symmetry

Article

Symmetrical Generalized Pareto Dominance and Adjusted Reference Vector Cooperative Evolutionary Algorithm for Many-Objective Optimization

Shuwei Zhu [1], Liusheng Zeng [1] and Meiji Cui [2,*]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Road, Wuxi 214122, China ; zhushuwei@jiangnan.edu.cn (S.Z.); 6233112053@stu.jiangnan.edu.cn (L.Z.)
[2] School of Intelligent Manufacturing, Nanjing University of Science and Technology, No. 200 Xiaolingwei Street, Nanjing 210094, China
* Correspondence: cui_mj@njust.edu.cn

Abstract: In Pareto-based many-objective evolutionary algorithms, performance usually degrades drastically as the number of objectives increases due to the poor discriminability of Pareto optimality. Although some relaxed Pareto domination relations have been proposed to relieve the loss of selection pressure, it is hard to maintain good population diversity, especially in the late phase of evolution. To solve this problem, we propose a symmetrical Generalized Pareto Dominance and Adjusted Reference Vectors Cooperative (GPDARVC) evolutionary algorithm to deal with many-objective optimization problems. The symmetric version of generalized Pareto dominance (GPD), as an efficient framework, provides sufficient selection pressure without degrading diversity, no matter of the number of objectives. Then, reference vectors (RVs), initially generated evenly in the objective space, guide the selection with good diversity. The cooperation of GPD and RVs in environmental selection in part ensures a good balance of convergence and diversity. Also, to further enhance the effectiveness of RV-guided selection, we regenerate more RVs according to the proportion of valid RVs; thereafter, we select the most valid RVs for adjustment after the association operation. To validate the performance of GPDARVC, we compare it with seven representative algorithms on commonly used sets of problems. This comprehensive analysis results in 26 test problems with different objective numbers and 6 practical problems, which show that GPDARVC outperforms other algorithms in most cases, indicating its great potential to solve many-objective optimization problems.

Keywords: evolutionary algorithms; generalized Pareto optimality; many-objective optimization; reference vector; cooperative evolution

Citation: Zhu, S.; Zeng, L.; Cui, M. Symmetrical Generalized Pareto Dominance and Adjusted Reference Vector Cooperative Evolutionary Algorithm for Many-Objective Optimization. *Symmetry* **2024**, *16*, 1484. https://doi.org/10.3390/sym16111484

Academic Editor: Marco Montemurro

Received: 9 October 2024
Revised: 31 October 2024
Accepted: 3 November 2024
Published: 6 November 2024

1. Introduction

Multi-objective optimization problems (MOPs) usually involve multiple conflicting objectives that need to be optimized simultaneously. Many real-world problems from engineering and science can be naturally modeled as MOPs [1,2], such as protein structure prediction [3], neural architecture searches [4] and ship hull form designs [5]. Without loss of generality, an MOP can be mathematically represented as follows:

$$\text{minimize } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})), \text{subject to } \mathbf{x} \in \Omega \tag{1}$$

where Ω is the search space of decision variables with solution vector $\mathbf{x} = (x_1, x_2, \ldots, x_D)$. D denotes the dimension of the decision variable and M represents the number of objectives. When M is larger than 3, these problems are regarded as many-objective optimization problems (MaOPs). Over the past two decades, evolutionary algorithms have been dedicated to dealing with MOPs, i.e., multi-objective evolutionary algorithms (MOEAs), have attracted a lot of attention and achieved some developments and applications [5–7]. Nevertheless, existing MOEAs are still faced with huge challenges when dealing with MaOPs. As the number of M increases, most individuals are non-dominated with each other, making it hard to discern their dominant relations and resulting in a loss of selection pressure [8]. Moreover, individuals tend to spread sparsely in the exponentially expanded objective space, posing challenges in maintaining population diversity.

Pareto-based evolutionary algorithms have shown great potential to solve MOPs [9], but their performance deteriorates in high-dimensional objective spaces. To address this issue, researchers have proposed various dominance-based many-objective evolutionary algorithms (MaOEAs), which usually adopt enhanced dominance strategies or introduce additional metrics to enhance selection pressure. Generally, these techniques can be divided into four categories. The first group modifies the definition of Pareto optimality, which can enhance selection pressure by designing a new domination relation [8], such as relaxed Pareto dominance criteria—e.g., CDAS [10], generalized pareto optimality (GPO) [11], and dual-distance dominance [12]. The principle of GPO is to expand the dominance area by enlarging the dominance angle and increasing the selection pressure. The dual-distance-based dominance relation combines with a niche technique that is based on the angle between individuals, where the niche size is dynamically adjusted according to the number of objectives and the evolution status. The second class of dominance-based MaOEAs incorporates additional convergence metrics that increase the selection pressure on the Pareto frontier. For example, Pi-MOEA [13] combines Pareto dominance and diversity estimation based on density [14] to maintain diversity and preserve convergence. KnEA [15] proposed a knee-point-driven strategy that could maintain diversity through a non-dominated solution's bias towards knee-points. The third group of methods, like NSGA-III [16,17] and MOEA/DD [18], introduce reference vectors to manage non-dominated solutions, which aim to make up for poor selection pressure by maintaining population diversity. Apart from the above, some extra strategies and mechanisms have been employed to improve the performance of original MOEAs on MaOPs, which can be considered the fourth group of dominance-based MaOEAs. For instance, evolutionary algorithms with multiple stages can obtain promising performance [19,20], since they focus on convergence in one phase and diversity in the other phase, which is beneficial in striking a good balance throughout the whole evolutionary process.

In comparison to the traditional Pareto dominance relation, relaxed Pareto dominance techniques expand domination regions to better discriminate non-dominated solutions. As discussed above, most existing Pareto-dominance evolutionary algorithms mainly focus on improving population convergence while ignoring their diversity. However, MultiGPO [21], as a parameter-free evolutionary framework, enhances selection pressure by adopting the generalized Pareto dominance relation [11]. Moreover, multiple symmetrical generalized Pareto optimalities are used in MultiGPO to maintain population diversity well [21]. Furthermore, reference vector-based MaOEAs [20,22,23] have achieved some success in solving MaOPs due to their ability to preserve population diversity. However, these algorithms perform worse when dealing with complex problems, especially those with irregular PFs. Therefore, we propose a symmetrical Generalized Pareto Dominance and Adjusted Reference Vector Cooperative (GPDARVC) evolutionary algorithm for MaOPs. The main contributions of our work are as follows:

(1) We propose a new evolutionary algorithm framework based on both symmetrical generalized Pareto dominance (GPD) and anadjusted reference vector cooperative strategy to deal with MaOPs more effectively, where the former enhances selection pressure and the latter maintains population diversity.
(2) To effectively address problems with different Pareto front (PF) shapes, we design an adjusted reference vector mechanism that generates and selects valid reference vectors based on historical evolutionary information.
(3) We conduct comprehensive experiments to validate the performance of our proposed algorithm and demonstrate its superiority on benchmark functions.

The remaining sections of this paper are organized as follows. We describe related works and our motivation in Section 2. The details of our proposed algorithm are given in Section 3. Section 4 presents our experimental design and analysis of the results. Finally, we give our conclusions and future directions in Section 5.

2. Related Works

2.1. Many-Objective Optimization Evolutionary Algorithms

MaOEAs can usually be categorized into three main types, i.e., Pareto dominance-based, decomposition-based, and indicator-based methods. We have briefly introduced Pareto dominance-based algorithms [8,10,11,13,16,17,21] in the Introduction section.

Of the second group, decomposition-based approaches, MOEA/D [24] is the most classic one, whose core principle is to decompose the multi-objective optimization problem into a series of simpler subproblems and then solve subproblems individually with the aim of improving population diversity. Due to its outstanding performance with MaOPs, many scholars have significantly improved and refined MOEA/D. For example, MOEA/AD [25] introduces a dual-population strategy, where the co-evolution of the dual populations promotes a balance between population diversity and convergence. MOEA/FC [26] employs flexible reference points and a novel density estimator in SPEA/R [27] to enhance population diversity. Regarding problems with irregular PF, decomposition-based MaOEAs have also shown advancements. In methods like CARV-MOEA [22], SPEA/ARP [28], and MaOEA/D-CIL [29], adaptive reference point strategies have been proposed to adjust reference vectors dynamically, enabling them to approximate the accurate PF distribution more accurately.

Indicator-based MaOEAs use performance indicators to find solutions that better balance convergence and diversity, thereby guiding the population toward the PF. Currently, popular performance indicators include hypervolume (HV), inverted generational distance (IGD), I^{ϵ^+}, R2, and enhanced IGD (IGD-NS). Due to the HV's favorable theoretical properties and Pareto compliance, several HV-based MaOEAs [30,31] have been proposed. Sun et al. proposed an IGD-based evolutionary algorithm, MaOEA-IGD [32]. IF-MaOEA [33] introduces the concept of optimal distribution of individuals based on the IGD indicator, ensuring the distribution of the evolutionary process and preventing the algorithm from converging to local optima. The authors of Ref. [34] designed an IGD-NS to select elite individuals for the next generation. In R2HCA-EMOA [35], R2 indicator variables are used to approximate HV contributions to select the next generation of individuals, and MaOEA-DISC [36] focuses on the spacing relationships among individuals within the population based on the I^{ϵ^+} indicator. It proposes a new enhanced diversity I^{ϵ^+} indicator to ensure increased diversity in the population while maintaining convergence. In addition, using only a single metric to select individuals is prone to bias and thus the reduced generality of the algorithm. Thus, some algorithms based on multiple indicators have been generated, such as 1by1EA [37] and 2REA [38].

AREA-APA [20] and MOEA/DD [18] belong to the hybrid class of algorithms that combine the advantages of the above methods and show promising results in handling MaOPs. Recently, many convergence and diversity strategies have been proposed. For example, RVEA-2DCES [39] introduces two new strategies, namely, adaptive sparse region detection and convergence-only selection. In CEEA [40], a cascading elimination strategy based on binary quality indicators and balanced fitness estimation is proposed. Additionally, some MaOEAs based on multi-stage mechanisms have been developed [41–43]. In addition, the idea of combining multi-/many-objective optimization and machine learning has become popular, based on which some learning-assisted MaOEAs [44,45] have been designed to address more complex problems [46].

However, for practical applications with expensive function evaluations, the observation of many objective functions with one algorithm needs more functional evaluations, which are time-consuming (i.e., computationally expensive). In order to balance the trade-off between time consumption and efficiency, surrogate-assisted methods can be used in

MaOEAs. In recent years, surrogate-assisted evolutionary algorithms (SAEAs) [1,2,47] have attracted much attention. Generally, surrogate models are trained using historical or real-time data of the optimization problems, and they can be used to replace the majority of actual models for the purpose of the rapid fitness evaluation.

2.2. Property Analysis of Symmetrical GPD

To gain a better understanding of Pareto-based MaOEAs, we give the definitions of traditional Pareto domination [9] and generalized Pareto domination [11] as follows.

Definition 1 (**Pareto dominance**). *For two solution vectors* **x** *and* **y***, the solution* **y** *is said to dominate the solution* **x***, denoted* $\mathbf{x} \prec \mathbf{y}$*, if and only if*

$$\begin{cases} \forall i \in \{1,2,\ldots,M\}, & f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \\ \exists j \in \{1,2,\ldots,M\}, & f_j(\mathbf{x}) < f_j(\mathbf{y}) \end{cases} \tag{2}$$

Definition 2 (**Generalized Pareto Dominance**). *A solution* **x** *is said to generally dominate another solution* **y** *with respect to (w.r.t.) the expanding angle vector* $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_M]$ *(denoted* $\mathbf{x} \prec_\varphi \mathbf{y}$*), if and only if* $f(\mathbf{x})_\varphi$ *is partially less than* $f(\mathbf{y})_\varphi$*, that is*

$$\begin{cases} \forall i \in \{1,2,\ldots,M\}: & f_i(\mathbf{x}) + \sum_{k \neq i} \delta_i f_k(\mathbf{x}) \leq f_i(\mathbf{y}) + \sum_{k \neq i} \delta_i f_k(\mathbf{y}) \\ \exists j \in \{1,2,\ldots,M\}: & f_j(\mathbf{x}) + \sum_{k \neq j} \delta_j f_k(\mathbf{x}) < f_j(\mathbf{y}) + \sum_{k \neq j} \delta_j f_k(\mathbf{y}) \end{cases} \tag{3}$$

where $\delta_i = \sqrt{M-1} \cdot \tan(\varphi_i/(M-1))$.

2.2.1. Symmetrical GPD-Based Ranking Scheme

The symmetrical GPD adopts multiple symmetric $(M-1)$-GPD versions for solution ranking of an M-objective optimization problem. Here, "$M-1$" indicates that $M-1$ objectives expand the dominance area of solutions to improve selection pressure. To take a bi-objective optimization problem as an example, Figure 1 shows a graphical explanation of symmetric $(M-1)$-GPD in the original f_1-f_2 objective space and the indirect f_1-Ω_2 (blue) and Ω_1-f_2 (green) objective spaces; for more details, please refer to [21].

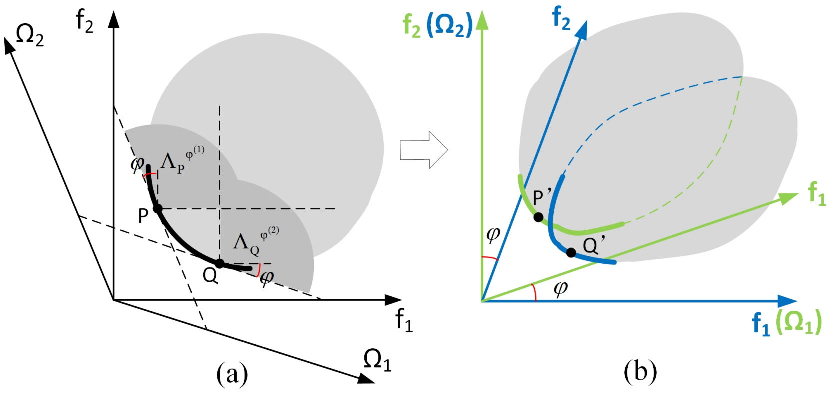


Figure 1. Pictorial illustration of a two-dimensional objective space and the shrunken space after performing two symmetrical $(M-1)$-GPD cases. (**a**) The original f_1-f_2 objective space and expanded Ω_1-Ω_2 objective space; (**b**) the two symmetrical generalized indirect f_1-Ω_2 (blue) and Ω_1-f_2 (green) objective spaces.

2.2.2. Additional Theoretical Study on Property Analysis

Given a vector Φ^i in which all elements take the values $\hat{\varphi}$ except the i-th element which is 0 (i.e., $\varphi_k = \hat{\varphi}$, $k \neq i$, and $\varphi_i = 0$), the theoretical proofs of asymmetric, transitive and irreflexive properties of $(M-1)$-GPD can be deduced according to the definition of GPO (Definition 2), as shown below.

Property 1 (Asymmetry). *For any two solutions x, y, if $x \prec^{\Phi^i} y$, then $y \nprec^{\Phi^i} x$.*

Proof. Suppose $F'(\mathbf{x}) = [f_1'(\mathbf{x}), \ldots, f_i(\mathbf{x}), \ldots, f_M'(\mathbf{x})]$, where $f_j'(\mathbf{x}) = f_j(\mathbf{x}) + \sum_{k \neq j} \delta_k f_k(\mathbf{x})$, $\delta_k = \frac{\tan\varphi_k}{\sqrt{M-1}}$ and is the same as $F'(\mathbf{y})$.

By $\mathbf{x} \prec^{\Phi^i} \mathbf{y}$, we have $f_k'(\mathbf{x}) < f_k'(\mathbf{y}), k \in [1, M], k \neq i$, and $f_i(\mathbf{x}) < f_i(\mathbf{y})$. That is to say, $f_k'(\mathbf{y}) \nless f_k'(\mathbf{x}), k \in [1, M], k \neq i$, and $f_i(\mathbf{y}) \nless f_i(\mathbf{x})$, so $\mathbf{y} \nprec^{\Phi^i} \mathbf{x}$. □

Property 2 (Transitivity). *For any three solutions x, y and z, if $x \prec^{\Phi^i} y$ and $y \prec^{\Phi^i} z$, then $x \prec^{\Phi^i} z$.*

Proof. Suppose $F'(\mathbf{x})$, $F'(\mathbf{y})$ and $F'(\mathbf{z})$ the same definition when proving Property 1. With the given conditions, we have

$$\begin{cases} f_k'(\mathbf{x}) < f_k'(\mathbf{y}), k \in [1, M], k \neq i, \text{ and } f_i(\mathbf{x}) < f_i(\mathbf{y}), \\ f_k'(\mathbf{y}) < f_k'(\mathbf{z}), k \in [1, M], k \neq i, \text{ and } f_i(\mathbf{y}) < f_i(\mathbf{z}). \end{cases}$$

Hence, it can be deduced that $f_k'(\mathbf{x}) < f_k'(\mathbf{z})$, $k \in [1, M]$, $k \neq i$, and $f_i(\mathbf{x}) < f_i(\mathbf{z})$, i.e., $\mathbf{x} \prec^{\Phi^i} \mathbf{z}$.

Moreover, by replacing $\mathbf{y}$ with another $\mathbf{x}$ in Property 1, we can deduce that if $\mathbf{x} \prec^{\Phi^i} \mathbf{x}$, then $\mathbf{x} \nprec^{\Phi^i} \mathbf{x}$, which is a contradiction. Thus, for any candidate solution $\mathbf{x}$, $\mathbf{x} \nprec^{\Phi^i} \mathbf{x}$, i.e., the $(M-1)$-GPD relationship is irreflexive. □

2.3. Reference Vector Adaptation

Multi-objective optimization algorithms with fixed reference vectors face challenges, including overly dense solution set distributions and below-standard convergence, especially when dealing with MOPs with irregular PFs. Many scholars have proposed strategies to adapt and adjust the reference vectors. These strategies aim to change the distribution of the reference vectors, explore promising regions, and achieve uniformly distributed and well-converged solution sets [23,48,49]. There are two main distinctions between these strategies, i.e., when to adapt the reference vectors and what methods are used to adapt the modified reference vectors.

The timing of adjusting the reference vectors is critical. Adjusting them frequently can lead to solution instability and slower convergence, while changing them too late can cause population searching in the wrong direction. Currently, most algorithms determine when the reference vector needs to be adjusted based on whether solutions reach some standard threshold. For example, SPARVEA [23] introduces the concept of solution potential to determine whether the convergence direction of an ideal solution has potential, based on which an adaptive strategy based on solution potential is designed. MaOEA/D-2ADV [48] proposes adjusting the number of reference vectors if the variation of the solution is less than 10^{-4} in each of the M directions, indicating that all subproblems have converged well.

The method of adjusting reference vectors is also crucial, especially when solving complex problems. In recent years, three strategieshave been proposed for adjusting reference vectors, as follows: (1) Adjusting based on existing reference vectors. Existing reference vectors are often partly valid reference vectors, which can guide the population to evolve in the right direction. The generation of new reference vectors can also rely on existing promising reference vectors. For example, the search is divided into two phases in [48,50]. In the first phase, the search proceeds along the boundary reference vectors. In the second stage, new reference vectors are generated by more promising reference vectors, while inactive reference vectors are replaced by interpolations based on the active reference vectors in MOEA/D-2ADV [48]; (2) Utilizing solution candidates to generate reference vectors. Candidate solutions are the promising solutions left after round-by-round elimination and often indicate potential regions of the true PF. Hence, the generation of reference vectors using this method is reliable. In MOEA/DAWA [51], a profile is kept to evaluate the sparsity of the reference vector using neighborhood distances. After a certain number of fixes are generated, the crowded reference vectors are removed, and new reference vectors are added to the sparse regions using the solutions in the archive; (3) Adjusting reference vectors by machine learning. There are often some hard-to-discover mapping relationships between the values of the objective function and the decision variables. We expect that mining the promising solutions in each generation of the population using machine learning can be used to understand the distribution of the PFs, as already demonstrated in the study of Suresh et al. [49]. Each solution's decision variables and objective function values are scaled to values between 0 and 1 by a specific deflation method. Then, the deflated decision variables and objective function are used as the inputs and outputs of the artificial neural network for training. The decision maker can predict promising solutions in any region using the learned model. Thus, this can help us to generate reliable reference vectors.

2.4. Motivation

Although many classic algorithms have been proposed to solve MaOPs, they are still faced with challenges in striking a balance between convergence and diversity. Some scholars have focused on increasing the algorithm's selection pressure by improving the traditional domination approach, which maintains population convergence yet lacks diversity at the late stage of evolution. The reference vector-based approach approximates the PF along the reference vector from multiple directions, allowing for good population diversity. In addition, some PF regions need to be explored sufficiently for some complex problems with irregular shapes. Based on the above analysis, a question arose—is it possible to develop an algorithm that can keep a good balance between convergence and diversity for MaOPs and perform efficiently on irregular or complex issues? Based on this question, we propose a Generalized Pareto Dominance and Adjusted Reference Vectors Cooperative evolutionary algorithm for MaOPs.

3. Proposed Algorithm

The key issue in solving MOPs lies in achieving a delicate balance between convergence and diversity. As mentioned before, the symmetrical GPD can provide enough selection pressure in a many-objective space, which achieves good convergence with problems of various scales. Additionally, reference vector-based methods maintain satisfactory population diversity by distributing reference points throughout the entire space. Therefore, we propose a cooperative strategy that combines GPD and reference vectors (RVs) for environmental selection, leveraging their respective strengths.

3.1. Overall Framework

The overall framework of GPDARVC is shown in Figure 2. The initial population is randomly generated using Latin hypervolume sampling (LHS), and a set of RVs are generated through Riesz *s*-Energy Method [52]. Then, mating selection and reproduction operations (crossover and mutation) are executed to generate offspring solutions. Binary tournament selection is employed to identify promising solutions for recombination, with simulated binary crossover (SBX) and polynomial mutation (PM) [16] serving as the reproduction operators within this framework. The main distinction between traditional MaOEAs and our proposed GPDARVC lies in the environmental selection part. Specifically, GPDARVC makes full use of GPD and RV cooperation to conduct efficient survival selection of population. Moreover, we can adjust the RVs once according to their validity estimated by historical evolutionary information if needed. We generate additional RVs based on the proportion of valid RVs and then adjust the original RVs accordingly. After environmental selection, the surviving population is obtained for the next generation. The pseudo-code of GPDARVC is presented in Algorithm 1.

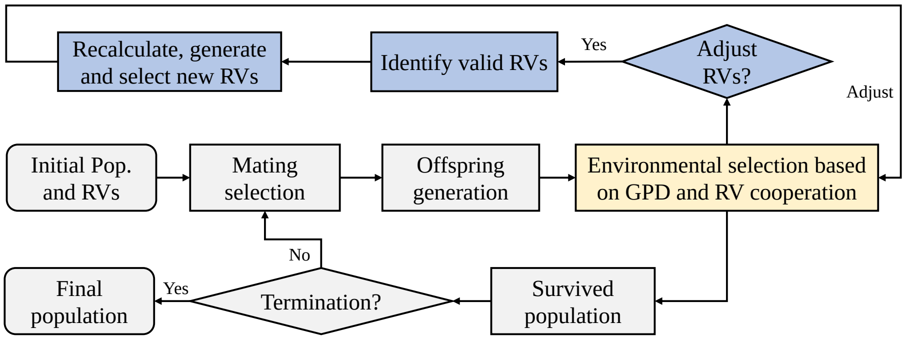


Figure 2. Overall framework of GPDARVC.

Algorithm 1 Pseudo-code of GPDARVC

Require: N: population size, φ: expanding angle, Max_FE: maximum number of fitness evaluations, FE=0: consumed fitness evaluations, α: control parameter;
Ensure: P: final population;
$P \leftarrow$ Initialize();
$Z \leftarrow$ Riesz-s-Energy(N,M); % % Generate RVs by energy minimization method
while $FE \leq \alpha \times$Max_FE **do**
    $P' \leftarrow \text{Mating\_Selection}(P)$;
    $Q \leftarrow \text{Reproduction}(P')$;
    $R \leftarrow P \cup Q$;
    $(P, FE) \leftarrow \text{EnvironmentalSelection}(R, N, \varphi, Z, FE)$;
end while
$(Z_{valid}, pi, d) \leftarrow \text{Association}(P, Z)$; % % Identify valid RVs through Algorithm 2
$N_Z \leftarrow \text{int}\ (N(N/|Z_{valid}|))$; % % Recalculate the number of required RVs
$Z_{new} \leftarrow \text{Riesz-s-Energy}(N_Z, M)$;
$(Z_{adjust}, pi, d) \leftarrow \text{Association}(P, Z_{new})$;
while $FE \leq$Max_FE **do**
    $P' \leftarrow \text{Mating\_Selection}(P)$;
    $Q \leftarrow \text{Reproduction}(P')$;
    $R \leftarrow P \cup Q$;
    $(P, FE) \leftarrow \text{EnvironmentalSelection}(R, N, \varphi, Z_{adjust}, FE)$; %% See Algorithm 3
end while

Given two parent individuals x_1 and x_2, the SBX operation can be expressed as follows:

$$y_1 = \frac{1}{2}(x_1 + x_2) + \frac{u}{2}(x_2 - x_1) \tag{4}$$

$$y_2 = \frac{1}{2}(x_1 + x_2) - \frac{u}{2}(x_2 - x_1) \tag{5}$$

where u is a random variable, calculated by Equation (6). Here, r is a uniformly distributed random number in the range $[0,1]$, and η is a parameter that controls the strength of the crossover operation.

$$u = \begin{cases} \left(\frac{2}{r}\right)^{\frac{1}{\eta+1}} & \text{if } r \leq 1 \\ \left(\frac{1}{\frac{2}{r}-1}\right)^{\frac{1}{\eta+1}} & \text{if } r > 1 \end{cases} \tag{6}$$

Given solution x and the mutation probability p_m, the PM operation is formulated as follows:

$$y = \begin{cases} x + \Delta & \text{if } r < p_m \\ x & \text{if } r \geq p_m \end{cases} \tag{7}$$

where the calculation of Δ is

$$\Delta = \begin{cases} (x_{\text{ub}} - x)^{\eta+1} \cdot rand & \text{if } r < 0.5 \\ (x - x_{\text{lb}})^{\eta+1} \cdot rand & \text{if } r \geq 0.5 \end{cases} \quad (8)$$

Here, x_{ub} and x_{lb} are the upper and lower bounds of the decision variable, respectively; *rand* is a random number in the range $[0, 1]$; and η is a parameter that controls the strength of the mutation.

3.2. GPD and RV Cooperative Environmental Selection Strategy

The purpose of environmental selection is to pick up promising solutions for the next generation, which plays a crucial rule in the whole evolutionary process. To enhance selection pressure, we propose a cooperative GPD and RV strategy, with the aim of speeding up the convergence and keeping population diversity.

Reference vector-guided searches are a widely employed strategy for MOPs. RVs can be uniformly distributed, as defined by users, and therefore, the solutions obtained tend to have good diversity. However, the definition of RVs significantly impacts the final results. Pre-defined RVs should consider the true PFs of the specific problem at hand. In other words, evenly distributed RVs, the commonly used ones, are effective only for MOPs with regular PFs and may not adequately handle MOPs with irregular or complex PFs. In contrast to traditional strategies guided by RVs, we propose two key points: (1) selecting only a subset of solutions according to RVs rather than all solution and (2) having each RV guide, at most, one solution throughout the process. Specifically, once an RV guides the selection of a solution in an iteration, it cannot guide any further selections within that same iteration. It is important to note that energy minimization method is utilized to generate these RVs due to its ability to produce any desired number of them.

The GPD technique, inspired by the CDAS strategy [10], effectively controls the dominance area of the solutions, thereby enhancing selection pressure. Zhu et al. [21] propose a framework based on it that performs M symmetric (M-1)-GPD-based sorting operations for all solutions simultaneously, resulting in M distinct solution sorting schemes.

Building upon this framework, we initially select solutions based on RVs and subsequently employ the max–min distance strategy to choose the remaining solutions. More specifically, our (M-1)-GPD framework ensures prior selection of non-dominant solutions, while remaining solutions can be obtained through guidance from uniformly distributed RVs. However, the true PFs are usually unknown, making the exact number of solutions selected by RVs indeterminate. MOPs with regular PFs tend to have more solutions selected in this step, whereas those with irregular or inverted PFs have fewer solutions selected. To address this, we use the maximum–minimum distance to determine the selection of remaining solutions. This strategy helps maintain a good balance between convergence and diversity, especially when some well-distributed solutions have already been selected.

The environmental selection process primarily involves the cooperation of two steps: reference vector-guided selection and GPD-based selection. The pseudo-code of environmental selection is shown in Algorithm 3. Firstly, some extreme points are determined, and the number of solutions selected by M GPD conditions is calculated. The candidate solutions for selection are firstly divided by the GPD according to their domination levels. For reference vector-guided selection, solutions are paired with RVs based on cosine distance (Association$(\cdot)$ in Algorithm 2). Specifically, each solution is associated with its closest RV, and each RV pairs with at most one solution. If an RV is associated with multiple solutions, it selects the solution closest to it as the final pairing. Once the pairing operation is completed, solutions with the paired RV are selected. Some RVs may not have any solution to pair with them, resulting in fewer selected solutions than the preset population size N. To address this, the remaining solutions are selected through the maximum–minimum distance strategy. Based on the selected solutions, the cosine distance between each candidate solution and the chosen ones is calculated. Candidate solutions are then selected one by one based on the minimum–maximum distance until the required number of solutions is met.

3.3. Reference Vector Adjustment

It is known that the adjustment of reference vectors every time means a lot of sensitivity issues; however, we adjust the reference vectors only once, as shown in Algorithm 1 (see line 12 of Algorithm 1). Our adjustment is intended to compensate for the shortcomings of the predefined reference vectors, which is partly according to [53]. In order to further improve the efficiency of reference vector-guided selection, we propose adjusting RVs during the evolutionary process. Firstly, RVs are considered valid if they are associated with solutions. Subsequently, additional RVs are generated based on the proportion of effective RVs, and increasing their number results in a higher density of RVs. We re-associate the newly generated RVs with the existing population, and those associated RVs are used for adjustment.

Here, we discuss the details of the RV adjustment process. Firstly, N evenly distributed RVs are generated by the energy minimization method, where N is the population size. As evolution continues, the GPD provides sufficient selection pressure, causing some RVs far from the real PFs to gradually cease working, as shown in Figure 3. Therefore, it it necessary to identify the valid RVs so as to guide efficient selection. We consider RVs invalid when they do not pair with any non-dominated solutions. After several iterations, RVs can be classified into valid and invalid groups according to the pairing condition. For problems with regular PFs, more effective RVs are identified at this stage. Correspondingly, fewer effective RVs are identified for problems with irregular or complex PFs. To ensure enough valid RVs, we need to re-generate some RVs after the identification process. To better understanding this, an illustrative example is provided, as shown in Figure 4, where 20 RVs are initially pre-defined. After several generations, eight valid RVs are determined (as shown in green dot). The proportion of valid to invalid RVs is 0.4. To make the number of effective RVs equal to the predefined RVs, we need to re-generate 2.5 times the predefined number of RVs. Thus, 50 predefined RVs need to be re-generated, thereby obtaining 20 valid RVs after this adjustment process. It is shown in Figure 4 that a few valid RVs are sparsely distributed in the objective space, which is not beneficial for effective environmental selection. Therefore, we re-generate more valid RVs according to the proportion, causing them to be more densely distributed in the objective space. Therefore, we can ensure a sufficient number of valid RVs according to the calculated proportion.

Algorithm 2 Association Operator

Require: P: population, Z: reference vectors;
Ensure: ARV: Associated RVs, pi: RVs' index associated with each solution, d: distance of the solution to its nearest RV;
1: Calculate the cosine distance D_{ij} from each solution to each RV; %% D_{ij} is the distance matrix
2: $(d, pi) = min(D_{ij})$; %% d is the minimal distance of each solution to all RVs, pi is the index of the minimal value;
3: $ARV = Z(pi)$; %% The RV with index pi is the associated RV;
4: Return (ARV, pi, d);

Algorithm 3 Environmental Selection

Require: N: population size, R: combined population, φ: expanding angle, Z: reference vector;
Ensure: P : $population$

$P = \emptyset$ and conduct fast non-dominated sorting on R;
Use the AGPO sorting method to produce each front PF_i, $i = 1, 2, \ldots, M$;
Suppose $t_i = \left\lfloor \frac{N-|Q|}{M} \right\rfloor$, $i = 1, 2, \ldots, M$;
$n_s = N - |Q| - M \times t$;
Randomly select n_s different t_i = $[t_1, t_2, \ldots, t_m]$ and increase their values by 1;
$(ARV, pi, d) \leftarrow$ Association(R, Z); %% through Algorithm 2;
Suppose $Zchoose_k = false$, $k = 1, 2, \ldots, |Z|$;
Let $Zchoose_k = true$ where k is the index of the RVs in the ARV; %% marks associated reference vectors as true
Identify the extreme solution set Q in terms of the minimum ASF value for each condition;
$P \leftarrow P \cup Q$, and $R \leftarrow R \setminus Q$;
for $j = 1$ to M **do**
  $PF^* = PF_j$;
  Let $TZchoose$ = $Zchoose$; %% generate a temporary reference vector marker
  $i = 0$;
  while $i < t_j$ and $|P| < N$ **do**
    Let R_{nd} be the set of all non-dominated solutions in R;
    $RT = \{x | x \in R_{nd}, PF^*(x) = \min(PF^*(R_{nd}))\}$;
    if RVs' markers $TZchoose$ are not all false **then**
      Randomly select one from the RVs marked as true and record its index as k;
      Select the solution set I from RN where the pi value is equal to k;%% Select all solutions associated with the reference vector index k;
      if I is not empty **then**
        $Zchoose_k = false$ and $TZchoose = Zchoose$
        $s = \arg\min(d_{min}(I))$;
        $P \leftarrow P \cup \{s\}$, $R \leftarrow R \setminus \{s\}$;
        $i = i + 1$;
      else
        $TZchoose_k = false$ and continue
      end if
    else
      Calculate cosine distance dc between any two solutions in RT and P,
      Calculate $dc_{min}(x) = \min_{y \in P} \text{dist}(x, y)$ for each solution $x \in RT$;
      $s = \arg\max(dc_{min}(RT))$;
      $P \leftarrow P \cup \{s\}$, $R \leftarrow R \setminus \{s\}$;
      $i = i + 1$;
    end if
  end while
end for

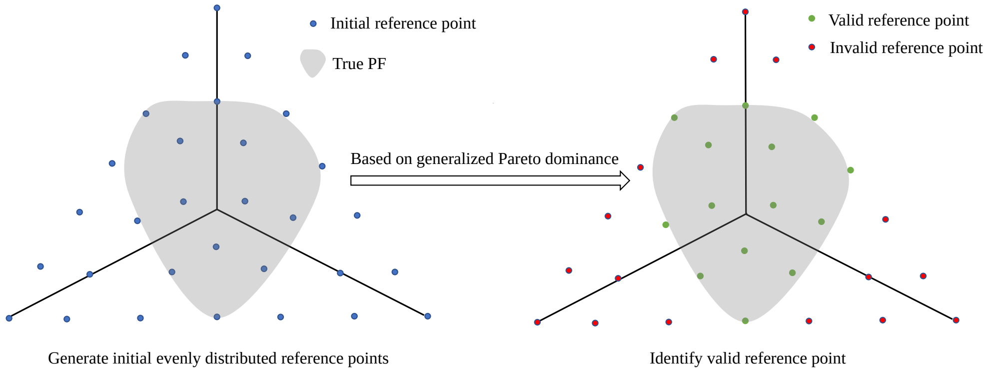


Figure 3. Valid reference vector identification.

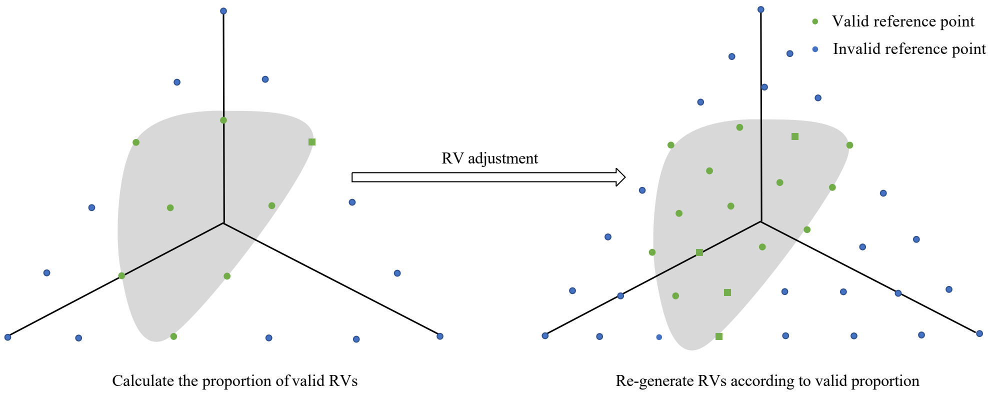


Figure 4. Reference vector adjustment.

4. Experimental Results and Analysis

In this section, we validate the performance of GPDARVC through a series of experiments. Firstly, we compare it with some state-of-the-art algorithms to show its superior performance. Then, ablation experiments are conducted to show the effectiveness of the proposed strategies and analyze the sensitivity of parameters. All experiments are performed on the PlatEMO platform 4.5 [54] using MATLAB R2023a on a personal computer, the manufacturer of which is DELL, made in China, with a built-in processor of Intel Core i5-12400F.

4.1. Experimental Design

To fully demonstrate the effectiveness and generalization of our algorithm, we test several representative MaOEAs on commonly used problems. Parameter settings and performance metrics are also introduced in this section.

4.1.1. Comparative Algorithms

We compare GPDARVC with some representative MaOEAs, including ANSGA-III [16], MaoEA-IGD [55], DEAGNG [32], LMPFE [56], TS-DGPD [57], RVEAiGNG [58], and Multi-GPO [21].These comparison algorithms cover different types of multi-objective optimization algorithms, ranging from Pareto dominance-based to metrics-based and decomposition-based algorithms.

- ANSGA-III [16] introduces an adaptive RV adjustment strategy to enhance the original NSGA-III. Specifically, new RVs are generated near the existing ones with more than two associations, while unassociated new RVs are removed.
- MaOEA-IGD [55] is an indicator-based approach that prioritizes solutions based on the IGD metric to guide the optimization process.
- DEAGNG [32] is a decomposition-based evolutionary algorithm that decomposes a multi-objective problem into several single-objective subproblems and guides the search process using neural networks and Gaussian process models.

- LMPFE [56] combines feedback mechanisms with Pareto optimization methods. By introducing a feedback evolutionary mechanism and Pareto dominance strategy, LMPFE effectively addresses the computational challenges of large-scale multi-objective optimization problems, offering an efficient, balanced, and diverse set of solutions.
- TS-DGPD [57] introduces a dynamic generalized Pareto dominance with two stages, where the first stage focuses on convergence and the second stage emphasizes solution diversity.
- RVEAiGNG [58] is an adaptive reference vector-based decomposition algorithm that presents a new approach to learning the distribution of reference vectors using a growing neural gas (GNG) network for automatic and stable adaptation.
- MultiGPO [21] utilizes M symmetric $(M-1)$-GPD scenarios, where each scenario enhances the selection pressure on $M-1$ objectives by expanding the dominance region of solutions while keeping the omitted objective constant. It demonstrates strong performance in handling unknown and irregular shapes of the PF.

4.1.2. Test Problems

In our study, we select three well-known test suites for experimental research, i.e., DTLZ, MaF, and WFG. The number of objectives varies from 5 to 15. For WFG test problems, the number of decision variables is set to $D = k + l$, where $k = M-1$ and $l = 10$. For MaF test problems, the setting of D is not very uniform, with $D = M + 9$ for MaF 1-MaF 6 and MaF 10; $D = M + 19$ for MaF 7; $D = 2$ for MaF 8 and MaF 9; and $D = 5$ for MaF 13. For DTLZ test problems, the number of decision variables is set to $D = M + 9$. These test MaOPs contain different problem characteristics, with regular and irregular FPs, such as convexity, linearity, degeneration, and disjointedness.

4.1.3. General Parameter Settings

GPDAEVC and other comparison algorithms use SBX as a crossover operator, with a crossover probability of 1.0 and a distribution exponent of 20, and PM as mutation operator, with an expected value of 1.0 and a distribution exponent of 20. For a fair comparison, population sizes for all algorithms are set to 210, 275, and 240 for 5-, 10-, and 15-objective test problems, respectively. The maximum number of fitness evaluations for each algorithm is set to $M \times 10{,}000$.

4.1.4. Performance Indicators

We employ inverse generation distance plus (IGD+) and hypervolume (HV) as the evaluation metrics for performance. IGD+ and HV are the most commonly used composite metrics in multi-objective optimization because they can reflect the convergence and diversity of the algorithms well. Generally, smaller IGD+ values indicate better results, while larger HVs indicate higher-quality solutions. To produce convincing results, all experiments are conducted 20 times, and the mean and standard deviation of results are recorded. The Wilcoxon rank sum test with a significance level of 0.05 is used for statistical analysis, where "+", "−", and "=" indicate that GPDARVC has worse, better, and similar performance compared to another algorithms. The formulas for HV and IGD+ are as follows.

$$HV(S, z^r) = \lambda\left(\bigcup_{i=1}^{|S|} v_i\right) \tag{9}$$

$$IGD^{+}(S, P^{*}) = \frac{1}{|P^{*}|}\sum_{p \in P^{*}} \min_{s \in S} d^{+}(p, s) \tag{10}$$

where λ is the Lebesgue measure, and v_i is the hypervolume consisting of the reference point $z^r = (1, 1, \ldots, 1)^{\mathrm{T}}$ and the solutions in the set of non-dominated solutions S. S is the solution set generated by the algorithm, P^* is the true PF's population, and $d^+(p, s)$

represents the directed distance from a reference point $p \in P^*$ to the closest solution $s \in S$, considering only the "dominated" distance.

We also adopt the performance score (PS), as suggested in [59], to compare the performance of all algorithms on all test suites. A smaller value of performance score indicates better performance. PS is defined as follows:

$$PS(MaOEA_i) = \frac{1}{P}\left(\sum_{j=1}^{P}\frac{1}{(N-1)}\left(\sum_{l=1}^{N}\delta_{j,l}^{i}\right)\right) \tag{11}$$

where N is the number of MaOEAs used for comparison, and P is the number of problems for testing. The parameter $\delta_{j,l}^{i}$ is defined as

$$\delta_{j,l}^{i} = \begin{cases} 1, & \text{if } MaOEA_j > MaOEA_i \text{ on a problem } l, \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

4.2. Experimental Results

In this section, Tables 1–6 give the statistical results obtained by each algorithm on different test suites, where the best results obtained by each algorithm are highlighted with dark background. Moreover, +, − and ≈ indicate that the result is significantly better, significantly worse and statistically similar to that obtained by GPDARVC, respectively. The experimental results show that our proposed GPDARVC achieves the best performance on most of the test suites. In the following, we analyze and discuss the results obtained by each algorithm in detail.

4.2.1. Comparison Results on DTLZ Test Problems

The mean and standard deviation of the IGD+ results and HV results obtained by all algorithms on the DTLZ test problems are given in Tables 1 and 2, respectively. The experimental results show that the overall performance of the proposed GPDARVC algorithm is significantly better than the other compared MaOEAs on these test suites. As shown in Table 1, GPDARVC obviously outperforms the compared algorithms on 14 out of 21 test problems. In HV metrics, GPDARVC also performs well, achieving the best performance on DTLZ1, DTLZ2, DTLZ3, DTLZ4, and DTLZ6. GPDARVC is better than ANSGA-III, MaOEA-IGD, DEAGNG, LMPFE, TS-DGPD, RVEAiGNG, and MultiGPO in 21 test problems on 19, 16, 18, 21, 21, 20, and 16 occasions and is defeated just 1, 4, 2, 0, 0, 1, and 0 times. This shows that GPDARVC is very effective in dealing with both simple and complex problems, which can be attributed to the fact that GPD provides enough selection pressure to keep fast convergence, while RVs maintain the diversity of the population through searching along different directions. In a word, the cooperation of GPD and RVs in the environmental selection part can ensure the outstanding performance of the final results even for complex problems.

To make a clear comparison, we illustrate the performance scores of all algorithms on different test suites in Figure 5. Figure 5a presents a bar chart of scores for GPDARVC and the compared algorithms on the DTLZ test suite. Clearly, the bars corresponding to GPDARVC have the lowest height in both HV and IGD+ metrics. This indicates that GPDARVC consistently ranks top, with performance scores significantly lower than the other MaOEAs. Therefore, we can conclude that the proposed GPDARVC performs exceptionally well on the DTLZ test suite. To ensure an intuitive understanding of GPDARVC, Figure 6 shows the final results of GPDARVC running against other competing algorithms on DTLZ5 with a 15-dimensional objective. It can be seen that GPDARVC not only approximates the final population to the true PF but also results in a very homogeneous population distribution. On the contrary, the experimental results of other competing algorithms either converge slowly or lack diversity.

Table 1. IGD+ values obtained by GPDARVC and other comparison algorithms on 5-, 10-, and 15-objective DTLZ 1-7. The best result for each test instance is shown with dark background.

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
DTLZ1	5	9	4.5940×10^{-2} (2.79×10^{-4}) −	1.6484×10^{-1} (2.26×10^{-1}) −	1.1869×10^{-1} (7.02×10^{-2}) −	3.8683×10^{-2} (3.65×10^{-3}) ≈	3.8479×10^{-2} (4.12×10^{-3}) −	4.1497×10^{-2} (4.26×10^{-4}) −	3.7186×10^{-2} (3.94×10^{-4}) ≈	3.7288×10^{-2} (3.53×10^{-4})
	10	14	1.0091×10^{-1} (4.95×10^{-2}) ≈	8.3402×10^{-2} (8.83×10^{-2}) −	1.6365×10^{-1} (2.21×10^{-2}) −	4.2190×10^{0} (2.41×10^{0}) −	8.1648×10^{-2} (2.14×10^{-2}) −	6.7808×10^{-2} (1.32×10^{-3}) +	7.3197×10^{-2} (1.64×10^{-3}) +	7.8383×10^{-2} (1.87×10^{-3})
	15	19	1.1110×10^{-1} (2.95×10^{-2}) −	1.3751×10^{-1} (1.16×10^{-1}) ≈	1.3841×10^{-1} (2.02×10^{-2}) −	5.8046×10^{0} (2.52×10^{0}) −	1.5316×10^{-1} (8.71×10^{-2}) −	9.0277×10^{-2} (4.85×10^{-3}) ≈	9.2294×10^{-2} (1.48×10^{-3}) −	8.9358×10^{-2} (3.19×10^{-3})
DTLZ2	5	14	7.2399×10^{-2} (1.90×10^{-3}) −	6.2558×10^{-2} (1.37×10^{-4}) +	9.1342×10^{-2} (3.53×10^{-3}) −	7.1808×10^{-2} (6.91×10^{-4}) −	7.5589×10^{-2} (1.67×10^{-3}) −	8.1319×10^{-2} (1.39×10^{-3}) −	7.2676×10^{-2} (9.48×10^{-4}) −	6.4962×10^{-2} (3.76×10^{-4})
	10	19	1.8723×10^{-1} (2.27×10^{-2}) −	1.7174×10^{-1} (2.20×10^{-3}) ≈	2.0901×10^{-1} (6.87×10^{-3}) −	1.7127×10^{-1} (1.81×10^{-3}) ≈	1.8453×10^{-1} (2.04×10^{-3}) −	1.7177×10^{-1} (1.98×10^{-3}) −	1.7985×10^{-1} (2.26×10^{-3}) −	1.7059×10^{-1} (4.57×10^{-4})
	15	24	2.7307×10^{-1} (1.14×10^{-2}) −	3.4632×10^{-1} (4.18×10^{-2}) −	2.7434×10^{-1} (1.29×10^{-2}) −	2.1462×10^{-1} (6.49×10^{-2}) +	2.3962×10^{-1} (3.42×10^{-3}) −	2.0609×10^{-1} (1.30×10^{-3}) +	2.3293×10^{-1} (3.69×10^{-3}) −	2.2582×10^{-1} (3.02×10^{-3})
DTLZ3	5	14	8.7895×10^{-2} (1.39×10^{-2}) −	9.3756×10^{0} (3.39×10^{0}) −	6.6271×10^{-1} (6.24×10^{-1}) −	8.7121×10^{-1} (1.27×10^{0}) −	8.2726×10^{-2} (7.27×10^{-3}) −	2.0207×10^{-1} (3.40×10^{-1}) −	7.5985×10^{-2} (4.31×10^{-3}) −	7.0722×10^{-2} (3.81×10^{-3})
	10	19	1.0563×10^{0} (1.46×10^{0}) −	5.7939×10^{0} (3.81×10^{0}) −	5.6195×10^{-1} (4.63×10^{-1}) −	1.7652×10^{2} (4.94×10^{1}) −	6.0260×10^{0} (2.83×10^{0}) −	1.9765×10^{-1} (1.14×10^{-2}) −	2.1157×10^{-1} (2.17×10^{-2}) −	1.7589×10^{-1} (3.25×10^{-3})
	15	24	1.0656×10^{0} (1.10×10^{0}) −	2.9275×10^{0} (1.44×10^{0}) −	1.2831×10^{0} (9.29×10^{-1}) −	2.2181×10^{2} (1.11×10^{2}) −	1.5036×10^{1} (5.70×10^{0}) −	5.0271×10^{-1} (1.24×10^{-1}) −	3.1552×10^{-1} (2.17×10^{-1}) −	2.3615×10^{-1} (4.27×10^{-3})
DTLZ4	5	14	8.2551×10^{-2} (3.36×10^{-2}) −	8.5652×10^{-2} (5.37×10^{-2}) −	8.3777×10^{-2} (1.88×10^{-3}) −	7.5100×10^{-2} (3.07×10^{-3}) −	7.5667×10^{-2} (1.43×10^{-3}) −	8.1083×10^{-2} (1.50×10^{-3}) −	7.1703×10^{-2} (9.50×10^{-4}) −	6.5065×10^{-2} (3.51×10^{-4})
	10	19	1.7301×10^{-1} (1.59×10^{-3}) −	1.6819×10^{-1} (2.34×10^{-3}) +	1.9234×10^{-1} (2.32×10^{-3}) −	3.4013×10^{-1} (1.19×10^{-1}) −	1.9482×10^{-1} (3.11×10^{-3}) −	1.6760×10^{-1} (5.38×10^{-3}) +	1.7873×10^{-1} (1.97×10^{-3}) −	1.7000×10^{-1} (4.38×10^{-4})
	15	24	2.4452×10^{-1} (1.69×10^{-2}) −	2.4560×10^{-1} (9.73×10^{-3}) −	2.3419×10^{-1} (1.85×10^{-3}) −	6.7045×10^{-1} (1.18×10^{-1}) −	2.4074×10^{-1} (3.77×10^{-3}) −	2.0871×10^{-1} (2.77×10^{-3}) +	2.2292×10^{-1} (1.84×10^{-3}) ≈	2.2433×10^{-1} (3.02×10^{-3})
DTLZ5	5	14	8.3713×10^{-2} (4.41×10^{-2}) −	1.9781×10^{-1} (1.34×10^{-1}) −	9.2131×10^{-2} (4.62×10^{-2}) −	1.1761×10^{-1} (4.57×10^{-2}) −	7.2629×10^{-2} (1.35×10^{-2}) −	7.8482×10^{-2} (2.96×10^{-2}) −	5.0725×10^{-2} (1.18×10^{-2}) −	4.2347×10^{-2} (8.12×10^{-3})
	10	19	2.5344×10^{-1} (7.96×10^{-2}) −	2.1530×10^{-1} (1.50×10^{-1}) −	1.7411×10^{-1} (4.27×10^{-2}) −	2.1220×10^{-1} (1.12×10^{-1}) −	2.0418×10^{-1} (9.42×10^{-2}) −	8.5474×10^{-2} (3.14×10^{-2}) ≈	1.0025×10^{-1} (1.87×10^{-2}) −	8.1812×10^{-2} (1.90×10^{-2})
	15	24	2.4511×10^{-1} (5.02×10^{-2}) −	2.7130×10^{-1} (1.38×10^{-1}) −	1.9716×10^{-1} (1.20×10^{-1}) −	1.7899×10^{-1} (1.14×10^{-1}) ≈	2.8725×10^{-1} (1.21×10^{-1}) −	1.5339×10^{-1} (6.18×10^{-2}) −	1.0616×10^{-1} (1.85×10^{-2}) ≈	1.0318×10^{-1} (2.35×10^{-2})
DTLZ6	5	14	1.4439×10^{-1} (7.92×10^{-2}) −	3.6647×10^{-1} (4.45×10^{-3}) −	2.0144×10^{-1} (9.35×10^{-2}) −	1.9246×10^{-1} (9.81×10^{-2}) −	8.5002×10^{-2} (2.09×10^{-2}) −	7.6623×10^{-2} (6.34×10^{-2}) −	6.7124×10^{-2} (1.76×10^{-2}) −	5.0886×10^{-2} (1.35×10^{-2})
	10	19	8.7498×10^{-1} (3.95×10^{-1}) −	3.7580×10^{-1} (4.29×10^{-4}) −	2.9028×10^{-1} (1.20×10^{-1}) −	4.8067×10^{-1} (2.88×10^{-1}) −	2.8921×10^{0} (6.01×10^{-1}) −	1.1994×10^{-1} (5.96×10^{-2}) −	1.0211×10^{-1} (2.30×10^{-2}) −	8.0732×10^{-2} (1.36×10^{-2})
	15	24	3.3181×10^{-1} (2.25×10^{-1}) −	3.6167×10^{-1} (6.54×10^{-2}) −	2.5802×10^{-1} (1.43×10^{-1}) −	4.4900×10^{-1} (2.58×10^{-1}) −	3.0743×10^{0} (8.00×10^{-1}) −	1.2655×10^{-1} (6.67×10^{-2}) −	8.9775×10^{-2} (1.62×10^{-2}) ≈	8.0777×10^{-2} (1.42×10^{-2})
DTLZ7	5	24	1.7507×10^{-1} (1.28×10^{-2}) −	3.5902×10^{-1} (6.20×10^{-2}) −	1.4210×10^{-1} (1.86×10^{-2}) −	7.2225×10^{-1} (4.21×10^{-1}) −	1.7335×10^{-1} (7.54×10^{-3}) −	1.3739×10^{-1} (3.03×10^{-2}) ≈	1.4859×10^{-1} (2.83×10^{-2}) −	1.2910×10^{-1} (3.47×10^{-3})
	10	29	7.3341×10^{-1} (6.21×10^{-2}) −	1.0440×10^{0} (5.25×10^{-2}) −	7.2080×10^{-1} (1.34×10^{-1}) ≈	2.8063×10^{0} (6.49×10^{-1}) −	1.0383×10^{0} (7.95×10^{-2}) −	6.0740×10^{-1} (1.18×10^{-2}) +	7.2405×10^{-1} (7.40×10^{-3}) −	6.7953×10^{-1} (8.44×10^{-3})
	15	34	4.1275×10^{0} (6.89×10^{-1}) −	1.4484×10^{0} (3.81×10^{-2}) +	6.2862×10^{0} (1.22×10^{0}) −	4.8606×10^{0} (2.50×10^{0}) −	4.1692×10^{0} (1.10×10^{0}) −	1.1196×10^{0} (6.47×10^{-2}) +	1.3657×10^{0} (4.60×10^{-2}) +	1.6508×10^{0} (2.65×10^{-1})
+/−/≈			0/20/1	3/16/2	0/20/1	1/17/3	0/21/0	6/12/3	2/15/4	

Table 2. HV values obtained by GPDARVC and other comparison algorithms on 5-, 10-, and 15-objective DTLZ 1-7. The best result for each test instance is shown with dark background.

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
DTLZ1	5	9	9.7467×10^{-1} (5.08×10^{-4}) −	6.9071×10^{-1} (3.72×10^{-1}) −	7.5756×10^{-1} (1.85×10^{-1}) −	9.7826×10^{-1} (2.34×10^{-3}) −	9.7401×10^{-1} (8.64×10^{-3}) −	9.7198×10^{-1} (1.03×10^{-3}) −	9.7613×10^{-1} (5.21×10^{-4}) −	9.7983×10^{-1} (5.83×10^{-4})
	10	14	9.6796×10^{-1} (6.81×10^{-2}) −	9.4663×10^{-1} (2.15×10^{-1}) −	7.9822×10^{-1} (7.72×10^{-2}) −	0.0000×10^{0} (0.00×10^{0}) −	9.8785×10^{-1} (4.45×10^{-2}) −	9.9889×10^{-1} (3.11×10^{-4}) −	9.9897×10^{-1} (6.31×10^{-4}) −	9.9973×10^{-1} (4.84×10^{-5})
	15	19	9.9638×10^{-1} (3.65×10^{-3}) −	8.0403×10^{-1} (2.74×10^{-1}) −	9.3265×10^{-1} (4.37×10^{-2}) −	0.0000×10^{0} (0.00×10^{0}) −	8.8154×10^{-1} (2.16×10^{-1}) −	9.9950×10^{-1} (1.93×10^{-4}) −	9.9977×10^{-1} (4.47×10^{-4}) ≈	9.9983×10^{-1} (1.58×10^{-4})
DTLZ2	5	14	7.9323×10^{-1} (2.86×10^{-3}) −	8.1165×10^{-1} (4.93×10^{-4}) +	7.3010×10^{-1} (6.83×10^{-3}) −	8.0043×10^{-1} (1.25×10^{-3}) −	7.9792×10^{-1} (2.30×10^{-3}) −	7.8661×10^{-1} (2.15×10^{-3}) −	8.0238×10^{-1} (1.56×10^{-3}) −	8.1113×10^{-1} (5.40×10^{-4})
	10	19	9.5697×10^{-1} (1.80×10^{-2}) −	9.6864×10^{-1} (1.31×10^{-3}) −	8.8748×10^{-1} (1.61×10^{-2}) −	9.6401×10^{-1} (6.48×10^{-4}) −	9.5638×10^{-1} (1.40×10^{-3}) −	9.6646×10^{-1} (6.49×10^{-4}) −	9.5936×10^{-1} (1.09×10^{-3}) −	9.7172×10^{-1} (1.84×10^{-4})
	15	24	9.7059×10^{-1} (8.21×10^{-3}) −	8.9483×10^{-1} (4.89×10^{-2}) −	9.5633×10^{-1} (1.60×10^{-2}) −	9.5715×10^{-1} (1.50×10^{-1}) −	9.7977×10^{-1} (1.01×10^{-3}) −	9.8529×10^{-1} (6.77×10^{-4}) −	9.8318×10^{-1} (1.06×10^{-3}) −	9.9344×10^{-1} (2.73×10^{-4})
DTLZ3	5	14	7.7094×10^{-1} (2.10×10^{-2}) −	0.0000×10^{0} (0.00×10^{0}) −	2.9351×10^{-1} (2.86×10^{-1}) −	2.7248×10^{-1} (2.66×10^{-1}) −	7.8876×10^{-1} (1.09×10^{-2}) −	6.7145×10^{-1} (1.98×10^{-1}) −	7.9903×10^{-1} (5.47×10^{-3}) −	8.0451×10^{-1} (4.55×10^{-3})
	10	19	5.0439×10^{-1} (3.65×10^{-1}) −	4.0281×10^{-3} (1.80×10^{-2}) −	5.2679×10^{-1} (2.72×10^{-1}) −	0.0000×10^{0} (0.00×10^{0}) −	1.8650×10^{-5} (8.34×10^{-5}) −	9.4986×10^{-1} (6.13×10^{-3}) −	9.3478×10^{-1} (2.28×10^{-2}) −	9.6892×10^{-1} (1.46×10^{-3})
	15	24	5.4395×10^{-1} (4.14×10^{-1}) −	1.6729×10^{-3} (5.73×10^{-3}) −	2.6632×10^{-1} (3.44×10^{-1}) −	0.0000×10^{0} (0.00×10^{0}) −	0.0000×10^{0} (0.00×10^{0}) −	6.0291×10^{-1} (2.38×10^{-1}) −	9.2194×10^{-1} (2.17×10^{-1}) −	9.9179×10^{-1} (1.37×10^{-3})
DTLZ4	5	14	7.7993×10^{-1} (4.53×10^{-2}) −	7.8450×10^{-1} (6.58×10^{-2}) −	7.6365×10^{-1} (6.95×10^{-3}) −	7.9756×10^{-1} (4.23×10^{-3}) −	7.9869×10^{-1} (2.22×10^{-3}) −	7.8782×10^{-1} (1.81×10^{-3}) −	8.0448×10^{-1} (1.66×10^{-3}) −	8.1079×10^{-1} (6.75×10^{-4})
	10	19	9.6968×10^{-1} (4.71×10^{-4}) −	9.7061×10^{-1} (2.27×10^{-3}) −	9.2911×10^{-1} (6.90×10^{-3}) −	6.9479×10^{-1} (2.81×10^{-1}) −	9.5244×10^{-1} (2.28×10^{-3}) −	9.7050×10^{-1} (2.76×10^{-3}) −	9.6239×10^{-1} (8.93×10^{-4}) −	9.7192×10^{-1} (2.17×10^{-4})
	15	24	9.8589×10^{-1} (6.32×10^{-3}) −	9.8759×10^{-1} (4.59×10^{-3}) −	9.8853×10^{-1} (7.04×10^{-4}) −	1.2436×10^{-1} (1.98×10^{-1}) −	9.8169×10^{-1} (1.17×10^{-3}) −	9.8978×10^{-1} (5.21×10^{-4}) −	9.8758×10^{-1} (3.91×10^{-4}) −	9.9367×10^{-1} (3.00×10^{-4})
DTLZ5	5	14	1.0344×10^{-1} (1.03×10^{-2}) ≈	9.7748×10^{-2} (1.25×10^{-4}) −	5.6660×10^{-2} (2.61×10^{-2}) −	4.0420×10^{-2} (3.69×10^{-2}) −	9.6143×10^{-2} (4.44×10^{-3}) −	4.9892×10^{-2} (2.86×10^{-2}) −	1.0154×10^{-1} (4.95×10^{-3}) −	1.0491×10^{-1} (4.70×10^{-3})
	10	19	7.7316×10^{-2} (8.11×10^{-3}) −	9.1297×10^{-2} (3.96×10^{-4}) +	5.5381×10^{-2} (3.00×10^{-2}) −	3.0640×10^{-2} (3.85×10^{-2}) −	4.7509×10^{-2} (2.89×10^{-2}) −	3.8559×10^{-2} (3.49×10^{-2}) −	8.6196×10^{-2} (3.13×10^{-3}) −	8.9374×10^{-2} (8.76×10^{-4})
	15	24	7.5368×10^{-2} (8.11×10^{-3}) −	9.0665×10^{-2} (2.94×10^{-4}) +	6.5311×10^{-2} (2.56×10^{-2}) −	4.7680×10^{-2} (4.09×10^{-2}) −	2.9159×10^{-2} (2.97×10^{-2}) −	1.0789×10^{-2} (2.56×10^{-2}) −	8.8644×10^{-2} (1.95×10^{-3}) ≈	8.9241×10^{-2} (1.18×10^{-3})

Table 2. *Cont.*

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
DTLZ6	5	14	$9.1198 \times 10^{-2} (7.20 \times 10^{-4})$ −	$8.7792 \times 10^{-2} (3.00 \times 10^{-2})$ ≈	$2.4239 \times 10^{-2} (3.00 \times 10^{-2})$ −	$2.8008 \times 10^{-2} (3.94 \times 10^{-2})$ −	$9.2434 \times 10^{-2} (1.83 \times 10^{-3})$ −	$8.3305 \times 10^{-2} (2.43 \times 10^{-2})$ −	$9.7489 \times 10^{-2} (6.38 \times 10^{-3})$ ≈	**$1.0093 \times 10^{-1} (6.61 \times 10^{-3})$**
	10	19	$4.5462 \times 10^{-3} (2.03 \times 10^{-2})$ −	$8.7055 \times 10^{-2} (2.05 \times 10^{-2})$ −	$9.4325 \times 10^{-3} (2.79 \times 10^{-2})$ −	$9.0904 \times 10^{-3} (2.80 \times 10^{-2})$ −	$0.0000 \times 10^{0} (0.00 \times 10^{0})$ −	$6.8785 \times 10^{-2} (3.34 \times 10^{-2})$ −	$8.9286 \times 10^{-2} (7.10 \times 10^{-3})$ ≈	**$9.0887 \times 10^{-2} (1.94 \times 10^{-4})$**
	15	24	$6.4410 \times 10^{-2} (3.89 \times 10^{-2})$ −	**$9.1216 \times 10^{-2} (1.50 \times 10^{-4})$ +**	$3.5501 \times 10^{-2} (4.37 \times 10^{-2})$ −	$1.6996 \times 10^{-2} (3.52 \times 10^{-2})$ −	$0.0000 \times 10^{0} (0.00 \times 10^{0})$ −	$8.3514 \times 10^{-2} (1.21 \times 10^{-2})$ −	$9.0959 \times 10^{-2} (2.43 \times 10^{-4})$ ≈	$9.0971 \times 10^{-2} (2.57 \times 10^{-4})$
DTLZ7	5	24	$2.4029 \times 10^{-1} (3.73 \times 10^{-3})$ −	$1.3352 \times 10^{-1} (3.74 \times 10^{-2})$ −	$2.5958 \times 10^{-1} (4.31 \times 10^{-3})$ ≈	$2.2551 \times 10^{-1} (1.10 \times 10^{-2})$ −	$2.3092 \times 10^{-1} (4.96 \times 10^{-3})$ −	$2.5195 \times 10^{-1} (2.68 \times 10^{-3})$ −	$2.5318 \times 10^{-1} (3.91 \times 10^{-3})$ −	**$2.6175 \times 10^{-1} (2.61 \times 10^{-3})$**
	10	29	$1.7112 \times 10^{-1} (6.30 \times 10^{-3})$ +	$2.2534 \times 10^{-3} (1.63 \times 10^{-3})$ −	**$1.8778 \times 10^{-1} (9.57 \times 10^{-3})$ +**	$1.1505 \times 10^{-1} (1.86 \times 10^{-2})$ −	$6.0844 \times 10^{-2} (1.57 \times 10^{-2})$ −	$1.5507 \times 10^{-1} (8.85 \times 10^{-3})$ +	$1.1874 \times 10^{-1} (2.20 \times 10^{-2})$ −	$1.3533 \times 10^{-1} (1.22 \times 10^{-2})$
	15	34	$6.5540 \times 10^{-2} (1.87 \times 10^{-2})$ −	$5.7682 \times 10^{-5} (1.02 \times 10^{-4})$ −	**$1.4783 \times 10^{-1} (7.86 \times 10^{-3})$ +**	$5.4693 \times 10^{-3} (8.87 \times 10^{-3})$ −	$2.5400 \times 10^{-2} (1.98 \times 10^{-2})$ −	$6.6297 \times 10^{-2} (1.95 \times 10^{-2})$ −	$9.4165 \times 10^{-2} (2.81 \times 10^{-2})$ −	$1.1945 \times 10^{-1} (1.29 \times 10^{-2})$
+/−/≈			1/19/1	4/16/1	2/18/1	0/21/0	0/21/0	1/20/0	0/16/5	

Table 3. IGD+ values obtained by GPDARVC and other comparison algorithms on 5-, 10-, and 15-objective MaF 1-10. The best result for each test instance is shown with dark background.

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
MaF1	5	14	$1.6606 \times 10^{-1} (5.20 \times 10^{-3})$ −	$2.3569 \times 10^{-1} (1.22 \times 10^{-3})$ −	**$7.5183 \times 10^{-2} (5.38 \times 10^{-3})$ +**	$7.9743 \times 10^{-2} (1.13 \times 10^{-3})$ +	$8.1819 \times 10^{-2} (1.51 \times 10^{-3})$ +	$7.5788 \times 10^{-2} (4.63 \times 10^{-4})$ +	$7.9782 \times 10^{-2} (1.83 \times 10^{-3})$ +	$8.3015 \times 10^{-2} (1.78 \times 10^{-3})$
	10	19	$2.2099 \times 10^{-1} (5.30 \times 10^{-3})$ −	$2.8747 \times 10^{-1} (3.60 \times 10^{-3})$ −	$1.6670 \times 10^{-1} (3.45 \times 10^{-3})$ −	$1.6617 \times 10^{-1} (3.32 \times 10^{-3})$ ≈	$1.6531 \times 10^{-1} (9.13 \times 10^{-4})$ −	$2.0729 \times 10^{-1} (2.73 \times 10^{-2})$ −	$1.6563 \times 10^{-1} (1.21 \times 10^{-3})$ −	**$1.6473 \times 10^{-1} (9.92 \times 10^{-4})$**
	15	24	$2.6978 \times 10^{-1} (5.98 \times 10^{-3})$ −	$3.3135 \times 10^{-1} (8.48 \times 10^{-3})$ −	$2.1353 \times 10^{-1} (6.58 \times 10^{-3})$ −	$2.1314 \times 10^{-1} (6.21 \times 10^{-3})$ −	$2.0234 \times 10^{-1} (2.03 \times 10^{-3})$ −	$3.3587 \times 10^{-1} (3.98 \times 10^{-2})$ −	$1.9735 \times 10^{-1} (1.55 \times 10^{-3})$ ≈	**$1.9732 \times 10^{-1} (1.36 \times 10^{-3})$**
MaF2	5	14	$7.0936 \times 10^{-2} (1.70 \times 10^{-3})$ −	$1.4727 \times 10^{-1} (7.92 \times 10^{-2})$ −	**$4.5175 \times 10^{-2} (1.22 \times 10^{-3})$ +**	$4.9204 \times 10^{-2} (1.35 \times 10^{-3})$ +	$6.1556 \times 10^{-2} (1.84 \times 10^{-3})$ −	$4.8672 \times 10^{-2} (1.20 \times 10^{-3})$ +	$5.5021 \times 10^{-2} (9.87 \times 10^{-4})$ −	$5.1897 \times 10^{-2} (5.97 \times 10^{-4})$
	10	19	$1.3271 \times 10^{-1} (1.17 \times 10^{-2})$ −	$2.1721 \times 10^{-1} (2.01 \times 10^{-2})$ −	$1.4806 \times 10^{-1} (1.43 \times 10^{-2})$ −	$1.0021 \times 10^{-1} (2.57 \times 10^{-3})$ +	$1.1581 \times 10^{-1} (4.41 \times 10^{-3})$ ≈	**$9.8642 \times 10^{-2} (2.21 \times 10^{-3})$ +**	$1.1747 \times 10^{-1} (4.81 \times 10^{-3})$ ≈	$1.1473 \times 10^{-1} (5.72 \times 10^{-3})$
	15	24	$1.4794 \times 10^{-1} (1.32 \times 10^{-2})$ −	$2.5160 \times 10^{-1} (8.31 \times 10^{-3})$ −	$1.6271 \times 10^{-1} (9.17 \times 10^{-3})$ −	**$1.0379 \times 10^{-1} (2.19 \times 10^{-3})$ +**	$1.2695 \times 10^{-1} (6.55 \times 10^{-3})$ +	$1.1942 \times 10^{-1} (4.77 \times 10^{-3})$ +	$1.3660 \times 10^{-1} (8.25 \times 10^{-3})$ ≈	$1.3594 \times 10^{-1} (8.00 \times 10^{-3})$
MaF3	5	14	$5.6303 \times 10^{-2} (1.88 \times 10^{-2})$ −	$9.4828 \times 10^{0} (1.55 \times 10^{1})$ −	$4.2387 \times 10^{0} (7.46 \times 10^{0})$ −	$4.9867 \times 10^{0} (6.61 \times 10^{0})$ −	$4.3086 \times 10^{-2} (1.29 \times 10^{-2})$ −	$1.0050 \times 10^{0} (4.06 \times 10^{0})$ −	$3.7722 \times 10^{-2} (1.24 \times 10^{-2})$ −	**$2.5824 \times 10^{-2} (1.93 \times 10^{-3})$**
	10	19	$1.7344 \times 10^{2} (2.53 \times 10^{2})$ −	$2.9170 \times 10^{2} (1.30 \times 10^{3})$ −	$1.2394 \times 10^{2} (3.30 \times 10^{2})$ −	$1.7889 \times 10^{6} (1.20 \times 10^{6})$ −	$9.4391 \times 10^{4} (1.02 \times 10^{5})$ −	$2.8265 \times 10^{-2} (5.13 \times 10^{-3})$ −	$2.8817 \times 10^{-2} (4.44 \times 10^{-3})$ −	**$2.2447 \times 10^{-2} (3.56 \times 10^{-18})$**
	15	24	$9.8689 \times 10^{1} (2.04 \times 10^{2})$ −	$8.9986 \times 10^{-1} (9.92 \times 10^{-1})$ −	$2.7306 \times 10^{2} (3.16 \times 10^{2})$ −	$5.5053 \times 10^{5} (5.42 \times 10^{5})$ −	$1.0747 \times 10^{6} (1.86 \times 10^{6})$ −	$1.8247 \times 10^{-2} (9.23 \times 10^{-3})$ ≈	**$1.8025 \times 10^{-2} (1.23 \times 10^{-3})$ ≈**	$1.8913 \times 10^{-2} (1.62 \times 10^{-3})$
MaF4	5	14	$1.4303 \times 10^{0} (4.48 \times 10^{-1})$ −	$8.1059 \times 10^{0} (5.98 \times 10^{0})$ −	$2.0018 \times 10^{0} (2.50 \times 10^{0})$ −	$1.2915 \times 10^{0} (2.00 \times 10^{0})$ −	$7.5510 \times 10^{-1} (6.43 \times 10^{-2})$ −	$2.5136 \times 10^{0} (3.53 \times 10^{0})$ −	$6.1668 \times 10^{-1} (3.32 \times 10^{-2})$ ≈	**$5.9851 \times 10^{-1} (3.03 \times 10^{-2})$**
	10	19	$5.3163 \times 10^{1} (9.76 \times 10^{0})$ −	$1.2294 \times 10^{2} (1.48 \times 10^{2})$ −	$2.6666 \times 10^{1} (1.05 \times 10^{1})$ −	$7.5703 \times 10^{1} (1.59 \times 10^{2})$ −	$1.9120 \times 10^{1} (2.97 \times 10^{0})$ −	$1.0889 \times 10^{1} (1.67 \times 10^{0})$ −	**$9.0576 \times 10^{0} (4.20 \times 10^{-1})$ +**	$9.4958 \times 10^{0} (1.82 \times 10^{-15})$
	15	24	$1.8827 \times 10^{3} (3.88 \times 10^{2})$ −	$1.4482 \times 10^{3} (8.58 \times 10^{2})$ −	$8.0599 \times 10^{2} (4.86 \times 10^{2})$ −	$2.1618 \times 10^{4} (4.35 \times 10^{4})$ −	$6.7287 \times 10^{2} (1.03 \times 10^{2})$ −	$3.2945 \times 10^{2} (7.85 \times 10^{1})$ −	**$1.7705 \times 10^{2} (8.70 \times 10^{0})$ ≈**	$1.7923 \times 10^{2} (1.04 \times 10^{1})$
MaF5	5	14	$4.5204 \times 10^{-1} (1.94 \times 10^{-1})$ +	$6.0440 \times 10^{-1} (1.07 \times 10^{-1})$ −	$5.2332 \times 10^{-1} (2.83 \times 10^{-2})$ −	**$4.0849 \times 10^{-1} (1.78 \times 10^{-2})$ +**	$4.1832 \times 10^{-1} (9.69 \times 10^{-3})$ +	$4.1927 \times 10^{-1} (5.88 \times 10^{-3})$ +	$4.2646 \times 10^{-1} (1.49 \times 10^{-2})$ +	$4.5367 \times 10^{-1} (3.03 \times 10^{-2})$
	10	19	**$1.0751 \times 10^{0} (2.69 \times 10^{-2})$ +**	$2.0641 \times 10^{1} (8.65 \times 10^{1})$ ≈	$1.7260 \times 10^{0} (4.22 \times 10^{-1})$ −	$2.3732 \times 10^{0} (7.81 \times 10^{-1})$ −	$1.3226 \times 10^{0} (2.82 \times 10^{-2})$ −	$2.0394 \times 10^{0} (6.50 \times 10^{-1})$ −	$1.2289 \times 10^{0} (8.87 \times 10^{-3})$ −	$1.2149 \times 10^{0} (2.28 \times 10^{-16})$
	15	24	**$1.3994 \times 10^{0} (2.82 \times 10^{-2})$ +**	$1.5208 \times 10^{0} (5.31 \times 10^{-2})$ ≈	$1.5103 \times 10^{0} (3.08 \times 10^{-2})$ ≈	$2.8187 \times 10^{0} (8.29 \times 10^{-1})$ −	$1.6232 \times 10^{0} (9.39 \times 10^{-3})$ −	$2.7160 \times 10^{0} (8.62 \times 10^{-1})$ −	$1.4946 \times 10^{0} (7.76 \times 10^{-3})$ ≈	$1.4939 \times 10^{0} (1.17 \times 10^{-2})$
MaF6	5	14	$2.2371 \times 10^{-2} (3.09 \times 10^{-3})$ −	$3.1356 \times 10^{-1} (1.21 \times 10^{-1})$ −	**$8.7643 \times 10^{-4} (1.94 \times 10^{-5})$ +**	$1.1064 \times 10^{-3} (4.61 \times 10^{-4})$ +	$2.7616 \times 10^{-3} (6.61 \times 10^{-4})$ −	$9.0313 \times 10^{-4} (1.67 \times 10^{-5})$ +	$1.1450 \times 10^{-3} (7.02 \times 10^{-5})$ ≈	$1.1069 \times 10^{-3} (5.97 \times 10^{-5})$
	10	19	$4.7019 \times 10^{-1} (1.87 \times 10^{-1})$ −	$2.8899 \times 10^{-1} (1.38 \times 10^{-1})$ −	$1.5260 \times 10^{0} (7.86 \times 10^{-1})$ −	$8.7070 \times 10^{-1} (5.09 \times 10^{-1})$ −	$1.0535 \times 10^{-1} (1.18 \times 10^{-1})$ ≈	**$6.6778 \times 10^{-4} (1.06 \times 10^{-5})$ +**	$8.5688 \times 10^{-2} (1.06 \times 10^{-1})$ ≈	$1.3317 \times 10^{-1} (3.24 \times 10^{-2})$
	15	24	$6.0390 \times 10^{-1} (4.03 \times 10^{-1})$ −	$3.2384 \times 10^{-1} (1.12 \times 10^{-1})$ −	$2.9242 \times 10^{0} (5.37 \times 10^{0})$ −	$8.1644 \times 10^{-1} (2.68 \times 10^{-1})$ −	$3.1620 \times 10^{-1} (1.70 \times 10^{-1})$ −	$8.7765 \times 10^{-2} (1.80 \times 10^{-1})$ −	$1.7838 \times 10^{-1} (9.80 \times 10^{-2})$ −	**$8.2901 \times 10^{-2} (4.18 \times 10^{-2})$**
MaF7	5	24	$1.7114 \times 10^{-1} (6.57 \times 10^{-3})$ −	$3.5034 \times 10^{-1} (6.11 \times 10^{-2})$ −	$1.5084 \times 10^{-1} (3.11 \times 10^{-2})$ −	$7.1766 \times 10^{-1} (4.16 \times 10^{-1})$ −	$2.2878 \times 10^{-1} (2.46 \times 10^{-1})$ −	$1.7890 \times 10^{-1} (2.08 \times 10^{-3})$ −	$1.4972 \times 10^{-1} (3.28 \times 10^{-2})$ −	**$1.3091 \times 10^{-1} (3.40 \times 10^{-3})$**
	10	29	$7.6578 \times 10^{-1} (8.08 \times 10^{-2})$ −	$9.4335 \times 10^{-1} (5.43 \times 10^{-2})$ −	$7.1346 \times 10^{-1} (1.06 \times 10^{-1})$ ≈	$3.1147 \times 10^{0} (4.29 \times 10^{-1})$ −	$1.0215 \times 10^{0} (5.72 \times 10^{-2})$ −	$7.0206 \times 10^{-1} (1.09 \times 10^{-2})$ ≈	$7.2159 \times 10^{-1} (1.30 \times 10^{-2})$ −	**$6.7698 \times 10^{-1} (2.28 \times 10^{-16})$**
	15	34	$3.8166 \times 10^{0} (4.97 \times 10^{-1})$ −	$1.4071 \times 10^{0} (3.05 \times 10^{-2})$ +	$6.1926 \times 10^{0} (1.23 \times 10^{0})$ −	$3.9574 \times 10^{0} (1.94 \times 10^{0})$ −	$4.6812 \times 10^{0} (1.16 \times 10^{0})$ −	**$1.0804 \times 10^{0} (4.88 \times 10^{-2})$ +**	$1.3687 \times 10^{0} (4.51 \times 10^{-2})$ +	$1.6345 \times 10^{0} (2.05 \times 10^{-1})$
MaF8	5	2	$1.1688 \times 10^{-1} (8.20 \times 10^{-3})$ −	$4.7808 \times 10^{-1} (1.24 \times 10^{-1})$ −	$6.9349 \times 10^{-2} (1.32 \times 10^{-2})$ −	$5.2867 \times 10^{-2} (1.18 \times 10^{-2})$ ≈	$4.9541 \times 10^{-2} (1.15 \times 10^{-3})$ −	$4.6795 \times 10^{-2} (2.49 \times 10^{-3})$ ≈	$4.6910 \times 10^{-2} (7.83 \times 10^{-4})$ ≈	**$4.6730 \times 10^{-2} (6.46 \times 10^{-4})$**
	10	2	$1.8739 \times 10^{-1} (2.32 \times 10^{-2})$ −	$7.5160 \times 10^{-1} (9.32 \times 10^{-2})$ −	$7.9392 \times 10^{-2} (1.43 \times 10^{-2})$ −	**$5.9367 \times 10^{-2} (2.97 \times 10^{-3})$ +**	$5.9573 \times 10^{-2} (5.65 \times 10^{-4})$ +	$5.9981 \times 10^{-2} (3.12 \times 10^{-4})$ +	$6.2770 \times 10^{-2} (8.61 \times 10^{-4})$ ≈	$6.2620 \times 10^{-2} (0.00 \times 10^{0})$
	15	2	$2.0426 \times 10^{-1} (3.44 \times 10^{-2})$ −	$1.0367 \times 10^{0} (1.21 \times 10^{-1})$ −	$1.0852 \times 10^{-1} (1.48 \times 10^{-2})$ −	$1.6307 \times 10^{-1} (1.16 \times 10^{-1})$ −	$1.2445 \times 10^{-1} (2.08 \times 10^{-3})$ −	**$8.0664 \times 10^{-2} (6.19 \times 10^{-4})$ +**	$9.7711 \times 10^{-2} (3.59 \times 10^{-3})$ −	$9.3734 \times 10^{-2} (3.91 \times 10^{-3})$
MaF9	5	2	$4.0719 \times 10^{-1} (1.34 \times 10^{-1})$ −	$1.0726 \times 10^{0} (1.52 \times 10^{0})$ −	$2.0417 \times 10^{-1} (9.78 \times 10^{-2})$ −	$2.0377 \times 10^{-1} (4.43 \times 10^{-2})$ −	$5.3614 \times 10^{-2} (7.26 \times 10^{-4})$ −	$5.2639 \times 10^{-2} (2.06 \times 10^{-3})$ ≈	**$5.1864 \times 10^{-2} (4.82 \times 10^{-4})$ +**	$5.2483 \times 10^{-2} (6.17 \times 10^{-4})$
	10	2	$3.5583 \times 10^{-1} (5.34 \times 10^{-2})$ −	$9.9237 \times 10^{-1} (7.95 \times 10^{-1})$ −	$3.7940 \times 10^{-1} (1.28 \times 10^{-1})$ −	$7.4453 \times 10^{-1} (1.38 \times 10^{-1})$ −	$9.0437 \times 10^{-2} (4.11 \times 10^{-3})$ −	$1.4580 \times 10^{-1} (4.37 \times 10^{-2})$ −	**$7.2169 \times 10^{-2} (3.68 \times 10^{-4})$ ≈**	$7.2283 \times 10^{-2} (4.99 \times 10^{-4})$
	15	2	$8.2976 \times 10^{-1} (2.55 \times 10^{0})$ −	$4.4101 \times 10^{0} (5.08 \times 10^{0})$ −	$2.5146 \times 10^{-1} (6.24 \times 10^{-2})$ −	$5.3184 \times 10^{-1} (2.69 \times 10^{-1})$ −	$6.6236 \times 10^{-1} (2.56 \times 10^{0})$ ≈	$9.3655 \times 10^{-2} (1.34 \times 10^{-3})$ −	**$9.0100 \times 10^{-2} (4.79 \times 10^{-4})$ ≈**	$9.0164 \times 10^{-2} (5.73 \times 10^{-4})$
MaF10	5	14	$1.6834 \times 10^{-1} (7.92 \times 10^{-3})$ −	$5.5384 \times 10^{-1} (1.67 \times 10^{-1})$ −	$2.9452 \times 10^{-1} (5.29 \times 10^{-2})$ −	$3.6327 \times 10^{-1} (5.15 \times 10^{-2})$ −	$1.6536 \times 10^{-1} (2.37 \times 10^{-2})$ −	$3.8897 \times 10^{-1} (8.26 \times 10^{-2})$ −	$1.2150 \times 10^{-1} (1.30 \times 10^{-2})$ −	**$1.0616 \times 10^{-1} (1.75 \times 10^{-3})$**
	10	19	$2.8409 \times 10^{-1} (6.00 \times 10^{-2})$ −	$6.4970 \times 10^{0} (4.06 \times 10^{0})$ −	$4.2634 \times 10^{-1} (1.01 \times 10^{-1})$ −	$3.5981 \times 10^{-1} (7.23 \times 10^{-2})$ −	$4.1446 \times 10^{-1} (6.29 \times 10^{-2})$ −	$4.9240 \times 10^{-1} (9.84 \times 10^{-2})$ −	$2.2751 \times 10^{-1} (7.09 \times 10^{-2})$ −	**$1.7741 \times 10^{-1} (7.35 \times 10^{-3})$**
	15	24	$3.5555 \times 10^{-1} (4.40 \times 10^{-2})$ −	$4.9849 \times 10^{0} (3.49 \times 10^{0})$ −	$5.8477 \times 10^{-1} (1.16 \times 10^{-1})$ −	$3.8232 \times 10^{-1} (9.02 \times 10^{-2})$ −	$4.9702 \times 10^{-1} (1.05 \times 10^{-1})$ −	$4.7740 \times 10^{-1} (8.53 \times 10^{-2})$ −	$2.7024 \times 10^{-1} (6.48 \times 10^{-2})$ −	**$2.0657 \times 10^{-1} (1.59 \times 10^{-2})$**
+/−/≈			3/27/0	1/27/2	3/25/2	7/21/2	4/23/3	10/16/4	5/12/13	

Table 4. HV values obtained by GPDARVC and other comparison algorithms on 5-, 10-, and 15-objective MaF 1-10. The best result for each test instance is shown with dark background.

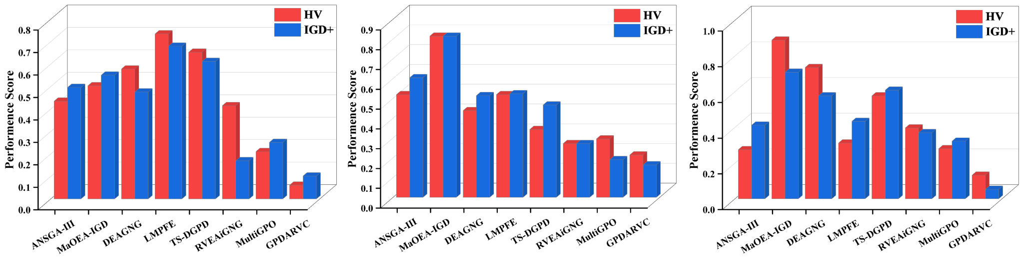
Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
MaF1	5	14	4.6121×10^{-3} (2.62×10^{-4}) −	3.8590×10^{-3} (5.59×10^{-5}) −	**1.2639×10^{-2} (5.83×10^{-4}) +**	1.1759×10^{-2} (1.79×10^{-4}) +	1.1479×10^{-2} (2.76×10^{-4}) +	1.2406×10^{-2} (1.32×10^{-4}) +	1.1759×10^{-2} (2.58×10^{-4}) +	1.1310×10^{-2} (2.42×10^{-4})
	10	19	4.6405×10^{-7} (2.30×10^{-8}) ≈	1.6880×10^{-7} (1.45×10^{-8}) ≈	**5.4529×10^{-7} (3.20×10^{-7}) ≈**	3.6437×10^{-7} (4.81×10^{-7}) ≈	4.9805×10^{-7} (6.87×10^{-7}) ≈	3.7271×10^{-7} (1.76×10^{-7}) ≈	3.9699×10^{-7} (7.47×10^{-7}) ≈	5.3666×10^{-7} (8.60×10^{-7})
	15	24	5.5603×10^{-12} (7.49×10^{-13}) +	1.3253×10^{-12} (4.46×10^{-13}) +	**7.2340×10^{-12} (3.24×10^{-11}) ≈**	0.0000×10^{0} (0.00×10^{0}) ≈	0.0000×10^{0} (0.00×10^{0}) ≈	$6.6561 \times 10^{-1}4$ (1.77×10^{-13}) +	0.0000×10^{0} (0.00×10^{0}) ≈	0.0000×10^{0} (0.00×10^{0})
MaF2	5	14	1.7072×10^{-1} (3.45×10^{-3}) −	1.1656×10^{-1} (4.02×10^{-2}) −	1.8755×10^{-1} (3.87×10^{-3}) −	1.8917×10^{-1} (3.82×10^{-3}) −	1.8086×10^{-1} (3.93×10^{-3}) −	**2.0240×10^{-1} (1.09×10^{-3}) +**	1.8752×10^{-1} (2.21×10^{-3}) −	1.9223×10^{-1} (2.51×10^{-3})
	10	19	**2.2275×10^{-1} (5.87×10^{-3}) +**	1.7563×10^{-1} (4.31×10^{-3}) −	1.9355×10^{-1} (6.66×10^{-3}) −	1.7354×10^{-1} (9.24×10^{-3}) −	2.0830×10^{-1} (5.63×10^{-3}) −	2.2038×10^{-1} (2.17×10^{-3}) ≈	2.0829×10^{-1} (3.53×10^{-3}) −	2.2006×10^{-1} (3.13×10^{-3})
	15	24	1.8107×10^{-1} (7.09×10^{-3}) −	1.4632×10^{-1} (1.16×10^{-2}) −	1.7262×10^{-1} (1.24×10^{-2}) −	1.2074×10^{-1} (1.72×10^{-2}) −	1.7189×10^{-1} (4.30×10^{-3}) −	2.1004×10^{-1} (3.40×10^{-3}) −	1.8708×10^{-1} (4.78×10^{-3}) −	**2.1259×10^{-1} (2.18×10^{-3})**
MaF3	5	14	9.9755×10^{-1} (2.26×10^{-3}) ≈	8.7352×10^{-2} (2.22×10^{-1}) −	3.9200×10^{-1} (4.68×10^{-1}) −	3.3797×10^{-1} (4.66×10^{-1}) −	9.9681×10^{-1} (2.65×10^{-3}) ≈	8.9967×10^{-1} (2.74×10^{-1}) ≈	9.9661×10^{-1} (2.73×10^{-3}) ≈	**9.9758×10^{-1} (7.20×10^{-4})**
	10	19	2.1740×10^{-1} (3.63×10^{-1}) −	2.4378×10^{-1} (3.77×10^{-1}) −	8.8125×10^{-2} (2.74×10^{-1}) −	0.0000×10^{0} (0.00×10^{0}) −	0.0000×10^{0} (0.00×10^{0}) −	9.9903×10^{-1} (1.33×10^{-4}) −	9.9902×10^{-1} (7.40×10^{-4}) −	**9.9962×10^{-1} (1.43×10^{-5})**
	15	24	2.2981×10^{-1} (3.91×10^{-1}) −	4.5466×10^{-1} (3.70×10^{-1}) −	8.9254×10^{-2} (2.75×10^{-1}) −	0.0000×10^{0} (0.00×10^{0}) −	0.0000×10^{0} (0.00×10^{0}) −	9.9969×10^{-1} (1.73×10^{-4}) −	9.9959×10^{-1} (3.20×10^{-4}) −	**9.9994×10^{-1} (6.73×10^{-5})**
MaF4	5	14	6.5321×10^{-2} (1.38×10^{-2}) −	4.8524×10^{-3} (1.04×10^{-2}) −	7.9295×10^{-2} (3.59×10^{-2}) −	8.5734×10^{-2} (2.41×10^{-2}) −	1.0895×10^{-1} (5.42×10^{-3}) −	7.7391×10^{-2} (4.09×10^{-2}) −	1.1445×10^{-1} (3.81×10^{-3}) −	**1.1694×10^{-1} (2.68×10^{-3})**
	10	19	2.5763×10^{-4} (1.37×10^{-5}) +	2.0149×10^{-7} (3.88×10^{-7}) −	2.3250×10^{-4} (7.69×10^{-5}) +	4.5033×10^{-5} (6.23×10^{-5}) −	4.9607×10^{-5} (1.31×10^{-5}) ≈	**2.7757×10^{-4} (5.74×10^{-5}) +**	5.9247×10^{-5} (1.40×10^{-5}) ≈	5.4507×10^{-5} (3.97×10^{-6})
	15	24	**2.0310×10^{-7} (1.74×10^{-8}) +**	8.1605×10^{-13} (2.18×10^{-12}) −	5.6619×10^{-8} (4.06×10^{-8}) +	1.1446×10^{-16} (3.57×10^{-16}) ≈	1.0256×10^{-9} (4.59×10^{-9}) ≈	1.4041×10^{-7} (7.65×10^{-8}) +	2.9827×10^{-9} (1.33×10^{-8}) ≈	9.3125×10^{-9} (2.30×10^{-8})
MaF5	5	14	7.8035×10^{-1} (4.54×10^{-2}) +	6.0920×10^{-1} (5.02×10^{-2}) −	7.6565×10^{-1} (5.20×10^{-3}) ≈	7.9639×10^{-1} (5.35×10^{-3}) +	7.9666×10^{-1} (2.83×10^{-3}) +	**8.0214×10^{-1} (1.15×10^{-3}) +**	7.7892×10^{-1} (5.41×10^{-3}) +	7.6540×10^{-1} (1.19×10^{-2})
	10	19	**9.6884×10^{-1} (8.73×10^{-4}) +**	5.8180×10^{-1} (1.15×10^{-1}) −	9.3248×10^{-1} (6.51×10^{-3}) +	4.5637×10^{-1} (3.35×10^{-1}) −	9.5282×10^{-1} (2.27×10^{-3}) +	9.4684×10^{-1} (4.90×10^{-3}) +	8.3378×10^{-1} (4.24×10^{-3}) −	8.3647×10^{-1} (3.10×10^{-4})
	15	24	**9.9033×10^{-1} (6.81×10^{-4}) +**	4.9608×10^{-1} (8.16×10^{-2}) −	9.8814×10^{-1} (1.06×10^{-3}) +	8.8308×10^{-2} (1.04×10^{-1}) −	9.8172×10^{-1} (1.38×10^{-3}) +	9.7544×10^{-1} (3.33×10^{-3}) +	8.7956×10^{-1} (2.56×10^{-3}) ≈	8.7930×10^{-1} (3.59×10^{-3})
MaF6	5	14	1.2294×10^{-1} (1.46×10^{-3}) −	5.3971×10^{-2} (5.00×10^{-2}) −	**1.3002×10^{-1} (3.45×10^{-4}) +**	1.3000×10^{-1} (4.03×10^{-4}) +	1.2983×10^{-1} (4.10×10^{-4}) ≈	1.2989×10^{-1} (4.02×10^{-4}) ≈	1.2964×10^{-1} (3.31×10^{-4}) ≈	1.2973×10^{-1} (4.61×10^{-4})
	10	19	1.5878×10^{-3} (7.10×10^{-3}) −	7.2840×10^{-2} (3.74×10^{-2}) +	0.0000×10^{0} (0.00×10^{0}) −	1.5109×10^{-2} (3.69×10^{-2}) −	6.2238×10^{-2} (4.16×10^{-2}) ≈	**1.0088×10^{-1} (3.48×10^{-4}) +**	6.8132×10^{-2} (4.00×10^{-2}) +	4.5524×10^{-2} (2.62×10^{-2})
	15	24	3.7735×10^{-4} (1.69×10^{-3}) −	6.7843×10^{-2} (4.02×10^{-2}) ≈	0.0000×10^{0} (0.00×10^{0}) −	0.0000×10^{0} (0.00×10^{0}) −	1.0869×10^{-2} (2.05×10^{-2}) −	**7.6645×10^{-2} (3.86×10^{-2}) +**	2.5879×10^{-2} (2.34×10^{-2}) −	7.4059×10^{-2} (2.31×10^{-2})
MaF7	5	24	2.4135×10^{-1} (4.02×10^{-3}) −	1.3913×10^{-1} (3.18×10^{-2}) −	2.5850×10^{-1} (5.25×10^{-3}) ≈	2.2738×10^{-1} (1.27×10^{-2}) −	2.3110×10^{-1} (6.25×10^{-3}) −	2.4795×10^{-1} (1.51×10^{-3}) −	2.5444×10^{-1} (3.21×10^{-3}) −	**2.6077×10^{-1} (2.75×10^{-3})**
	10	29	1.6975×10^{-1} (4.03×10^{-3}) +	1.0346×10^{-2} (5.32×10^{-3}) −	**1.8894×10^{-1} (1.31×10^{-2}) +**	1.1863×10^{-1} (1.97×10^{-2}) −	6.4935×10^{-2} (1.94×10^{-2}) −	1.2174×10^{-1} (8.44×10^{-3}) −	1.1235×10^{-1} (2.10×10^{-2}) −	1.5393×10^{-1} (3.05×10^{-4})
	15	34	6.9636×10^{-2} (1.52×10^{-2}) −	2.4443×10^{-4} (5.55×10^{-4}) −	**1.4706×10^{-1} (7.54×10^{-3}) +**	3.3819×10^{-3} (4.85×10^{-3}) −	4.3577×10^{-2} (3.11×10^{-2}) −	6.7486×10^{-2} (1.33×10^{-2}) −	8.6084×10^{-2} (2.96×10^{-2}) −	1.1863×10^{-1} (2.22×10^{-2})
MaF8	5	2	1.0443×10^{-1} (2.13×10^{-3}) −	4.8073×10^{-2} (1.47×10^{-2}) −	1.1934×10^{-1} (3.72×10^{-3}) −	1.2380×10^{-1} (2.55×10^{-3}) −	1.2531×10^{-1} (3.18×10^{-4}) −	1.2522×10^{-1} (8.78×10^{-4}) −	1.2584×10^{-1} (3.65×10^{-4}) ≈	**1.2592×10^{-1} (4.35×10^{-4})**
	10	2	9.2756×10^{-3} (1.86×10^{-4}) −	2.7814×10^{-3} (7.32×10^{-4}) −	1.0409×10^{-2} (3.56×10^{-4}) −	1.0992×10^{-2} (1.07×10^{-4}) ≈	**1.1062×10^{-2} (9.13×10^{-5}) ≈**	1.0665×10^{-2} (1.27×10^{-4}) −	1.1005×10^{-2} (8.95×10^{-5}) ≈	1.1005×10^{-2} (1.15×10^{-4})
	15	2	5.1133×10^{-4} (2.33×10^{-5}) −	9.6382×10^{-5} (4.48×10^{-5}) −	5.7226×10^{-4} (2.98×10^{-5}) −	6.0939×10^{-4} (5.90×10^{-5}) −	**6.6263×10^{-4} (7.65×10^{-6}) +**	5.8681×10^{-4} (2.64×10^{-5}) −	6.4957×10^{-4} (1.89×10^{-5}) ≈	6.5171×10^{-4} (1.13×10^{-5})
MaF9	5	2	1.5920×10^{-1} (4.41×10^{-2}) −	9.8072×10^{-2} (5.14×10^{-2}) −	2.4131×10^{-1} (4.13×10^{-2}) −	2.2129×10^{-1} (2.74×10^{-2}) −	3.2379×10^{-1} (8.96×10^{-4}) ≈	**3.2550×10^{-1} (1.43×10^{-3}) +**	3.2532×10^{-1} (7.10×10^{-4}) +	3.2427×10^{-1} (1.01×10^{-3})
	10	2	8.9235×10^{-3} (1.42×10^{-3}) −	4.5008×10^{-3} (2.11×10^{-3}) −	9.0017×10^{-3} (2.06×10^{-3}) −	4.3945×10^{-3} (1.24×10^{-3}) −	1.7446×10^{-2} (2.93×10^{-4}) −	1.5473×10^{-2} (1.47×10^{-3}) −	**1.8576×10^{-2} (1.20×10^{-4}) ≈**	1.8572×10^{-2} (1.21×10^{-4})
	15	2	8.3268×10^{-4} (2.12×10^{-4}) −	1.1835×10^{-4} (1.57×10^{-4}) −	8.3976×10^{-4} (1.67×10^{-4}) −	7.8030×10^{-4} (2.67×10^{-4}) −	1.2383×10^{-3} (2.94×10^{-4}) ≈	1.2736×10^{-3} (4.44×10^{-5}) −	1.2938×10^{-3} (4.12×10^{-5}) ≈	**1.3010×10^{-3} (4.09×10^{-5})**
MaF10	5	14	**9.9739×10^{-1} (3.30×10^{-4}) +**	7.9512×10^{-1} (7.07×10^{-2}) −	9.1315×10^{-1} (2.72×10^{-2}) −	9.1388×10^{-1} (3.33×10^{-2}) −	9.8879×10^{-1} (1.71×10^{-2}) ≈	8.6430×10^{-1} (4.35×10^{-2}) −	9.9565×10^{-1} (2.68×10^{-3}) ≈	9.9591×10^{-1} (5.39×10^{-4})
	10	19	9.9791×10^{-1} (2.29×10^{-3}) ≈	6.1410×10^{-1} (2.47×10^{-1}) −	9.7668×10^{-1} (4.61×10^{-2}) −	**9.9917×10^{-1} (8.22×10^{-4}) +**	9.9636×10^{-1} (1.06×10^{-2}) ≈	9.6461×10^{-1} (4.01×10^{-2}) −	9.9835×10^{-1} (7.74×10^{-4}) ≈	9.9818×10^{-1} (8.12×10^{-4})
	15	24	9.9938×10^{-1} (2.49×10^{-4}) +	8.6770×10^{-1} (1.05×10^{-1}) −	9.7960×10^{-1} (3.58×10^{-2}) −	**9.9968×10^{-1} (1.66×10^{-4}) +**	9.9900×10^{-1} (5.05×10^{-4}) +	9.9777×10^{-1} (6.09×10^{-4}) −	9.9839×10^{-1} (4.74×10^{-4}) ≈	9.9846×10^{-1} (6.07×10^{-4})
+/−/≈			10/17/3	2/26/2	8/18/4	5/21/4	6/12/12	11/15/4	4/11/15	

Table 5. IGD+ values obtained by GPDARVC and other comparison algorithms on 5-, 10-, and 15-objective WFG 1-9. The best result for each test instance is shown with dark background.

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
WFG1	5	14	$1.6879 \times 10^{-1} (1.04 \times 10^{-2})$ −	$8.4047 \times 10^{-1} (7.93 \times 10^{-1})$ −	$2.7932 \times 10^{-1} (5.14 \times 10^{-2})$ −	$3.4056 \times 10^{-1} (5.90 \times 10^{-2})$ −	$1.7181 \times 10^{-1} (3.53 \times 10^{-2})$ −	$3.9556 \times 10^{-1} (7.17 \times 10^{-2})$ −	$1.2037 \times 10^{-1} (1.68 \times 10^{-2})$ −	$1.0612 \times 10^{-1} (1.51 \times 10^{-3})$
	10	19	$2.7284 \times 10^{-1} (3.52 \times 10^{-2})$ −	$4.3105 \times 10^{0} (3.58 \times 10^{0})$ −	$4.2434 \times 10^{-1} (8.32 \times 10^{-2})$ −	$3.5470 \times 10^{-1} (7.15 \times 10^{-2})$ −	$3.9092 \times 10^{-1} (6.16 \times 10^{-2})$ −	$4.3791 \times 10^{-1} (5.09 \times 10^{-2})$ −	$2.3283 \times 10^{-1} (6.33 \times 10^{-2})$ −	$1.8100 \times 10^{-1} (5.32 \times 10^{-3})$
	15	24	$3.7314 \times 10^{-1} (5.29 \times 10^{-2})$ −	$6.8503 \times 10^{0} (3.59 \times 10^{0})$ −	$5.4497 \times 10^{-1} (8.73 \times 10^{-2})$ −	$4.1353 \times 10^{-1} (9.65 \times 10^{-2})$ −	$4.8966 \times 10^{-1} (8.79 \times 10^{-2})$ −	$4.8920 \times 10^{-1} (1.24 \times 10^{-1})$ −	$2.6991 \times 10^{-1} (6.09 \times 10^{-2})$ −	$2.2207 \times 10^{-1} (2.85 \times 10^{-2})$
WFG2	5	14	$1.7096 \times 10^{-1} (5.95 \times 10^{-3})$ −	$4.0120 \times 10^{-1} (1.60 \times 10^{-1})$ −	$1.4609 \times 10^{-1} (1.20 \times 10^{-2})$ −	$2.3480 \times 10^{-1} (1.77 \times 10^{-2})$ −	$1.6718 \times 10^{-1} (1.33 \times 10^{-2})$ −	$1.2269 \times 10^{-1} (4.17 \times 10^{-3})$ −	$1.1988 \times 10^{-1} (4.92 \times 10^{-3})$ −	$1.0821 \times 10^{-1} (3.25 \times 10^{-3})$
	10	19	$2.7613 \times 10^{-1} (3.52 \times 10^{-2})$ −	$8.7626 \times 10^{-1} (2.54 \times 10^{-1})$ −	$3.1333 \times 10^{-1} (5.85 \times 10^{-2})$ −	$6.4335 \times 10^{-1} (1.00 \times 10^{-1})$ −	$3.0931 \times 10^{-1} (1.89 \times 10^{-2})$ −	$2.5557 \times 10^{-1} (1.54 \times 10^{-2})$ −	$2.4792 \times 10^{-1} (1.31 \times 10^{-2})$ −	$1.8291 \times 10^{-1} (1.39 \times 10^{-2})$
	15	24	$4.6028 \times 10^{-1} (7.80 \times 10^{-2})$ −	$5.3890 \times 10^{-1} (4.74 \times 10^{-1})$ −	$9.8919 \times 10^{-1} (2.92 \times 10^{-1})$ −	$8.0519 \times 10^{-1} (1.72 \times 10^{-1})$ −	$4.0059 \times 10^{-1} (4.11 \times 10^{-2})$ −	$3.8666 \times 10^{-1} (7.76 \times 10^{-2})$ −	$2.6561 \times 10^{-1} (1.34 \times 10^{-2})$ −	$2.3494 \times 10^{-1} (3.23 \times 10^{-2})$
WFG3	5	14	$4.7816 \times 10^{-1} (7.93 \times 10^{-2})$ ≈	$3.0563 \times 10^{0} (2.44 \times 10^{0})$ ≈	$4.2345 \times 10^{-1} (1.32 \times 10^{-1})$ +	$3.8317 \times 10^{-1} (5.46 \times 10^{-2})$ +	$6.5899 \times 10^{-1} (1.41 \times 10^{-1})$ −	$2.3907 \times 10^{-1} (2.25 \times 10^{-2})$ +	$6.0440 \times 10^{-1} (1.31 \times 10^{-1})$ −	$4.6603 \times 10^{-1} (5.86 \times 10^{-2})$
	10	19	$1.1414 \times 10^{0} (2.82 \times 10^{-1})$ +	$2.6230 \times 10^{0} (3.19 \times 10^{0})$ ≈	$9.0995 \times 10^{-1} (2.16 \times 10^{-1})$ +	$1.7089 \times 10^{0} (4.31 \times 10^{-1})$ −	$2.1616 \times 10^{0} (2.72 \times 10^{-1})$ −	$6.7516 \times 10^{-1} (1.04 \times 10^{-1})$ +	$2.0015 \times 10^{0} (2.42 \times 10^{-1})$ −	$1.3803 \times 10^{0} (1.44 \times 10^{-1})$
	15	24	$8.8343 \times 10^{-1} (4.40 \times 10^{-1})$ +	$5.8458 \times 10^{0} (4.73 \times 10^{0})$ −	$1.3142 \times 10^{0} (4.11 \times 10^{-1})$ +	$2.7530 \times 10^{0} (5.88 \times 10^{-1})$ −	$2.8995 \times 10^{0} (3.69 \times 10^{-1})$ −	$1.1996 \times 10^{0} (1.55 \times 10^{-1})$ +	$2.6914 \times 10^{0} (4.05 \times 10^{-1})$ −	$2.1122 \times 10^{0} (2.98 \times 10^{-1})$
WFG4	5	14	$3.4865 \times 10^{-1} (6.25 \times 10^{-3})$ −	$1.9450 \times 10^{0} (1.82 \times 10^{0})$ −	$4.3914 \times 10^{-1} (1.37 \times 10^{-2})$ −	$3.3610 \times 10^{-1} (4.05 \times 10^{-3})$ −	$4.2041 \times 10^{-1} (1.36 \times 10^{-2})$ −	$3.6107 \times 10^{-1} (5.60 \times 10^{-3})$ −	$3.7270 \times 10^{-1} (8.00 \times 10^{-3})$ −	$3.2184 \times 10^{-1} (6.14 \times 10^{-3})$
	10	19	$1.2076 \times 10^{0} (2.07 \times 10^{-1})$ −	$1.5911 \times 10^{0} (7.62 \times 10^{-3})$ −	$1.6910 \times 10^{0} (1.94 \times 10^{-1})$ −	$1.0553 \times 10^{0} (1.51 \times 10^{-2})$ −	$1.4249 \times 10^{0} (4.86 \times 10^{-2})$ −	$1.2323 \times 10^{0} (1.85 \times 10^{-2})$ −	$9.7903 \times 10^{-1} (1.57 \times 10^{-2})$ −	$9.6320 \times 10^{-1} (2.24 \times 10^{-2})$
	15	24	$2.1382 \times 10^{0} (5.64 \times 10^{-1})$ −	$7.9585 \times 10^{0} (7.55 \times 10^{0})$ −	$4.3857 \times 10^{0} (6.79 \times 10^{-1})$ −	$1.5059 \times 10^{0} (1.85 \times 10^{-1})$ −	$2.1251 \times 10^{0} (1.42 \times 10^{-1})$ −	$1.6988 \times 10^{0} (1.33 \times 10^{-1})$ −	$1.4436 \times 10^{0} (3.42 \times 10^{-2})$ −	$1.3992 \times 10^{0} (4.20 \times 10^{-2})$
WFG5	5	14	$4.0436 \times 10^{-1} (5.52 \times 10^{-3})$ −	$5.7518 \times 10^{-1} (7.70 \times 10^{-2})$ −	$4.9843 \times 10^{-1} (1.40 \times 10^{-2})$ −	$3.9344 \times 10^{-1} (3.90 \times 10^{-3})$ −	$4.7974 \times 10^{-1} (7.81 \times 10^{-3})$ −	$4.0874 \times 10^{-1} (3.66 \times 10^{-3})$ −	$4.4547 \times 10^{-1} (6.82 \times 10^{-3})$ −	$3.6670 \times 10^{-1} (5.91 \times 10^{-3})$
	10	19	$1.2411 \times 10^{0} (1.84 \times 10^{-2})$ −	$6.1110 \times 10^{0} (7.10 \times 10^{0})$ ≈	$1.9677 \times 10^{0} (4.08 \times 10^{-1})$ −	$1.1390 \times 10^{0} (1.53 \times 10^{-2})$ −	$1.6906 \times 10^{0} (6.26 \times 10^{-2})$ −	$1.3560 \times 10^{0} (6.68 \times 10^{-2})$ −	$1.2222 \times 10^{0} (4.16 \times 10^{-2})$ −	$1.0257 \times 10^{0} (1.58 \times 10^{-2})$
	15	24	$2.0035 \times 10^{0} (5.31 \times 10^{-1})$ −	$1.9696 \times 10^{1} (9.54 \times 10^{0})$ −	$4.8177 \times 10^{0} (1.46 \times 10^{0})$ −	$1.5583 \times 10^{0} (2.10 \times 10^{-2})$ −	$2.8805 \times 10^{0} (2.43 \times 10^{-1})$ −	$1.9200 \times 10^{0} (2.25 \times 10^{-1})$ −	$1.8096 \times 10^{0} (7.35 \times 10^{-2})$ −	$1.4755 \times 10^{0} (2.33 \times 10^{-2})$
WFG6	5	14	$4.4307 \times 10^{-1} (1.69 \times 10^{-2})$ −	$1.4079 \times 10^{0} (9.69 \times 10^{-1})$ −	$5.4214 \times 10^{-1} (3.37 \times 10^{-2})$ −	$4.1629 \times 10^{-1} (1.42 \times 10^{-2})$ −	$5.5529 \times 10^{-1} (3.56 \times 10^{-2})$ −	$4.4503 \times 10^{-1} (2.58 \times 10^{-2})$ −	$4.9629 \times 10^{-1} (2.58 \times 10^{-2})$ −	$4.0251 \times 10^{-1} (2.25 \times 10^{-2})$
	10	19	$1.2027 \times 10^{0} (4.56 \times 10^{-2})$ −	$5.6966 \times 10^{0} (4.84 \times 10^{0})$ −	$2.0388 \times 10^{0} (8.15 \times 10^{-1})$ −	$1.1168 \times 10^{0} (2.50 \times 10^{-2})$ −	$1.7499 \times 10^{0} (9.49 \times 10^{-2})$ −	$1.3074 \times 10^{0} (5.77 \times 10^{-2})$ −	$1.1484 \times 10^{0} (5.35 \times 10^{-2})$ −	$1.0760 \times 10^{0} (3.40 \times 10^{-2})$
	15	24	$1.5688 \times 10^{0} (1.02 \times 10^{-1})$ −	$8.7736 \times 10^{0} (9.46 \times 10^{0})$ ≈	$3.7892 \times 10^{0} (2.28 \times 10^{0})$ −	$1.5028 \times 10^{0} (2.64 \times 10^{-2})$ −	$2.5564 \times 10^{0} (2.42 \times 10^{-1})$ −	$1.9234 \times 10^{0} (2.86 \times 10^{-1})$ −	$1.5690 \times 10^{0} (1.11 \times 10^{-1})$ −	$1.4595 \times 10^{0} (2.50 \times 10^{-2})$
WFG7	5	14	$3.5370 \times 10^{-1} (7.85 \times 10^{-3})$ −	$1.0023 \times 10^{0} (9.38 \times 10^{-1})$ −	$4.4150 \times 10^{-1} (2.04 \times 10^{-2})$ −	$3.2445 \times 10^{-1} (3.17 \times 10^{-3})$ −	$3.8391 \times 10^{-1} (7.99 \times 10^{-3})$ −	$3.4186 \times 10^{-1} (4.06 \times 10^{-3})$ −	$3.4372 \times 10^{-1} (6.61 \times 10^{-3})$ −	$3.1604 \times 10^{-1} (5.65 \times 10^{-3})$
	10	19	$1.2405 \times 10^{0} (1.83 \times 10^{-1})$ −	$2.0352 \times 10^{0} (2.06 \times 10^{0})$ −	$2.6192 \times 10^{0} (7.96 \times 10^{-1})$ −	$1.0307 \times 10^{0} (1.52 \times 10^{-2})$ −	$1.2931 \times 10^{0} (4.82 \times 10^{-2})$ −	$1.2469 \times 10^{0} (7.02 \times 10^{-2})$ −	$9.6213 \times 10^{-1} (1.46 \times 10^{-2})$ ≈	$9.6175 \times 10^{-1} (1.99 \times 10^{-2})$
	15	24	$2.6539 \times 10^{0} (3.19 \times 10^{-1})$ −	$1.4996 \times 10^{1} (6.78 \times 10^{0})$ −	$7.2836 \times 10^{0} (1.22 \times 10^{0})$ −	$1.6876 \times 10^{0} (3.15 \times 10^{-1})$ −	$1.8703 \times 10^{0} (1.92 \times 10^{-1})$ −	$1.9153 \times 10^{0} (2.48 \times 10^{-1})$ −	$1.3936 \times 10^{0} (2.58 \times 10^{-2})$ −	$1.3687 \times 10^{0} (3.25 \times 10^{-2})$
WFG8	5	14	$7.0606 \times 10^{-1} (1.94 \times 10^{-2})$ −	$1.6805 \times 10^{0} (4.10 \times 10^{-1})$ −	$8.1117 \times 10^{-1} (4.86 \times 10^{-2})$ −	$6.1329 \times 10^{-1} (7.03 \times 10^{-3})$ −	$7.5665 \times 10^{-1} (1.82 \times 10^{-2})$ −	$6.3778 \times 10^{-1} (9.83 \times 10^{-3})$ −	$7.0094 \times 10^{-1} (1.09 \times 10^{-2})$ −	$6.0005 \times 10^{-1} (3.45 \times 10^{-3})$
	10	19	$2.9083 \times 10^{0} (3.54 \times 10^{-1})$ −	$8.3831 \times 10^{0} (1.89 \times 10^{0})$ −	$4.4566 \times 10^{0} (3.02 \times 10^{-1})$ −	$2.3124 \times 10^{0} (1.04 \times 10^{-1})$ −	$2.9543 \times 10^{0} (9.12 \times 10^{-2})$ −	$1.7865 \times 10^{0} (2.62 \times 10^{-1})$ −	$2.2224 \times 10^{0} (4.60 \times 10^{-1})$ −	$1.4763 \times 10^{0} (2.26 \times 10^{-1})$
	15	24	$5.2552 \times 10^{0} (1.03 \times 10^{0})$ ≈	$1.7779 \times 10^{1} (3.46 \times 10^{0})$ −	$9.8270 \times 10^{0} (6.85 \times 10^{-1})$ −	$4.1964 \times 10^{0} (5.71 \times 10^{-1})$ ≈	$6.0122 \times 10^{0} (3.44 \times 10^{-1})$ ≈	$3.2768 \times 10^{0} (7.39 \times 10^{-1})$ +	$6.3542 \times 10^{0} (3.78 \times 10^{-1})$ −	$4.9284 \times 10^{0} (1.58 \times 10^{0})$
WFG9	5	14	$4.3455 \times 10^{-1} (6.95 \times 10^{-2})$ −	$6.3084 \times 10^{-1} (2.68 \times 10^{-1})$ −	$4.8013 \times 10^{-1} (1.83 \times 10^{-2})$ −	$3.8708 \times 10^{-1} (1.11 \times 10^{-2})$ −	$4.4516 \times 10^{-1} (1.40 \times 10^{-2})$ −	$4.0339 \times 10^{-1} (7.93 \times 10^{-3})$ −	$4.2105 \times 10^{-1} (1.12 \times 10^{-2})$ −	$3.7320 \times 10^{-1} (1.04 \times 10^{-2})$
	10	19	$1.9231 \times 10^{0} (4.06 \times 10^{-1})$ −	$1.3237 \times 10^{0} (1.26 \times 10^{0})$ −	$2.5773 \times 10^{0} (3.65 \times 10^{-1})$ −	$1.3674 \times 10^{0} (5.77 \times 10^{-2})$ −	$1.6376 \times 10^{0} (1.57 \times 10^{-1})$ −	$1.5029 \times 10^{0} (6.31 \times 10^{-2})$ −	$1.2111 \times 10^{0} (4.62 \times 10^{-2})$ −	$1.0912 \times 10^{0} (4.55 \times 10^{-2})$
	15	24	$3.4744 \times 10^{0} (1.87 \times 10^{-1})$ −	$5.0539 \times 10^{0} (7.16 \times 10^{0})$ −	$5.0051 \times 10^{0} (6.87 \times 10^{-1})$ −	$2.7106 \times 10^{0} (3.24 \times 10^{-1})$ −	$2.5771 \times 10^{0} (1.51 \times 10^{-1})$ −	$2.5878 \times 10^{0} (3.32 \times 10^{-1})$ −	$1.9573 \times 10^{0} (1.53 \times 10^{-1})$ −	$1.7683 \times 10^{0} (1.28 \times 10^{-1})$
+/−/≈			2/23/2	0/23/4	3/24/0	1/25/1	0/26/1	4/23/0	0/26/1	

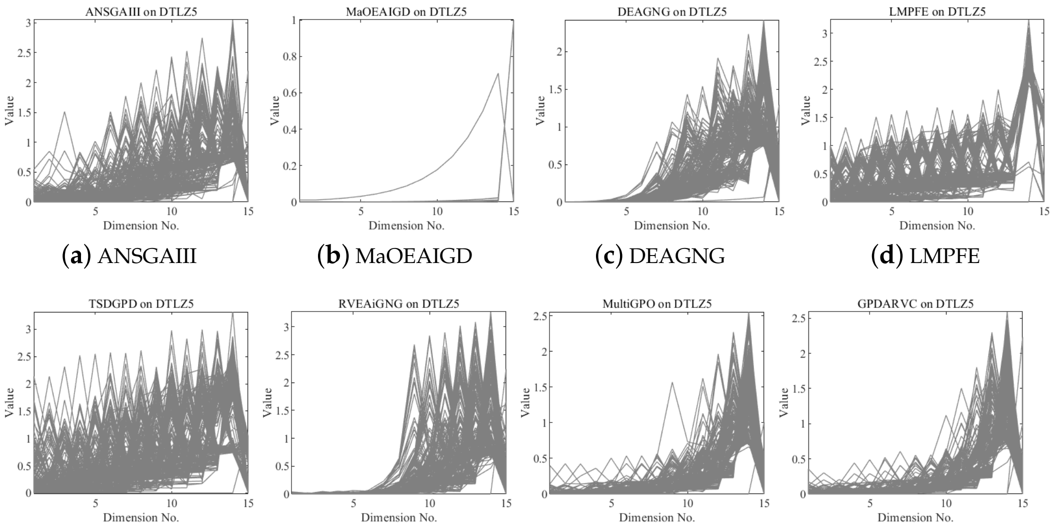
Table 6. HV values obtained by GPDARVC and other comparison algorithms on 5-, 10-, and 15-objective WFG 1-9. The best result for each test instance is shown with dark background.

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
WFG1	5	14	**$9.9729 \times 10^{-1}\ (4.79 \times 10^{-4})$ +**	$7.5701 \times 10^{-1}\ (1.13 \times 10^{-1})$ −	$9.2224 \times 10^{-1}\ (3.02 \times 10^{-2})$ −	$9.2482 \times 10^{-1}\ (3.82 \times 10^{-2})$ −	$9.8940 \times 10^{-1}\ (1.93 \times 10^{-2})$ ≈	$8.6548 \times 10^{-1}\ (4.29 \times 10^{-2})$ −	$9.9357 \times 10^{-1}\ (1.06 \times 10^{-2})$ −	$9.9624 \times 10^{-1}\ (4.40 \times 10^{-4})$
	10	19	$9.9880 \times 10^{-1}\ (4.02 \times 10^{-4})$ +	$7.7590 \times 10^{-1}\ (2.05 \times 10^{-1})$ −	$9.7484 \times 10^{-1}\ (3.39 \times 10^{-2})$ −	$9.9692 \times 10^{-1}\ (1.13 \times 10^{-2})$ −	**$9.9886 \times 10^{-1}\ (6.89 \times 10^{-4})$ +**	$9.8202 \times 10^{-1}\ (3.12 \times 10^{-2})$ −	$9.9846 \times 10^{-1}\ (6.07 \times 10^{-4})$ ≈	$9.9805 \times 10^{-1}\ (7.91 \times 10^{-4})$
	15	24	$9.9938 \times 10^{-1}\ (2.46 \times 10^{-4})$ +	$8.1670 \times 10^{-1}\ (1.42 \times 10^{-1})$ −	$9.9112 \times 10^{-1}\ (2.96 \times 10^{-3})$ −	**$9.9966 \times 10^{-1}\ (2.22 \times 10^{-4})$ +**	$9.9919 \times 10^{-1}\ (5.06 \times 10^{-4})$ +	$9.9779 \times 10^{-1}\ (4.44 \times 10^{-4})$ −	$9.9846 \times 10^{-1}\ (5.37 \times 10^{-4})$ ≈	$9.9850 \times 10^{-1}\ (8.21 \times 10^{-4})$
WFG2	5	14	**$9.9385 \times 10^{-1}\ (1.63 \times 10^{-3})$ +**	$9.4873 \times 10^{-1}\ (1.92 \times 10^{-2})$ −	$9.6991 \times 10^{-1}\ (4.15 \times 10^{-3})$ −	$9.9260 \times 10^{-1}\ (1.75 \times 10^{-3})$ +	$9.8210 \times 10^{-1}\ (2.76 \times 10^{-3})$ −	$9.8640 \times 10^{-1}\ (1.93 \times 10^{-3})$ ≈	$9.8536 \times 10^{-1}\ (2.11 \times 10^{-3})$ −	$9.8743 \times 10^{-1}\ (1.98 \times 10^{-3})$
	10	19	$9.9456 \times 10^{-1}\ (2.12 \times 10^{-3})$ +	$9.7842 \times 10^{-1}\ (1.32 \times 10^{-2})$ −	$9.8304 \times 10^{-1}\ (2.92 \times 10^{-3})$ −	**$9.9771 \times 10^{-1}\ (7.09 \times 10^{-4})$ +**	$9.9329 \times 10^{-1}\ (1.21 \times 10^{-3})$ +	$9.9215 \times 10^{-1}\ (1.39 \times 10^{-3})$ ≈	$9.9307 \times 10^{-1}\ (2.10 \times 10^{-3})$ +	$9.9216 \times 10^{-1}\ (1.81 \times 10^{-3})$
	15	24	$9.9446 \times 10^{-1}\ (2.16 \times 10^{-3})$ +	$9.5815 \times 10^{-1}\ (1.77 \times 10^{-2})$ −	$9.7537 \times 10^{-1}\ (9.91 \times 10^{-3})$ −	**$9.9633 \times 10^{-1}\ (1.29 \times 10^{-3})$ +**	$9.9418 \times 10^{-1}\ (1.55 \times 10^{-3})$ +	$9.9488 \times 10^{-1}\ (1.08 \times 10^{-3})$ +	$9.9345 \times 10^{-1}\ (3.26 \times 10^{-3})$ ≈	$9.9217 \times 10^{-1}\ (3.01 \times 10^{-3})$
WFG3	5	14	$1.2313 \times 10^{-1}\ (2.34 \times 10^{-2})$ ≈	**$1.3726 \times 10^{-1}\ (6.71 \times 10^{-2})$ ≈**	$7.9633 \times 10^{-2}\ (3.04 \times 10^{-2})$ −	$1.3248 \times 10^{-1}\ (2.30 \times 10^{-2})$ ≈	$2.9777 \times 10^{-2}\ (2.79 \times 10^{-2})$ −	$1.3689 \times 10^{-1}\ (2.05 \times 10^{-2})$ ≈	$3.3277 \times 10^{-2}\ (2.60 \times 10^{-2})$ −	$1.3093 \times 10^{-1}\ (1.75 \times 10^{-2})$
	10	19	**$1.0038 \times 10^{-3}\ (3.09 \times 10^{-3})$ ≈**	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$4.0794 \times 10^{-4}\ (1.39 \times 10^{-3})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$
	15	24	**$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈**	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$ ≈	$0.0000 \times 10^{0}\ (0.00 \times 10^{0})$
WFG4	5	14	$7.8150 \times 10^{-1}\ (3.53 \times 10^{-3})$ −	$2.7809 \times 10^{-1}\ (2.17 \times 10^{-1})$ −	$7.2951 \times 10^{-1}\ (5.59 \times 10^{-3})$ −	$7.8987 \times 10^{-1}\ (2.33 \times 10^{-3})$ −	$7.3936 \times 10^{-1}\ (7.22 \times 10^{-3})$ −	$7.7712 \times 10^{-1}\ (3.86 \times 10^{-3})$ −	$7.6398 \times 10^{-1}\ (5.03 \times 10^{-3})$ −	**$7.9304 \times 10^{-1}\ (3.16 \times 10^{-3})$**
	10	19	$8.9006 \times 10^{-1}\ (1.29 \times 10^{-2})$ −	$1.1465 \times 10^{-1}\ (3.80 \times 10^{-2})$ −	$8.2631 \times 10^{-1}\ (9.69 \times 10^{-3})$ −	$9.4904 \times 10^{-1}\ (2.03 \times 10^{-3})$ −	$8.3550 \times 10^{-1}\ (9.08 \times 10^{-3})$ −	$9.0372 \times 10^{-1}\ (4.48 \times 10^{-3})$ −	**$9.5702 \times 10^{-1}\ (2.44 \times 10^{-3})$ +**	$9.5320 \times 10^{-1}\ (5.39 \times 10^{-3})$
	15	24	$9.2280 \times 10^{-1}\ (1.76 \times 10^{-2})$ −	$1.6712 \times 10^{-1}\ (1.57 \times 10^{-1})$ −	$8.3298 \times 10^{-1}\ (2.58 \times 10^{-2})$ −	$9.7464 \times 10^{-1}\ (3.52 \times 10^{-3})$ −	$8.2140 \times 10^{-1}\ (1.34 \times 10^{-2})$ −	$9.1813 \times 10^{-1}\ (6.86 \times 10^{-3})$ −	**$9.8614 \times 10^{-1}\ (9.74 \times 10^{-4})$ +**	$9.8307 \times 10^{-1}\ (5.42 \times 10^{-3})$
WFG5	5	14	$7.3641 \times 10^{-1}\ (2.74 \times 10^{-3})$ −	$5.9047 \times 10^{-1}\ (4.59 \times 10^{-2})$ −	$6.7534 \times 10^{-1}\ (7.28 \times 10^{-3})$ −	$7.4162 \times 10^{-1}\ (2.72 \times 10^{-3})$ −	$7.0602 \times 10^{-1}\ (4.99 \times 10^{-3})$ −	$7.3525 \times 10^{-1}\ (3.38 \times 10^{-3})$ −	$7.2390 \times 10^{-1}\ (4.27 \times 10^{-3})$ −	**$7.5300 \times 10^{-1}\ (2.97 \times 10^{-3})$**
	10	19	$8.4977 \times 10^{-1}\ (5.61 \times 10^{-3})$ −	$4.8967 \times 10^{-1}\ (3.06 \times 10^{-1})$ −	$7.6801 \times 10^{-1}\ (1.44 \times 10^{-2})$ −	$8.8577 \times 10^{-1}\ (2.36 \times 10^{-3})$ −	$7.9606 \times 10^{-1}\ (7.04 \times 10^{-3})$ −	$8.5903 \times 10^{-1}\ (4.45 \times 10^{-3})$ −	$8.8579 \times 10^{-1}\ (2.02 \times 10^{-3})$ −	**$9.0141 \times 10^{-1}\ (1.28 \times 10^{-3})$**
	15	24	$8.7862 \times 10^{-1}\ (1.01 \times 10^{-2})$ −	$2.2781 \times 10^{-1}\ (2.79 \times 10^{-1})$ −	$7.8038 \times 10^{-1}\ (3.15 \times 10^{-2})$ −	$9.0388 \times 10^{-1}\ (2.08 \times 10^{-3})$ −	$7.5518 \times 10^{-1}\ (1.07 \times 10^{-2})$ −	$8.7299 \times 10^{-1}\ (3.09 \times 10^{-3})$ −	$8.9638 \times 10^{-1}\ (2.99 \times 10^{-3})$ −	**$9.1473 \times 10^{-1}\ (1.00 \times 10^{-3})$**
WFG6	5	14	$7.1109 \times 10^{-1}\ (1.12 \times 10^{-2})$ −	$2.0909 \times 10^{-1}\ (9.44 \times 10^{-2})$ −	$6.5917 \times 10^{-1}\ (1.82 \times 10^{-2})$ −	$7.3093 \times 10^{-1}\ (1.06 \times 10^{-2})$ ≈	$6.8227 \times 10^{-1}\ (1.56 \times 10^{-2})$ −	$7.1647 \times 10^{-1}\ (1.75 \times 10^{-2})$ −	$7.0659 \times 10^{-1}\ (1.33 \times 10^{-2})$ −	**$7.3125 \times 10^{-1}\ (1.50 \times 10^{-2})$**
	10	19	$8.3944 \times 10^{-1}\ (1.14 \times 10^{-2})$ −	$3.7260 \times 10^{-1}\ (1.97 \times 10^{-1})$ −	$7.5420 \times 10^{-1}\ (3.52 \times 10^{-2})$ −	$8.7926 \times 10^{-1}\ (1.21 \times 10^{-2})$ +	$7.8060 \times 10^{-1}\ (1.81 \times 10^{-2})$ −	$8.4265 \times 10^{-1}\ (2.35 \times 10^{-2})$ −	**$8.8029 \times 10^{-1}\ (1.79 \times 10^{-2})$ +**	$8.6612 \times 10^{-1}\ (1.85 \times 10^{-2})$
	15	24	$8.7588 \times 10^{-1}\ (1.13 \times 10^{-2})$ +	$4.1887 \times 10^{-1}\ (2.13 \times 10^{-1})$ −	$7.7995 \times 10^{-1}\ (4.51 \times 10^{-2})$ −	**$9.0298 \times 10^{-1}\ (2.09 \times 10^{-2})$ +**	$7.7459 \times 10^{-1}\ (1.38 \times 10^{-2})$ −	$8.4469 \times 10^{-1}\ (1.84 \times 10^{-2})$ ≈	$8.6505 \times 10^{-1}\ (4.74 \times 10^{-2})$ ≈	$8.5845 \times 10^{-1}\ (2.92 \times 10^{-2})$
WFG7	5	14	$7.8156 \times 10^{-1}\ (3.89 \times 10^{-3})$ −	$4.2979 \times 10^{-1}\ (1.33 \times 10^{-1})$ −	$7.3228 \times 10^{-1}\ (8.15 \times 10^{-3})$ −	**$7.9954 \times 10^{-1}\ (1.12 \times 10^{-3})$ +**	$7.6415 \times 10^{-1}\ (4.30 \times 10^{-3})$ −	$7.9277 \times 10^{-1}\ (1.76 \times 10^{-3})$ −	$7.8435 \times 10^{-1}\ (3.56 \times 10^{-3})$ −	$7.9760 \times 10^{-1}\ (3.31 \times 10^{-3})$
	10	19	$9.0354 \times 10^{-1}\ (1.29 \times 10^{-2})$ −	$6.7282 \times 10^{-1}\ (1.41 \times 10^{-1})$ −	$8.2953 \times 10^{-1}\ (3.98 \times 10^{-2})$ −	$9.5925 \times 10^{-1}\ (9.79 \times 10^{-4})$ ≈	$8.7871 \times 10^{-1}\ (1.03 \times 10^{-2})$ −	$9.2998 \times 10^{-1}\ (3.85 \times 10^{-3})$ −	**$9.6340 \times 10^{-1}\ (2.77 \times 10^{-3})$ +**	$9.5956 \times 10^{-1}\ (3.66 \times 10^{-3})$
	15	24	$9.7311 \times 10^{-1}\ (4.01 \times 10^{-3})$ −	$2.5560 \times 10^{-1}\ (2.54 \times 10^{-1})$ −	$8.3713 \times 10^{-1}\ (4.21 \times 10^{-2})$ −	$9.7966 \times 10^{-1}\ (3.53 \times 10^{-3})$ −	$8.9319 \times 10^{-1}\ (1.65 \times 10^{-2})$ −	$9.3806 \times 10^{-1}\ (5.73 \times 10^{-3})$ −	**$9.9137 \times 10^{-1}\ (4.88 \times 10^{-4})$ +**	$9.9028 \times 10^{-1}\ (1.29 \times 10^{-3})$
WFG8	5	14	$6.1886 \times 10^{-1}\ (1.06 \times 10^{-2})$ −	$7.2921 \times 10^{-2}\ (7.56 \times 10^{-2})$ −	$6.0607 \times 10^{-1}\ (1.52 \times 10^{-2})$ −	$6.8035 \times 10^{-1}\ (4.13 \times 10^{-3})$ −	$6.0196 \times 10^{-1}\ (1.26 \times 10^{-2})$ −	$6.6704 \times 10^{-1}\ (3.48 \times 10^{-3})$ −	$6.3234 \times 10^{-1}\ (6.33 \times 10^{-3})$ −	**$6.8976 \times 10^{-1}\ (2.81 \times 10^{-3})$**
	10	19	$8.5186 \times 10^{-1}\ (1.64 \times 10^{-2})$ −	$2.6028 \times 10^{-1}\ (1.24 \times 10^{-1})$ −	$7.1378 \times 10^{-1}\ (2.35 \times 10^{-2})$ −	$8.2294 \times 10^{-1}\ (6.10 \times 10^{-3})$ −	$5.9263 \times 10^{-1}\ (2.46 \times 10^{-2})$ −	$8.2116 \times 10^{-1}\ (2.60 \times 10^{-2})$ −	$7.3630 \times 10^{-1}\ (7.71 \times 10^{-2})$ −	**$8.9386 \times 10^{-1}\ (1.23 \times 10^{-2})$**
	15	24	$9.1743 \times 10^{-1}\ (6.21 \times 10^{-3})$ −	$1.8712 \times 10^{-1}\ (5.43 \times 10^{-2})$ −	$7.2178 \times 10^{-1}\ (5.02 \times 10^{-2})$ −	$9.0189 \times 10^{-1}\ (6.90 \times 10^{-3})$ −	$8.7673 \times 10^{-1}\ (8.16 \times 10^{-3})$ −	$8.6506 \times 10^{-1}\ (2.08 \times 10^{-2})$ −	$8.9671 \times 10^{-1}\ (6.03 \times 10^{-3})$ −	**$9.3230 \times 10^{-1}\ (2.65 \times 10^{-3})$**
WFG9	5	14	$7.1726 \times 10^{-1}\ (4.39 \times 10^{-2})$ −	$5.7438 \times 10^{-1}\ (1.18 \times 10^{-1})$ −	$6.9840 \times 10^{-1}\ (6.08 \times 10^{-3})$ −	$7.4910 \times 10^{-1}\ (4.64 \times 10^{-3})$ ≈	$7.2015 \times 10^{-1}\ (7.50 \times 10^{-3})$ −	$7.4269 \times 10^{-1}\ (4.35 \times 10^{-3})$ −	$7.3180 \times 10^{-1}\ (6.41 \times 10^{-3})$ −	**$7.5112 \times 10^{-1}\ (6.66 \times 10^{-3})$**
	10	19	$8.4119 \times 10^{-1}\ (5.81 \times 10^{-2})$ −	$6.9320 \times 10^{-1}\ (8.26 \times 10^{-2})$ −	$7.9149 \times 10^{-1}\ (1.22 \times 10^{-2})$ −	$8.6672 \times 10^{-1}\ (3.58 \times 10^{-2})$ −	$7.9195 \times 10^{-1}\ (5.95 \times 10^{-2})$ −	$8.5381 \times 10^{-1}\ (8.84 \times 10^{-3})$ −	$8.9091 \times 10^{-1}\ (3.76 \times 10^{-2})$ −	**$9.1675 \times 10^{-1}\ (8.03 \times 10^{-3})$**
	15	24	$8.9521 \times 10^{-1}\ (1.22 \times 10^{-2})$ −	$6.0997 \times 10^{-1}\ (2.11 \times 10^{-1})$ −	$8.1071 \times 10^{-1}\ (1.60 \times 10^{-2})$ −	$8.8163 \times 10^{-1}\ (1.14 \times 10^{-2})$ −	$7.9180 \times 10^{-1}\ (1.41 \times 10^{-2})$ −	$8.4359 \times 10^{-1}\ (3.81 \times 10^{-2})$ −	$8.9479 \times 10^{-1}\ (4.57 \times 10^{-2})$ −	**$9.2016 \times 10^{-1}\ (4.61 \times 10^{-2})$**
+/−/≈			7/17/3	0/24/3	0/25/2	7/14/6	4/20/3	1/20/6	6/15/6	



(**a**) DTLZ (**b**) MaF (**c**) WFG

Figure 5. Performance score of all algorithms on different test suites.



(**a**) ANSGAIII (**b**) MaOEAIGD (**c**) DEAGNG (**d**) LMPFE

(**e**) TSDGPD (**f**) RVEAiGNG (**g**) MultiGPO (**h**) GPDARVC

Figure 6. Plot of results for different algorithms on 15-objective DTLZ5.

4.2.2. Comparison Results on MaF Test Problems

The mean and standard deviation of the IGD+ and HV results obtained by all algorithms on the MaF test problems are given in Tables 3 and 4, respectively. Regarding IGD+, GPDARVC outperforms ANSGA-III, MaOEA-IGD, DEAGNG, LMPFE, TS-DGPD, RVEAiGNG, and MultiGPO by 27, 27, 25, 21, 23, 16, and 12 times over the 30 test problems. In terms of HV, GPDARVC beats ANSGA-III, MaOEA-IGD, DEAGNG, LMPFE, TS-DGPD, RVEAiGNG, and MultiGPO on 17, 26, 18, 21, 12, 15, and 11 occasions out of 30 problems. It is not difficult to find that these MaOEAs using reference vectors, like ANSGA-III and MaOEA-IGD, perform poorly on this test suite. This is because the MaF suite has such complex features that the uniformly distributed reference points cannot correctly match the true PF of the MaF suite. In contrast, our proposed GDPARVC performs well on different test problems, and this is because the later adjustment of the reference vectors can correctly guide the algorithms to explore the promising regions that were not explored earlier. This compensates for the lack of early reference vector exploration. The experimental results indicate that the overall performance of GPDARVC is significantly better than the other compared MaOEAs.

Figure 5b shows that GPDARVC achieves the lowest scores in both HV and IGD+, signifying that GPDARVC consistently ranked first. MultioGPO and TSDGPD also perform well, likely because they both utilize the GPD strategy to help them maintain convergence in high-dimensional objective space. In contrast, GPDARVC incorporates the GPD strategy with adjusted RVs, enabling it to balance convergence and diversity well for high-dimensional multi-objective problems, which explains its superiority. Although GPDARVC achieves good results on the MaF test suite, it falls slightly short compared to the DTLZ on the above test suite. Aside from the high complexity of the MaF test suite, another reason for this may be that for some complex problems, the adjusted reference vectors in the later stages do not provide comprehensive exploration. To ensure an intuitive understanding of GPDARVC, Figure 7 shows the final results of GPDARVC running against other competing algorithms on MaF1 with three objectives. Obviously, compared with other algorithms, GPDARVC has well-distributed results and closely approximates the true PF.

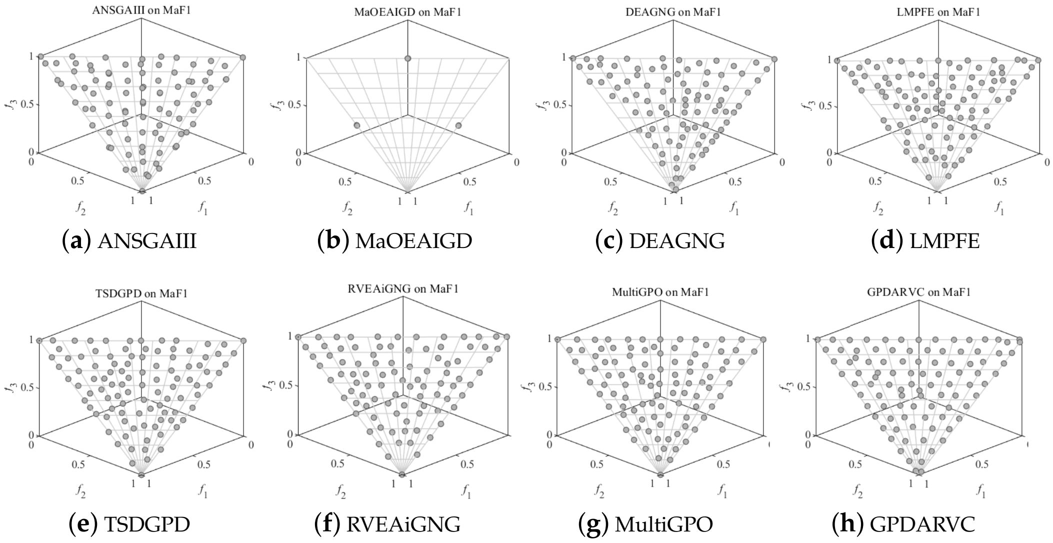


Figure 7. Plot of results for different algorithms on three-objective MaF1.

4.2.3. Comparison Results on WFG Test Problems

The IGD+ and HV values for all algorithms on the WFG test suite are given in Tables 5 and 6, respectively. As shown, GPDARVC significantly outperform its competitors on most cases. WFG 1-3 has irregular PFs, i.e., mixed PFs, disconnected PFs, and degenerate PFs, which pose a significant challenge to algorithms. Even so, GPDARVC still achieves good results in most test instances. For WFGs 4-9, they have the same concave PFs, but their properties in the decision space are entirely different. WFG 4 and WFG 5 tend to make the algorithms susceptible to local optimality due to their multi-peak and deceptive nature, respectively. Nevertheless, GPDARVC still achieves the best IGD+ and HV results in most test instances. The fundamental reason for this is that the distribution of the reference points is consistent with that of the real PFs, where reference vector bootstrapping plays a key role. For WFG 6, whose decision variables are non-separable, GPDARVC has a clear advantage over its competitors. WFG 7-9 tests the algorithm's ability to maintain individuals with good diversity due to its different bias properties. As seen from the HV and IGD+ results, GPDARVC shows the best performance in almost all test instances, which is mainly due to the adjustment of RVs in the later stage, guiding individuals to search the regions not explored before.

Figure 5c presents a bar chart of the HV and IGD+ performance scores achieved by each MaOEA on the WFG test suite, which shows that GPDARVC secured the top results in terms of both HV and IGD+. In particular, the scores of GPDARVC are significantly lower than those of the leading algorithms, such as MultiGPO and ANSGA-III. All of these results demonstrate that GPDARVC outperforms the other competing algorithms on the WFG test suite. Figure 8 plots the results of different algorithms on the 10-objective WFG5 problem.

4.2.4. Comparison Results on Real-World Problems

This section discusses six practical multi-objective engineering problems to validate GPDARVC's effectiveness further. The six practical multi-objective engineering problems are the car side impact design problem, liquid-rocket single-element injector design, location of a pollution monitoring system, the machining problem, the single-pass work roll cooling design problem, and the development of water- and oil-repellent fabric. We name each of these six real multi-objective engineering problems Ma_RW1-6 for convenience. Note that the design of this paper's six real multi-objective engineering problems can be found in [59,60].

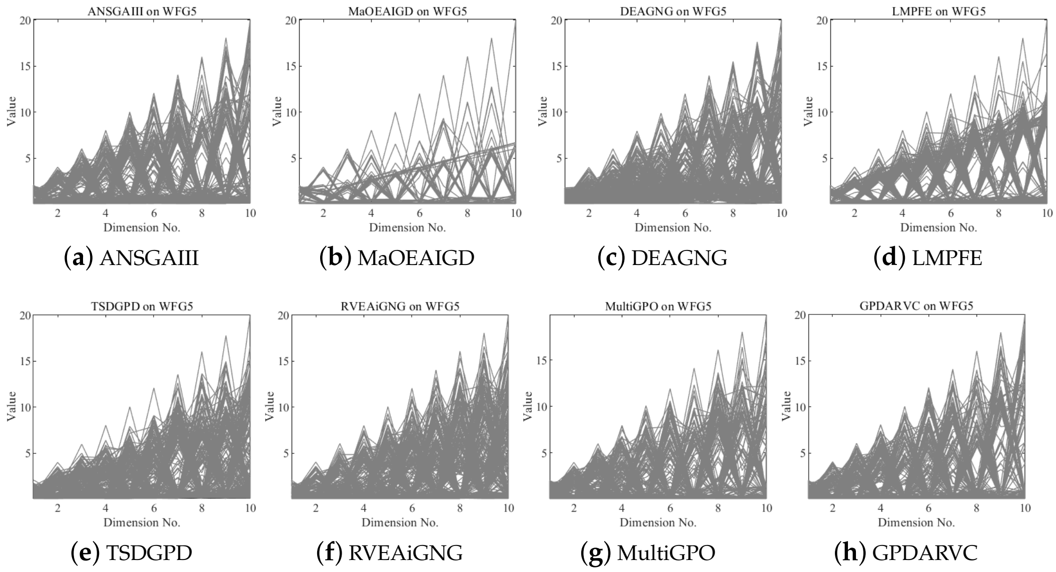


Figure 8. Plot of results for different algorithms on 10-objective WFG5.

GPDARVC is compared with seven algorithms, ANSGA-III, MaOEA-IGD, DEAGNG, LMPFE, TS-DGPD, RVEAiGNG, and MultiGPO, regarding HV. The parameters of the algorithms are set as follows: the population size is set to 200, and the maximum number of fitness evaluations for each algorithm is set to $M\times$ 10,000. To compute HV, the maximum and minimum objective values obtained from the final set of solutions are used to normalize the objective values of all the solutions, followed by the reference point $(1, 1, ..., 1, 1)^T$ to compute the HV.

The HV results of GPDARVC and other competing algorithms for six real-world multi-objective engineering problems are given in Table 7, from which it can be seen that GPDARVC achieves optimal HV results for two real-world problems. GPDARVC performs better than or equal to ANSGA-III, MaOEA-IGD, DEAGNG, LMPFE, TS-DGPD, RVEAiGNG, and MultiGPO in six real-world problems on 4, 5, 4, 3, 4, 4, and 6 occasions. This indicates that GPDARVC shows good potential for future practical multi-objective engineering problems.

4.3. Validation of Proposed Strategies

4.3.1. Validation of Cooperative GPD and RV Strategy

In this subsection, we aim to verify the effectiveness of the cooperative GPD and RV strategy. This strategy first makes a preliminary ranking of the solutions through an (M-1)-GPD dominance framework, and then the environment selection is accomplished through reference vector guidance with the max–min strategy. We mainly verify the effectiveness of embedding this strategy into the developed framework (M-1)-GPD, as (M-1)-GPD was first proposed in [21]. We denote the GPD and RV cooperative framework MultiGPO-RV and compare it with MultiGPO. To ensure fairness, we verify the performance of MultiGPO with MultiGPO-RV by using the same evolutionary operators and keeping other parameters consistent.

We test MultiGPO and MultiGPO_RV on DTLZ, MaF, and WFG with 5 and 10 objectives. Table 7 shows the HV and IGD+ results of MultiGPO and MultiGPO_RV on the DTLZ, MaF, and WFG test suites, respectively. For HV, MultiGPO_RV performed better in 23 out of 52 test problems and worse in 14. In terms of IGD+, MultiGPO_RV achieved better results in 28 out of 52 test problems and worse results in 10. It can be seen that the performance of MultiGPO-RV is better than that of MultiGPO on most cases, which validates the effectiveness and competitiveness of the cooperative GPD and RV strategy.

Table 7. HV values obtained by GPDARVC and other comparison algorithms on real-world problems. The best result for each test instance is shown with dark background.

Problem	M	D	ANSGAIII	MaOEAIGD	DEAGNG	LMPFE	TSDGPD	RVEAiGNG	MultiGPO	GPDARVC
Ma_RW1	4	7	3.0062×10^{-2} (9.12×10^{-4}) −	9.8681×10^{-3} (5.55×10^{-4}) −	2.5086×10^{-2} (1.93×10^{-3}) −	3.0006×10^{-2} (9.72×10^{-4}) −	3.2535×10^{-2} (3.36×10^{-4}) −	3.0318×10^{-2} (9.07×10^{-4}) −	3.2168×10^{-2} (2.14×10^{-3}) −	3.3675×10^{-2} (1.88×10^{-3})
Ma_RW2	4	4	3.7618×10^{-1} (3.15×10^{-3}) +	2.4753×10^{-2} (3.05×10^{-2}) −	3.8277×10^{-1} (6.55×10^{-3}) +	3.8857×10^{-1} (6.32×10^{-3}) +	3.1537×10^{-1} (2.60×10^{-3}) −	3.8650×10^{-1} (1.93×10^{-3}) +	3.6983×10^{-1} (4.73×10^{-3}) ≈	3.6840×10^{-1} (2.76×10^{-3})
Ma_RW3	5	2	4.8459×10^{-3} (8.26×10^{-3}) +	8.0463×10^{-2} (1.14×10^{-1}) +	3.4897×10^{-3} (5.40×10^{-3}) +	7.3549×10^{-3} (6.47×10^{-3}) +	6.5190×10^{-3} (6.71×10^{-3}) +	2.8039×10^{-2} (3.45×10^{-2}) +	3.3680×10^{-4} (5.15×10^{-5}) ≈	4.2021×10^{-4} (2.48×10^{-4})
Ma_RW4	5	3	3.1830×10^{-1} (8.96×10^{-3}) −	3.9165×10^{-2} (3.70×10^{-3}) −	3.2927×10^{-1} (8.65×10^{-3}) −	3.3882×10^{-1} (5.37×10^{-3}) −	3.4034×10^{-1} (4.87×10^{-3}) −	3.3191×10^{-1} (1.11×10^{-3}) −	3.4164×10^{-1} (5.49×10^{-3}) −	3.4504×10^{-1} (4.81×10^{-3})
Ma_RW5	6	7	3.2322×10^{-2} (1.62×10^{-3}) −	4.5537×10^{-3} (1.06×10^{-3}) −	3.0394×10^{-2} (4.77×10^{-3}) −	4.0883×10^{-2} (9.97×10^{-4}) +	4.2664×10^{-2} (3.20×10^{-4}) +	3.1140×10^{-2} (1.71×10^{-3}) −	3.1904×10^{-2} (1.83×10^{-3}) −	3.5359×10^{-2} (9.88×10^{-4})
Ma_RW6	7	3	1.1146×10^{-2} (1.31×10^{-3}) −	7.2504×10^{-6} (1.36×10^{-7}) −	7.8898×10^{-3} (1.24×10^{-3}) −	1.1674×10^{-2} (1.27×10^{-3}) −	1.2924×10^{-2} (1.27×10^{-3}) ≈	1.2808×10^{-2} (8.85×10^{-4}) ≈	1.2569×10^{-2} (7.63×10^{-4}) ≈	1.2942×10^{-2} (6.15×10^{-4})
+/−/≈			2/4/0	1/5/0	2/4/0	3/3/0	2/3/1	2/3/1	0/2/4	

4.3.2. Validation of Adjusted Reference Vector Selection

To verify the effectiveness of reference vector adjustment in our algorithm, we compare the algorithm with static RVs, denoted MultiGPO-SRV, with the algorithm with adjusted RVs, called MultiGPO-ARV. Other operations and parameter settings are the same for MultiGPO-RV and MultiGPO-ARV. From the experimental results shown ins Tables 8 and 9, it can be seen that the performance of MultiGPO-ARV is superior to that of MultiGPO-SRV. For HV, MultiGPO_ARV performed better in 28 out of 52 test problems and worse in 5. In terms of IGD+, MultiGPO_ARV achieved better results in 25 out of 52 test problems and worse results in 7. Both the HV and IGD+ values of MultiGPO_ARV are better than those of MultiGPO-SRV for most of the test problems. This validates that the adjusted RV in the later stage plays a very significant role in the final performance.

Table 8. Comparative results of MultiGPO and MultiGPO_RV on various problems. For each pair of comparison, the best result for each test instance is shown with dark background.

Problem	M	D	HV		IGD+	
			MultiGPO_RV	**MultiGPO**	**MultiGPO_RV**	**MultiGPO**
DTLZ1	5	9	9.7957×10^{-1} (4.26×10^{-4}) +	9.7613×10^{-1} (5.21×10^{-4})	3.7875×10^{-2} (7.35×10^{-4}) −	3.7186×10^{-2} (3.94×10^{-4})
	10	14	9.9968×10^{-1} (5.99×10^{-5}) +	9.9897×10^{-1} (6.31×10^{-4})	8.2646×10^{-2} (9.27×10^{-4}) −	7.3197×10^{-2} (1.64×10^{-3})
DTLZ2	5	14	8.1147×10^{-1} (4.91×10^{-4}) +	8.0238×10^{-1} (1.56×10^{-3})	6.4032×10^{-2} (1.56×10^{-4}) +	7.2676×10^{-2} (9.48×10^{-4})
	10	19	9.7023×10^{-1} (2.36×10^{-4}) +	9.5936×10^{-1} (1.09×10^{-3})	1.7666×10^{-1} (3.08×10^{-4}) +	1.7985×10^{-1} (2.26×10^{-3})
DTLZ3	5	14	7.9170×10^{-1} (9.61×10^{-3}) −	7.9903×10^{-1} (5.47×10^{-3})	7.9411×10^{-2} (7.69×10^{-3}) ≈	7.5985×10^{-2} (4.31×10^{-3})
	10	19	9.6339×10^{-1} (6.88×10^{-3}) +	9.3478×10^{-1} (2.28×10^{-2})	1.8541×10^{-1} (7.87×10^{-3}) +	2.1157×10^{-1} (2.17×10^{-2})
DTLZ4	5	14	8.1151×10^{-1} (2.92×10^{-4}) +	8.0448×10^{-1} (1.66×10^{-3})	6.3928×10^{-2} (1.53×10^{-4}) +	7.1703×10^{-2} (9.50×10^{-4})
	10	19	9.7021×10^{-1} (2.63×10^{-4}) +	9.6239×10^{-1} (8.93×10^{-4})	1.7579×10^{-1} (4.26×10^{-4}) +	1.7873×10^{-1} (1.97×10^{-3})
DTLZ5	5	14	1.0449×10^{-1} (4.85×10^{-3}) ≈	1.0154×10^{-1} (4.95×10^{-3})	4.3883×10^{-2} (7.30×10^{-3}) +	5.0725×10^{-2} (1.18×10^{-2})
	10	19	8.7696×10^{-2} (2.12×10^{-3}) ≈	8.6196×10^{-2} (3.13×10^{-3})	8.6392×10^{-2} (1.95×10^{-2}) +	1.0025×10^{-1} (1.87×10^{-2})
DTLZ6	5	14	1.0290×10^{-1} (5.19×10^{-3}) +	9.7489×10^{-2} (6.38×10^{-3})	4.9008×10^{-2} (1.04×10^{-2}) +	6.7124×10^{-2} (1.76×10^{-2})
	10	19	9.0892×10^{-2} (2.61×10^{-4}) ≈	8.9286×10^{-2} (7.10×10^{-3})	1.0316×10^{-1} (2.83×10^{-2}) ≈	1.0211×10^{-1} (2.30×10^{-2})
DTLZ7	5	24	2.5228×10^{-1} (6.37×10^{-3}) ≈	2.5318×10^{-1} (3.91×10^{-3})	2.0991×10^{-1} (1.41×10^{-1}) ≈	1.4859×10^{-1} (2.83×10^{-2})
	10	29	1.2967×10^{-1} (1.59×10^{-2}) ≈	1.1874×10^{-1} (2.20×10^{-2})	6.7916×10^{-1} (8.51×10^{-3}) +	7.2405×10^{-1} (7.40×10^{-3})
MaF1	5	14	1.1724×10^{-2} (2.69×10^{-4}) ≈	1.1759×10^{-2} (2.58×10^{-4})	8.0030×10^{-2} (1.75×10^{-3}) ≈	7.9782×10^{-2} (1.83×10^{-3})
	10	19	2.4953×10^{-7} (4.43×10^{-7}) ≈	3.9699×10^{-7} (7.47×10^{-7})	1.6657×10^{-1} (8.11×10^{-4}) −	1.6563×10^{-1} (1.21×10^{-3})
MaF2	5	14	1.9181×10^{-1} (1.81×10^{-3}) +	1.8752×10^{-1} (2.21×10^{-3})	5.2287×10^{-2} (1.12×10^{-3}) +	5.5021×10^{-2} (9.87×10^{-4})
	10	19	2.2185×10^{-1} (2.93×10^{-3}) +	2.0829×10^{-1} (3.53×10^{-3})	1.1269×10^{-1} (4.54×10^{-3}) +	1.1747×10^{-1} (4.81×10^{-3})
MaF3	5	14	9.9617×10^{-1} (2.20×10^{-3}) −	9.9661×10^{-1} (2.73×10^{-3})	3.2517×10^{-2} (1.82×10^{-2}) +	3.7722×10^{-2} (1.24×10^{-2})
	10	19	9.9910×10^{-1} (7.12×10^{-4}) ≈	9.9902×10^{-1} (7.40×10^{-4})	2.1471×10^{-2} (2.12×10^{-3}) +	2.8817×10^{-2} (4.44×10^{-3})
MaF4	5	14	1.1050×10^{-1} (5.30×10^{-3}) −	1.1445×10^{-1} (3.81×10^{-3})	6.5158×10^{-1} (4.85×10^{-2}) −	6.1668×10^{-1} (3.32×10^{-2})
	10	19	7.1109×10^{-5} (2.01×10^{-5}) +	5.9247×10^{-5} (1.40×10^{-5})	9.2043×10^{0} (8.02×10^{-1}) ≈	9.0576×10^{0} (4.20×10^{-1})
MaF5	5	14	7.6460×10^{-1} (2.49×10^{-2}) −	7.7892×10^{-1} (5.41×10^{-3})	4.5095×10^{-1} (5.02×10^{-2}) −	4.2646×10^{-1} (1.49×10^{-2})
	10	19	8.3359×10^{-1} (3.66×10^{-3}) ≈	8.3378×10^{-1} (4.24×10^{-3})	1.2346×10^{0} (1.26×10^{-2}) ≈	1.2289×10^{0} (8.87×10^{-3})
MaF6	5	14	1.2950×10^{-1} (4.39×10^{-4}) ≈	1.2964×10^{-1} (3.31×10^{-4})	1.1968×10^{-3} (6.96×10^{-5}) −	1.1450×10^{-3} (7.02×10^{-5})
	10	19	7.2304×10^{-2} (3.72×10^{-2}) ≈	6.8132×10^{-2} (4.00×10^{-2})	7.7691×10^{-2} (1.02×10^{-1}) ≈	8.5688×10^{-2} (1.06×10^{-1})
MaF7	5	24	2.5643×10^{-1} (2.73×10^{-3}) +	2.5444×10^{-1} (3.21×10^{-3})	1.5877×10^{-1} (3.95×10^{-2}) ≈	1.4972×10^{-1} (3.28×10^{-2})
	10	29	1.3458×10^{-1} (1.66×10^{-2}) +	1.1235×10^{-1} (2.10×10^{-2})	6.7943×10^{-1} (5.75×10^{-3}) +	7.2159×10^{-1} (1.30×10^{-2})
MaF8	5	2	1.2478×10^{-1} (4.67×10^{-3}) ≈	1.2584×10^{-1} (3.65×10^{-4})	5.3530×10^{-2} (2.84×10^{-2}) ≈	4.6910×10^{-2} (7.83×10^{-4})
	10	2	1.0978×10^{-2} (9.92×10^{-5}) ≈	1.1005×10^{-2} (8.95×10^{-5})	6.3002×10^{-2} (7.59×10^{-4}) ≈	6.2770×10^{-2} (8.61×10^{-4})
MaF9	5	2	3.2424×10^{-1} (2.67×10^{-3}) −	3.2532×10^{-1} (7.10×10^{-4})	5.3077×10^{-2} (4.51×10^{-3}) ≈	5.1864×10^{-2} (4.82×10^{-4})
	10	2	1.8569×10^{-2} (1.48×10^{-4}) ≈	1.8576×10^{-2} (1.20×10^{-4})	7.2281×10^{-2} (3.44×10^{-4}) ≈	7.2169×10^{-2} (3.68×10^{-4})
MaF10	5	14	5.9066×10^{-1} (7.86×10^{-2}) −	9.9565×10^{-1} (2.68×10^{-3})	9.8594×10^{-1} (2.10×10^{-1}) −	1.2150×10^{-1} (1.30×10^{-2})
	10	19	7.2468×10^{-1} (7.59×10^{-2}) −	9.9835×10^{-1} (7.74×10^{-4})	8.5126×10^{-1} (1.97×10^{-1}) −	2.2751×10^{-1} (7.09×10^{-2})
WFG1	5	14	9.7518×10^{-1} (3.08×10^{-2}) −	9.9357×10^{-1} (1.06×10^{-2})	1.4498×10^{-1} (4.70×10^{-2}) ≈	1.2037×10^{-1} (1.68×10^{-2})
	10	19	9.9740×10^{-1} (6.59×10^{-4}) −	9.9846×10^{-1} (6.07×10^{-4})	1.8820×10^{-1} (1.36×10^{-2}) ≈	2.3283×10^{-1} (6.33×10^{-2})

Table 8. *Cont.*

Problem	M	D	HV		IGD+	
			MultiGPO_RV	MultiGPO	MultiGPO_RV	MultiGPO
WFG2	5	14	9.8416×10^{-1} (2.26×10^{-3}) −	9.8536×10^{-1} (2.11×10^{-3})	1.0870×10^{-1} (3.80×10^{-3}) +	1.1988×10^{-1} (4.92×10^{-3})
	10	19	9.8845×10^{-1} (2.30×10^{-3}) −	9.9307×10^{-1} (2.10×10^{-3})	1.7426×10^{-1} (1.03×10^{-2}) +	2.4792×10^{-1} (1.31×10^{-2})
WFG3	5	14	1.3404×10^{-1} (1.65×10^{-2}) +	3.3277×10^{-2} (2.60×10^{-2})	3.7931×10^{-1} (4.14×10^{-2}) +	6.0440×10^{-1} (1.31×10^{-1})
	10	19	0.0000×10^{0} (0.00×10^{0}) ≈	0.0000×10^{0} (0.00×10^{0})	1.3479×10^{0} (1.20×10^{-1}) +	2.0015×10^{0} (2.42×10^{-1})
WFG4	5	14	7.8988×10^{-1} (3.82×10^{-3}) +	7.6398×10^{-1} (5.03×10^{-3})	3.2773×10^{-1} (6.20×10^{-3}) +	3.7270×10^{-1} (8.00×10^{-3})
	10	19	9.4417×10^{-1} (8.64×10^{-3}) −	9.5702×10^{-1} (2.44×10^{-3})	1.0054×10^{0} (3.41×10^{-2}) −	9.7903×10^{-1} (1.57×10^{-2})
WFG5	5	14	7.5198×10^{-1} (1.61×10^{-3}) +	7.2390×10^{-1} (4.27×10^{-3})	3.7129×10^{-1} (3.55×10^{-3}) +	4.4547×10^{-1} (6.82×10^{-3})
	10	19	8.9827×10^{-1} (1.38×10^{-3}) +	8.8579×10^{-1} (2.02×10^{-3})	1.0346×10^{0} (1.46×10^{-2}) +	1.2222×10^{0} (4.16×10^{-2})
WFG6	5	14	7.2172×10^{-1} (1.60×10^{-2}) +	7.0659×10^{-1} (1.33×10^{-2})	4.1960×10^{-1} (2.33×10^{-2}) +	4.9629×10^{-1} (2.58×10^{-2})
	10	19	8.5851×10^{-1} (1.47×10^{-2}) −	8.8029×10^{-1} (1.79×10^{-2})	1.1139×10^{0} (2.98×10^{-2}) +	1.1484×10^{0} (5.35×10^{-2})
WFG7	5	14	7.9764×10^{-1} (3.30×10^{-3}) +	7.8435×10^{-1} (3.56×10^{-3})	3.1749×10^{-1} (5.84×10^{-3}) +	3.4372×10^{-1} (6.61×10^{-3})
	10	19	9.5475×10^{-1} (5.47×10^{-3}) −	9.6340×10^{-1} (2.77×10^{-3})	9.8868×10^{-1} (2.40×10^{-2}) −	9.6213×10^{-1} (1.46×10^{-2})
WFG8	5	14	6.8800×10^{-1} (2.04×10^{-3}) +	6.3234×10^{-1} (6.33×10^{-3})	5.9836×10^{-1} (3.20×10^{-3}) +	7.0094×10^{-1} (1.09×10^{-2})
	10	19	8.7396×10^{-1} (1.40×10^{-2}) +	7.3630×10^{-1} (7.71×10^{-2})	1.6441×10^{0} (2.66×10^{-1}) +	2.2224×10^{0} (4.60×10^{-1})
WFG9	5	14	7.5271×10^{-1} (5.71×10^{-3}) +	7.3180×10^{-1} (6.41×10^{-3})	3.7297×10^{-1} (8.77×10^{-3}) +	4.2105×10^{-1} (1.12×10^{-2})
	10	19	9.0123×10^{-1} (4.14×10^{-2}) +	8.9091×10^{-1} (3.76×10^{-2})	1.1322×10^{0} (8.26×10^{-2}) +	1.2111×10^{0} (4.62×10^{-2})
+/−/≈			23/14/15		28/10/14	

Table 9. Comparative results of MultiGPO_SRV and MultiGPO_ARV on various problems. For each pair of comparison, the best result for each test instance is shown with dark background.

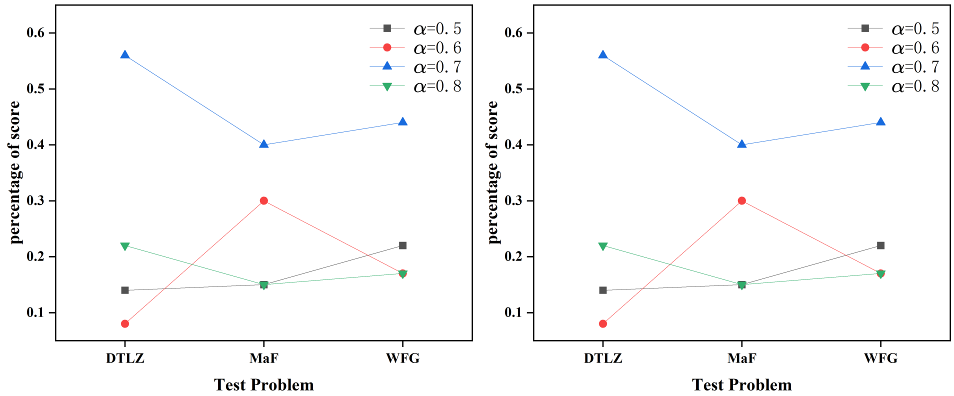
Problem	M	D	HV		IGD+	
			MultiGPO_ARV	MultiGPO_SRV	MultiGPO_ARV	MultiGPO_SRV
DTLZ1	5	9	9.7983×10^{-1} (5.83×10^{-4}) +	9.7957×10^{-1} (4.26×10^{-4})	3.7288×10^{-2} (3.53×10^{-4}) +	3.7875×10^{-2} (7.35×10^{-4})
	10	14	9.9973×10^{-1} (4.84×10^{-5}) +	9.9968×10^{-1} (5.99×10^{-5})	7.8383×10^{-2} (1.87×10^{-3}) +	8.2646×10^{-2} (9.27×10^{-4})
DTLZ2	5	14	8.1113×10^{-1} (5.40×10^{-4}) −	8.1147×10^{-1} (4.91×10^{-4})	6.4962×10^{-2} (3.76×10^{-4}) −	6.4032×10^{-2} (1.56×10^{-4})
	10	19	9.7172×10^{-1} (1.84×10^{-4}) +	9.7023×10^{-1} (2.36×10^{-4})	1.7059×10^{-1} (4.57×10^{-4}) +	1.7666×10^{-1} (3.08×10^{-4})
DTLZ3	5	14	8.0451×10^{-1} (4.55×10^{-3}) +	7.9170×10^{-1} (9.61×10^{-3})	7.0722×10^{-2} (3.81×10^{-3}) +	7.9411×10^{-2} (7.69×10^{-3})
	10	19	9.6892×10^{-1} (1.46×10^{-3}) +	9.6339×10^{-1} (6.88×10^{-3})	1.7589×10^{-1} (3.25×10^{-3}) +	1.8541×10^{-1} (7.87×10^{-3})
DTLZ4	5	14	8.1079×10^{-1} (6.75×10^{-4}) −	8.1151×10^{-1} (2.92×10^{-4})	6.5065×10^{-2} (3.51×10^{-4}) −	6.3928×10^{-2} (1.53×10^{-4})
	10	19	9.7192×10^{-1} (2.17×10^{-4}) +	9.7021×10^{-1} (2.63×10^{-4})	1.7000×10^{-1} (4.38×10^{-4}) +	1.7579×10^{-1} (4.26×10^{-4})
DTLZ5	5	14	1.0491×10^{-1} (4.70×10^{-3}) ≈	1.0449×10^{-1} (4.85×10^{-3})	4.2347×10^{-2} (8.12×10^{-3}) ≈	4.3883×10^{-2} (7.30×10^{-3})
	10	19	8.9374×10^{-2} (8.76×10^{-4}) +	8.7696×10^{-2} (2.12×10^{-3})	8.1812×10^{-2} (1.90×10^{-2}) ≈	8.6392×10^{-2} (1.95×10^{-2})
DTLZ6	5	14	1.0093×10^{-1} (6.61×10^{-3}) ≈	1.0290×10^{-1} (5.19×10^{-3})	5.0886×10^{-2} (1.35×10^{-2}) ≈	4.9008×10^{-2} (1.04×10^{-2})
	10	19	9.0887×10^{-2} (1.94×10^{-4}) ≈	9.0892×10^{-2} (2.61×10^{-4})	8.0732×10^{-2} (1.36×10^{-2}) +	1.0316×10^{-1} (2.83×10^{-2})
DTLZ7	5	24	2.6175×10^{-1} (2.61×10^{-3}) +	2.5228×10^{-1} (6.37×10^{-3})	1.2910×10^{-1} (3.47×10^{-3}) +	2.0991×10^{-1} (1.41×10^{-1})
	10	29	1.3533×10^{-1} (1.22×10^{-2}) ≈	1.2967×10^{-1} (1.59×10^{-2})	6.7953×10^{-1} (8.44×10^{-3}) ≈	6.7916×10^{-1} (8.51×10^{-3})
MaF1	5	14	1.1310×10^{-2} (2.42×10^{-4}) −	1.1724×10^{-2} (2.69×10^{-4})	8.3015×10^{-2} (1.78×10^{-3}) −	8.0030×10^{-2} (1.75×10^{-3})
	10	19	5.3666×10^{-7} (8.60×10^{-7}) ≈	2.4953×10^{-7} (4.43×10^{-7})	1.6473×10^{-1} (9.92×10^{-4}) +	1.6657×10^{-1} (8.11×10^{-4})
MaF2	5	14	1.9223×10^{-1} (2.51×10^{-3}) ≈	1.9181×10^{-1} (1.81×10^{-3})	5.1897×10^{-2} (5.97×10^{-4}) ≈	5.2287×10^{-2} (1.12×10^{-3})
	10	19	2.2006×10^{-1} (3.13×10^{-3}) ≈	2.2185×10^{-1} (2.93×10^{-3})	1.1473×10^{-1} (5.72×10^{-3}) ≈	1.1269×10^{-1} (4.54×10^{-3})
MaF3	5	14	9.9758×10^{-1} (7.20×10^{-4}) +	9.9617×10^{-1} (2.20×10^{-3})	2.5824×10^{-2} (1.93×10^{-3}) ≈	3.2517×10^{-2} (1.82×10^{-2})
	10	19	9.9962×10^{-1} (1.43×10^{-5}) +	9.9910×10^{-1} (7.12×10^{-4})	2.2447×10^{-2} (3.56×10^{-18}) ≈	2.1471×10^{-2} (2.12×10^{-3})
MaF4	5	14	1.1694×10^{-1} (2.68×10^{-3}) +	1.1050×10^{-1} (5.30×10^{-3})	5.9851×10^{-1} (3.03×10^{-2}) +	6.5158×10^{-1} (4.85×10^{-2})
	10	19	5.4507×10^{-5} (3.97×10^{-6}) −	7.1109×10^{-5} (2.01×10^{-5})	9.4958×10^{0} (1.82×10^{-15}) −	9.2043×10^{0} (8.02×10^{-1})
MaF5	5	14	7.6540×10^{-1} (1.19×10^{-2}) ≈	7.6460×10^{-1} (2.49×10^{-2})	4.5367×10^{-1} (3.03×10^{-2}) ≈	4.5095×10^{-1} (5.02×10^{-2})
	10	19	8.3647×10^{-1} (3.10×10^{-4}) +	8.3359×10^{-1} (3.66×10^{-3})	1.2149×10^{0} (2.28×10^{-16}) +	1.2346×10^{0} (1.26×10^{-2})
MaF6	5	14	1.2973×10^{-1} (4.61×10^{-4}) ≈	1.2950×10^{-1} (4.39×10^{-4})	1.1069×10^{-3} (5.97×10^{-5}) +	1.1968×10^{-3} (6.96×10^{-5})
	10	19	4.5524×10^{-2} (2.62×10^{-2}) −	7.2304×10^{-2} (3.72×10^{-2})	1.3317×10^{-1} (3.24×10^{-2}) −	7.7691×10^{-2} (1.02×10^{-1})

Table 9. *Cont.*

Problem	M	D	HV		IGD+	
			MultiGPO_ARV	MultiGPO_SRV	MultiGPO_ARV	MultiGPO_SRV
MaF7	5	24	2.6077×10^{-1} (2.75×10^{-3}) +	2.5643×10^{-1} (2.73×10^{-3})	1.3091×10^{-1} (3.40×10^{-3}) +	1.5877×10^{-1} (3.95×10^{-2})
	10	29	1.5393×10^{-1} (3.05×10^{-4}) +	1.3458×10^{-1} (1.66×10^{-2})	6.7698×10^{-1} (2.28×10^{-16}) ≈	6.7943×10^{-1} (5.75×10^{-3})
MaF8	5	2	1.2592×10^{-1} (4.35×10^{-4}) ≈	1.2478×10^{-1} (4.67×10^{-3})	4.6730×10^{-2} (6.46×10^{-4}) +	5.3530×10^{-2} (2.84×10^{-2})
	10	2	1.1005×10^{-2} (1.15×10^{-4}) ≈	1.0978×10^{-2} (9.92×10^{-5})	6.2620×10^{-2} (0.00×10^{0}) ≈	6.3002×10^{-2} (7.59×10^{-4})
MaF9	5	2	3.2427×10^{-1} (1.01×10^{-3}) ≈	3.2424×10^{-1} (2.67×10^{-3})	5.2483×10^{-2} (6.17×10^{-4}) ≈	5.3077×10^{-2} (4.51×10^{-3})
	10	2	1.8572×10^{-2} (1.21×10^{-4}) ≈	1.8569×10^{-2} (1.48×10^{-4})	7.2283×10^{-2} (4.99×10^{-4}) ≈	7.2281×10^{-2} (3.44×10^{-4})
MaF10	5	14	9.9591×10^{-1} (5.39×10^{-4}) +	5.9066×10^{-1} (7.86×10^{-2})	1.0616×10^{-1} (1.75×10^{-3}) +	9.8594×10^{-1} (2.10×10^{-1})
	10	19	9.9818×10^{-1} (8.12×10^{-4}) +	7.2468×10^{-1} (7.59×10^{-2})	1.7741×10^{-1} (7.35×10^{-3}) +	8.5126×10^{-1} (1.97×10^{-1})
WFG1	5	14	9.9624×10^{-1} (4.40×10^{-4}) +	9.7518×10^{-1} (3.08×10^{-2})	1.0612×10^{-1} (1.51×10^{-3}) +	1.4498×10^{-1} (4.70×10^{-2})
	10	19	9.9805×10^{-1} (7.91×10^{-4}) +	9.9740×10^{-1} (6.59×10^{-4})	1.8100×10^{-1} (5.32×10^{-3}) +	1.8820×10^{-1} (1.36×10^{-2})
WFG2	5	14	9.8743×10^{-1} (1.98×10^{-3}) +	9.8416×10^{-1} (2.26×10^{-3})	1.0821×10^{-1} (3.25×10^{-3}) ≈	1.0870×10^{-1} (3.80×10^{-3})
	10	19	9.9216×10^{-1} (1.81×10^{-3}) +	9.8845×10^{-1} (2.30×10^{-3})	1.8291×10^{-1} (1.39×10^{-2}) −	1.7426×10^{-1} (1.03×10^{-2})
WFG3	5	14	1.3093×10^{-1} (1.75×10^{-2}) ≈	1.3404×10^{-1} (1.65×10^{-2})	4.6603×10^{-1} (5.86×10^{-2}) −	3.7931×10^{-1} (4.14×10^{-2})
	10	19	0.0000×10^{0} (0.00×10^{0}) ≈	0.0000×10^{0} (0.00×10^{0})	1.3803×10^{0} (1.44×10^{-1}) ≈	1.3479×10^{0} (1.20×10^{-1})
WFG4	5	14	7.9304×10^{-1} (3.16×10^{-3}) +	7.8988×10^{-1} (3.82×10^{-3})	3.2184×10^{-1} (6.14×10^{-3}) +	3.2773×10^{-1} (6.20×10^{-3})
	10	19	9.5320×10^{-1} (5.39×10^{-3}) +	9.4417×10^{-1} (8.64×10^{-3})	9.6320×10^{-1} (2.24×10^{-2}) +	1.0054×10^{0} (3.41×10^{-2})
WFG5	5	14	7.5300×10^{-1} (2.97×10^{-3}) +	7.5198×10^{-1} (1.61×10^{-3})	3.6670×10^{-1} (5.91×10^{-3}) +	3.7129×10^{-1} (3.55×10^{-3})
	10	19	9.0141×10^{-1} (1.28×10^{-3}) +	8.9827×10^{-1} (1.38×10^{-3})	1.0257×10^{0} (1.58×10^{-2}) ≈	1.0346×10^{0} (1.46×10^{-2})
WFG6	5	14	7.3125×10^{-1} (1.50×10^{-2}) +	7.2172×10^{-1} (1.60×10^{-2})	4.0251×10^{-1} (2.25×10^{-2}) +	4.1960×10^{-1} (2.33×10^{-2})
	10	19	8.6612×10^{-1} (1.85×10^{-2}) ≈	8.5851×10^{-1} (1.47×10^{-2})	1.0760×10^{0} (3.40×10^{-2}) +	1.1139×10^{0} (2.98×10^{-2})
WFG7	5	14	7.9760×10^{-1} (3.31×10^{-3}) ≈	7.9764×10^{-1} (3.30×10^{-3})	3.1604×10^{-1} (5.65×10^{-3}) ≈	3.1749×10^{-1} (5.84×10^{-3})
	10	19	9.5956×10^{-1} (3.66×10^{-3}) +	9.5475×10^{-1} (5.47×10^{-3})	9.6175×10^{-1} (1.99×10^{-2}) +	9.8868×10^{-1} (2.40×10^{-2})
WFG8	5	14	6.8976×10^{-1} (2.81×10^{-3}) ≈	6.8800×10^{-1} (2.04×10^{-3})	6.0005×10^{-1} (3.45×10^{-3}) ≈	5.9836×10^{-1} (3.20×10^{-3})
	10	19	8.9386×10^{-1} (1.23×10^{-2}) +	8.7396×10^{-1} (1.40×10^{-2})	1.4763×10^{0} (2.26×10^{-1}) +	1.6441×10^{0} (2.66×10^{-1})
WFG9	5	14	7.5112×10^{-1} (6.66×10^{-3}) ≈	7.5271×10^{-1} (5.71×10^{-3})	3.7320×10^{-1} (1.04×10^{-2}) ≈	3.7297×10^{-1} (8.77×10^{-3})
	10	19	9.1675×10^{-1} (8.03×10^{-3}) +	9.0123×10^{-1} (4.14×10^{-2})	1.0912×10^{0} (4.55×10^{-2}) ≈	1.1322×10^{0} (8.26×10^{-2})
+/−/≈			28/5/19		25/7/20	

4.3.3. Parameter Sensitivity Analysis

The purpose of adjusting RVs at a later stage is to improve the search capability of the algorithm so that it can perform well when dealing with more complex problems. The timing of adjusting the RVs needs to be determined, i.e., α in Algorithm 1. This parameter plays a critical role in affecting the algorithm's performance. We have tested different values to investigate the impact on algorithm's performance so as to determine the optimal value. It should be noted that the accuracy of the region explored by our algorithm in the early stage affects the effectiveness of the RV adjustment in the later stage. The algorithm should explore all the promising areas as much as possible in the early stage. That said, we do not suggest starting the adjustment of RVs too early. Therefore, we tested the performance of the algorithm when α = 0.5, 0.6, 0.7, and 0.8. For each different value of α, one point is added whenever they achieve the best performance in a test problem; eventually, we counted the percentage of their scores in each category of test problem and plotted a line graph, where the higher the line, the better the performance of the parameter. As shown in Figure 9, the algorithm performs best when α = 0.7. Hence, we recommend using α = 0.7 in this paper.



(**a**) HV (**b**) IGD+

Figure 9. Performance score of GPDARVC with varying α on different test suites.

5. Conclusions

In this article, we propose a Generalized Pareto Dominance and Reference Vector Cooperative evolutionary algorithm to deal with many-objective optimization problems. A generalized Pareto dominance relation can provide enough selection pressure to enhance convergence, while the guidance of reference vectors approximates the actual PF from different directions, ensuring population diversity. Additionally, we adjust the reference vectors in the later stage of the algorithm, exploring previously uncharted promising regions and thus significantly improving the algorithm's ability to handle complex problems. The cooperation of GPD and RV provides a good balance between convergence and diversity. Compared with state-of-the art algorithms, after conducting comprehensive experiments, we can confirm that GPDARVC shows great performance in most cases.

In future studies, we need to investigate whether adaptively updating reference vectors during the evolution process can further improve the overall performance. Given the performance of GPDARVC in real multi-objective engineering problems, we are interested in extending the application of GPDARVC to more complex engineering problems, such as protein structure prediction [3], many-objective recommendation systems [61] and many-objective mobile edge computing problems [62].

Author Contributions: Conceptualization, S.Z.; methodology, S.Z. and M.C.; software, L.Z.; validation, L.Z.; formal analysis, L.Z. and M.C.; writing—original draft, S.Z.; writing—review and editing, S.Z., L.Z. and M.C.; supervision, M.C.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of Jiangsu Province, China (No. BK20221067 and No. BK20230923), the Natural Science Foundation of China (No. 62206113), and the High-End Foreign Expert Recruitment Plan of China (No. G2023144007L).

Data Availability Statement: Data are contained with the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Cui, M.; Li, L.; Zhou, M.; Li, J.; Abusorrah, A.; Sedraoui, K. A bi-population cooperative optimization algorithm assisted by an autoencoder for medium-scale expensive problems. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1952–1966. [CrossRef]
2. Zhou, M.; Cui, M.; Xu, D.; Zhu, S.; Zhao, Z.; Abusorrah, A. Evolutionary optimization methods for high-dimensional expensive problems: A survey. *IEEE/CAA J. Autom. Sin.* **2024**, *11*, 1092–1105. [CrossRef]
3. Lei, Z.; Gao, S.; Zhang, Z.; Zhou, M.; Cheng, J. MO4: A many-objective evolutionary algorithm for protein structure prediction. *IEEE Trans. Evol. Comput.* **2022**, *26*, 417–430. [CrossRef]

4. Lu, Z.; Sreekumar, G.; Goodman, E.; Banzhaf, W.; Deb, K.; Boddeti, V.N. Neural architecture transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 2971–2989. [CrossRef] [PubMed]
5. Zhu, S.; Sun, N.; Lv, S.; Chen, K.; Fang, W.; Cao, L. Research progress on intelligent optimization techniques for energy-efficient design of ship hull forms. *J. Membr. Comput.* **2024**, 1–17. [CrossRef]
6. Zhu, S.; Xu, L.; Goodman, E.D. Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy. *Knowl.-Based Syst.* **2020**, *188*, 105018. [CrossRef]
7. Zhou, Z.; Zhu, S. Kernel-based multiobjective clustering algorithm with automatic attribute weighting. *Soft Comput.* **2018**, *22*, 3685–3709. [CrossRef]
8. Zhu, S.; Xu, L.; Goodman, E.; Deb, K.; Lu, Z. A general framework for enhancing relaxed Pareto dominance methods in evolutionary many-objective optimization. *Nat. Comput.* **2023**, *22*, 287–313. [CrossRef]
9. 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]
10. Sato, H.; Aguirre, H.E.; Tanaka, K. Controlling dominance area of solutions and its impact on the performance of MOEAs. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Matsushima, Japan, 5–8 March 2007 ; Springer: Berlin/Heidelberg, Germany, 2007; pp. 5–20.
11. Zhu, C.; Xu, L.; Goodman, E.D. Generalization of Pareto-optimality for many-objective evolutionary optimization. *IEEE Trans. Evol. Comput.* **2016**, *20*, 299–315. [CrossRef]
12. Zhang, W.; Liu, J.; Liu, J.; Liu, Y.; Tan, S. A dual distance dominance based evolutionary algorithm with selection-replacement operator for many-objective optimization. *Expert Syst. Appl.* **2024**, *237*, 121244. [CrossRef]
13. Palakonda, V.; Kang, J.M.; Jung, H. An adaptive neighborhood based evolutionary algorithm with pivot-solution based selection for multi-and many-objective optimization. *Inf. Sci.* **2022**, *607*, 126–152. [CrossRef]
14. Li, M.; Yang, S.; Liu, X. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans. Evol. Comput.* **2014**, *18*, 348–365. [CrossRef]
15. Zhang, X.; Tian, Y.; Jin, Y. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2015**, *19*, 761–776. [CrossRef]
16. 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.* **2014**, *18*, 577–601. [CrossRef]
17. Xu, X.; Cheng, D.; Wang, D.; Li, Q.; Yu, F. An Improved NSGA-III with a Comprehensive Adaptive Penalty Scheme for Many-Objective Optimization. *Symmetry* **2024**, *16*, 1289. [CrossRef]
18. Li, K.; Deb, K.; Zhang, Q.; Kwong, S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evol. Comput.* **2015**, *19*, 694–716. [CrossRef]
19. Lin, Q.; Li, J.; Liu, S.; Ma, L.; Li, J.; Chen, J. An adaptive two-stage evolutionary algorithm for large-scale continuous multi-objective optimization. *Swarm Evol. Comput.* **2023**, *77*, 101235. [CrossRef]
20. Wang, X.; Xie, Z.; Zhou, X.; Gu, X. A two-stage adaptive reference direction guided evolutionary algorithm with modified dominance relation for many-objective optimization. *Swarm Evol. Comput.* **2023**, *78*, 101272. [CrossRef]
21. Zhu, S.; Xu, L.; Goodman, E.D.; Lu, Z. A new many-objective evolutionary algorithm based on generalized Pareto dominance. *IEEE Trans. Cybern.* **2022**, *52*, 7776–7790. [CrossRef] [PubMed]
22. Liu, Q.; Jin, Y.; Heiderich, M.; Rodemann, T. Coordinated adaptation of reference vectors and scalarizing functions in evolutionary many-objective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *53*, 763–775. [CrossRef]
23. Li, W.; Chen, Y.; Dong, Y.; Huang, Y. A solution potential-based adaptation reference vector evolutionary algorithm for many-objective optimization. *Swarm Evol. Comput.* **2024**, *84*, 101451. [CrossRef]
24. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
25. Wu, M.; Li, K.; Kwong, S.; Zhang, Q. Evolutionary many-objective optimization based on adversarial decomposition. *IEEE Trans. Cybern.* **2018**, *50*, 753–764. [CrossRef]
26. Liu, S.; Zheng, J.; Lin, Q.; Tan, K.C. Evolutionary multi and many-objective optimization via clustering for environmental selection. *Inf. Sci.* **2021**, *578*, 930–949. [CrossRef]
27. Jiang, S.; Yang, S. A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Trans. Evol. Comput.* **2017**, *21*, 329–346. [CrossRef]
28. Li, X.; Li, X.; Wang, K.; Yang, S. A strength pareto evolutionary algorithm based on adaptive reference points for solving irregular fronts. *Inf. Sci.* **2023**, *626*, 658–693. [CrossRef]
29. Wang, M.; Li, X.; Dai, Y.; Chen, L.; Chen, H.; Ruiz, R. An incremental learning evolutionary algorithm for many-objective optimization with irregular Pareto fronts. *Inf. Sci.* **2023**, *642*, 119115. [CrossRef]
30. Beume, N.; Naujoks, B.; Emmerich, M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **2007**, *181*, 1653–1669. [CrossRef]
31. Bader, J.; Zitzler, E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **2011**, *19*, 45–76. [CrossRef]
32. 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]

33. Chao, T.; Wang, S.; Wang, S.; Yang, M. A many-objective evolutionary algorithm combining simplified hypervolume and a method for reference point sampling based on angular relationship. *Appl. Soft Comput.* **2024**, *163*, 111881. [CrossRef]
34. 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]
35. Shang, K.; Ishibuchi, H. A new hypervolume-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2020**, *24*, 839–852. [CrossRef]
36. Cao, J.; Yang, L.; Li, K.; Zhang, Y.; Tian, J.; Wang, D. An enhanced diversity indicator-based many-objective evolutionary algorithm with shape-conforming convergence metric. *Appl. Soft Comput.* **2024**, *166*, 112161. [CrossRef]
37. Liu, Y.; Gong, D.; Sun, J.; Jin, Y. A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. Cybern.* **2017**, *47*, 2689–2702. [CrossRef]
38. Liang, Z.; Hu, K.; Ma, X.; Zhu, Z. A many-objective evolutionary algorithm based on a two-round selection strategy. *IEEE Trans. Cybern.* **2021**, *51*, 1417–1429. [CrossRef]
39. Yang, L.; Zhang, Y.; Cao, J.; Li, K.; Wang, D. A many-objective evolutionary algorithm based on reference vector guided selection and two diversity and convergence enhancement strategies. *Appl. Soft Comput.* **2024**, *154*, 111369. [CrossRef]
40. Zhang, W.; Liu, J.; Yang, W.; Tan, S. A cascading elimination-based evolutionary algorithm with variable classification mutation for many-objective optimization. *Inf. Sci.* **2024**, *662*, 120200. [CrossRef]
41. Zhang, Z.; Wang, H.; Zhang, W.; Cui, Z. Cooperative-competitive two-stage game mechanism assisted many-objective evolutionary algorithm. *Inf. Sci.* **2023**, *647*, 119559. [CrossRef]
42. Shen, J.; Wang, P.; Dong, H.; Li, J.; Wang, W. A multistage evolutionary algorithm for many-objective optimization. *Inf. Sci.* **2022**, *589*, 531–549. [CrossRef]
43. Ming, F.; Gong, W.; Wang, L. A two-stage evolutionary algorithm with balanced convergence and diversity for many-objective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 6222–6234. [CrossRef]
44. Zhou, Y.; Li, S.; Pedrycz, W.; Feng, G. ACDB-EA: Adaptive convergence-diversity balanced evolutionary algorithm for many-objective optimization. *Swarm Evol. Comput.* **2022**, *75*, 101145. [CrossRef]
45. Zhu, S.; Xu, L. Many-objective fuzzy centroids clustering algorithm for categorical data. *Expert Syst. Appl.* **2018**, *96*, 230–248. [CrossRef]
46. Zhu, S.; Xu, L.; Goodman, E.D. Hierarchical Topology-Based Cluster Representation for Scalable Evolutionary Multiobjective Clustering. *IEEE Trans. Cybern.* **2022**, *52*, 9846–9860. [CrossRef] [PubMed]
47. Cui, M.; Li, L.; Zhou, M.; Abusorrah, A. Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems. *IEEE Trans. Evol. Comput.* **2022**, *26*, 676–689. [CrossRef]
48. Cai, X.; Mei, Z.; Fan, Z. A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors. *IEEE Trans. Cybern.* **2018**, *48*, 2335–2348. [CrossRef] [PubMed]
49. Suresh, A.; Deb, K. Machine learning based prediction of new Pareto-optimal solutions from pseudo-weights. *IEEE Trans. Evol. Comput.* **2024**, *28*, 1351–1365. [CrossRef]
50. Zhu, C.; Cai, X.; Fan, Z.; Sulaman, M. A two-phase many-objective evolutionary algorithm with penalty based adjustment for reference lines. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2161–2168.
51. 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]
52. Blank, J.; Deb, K.; Dhebar, Y.; Bandaru, S.; Seada, H. Generating well-spaced points on a unit simplex for evolutionary many-objective optimization. *IEEE Trans. Evol. Comput.* **2020**, *25*, 48–60. [CrossRef]
53. Deb, K.; do Val Lopes, C.L.; Martins, F.V.C.; Wanner, E.F. Identifying Pareto Fronts Reliably Using a Multistage Reference-Vector-Based Framework. *IEEE Trans. Evol. Comput.* **2024**, *28*, 252–266. [CrossRef]
54. 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]
55. Sun, Y.; Yen, G.G.; Yi, Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2019**, *23*, 173–187. [CrossRef]
56. Tian, Y.; Si, L.; Zhang, X.; Tan, K.C.; Jin, Y. Local model-based Pareto front estimation for multiobjective optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 623–634. [CrossRef]
57. Wang, H.; Wei, Z.; Yu, G.; Wang, S.; Wu, J.; Liu, J. A two-stage many-objective evolutionary algorithm with dynamic generalized Pareto dominance. *Int. J. Intell. Syst.* **2022**, *37*, 9833–9862. [CrossRef]
58. Liu, Q.; Jin, Y.; Heiderich, M.; Rodemann, T.; Yu, G. An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems. *IEEE Trans. Cybern.* **2022**, *52*, 2698–2711. [CrossRef]
59. Palakonda, V.; Kang, J.M.; Jung, H. Benchmarking Real-World Many-Objective Problems: A Problem Suite With Baseline Results. *IEEE Access* **2024**, *12*, 49275–49290. [CrossRef]
60. Palakonda, V.; Kang, J.M. Many-Objective Real-World Engineering Problems: A Comparative Study of State-of-the-Art Algorithms. *IEEE Access* **2023**, *11*, 111636–111654. [CrossRef]

61. Zhang, L.; Zhang, H.; Chen, Z.; Liu, S.; Yang, H.; Zhao, H. A Multi-Population Based Evolutionary Algorithm for Many-Objective Recommendations. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 1969–1982. [CrossRef]
62. Zhang, J.; Gong, B.; Waqas, M.; Tu, S.; Han, Z. A hybrid many-objective optimization algorithm for task offloading and resource allocation in multi-server mobile edge computing networks. *IEEE Trans. Serv. Comput.* **2023**, *16*, 3101–3114. [CrossRef]