector $\boldsymbol{\lambda}^i$ ($i \in \{1, ..., N\}$, $j \in \{1, ..., |\boldsymbol{U}|\}$). Since the weight vector prefers the solution with small scalarizing function value, a_{ij} is calculated as the opposite of the scalarizing function value, i.e.,

$$a_{ij} = -g^{TCH}(\boldsymbol{x}^j \Big| \boldsymbol{\lambda}^i, \boldsymbol{z}^*, \boldsymbol{z}^{nad}). \tag{11}$$

- p_j: the price of solution $\boldsymbol{x}^j$ ($j \in \{1, ..., |\boldsymbol{U}|\}$). The price p_j is initialized to 0 and updated with the auction process.
- b_{ij}: the bidding price of the weight vector $\boldsymbol{\lambda}^i$ on solution $\boldsymbol{x}^j$ ($i \in \{1, ..., N\}, j \in \{1, ..., |\boldsymbol{U}|\}$).
- v_{ij}: the assignment decision variables ($i \in \{1, ..., N\}, j \in \{1, ..., |\boldsymbol{U}|\}$). $v_{ij} = 1$ if solution $\boldsymbol{x}^j$ is assigned to weight vector $\boldsymbol{\lambda}^i$; $v_{ij} = 0$ otherwise.

The relationship among the above symbols can be represented by the network shown in Figure 2.

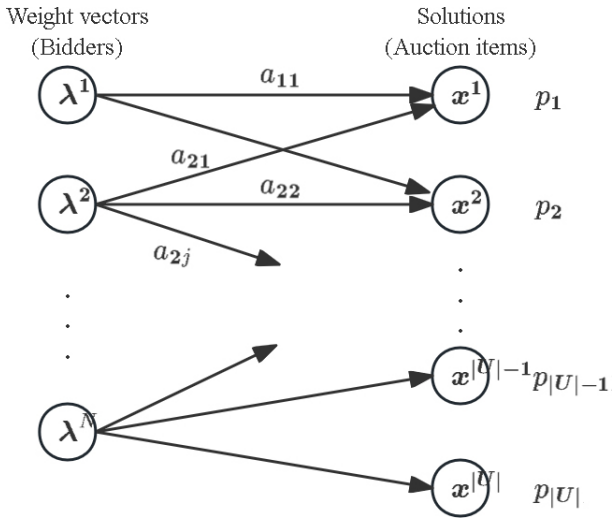


Figure 2. The relationship of the relevant defined symbols in ABM.

The pseudo-code and flowchart of ABM are presented in Algorithm 5 and Figure 3, respectively. The ABM mechanism is mainly composed of two crucial stages: the bidding stage (Algorithm 5, lines 8–14) and the assignment stage (Algorithm 5, lines 15–19).

(1) Bidding stage (Algorithm 5, lines 8–14)
In the bidding stage, each weight vector not assigning a solution selects its preferred solution from $\boldsymbol{U}$ and bids on it. Firstly, for the unassigned weight vector $\boldsymbol{\lambda}^i$, we calculate the earning value of each solution to it under current item prices. The earning value is $a_{ij} - p_j$. Motivated by self-interest, each unassigned weight vector prefers the solution with better scalarizing function value, i.e., $\boldsymbol{\lambda}^i$ prefers $\boldsymbol{x}^j$ with the great value of a_{ij}. Therefore, $\boldsymbol{\lambda}^i$ selects the solution from $\boldsymbol{U}$ that offers it the maximum earning value as its preferred one, which can be expressed as

$$j_i^* = \arg\max_j \{a_{ij} - p_j\}, \tag{12}$$

where j_i^* denotes the index of the solution $\boldsymbol{\lambda}^i$ prefers. The corresponding earning value is denoted by π_i, i.e.,

$$\pi_i = \max_j \{a_{ij} - p_j\} = a_{ij_i^*} - p_{j_i^*}. \tag{13}$$

Secondly, the unassigned weight vector $\boldsymbol{\lambda}^i$ bids on its preferred solution $\boldsymbol{x}^{j_i^*}$, and the bidding price is calculated as follows:

$$b_{ij_i^*} = p_{j_i^*} + \pi_i - \omega_i + \varepsilon, \tag{14}$$

where ε is a relaxation parameter and set to 10^{-2} herein. ω_i represents the second-best earning value for the weight vector $\boldsymbol{\lambda}^i$, which is calculated as follows:

$$\omega_i = \max_{j \neq j_i^*} \{a_{ij} - p_j\}. \tag{15}$$

Through Equations (13)–(15), we can deduce that

$$b_{ij_i^*} \geqslant p_{j_i^*} + \varepsilon, \tag{16}$$

implying that the bidding price on solution $\boldsymbol{x}^{j_i^*}$ is at least ε higher than its current price $p_{j_i^*}$. After every unassigned weight vector expresses its preferred solution and bids on it through the above process, we proceed to the following assignment stage.

(2) Assignment stage (Algorithm 5, lines 15–20)

In the assignment stage, the auctioneer assigns the bidded solutions to the weight vectors with the highest bidding price on it and updates the item price accordingly. Let j be the index of a bidded solution, i.e., $j \in \boldsymbol{J}$ ($\boldsymbol{J}$ denotes the set of indexes of the bidded solutions), it is assigned to the weight vector $\boldsymbol{\lambda}^{i^p}$, which is expressed as follows:

$$i^p = \arg \max_{i \in \boldsymbol{A}(j)} b_{ij}, \tag{17}$$

where $\boldsymbol{A}(j)$ denotes the collection of indexes of the bidders that bid on $\boldsymbol{x}^j$ in the bidding stage. As a result, the assignment decision variables are updated as follows:

$$\begin{cases} v_{ij} = 1 & \text{if} \quad i = i^p, \\ v_{ij} = 0 & \text{if} \quad i \neq i^p. \end{cases} \tag{18}$$

Equation (18) indicates that once solution $\boldsymbol{x}^j$ is assigned to weight vector $\boldsymbol{\lambda}^{i_p}$, it withdraws its previous assignment to other weight vectors. Finally, the highest bidding price for solution $\boldsymbol{x}^j$ is taken as its updated price, which can be written as

$$p'_j = \max_{i \in \boldsymbol{A}(i)} \{b_{ij}\} = b_{i^p j}. \tag{19}$$

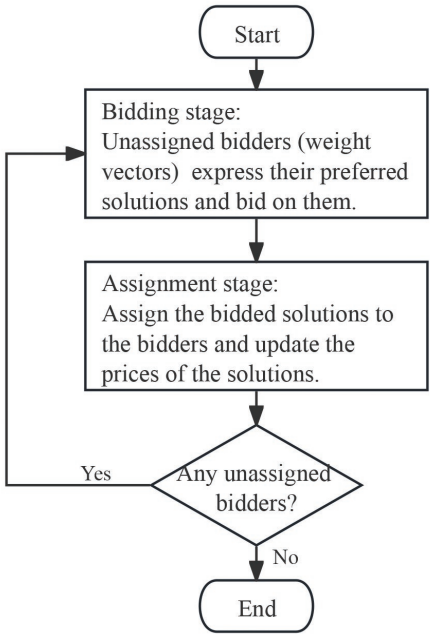


Figure 3. The flowchart of the ABM mechanism.

We iterate the auction process including the above two stages until each weight vector is assigned a solution (Algorithm 5, lines 4–20). It can be predicted from (16) that, as the auction process iterates, the price of the solution will gradually increase, resulting in decreased earning values for the weight vectors. Referring to [53], it can be proved that ABM must terminate after a finite number of iterations, i.e., every weight vector will be assigned a solution after a finite number of auction rounds.

The rationality of ABM is supported by the following derivation: after each round of auction, if the weight vector $\boldsymbol{\lambda}^i$ is assigned its preferred solution $\boldsymbol{x}^{j_i^*}$, the following inequality can be derived from (12)–(16) and (19):

$$\begin{aligned} a_{ij_i^*} - p'_{j_i^*} &= a_{ij_i^*} - (p_{j_i^*} + \pi_i - \omega_i + \varepsilon) \\ &= \omega_i - \varepsilon = \max_{j \neq j_i^*} \{a_{ij} - p_j\} - \varepsilon \\ &\geqslant \max_{j \neq j_i^*} \{a_{ij} - p'_j\} - \varepsilon \quad (\text{Since } p'_{j_i^*} = b_{ij_i^*} > p_{j_i^*}) \\ &\geqslant \max_{j} \{a_{ij} - p'_j\} - \varepsilon. \end{aligned} \tag{20}$$

When the ABM algorithm terminates, each weight vector is assigned its preferred solution; thus, it can be further derived from (20) that

$$\sum_{i=1}^{N} a_{ij_i^*} - p'_{j_i^*} \geqslant \sum_{i=1}^{N} \max_{j} \{a_{ij} - p'_j\} - N\varepsilon. \tag{21}$$

Inequality (21) indicates that, when the ABM algorithm terminates, the obtained assignment scheme is optimal for the following one-to-one assignment problem within a maximum error range of $N\varepsilon$:

$$\begin{aligned} &\max \sum_{i=1}^{N} \sum_{j=1}^{M} (a_{ij} - p'_j) v_{ij} \\ &s.t. \quad \sum_{j=1}^{M} v_{ij} = 1, \forall i \in I. \end{aligned} \tag{22}$$

In summary, the assignment scheme obtained by ABM ensures the maximization of the earning value of all weight vectors as much as possible. In other words, through the employment of ABM, each weight vector is associated with a solution that offers a better scalarizing function value in a competitive manner. Therefore, we reasonably believe that ABM can accelerate the convergence of the population. Meanwhile, the one-to-one assignment strategy avoids the presence of repeated solutions in the population, maintaining the diversity. The superiority of ABM has been verified in the experimental study.

3.6. Weight Vector Adjustment

Adjusting weight vectors based on ***EP*** during the evolution process is a good method for a more appropriate distribution, and, recently, some MOEA/Ds have adopted this strategy [37,45,49]. However, studies have indicated that premature or too late of an adjustment of the weight vectors may severely affect the optimization process by deteriorating the search [19]. For this reason, in MOEA/D-ABM, the weight vector adjustment is only permitted every 5% between 10% and 90% of the whole evolution process. Meanwhile, a measure called convergence activity, denoted as Ca, is defined to determine whether weight vectors need to be adjusted or not. It can be formulated as follows:

$$Ca = \frac{1}{n_{ca}} \sum_{i=t-n_{ca}+1}^{t} N_i^u, \tag{23}$$

where N_i^u denotes the proportion of the subproblems that are improved in the i-th generation; t is the number of current generation; Ca represents the average of N_i^u in the latest n_{ca} generations. When Ca is smaller than a given coefficient ρ, we believe that the population cannot achieve better solutions for the subproblems with current weight vectors. In this condition, the weight vectors need to be adjusted.

The detailed procedure of adjusting weight vectors is described in Algorithm 6. Firstly, we calculate the sparsity level of each individual in population $\boldsymbol{P}$ among $\boldsymbol{P}$ using (7), and then remove the individual with the lowest sparsity level, $\boldsymbol{x}^c$, and its associated weight vector, $\boldsymbol{\lambda}^c$. The deletion process will be repeated until nus individuals in $\boldsymbol{P}$ are removed (Algorithm 6, lines 2–6). The reason for this operator is that the subproblem associated with the crowded individual is relatively redundant. After that, we calculate the sparsity level among of each non-dominated individual in $\boldsymbol{EP}$ among the population $\boldsymbol{P}$, and select the sparsest one with the largest sparsity level, $\boldsymbol{x}^s$, as the new individual to be added into $\boldsymbol{P}$ (Algorithm 6, lines 7–12). Meanwhile, since the optimal solution of Tchebycheff scalarizing function is the intersection point of vector $(1/\lambda_1, ..., 1/\lambda_m)^\top$ and normalized PF, the new weight vector associated with $\boldsymbol{x}^s$ is generated by using the Tchebycheff method:

$$\boldsymbol{\lambda}^s = \left(\frac{\frac{1}{\bar{f}_1(\boldsymbol{x}^s)}}{\sum_{j=1}^m \frac{1}{\bar{f}_j(\boldsymbol{x}^s)}}, ..., \frac{\frac{1}{\bar{f}_m(\boldsymbol{x}^s)}}{\sum_{j=1}^m \frac{1}{\bar{f}_j(\boldsymbol{x}^s)}}\right)^\top. \tag{24}$$

The addition process will also be repeated until nus individuals are added (lines 7–12). The reason for this operator is that the sparsest non-dominated solution has relative potential in exploring PF comprehensively. Finally, the neighborhood of each weight vector is updated (line 13).

Algorithm 6: AdjustWeights($\boldsymbol{P}$, $\boldsymbol{EP}$, $\boldsymbol{\Lambda}$, nus)

Input: Population $\boldsymbol{P}$; external population $\boldsymbol{EP}$; weight vector set $\boldsymbol{\Lambda}$; the number of weight vectors to be updated nus
Output: Updated population $\boldsymbol{P}$; Updated weight vector set $\boldsymbol{\Lambda}$; Updated neighbors $\boldsymbol{B}$.

1 $adjust \leftarrow 0$;
2 **while** $adjust < nus$ **do**
3 Calculate the sparsity level of each individual in $\boldsymbol{P}$ among $\boldsymbol{P}$ by (7);
4 Remove the individual with the lowest sparsity level, $\boldsymbol{x}^c$, from $\boldsymbol{P}$ and it associated weight vector, $\boldsymbol{\lambda}^c$ from $\boldsymbol{\Lambda}$;
5 $adjust \leftarrow adjust + 1$;
6 **end**
7 **while** $adjust > 0$ **do**
8 Calculate the sparsity level of each individual in $\boldsymbol{EP}$ among $\boldsymbol{P}$ by (7);
9 Add the individual with the largest sparsity level in $\boldsymbol{EP}$, $\boldsymbol{x}^s$, to $\boldsymbol{P}$;
10 Generate a new weight vector, $\boldsymbol{\lambda}^s$, by (24) and add it to $\boldsymbol{\Lambda}$;
11 $adjust \leftarrow adjust - 1$;
12 **end**
13 Update the neighborhood $\boldsymbol{B}(i)$ for each weight vector, $i = 1, ..., N$.

3.7. Computational Complexity

The computational complexity is considered for one generation of the proposed MOEA/D-ABM, whose computational cost mainly comes from the MOEA/D (Algorithm 2, lines 5–15), the $\boldsymbol{EP}$ update (Algorithm 2, line 16), the ABM, and the weight vector adjustment (Algorithm 2, line 25). The complexity of MOEA/D is $O(mTN)$; the complexity of the $\boldsymbol{EP}$ update is $O(2mN^2)$; the complexity of the weight vector adjustment is $O(nus \cdot mN^2)$; for the ABM, the complexity of initializing a_{ij} (Algorithm 5, line 1) is $O(mN|\boldsymbol{U}|)$, and the maximum complexity of each auction round (Algorithm 5, lines 4-20) is $O(2N^2 + N)$ (N bidders are unassigned and N auction items are bidded). The number of auction rounds, denoted as R, is finite but not fixed, primarily influenced by the relaxation parameter ε, the number of bidders N and the number of auction items $|\boldsymbol{U}|$; thus, the complexity of ABM is $O(2RN^2)$. To sum up the findings, the maximum complexity of MOEA/D-ABM is $O((m \cdot nus + 2R)N^2)$.

4. Experimental Study

4.1. Test Problem

To evaluate the performance of MOEA/D-ABM in solving MOPs, three benchmark test suites were employed, including the DTLZ test suite [55] (DTLZ1-DTLZ7), WFG test suite [56] (WFG1-WFG9), and BT test suite [57] (BT3-BT5, BT8-BT9). The number of objectives m varied from two to ten. The dimension of decision variables n is related to m. These test suites possess various features such as linear, concave, convex, multi-modal, degenerate, discontinuous, and biased features. BT1-BT2 and BT6-BT7 problems were not employed in our experiment, due to their lack of distinctive features. See Table 1 for specific parameters settings and features of these MOP test suites.

Table 1. The parameter settings and features of MOP test suits.

Problem	Number of Objectives (m)	Dimension of Decision Variables (n)	Features
DTLZ1	3, 5, 8, 10	$m-1+5$	Linear, multi-modal
DTLZ2	3, 5, 8, 10	$m-1+10$	Concave
DTLZ3	3, 5, 8, 10	$m-1+10$	Concave, multi-modal
DTLZ4	3, 5, 8, 10	$m-1+10$	Concave, bias
DTLZ5	3, 5, 8, 10	$m-1+10$	Degenerate
DTLZ6	3, 5, 8, 10	$m-1+10$	Degenerate, bias
DTLZ7	3, 5, 8, 10	$m-1+10$	Discontinuous
WFG1	3, 5, 8, 10	$m-1+10$	Bias, mixed
WFG2	3, 5, 8, 10	$m-1+10$	Convex, discontinuous, multi-modal
WFG3	3, 5, 8, 10	$m-1+10$	Linear, degenerate
WFG4-9	3, 5, 8, 10	$m-1+10$	Concave
BT3-4	2	30	Bias
BT5	2	30	Discontinuous, bias
BT8	2	30	Multi-modal, bias
BT9	3	30	Bias

4.2. Performance Metrics

The inverted generational distance (IGD) [17] and hypervolume (HV) [16] are two widely utilized evaluation metrics for assessing the performance of MOEAs. They are also used in our experiments.

IGD measures the mean distance from the points in the true PF to the nearest individual in the output population, which is calculated using the following formula:

$$IGD(\boldsymbol{P}, \boldsymbol{P}^*) = \frac{\sum_{\boldsymbol{x}^* \in \boldsymbol{P}^*} d(\boldsymbol{x}^*, \boldsymbol{P})}{|\boldsymbol{P}^*|} \tag{25}$$

$$d(\boldsymbol{x}^*, \boldsymbol{P}) = \min_{\boldsymbol{x} \in \boldsymbol{P}} \|\boldsymbol{x}^* - \boldsymbol{x}\|, \tag{26}$$

where $\boldsymbol{P}$ and $\boldsymbol{P}^*$ are the output population and the set of points in the true PF, respectively; $d(\boldsymbol{x}^*, \boldsymbol{P})$ denotes the Euclidean distance between $\boldsymbol{x}^*$ and its nearest individual in $\boldsymbol{P}$. The experiments use 10,000 points uniformly sampled on the true PF. A smaller IGD value implies that the population $\boldsymbol{P}$ approximates the true PF more effectively and, generally, has a superior convergence and diversity.

HV measures the volume of the objective space surrounded by the output population $\boldsymbol{P}$ and the nadir point $\boldsymbol{z}^{nad}$, which can be calculated by the following formula:

$$HV(\boldsymbol{P}) = VOL\left(\bigcup_{\boldsymbol{x}\in\boldsymbol{P}}[f_1(\boldsymbol{x}), z_1^{nad}] \times \cdots \times [f_m(\boldsymbol{x}), z_m^{nad}]\right) \tag{27}$$

where VOL denotes the Lebesgue measure. A higher HV value indicates a better quality of the population in approximating the whole PF. We used a modified nadir point 10% higher than the upper bound of the true PF in our experiments.

4.3. Comparison Algorithms

In our experiments, six state-of-the-art MOEAs were considered as the comparison algorithms for MOEA/D-ABM, which included MOEA/D-URWA [49], MOEA/D-DU [28], MOEA/D-UR [45], NMPSO [67], AR-MOEA [47], and NSGA-III [68]. The main unique aspects of each algorithm are introduced as follows:

- **MOEA/D-URAW** [49] is a decomposition-based MOEA in which the weight vectors are adjusted once every certain number of generations. The sparsity level is also used to determine which solutions and weight vectors are deleted and added, but the trigger timing of adjustment is predefined.
- **MOEA/D-DU** [28] uses a modified replacement strategy in the evolution process explicitly. It exploits the perpendicular distance from the solution to the weight vector in the objective space to determine the solutions for the update, aiming to better balance between convergence and diversity.
- **MOEA/D-UR** [45] is a decomposition-based MOEA that updates the weight vectors when required. It uses an improvement metric to determine when to adjust the weight vectors and an objective space dividing procedure to increase the diversity of the population.
- **NMPSO** [67] integrates the particle swarm optimizer into the framework of MOEA. It employs the balanceable fitness estimation (BFE) and two novel operators. Specifically, the BFE combines the convergence and diversity distances to guide the particles to approximate to the PF. The two operators are respectively responsible for searching on the external archive and updating the velocity equation.
- **AR-MOEA** [47] is a reference point adaptation MOEA based on IGD-NS indicator. The reference adaptation mechanism is used to adjust the reference points at each generation based on their contributions to the IGD-NS value, and the population is updated according to the adjusted reference points.
- **NSGA-III** [68] introduces a reference point-based approach in the NSGA-II framework. It implements the non-dominated sorting approach on the population and, then, the individuals on the selection frontier are selected and retained based on their vector angle with the reference point.

4.4. Experimental Settings

All code of MOEA/D-ABM and comparison algorithms were conducted in MATLAB_R2018b on the PlatEMO platform [69] with Intel Core i5 CPU 2.30GHz and 12GB internal memory. The basic experimental settings are shown in Table 2, where the Wilcoxon rank test [70] with 0.05 significance level is adopted to statistically compare the performance. The SBX crossover and polynomial mutation were used in the seven MOEAs to produce new solutions, with the parameter settings detailed in Table 3. For a fair comparison, the selection probability δ and replacement size nr, which are adopted in MOEA/D-URAW, MOEA/D-DU, MOEA/D-UR, and MOEA/D-ABM, were set to 0.9 and 2, respectively. The adjustment size nus, which is used in MOEA/D-URAW and MOEA/D-ABM, was set to $0.05N$. The other parameters of the algorithms to be compared were set according to the

recommended values in the original papers. Each algorithm ran 30 times independently for each test problem.

Table 2. Basic experimental settings.

Parameter	Setup
Maximum evaluations $maxFE$	50,000
Population size N	100 ($m = 2, 3, 5$), 200 ($m = 8, 10$)
Neighborhood size T	$0.1N$
Significance test	Wilcoxon rank test [70], significance level is 0.05

Table 3. Parameter settings for SBX crossover and polynomial mutation.

Parameter	Setup
Crossover probability p_c	1
Mutation probability p_m	$1/n$
Distribution index of crossover η_c	20
Distribution index of mutation η_m	20

4.5. Sensitivity Analysis

The convergence activity coefficients ρ and n_{ca} are important to determining the trigger timing of weight vector adjustment. Intuitively, a smaller value of n_{ca} or a larger value of ρ (or both) leads to more frequent adjustments of the weight vector. To achieve suitable settings, we analyzed the effect of these two coefficients on the performance of MOEA/D-ABM. Based on different values of ρ and n_{ca}, we used the Friedman test to obtain the average rank of MOEA/D-ABM in solving the above test problems. Figure 4 presents the average rank of HV and IGD metrics for various coefficient vectors (ρ, n_{ca}), where $\rho \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and $n_{ca} \in \{1, 2, 3, 4, 5\}$. It can be observed that, when $\rho = 0.7$ and $n_{ca} = 4$ (the 14th coefficient vector), MOEA/D-ABM achieves favorable average rank in both HV and IGD metrics. Therefore, we adopted these coefficient value settings in our experiments.

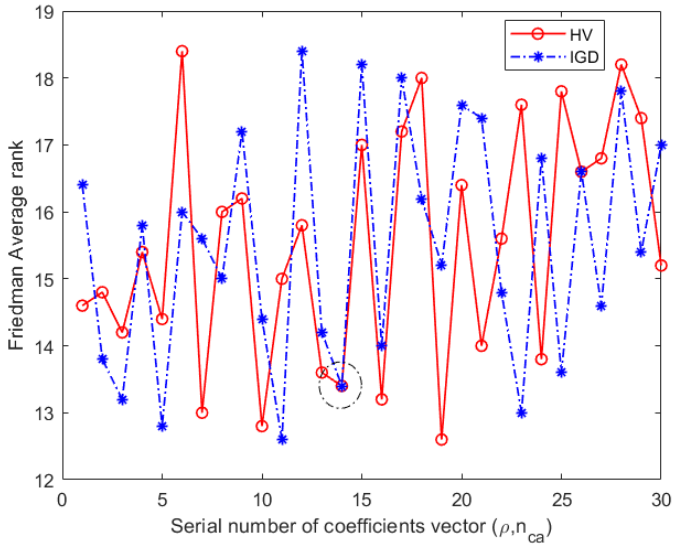


Figure 4. The average ranks of MOEA/D-ABM with different values of (ρ, n_{ca}).

4.6. Experimental Results and Analysis

The statistical results of the IGD and HV metrics obtained by the seven algorithms are summarized in Tables A1–A4, where the best average value for each test problem is in bold. The "+", "−", and "=" indicate that the result obtained by this comparison algorithm is significantly better, significantly worse, and statistically similar to that obtained by MOEA/D-ABM. For the IGD results shown in Tables A1 and A2, it can be seen that MOEA/D-ABM shows superior performance in fourteen regular test problems and five irregular test problems, accounting for 27.5% of the total number of instances. MOEA/D-URAW, MOEA/D-DU, MOEA/D-UR, NMPSO, ARMOEA, and NSGA-III have the best performance on 16, 3, 2, 3, 5, and 1 regular test problems and 4, 0, 11, 2, 3, and 0 irregular test problems, respectively. For the HV results shown in Tables A3 and A4, MOEA/D-ABM shows superior performance in 13 regular test problems and 11 irregular test problems, accounting for 34.8% of the total number of instances. MOEA/D-URAW, MOEA/D-DU, MOEA/D-UR, NMPSO, ARMOEA, and NSGA-III have the best performance on 4, 3, 0, 18, 4, and 2 regular instances and 2, 0, 8, 4, 0, and 0 irregular instances, respectively. Herein, the distribution figures of non-dominated solutions shown in Figures 5–12 adopt the result with the median HV value in 30 runs for each algorithm. A more intuitive comparison of these algorithms is presented in Figure 13, which shows the average ranks of the seven algorithms for the test problems based on all IGD and HV results. Furthermore, the convergence rate of the algorithms can be analyzed and compared by the process IGD mean value presented in Figures A1–A3.

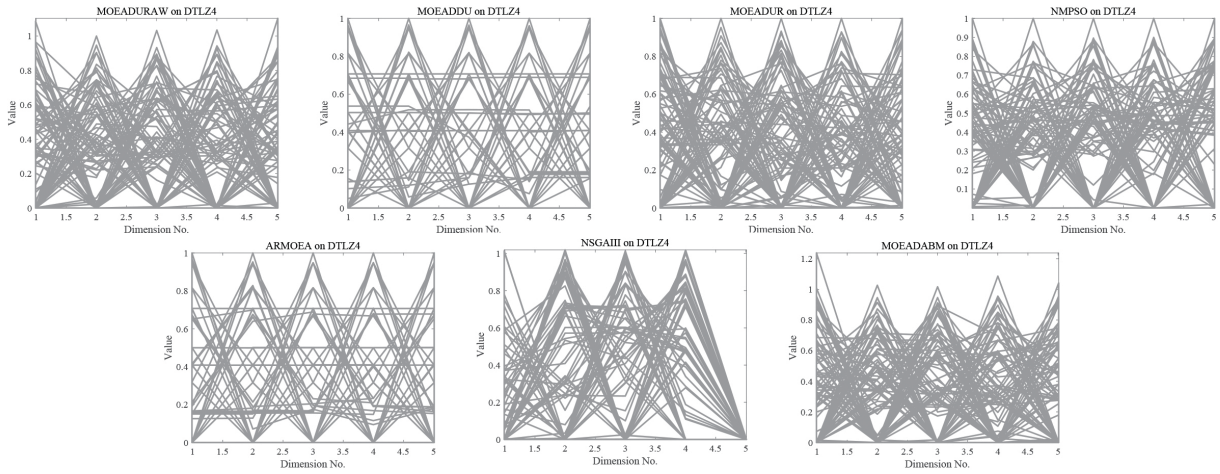


Figure 5. Non-dominated solutions obtained by seven MOEAs on DTLZ4 with five objectives.

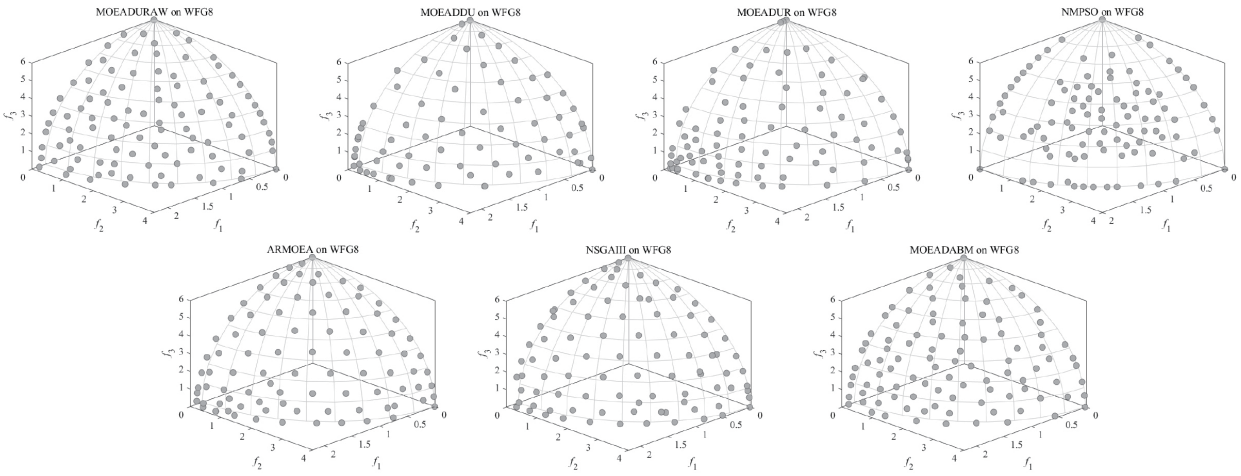


Figure 6. Non-dominated solutions obtained by seven MOEAs on WFG8 with three objectives.

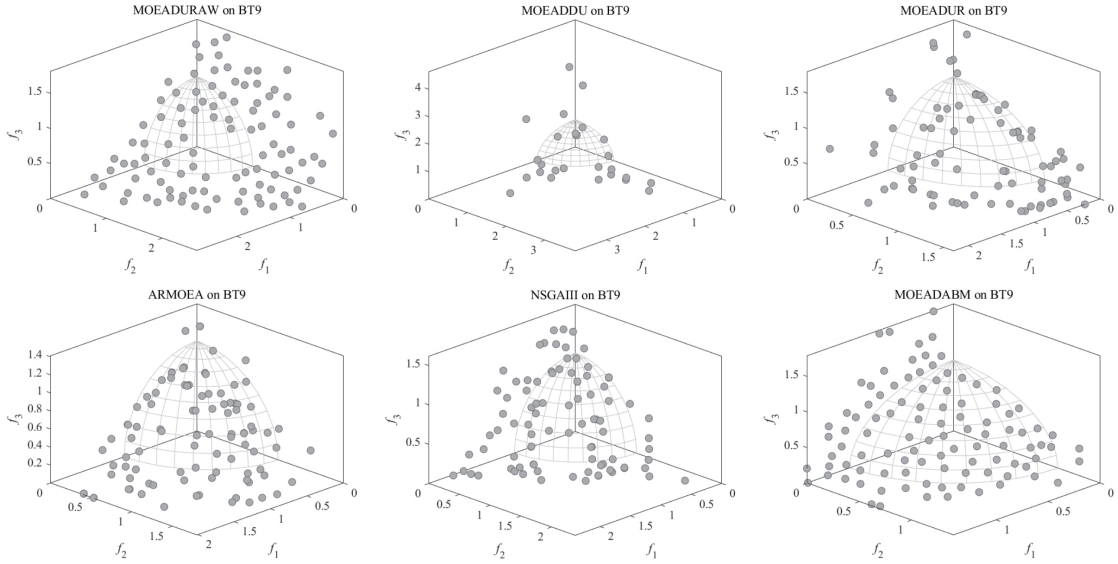


Figure 7. Non-dominated solutions obtained by six MOEAs on BT9 (NMPSO fails).

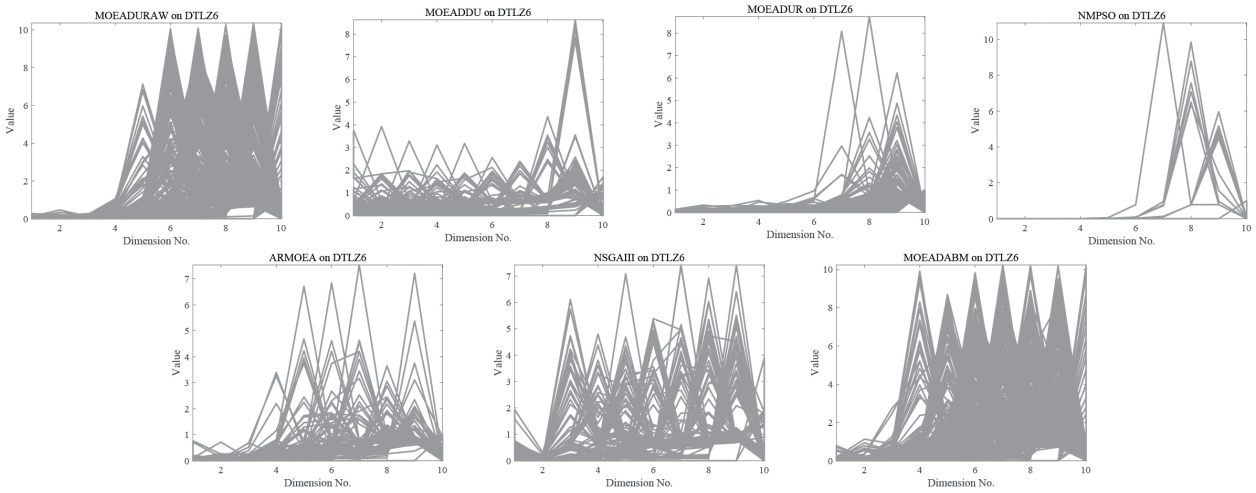


Figure 8. Non-dominated solutions obtained by seven MOEAs on DTLZ6 with ten objectives.

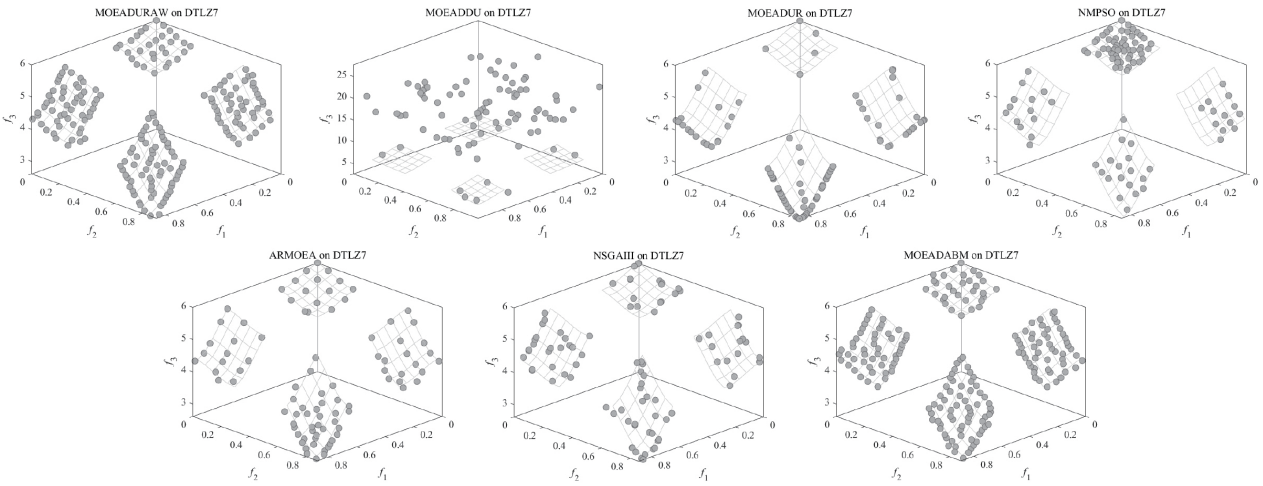


Figure 9. Non-dominated solutions obtained by seven MOEAs on DTLZ7 with three objectives.

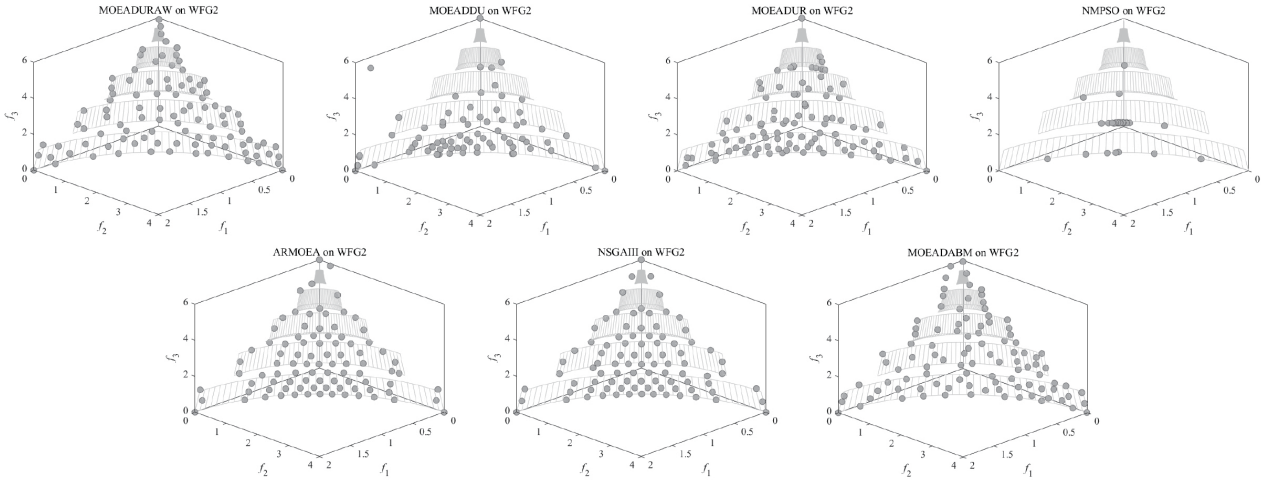


Figure 10. Non-dominated solutions obtained by seven MOEAs on WFG2 with three objectives.

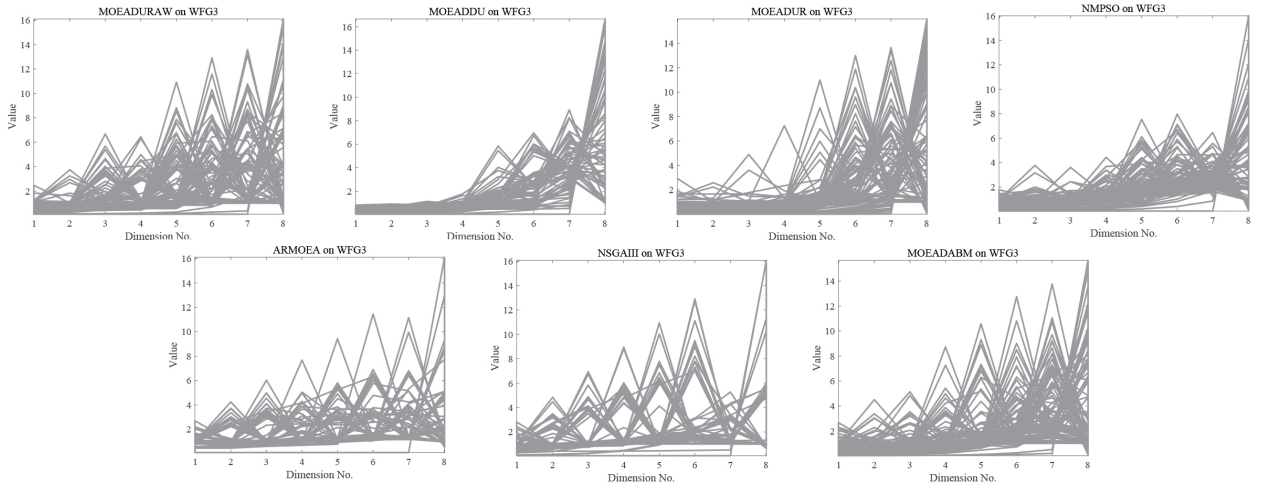


Figure 11. Non-dominated solutions obtained by seven MOEAs on WFG3 with eight objectives.

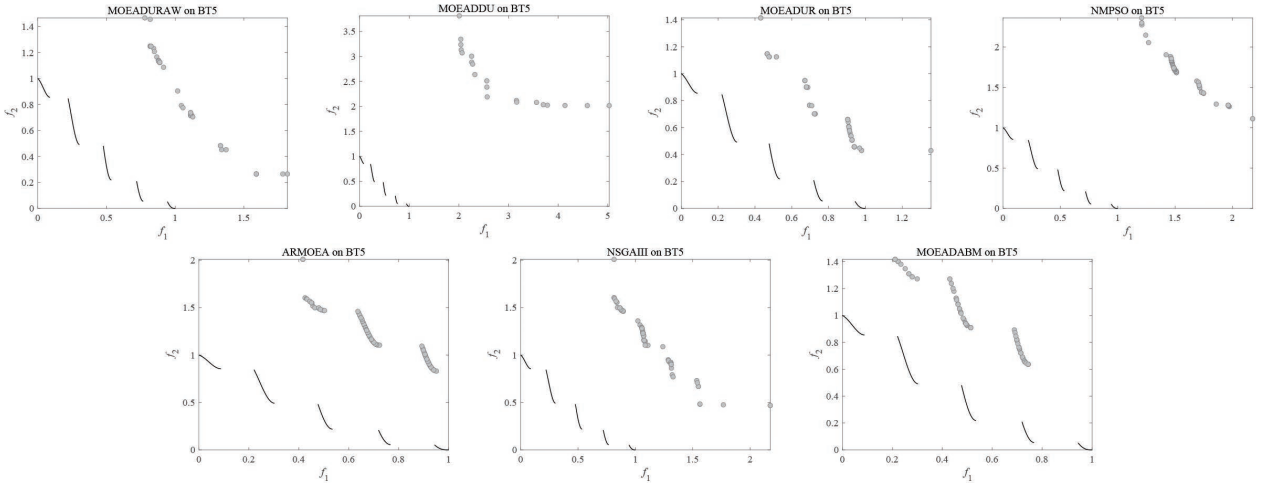
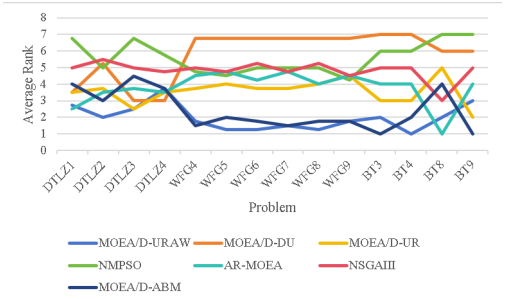
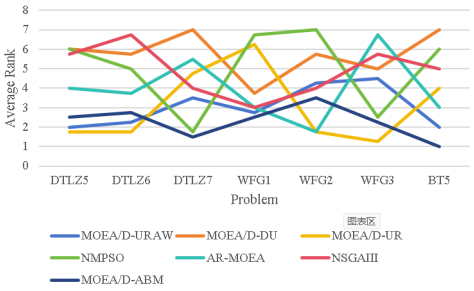


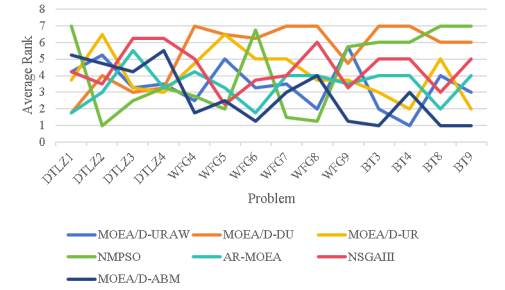
Figure 12. Non-dominated solutions obtained by seven MOEAs on BT5.



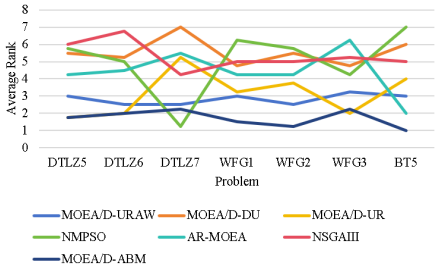
(**a**) Average ranks of IGD results on regular problems



(**b**) Average ranks of IGD results on irregular problems



(**c**) Average ranks of HV results on regular problems



(**d**) Average ranks of HV results on irregular problems

Figure 13. Average ranks of IGD and HV results for seven MOEAs in solving DTLZ, WFG, and BT test suites.

Next, we will analysis the correlation between the characteristics of the test problems and the performance of the algorithms through the experimental results.

4.6.1. Results and Analysis on Regular Test Problems

DTLZ1-DTLZ4, WFG4-WFG9, and BT3-BT4, BT8-BT9 are the 14 test problems with regular PFs, and the comparison results of their HV and IGD metrics obtained by the seven algorithms are shown in Tables A1 and A3.

DTLZ1 has a linear hyper-plane PF and DTLZ3 has a concave PF, but their multi-modal feather causes the existence of multiple local optimal solutions, increasing the difficulty in convergence. It should be noted that, presented in Figure A1, MOEA/D-ABM obtains the best convergence rate in terms of the process IGD mean value. NMPSO performs worse than the other algorithms in both IGD and HV metrics.

DTLZ2 is a simple MOP with concave PF, and all the algorithms have good performance on the IGD metric. Table A3 shows that NMPSO stably obtains the best performance on the HV metric, while MOEA/D-ABM is competitive. DTLZ4 also has a concave PF, but the bias feature makes it hard to maintain a proper distribution of the candidate solutions. From Table A1, Table A3, and Figure A1, we can observe that both IGD and HV results obtained by MOEA/D-ABM in solving DTLZ4 are not backward. Figure 5 plots the non-dominated solutions obtained by these algorithms on DTLZ4 with five objectives. Apparently, MOEA/D-URAW, MOEA/D-UR, and MOEA/D-ABM show better performance in diversity. The results verify the superiority of decomposition-based MOEAs in maintaining diversity.

WFG4-WFG9 have regular PFs with a scaled concave shape and decision variables with high coupling. As can be seen from Figure 13a,c, MOEA/D-ABM and MOEAD-URAW deal well with these problems and have the highest average ranks in both IGD and HV metrics, especially in the IGD metric. Figures A2 and A3 show that the process IGD mean value of MOEA/D-ABM and MOEA/D-URAW fluctuates periodically. The underlying reason is that the population distribution is changed by the weight vector adjustment mechanism at every specified calculation. Overall, the performance of MOEA/D-DU in terms of both IGD and HV metrics is not satisfactory. NMPSO performs poorly in the IGD metric but well in the HV metric. The reason for this phenomenon is that the non-

dominated solutions obtained by NMPSO have a wide distribution but bad approximation to PF. Figure 6 plots the final non-dominated solutions obtained by the algorithms on WFG8 with three objectives. Apparently, the distribution of results obtained by MOEA/D-URAW, ARMOEA, NSGA-III, and MOEA/D-ABM is relatively more uniform.

BT3-BT5 and BT9 are the test problems with twp or three objectives and have union PFs. However, their strong bias feature and highly deceptive PF pose a great challenge to the convergence of MOEAs. From Tables A1 and A3, MOEA/D-ABM is apparently superior to the other MOEAs in both IGD and HV metrics. Meanwhile, as shown in Figure A3, in solving the BT3, BT4, and BT9 problems, the convergence rate of the process IGD mean value obtained by MOEA/D-ABM is obviously superior to the other algorithms throughout the whole evolution. It is mainly due to the fact that the ABM mechanism can better reassociate subproblems with the current solutions in each generation, further optimizing each subproblem while maintaining population diversity. In this way, the exploitation capability of the population is further enhanced. Additionally, MOEA/D-DU obtains the worst performance overall, and NMPSO fails in dealing with BT8 and BT9. To visually understand the experimental results, the non-dominated solutions of BT9 obtained by the MOEAs, except NMPSO, are plotted in Figure 7. Clearly, the result obtained by MOEA/D-ABM performs best in approximating PF, while being evenly distributed.

4.6.2. Results and Analysis on Irregular Test Problems

DTLZ5-DTLZ7, WFG1-WFG3, and BT5 are the seven test problems with irregular PFs. It is clear from Figure 13b,d that MOEA/D-ABM exhibits outstanding performance in the average rank in both IGD and HV metrics.

For the DTLZ5 and DTLZ6 with degenerate PFs, MOEA/D-UR presents the best average rank and MOEA/D-ABM obtains the third average rank in the IGD metric. Figure A1 shows that, when dealing with the two problems, the mean values of the IGD obtained by MOEA/D-ABM and MOEA/D-URAW experience a rapid decline in the early evolution but rebound later. This negativity is mainly caused by the excessive number of weight vectors adjusted in the later evolution process. In addition, MOEA/D-DU has the worst overall performance. Figure 8 plots the final obtained non-dominated solutions for these algorithms concerning DTLZ6 with 10 objectives. Evidently, the distribution of results obtained by MOEA/D-ABM and MOEA/D-URAW is more diverse, but performs relatively poorer in terms of convergence.

DTLZ7 has a discontinuous PF. As shown in Figure 13, MOEA/D-ABM performs best in the average rank of the IGD metric and second in the average rank of HV metric. Furthermore, as depicted in Figure A1, MOEA/D-ABM always demonstrates the best convergence rate in terms of the process IGD mean value. Conversely, MOEA/D-DU shows the worst overall performance. The distribution of the non-dominated solutions obtained by these algorithms concerning three-objective DTLZ7 is presented in Figure 9. It is clear that MOEA/D-ABM and MOEA/D-URAW achieve more uniformly distributed non-dominated solutions, which verifies that the adaptive weight vector adjustment is conducive to increasing diversity when dealing with irregular problems.

WFG1 has a PF with bias and mixed features. As shown in Figures 13 and A2, MOEA/D-ABM obtains the best average ranks on both IGD and HV metrics, and also possesses the highest convergence rate of process IGD mean value in the early evolution. MOEA/D-URAW achieves competitive results on this problem. NMPSO fails in approximating PF.

WFG2 has a discontinuous PF. MOEA/D-ABM performs best in the HV metric for three, five, and ten objective instances, while MOEA/D-DU and NMPSO exhibit poorer performance compared to other algorithms. Figure 10 visually shows the distribution of the non-dominated solutions obtained by these algorithms on three-objective WFG2. Notably, the results obtained by MOEA/D-ABM and MOEA/D-URAW have the wider distribution, which validates the superiority of adaptive weight vector adjustment in terms of diversity when dealing with irregular MOPs. The results obtained by ARMOEA and

NSGA-III perform well in uniformity, but are sparsely distributed in the region near the axis. This is because the fixed weight vectors used in these two algorithms weaken the exploitation ability of the population in the boundary regions.

WFG3 is a complex MOP with a linear degenerate PF. It is difficult for the most existing MOEAs to obtain satisfactory results. As for the IGD metric, MOEA/D-UR obtains the overall best performance. Specifically, it has the best performance on five, eight, and ten objective instances. As shown in Figure A2, the process IGD mean value obtained by MOEA/D-ABM converges most rapidly in the early evolution but rebounds later. This phenomenon also occurs when dealing with DTLZ4 and DTLZ5 problems. It suggests that, in the decomposition-based MOEAs, the smaller-scale weight vector adjustment in the later evolution is more suitable for dealing with the MOPs with a degenerate PF. ARMOEA and NSGA-III achieve poor performance when the dimension of objectives exceeds five. As for the HV metric, Table A4 shows that all the algorithms are inferior in solving WFG3. Especially when solving the instance with 10 objectives, only MOEA/D-ABM obtains a non-zero value. The distribution of the non-dominated solutions obtained by the algorithms is shown in Figure 11. It can be observed that MOEA/D-DU and NMPSO face a challenge in maintaining diversity.

BT5 has a discontinuous PF with a strong bias feature. In terms of the IGD results presented in Table A2 and Figure A3, MOEA/D-ABM obtains the best performance and has the highest convergence rate of the process IGD mean value, while MOEA/D-DU performs the worst. As for the HV metric, MOEA/D-ABM also obtains the best result, while MOEA/D-DU and NMPSO obtain zero HV values. The distribution of the obtained non-dominated solutions shown in Figure 12 indicate that MOEA/D-ABM and ARMOEA perform relatively well in terms of diversity, while MOEA/D-DU and NMPSO perform poorly in both diversity and convergence.

4.7. Overall Discussion of the Results

As analyzed in the previous subsections, MOEA/D-ABM has achieved good performance in solving most test problems with various PFs.

As for the test problems with a regular PF, Table A1 presents that MOEA/D-ABM and MOEA/D-URAW work well and stably on different test problems. They have the best overall performance on both IGD and HV metrics when dealing with WFG4, WFG6, WFG9, BT3, BT4, and BT9 problems. This result demonstrates that MOEA/Ds with adaptive weight vectors achieve a good balance between convergence and diversity. Although NMPSO has good overall performance on the HV metric when dealing with DTLZ2, DTLZ4, WFG5, WFG7, and WFG8 problems, its performance on the IGD metric is poor. This is mainly due to the limitation of its selection strategy-based on non-dominated sorting in convergence efficiency.

As for the test problems with irregular PFs, the distribution of results obtained by MOEAs with adaptive weight vectors, including MOEA/D-URAW, MOEA/D-ABM, and MOEA/D-UR, are always good in the performance of diversity. Especially when dealing with DTLZ7 and WFG2 problems with discontinuous PFs, MOEA/D-ABM and MOEA/D-URAW are evidently able to obtain non-dominated solutions with broader distribution (Figures 9 and 10). This is due to the fact that, in these two algorithms, the weight vectors associated with crowded solutions can be adaptively adjusted to the region with sparse non-dominated solutions periodically, which contributes to exploiting the regions with a sparse PF.

Also, it can be observed that the IGD results obtained by MOEA/D-ABM and MOEA/D-URAW in solving DTLZ5, DTLZ6, and WFG3 problems with degenerate PFs, are good in the early evolution process but deteriorate later (Figures A1 and A2). One possible explanation for this phenomenon is that these two algorithms make excessive adjustments to the weight vectors, which spread some weight vectors away from the degenerate PF and lead to many inappropriate convergence tendencies. In contrast, MOEA/D-UR has a

better performance in solving these problems, since it adopts a smaller-scale weight vector adjustment in the later evolution process.

By observing the process IGD mean values of the seven algorithms presented in Figures A1–A3, it can be discerned that MOEA/D-ABM has the best convergence rate in terms of the IGD metric for most test problems in the early evolution process. Especially when dealing with the BT test suit, which possesses significant convergence challenges, MOEA/D-ABM shows an obvious superiority (Figure A3). The underlying reason is that the ABM mechanism reassociates the subproblems with current existing solutions in a competitive manner, enhancing convergence. Meanwhile, it prevents the presence of repeated solutions, thereby maintaining diversity. To more intuitively illustrate the contribution of the ABM mechanism in enhancing convergence, we plot the box-plot corresponding to the improvement ratio (IR) of the total scalarizing functions value attributed to the ABM mechanism in the 30 runs at every 5% of the whole evolution process, as shown in Figure A4. Herein, IR is calculated by the following formula:

$$IR = \frac{\sum_{i=1}^{N} g_{i,old}^{TCH} - \sum_{i=1}^{N} g_{i,new}^{TCH}}{\sum_{i=1}^{N} g_{i,old}^{TCH}}, \tag{28}$$

where $g_{i,old}^{TCH}$ and $g_{i,new}^{TCH}$ denote the scalarizing function value of the i-th subproblem in the population before and after ABM is triggered, respectively. It can be concluded that the implementation of ABM can stably improve the total scalarizing function value of subproblems, and the mean IR is greater in the early evolution process (evident in the results of BT3, BT4, and BT9 problems).

5. Conclusions

In this paper, a modified MOEA/D with an auction-based matching mechanism (MOEA/D-ABM) was proposed for solving MOPs. The proposed auction-based matching (ABM) mechanism is an extension of the auction algorithm. In ABM, each weight vector is associated with its preferred solution by simulating the auction process in economic activities. The preferences of subproblems enhance convergence, while the one-to-one matching model maintains diversity. Furthermore, some additional modifications were employed in MOEA/D-ABM for the enhancement of algorithm performance, including a dynamic normalized procedure and a novel weight vector adjustment strategy. Finally, the experimental study was conducted to compare our proposed MOEA/D-ABM with six state-of-the-art MOEAs in solving some MOP test suits, analyzing the ability and behavior of MOEA/D-ABM. The experimental results demonstrate the competitiveness of MOEA/D-ABM, particularly in solving the DTLZ7 problem and BT test suit, which pose significant convergence challenges.

This paper represents the first attempt to apply auction algorithm theory to MOEA/D in order to enhance the performance of the algorithm. Despite notable achievements, it is apparent that MOEA/D-ABM exhibits some limitations, which provide guidance for future research:

(1) In the cases of DTLZ5, DTLZ6, and WFG3 with degenerate PFs, the IGD results obtained by MOEA/D-ABM deteriorated in the later phase of evolution. As previously analyzed, this may be attributable to the oversized proportion of adjusting weight vectors. One promising method for enhancement would be to adaptively alter the adjustment size based on the estimation of PF.

(2) It can be theoretically proven that the ABM process will terminate after a finite number of auction iterations. However, when the scale of solutions approaches that of subproblems, intensified competition may result in an increase in the number of auction iterations within ABM, leading to a waste of computational resources. Controlling the value of the relaxation parameter ε is the key to addressing this issue.

Author Contributions: G.L. conceived the algorithm; G.L. and G.H. conducted the experiment; M.Z., G.S., Y.M. and H.Z. analyzed the results. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Basic Research Program of Shaanxi (Program No. 2023-JC-YB-629) and the Research Fund of Fundamentals Department of Air Force Engineering University (Program No. JK2022204).

Data Availability Statement: Data available on request from the corresponding author, M.Z., at mingfazheng@stu.xjtu.edu.cn.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. IGD mean and standard deviation values obtained by seven MOEAs for regular MOPs.

Problem	*m*	MOEA/D-URAW	MOEA/D-DU	MOEA/D-UR	NMPSO	AR-MOEA	NSGA-III	MOEA/D-ABM
DTLZ1	3	2.05e-2(2.15e-4)=	2.06e-2(4.71e-5)-	2.19e-2(8.42e-4)-	2.27e-2(9.92e-4)-	2.06e-2(7.35e-5)=	2.06e-2(5.20e-5)=	**2.05e-2(3.67e-4)**
	5	**6.59e-2(1.30e-3)+**	7.10e-2(4.74e-4)-	6.74e-2(1.20e-3)=	6.94e-2(3.30e-3)-	6.87e-2(3.39e-4)=	6.89e-2(2.80e-3)-	6.89e-2(4.32e-3)
	8	1.27e-1(1.24e-2)+	1.15e-1(7.33e-3)+	1.25e-1(1.07e-2)+	3.46e-1(4.64e-1)-	**1.11e-1(5.50e-3)+**	1.59e-1(7.87e-2)=	1.40e-1(1.88e-2)
	10	1.36e-1(2.00e-2)=	1.24e-1(5.10e-3)+	1.36e-1(1.70e-2)=	1.55e+0(6.67e+0)-	**1.14e-1(3.40e-3)+**	1.63e-1(6.86e-2)-	1.41e-1(2.40e-2)
DTLZ2	3	5.34e-2(4.37e-4)+	5.47e-2(1.83e-4)=	**5.28e-2(3.25e-4)+**	7.62e-2(3.00e-3)-	5.45e-2(8.55e-6)=	5.45e-2(2.94e-6)=	5.43e-2(3.90e-3)
	5	2.04e-1(1.80e-3)=	2.18e-1(1.50e-3)-	2.05e-1(1.30e-3)=	2.28e-1(2.00e-3)-	2.12e-1(6.94e-5)-	2.12e-1(1.73e-5)-	**2.02e-1(5.30e-3)**
	8	3.88e-1(3.70e-3)=	3.90e-1(1.30e-3)=	3.93e-1(1.39e-2)-	3.92e-1(2.30e-3)=	**3.87e-1(7.35e-4)+**	4.12e-1(4.81e-2)-	3.90e-1(9.11e-3)
	10	4.37e-1(9.60e-3)-	4.54e-1(1.62e-3)-	4.48e-1(2.07e-2)+	**4.32e-1(2.70e-3)+**	4.38e-1(1.21e-3)=	4.71e-1(4.88e-2)-	4.44e-1(1.16e-2)
DTLZ3	3	5.83e-2(2.50e-3)=	6.11e-2(3.00e-3)-	6.43e-2(3.50e-3)-	2.72e+0(1.99e+0)-	5.85e-2(3.10e-3)=	**5.79e-2(4.60e-3)=**	1.07e-1(3.42e-2)
	5	2.85e-1(3.45e-2)+	2.80e-1(1.56e-1)+	**2.43e-1(2.34e-2)+**	2.36e+0(1.39e+0)-	2.55e-1(1.63e-1)+	8.46e-1(6.99e-1)-	4.57e-1(5.52e-2)
	8	6.36e-1(1.83e-1)=	**4.45e-1(1.21e-1)+**	5.29e-1(8.21e-2)+	1.25e+1(2.24e+1)-	7.12e-1(4.79e-1)-	3.73e+0(2.43e+0)-	6.39e-1(3.20e-1)
	10	**7.10e-1(7.76e-2)+**	7.62e-1(4.33e-1)=	7.11e-1(2.20e-1)+	4.94e+1(5.75e+1)-	1.05e+0(9.27e-1)-	5.26e+0(3.75e+0)-	7.47e-1(8.29e-2)
DTLZ4	3	1.94e-1(2.74e-1)-	**5.47e-2(1.59e-4)+**	1.52e-1(2.24e-1)+	7.70e-2(2.70e-3)=	3.38e-1(3.11e-1)-	1.97e-1(2.52e-1)=	2.05e-1(3.75e-2)
	5	2.65e-1(1.24e-1)-	**2.21e-1(1.20e-3)+**	2.64e-1(8.80e-2)-	2.92e-1(1.11e-1)-	2.54e-1(8.40e-2)=	2.73e-1(1.03e-1)-	2.92e-1(1.29e-1)
	8	4.53e-1(8.19e-2)-	4.50e-1(7.92e-2)-	4.44e-1(4.78e-2)-	5.32e-1(1.20e-1)-	**3.89e-1(1.55e-2)+**	4.85e-1(8.43e-2)-	4.22e-1(6.43e-2)
	10	4.66e-1(2.65e-2)=	4.93e-1(3.61e-2)-	4.70e-1(2.73e-2)=	**4.52e-1(1.93e-2)+**	4.55e-1(4.90e-3)=	4.87e-1(6.76e-2)-	4.74e-1(2.78e-2)
WFG4	3	2.10e-1(2.00e-3)=	2.56e-1(7.60e-3)-	2.27e-1(4.30e-3)-	3.05e-1(1.09e-2)-	2.21e-1(1.88e-4)-	2.21e-1(1.62e-4)-	**2.08e-1(3.40e-3)**
	5	1.16e+0(6.91e-3)=	1.41e+0(4.54e-2)-	1.19e+0(1.14e-2)=	1.35e+0(2.36e-2)-	1.22e+0(9.96e-4)=	1.23e+0(9.46e-4)-	**1.16e+0(5.93e-3)**
	8	3.20e+0(1.49e-2)=	3.76e+0(3.69e-2)-	3.42e+0(3.50e-2)-	3.42e+0(5.80e-2)-	3.57e+0(1.09e-2)-	3.57e+0(3.62e-2)-	**3.20e+0(2.46e-2)**
	10	**4.29e+0(1.96e-2)=**	5.58e+0(6.75e-2)-	4.80e+0(5.27e-2)-	4.31e+0(2.67e-2)=	5.10e+0(2.40e-2)-	5.10e+0(2.44e-2)-	4.36e+0(3.20e-2)
WFG5	3	**2.17e-1(2.20e-3)=**	2.47e-1(3.90e-3)-	2.33e-1(4.60e-3)-	2.93e-1(7.10e-3)-	2.30e-1(1.00e-4)-	2.30e-1(1.00e-4)-	2.23e-1(3.30e-3)
	5	1.12e+0(6.00e-3)=	1.35e+0(4.06e-2)-	1.17e+0(1.07e-2)-	1.27e+0(1.61e-2)-	1.21e+0(1.10e-3)-	1.21e+0(1.30e-3)-	**1.11e+0(9.05e-3)**
	8	**3.16e+0(1.30e-2)=**	3.69e+0(4.06e-2)-	3.35e+0(4.91e-2)-	3.32e+0(3.79e-2)-	3.54e+0(1.50e-2)-	3.52e+0(6.70e-3)-	3.20e+0(2.09e-2)
	10	**4.21e+0(2.37e-2)=**	5.40e+0(5.21e-2)-	4.65e+0(9.21e-2)-	4.27e+0(3.24e-2)=	5.07e+0(3.84e-2)-	5.08e+0(1.63e-2)-	4.29e+0(2.94e-2)
WFG6	3	2.31e-1(1.24e-2)-	2.82e-1(1.33e-2)-	2.47e-1(1.48e-2)-	3.99e-1(3.17e-2)-	2.39e-1(6.80e-3)-	2.39e-1(7.80e-3)-	**2.27e-1(1.19e-2)**
	5	**1.14e+0(6.05e-3)+**	1.54e+0(6.19e-2)-	1.19e+0(1.63e-2)=	1.37e+0(2.18e-2)-	1.22e+0(9.53e-4)=	1.22e+0(1.03e-3)=	1.18e+0(8.15e-3)
	8	**3.20e+0(2.43e-2)=**	3.79e+0(2.19e-2)-	3.40e+0(4.03e-2)-	3.51e+0(4.48e-2)-	3.57e+0(1.13e-2)-	3.61e+0(2.73e-1)-	3.28e+0(2.51e-2)
	10	**4.25e+0(2.84e-2)=**	5.40e+0(3.64e-2)-	4.78e+0(7.37e-2)-	4.74e+0(1.12e-1)-	5.09e+0(4.36e-2)-	5.12e+0(6.49e-2)-	4.28e+0(2.51e-2)
WFG7	3	**2.09e-1(1.85e-3)=**	2.74e-1(1.31e-2)-	2.26e-1(3.64e-3)-	3.09e-1(1.04e-2)-	2.22e-1(2.31e-4)-	2.22e-1(2.00e-4)-	2.13e-1(2.16e-3)
	5	1.16e+0(9.61e-3)=	1.44e+0(7.47e-2)-	1.19e+0(1.22e-2)=	1.37e+0(2.30e-2)-	1.23e+0(2.71e-3)-	1.23e+0(1.58e-3)-	**1.13e+0(8.87e-3)**
	8	3.22e+0(2.11e-2)=	3.80e+0(2.51e-2)-	3.46e+0(3.64e-2)-	3.47e+0(7.45e-2)-	3.56e+0(1.89e-2)-	3.61e+0(1.23e-1)-	**3.20e+0(2.04e-2)**
	10	**4.27e+0(2.42e-2)+**	5.58e+0(6.46e-2)-	4.87e+0(5.83e-2)-	4.43e+0(3.80e-2)-	5.10e+0(4.20e-2)-	5.19e+0(1.72e-1)-	4.40e+0(3.13e-1)
WFG8	3	**2.62e-1(3.41e-3)=**	3.17e-1(6.96e-3)-	3.06e-1(9.61e-3)-	3.47e-1(1.18e-2)-	2.78e-1(2.46e-3)=	2.87e-1(4.85e-3)-	2.68e-1(4.93e-3)
	5	**1.16e+0(7.32e-3)=**	1.60e+0(2.99e-2)-	1.23e+0(1.87e-2)-	1.33e+0(1.84e-2)-	1.23e+0(1.58e-3)=	1.26e+0(5.29e-2)-	1.17e+0(1.07e-2)
	8	3.31e+0(5.66e-2)=	3.97e+0(1.11e-1)-	3.51e+0(1.01e-1)-	3.52e+0(3.83e-2)-	3.64e+0(2.40e-2)-	3.74e+0(1.50e-1)-	**3.30e+0(3.23e-2)**
	10	**4.44e+0(5.27e-2)=**	5.81e+0(1.38e-1)-	4.70e+0(1.17e-1)-	4.65e+0(4.60e-2)-	5.22e+0(3.75e-2)-	5.27e+0(1.11e-1)-	4.48e+0(9.00e-2)
WFG9	3	**2.14e-1(2.33e-2)=**	2.61e-1(5.13e-3)-	2.36e-1(6.31e-3)-	3.34e-1(5.33e-2)-	2.26e-1(2.24e-2)-	2.22e-1(1.68e-3)=	2.16e-1(3.12e-3)
	5	1.14e+0(2.07e-2)=	1.52e+0(3.41e-2)-	1.22e+0(2.83e-2)-	1.27e+0(2.08e-2)-	1.21e+0(7.02e-3)=	1.20e+0(6.93e-3)=	**1.11e+0(1.54e-2)**
	8	3.29e+0(3.74e-2)=	3.73e+0(3.43e-2)-	3.54e+0(6.57e-2)-	3.31e+0(3.63e-2)-	3.55e+0(2.50e-2)-	3.57e+0(6.25e-2)-	**3.24e+0(4.70e-2)**
	10	4.35e+0(5.69e-2)-	5.46e+0(6.91e-2)-	4.82e+0(8.15e-2)-	**4.25e+0(3.37e-2)=**	5.02e+0(4.45e-2)-	5.03e+0(7.12e-2)-	4.42e+0(7.18e-2)
BT3	2	4.67e-2(4.83e-2)-	1.09e+0(2.53e-1)-	5.08e-2(5.30e-2)-	3.59e-1(1.54e-1)-	6.79e-2(5.32e-2)-	1.01e-1(6.69e-2)-	**3.13e-2(4.05e-2)**
BT4	2	**4.94e-2(3.65e-2)+**	9.10e-1(1.76e-1)-	5.78e-2(5.10e-2)=	2.79e-1(1.43e-1)-	6.48e-2(4.51e-2)=	8.78e-2(7.16e-2)-	5.27e-2(5.07e-2)
BT8	2	7.06e-1(2.74e-1)=	1.65e+0(5.35e-1)-	8.01e-1(2.64e-1)-	n.a.	**6.03e-1(2.28e-1)+**	7.11e-1(3.83e-1)=	7.25e-1(3.06e-1)
BT9	3	4.12e-1(1.05e-1)-	2.59e+0(1.20e-1)-	3.20e-1(8.10e-2)-	n.a.	5.79e-1(1.01e-1)-	6.69e-1(1.21e-1)-	**2.31e-1(6.08e-2)**
+/−/=		8/8/28	6/35/3	7/29/8	2/35/5	5/25/14	0/35/9	

Table A2. IGD mean and standard deviation values obtained by seven MOEAs for irregular MOPs.

Problem	*m*	MOEA/D-URAW	MOEA/D-DU	MOEA/D-UR	NMPSO	AR-MOEA	NSGA-III	MOEA/D-ABM
DTLZ5	3	**4.31e-3(4.20e-5)=**	3.17e-2(5.42e-3)-	6.45e-3(3.88e-4)-	1.44e-2(2.81e-3)-	5.48e-3(1.31e-4)-	1.26e-2(2.02e-3)-	4.35e-3(1.01e-4)
	5	4.16e-2(1.01e-2)+	1.89e-1(3.82e-2)-	**2.47e-2(5.20e-3)+**	4.28e-2(6.82e-3)=	7.45e-2(1.04e-2)-	1.10e-1(3.46e-2)-	2.47e-2(7.28e-3)
	8	5.99e-2(1.36e-2)=	2.50e-1(6.87e-2)-	**3.19e-2(7.71e-3)+**	5.97e-1(1.96e-1)-	1.10e-1(2.67e-2)-	2.96e-1(6.79e-2)-	6.15e-2(1.69e-2)
	10	3.22e-2(6.59e-3)+	2.53e-1(3.64e-2)-	**1.72e-2(4.21e-3)**+	7.65e-1(1.04e-1)-	9.35e-2(2.35e-2)-	3.19e-1(9.46e-2)-	4.34e-2(6.09e-3)
DTLZ6	3	**4.32e-3(3.45e-5)=**	2.97e-2(2.81e-3)-	6.77e-3(4.69e-4)-	1.45e-2(2.66e-3)-	5.01e-3(8.17e-5)-	1.92e-2(2.77e-3)-	4.35e-3(8.82e-5)
	5	4.29e-2(1.67e-2)+	1.51e-1(1.16e-1)-	**2.37e-2(3.58e-3)+**	4.96e-2(6.15e-3)+	8.24e-2(1.83e-2)-	3.90e-1(2.3e-1)-	7.35e-2(3.20e-2)
	8	5.41e-2(2.44e-2)=	1.68e-1(3.06e-2)-	**3.26e-2(9.35e-3)+**	6.97e-1(1.62e-1)-	1.35e-1(2.75e-2)-	1.21e+0(7.05e-1)-	5.29e-2(2.35e-2)
	10	1.95e-2(4.52e-3)-	2.04e-1(4.43e-2)-	**1.73e-2(3.55e-3)=**	7.42e-1(3.29e-1)-	1.16e-1(3.58e-2)-	1.46e+0(8.30e-1)-	1.78e-2(3.22e-2)
DTLZ7	3	7.04e-2(5.27e-2)+	1.52e+0(2.31e+0)-	9.64e-2(3.28e-3)-	**7.01e-2(4.52e-3)-**	2.00e-1(2.07e-1)-	1.30e-1(1.56e-1)-	8.27e-2(4.66e-3)
	5	3.22e-1(4.11e-2)-	8.78e+0(2.18e+0)-	4.82e-1(2.61e-2)-	3.08e-1(9.91e-3)=	3.51e-1(8.33e-3)-	3.87e-1(1.23e-2)-	**3.07e-1(1.81e-2)**
	8	1.16e+0(1.83e-1)-	9.14e+0(3.78e+0)-	1.30e+0(2.02e-1)-	8.87e-1(1.22e-1)-	1.73e+0(1.34e-1)-	9.37e-1(4.86e-2)-	**7.79e-1(2.90e-2)**
	10	1.73e+0(2.40e-1)-	2.36e+1(2.71e+0)-	1.61e+0(3.00e-1)-	9.55e-1(8.49e-2)=	2.15e+0(4.79e-1)-	1.46e+0(3.47e-1)-	**9.52e-1(2.99e-2)**
WFG1	3	1.79e-1(2.62e-2)-	1.74e-1(9.92e-3)=	2.08e-1(3.18e-2)-	6.74e-1(2.22e-1)-	**1.59e-1(1.54e-2)+**	1.71e-1(2.02e-2) =	1.65e-1(2.35e-2)
	5	4.89e-1(2.40e-2)=	5.41e-1(2.12e-2)-	6.12e-1(5.92e-2)-	9.89e-1(2.77e-1)-	**4.78e-1(8.15e-3)=**	4.83e-1(1.84e-2)=	4.82e-1(2.42e-2)
	8	**1.06e+0(6.11e-2)=**	1.15e+0(1.22e-1)-	1.44e+0(1.11e-1)-	2.54e+0(8.96e-1)-	1.19e+0(5.88e-1)-	1.09e+0(5.86e-2)=	1.09e+0(6.17e-2)
	10	**1.16e+0(5.04e-2)+**	1.20e+0(6.47e-2)=	1.63e+0(1.31e-1)-	1.47e+0(1.55e-1)-	1.45e+0(6.64e-2)-	1.41e+0(1.41e-1)-	1.25e+0(8.14e-2)
WFG2	3	1.64e-1(3.92e-3)=	1.95e-1(2.95e-3)-	**1.62e-1(4.48e-3)=**	4.44e-1(4.85e-2)-	1.63e-1(1.32e-3)=	1.64e-1(1.18e-3)=	1.68e-1(7.71e-3)
	5	5.15e-1(1.23e-2)=	6.22e-1(2.49e-2)-	**4.93e-1(9.95e-3)+**	1.07e+0(1.52e-1)-	4.98e-1(5.71e-3)=	5.02e-1(3.95e-3)=	5.17e-1(1.54e-2)
	8	1.21e+0(6.14e-2)-	1.21e+0(7.91e-2)-	1.14e+0(3.57e-2)=	1.85e+0(1.77e-1)-	**1.08e+0(2.55e-2)=**	1.19e+0(1.54e-1)=	1.15e+0(5.23e-2)
	10	1.24e+0(2.73e-2)=	1.27e+0(5.97e-2)-	1.23e+0(5.27e-2)-	1.97e+0(1.88e-1)-	1.21e+0(3.94e-2)=	1.31e+0(1.37e-1)-	**1.21e+0(3.73e-2)**
WFG3	3	6.70e-2(3.21e-3)=	2.48e-1(2.27e-2)-	5.45e-2(4.68e-3)+	**4.37e-2(5.14e-3)+**	1.24e-1(7.21e-3)-	1.18e-1(8.55e-3)-	6.67e-2(3.71e-3)
	5	4.05e-1(4.38e-2)-	5.39e-1(2.80e-2)-	**1.82e-1(2.95e-2)+**	3.01e-1(2.88e-2)-	7.01e-1(4.73e-2)-	6.15e-1(9.05e-2)-	2.43e-1(5.06e-2)
	8	1.15e+0(1.69e-1)-	1.15e+0(5.44e-2)-	**5.11e-1(9.77e-2)+**	7.31e-1(1.40e-1)=	2.24e+0(1.92e-1)-	2.09e+0(3.30e-1)-	7.11e-1(1.03e-1)
	10	1.49e+0(1.60e-1)-	1.37e+0(7.92e-2)-	**5.36e-1(9.24e-2)+**	9.77e-1(1.85e-1)-	2.89e+0(2.20e-1)-	2.07e+0(5.68e-1)-	6.41e-1(1.30e-1)
BT5	2	5.47e-1(1.56e-1)-	2.59e+0(2.40e-1)-	6.12e-1(1.44e-1)-	1.59e+0(1.49e-1)-	5.79e-1(1.85e-1)-	7.73e-1(2.37e-1)-	**4.41e-1(1.40e-1)**
+/−/=		5/10/10	0/23/2	10/12/3	2/19/4	1/19/5	0/19/6	

Table A3. HV mean and standard values deviation obtained by seven MOEAs for regular MOPs.

Problem	*m*	MOEA/D-URAW	MOEA/D-DU	MOEA/D-UR	NMPSO	AR-MOEA	NSGA-III	MOEA/D-ABM
DTLZ1	3	8.40e-1(1.09e-3)=	**8.41e-1(4.86e-4)+**	8.37e-1(1.55e-3)=	8.33e-1(2.61e-3)-	8.41e-1(7.65e-4)=	8.41e-1(5.73e-4)=	8.40e-1(1.22e-3)
	5	9.61e-1(2.61e-3)=	**9.71e-1(4.63e-4)+**	9.63e-1(2.75e-3)=	9.51e-1(5.14e-3)-	9.70e-1(4.55e-4)+	9.69e-1(5.36e-3)+	9.61e-1(5.40e-3)
	8	9.81e-1(5.62e-3)=	9.86e-1(3.68e-3)+	9.82e-1(5.91e-3)+	7.18e-1(3.37e-1)-	**9.89e-1(3.46e-3)+**	8.95e-1(2.13e-1)-	9.76e-1(3.58e-2)
	10	9.87e-1(3.24e-2)+	9.87e-1(4.21e-3)=	9.92e-1(4.18e-3)+	8.21e-1(3.06e-1)-	**9.94e-1(2.69e-3)+**	9.38e-1(1.39e-1)-	9.80e-1(2.38e-2)
DTLZ2	3	5.61e-1(1.32e-3)-	5.60e-1(3.59e-5)=	5.60e-1(9.54e-4)=	**5.61e-1(9.65e-4)=**	5.60e-1(5.30e-5)=	5.60e-1(9.07e-6)=	5.51e-1(6.61e-3)
	5	7.65e-1(4.86e-3)=	7.74e-1(6.11e-4)=	7.65e-1(4.52e-3)=	**7.85e-1(2.86e-3)+**	7.74e-1(5.57e-4)=	7.74e-1(4.22e-4)=	7.68e-1(6.85e-3)
	8	8.62e-1(8.75e-3)=	8.84e-1(2.50e-3)+	8.59e-1(1.50e-2)=	**9.14e-1(2.86e-3)+**	8.85e-1(5.67e-4)+	8.66e-1(3.46e-2)=	8.67e-1(1.06e-2)
	10	9.18e-1(1.44e-2)-	9.18e-1(7.71e-3)=	9.15e-1(1.34e-2)-	**9.61e-1(2.17e-3)+**	9.41e-1(8.26e-4)=	9.22e-1(2.68e-2)=	9.27e-1(1.43e-2)
DTLZ3	3	**5.50e-1(4.52e-3)=**	5.48e-1(7.49e-3)=	5.42e-1(8.07e-3)=	7.52e-2(1.76e-1)-	5.39e-1(9.54e-3)=	5.43e-1(1.47e-2)=	5.45e-1(3.20e-2)
	5	6.95e-1(2.12e-2)=	7.19e-1(1.37e-1)+	**7.34e-1(2.12e-2)+**	8.46e-2(2.11e-1)-	7.08e-1(1.38e-1)+	3.23e-1(3.29e-1)-	6.87e-1(5.00e-2)
	8	6.56e-1(1.84e-1)-	**8.11e-1(1.50e-1)+**	7.60e-1(1.03e-1)+	8.73e-2(2.29e-1)-	5.43e-1(3.56e-1)-	8.23e-2(2.32e-1)+	6.90e-1(1.49e-1)
	10	**7.21e-1(5.30e-2)=**	6.18e-1(3.24e-1)-	7.13e-1(2.48e-1)=	0(0)-	5.00e-1(3.91e-1)-	1.51e-2(8.29e-2)-	7.10e-1(2.05e-1)
DTLZ4	3	4.95e-1(1.36e-1)+	5.59e-1(9.17e-5)+	5.13e-1(1.08e-1)+	**5.61e-1(1.30e-3)+**	4.28e-1(1.53e-1)-	4.93e-1(1.21e-1)=	4.70e-1(8.76e-2)
	5	7.53e-1(5.92e-2)=	**7.74e-1(4.68e-4)+**	7.49e-1(3.53e-2)=	7.52e-1(6.33e-2)=	7.56e-1(3.67e-2)=	7.40e-1(5.95e-2)=	7.47e-1(7.16e-2)
	8	8.67e-1(4.16e-2)+	8.57e-1(4.98e-2)=	8.72e-1(2.07e-2)=	7.79e-1(1.60e-1)-	**8.84e-1(5.71e-3)+**	8.24e-1(5.74e-2)-	8.52e-1(2.97e-2)
	10	9.45e-1(8.95e-3)=	9.38e-1(1.55e-2)=	9.50e-1(8.10e-3)=	**9.64e-1(6.25e-3)+**	9.48e-1(1.84e-3)=	9.22e-1(2.96e-2)-	9.41e-1(1.01e-2)
WFG4	3	5.59e-1(1.12e-3)=	5.40e-1(3.05e-3)-	5.53e-1(2.18e-3)=	5.57e-1(1.74e-3)=	5.56e-1(6.19e-4)=	5.56e-1(7.21e-4)=	**5.61e-1(1.31e-3)**
	5	7.66e-1(3.44e-3)=	7.38e-1(9.86e-3)-	7.52e-1(4.32e-3)=	**7.67e-1(3.88e-3)=**	7.62e-1(1.89e-3)=	7.64e-1(1.41e-3)=	7.65e-1(3.63e-3)
	8	8.73e-1(4.01e-3)-	7.76e-1(1.61e-2)-	8.62e-1(5.44e-3)-	**8.83e-1(8.12e-3)=**	8.66e-1(3.52e-3)-	8.62e-1(1.31e-2)-	8.80e-1(5.32e-3)
	10	9.12e-1(5.61e-3)-	8.74e-1(1.31e-2)-	9.17e-1(8.25e-3)=	8.87e-1(7.14e-3)-	9.03e-1(6.87e-3)-	9.03e-1(9.11e-3)-	**9.23e-1(5.61e-3)**
WFG5	3	5.15e-1(3.91e-3)=	5.11e-1(2.14e-3)=	5.08e-1(3.88e-3)=	**5.20e-1(1.17e-3)=**	5.18e-1(1.44e-4)=	5.18e-1(2.39e-5)=	5.18e-1(1.84e-3)
	5	7.08e-1(3.91e-3)-	6.97e-1(1.00e-2)-	6.95e-1(6.14e-3)-	7.20e-1(3.98e-3)=	7.21e-1(1.62e-3)=	**7.23e-1(1.20e-3)=**	7.21e-1(4.55e-3)
	8	8.06e-1(5.01e-3)-	6.87e-1(1.53e-2)-	7.83e-1(8.26e-3)-	**8.21e-1(5.02e-3)=**	8.17e-1(4.18e-3)=	8.18e-1(2.55e-3)=	8.20e-1(4.83e-3)
	10	8.49e-1(4.01e-3)-	7.73e-1(1.30e-2)-	8.43e-1(6.54e-3)-	8.57e-1(3.26e-3)=	8.52e-1(4.23e-3)-	8.55e-1(3.95e-3)-	**8.65e-1(3.84e-3)**
WFG6	3	5.05e-1(1.64e-2)=	4.90e-1(1.55e-2)-	4.97e-1(1.99e-2)=	4.30e-1(3.94e-2)-	5.06e-1(9.40e-3)=	5.06e-1(1.12e-2)=	**5.07e-1(1.65e-2)**
	5	7.00e-1(2.19e-2)=	6.43e-1(3.00e-2)-	6.78e-1(2.24e-2)-	6.10e-1(3.48e-3)-	7.00e-1(1.19e-2)=	6.97e-1(1.23e-2)=	**7.03e-1(2.12e-2)**
	8	7.90e-1(2.79e-2)=	6.52e-1(2.59e-2)-	7.79e-1(2.76e-2)=	7.10e-1(2.35e-3)-	7.95e-1(2.59e-2)=	7.90e-1(2.81e-2)=	**8.01e-1(2.41e-2)**
	10	8.33e-1(2.48e-2)-	7.48e-1(2.63e-2)-	8.24e-1(2.18e-2)=	7.46e-1(1.66e-3)-	**8.41e-1(2.10e-2)=**	8.27e-1(2.01e-2)=	8.40e-1(2.56e-2)
WFG7	3	**5.60e-1(9.01e-4)+**	5.31e-1(5.21e-3)-	5.54e-1(1.85e-3)=	5.59e-1(1.26e-3)=	5.56e-1(5.16e-4)=	5.57e-1(4.84e-4)=	5.53e-1(1.78e-3)
	5	7.65e-1(3.26e-3)=	7.32e-1(1.36e-2)-	7.52e-1(7.88e-3)=	**7.78e-1(3.14e-3)+**	7.60e-1(2.23e-3)=	7.67e-1(1.56e-3)=	7.67e-1(3.11e-3)
	8	8.68e-1(7.95e-3)-	7.84e-1(1.08e-2)-	8.61e-1(5.99e-3)-	**8.99e-1(1.52e-2)+**	8.75e-1(2.71e-3)=	8.67e-1(1.93e-2)-	8.84e-1(5.54e-3)
	10	9.10e-1(7.55e-3)-	8.72e-1(1.54e-2)-	9.22e-1(6.02e-3)=	9.31e-1(4.28e-3)=	9.15e-1(6.65e-3)-	9.01e-1(2.61e-2)-	**9.33e-1(6.39e-3)**
WFG8	3	4.79e-1(3.22e-3)=	4.61e-1(2.09e-3)=	4.69e-1(3.06e-3)=	**4.80e-1(1.64e-3)+**	4.75e-1(1.85e-3)=	4.68e-1(3.03e-3)=	4.71e-1(3.81e-3)
	5	6.58e-1(5.14e-3)=	6.03e-1(6.59e-3)-	6.47e-1(6.11e-3)=	**6.67e-1(3.96e-3)+**	6.57e-1(2.14e-3)=	6.39e-1(8.93e-3)=	6.54e-1(4.66e-3)
	8	**7.74e-1(8.55e-3)+**	6.11e-1(8.94e-3)-	7.54e-1(6.82e-3)=	7.64e-1(6.33e-3)=	7.44e-1(1.91e-2)=	6.83e-1(1.08e-2)-	7.53e-1(7.41e-3)
	10	8.20e-1(9.66e-3)=	6.72e-1(1.23e-2)-	8.20e-1(9.57e-3)=	**8.42e-1(1.19e-2)=**	7.94e-1(1.89e-2)-	7.39e-1(2.04e-2)-	8.17e-1(6.88e-3)
WFG9	3	5.27e-1(2.29e-2)=	5.21e-1(3.22e-3)-	5.28e-1(4.65e-3)=	4.83e-1(6.38e-2)-	5.27e-1(2.27e-2)=	5.32e-1(3.57e-3)=	**5.33e-1(3.44e-3)**
	5	6.76e-1(4.19e-2)-	6.74e-1(4.05e-2)-	6.79e-1(3.01e-2)-	6.68e-1(7.37e-2)-	6.75e-1(2.53e-2)-	**7.05e-1(1.34e-2)=**	7.00e-1(5.04e-2)
	8	6.98e-1(4.55e-2)-	7.10e-1(4.50e-2)-	7.16e-1(5.35e-2)-	7.35e-1(8.27e-2)=	7.40e-1(3.40e-2)=	7.00e-1(5.98e-2)-	**7.56e-1(5.27e-2)**
	10	7.32e-1(3.67e-2)-	7.81e-1(4.83e-2)=	7.54e-1(4.57e-2)-	7.54e-1(7.80e-2)-	7.70e-1(4.97e-2)-	7.67e-1(5.32e-2)-	**8.09e-1(3.88e-2)**

Table A3. *Cont.*

Problem	*m*	MOEA/D-URAW	MOEA/D-DU	MOEA/D-UR	NMPSO	AR-MOEA	NSGA-III	MOEA/D-ABM
BT3	2	6.51e-1(6.37e-2)-	5.20e-3(1.76e-2)-	6.44e-1(7.45e-2)-	2.80e-1(1.30e-1)-	6.08e-1(7.56e-2)-	5.65e-1(8.99e-2)-	**6.72e-1(1.48e-2)**
BT4	2	**6.59e-1(4.53e-2)=**	1.39e-2(2.53e-2)-	6.55e-1(5.04e-2)=	3.63e-1(1.52e-1)-	6.36e-1(6.43e-2)-	6.07e-1(9.57e-2)-	6.54e-1(7.25e-2)
BT8	2	8.43e-2(1.09e-1)-	1.01e-3(4.72e-3)-	4.30e-2(7.08e-2)-	n.a.	1.23e-1(1.10e-1)-	9.86e-2(1.11e-1)-	**6.38e-1(6.54e-2)**
BT9	3	7.80e-2(5.13e-2)-	0(0)-	1.36e-1(7.23e-2)-	n.a.	1.14e-2(2.00e-2)-	8.02e-3(2.00e-2)-	**2.54e-1(1.12e-1)**
+/−/=		4/17/23	8/26/10	5/13/26	9/19/14	6/14/24	2/19/23	

Table A4. HV mean and standard deviation values obtained by seven MOEAs for irregular MOPs.

Problem	*m*	MOEA/D-URAW	MOEA/D-DU	MOEA/D-UR	NMPSO	AR-MOEA	NSGA-III	MOEA/D-ABM
DTLZ5	3	1.99e-1(2.51e-4)=	1.84e-1(3.81e-3)-	1.98e-1(3.03e-4)=	1.96e-1(7.41e-4)=	1.99e-1(2.46e-4)=	1.94e-1(1.15e-3)=	**2.00e-1(3.81e-4)**
	5	1.18e-1(3.35e-3)=	9.98e-2(9.96e-3)-	**1.25e-1(9.43e-4)=**	1.15e-1(2.93e-3)-	1.08e-1(2.82e-3)-	1.06e-1(3.36e-3)-	1.23e-1(3.44e-3)
	8	9.86e-2(1.72e-3)-	9.82e-2(1.24e-3)-	**1.04e-1(6.41e-4)=**	6.37e-2(1.71e-2)-	9.23e-2(1.58e-3)-	8.84e-2(3.55e-3)-	1.01e-1(5.20e-3)
	10	9.67e-2(2.10e-3)=	9.54e-2(6.45e-4)=	**1.01e-1(2.87e-4)-**	1.62e-2(1.94e-2)-	9.20e-2(1.02e-3)-	8.03e-2(7.78e-3)-	9.78e-2(1.90e-3)
DTLZ6	3	1.99e-1(7.00e-5)=	1.86e-1(1.82e-3)-	1.98e-1(2.94e-4)=	1.96e-1(8.70e-4)=	1.99e-1(8.57e-5)=	1.91e-1(1.34e-3)-	**2.00e-1(3.47e-4)**
	5	1.16e-1(8.55e-3)=	1.02e-1(1.97e-2)-	**1.25e-1(1.20e-3)+**	1.15e-1(2.54e-3)=	1.09e-1(3.05e-3)-	7.77e-2(2.86e-2)-	1.15e-1(3.01e-3)
	8	9.60e-2(7.92e-3)-	9.41e-2(5.02e-4)-	1.04e-1(6.95e-4)=	9.10e-2(5.26e-4)-	9.17e-2(1.13e-3)-	1.25e-2(3.13e-2)-	**1.06e-1(2.14e-3)**
	10	9.65e-2(4.55e-3)=	9.35e-2(4.43e-4)-	**1.01e-1(3.16e-4)+**	9.09e-2(6.15e-4)-	9.15e-2(8.15e-4)-	3.05e-3(1.66e-2)-	9.78e-2(8.56e-4)
DTLZ7	3	**2.77e-1(6.21e-3)=**	1.41e-1(7.98e-2)-	2.68e-1(1.35e-3)=	2.75e-1(1.22e-3)=	2.61e-1(2.21e-2)-	2.65e-1(1.58e-2)-	2.72e-1(1.33e-3)
	5	2.46e-1(4.33e-3)-	0(0)-	1.99e-1(8.74e-3)-	**2.59e-1(2.44e-3)=**	2.26e-1(2.38e-3)-	2.33e-1(3.21e-3)-	2.55e-1(3.71e-3)
	8	1.83e-1(4.21e-3)-	2.02e-2(4.11e-2)=	1.51e-1(5.64e-3)-	**2.06e-1(6.44e-3)=**	1.70e-1(1.10e-2)-	1.82e-1(7.96e-3)-	1.98e-1(3.96e-3)
	10	1.73e-1(4.71e-3)-	0(0)-	1.09e-1(2.05e-2)-	**1.99e-1(2.72e-3)+**	1.06e-1(2.04e-2)-	1.42e-1(3.20e-2)-	1.80e-1(6.43e-3)
WFG1	3	9.35e-1(4.95e-3)=	9.31e-1(1.28e-2)-	9.32e-1(5.16e-3)=	7.84e-1(7.73e-2)-	9.41e-1(2.24e-3)=	9.39e-1(7.26e-3)=	**9.43e-1(2.39e-3)**
	5	9.95e-1(7.18e-4)=	9.96e-1(8.31e-3)=	9.93e-1(8.22e-3)=	8.82e-1(6.05e-2)-	9.94e-1(1.32e-2)=	9.92e-1(1.27e-2)=	**9.96e-1(5.86e-4)**
	8	9.99e-1(4.22e-4)=	9.83e-1(3.55e-2)-	9.99e-1(3.20e-4)=	8.96e-1(6.80e-2)-	9.95e-1(1.64e-2)=	9.94e-1(1.62e-2)=	**9.99e-1(1.96e-4)**
	10	9.97e-1(8.29e-3)=	9.89e-1(2.88e-2)=	**9.99e-1(2.35e-4)=**	8.02e-1(9.07e-2)-	9.28e-1(7.58e-2)-	9.78e-1(3.69e-2)-	9.94e-1(5.13e-4)
WFG2	3	9.33e-1(1.15e-3)=	9.19e-1(4.25e-3)-	9.33e-1(8.65e-4)=	8.61e-1(2.01e-2)-	9.30e-1(1.08e-3)=	9.30e-1(8.56e-4)=	**9.33e-1(1.11e-3)**
	5	9.93e-1(1.24e-3)=	9.91e-1(1.90e-3)=	9.92e-1(1.44e-3)=	9.56e-1(9.65e-3)-	9.92e-1(1.44e-3)=	9.92e-1(1.52e-3)=	**9.94e-1(1.46e-3)**
	8	**9.96e-1(1.31e-3)=**	9.89e-1(3.66e-3)=	9.93e-1(3.28e-3)=	9.76e-1(1.00e-2)-	9.92e-1(3.01e-3)=	9.91e-1(3.96e-3)=	9.95e-1(1.75e-3)
	10	9.94e-1(1.70e-3)=	9.92e-1(2.99e-3)=	9.90e-1(3.91e-3)=	9.91e-1(3.88e-3)=	9.83e-1(5.46e-3)-	9.89e-1(5.79e-3)=	**9.95e-1(1.50e-3)**
WFG3	3	3.99e-1(2.51e-3)=	3.17e-1(1.25e-2)-	4.06e-1(1.92e-3)=	**4.07e-1(3.55e-3)=**	3.75e-1(3.80e-3)-	3.79e-1(4.53e-3)-	3.99e-1(2.71e-3)
	5	1.97e-1(1.88e-2)-	1.14e-1(2.78e-2)-	**2.62e-1(6.44e-3)+**	1.81e-1(1.60e-2)-	1.13e-1(1.52e-2)-	1.56e-1(2.61e-2)-	2.21e-1(1.53e-2)
	8	2.01e-3(4.50e-3)-	8.71e-3(1.81e-2)-	**7.29e-2(3.56e-2)+**	4.88e-4(1.50e-3)-	5.15e-4(2.88e-3)-	5.83e-3(1.04e-2)-	2.16e-2(3.00e-2)
	10	0(0)-	0(0)-	0(0)-	0(0)-	0(0)-	0(0)-	**1.62e-3(8.94e-3)**
BT5	2	1.17e-1(9.30e-2)-	0(0)-	8.27e-2(5.84e-2)-	0(0)-	1.18e-1(1.17e-1)-	3.90e-2(5.08e-2)-	**2.06e-1(1.26e-1)**
+/−/=		0/9/16	0/18/7	4/6/15	1/16/8	0/17/8	0/17/8	

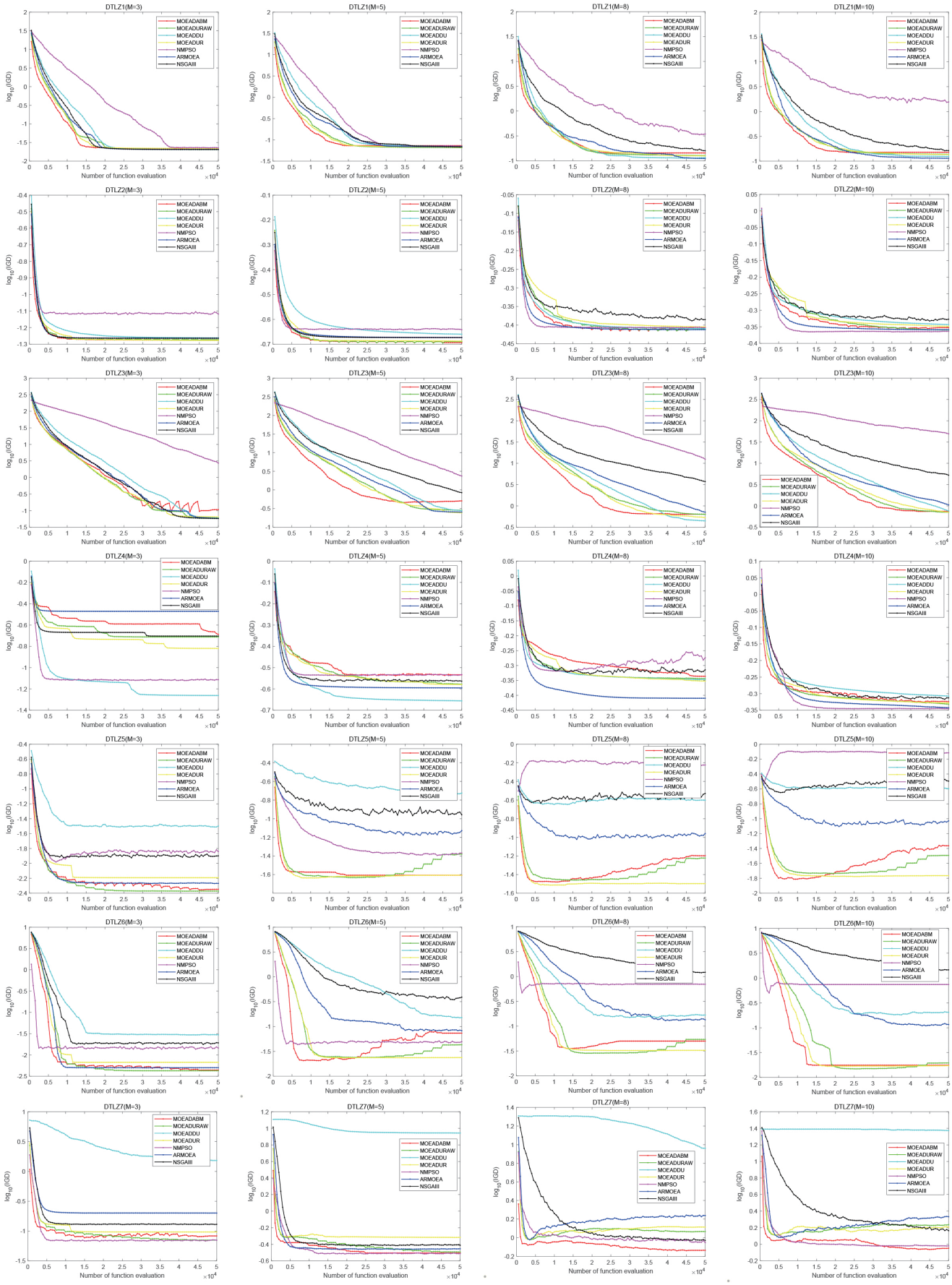


Figure A1. Process IGD mean value of DTLZ1−DTLZ7 for the comparison of seven MOEAs.

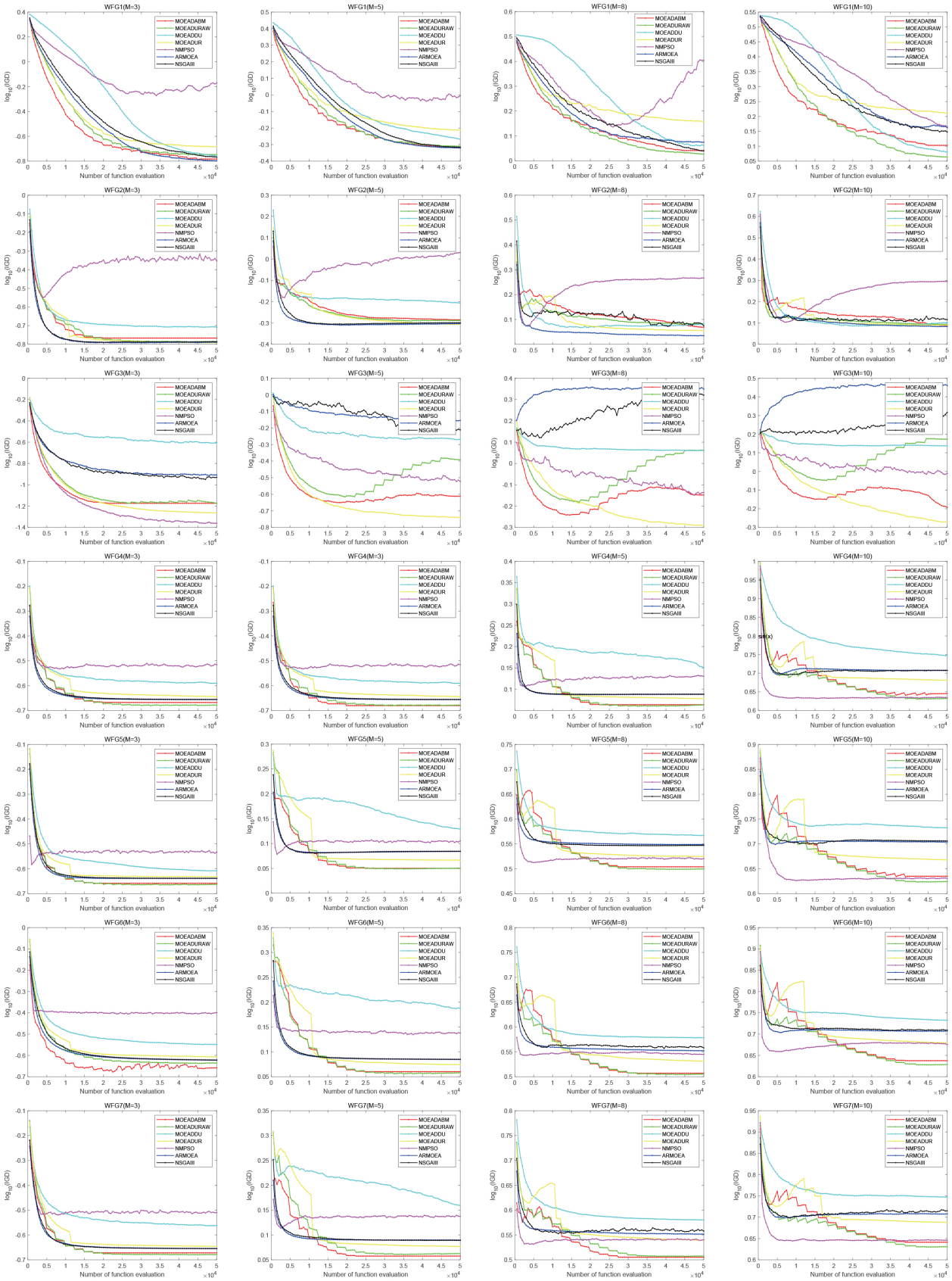


Figure A2. Process IGD mean value of WFG1−WFG7 for the comparison of seven MOEAs.

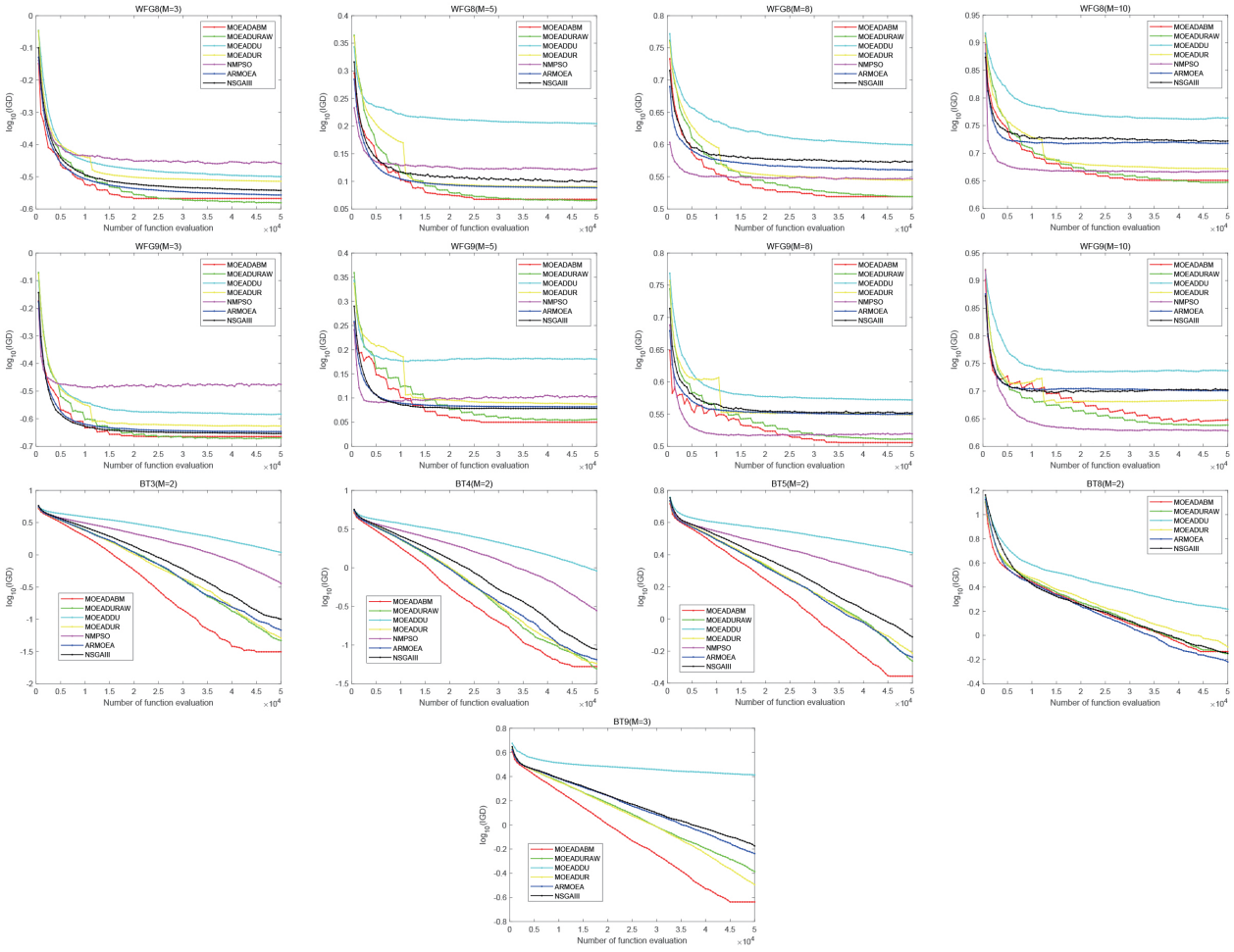


Figure A3. Process IGD mean value of WFG8−WFG9, BT3−BT5, and BT8−BT9 for the comparison of seven MOEAs.

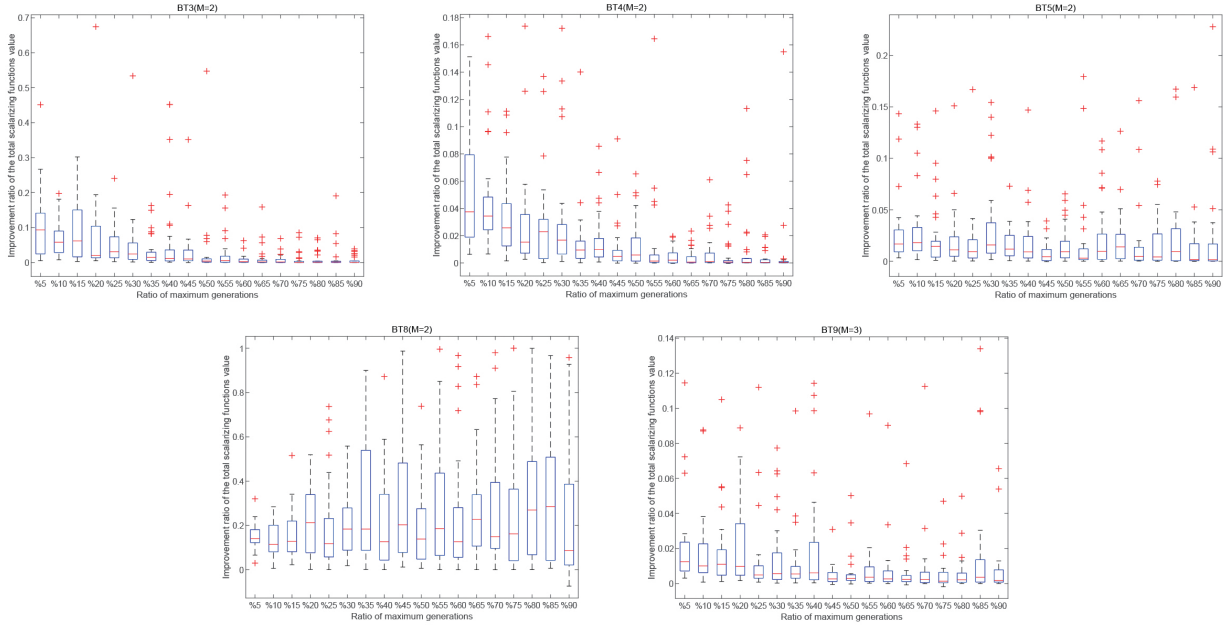


Figure A4. Box-plot corresponding to the improvement ratio of the total scalarizing functions value for solving the BT test suit.

References

1. Huang, Y.; Fei, M.; Zhou, W. Enhanced MOEA/D for trajectory planning improvement of robot manipulator. *Int. J. Robot. Autom.* **2021**, *36*, 91–102.
2. Jiao, H.; Wei, H.; Yang, Q.; Li, M. Application Research of CFD-MOEA/D Optimization Algorithm in Large-Scale Reservoir Flood Control Scheduling. *Processes* **2022**, *10*, 2318. [CrossRef]
3. Kaleybar, H.J.; Davoodi, M.; Brenna, M.; Zaninelli, D. Applications of genetic algorithm and its variants in rail vehicle systems: A bibliometric analysis and comprehensive review. *IEEE Access* **2023**, *11*, 68972–68993. [CrossRef]
4. Tong, S.; Yan, X.; Yang, L.; Yang, X. A Novel Multi-Objective Dynamic Reliability Optimization Approach for a Planetary Gear Transmission Mechanism. *Axioms* **2024**, *13*, 560. [CrossRef]
5. Hernández-Ramírez, L.; Frausto-Solís, J.; Castilla-Valdez, G.; González-Barbosa, J.; Sánchez Hernández, J.P. Three hybrid scatter search algorithms for multi-objective job shop scheduling problem. *Axioms* **2022**, *11*, 61. [CrossRef]
6. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
7. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the strength Pareto evolutionary algorithm. *Evol. Comput.* **2001**, *5*, 121.
8. Wang, X.; Dong, Z.; Tang, L. Multiobjective differential evolution with personal archive and biased self-adaptive mutation selection. *IEEE Trans. Syst. Man, Cybern. Syst.* **2018**, *50*, 5338–5350. [CrossRef]
9. Li, B.; Tang, K.; Li, J.; Yao, X. Stochastic ranking algorithm for many-objective optimization based on multiple indicators. *IEEE Trans. Evol. Comput.* **2016**, *20*, 924–938. [CrossRef]
10. Bader, J.; Zitzler, E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **2011**, *19*, 45–76. [CrossRef]
11. Pamulapati, T.; Mallipeddi, R.; Suganthan, P.N. I_{SDE}+—An Indicator for Multi and Many-Objective Optimization. *IEEE Trans. Evol. Comput.* **2018**, *23*, 346–352. [CrossRef]
12. Falcón-Cardona, J.G.; Coello, C.A.C. Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–35. [CrossRef]
13. Zhang, Q.; Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evol. Comput.* **2008**, *11*, 712–731. [CrossRef]
14. Cheng, R.; Jin, Y.; Olhofer, M.; Sendhoff, B. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 773–791. [CrossRef]
15. Qi, Y.; Li, X.; Yu, J.; Miao, Q. User-preference based decomposition in MOEA/D without using an ideal point. *Swarm Evol. Comput.* **2019**, *44*, 597–611. [CrossRef]
16. Emmerich, M.; Beume, N.; Naujoks, B. An EMO algorithm using the hypervolume measure as selection criterion. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization;* Springer: Berlin/Heidelberg, Germany, 2005; pp. 62–76.
17. Bosman, P.A.; Thierens, D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 174–188. [CrossRef]
18. Wang, R.; Zhang, Q.; Zhang, T. Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods. *IEEE Trans. Evol. Comput.* **2016**, *20*, 821–837. [CrossRef]
19. Junqueira, P.P.; Meneghini, I.R.; Guimarães, F.G. Multi-objective evolutionary algorithm based on decomposition with an external archive and local-neighborhood based adaptation of weights. *Swarm Evol. Comput.* **2022**, *71*, 101079. [CrossRef]
20. Jiang, S.; Yang, S. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts. *IEEE Trans. Cybern.* **2015**, *46*, 421–437. [CrossRef]
21. Jiang, S.; Yang, S.; Wang, Y.; Liu, X. Scalarizing functions in decomposition-based multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2017**, *22*, 296–313. [CrossRef]
22. Wang, J.; Su, Y.; Lin, Q.; Ma, L.; Gong, D.; Li, J.; Ming, Z. A survey of decomposition approaches in multiobjective evolutionary algorithms. *Neurocomputing* **2020**, *408*, 308–330. [CrossRef]
23. Pazhaniraja, N.; Basheer, S.; Thirugnanasambandam, K.; Ramalingam, R.; Rashid, M.; Kalaivani, J. Multi-objective Boolean grey wolf optimization based decomposition algorithm for high-frequency and high-utility itemset mining. *AIMS Math* **2023**, *8*, 18111–18140. [CrossRef]
24. Pavelski, L.M.; Delgado, M.R.; Almeida, C.P.; Gonçalves, R.A.; Venske, S.M. Extreme learning surrogate models in multi-objective optimization based on decomposition. *Neurocomputing* **2016**, *180*, 55–67. [CrossRef]
25. Zhang, S.X.; Zheng, L.M.; Liu, L.; Zheng, S.Y.; Pan, Y.M. Decomposition-based multi-objective evolutionary algorithm with mating neighborhood sizes and reproduction operators adaptation. *Soft Comput.* **2017**, *21*, 6381–6392. [CrossRef]
26. Zhu, Y.; Qin, Y.; Yang, D.; Xu, H.; Zhou, H. An enhanced decomposition-based multi-objective evolutionary algorithm with a self-organizing collaborative scheme. *Expert Syst. Appl.* **2023**, *213*, 118915. [CrossRef]
27. Li, K.; Zhang, Q.; Kwong, S.; Li, M.; Wang, R. Stable matching-based selection in evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput.* **2013**, *18*, 909–923.
28. Yuan, Y.; Xu, H.; Wang, B.; Zhang, B.; Yao, X. Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans. Evol. Comput.* **2015**, *20*, 180–198. [CrossRef]
29. Hong, R.; Xing, L.; Zhang, G. Ensemble of selection operators for decomposition-based multi-objective evolutionary optimization. *Swarm Evol. Comput.* **2022**, *75*, 101198. [CrossRef]

30. Zhou, J.; Yao, X.; Chan, F.T.; Gao, L.; Jing, X.; Li, X.; Lin, Y.; Li, Y. A decomposition based evolutionary algorithm with direction vector adaption and selection enhancement. *Inf. Sci.* **2019**, *501*, 248–271. [CrossRef]
31. Wang, J.; Zheng, Y.; Huang, P.; Peng, H.; Wu, Z. A stable-state multi-objective evolutionary algorithm based on decomposition. *Expert Syst. Appl.* **2024**, *239*, 122452. [CrossRef]
32. Wei, Z.; Yang, J.; Hu, Z.; Sun, H. An adaptive decomposition evolutionary algorithm based on environmental information for many-objective optimization. *ISA Trans.* **2021**, *111*, 108–120. [CrossRef] [PubMed]
33. Zhou, A.; Zhang, Q. Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2015**, *20*, 52–64. [CrossRef]
34. Qi, Y.; Ma, X.; Liu, F.; Jiao, L.; Sun, J.; Wu, J. MOEA/D with adaptive weight adjustment. *Evol. Comput.* **2014**, *22*, 231–264. [CrossRef]
35. Asafuddoula, M.; Singh, H.K.; Ray, T. An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors. *IEEE Trans. Cybern.* **2017**, *48*, 2321–2334. [CrossRef] [PubMed]
36. Luque, M.; Gonzalez-Gallardo, S.; Saborido, R.; Ruiz, A.B. Adaptive global WASF-GA to handle many-objective optimization problems. *Swarm Evol. Comput.* **2020**, *54*, 100644. [CrossRef]
37. Li, M.; Yao, X. What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation. *Evol. Comput.* **2020**, *28*, 227–253. [CrossRef]
38. Ni, Q.; Kang, X. A novel decomposition-based multi-objective evolutionary algorithm with dual-population and adaptive weight strategy. *Axioms* **2023**, *12*, 100. [CrossRef]
39. Jin, Y.; Zhang, Z.; Xie, L.; Cui, Z. Decomposition-based interval multi-objective evolutionary algorithm with adaptive adjustment of weight vectors and neighborhoods. *Egypt. Inform. J.* **2023**, *24*, 100405. [CrossRef]
40. Wang, Z.; Zhang, Q.; Zhou, A.; Gong, M.; Jiao, L. Adaptive replacement strategies for MOEA/D. *IEEE Trans. Cybern.* **2015**, *46*, 474–486. [CrossRef]
41. Wang, Z.; Zhang, Q.; Li, H.; Ishibuchi, H.; Jiao, L. On the use of two reference points in decomposition based multiobjective evolutionary algorithms. *Swarm Evol. Comput.* **2017**, *34*, 89–102. [CrossRef]
42. Wu, M.; Li, K.; Kwong, S.; Zhou, Y.; Zhang, Q. Matching-based selection with incomplete lists for decomposition multiobjective optimization. *IEEE Trans. Evol. Comput.* **2017**, *21*, 554–568. [CrossRef]
43. Li, K.; Deb, K.; Zhang, Q.; Kwong, S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **2014**, *19*, 694–716. [CrossRef]
44. Bao, Q.; Wang, M.; Dai, G.; Chen, X.; Song, Z. Dynamical decomposition and selection based evolutionary algorithm for many-objective optimization. *Appl. Soft Comput.* **2023**, *141*, 110295. [CrossRef]
45. de Farias, L.R.; Araújo, A.F. A decomposition-based many-objective evolutionary algorithm updating weights when required. *Swarm Evol. Comput.* **2022**, *68*, 100980. [CrossRef]
46. Li, M.; Yang, S.; Liu, X. Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 645–665. [CrossRef]
47. Tian, Y.; Cheng, R.; Zhang, X.; Cheng, F.; Jin, Y. An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans. Evol. Comput.* **2017**, *22*, 609–622. [CrossRef]
48. Liu, H.L.; Chen, L.; Zhang, Q.; Deb, K. Adaptively allocating search effort in challenging many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2017**, *22*, 433–448. [CrossRef]
49. de Farias, L.R.; Braga, P.H.; Bassani, H.F.; Araújo, A.F. MOEA/D with uniformly randomly adaptive weights. In Proceedings of the Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 641–648.
50. Liu, Y.; Ishibuchi, H.; Masuyama, N.; Nojima, Y. Adapting reference vectors and scalarizing functions by growing neural gas to handle irregular Pareto fronts. *IEEE Trans. Evol. Comput.* **2019**, *24*, 439–453. [CrossRef]
51. Gu, F.; Cheung, Y.M. Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Trans. Evol. Comput.* **2017**, *22*, 211–225. [CrossRef]
52. Dong, Z.; Wang, X.; Tang, L. MOEA/D with a self-adaptive weight vector adjustment strategy based on chain segmentation. *Inf. Sci.* **2020**, *521*, 209–230. [CrossRef]
53. Bertsekas, D.P. The auction algorithm: A distributed relaxation method for the assignment problem. *Ann. Oper. Res.* **1988**, *14*, 105–123. [CrossRef]
54. Blank, J.; Deb, K.; Roy, P.C. Investigating the normalization procedure of NSGA-III. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 229–240.
55. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. *Scalable Test Problems for Evolutionary Multiobjective Optimization*; Springer: London, UK, 2005; pp. 105–145.
56. Huband, S.; Hingston, P.; Barone, L.; While, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **2006**, *10*, 477–506. [CrossRef]
57. Li, H.; Zhang, Q.; Deng, J. Biased Multiobjective Optimization and Decomposition Algorithm. *IEEE Trans. Cybern.* **2017**, *47*, 52–66. [CrossRef] [PubMed]
58. Miettinen, K. *Nonlinear Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 1999; Volume 12.
59. Pescador-Rojas, M.; Hernández Gómez, R.; Montero, E.; Rojas-Morales, N.; Riff, M.C.; Coello Coello, C.A. An overview of weighted and unconstrained scalarizing functions. In Proceedings of the Evolutionary Multi-Criterion Optimization: 9th

International Conference, EMO 2017, Münster, Germany, 19–22 March 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 499–513.
60. Miettinen, K.; Kirilov, L. Interactive reference direction approach using implicit parametrization for nonlinear multiobjective optimization. *J. Multi-Criteria Decis. Anal.* **2010**, *13*, 115–123. [CrossRef]
61. Das, I. Normal-Boundary Intersection: A New Method for Generating The Pareto Surface in Nonlinear Multicriteria Optimization Problems. *Soc. Ind. Appl. Math.* **1998**, *8*, 631–657. [CrossRef]
62. Jaszkiewicz, A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—A comparative experiment. *IEEE Trans. Evol. Comput.* **2002**, *6*, 402–412. [CrossRef]
63. Zhang, Q.; Liu, W.; Li, H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In Proceedings of the IEEE Congress on Evolutionary Computation, Trondheim, Norway, 18–21 May 2009; pp. 203–208.
64. Agrawal, R.B.; Deb, K.; Agrawal, R.B. Simulated Binary Crossover for Continuous Search Space. *Complex Syst.* **1994**, *9*, 115–148.
65. Kukkonen, S.; Deb, K. A fast and effective method for pruning of non-dominated solutions in many-objective problems. *Parallel Probl. Solving Nat.* **2006**, *4193*, 553–562.
66. He, L.; Ishibuchi, H.; Trivedi, A.; Srinivasan, D. Dynamic normalization in MOEA/D for multiobjective optimization. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8.
67. Lin, Q.; Liu, S.; Zhu, Q.; Tang, C.; Song, R.; Chen, J.; Coello, C.A.C.; Wong, K.C.; Zhang, J. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2016**, *22*, 32–46. [CrossRef]
68. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]
69. Tian, Y.; Cheng, R.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput. Intell. Mag.* **2017**, *12*, 73–87. [CrossRef]
70. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 196–202.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.