

Research Article

A Knee Point-Driven Many-Objective Evolutionary Algorithm with Adaptive Switching Mechanism

Maowei He^D,¹ Xu Wang,¹ Hanning Chen^D,^{2,3} and Xuguang Li⁴

¹Tiangong University, Tianjin 300387, China

²Engineering Research Center of Integration and Application of Digital Learning Technology, Ministry of Education, Beijing 100039, China

³Tianjin Swom Technology Development Co., Ltd., Tianjin 300387, China

⁴Central Committee of the Communist Party of China, Beijing 100010, China

Correspondence should be addressed to Hanning Chen; perfect_chn@hotmail.com

Received 4 September 2023; Revised 31 October 2023; Accepted 30 November 2023; Published 3 January 2024

Academic Editor: Wei-Chiang Hong

Copyright © 2024 Maowei He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Pareto dominance-based evolutionary algorithms can effectively address multiobjective optimization problems (MOPs). However, when dealing with many-objective optimization problems with more than three objectives (MaOPs), the Pareto dominance relationships cannot effectively distinguish the nondominated solutions in high-dimensional spaces. With the increase of the number of objectives, the proportion of dominance-resistant solutions (DRSs) in the population rapidly increases, which leads to insufficient selection pressure. In this paper, to address the challenges on MaOPs, a knee point-driven many-objective evolutionary algorithm with adaptive switching mechanism (KPEA) is proposed. In KPEA, the knee points determined by an adaptive strategy are introduced for not only mating selection but also environmental selection, which increases the probability of generating excellent offspring. In addition, to remove dominance-resistant solutions (DRSs) in the population, an interquartile range method is adopted, which enhances the selection pressure. Moreover, a novel adaptive switching mechanism between angle-based selection and penalty for selecting solutions is proposed, which is aimed at achieving a balance between convergence and diversity. To validate the performance of KPEA, it is compared with five state-of-the-art many-objective evolutionary algorithms. All algorithms are evaluated on 20 benchmark problems, i.e., WFG1-9, MaF1, and MaF4-13 with 3, 5, 8, and 10 objectives. The experimental results demonstrate that KPEA outperforms the compared algorithms in terms of HV and IGD in most of the test instances.

1. Introduction

Multiobjective optimization problems (MOPs) are ubiquitous in the real-world, such as wireless sensor networks [1], engineering design [2], workflow applications [3], and robot path planning [4]. In MOPs, the objectives usually conflict with each other. Generally, an MOP with M conflicting objectives can be defined as follows:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_M(x))^T,$$

s.t.x $\in \Omega$, (1)

where $x = (x_1, x_2, \dots, x_n)$ is an *n*-dimensional decision vector in the decision space Ω and *M* is the number of objective functions. MOPs with more than three-objectives (i.e., M > 3) are also referred to as many-objective optimization problems (MaOPs) [5, 6].

Generally, many-objective evolutionary algorithms (MaOEAs) still adopt the Pareto dominance as the criterion for selecting solutions. However, in high-dimensional spaces, MaOEAs may face two significant challenges due to the curse of dimensionality. The first challenge is the phenomenon of dominance resistance. It means that the number of nondominated solutions in the population rapidly increases as the number of objectives increases, which leads to the incomparability of solutions [7, 8]. The resulting solutions that may survive for many generations in the population are called dominance resistance solutions (DRSs). The presence of DRSs

will further expand the search space, which causes the convergence deterioration in the local search space. The second challenge is the contradiction between maintaining population convergence and diversity [9, 10].

To address these challenges, many novel MaOEAs have been proposed [11, 12]. These methods can be broadly classified into the following four categories.

The first category involves the redefining or relaxing the Pareto dominance relationship. Examples of improved dominance relationships include angle dominance [13], grid dominance [14], scalarization-based dominance relationship [15], RPS-dominance relationship [16], and reinforced dominance relationship [17]. NSGA-II+AD [13] proposes an angle dominance criterion that is insensitive to parameters. And the involved angle is composed of the solution and each objective axis. This dominance relationship cleverly expands the dominance region of a solution, which could increase the convergence pressure. In CDR-MOEA [18], a dominance relationship based on a convergence metric (CDR) is proposed, which combines the convergence metric and the adaptive parameter λ based on cosine similarity. This dominance relationship effectively balances the convergence and diversity of the nondominated solution set. MOEA/D-AED [19] uses adaptive Epsilon dominance to control the number of nondominated solutions, which effectively addresses problems with complex Pareto fronts (PFs).

The second type involves the decomposition-based algorithms, whose core idea is to solve MOP by decomposing a MOP into many single-objective subproblems. MOEA/D [20] is a typical decomposition-based algorithm that guides candidate solutions towards the Pareto front by defining a set of uniformly distributed weight vectors. However, this approach relies on the degree of match between the adopted reference vectors and the true Pareto front. To address the problems with irregular PFs, various adaptive decomposition-based methods have emerged. For example, AMOEA/D [21] adopts an adaptive addition-first-and-deletion-second strategy to dynamically adjust the weight vector. In addition, the reference points are widely used in most decomposition-based algorithms, such as NSGA-III [22], MOEA/DD [23], and SPEA/R [24]. SPEAR proposes a density estimation technique based on reference vectors to maintain population diversity. NSGA-III, as an extension of NSGA-II [25], features a combination of nondominated sorting with reference vectors. MOEA/DD, as a variant of MOEA/D, introduces the Pareto dominance and adaptively associates solutions with subregions. In summary, the decomposition-based method is suitable for solving MaOPs, but it is inevitable to preset the appropriate weight vectors and the scalar functions for guaranteeing their performance.

The third type is the indicator-based algorithm. Unlike the Pareto-dominance and decomposition-based algorithms, this method usually evaluates the quality of solutions using predefined performance indicators. Some well-known indicators are used as criteria in environmental selection, including inverted generational distance (IGD), hypervolume (HV), and generational distance (GD). The indicator-based method can provide more selection pressure towards the Pareto front. Some recently popular indicator-based algorithms include HypE [26], R2HCA-EMOA [27], and MaOEA/IGD [28]. HypE adopts the Lebesgue measure to calculate the fitness values of solutions. To reduce computational complexity, R2HCA-EMOA uses the R2 indicator to approximate the value of HV. However, this method cannot effectively balance convergence and diversity. MaOEA-IGD proposes a novel nondominance comparison strategy based on the IGD indicator. This strategy compares solutions to reference points, which improves the quality of population mating.

The final type is knee point-driven algorithms, which are aimed at improving the performance of MaOEAs by identifying knee points that contribute to a larger HV [29]. These algorithms employ strategies such as reflex angle [30], extended angle dominance [8], niching-based method [31], distance-based strategies [32], and offspring generation strategy [33] to identify knee points, which increase selection pressure and improve convergence. KnEA [34] identifies knee points based on the distance from the solutions to the hyperplane. And the number of knee points involved in this strategy is controlled by threshold parameter T. However, the performance of KnEA is greatly influenced by the variation of parameter T. Pi-MOEA [35] uses the average ranking method to identify pivot solutions within the neighborhood, which is insensitive to parameters. k-NSGAII [36] applies a distancebased metric to identify convex and edge knee points. To improve the accuracy of the knee-detection method, k-NSGAII designs a specific parameter η to control the width of the knee region. However, these strategies emphasize convergence too much but neglect the importance of diversity. In contrast, KnMAPIO [29] designs a novel environmental selection based on knee point-oriented dominance, which uses extreme and boundary points to select solutions from the critical layer. This selection strategy effectively balances the convergence and diversity.

In the past few years, many novel strategies for addressing MaOPs have been proposed. However, most MaOEAs still cannot meet the performance requirements. There are two main challenges.

First, without any prior knowledge, the diversity-first-convergence-second principle adopted by most Pareto dominancebased MaOEAs leads to the loss of population convergence.

Second, a strategy that overly emphasizes diversity may select many DRSs into the next-generation population, which leads to the population gradually moving away from the PF.

To address the above challenges, a knee point-driven many-objective evolutionary algorithm with adaptive switching mechanism (KPEA) is proposed in this paper, which is aimed at achieving a balance between convergence and diversity. Specifically, to increase the probability of generating excellent offspring, KPEA incorporates the knee points and the weighted distances into its mating strategy. During the environmental selection, an adaptive switching mechanism between angle-based selection and penalty is designed to balance convergence and diversity. The main contributions of this study can be summarized as follows:

 Unlike the methods that indirectly identify DRS based on extreme solutions, to detect and eliminate DRSs in the population, an interquartile range method based on population convergence is used, which enhances the quality of population mating and ensures even scaling of each objective

(2) Compared with selection mechanisms based on convergence-first or diversity-first, in the environmental selection process, an adaptive switching mechanism between angle-based selection and penalty can maintain a balance between convergence and diversity, instead of excessively emphasizing either of them

The remainder of this paper is organized as follows. The related works and the motivation of this study are presented in Section 2. A detailed description of the framework and core strategies of KPEA is presented in Section 3. The experimental setup is presented in Section 4. The experimental results and related analysis are presented in Section 5. Further investigation and discussion on KPEA are provided in Section 6. Finally, the conclusions of this paper and future work are given in Section 7.

2. Related Work and Motivation

In this section, the relevant research on dominance resistance solutions (DRSs) and angle-based selection strategies are reviewed, and the research motivation is elaborated briefly.

2.1. Impact of DRSs. In general, the Pareto-based algorithms regard the nondominated solutions as suitable candidates for the next generation. When some solutions are barely dominated but clearly inferior to others, then they will not be removed by the Pareto dominance-based strategies [7, 37, 38]. This phenomenon is called dominance resistance, and these solutions are called DRSs.

Regarding the specific definition of DRSs, various researchers have provided various descriptions. In literature [7], "DRSs are extremely inferior to others in at least one objective, and therefore, they are apart from Pareto optimums." DRSs were rephrased in literature [38] as "These are points in the nondominated set that are extremely poor in at least one of the objectives, while being extremely good in some others." Since DRSs are located on certain boundaries of the objective space, most may have the best density values, which causes these solutions to be always reserved for the next generation [39]. Furthermore, the proportion of nondominated solutions in a population grows exponentially as the number of objectives increases [40]. Meanwhile, the number of DRSs may also increase with the number of objectives, which results in more intense dominance resistance [5]. Even worse, DRSs can seriously inflate the nadir in specific axial directions, leading to an uneven scaling of each objective [8].

Figure 1 illustrates the situation of DRSs in the population in biobjective and triobjective spaces. The solid purple dots represent the nondominated solutions in the search space. In Figure 1, each axis shows the fitness value on each objective, and the nondominated solutions within the red dashed box are DRSs. With the increase of the number of objectives, the presence of DRSs will further expand the search space, which causes the convergence deterioration in local search space.

Some researchers have tried to identify the extreme solutions directly with specific methods. In many-objective particle swarm optimization (MaOPSO) [41], the solutions in the external archive that minimize the achievement scalarizing function (ASF) [42] are considered as the extreme solutions. In the multiphase balance of diversity and convergence in many-objective optimization (B-NSGA-III) [43], the extreme-LS operator is used to identify the extreme solutions in objective directions. Nevertheless, because MaOPSO and B-NSGA-III are all Pareto-based algorithms, their effectiveness is significantly impacted by DRSs. Different from directly determining the precise extreme solutions, some researchers are committed to deleting the DRSs in the current population to indirectly determine the extreme solutions.

Bhattacharjee et al. [38] propose a six-sigma-based method to eliminate DRSs. In this method, a solution is classified as DRSs when its value on the objective exceeds the average value plus six times the standard deviation of all nondominated solutions on that objective. However, the strategy still has some drawbacks. Xiang et al. [44] demonstrate experimentally that the method produces solutions with very poor convergence for some problems. Furthermore, Zhou et al. [45] propose a method to eliminate DRS by modifying the current candidate extreme solutions. Unfortunately, when there are several DRSs on an objective axis, this strategy cannot eliminate all DRSs.

In summary, the elimination of DRSs is beneficial to finding the exact extremal solutions. As Bhattacharjee et al. stated in literature [38], "the detection and elimination of DRSs is far from trivial, and further research is needed to address this issue." In the paper, an interquartile range method (IQR) is introduced to eliminate certain nondominated solutions in the population whose convergence values are greater than the upper third quartile (Q3).

2.2. Angle-Based Selection Strategy. Basing on the embedding position of the angle selection criterion, the angle-based evolutionary algorithm can be classified into three categories. The first category involves applying the angle selection to environmental selection. The decomposition-based ranking and angle-based selection evolutionary algorithm (MOEA/ D-SAS) [46] employs the angle information between solutions in the objective space to maintain population diversity. The vector angle-based evolutionary algorithm (VaEA) [47] uses the maximum-vector-angle-first strategy to select elite individuals. However, in certain cases, the selected solutions may have poor convergence. Thus, VaEA adopts the worseelimination principle and conditionally uses alternative solutions to replace poorly converging ones. The scalar projection and perspective-based evolutionary algorithm (PAEA) [45] uses the minimum angle formed between a solution and a unit vector as a control threshold to maintain a balance between convergence and diversity. However, how to determine the threshold for the search direction in these strategies is usually an extremely complex problem.

The second category involves incorporating angle selection into the dominance relationship. Wang et al. [37]



FIGURE 1: The existence of DRSs in the population at different objectives.

propose the generalized Pareto optimum (GPO) that uses a parameter ϕ_i , the extension angle on the *i*th objective to redefine the dominance relationship. Yuan et al. [13] propose an angle dominance criterion that is insensitive to parameters. The involved angle is composed of the solution and each objective axis. This dominance relationship cleverly expands the dominance region of a solution, which could increase the convergence pressure.

The third category involves considering angle selection in both environmental selection and dominance relationships. He and Yen [48] propose a coordinated selection strategy to solve MaOP problems. This strategy uses ASF distance as the convergence metric and the angle formed between individuals as the diversity metric, which is applied to both environmental selection and mating selection. During the process of evolution, this strategy can balance the convergence and diversity of the whole population. Li et al. [49] use an archive population to update reference points and construct a distance scaling function based on the angle between individuals and reference points. This function is employed in mating selection, which improves the quality of offspring in the population.

Based on the above research, this paper presents a novel environmental selection strategy that regards the relationship between population size and the number of nondominated solutions as two criteria for selecting the solutions. The knee points determined by an adaptive strategy are introduced for not only mating selection but also environmental selection.

2.3. Motivation of This Paper. When dealing with manyobjective optimization problems (MaOPs), dominancebased algorithms often face difficulties caused by DRSs and active diversity promotion [50, 51]. Decomposition-based algorithms can avoid the problem of insufficient selection pressure but require a predefined set of weight vectors. However, uniformly distributed weight vectors cannot guarantee well-distributed solutions. In contrast, angle-based selection strategies perform well on enhancing population diversity, especially when dealing with MaOPs with complex Pareto fronts. The studies in Section 2.2 indicate that many algorithms using angle-based ecological niche protection techniques seem effective, but there are still many unresolved problems. When dealing with nonlinear optimization problems, for the strategy proposed by the hyperplane-assisted evolutionary algorithm (hpaEA) [52] to protect prominent solutions, it may be invalid to maintain the diversity of the population.

The strategy of protecting prominent solutions proposed by the hyperplane-assisted evolutionary algorithm (hpaEA) [52] is not suitable for MaOPs with nonlinear Pareto fronts. On this type of problem, it is unable to maintain the diversity of the population for this strategy. The strategy proposed by VaEA overemphasizes diversity and lacks a reasonable design for angle thresholds, which results in difficultly achieving effective convergence in certain high-dimensional problems.

To address the above issues, this paper proposes a knee point-driven many-objective evolutionary algorithm with adaptive switching mechanism (KPEA). It incorporates the knee points and weighted distance into its mating strategy to increase the probability of generating excellent offspring. An interquartile range method is introduced to detect and eliminate DRSs in the population. During the environmental selection, an adaptive switching mechanism between anglebased selection and penalty is designed to balance convergence and diversity.

Up to now, most selection mechanisms focus on either convergence or diversity optimization. Generally, without any prior knowledge, a single strategy cannot offer a universal and acceptable performance on most MaOPs.

With this motivation, an adaptive switching mechanism between angle-based selection and penalty proposed in this paper could keep a balance between convergence and diversity, instead of excessively emphasizing either of them. The selection mechanism first preserves knee points identified by adaptive recognition and then selects individuals based on the maximum-vector-angle-first principle. The adaptive penalty mechanism selects the individuals with the best convergence one by one. Then, it adaptively adjusts the penalty set based on the distribution of solutions, which achieves comprehensive coverage of the Pareto front. As mentioned above, KPEA not only eliminates the influence of DRSs throughout the entire evolutionary process but also achieves a balance between diversity and convergence.

3. Proposed Algorithm

This section provides a detailed description of the proposed KPEA. Firstly, the main framework of the algorithm KPEA is given. Subsequently, the quartile detection method is regarded as a preprocessing tool to detect DRSs, which enhances the quality of population mating and ensures even scaling of each objective. In addition, in the mating selection, knee points and weighted distances are introduced as secondary selection criteria. Finally, an environment selection strategy is proposed with an adaptive switching mechanism, which can adaptively switch according to the relationship between population size and the number of nondominated solutions.

3.1. General Framework. A common Pareto-based MaOEA framework is employed in the proposed KPEA, as shown in Algorithm 1. First of all, an initial parent population P of size N is randomly generated. Then, after individuals are selected from P using the binary tournament selection strategy, an offspring population Q is generated using mutation (lines 5 and 6). Three metrics, namely, dominance relation, knee point criterion, and weighted distance metric, are employed as the selection criteria for the binary tournament selection. If there are some checked DRSs in the combined population S, a preprocessing mechanism based on IQR is used to detect and eliminate DRSs (line 8). Next, the nondominated sorting is performed on the preprocessed new combined population P (line 9), and an adaptive strategy is introduced to identify the knee points of each nondominated front in the combined population P (line 10). Finally, with an adaptive switching mechanism, N individuals are selected from the population based on the relationship between population size and the number of nondominated solutions. As a vital contribution of this study, the niche protection operation different from VaEA [47] considers both convergence and diversity of each solution in the population.

In order to clearly illustrate the framework of KPEA, the specific flow of KPEA is shown in Figure 2. Specifically, KPEA consists of five evolution parts: mating selection, preprocessing, nondominated sorting, identification of knee points, and environment selection. During the environmental selection, an adaptive switching mechanism between angle-based selection and penalty is designed to balance convergence and diversity. In addition, the switching conditions and the specific evolutionary process of this novel adaptive switching mechanism are visualized. 3.2. Preprocessing Operation. Preprocessing is one of the most important steps in data mining, which is aimed at eliminating outliers [53]. In fact, during the optimization process of MaOPs, there are many DRSs (i.e., outliers) within the population. As discussed in Section 2, these DRSs have a negative impact on the quality of the final solutions. Thus, removing these DRSs is a crucial step to improve the performance of the algorithm.

In this paper, an interquartile range (IQR) method is introduced to detect and eliminate DRSs. IQR is a descriptive statistic for identifying outliers in any dataset. Given an ordered one-dimensional dataset R, the lower quartile (Q_1) and upper quartile (Q_3) of the dataset are determined first, and then the IQR is calculated as follows.

$$\begin{cases} UB = Q_3 + r(Q_3 - Q_1), \\ LB = Q_1 - r(Q_3 - Q_1), \end{cases}$$
(2)

where UB and LB denote the upper and lower boundaries of the dataset, respectively. The value of parameter r is set to 1.5, which will be comparatively analyzed in Section 4. In addition, Q_1 and Q_3 are defined as

$$\begin{cases} i = \left\lceil \left(\frac{||R||}{4}\right) \right\rceil, \\ j = \left\lceil \left(\frac{3||R||}{4}\right) \right\rceil, \\ Q_1 = R_i, \\ Q_3 = R_j, \end{cases}$$
(3)

where $\lceil \bullet \rceil$ indicates rounding up and $||\bullet||$ represents the size of the set. The specific process of preprocessing aimed at the dataset $R = \{48, 32, \dots, 53, 17\}$, which is composed of convergence metrics of solutions, is visually illustrated in Figure 3.

First, the convergence metrics of all solutions are calculated and sorted in ascending order. According to the quartile method, the index positions *i* and *j* locate at 1/4 and 3/4 of the sorted convergence metrics, respectively. In Figure 3, $i = \lceil 13^*1/4 \rceil = 4$ and $j = \lceil 13^*3/4 \rceil = 10$, where $\lceil \bullet \rceil$ indicates rounding up. Q_1 and Q_3 are calculated according to Eq. (3). In Figure 3, $Q_1 = 32$ and $Q_3 = 55$. Then, the lower and upper boundaries are set as are calculated according to Eq. (2). In this paper, a solution is considered a DRS when its convergence metric is greater than the upper boundary. Finally, the blue point with a convergence metric of 92 will be regarded as DRSs by the preprocessing mechanism and will be eliminated from the population.

Algorithm 2 presents the specific process of the preprocessing operation. In the first step, the convergence metric for each solution is calculated and stored in the set CM (line 1). Then, CM are sorted in ascending order (line 2). Subsequently, the corresponding index position *i* of Q_1 in *P* is calculated according to the quartile method, and then the value of Q_1 in the ordered CM is calculated using Eq. (3) (line 3). As the same as the solving procedure of Q_1 , Q_3 is calculated **Input:***P* (population), *N* (population size), *M* (number of objectives), *K* (set of knee points), *T* (rate of knee points in population) 1 $r \leftarrow 1, t \leftarrow 0$ /*Adaptive parameters for finding knee points*/ 2 $K \leftarrow \emptyset$ 3 $P \leftarrow Initialize(N)$ 4 While termination criterion not fulfilled do 5 $P' \leftarrow Mating_selection(P, K, N)$ $Q \leftarrow Variation(P')$ 6 7 $S \leftarrow P \cup Q$ 8 P = Preprocessing(S) /* A preprocessing operation to eliminate the DRSs. */9 $F \leftarrow Nondominated_sort(P)$ /* the preprocessed population is assigned to N_F non-dominated fronts, F_i , $1 \le i \le N_F$ */ 10 $[K, r, t] \leftarrow Finding_Knee_point(F, T, r, t)$ $P \leftarrow Environmental_selection(F, K, N)$ 11 12 End While 13 Return P





FIGURE 2: The running process of KPEA.

(line 4). To avoid the situation that the number of candidate solutions after preprocessing is less than the population size N, Q_1 and Q_3 may be adjusted. If j > N, the upper and lower quartiles of the CM are adjusted accordingly (lines 5-7). After determining the upper and lower quartiles, UB is calculated using Eq. (2) (line 9). Finally, any solutions with a convergence metric larger than UB are eliminated, leading to a new set of solutions P' (line 10). P' is an auxiliary set for detecting whether there are DRSs in P. The disparity indicator I is employed, which calculates the diagonal length of the hypercube formed by the extreme objective values in population (line 11). The disparity indicator I is defined as

$$I = \left[\sum_{i=1}^{m} \left(\max_{x \in P} f_i(x) - \min_{x \in P} f_i(x)\right)^2\right]^{1/2},$$
 (4)

where *m* represents the number of objectives and *x* is a candidate solution that belongs to the population. After calculating the disparity indicator I_p and I_p of *P* and *P*, the difference ratio γ between two sets is calculated as follows:

$$\gamma = \frac{I_P}{I_{P'}}.$$
(5)



FIGURE 3: An illustration of identifying the DRSs with the interquartile range method, where LB and UB represent the lower and upper boundaries, respectively, and Q_1 and Q_3 represent the lower and upper quartiles, respectively.

Input: *P* (population), *N* (population size) Compute the convergence metrics of solutions $P \rightarrow CM$ 1 Sort CM; 2 3 The index position *i* is calculated, and then Q_1 is calculated using Eq. (3). 4 The index position j is calculated, and then Q_3 is calculated using Eq. (3). 5 If j > N then i = ||CM|| - N and $Q_1 = CM(i)$ 6 7 j = N and $Q_3 = CM(j)$ 8 End Compute the UB of the ordered set CM 9 10 Eliminate the solutions whose convergence metrics are greater than $UB \rightarrow P'$ Calculate the disparity indicators Is of P and P': /*Detect if there are DRSs in P */ 11 12 If $\gamma > 1$ then 13 P = P'14 End

ALGORITHM 2: Preprocessing (P, N).

Obviously, if γ is smaller than or equal to 1, it implies that there are no DRSs in *P*. On the contrary, if γ is greater than 1, it indicates that one or more DRSs exist in *P*, and *P* will be updated to *P* with no DRSs (lines 12 and 13).

3.3. Adaptive Strategy of Identifying Knee Points. In KPEA, the knee points play a crucial role. According to literature [34], knee point-based selection has more advantages than using HV-based selection.

The core idea of determining the knee point is demonstrated in Figure 4. In biobjective minimization problems, the extreme solutions with the maximum value on the respective objective axes f_1 and f_2 need to be first determined. Then, the extreme line L is defined as

$$L = a\mathbf{x} + b\mathbf{y} + c. \tag{6}$$

Finally, the distance *D* from each solution $A(x_A, y_A)$ to *L* is calculated as

$$D(A,L) = \begin{cases} \frac{|ax_{A} + by_{A} + c|}{\sqrt{a^{2} + b^{2}}}, & \text{if } ax_{A} + by_{A} + c < 0, \\ -\frac{|ax_{A} + by_{A} + c|}{\sqrt{a^{2} + b^{2}}}, & \text{otherwise.} \end{cases}$$
(7)

Within the neighborhood range, the solution with the maximum distance to L will be identified as the knee point. In many-objective optimization problems, L refers to the hyperplane [54].

As shown in Figure 4, solutions A, B, and C are located within the neighborhood enclosed by dashed lines. Among these, solution A has the maximum distance to L. Therefore, A is considered as the knee point within the neighborhood. When there is only one solution within the neighborhood (e.g., solution H), that solution is also considered as the knee point.

The results of identifying knee points will be significantly affected by the size of the neighborhood of the solutions. Solutions B, E, and H are identified as knee points based on the neighborhood size defined in Figure 4. However, when all solutions are included in the same neighborhood, only solution E is identified as the knee point. To address this issue, a strategy is introduced to adjust the neighborhood size of solutions.

Assuming that at the *g*th generation, there are N_F -nondominated fronts within the population. Each nondominated front is composed of a set of nondominated solutions F_i . The neighborhood of a solution is defined by a hypercube of dimension $V_g^{-1} \times V_g^{-2} \times \cdots \times V_g^{-j} \times \cdots \times V_g^{-M}$, where $1 \le j \le M$ and *M* denotes the number of objectives. Specifically, the neighborhood size V_g^{-j} of the *j*th objective is calculated as follows:

$$V_g^j = \left(f \max_g^j + f \min_g^j\right) \cdot r_g,\tag{8}$$

where $f \min_{g}^{j}$ and $f \max_{g}^{j}$ represent the minimum and maximum values of the nondominated set F_{i} on the *j*th objective at the *g*th generation, respectively. The parameter r_{g} denotes the ratio of the size of the neighborhood to the span of *j*th objective in nondominated front F_{i} at the *g*th generation, which is defined as

$$r_g = r_{g-1} * e^{-(1 - t_{g-1}/T)/M}, \qquad (9)$$



FIGURE 4: Description of identifying the knee points in KPEA.

where r_{g-1} represents the ratio of the size of the neighborhood to the span of *j*th objective in F_i at the (g-1)th generation and t_{g-1} indicates the ratio of knee points to the number of nondominated solutions in F_i at the (g-1)th generation. The threshold T (0 < T < 1) is used to control the proportion of knee points in F_i .

The main steps for identifying the knee points are presented in Algorithm 3. The knee points are identified starting from the first nondominated front (lines 2). First, the extreme points in F_i are identified, and the hyperplane L is calculated (lines 3 and 4). Next, the size of the neighborhood for the solutions in F_i is calculated (lines 5-8). The distances from each solution to L are calculated and sorted in descending order (lines 9¢-10). Then, the knee points within each neighborhood are identified and added into the knee point set K (lines 12-15). Finally, the ratio between the number of knee points and the number of nondominated solutions in F_i is updated (line 17).

3.4. Binary Tournament Mating Selection. In KPEA, a binary tournament selection based on three metrics, including the Pareto dominance, knee point criterion, and weighted distance, is designed, as shown in Algorithm 4.

Two individuals, x and y, are randomly selected from the parent population P. If y is dominated by x, then x will be preferred (lines 4-7). If x and y are nondominated, the individual belonging to the knee point set K will be selected (lines 9-12). If both or none of x and y belongs to the knee point set K, the weighted distances will be used to compare x and y. The weighted distances DWs of x and y are calculated as follows:

$$DW(x) = \sum_{i=1}^{k} w_{xi} dis_{x,xi},$$
 (10)

$$w_{\rm xi} = \frac{r_{\rm xi}}{\sum_{i=1}^{k} r_{\rm xi}},\tag{11}$$

Input: *F* (sorted population), *T* (rate of knee points in population), *r*, *t* (adaptive parameters) 1 $K \leftarrow \emptyset / *$ knee points */ 2 For all $F_i \in F$ do 3 $E \leftarrow Find_extreme_solution(F_i)$ 4 $L \leftarrow Calculate_extreme_hyperplane(E)$ 5 Update r by Eq. (9) 6 $fmax \leftarrow Maximum$ value of each objective in F_i 7 $fmin \leftarrow Minimum$ value of each objective in F_i 8 Calculate V by Eq. (8) 9 Calculate the distance from each solution in F_i to L by Eq. (7) 10 Sort F_i in a descending order according to the distances 11 $Size_{Fi} \leftarrow |F_i|$ 12 For all $p \in F_i$ do $NB \longleftarrow \{a | a \in F_i \longrightarrow |f_a^j - f_p^j| \le V^j, 1 \le j \le M\}$ 13 $K \leftarrow K \bigcup \{p\}$ 14 $F_i \leftarrow F_i \backslash \overline{NB}$ 15 16 **End For** $t = |K| / Size_{F_i}$ 17 End for 18 19 Return K, r and t

ALGORITHM 3: Finding_knee_point (F, T, r, and t).

```
Input: P (population), K (set of knee points), N (population size)
    Q \leftarrow \emptyset
1
     While |Q| < N do
2
3
           Randomly choose x and y from P
4
           If x \prec y then
                Q \leftarrow Q \cup \{x\}
5
6
           Else If y \prec x then
7
                 Q \leftarrow Q \cup \{y\}
8
           Else
9
                 If x \in K and y \notin K then
                        Q \leftarrow Q \cup \{x\}
10
                  Else If y \in K and x \notin K then
11
12
                        Q \leftarrow Q \cup \{y\}
13
                  Else
14
                        If DW(x) > DW(y) then
15
                                 Q \leftarrow Q \cup \{x\}
                        Else If DW(x) < DW(y) then
16
17
                                 Q \leftarrow Q \cup \{y\}
18
                        Else
19
                                 If rand() < 0.5 then
20
                                        Q \leftarrow Q \cup \{x\}
21
                                 Else
22
                                        Q \leftarrow Q \cup \{y\}
23
                                 End If
                        End If
24
25
                  End If
26
            End If
27
      End While
28
      Return Q
```

ALGORITHM 4: Mating_selection (P, K, and N).

$$r_{\rm xi} = \frac{1}{\left| \operatorname{dis}_{{\rm x},{\rm xi}} - 1/k \sum_{i=1}^{k} \operatorname{dis}_{{\rm x},{\rm xi}} \right|},\tag{12}$$

where x_i is the *i*th nearest neighbor of x in the population, w_{xi} is the weight of x_i , $dis_{x,xi}$ is the Euclidean distance between x and x_i , and r_{xi} is the rank of distance $dis_{x,xi}$ among all distances $dis_{x,xi}$, $1 \le j \le k$.

Then, the individual with a large value of DW is selected (lines 14-17). If DW(x) = DW(y), either is selected (lines 19-23). This procedure is repeated until *N* individuals are selected.

3.5. Environmental Selection. In this paper, a novel environmental selection strategy is proposed, which is aimed at enhancing selection pressure while balancing convergence and diversity.

The details of the novel environmental selection strategy are provided in Algorithm 5. Firstly, the sorted population Fwill be normalized (line 1). The objective vector $F(x_j)$ of a solution x_j in the sorted population F can be normalized to $F'(x_j) = (f'_1(x_j), f'_2(x_j), \dots, f'_m(x_j))^T$ using the following formula [55, 56]:

$$f_{i}'(x_{j}) = \frac{f_{i}(x_{j}) - z_{i}^{\max}}{z_{i}^{\max} - z_{i}^{\min}},$$
(13)

where z_i^{\min} denotes the ideal point, that is, $z_i^{\min} = \min f_i(x_j)$ $i \in \{1, \dots, M\}, j \in \{1, \dots, |F|\}; z_i^{\max}$ represents the worst point, that is, $z_i^{\max} = \max f_i(x_j)i \in \{1, \dots, M\}, j \in \{1, \dots, |F|\}.$

When the number of solutions in F_1 exceeds the population size N, the solutions based on the Pareto dominance relationship are incomparable. In this situation, the convergence and diversity of KPEA are maintained alternately by the penalty mechanism (introduced in Section 3.5.1), which also decreases the sensitivity to the distribution threshold (lines 2 and 3).

In contrast, when the number of nondominated solutions in F_1 is less than the population size, knee points are first preserved by the selection mechanism (introduced in Section 3.5.2), which increases selection pressure and improves convergence. Then, the remaining solutions are selected with association and niching functions based on the principle of maximum-vector-angle-first, which maintains diversity (lines 4 and 5). The detailed explanations of these methods will be provided in the following sections.

3.5.1. Penalty. When the number of solutions in F_1 is greater than the population size N, the function Penalty() is triggered in KPEA, as shown in Algorithm 6. The excellent solutions for the next generation are selected based on the penalty principle, which considers both the diversity and convergence of the population. First, the angles $DM_i = \{\mu_1^i, \mu_2^i, \dots, \mu_{|F|-1}^i\}$ between p_i and $\{p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_{|F|}\}$ is computed and stored in $S_{DM} = \{DM_1, DM_2, \dots, DM_{|F|}\}$ in ascending order (lines 2-3). The distance CM_i from p_i to the idle point z^* is computed and stored in $S_{CM} = \{CM_1, CM_2, \dots, CM_2, \dots\}$

 \cdots , CM_{|*F*|} (lines 4-5). Then, the solution with the minimum value in S_{CM} is selected from the candidate solution set and put into the next-generation population P (lines 7 and 8). Once a solution is selected, its neighbors N_i are identified, where $\mu^{i}_{j} < \alpha$ and $\mu^{i}_{j} \in {\{\mu^{i}_{1}, \mu^{i}_{2}, \cdots, \mu^{i}_{|F|-1}\}}$. These neighbors N_i are put into the penalty set *E* (lines 9 and 10). The size of *E* is determined by the size of the initial candidate set *S* and the desired population size N. If the number of solutions in *E* exceeds its capacity limit, the penalty set *E* must be updated based on the similarity between the penalized solutions N_i and the selected solution set P. Specifically, the closer the solution p_i in E is to the candidate solution, the smaller the angle value DM_i (stored in S_{DM}) between p_i and the candidate solution. Therefore, among the neighbors of the candidate solutions, the top $N_i = S - N$ neighbors with smaller angle values are kept in the penalty set (lines 11-13). Then, the remaining well-distributed solutions N_F with larger angle values (stored in S_{DM}) will be moved from *E* to the candidate set *F* (line 14).

The process of the penalty mechanism in a 2D objective space is illustrated in Figure 5. In this example, there are 9 nondominated solutions in F_1 , and the population size N is 5. Firstly, solution A with min (S_{CM}) is selected. And then, as the angles between the solutions (B and C) and A are smaller than α , B and C will be penalized, as shown in Figure 5(b). Next, as the same as the method in Figure 5(b), solution I with min (S_{CM}) is selected. As the number of solutions in the penalty set does not exceed the limit, the neighbor H of I can be directly punished, as shown in Figure 5(c). Subsequently, solution F with min (S_{CM}) is selected. At present, solutions A, I, and F are contained in the population P. Solutions B, C, and H are included in the penalty set. Because the penalty set can only accommodate four solutions, the neighbors *E* and *G* cannot be directly penalized by the mechanism. Therefore, it is necessary to update the penalty set by removing solution H with max (S_{DM}) . After the update, the penalty set is composed of solutions B, C, E, and G. The selected solution may have no neighbors, which means there is no angle between the candidate solutions and the selected solutions smaller than α . In this case, no solution is penalized. Finally, based on the above steps, solutions D and H are selected, as shown in Figures 5(e) and 5(f).

3.5.2. Selection. Once the number of nondominated solutions in F_1 is less than the population size, the selection operation will be triggered, as given in Algorithm 7. The knee points of each nondominated front are selected first and added into P, which can increase selection pressure (line 2). Then, the remaining nondominated solutions from $F_1 \cup \cdots \cup F_{l-1}$ are selected and added into P. If the number of solutions in F_l exceeds the population size N, then F_l is considered as the critical front. N - |P| knee points with the minimum distance to the hyperplane are removed (lines 4 and 5). Otherwise, the association function that constructs the search direction based on P and the niching function that is aimed at ensuring the diversity of the population are used to select N - |P| solutions one by one from F_l (lines 6–8). The association and niching functions will be described in detail in the following sections. Input: *F* (sorted population), *K* (set of knee points), *N* (population size); Output: *P* (next population) 1 Normalization (*F*) 2 If $|F_1| > N$ then 3 $P \leftarrow Penalty(F_1, N)$; /* Solutions are selected one by one using penalty mechanism. */ 4 Else 5 $P \leftarrow Selection(F, K, N)$; /* The knee points are first preserved, and then each solution is selected based on the principle of maximum-vector-angle-first. */

6 End

7 Return P



Input: *F* (sorted population), *N* (population size); **Output**: *P* (next population) while |P| < N do 1 Compute the angles $DM_i = \{\mu_1^i, \mu_2^i, \dots, \mu_{|F|-1}^i\}$ between p_i and $\{p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_{|F|}\}$ 2 3 $S_{DM} = \{DM_1, DM_2, ..., DM_{|F|}\}$ Compute the distance CM_i from p_i to the idle point z^* ; 4 5 $S_{CM} = \{CM_1, CM_2, ..., CM_{|F|}\}$ 6 S = |F|Select p_i with min(S_{CM}) 7 8 Move p_i into P Identify the neighbors N_i of p_i , where $\mu_i^i < \alpha$ and $\mu_i^i \in {\{\mu_1^i, \mu_2^i, ..., \mu_{|F|-1}^i\}}$ 9 10 Move N_i into the penalty set E 11 If Size(E) > S - N then Identify the neighbors N_i of P, where the DM_i of N_i is smaller compared to the other neighbors of P, $DM_i \in \{DM_1, DM_2, DM$ 12 ..., $DM_{|F|}$ } 13 Penalize N_i into the penalty set E 14 Move the remaining solutions N_i into the solution set F, where the DM_i of N_i is larger compared to the other neighbors of $P, DM_i \in \{DM_1, DM_2, ..., DM_{|F|}\}$ End 15 End 16 Return P 17



3.5.3. Association. The angle-based association process is described in this section. After selecting the knee points into P, if P is not full, N - |P| solutions will be selected from F_l based on the vector angles. Before describing the association operation, some relevant formulas are first presented in this section.

The norm could calculate the vector angle between two solutions in the normalized objective space. The objective vector $F(x_j)$ in the sorted population F has been normalized to $F'(x_j) = (f'_1(x_j), f'_2(x_j), \dots, f'_m(x_j))^T$, as introduced at the beginning of Section 3.5. The norm of solution x_j in the sorted population is defined as

norm
$$(x_j) \triangleq \sqrt{\sum_{i=1}^{m} f'(x_j)^2}.$$
 (14)

The vector angle μ between solutions x_j and c_k is defined as

$$\mu(x_j, c_k) \triangleq \arccos \left| \frac{F'(x_j) \cdot F'(c_k)}{\operatorname{norm}(x_j) \cdot \operatorname{norm}(c_k)} \right|, \quad (15)$$

where $F'(x_j) \cdot F'(c_k)$ is to calculate the inner product between $F'(x_j)$ and $F'(c_k)$. And $F'(x_j) \cdot F'(c_k)$ is defined as

$$F'(x_{j}) \cdot F'(c_{k}) = \sum_{i=1}^{m} f_{i}'(x_{j}) \cdot f_{i}'(c_{k}).$$
(16)

The main steps of the association process are provided in Algorithm 8. For each remaining member in F_i , for example, x_j , the minimum vector angle (denoted as $\theta(x_j)$) between x_j and the solutions in *P* is calculated. The index of the solution in *P* that has the minimum vector angle to x_j is denoted as $\gamma(x_i)$.

Firstly, for each $x_j \in F_l$, its $\theta(x_j)$ and $\gamma(x_j)$ are initialized as ∞ and -1, respectively (lines 1-3). Then, for each $c_k \in P$,



FIGURE 5: An illustration of selecting solutions in the penalty mechanism. Five superior solutions need to be selected from 9 nondominated solutions *A*-*I*, while the penalty set can accommodate only four solutions. The threshold α represents the neighborhood range for each solution. (a) There are 9 nondominated solutions in the population. (b) *A* with min (S_{CM}) is selected. As $\angle Bz^*A < \alpha$ and $\angle Cz^*A < \alpha$, *B* and *C* are penalized. (c) *I* is selected, and then *H* is penalized. (d) *F* is selected, then *E* and *G* are penalized. (e) *D* is selected. (f) *H* is selected.

Input: F (sorted population), K (set of knee points), N (population size); **Output**: *P* (next population) $P = \emptyset, i = 1$ $P \leftarrow P \cup (K \cap F_l)$ /* Select the knee point and the solutions in $F_1 \dots F_l$ into P */ 2 $P \leftarrow F_1 \cup \ldots \cup F_{l-1}$ 3 4 If |P| > N then 5 Delete |P| - N solutions from $K \cap F_p$ which have the minimum distances to the hyperplane 6 Else If |P| < N then 7 Associate each member of F_1 with a solution in P: Association (P, F_1) 8 Choose N - |P| solutions one by one from F_l to construct final P: $P = Niching (P, F_b, N - |P|)$ 10 End If Return P 11

ALGORITHM 7: Selection (F, K, and N).



ALGORITHM 8: Association (P, F_l) .

the angle μ between x_j and c_k is calculated (lines 4 and 5). If $\mu < \theta(x_j), \theta(x_j)$ and $\gamma(x_j)$ are updated accordingly (lines 6–8). The association process in a 2D objective space is illustrated in Figure 6. In this example, there are 4 nondominated solutions, and one knee point has been selected and put into the next-generation population *P*. Based on the vector angle association, both x_1 and x_2 are associated with c_1, x_3 , and x_4 are associated with c_2 , and x_5 is associated with c_5 . Taking x_3 as an example, $\theta(x_3)$ and $\gamma(x_3)$ are β and 2, respectively.

3.5.4. Niche Preservation. A niche-based diversity maintenance strategy is described in Algorithm 9, which is based on the principle of maximum-vector-angle-first.

First, each solution in F_l has a flag value, which indicates whether the solution has been added into P. These flag values should be initialized to false at the beginning of the process (lines 1 and 2). Subsequently, for each $k \le (N - |P|)$, the index of the solution in F_l that has the maximum vector angle to P is calculated, which is denoted by ρ (lines 3 and 4).

Finally, based on the principle of maximum-vectorangle-first, the best solution is selected and added to *P*, as specified in Algorithm 10. When ρ is null, it means that all solutions in F_l have been added into *P*. Otherwise, x_{ρ} is added into *P* and the corresponding flag is updated to true (lines 4 and 5). When adding x_{ρ} to the population *P*, the vector angles between the remaining solutions in F_l and the



FIGURE 6: Description of updating angles in the maximum-vectorangle-first principle.

solutions in *P* must be adjusted accordingly. This adjustment can be achieved by computing the vector angle between x_{ρ} and each solution in F_l whose flag value is false (lines 7-11). The specific updating process has been described in Section 3.5.3. As shown in Figure 6, compared to the other remaining solutions, the angle between the solution c_1 in *P* and x_2 in F_l is the maximum. Based on the principle of maximum-vectorInput: *P* (population), F_l (critical front), N - |P| (solutions to be chosen) Output: *P* (next population) 1 $T = |F_l| / * |F_l|$ returns the cardinality of $F_l * /$ 2 $flag(x_j) = false, x_j \in F_p \ j = 1, 2, ..., T$ 3 For k = 1 to (N - |P|) / * Find the index with the maximum vector angle*/ 4 $\rho = \arg \max \{\theta(x_j) | x_j \in F_l \land (flag(x_j) == false)\}$ 5 $P = Maximum-Vector-Angle-First (P, \rho, T)$ 6 End for 7 Return *P*

ALGORITHM 9: Niching $(P, F_1, \text{ and } N - |P|)$.

```
Input: P (population), \rho (solution index); T (number of solutions)
Output: P (next population)
1 If \rho == null,
2
          Return P
3 End If
   P = P \cup \{\rho\} / {}^*_{\rho} is orderly added*/
4
   flag(_{\rho}) = true
5
   For j = 1 to T
6
7
          If flag(x_i) == false / x_i is a member of F_l / x_i
8
               Calculate angle between x_i and _{\rho} by Eq. (15)
9
               If angle < \theta(x_i) / x_i is associated with \rho^* / \rho^*
10
                         \theta(x_i) = angle / Update the vector angle from x_i to P */
11
                         \gamma(x_i) = |P| / Update the corresponding index.*/
12
                 End if
13
           End if
14
    End for
15
     Return P
```

ALGORITHM 10: Maximum-vector-angle-first (P, ρ , and T).

angle-first, x_2 will be added into *P*. Obviously, compared to solutions such as x_1 , x_3 , and x_4 , adding x_2 is more likely to improve the distribution of the population. In addition, the potential solutions may be obtained along the direction of x_2 . Assuming that two more solutions need to be added into *P*, x_1 and x_3 will be chosen.

In contrast, x_5 can never be considered because the vector angle between x_5 and the associated solution c_5 is zero. This means that c_5 with the same direction as x_5 has good convergence in *P*.

3.6. Time Complexity Analysis. During each evolutionary generation, KPEA performs five main operations: mating selection, preprocessing, nondominated sorting, identification of the knee points, and environmental selection. The preprocessing operation takes a runtime of $O(N \log N)$. Nondominated sorting operation takes a runtime of $O(N \log^{m-2}N)$. The identification of the knee points requires at most a runtime of $O(mN^2)$. Specifically, it takes at most a runtime of O(mN) to compute the distance between each nondominated solution and the hyperplane. And it takes a runtime of $O(mN^2)$ to identify the knee points in the nondominated solutions set.

During the environment selection, the time complexity of KPEA is mainly determined by the adaptive switching strat-

egy, as described in Algorithm 5. When the number of nondominated solutions in F_1 exceeds the population size N, the function penalty operation is triggered. First, the computation of cosine similarity requires a runtime of $O(mN^2)$. Next, it takes a runtime of $O(N^2)$ to identify the elite solutions. When the number of nondominated solutions in F_1 is less than N, the selection operation will be triggered. First, it takes a runtime of $O(N \log N)$ to identify the knee points into the next generation of the population. Next, the association and niching functions require at most a runtime of $O(mN^2)$.

Therefore, the environment selection stage requires at most a runtime of $O(mN^2)$. Overall, in each generation, the worst time complexity of KPEA is max $\{O(N \log^{m-2} N), O(mN^2)\}$.

4. Experimental Setup

The experiment to validate the performance of the proposed KPEA on MaOPs is set up in this section. First, in Section 4.1, five representative algorithms are briefly introduced, namely, k-NSGAII [36], KnEA [34], Pi-MOEA [35], hpaEA [52], and VaEA [47]. Then, the specific characteristics of the test problems in the MaF and WFG benchmark suites are listed in Section 4.2. Finally, performance metrics and

parameter settings are described in Section 4.3 and 4.4, respectively.

4.1. Comparison Algorithms. To fully validate the effectiveness of KPEA, five state-of-the-art algorithms are selected for performance comparison in this paper. The core ideas of these five algorithms are presented as follows.

- (i) *k-NSGAII* [36]. The algorithm is based on NSGA-II and introduces a knee point identification strategy, which uses a user-defined parameter η to control the width of the neighborhood of knee points. This strategy can improve the coverage of the population on the Pareto front
- (ii) KnEA [34]. The algorithm adopts an adaptive strategy to identify knee points for each nondominated front and uses a custom threshold parameter T to control the number of knee points recognized. This strategy not only enhances selection pressure on the Pareto front but also promotes population diversity
- (iii) Pi-MOEA [35]. The algorithm uses an average ranking method to identify pivot solutions within the adaptive neighborhood, which is aimed at achieving a balance between convergence and diversity
- (iv) *VaEA* [47]. The selection of solutions is based on the principles of maximum-vector-angle-first and worse-elimination without requiring additional weight vectors and parameters
- (v) hpaEA [52]. With the assistance of hyperplanes, prominent solutions are selected first during both mating selection and environmental selection, which increases the selection pressure

4.2. Description of Benchmark Problems. In this paper, 20 scalable test problems from two well-known benchmark test suites, WFG [57] and MaF [58], are used in this study to verify the performance of KPEA. The characteristics of the test problems are detailed in Table 1.

4.3. Performance Indicators. Two widely used performance metrics, namely, hypervolume (HV) [59] and inverse generational distance (IGD) [60, 61], are considered as they can simultaneously evaluate the convergence and diversity of solution sets. The IGD measures the average distance between the nondominated solution set P and the nondominated solution set P^* uniformly distributed along the true Pareto front (PF) [26]. IGD is defined as follows:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|},$$
(17)

where d(v, P) refers to the minimum distance between the solution in v and its closest solution in P. Therefore, the smaller the value of IGD, the closer the solution set is to the true PF.

HV refers to the volume of a hypercube in the objective space that is enclosed by a reference point and a set of non-

TABLE 1: Characteristics of the benchmark problems.

Problem	Characteristics
WFG1	Mixed, biased, scaled
WFG2	Convex, disconnected, multimodal, nonseparable, scaled
WFG3	Linear, degenerate, nonseparable, scaled
WFG4	Concave, multimodal, scaled
WFG5	Concave, deceptive, scaled
WFG6	Concave, nonseparable, scaled
WFG7	Concave, biased, scaled
WFG8	Concave, biased, nonseparable, scaled
WFG9	Concave, biased, multimodal, deceptive, nonseparable, scaled
MaF1	Linear
MaF4	Convex, multimodal
MaF5	Convex, biased
MaF6	Concave, degenerate
MaF7	Mixed, disconnected, multimodal
MaF8	Linear, degenerate
MaF9	Linear, degenerate
MaF10	Mixed, biased
MaF11	Convex, disconnected, nonseparable
MaF12	Concave, nonseparable, biased, deceptive
MaF13	Concave, unimodal, nonseparable, degenerate

TABLE 2: Setting of the population size.

M	Ν
3	120
5	126
8	156
10	275

dominated solutions P, which is a common indicator for measuring the performance of the algorithm. It can be defined as follows:

$$HV = volume\left(\bigcup_{i=1}^{|P|} v_i\right), \tag{18}$$

where v_i is the hypercube enclosed by $z^r = (1, 1, \dots, 1)$ and the individual *i* in *P*. The quality of the solution set can be judged by the HV value.

In order to obtain reliable statistical conclusions, the Wilcoxon rank-sum test [62] with a significance level of 0.05 is conducted to evaluate whether the proposed KPEA has a significant advantage over other comparative algorithms. The symbols "+," "-," and "=" indicate whether the results obtained by the compared algorithms are significantly better, similar, and worse than those obtained by KPEA, respectively. Moreover, the Friedman test [63] is employed in this study to obtain the average rankings of

)		
Problem	Μ	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	3	7.9923e - 1 (4.15e - 2) -	8.6653e - 1(1.70e - 2) -	8.4983e - 1 (3.52e - 2) -	1.6363e - 1 (1.66e - 1) -	4.1765e - 1 (4.49e - 2) -	9.1525e - 1(1.66e - 2)
	5	7.1681e - 1 (6.16e - 2) -	7.1316e - 1 (5.94e - 2) -	8.3420e - 1(5.31e - 2) -	$4.4593e - 1 \left(1.46e - 1 ight)$ -	3.3527e - 1(2.60e - 2) -	9.3993e - 1 (3.10e - 2)
WFGI	8	$4.9977e - 1 \left(7.70e - 2 ight)$ –	6.0197e - 1 (4.61e - 2) -	6.8013e - 1(8.49e - 2) -	$4.8784e - 1 \left(7.61e - 2 ight)$ –	2.6537e - 1 (1.71e - 2) -	8.2049e - 1 (7.08e - 2)
	10	3.4662e - 1 (3.86e - 2) -	4.1231e - 1(2.82e - 2) -	$4.8184e - 1 \left(1.06e - 1 \right) -$	2.5027e - 1(1.17e - 1) -	2.3979e - 1 (1.39e - 2) -	5.6118e - 1(6.64e - 2)
	3	8.7083e - 1(8.65e - 3) -	9.2764e - 1(2.34e - 3) -	9.3243e - 1(1.60e - 3) -	$6.6487e - 1 \left(6.45e - 2 ight) -$	9.0644e - 1(4.97e - 3) -	9.3448e - 1(1.23e - 3)
	5	9.8040e - 1 (4.46e - 3) -	9.8329e - 1(3.24e - 3) -	$9.9178e - 1 \left(1.85e - 3 ight) -$	$9.2587e - 1 \left(2.88e - 2 ight) -$	9.5001e - 1 (9.84e - 3) -	$9.9390e - 1 \left(1.20e - 3 ight)$
WFG2	8	9.8856e - 1(3.11e - 3) -	$9.8397e - 1 \left(2.59e - 3 ight) -$	9.9194e - 1 (2.23e - 3) -	$9.1348e - 1 \left(7.86e - 2 ight) -$	$9.6984e - 1 \left(1.05e - 2 ight) -$	$9.9510e - 1 \left(1.43e - 3 ight)$
	10	9.7647 <i>e</i> - 1 (8.71 <i>e</i> - 3) -	9.7811e - 1(4.15e - 3) -	9.8756e - 1 (4.00e - 3) -	7.6630e - 1(8.02e - 2) -	9.5152e - 1(1.12e - 2) -	9.9133e - 1 (2.57e - 3)
	3	2.9525e - 1 (2.88e - 2) -	3.8732e - 1(6.57e - 3) -	3.9567e - 1(2.03e - 3) =	$1.0132e - 1 \left(6.63e - 2 ight) -$	$3.5119e - 1 \left(7.47e - 3 ight) -$	3.9535e - 1 (3.12e - 3)
	5	1.8512e - 1 (1.40e - 2) +	1.1099e - 1 (3.06e - 2) =	9.3276e - 2(3.16e - 2) =	1.0222e - 1 (4.85e - 2) =	$6.5594e - 2 \left(1.72e - 2 ight) -$	1.0036e - 1 (3.57e - 2)
WFG5	8	1.9235e - 2(1.99e - 2) +	0.0000e + 0(0.00e + 0) =	0.0000e + 0(0.00e + 0) =	9.6694e - 4(4.32e - 3) =	1.5858e - 2(2.01e - 2) +	0.0000e + 0(0.00e + 0)
	10	0.0000e + 0(0.00e + 0) =	0.0000e + 0(0.00e + 0) =	0.0000e + 0(0.00e + 0) =	0.0000e + 0(0.00e + 0) =	0.0000e + 0(0.00e + 0) =	0.0000e + 0(0.00e + 0)
	3	4.4006e - 1 (2.58e - 2) -	5.3642e - 1(1.86e - 2) -	5.4980e - 1(2.15e - 2) =	$1.2891e - 1 \left(8.86e - 2 ight) -$	$5.2262e - 1 \left(4.69e - 3 ight) -$	5.5441e - 1(3.20e - 3)
	5	6.4972e - 1 (1.21e - 2) -	7.5358e - 1 (4.45e - 3) =	7.5557e - 1(9.63e - 3) =	$2.4462e - 1 \left(1.80e - 1 ight) -$	6.7881e - 1 (7.70e - 3) -	$7.5305e - 1 \left(7.94e - 3 ight)$
W FG4	8	6.4146e - 1 (2.81e - 2) -	8.8218e - 1(5.30e - 3) +	8.7175e - 1(1.12e - 2) =	5.6647e - 1(1.17e - 1) -	7.5986e - 1 (2.19e - 2) -	8.7061e - 1 (9.47e - 3)
	10	5.9718e - 1 (2.75e - 2) -	8.9185e - 1(5.77e - 3) -	8.9828e - 1(3.33e - 2) -	3.4895e - 1(5.14e - 2) -	7.6913e - 1(1.26e - 2) -	9.2021e - 1(5.09e - 3)
	б	$3.8063e - 1 \left(2.65e - 2 ight)$ -	5.0598e - 1 (2.53e - 3) +	5.0545e - 1 (3.83e - 3) +	$3.5537e - 1 \left(1.53e - 1 ight) -$	4.9911e - 1 (3.83e - 3) =	$4.9951e-1\left(5.43e-3\right)$
	5	5.9620e - 1 (1.59e - 2) -	7.1988e - 1(2.91e - 3) +	6.7709e - 1 (1.27e - 2) =	5.3354e - 1 (1.59e - 1) -	6.5401e - 1 (7.71e - 3) -	6.7536e - 1 (1.36e - 2)
601 M	8	5.7879e - 1 (3.33e - 2) -	8.3592e - 1 (4.39e - 3) +	8.0519e - 1 (1.24e - 2) =	5.9647e - 1 (2.87e - 2) -	7.2821e - 1 (2.03e - 2) -	8.0494e - 1 (1.07e - 2)
	10	5.3280e - 1 (2.41e - 2) -	8.7303e - 1(3.08e - 3) +	8.5338e - 1(8.80e - 3) =	3.1109e - 1(3.54e - 2) -	7.1555e - 1 (1.13e - 2) -	8.4634e - 1(2.47e - 2)
	3	3.8394e - 1 (3.60e - 2) -	4.9384e - 1 (1.18e - 2) =	5.0442e - 1(1.43e - 2) =	2.4253e - 1(1.61e - 1) -	4.7249e - 1(1.16e - 2) -	4.9931e - 1 (1.88e - 2)
	5	5.9743e - 1 (2.40e - 2) -	6.8892e - 1 (1.42e - 2) -	(6.9493e - 1(1.63e - 2)) =	$1.7389e - 1 \left(9.46e - 2 ight) -$	6.1901e - 1 (1.39e - 2) -	$7.0067e - 1 \left(1.52e - 2 ight)$
05J M	8	6.0435e - 1 (3.83e - 2) -	8.0255e - 1(2.06e - 2) =	8.0654e - 1(1.98e - 2) =	2.9189e - 1 (1.53e - 1) -	$7.0598e - 1 \left(2.40e - 2 ight) -$	$7.9697e - 1 \left(2.10e - 2 ight)$
	10	5.8746e - 1 (3.44e - 2) -	8.3649e - 1(1.74e - 2) =	8.3967e - 1(2.19e - 2) =	2.8991e - 1 (2.66e - 2) -	7.0153e - 1(2.15e - 2) -	8.5007e - 1(2.17e - 2)
	3	4.1956e - 1 (2.48e - 2) -	$5.4777e - 1 \left(1.83e - 3 ight) -$	$5.5524e - 1 \left(1.58e - 3 ight) -$	$2.8235e - 1 \left(1.17e - 1 ight) -$	$5.3095e - 1 \left(3.77e - 3 ight) -$	5.5732e - 1(2.70e - 3)
MEC7	5	6.3766e - 1 (1.57e - 2) -	7.7183e - 1(3.23e - 3) +	7.2358e - 1 (2.02e - 2) +	3.7400e - 1(1.20e - 1) -	6.8315e - 1 (1.51e - 2) -	$7.0416e - 1 \left(2.00e - 2 ight)$
DIM.	8	5.8493e - 1 (4.02e - 2) -	8.9217e - 1(5.80e - 3) +	8.8207e - 1(1.32e - 2) =	5.6287e - 1 (6.06e - 2) -	7.7972e - 1 (3.65e - 2) -	8.8382e - 1 (7.86e - 3)
	10	5.5977e - 1 $(1.79e - 2) -$	9.2574e - 1(6.76e - 3) =	9.1000e - 1(1.49e - 2) -	3.3957e - 1 (2.32e - 2) -	7.7302e - 1 (1.85e - 2) -	9.2624e - 1(7.90e - 3)

Problem	M	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	3	3.7102e - 1 (2.18e - 2) -	4.4987e - 1 (2.89e - 3) -	4.6793e - 1(1.58e - 3) =	4.4079e - 2(9.88e - 2) -	$4.4274e - 1 \left(4.33e - 3 ight) -$	4.6704e - 1(3.01e - 3)
	5	5.2630e - 1 (1.20e - 2) -	$6.2253e - 1 \left(7.92e - 3 ight) -$	6.3323e - 1(1.72e - 2) -	4.2707e - 2(5.26e - 2) -	5.5209e - 1 (1.22e - 2) -	$6.4386e - 1 \left(7.16e - 3 ight)$
WFG&	8	5.3046e - 1 (1.23e - 2) -	7.2344e - 1 (1.63e - 2) -	7.2962e - 1 (1.86e - 2) =	5.2007e - 1(1.25e - 1) -	6.1173e - 1 (2.58e - 2) -	$7.3440e - 1 \left(9.52e - 3 \right)$
	10	5.5626e - 1 (2.12e - 2) -	7.5604e - 1 (1.37e - 2) -	7.8833e - 1(1.80e - 2) -	$2.8945e - 1 \left(2.94e - 2 ight) -$	6.3981e - 1 (2.80e - 2) -	8.0096e - 1 (2.12e - 2)
	3	4.2372e - 1(3.61e - 2) -	5.3077e - 1 (2.66e - 3) +	5.2913e - 1(3.23e - 2) =	$2.1958e - 1 \left(6.50e - 2 ight) -$	5.0551e - 1 (1.04e - 2) -	5.3632e - 1 (5.50e - 3)
	5	5.7689e - 1 (3.05e - 2) -	7.2456e - 1 (2.64e - 2) +	7.0197e - 1(3.35e - 2) =	3.9623e - 1(1.70e - 1) -	5.7333e - 1 (3.49e - 2) -	7.1115e - 1 (1.22e - 2)
WFG9	8	$4.8285e - 1 \left(4.03e - 2 \right) -$	8.2677e - 1(3.07e - 2) +	7.8731e - 1 (2.08e - 2) =	6.3628e - 1 (5.68e - 2) -	5.9553e - 1 (3.58e - 2) -	$7.8321e - 1 \big(4.65e - 2\big)$
	10	5.4468e - 1 (3.22e - 2) -	8.3303e - 1(2.97e - 2) =	8.1791e - 1(2.04e - 2) =	$3.1355e - 1 \left(3.69e - 2 ight) -$	6.1249e - 1 (4.64e - 2) -	$8.2342e - 1 \big(1.40e - 2 \big)$
=/-/+		2/33/1	10/17/9	2/13/21	0/33/3	1/33/2	

TABLE 3: Continued.

Problem	Μ	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	ε	5.8860e - 1 (1.02e - 1) - 102405 + 0(1.62e - 1) - 102405 + 0(1.62e - 1) - 102405 + 0(1.62e - 1) - 100405 + 0(1.62e - 1)) - 100405 + 0(1.62e - 1) - 100405 + 0(1.62e - 1)) - 100405 + 0(1.62e - 1) - 100405 + 0(1.62e - 1))	2.5276e - 1 (2.29e - 2) = 7.5526 - 1 (2.29e	3.0350e - 1 (5.05e - 2) - 664162 - 16622 - 20000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000	2.0717e + 0(3.71e - 1) - 2.07202 + 0(1.472 + 0)	1.1143e + 0(1.05e - 1) - 1143e + 0(1.05e - 1) - 1143e + 0(1.05e - 1) - 1143e + 0(1.05e - 1) - 1142e + 0(1.05e - 1) - 1042e + 0(1.05e -	2.3463e - 1 (2.79e - 2)
WFG1	റം	-1.03406 + 0(1.346 - 1) - 00000000000000000000000000000000	-(2 - 300.6) - 1 - 3800.6.7	-(2 - 3000 - 1) - 3000 - 20000 - 2000 - 2000 - 2000 - 2000 - 2000 - 2000 - 20	$2.0700e \pm 0(1.47e \pm 0) - 2.379e \pm 0(8.98e - 1) - 2.379e \pm 0(8.98e - 1) - 2.379e \pm 0.080e - 1) - 2.379e + 0.080e $	1.82916 ± 0 (1.01 $e - 1$) $- 1.82916 \pm 0$ (6.4 $e - 2$) $- 2.866 \pm 0$	(2 - 3c4) = 1 (3.426 - 2) (2
	° 10	2.6103e + 0(1.38e - 1) - 2.6103e + 0(1.38e - 1) - 10000000000000000000000000000000	2.1096e + 0(1.05e - 1) - 2.1096e + 0(1.05e -	1.9574e + 0(1.79e - 1) - 1.9566 + 0(1.79e - 1) -	3.0553e + 0(3.03e - 1) -	2.9065e + 0(7.07e - 2) - 2.9065e + 0(7.07e - 2) - 2.9065e + 0(7.07e - 2) - 20065e + 0(7.07e - 2) - 20065e + 0(7.07e - 2) - 20065e + 0(7.07e - 2)) - 200656e + 0(7.07e - 2)	1.7017e + 0(1.49e - 1)
	6	5.1025e - 1(3.05e - 2) -	1.8508e - 1 (6.50e - 3) =	1.8248e - 1(1.01e - 2) +	1.1824e + 0(2.88e - 1) -	1.6961e - 1 (5.53e - 3) +	1.8600e - 1(5.79e - 3)
	Ŋ	7.5021e - 1(3.28e - 2) -	5.4313e - 1(2.80e - 2) =	5.8522e - 1(2.25e - 2) -	1.1054e + 0(3.40e - 1) -	5.0043e - 1(1.29e - 2) +	5.3209e - 1(2.09e - 2)
WFG2	×	1.4018e + 0(5.45e - 2) -	1.0697e + 0(2.39e - 2) =	1.0850e + 0(3.26e - 2) -	2.0717e + 0(5.22e - 1) -	9.5142e - 1(4.33e - 2) +	1.0568e + 0(1.85e - 2)
	10	1.5624e + 0(5.16e - 2) -	1.1578e + 0(2.82e - 2) =	1.1899e + 0(6.76e - 2) =	1.8478e + 0(7.03e - 1) -	1.2716e + 0(3.96e - 2) -	1.1759e + 0(3.17e - 2)
	6	5.2940e - 1(1.44e - 1) -	9.4299e - 2(1.11e - 2) -	8.1490e - 2(6.12e - 3) -	2.9862e + 0(6.24e - 1) -	1.5674e - 1 $(1.30e - 2) -$	7.5795e - 2(6.26e - 3)
	5	6.3059e - 1 (1.49e - 1) -	5.0196e - 1(1.15e - 1) =	4.8471e - 1(1.27e - 1) =	4.4171e + 0(2.08e + 0) -	$6.5987e - 1 \left(7.46e - 2 ight) -$	$5.1572e - 1 \; (1.54e - 1)$
WFG3	8	1.1936e + 0(2.51e - 1) =	1.1017e + 0(1.67e - 1) =	1.1109e + 0(2.45e - 1) =	3.2819e + 0(3.36e + 0) -	1.5478e + 0(1.70e - 1) -	$1.0966e+0\left(3.46e-1\right)$
	10	1.3744e + 0(3.13e - 1) =	1.4816e + 0(2.39e - 1) -	1.4150e + 0(1.39e - 1) -	1.6785e + 0(1.51e - 1) -	1.5482e + 0(2.18e - 1) -	1.2814e + 0(2.47e - 1)
	6	7.1881e - 1 (1.26e - 1) -	2.4712e - 1(7.89e - 3) -	2.2804e - 1(9.09e - 3) -	3.5970e + 0(8.92e - 1) -	2.3578e - 1 (4.07e - 3) -	2.0706e - 1 (7.01e - 3)
	5	1.6291e + 0(6.10e - 2) -	1.2119e + 0(1.38e - 2) -	1.1395e + 0(3.41e - 2) =	5.1480e + 0(2.02e + 0) -	1.1950e + 0(1.14e - 2) -	1.1263e + 0(2.81e - 2)
WFG4	8	3.8985e + 0(7.41e - 2) -	3.2822e + 0(3.05e - 2) =	3.0568e + 0(9.16e - 2) +	4.7308e + 0(1.21e + 0) -	3.3021e + 0(3.13e - 2) =	3.2628e + 0(1.46e - 1)
	10	4.9675e + 0(6.53e - 2) -	4.1444e + 0(3.60e - 2) =	4.0909e + 0(3.37e - 1) +	9.3937e + 0(1.21e + 0) -	4.8852e + 0(3.55e - 2) -	4.1295e + 0(1.35e - 1)
	3	7.9537e - 1 (1.04e - 1) -	2.5593e - 1 (6.54e - 3) -	2.3206e - 1 (4.73e - 3) -	1.6145e + 0(1.18e + 0) -	2.3844e - 1(2.34e - 3) -	2.2002e - 1 (6.55e - 3)
MTEO E	5	1.7181e + 0(8.19e - 2) -	1.1976e + 0(1.90e - 2) -	1.1131e + 0(3.44e - 2) =	2.6448e + 0(1.92e + 0) -	1.1884e + 0(8.77e - 3) -	1.1124e + 0(2.55e - 2)
CD1 M	×	3.8137e + 0(7.73e - 2) -	3.2910e + 0(2.86e - 2) -	3.0747e + 0(7.86e - 2) =	4.2447e + 0(1.34e - 1) -	3.3345e + 0(2.17e - 2) -	3.1129e + 0(8.21e - 2)
	10	5.0150e + 0(6.48e - 2) -	$4.1436e + 0 \left(4.31e - 2\right) -$	4.0455e + 0(6.71e - 2) -	8.2162e + 0(1.04e + 0) -	4.9026e + 0(4.46e - 2) -	4.0425e + 0(3.54e - 1)
	3	7.2976e - 1(1.21e - 1) -	2.7872e - 1(1.23e - 2) -	2.3569e - 1(1.20e - 2) +	1.9440e + 0(1.40e + 0) -	2.7832e - 1 (1.28e - 2) -	2.5265e - 1 (1.30e - 2)
	Ŋ	1.6532e + 0(7.64e - 2) -	1.2356e + 0(1.64e - 2) -	1.1440e + 0(3.02e - 2) =	4.9760e + 0(1.30e + 0) -	1.2337e + 0(1.14e - 2) -	1.1403e + 0(1.87e - 2)
WFG0	8	3.8452e + 0(1.02e - 1) -	3.4528e + 0(6.06e - 2) =	3.2090e + 0(9.17e - 2) +	7.7825e + 0(3.25e + 0) -	3.4112e + 0(1.49e - 1) =	3.5154e + 0(3.77e - 1)
	10	4.9986e + 0(6.23e - 2) -	4.3440e + 0(6.64e - 2) =	4.3754e + 0(3.89e - 1) +	8.1580e + 0(7.98e - 1) -	5.0176e + 0(4.41e - 2) -	4.5569e + 0(3.33e - 1)
	6	8.5091e - 1(1.11e - 1) -	2.3750e - 1 (5.88e - 3) -	2.1601e - 1(4.93e - 3) -	2.1061e + 0(1.08e + 0) -	2.2974e - 1(4.91e - 3) -	2.0348e - 1(5.41e - 3)
	5	1.7699e + 0(8.08e - 2) =	1.2095e + 0(1.63e - 2) +	1.3471e + 0(1.63e - 1) +	3.4703e + 0(1.25e + 0) -	1.2122e + 0(1.09e - 2) +	1.6581e + 0(2.51e - 1)
15J M	×	4.0868e + 0(8.95e - 2) -	3.2499e + 0(2.93e - 2) -	3.1077e + 0(8.97e - 2) =	4.5537e + 0(3.32e - 1) -	3.3076e + 0(1.40e - 1) -	3.1905e + 0(1.62e - 1)
	10	5.1774e + 0(8.17e - 2) -	4.1428e + 0(5.60e - 2) =	4.2022e + 0(2.48e - 1) =	8.5753e + 0(6.34e - 1) -	4.8932e + 0(4.33e - 2)	4.1245e + 0(6.55e - 2)
	3	6.6030e - 1 (1.29e - 1) -	3.2937e-1 (8.43e-3) –	2.9268e - 1 (4.55e - 3) +	3.5523e + 0(8.05e - 1) -	3.2313e - 1 (6.56e - 3) -	3.0248e - 1(5.58e - 3)
WEG8	Ŋ	1.6796e + 0(4.62e - 2) -	1.2737e+0 (1.58e-2) -	1.1788e + 0(3.01e - 2) -	4.5928e + 0(9.17e - 1) -	1.3140e + 0(1.81e - 2) -	1.1674e + 0(1.41e - 2)
00.14	8	4.0973e + 0(6.59e - 2) -	3.4668e+0 (5.41e-2) -	3.4445e + 0(1.80e - 1) -	5.1918e + 0(1.93e + 0) -	3.6081e + 0(3.66e - 2) -	3.2830e + 0(1.01e - 1)
	10	5.1965e + 0(5.33e - 2) -	4.3376e+0 (6.49e-2) +	4.6386e + 0(2.97e - 1) =	9.8605e + 0(5.72e - 1) -	5.2287e + 0(4.11e - 2) -	4.6565e + 0(1.52e - 1)

TABLE 4: Mean and standard deviation values of IGD for KPEA and state-of-art algorithms on WFG problems.

Problem	Μ	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	ю	$7.4127e - 1 \left(1.80e - 1 ight) -$	2.2145e-1 (6.30e-3) -	2.1681e - 1(3.85e - 2) -	2.0376e + 0(3.69e - 1) -	2.3054e - 1 (9.36e - 3) -	1.9810e - 1 (6.52e - 3)
	S	1.6999e + 0(6.68e - 2) -	1.1428e+0 (1.57e-2) -	1.0933e + 0(2.98e - 2) =	3.3204e + 0(1.15e + 0) -	1.2154e + 0(2.87e - 2) -	1.0917e + 0(1.67e - 2)
WFGY	8	4.1823e + 0(9.62e - 2) -	3.1190e+0 (3.61e-2) -	3.1037e + 0(8.67e - 2) -	3.9309e + 0(3.86e - 1) -	3.3707e + 0(6.48e - 2) -	3.0483e + 0(5.88e - 2)
	10	5.3119e + 0(6.75e - 2) -	3.9926e+0 (3.81e-2) +	4.2361e + 0(2.52e - 1) =	9.1295e + 0(8.33e - 1) -	4.9137e + 0(4.53e - 2) -	$4.2126e + 0 \bigl(1.17e - 1\bigr)$
=/-/+		0/33/3	3/21/12	8/16/12	0/36/0	4/30/2	

TABLE 4: Continued.

					c	T	
Problem	Μ	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	3	1.5219e - 1 (3.70e - 2) -	2.1690e - 1(1.88e - 3) -	2.1891e - 1(6.79e - 4) -	2.2054e - 1(5.97e - 4) +	2.1662e - 1 (8.63e - 4) -	2.1988e - 1 (6.02e - 4)
MALE1	5	6.3281e - 3(1.01e - 3) -	1.0598e - 2(1.83e - 4) -	1.0047e - 2(2.25e - 4) -	9.5277e - 3(2.36e - 4) -	8.3063e - 3 (2.84e - 4) -	1.0997e - 2(2.40e - 4)
Marı	8	1.1000e - 5(4.19e - 6) -	2.3299e - 5(2.18e - 6) -	2.2538e - 5(2.67e - 6) -	2.0550e - 5(3.33e - 6) -	1.0195e - 5(2.65e - 6) -	3.1775e - 5(2.88e - 6)
	10	1.9360e - 7(3.97e - 7) -	2.2887e - 7(1.46e - 7) -	3.1723e - 7(1.50e - 7) -	1.5885e - 7(2.76e - 7) -	4.0284e - 8(1.80e - 7) -	5.5111e - 7(2.81e - 7)
	ŝ	1.6350e - 1 (2.27e - 1) -	4.4116e - 1(1.16e - 1) -	3.6413e - 1(2.00e - 1) -	3.2979e - 1(1.97e - 1) -	7.4703e - 2(1.77e - 1) -	4.5241e - 1 (1.50e - 1)
	2	7.2382e - 3(1.40e - 2) -	2.0212e - 2(2.49e - 2) -	2.2498e - 2(3.20e - 2) -	3.2080e - 3(9.85e - 3) -	5.9175e - 3(1.90e - 2) -	4.6754e - 2(3.60e - 2)
Mar4	8	9.2592e - 7(4.14e - 6) -	0.0000e + 0(0.00e + 0) -	2.7612e - 4(4.35e - 4) -	1.5883e - 5(4.12e - 5) -	1.9949e - 7 (6.65e - 7) -	8.3691e - 4(8.10e - 4)
	10	2.1465e - 9(7.99e - 9) -	0.0000e + 0(0.00e + 0) -	6.7485e - 6(2.41e - 5) =	1.7541e - 7 (7.84e - 7) -	1.9186e - 9(8.58e - 9) -	8.1403e - 6(2.07e - 5)
	3	$4.1654e - 1 \left(8.04e - 2 \right) -$	5.2349e - 1 (1.02e - 1) -	5.2869e - 1(1.04e - 1) =	$3.9850e - 1 \left(1.35e - 1 ight) -$	$4.8847e - 1 \left(1.26e - 1 \right) -$	$5.3746e - 1 \left(5.85e - 2 ight)$
	Ŋ	6.4424e - 1 (1.32e - 2) -	7.7631e - 1(5.06e - 3) =	7.6272e - 1 (1.95e - 2) -	6.0635e - 1(1.38e - 1) -	7.3960e - 1 (6.85e - 3) -	7.7107e - 1 ($2.58e - 2$)
Clari	8	2.1369e - 4 (9.56e - 4) -	9.0603e - 1(4.34e - 3) -	8.8761e - 1(1.12e - 2) -	7.7748e - 1 (5.49e - 2) -	8.6000e - 1 (2.62e - 2) -	9.1176e - 1(8.41e - 3)
	10	0.0000e + 0(0.00e + 0) - 0	$9.5206e - 1 \left(3.74e - 3 ight) -$	9.1107e - 1(2.47e - 2) -	8.2543e - 1(5.01e - 2) -	7.0388e - 1 (6.17e - 2) -	9.5801e - 1 (6.93e - 3)
	3	1.8023e - 1 (3.54e - 2) -	$1.8050e - 1 \left(1.86e - 2 ight) -$	1.9861e - 1(4.10e - 4) -	1.9981e - 1(1.04e - 4) +	1.9659e - 1 (1.28e - 3) -	1.9890e - 1 (2.06e - 4)
	5	1.2132e - 1 (1.19e - 2) -	$1.2790e - 1 \left(7.02e - 4 ight) -$	1.2831e - 1 (7.90e - 4) -	1.2939e - 1 (3.91e - 4) +	1.2720e - 1 (1.02e - 3) -	1.2882e - 1 (2.55e - 4)
Maro	8	3.3740e - 2(4.73e - 2) -	5.2829e - 2(5.42e - 2) -	9.9379e - 2(2.35e - 2) +	1.0063e - 1(2.37e - 2) =	6.3245e - 2(5.30e - 2) -	9.5296e - 2 (3.26e - 2)
	10	0.0000e + 0(0.00e + 0) - 0	0.0000e + 0(0.00e + 0) - 0	2.5867e - 2(4.42e - 2) =	3.0225e - 2(4.74e - 2) =	1.4859e - 2(3.63e - 2) =	$2.0106e - 2\left(4.13e - 2\right)$
	б	$1.7773e - 1 \left(1.73e - 2 ight) -$	2.7589e - 1(8.57e - 3) +	2.6950e - 1(1.25e - 2) =	2.6988e - 1 (9.12e - 3) =	$2.5203e - 1 \left(7.80e - 3 ight) -$	2.6615e - 1 (2.06e - 2)
	5	1.5133e - 1 (2.95e - 2) -	2.5023e - 1 (5.52e - 3) =	1.9700e - 1 (2.06e - 2) -	2.3099e - 1 (1.40e - 2) -	1.5718e - 1 (1.87e - 2) -	$2.4103e - 1 \left(1.75e - 2 ight)$
Mar/	8	3.9354e - 6(6.09e - 6) -	1.3944e - 1(1.83e - 2) +	5.3852e - 2(5.07e - 3) -	1.1778e - 1 (2.52e - 2) +	4.0480e - 3(3.97e - 3) -	7.7005e - 2(1.87e - 2)
	10	1.5466e - 8(3.45e - 8) -	4.5983e - 2(1.94e - 2) =	3.2404e - 2(2.24e - 2) -	1.1475e - 1(1.77e - 2) +	8.4559e - 4(1.15e - 3) -	5.7313e - 2(9.76e - 3)
	3	1.1156e - 1 (6.69e - 2) -	$1.3830e - 1 \left(6.92e - 2 ight) -$	$1.0925e - 1 \left(6.52e - 2 ight) -$	$1.5173e - 1 \left(8.34e - 2 ight) -$	$1.3188e - 1 \left(7.82e - 2 ight) -$	2.1181e - 1 (5.46e - 2)
MACEO	S.	5.5082e - 2(3.29e - 2) =	5.4001e - 2(3.77e - 2) =	6.4162e - 2(3.45e - 2) =	5.7578e - 2(3.03e - 2) =	4.2186e - 2(3.66e - 2) -	7.0910e - 2(3.54e - 2)
IVIALO	×	1.0488e - 2(8.47e - 3) =	1.2220e - 2(1.11e - 2) =	1.2215e - 2(7.35e - 3) =	1.2189e - 2(8.57e - 3) =	1.0881e - 2(8.52e - 3) =	1.3337e - 2(1.05e - 2)
	10	3.1893e - 3(3.01e - 3) =	3.1394e - 3(3.18e - 3) =	4.1888e - 3(2.76e - 3) =	3.0522e - 3(2.89e - 3) =	3.4280e - 3(2.27e - 3) =	4.4683e - 3(4.01e - 3)
	3	6.4312e - 1(1.05e - 1) -	$4.7809e - 1 \left(1.71e - 1 ight) -$	7.5994e - 1 (4.61e - 2) -	7.9322e - 1 (5.65e - 2) =	5.3049e - 1 (1.68e - 1) -	8.0297e - 1 (1.26e - 2)
MAFO	5	$1.4392e - 1 \left(6.76e - 2 ight) -$	$1.8494e - 1 \left(4.34e - 2 ight) -$	1.8294e - 1 (4.13e - 2) -	2.7154e - 1(3.24e - 2) =	7.3405e - 2(8.11e - 2) -	$2.6738e - 1 \left(1.58e - 2 ight)$
INIAL	8	0.0000e + 0(0.00e + 0) -	8.9774e - 4(1.95e - 3) -	1.8513e - 3(2.16e - 3) -	7.1614e - 3(1.02e - 2) -	1.8034e - 3(4.38e - 3) -	3.3563e - 2(3.16e - 3)
	10	3.5457e - 5(9.14e - 5) -	2.9432e - 5(1.32e - 4) -	2.9196e - 4(3.12e - 4) -	8.8388e - 3(3.26e - 3) =	9.3511e - 3(3.56e - 3) =	8.6205e - 3(1.19e - 3)

				I ABLE 5: Continued.			
Problem	M	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	3	8.0737e - 1 (3.86e - 2) -	8.5207e - 1 (4.32e - 2) -	8.7596e - 1 (1.82e - 2) -	5.3517e - 1 (9.89e - 2) -	$4.2780e - 1 \left(4.03e - 2 \right) -$	9.1376e - 1 (1.92e - 2)
	Ŋ	$7.0120e - 1 \left(7.79e - 2 ight) -$	$7.5349e - 1 \left(4.11e - 2 \right) -$	8.6799e - 1 (5.38e - 2) -	$5.4315e - 1 \left(5.34e - 2 ight) -$	3.2681e - 1(1.57e - 2) -	9.3890e - 1 (3.20e - 2)
Mariu	8	5.1604e - 1 (8.08e - 2) -	5.9971e - 1 (4.60e - 2) -	8.2349e - 1 (6.09e - 2) =	$4.6576e - 1 \left(5.84e - 2 \right) -$	2.6138e - 1 (3.29e - 3) -	8.2514e - 1 (6.79e - 2)
	10	3.4038e - 1(4.49e - 2) -	4.2104e - 1(2.77e - 2) -	7.9320e - 1 (6.54e - 2) +	4.8083e - 1(7.15e - 2) -	2.4531e - 1 (1.72e - 2)	5.6479e - 1 (9.82e - 2)
	3	8.7060e - 1 (1.17e - 2) -	9.2784e - 1(1.76e - 3) -	9.3154e - 1(1.72e - 3) -	9.2308e - 1 (3.44e - 3) -	9.0705e - 1 (5.74e - 3) -	9.3479e - 1 (1.03e - 3)
111 J.V.	S	9.8074e - 1(3.73e - 3) -	$9.8426e - 1 \left(2.59e - 3 ight) -$	$9.9234e - 1 \left(1.64e - 3 ight) -$	$9.7206e - 1 \left(7.66e - 3 ight) -$	9.5271e - 1 (1.18e - 2) -	$9.9377e - 1 \left(9.73e - 4 ight)$
Marii	8	9.8846e - 1 (3.39e - 3) -	9.8195e - 1 (3.85e - 3) -	9.9266e - 1 (2.32e - 3) -	$9.5138e - 1 \left(9.27e - 3 ight) -$	$9.4293e - 1 \left(1.85e - 2 ight) -$	9.9471e - 1(1.79e - 3)
	10	$9.7417e - 1 \left(7.06e - 3 ight) -$	9.7857e - 1 (4.68e - 3) -	9.9010e - 1(3.23e - 3) =	$9.6377e - 1 \left(1.02e - 2 ight) -$	$9.5395e - 1 \left(1.29e - 2 ight) -$	9.9092e - 1 (2.46e - 3)
	3	4.1692e - 1(6.46e - 2) -	5.2386e - 1 (2.74e - 2) -	5.3095e - 1 (4.59e - 3) -	5.1836e - 1 (2.68e - 2) -	$5.0289e - 1 \left(7.84e - 3 ight) -$	5.3619e - 1 (4.26e - 3)
	5	5.5755e - 1 (5.08e - 2) -	7.2927e - 1 (7.24e - 3) +	6.8676e - 1 (1.38e - 2) -	6.8031e - 1 (1.46e - 2) -	5.6878e - 1 (2.55e - 2) -	$7.0763e - 1 \left(1.08e - 2 ight)$
Mar 12	8	4.7393e - 1 (4.73e - 2) -	8.0365e - 1 (4.93e - 2) +	$7.1568e - 1 \left(6.54e - 2 \right) -$	6.0024e - 1(3.97e - 2) -	6.0264e - 1 (4.17e - 2) -	$7.9307e - 1 \left(1.46e - 2 ight)$
	10	5.3469e - 1 (2.39e - 2) -	8.4051e - 1(1.92e - 2) +	7.5303e - 1 (3.86e - 2) -	6.5491e - 1 (2.89e - 2) -	6.1413e - 1 (4.94e - 2) -	8.2902e - 1(1.37e - 2)
	3	2.6807e - 1 $(3.87e - 2) -$	$4.5762e - 1 \left(9.03e - 3 ight) -$	$4.7636e - 1 \left(1.74e - 2 ight) -$	4.9983e - 1(1.41e - 2) =	$4.6880e - 1 \left(1.70e - 2 ight) -$	5.0276e - 1 (1.22e - 2)
011 J.	S	2.1896e - 1 (1.06e - 2) -	2.1548e - 1(1.40e - 2) -	2.2809e - 1 (1.38e - 2) -	2.3806e - 1 (1.78e - 2) -	2.1085e - 1 (1.60e - 2) -	2.5636e - 1 (8.44e - 3)
Mar 15	8	1.3697e - 1 (7.97e - 3) -	$1.1544e - 1 \left(1.87e - 2 ight) -$	1.3604e - 1 (7.69e - 3) -	$1.4317e - 1 \left(9.02e - 3 ight) -$	1.2017e - 1(2.15e - 2) -	$1.6289e - 1 \left(4.20e - 3 \right)$
	10	1.1848e - 1(8.18e - 3) -	1.0885e - 1(1.16e - 2) -	1.0926e - 1 (7.06e - 3) -	1.1439e - 1(1.32e - 2) -	8.7533e - 2(1.60e - 2) -	1.3634e - 1 (1.59e - 3)
=/-/+		0/41/3	5/33/6	2/33/9	5/29/10	0/40/4	

TABLE 5: Continued.

					2	4	
Problem	Μ	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	ю	1.3706e - 1 (7.03e - 2) -	4.3883e - 2(2.74e - 3) =	$4.4146e - 2\left(8.23e - 4\right) -$	$4.1959e - 2\left(5.91e - 4\right) +$	$4.2516e - 2\left(5.28e - 4\right) +$	4.3241e - 2(8.17e - 4)
LT.L.	5	1.7640e - 1 (1.92e - 2) -	1.2393e - 1 (1.62e - 3) -	1.3154e - 1(2.81e - 3) -	1.3789e - 1 (2.27e - 3) -	1.3903e - 1 (2.09e - 3) -	1.2206e - 1 (2.51e - 3)
Marı	8	2.5964e - 1 (9.81e - 3) -	2.0822e - 1(3.20e - 3) -	2.2142e - 1(6.20e - 3) -	2.3615e - 1(1.95e - 3) -	2.4331e - 1 (2.82e - 3) -	$1.9997e - 1 \left(1.94e - 3 ight)$
	10	$2.5879e - 1 \left(7.02e - 3 ight) -$	2.2952e - 1(6.21e - 3) -	2.6266e - 1(5.91e - 3) -	2.6073e - 1(3.43e - 3) -	2.7490e - 1 (4.48e - 3) -	2.1629e - 1(1.76e - 3)
	3	4.4728e + 0(4.62e + 0) -	6.6254e - 1 (4.34e - 1) -	1.1544e + 0(1.81e + 0) -	1.0843e + 0(1.22e + 0) -	2.3789e + 1 (1.80e + 1) -	5.3802e - 1 (6.35e - 1)
	5	4.2209e + 1 (3.10e + 1) -	1.8388e + 1(1.91e + 1) -	1.0116e + 1 (7.90e + 0) -	2.4496e + 1 (1.65e + 1) -	1.2139e + 2(9.70e + 1) -	6.0429e + 0(6.80e + 0)
Mar4	8	3.4828e + 2(2.43e + 2) -	3.5033e + 2(2.80e + 2) -	1.0153e + 2(9.44e + 1) -	3.3545e + 2(3.06e + 2) -	2.3475e + 2(1.77e + 2) -	5.6954e + 1 (7.72e + 1)
	10	1.5062e + 3(9.62e + 2) -	2.0925e + 3(1.30e + 3) -	8.2097e + 2(8.33e + 2) =	1.4983e + 3(9.26e + 2) -	3.1758e + 3(2.67e + 3) -	5.0026e + 2(4.69e + 2)
	3	$1.0653e+0\left(9.08e-1\right)-$	5.2659e - 1(1.03e + 0) -	5.1328e - 1((1.03e + 0)) =	1.5113e + 0(1.21e + 0) -	7.6498e - 1(1.12e + 0) =	3.6738e - 1 (3.25e - 1)
	5	3.1729e + 0(1.79e - 1) -	2.4414e + 0(7.17e - 2) -	2.3438e + 0(2.68e - 1) +	3.7936e + 0(1.72e + 0) -	2.3488e + 0(5.34e - 2) +	2.3799e + 0(8.51e - 1)
CABIM	8	3.5165e + 1 (2.80e + 0) -	2.0408e + 1(1.11e + 0) -	1.9405e + 1 (4.03e + 0) -	2.3445e + 1 (5.89e + 0) -	1.8715e + 1 (1.50e + 0) -	1.7302e + 1 (4.28e + 0)
	10	9.3598e + 1 (5.88e + 0) -	5.7716e + 1 (3.04e + 0) -	7.7476e + 1 (8.32e + 0) -	7.4645e + 1(6.35e + 0) -	7.6310e + 1 (6.50e + 0) -	5.4777e + 1 (4.58e + 0)
	3	5.4825e - 2(9.75e - 2) -	2.9621e - 2(3.30e - 2) -	6.0431e - 3(3.51e - 4) =	4.6042e - 3(1.07e - 4) +	7.7203e - 3(1.38e - 3) -	5.8587e - 3(3.59e - 4)
ALLC V	5	3.5663e - 2(4.78e - 2) -	6.8522e - 3(2.61e - 3) -	5.9527e - 3(3.75e - 4) -	$5.4502e - 3 \left(1.98e - 4 ight) -$	7.8671e - 3(2.82e - 3) -	4.8327e - 3(2.76e - 4)
Maro	8	6.8028e - 1 (5.64e - 1) -	8.8652e - 1(1.05e + 0) -	2.5870e - 2(8.96e - 2) +	1.9638e - 1 (8.46e - 1) -	$7.5251e - 1 \left(1.25e + 0 ight) -$	9.3761e - 2(2.90e - 1)
	10	2.1392e + 0(9.30e - 1) =	4.9617e + 0(3.10e + 0) -	4.2340e - 1(2.90e - 1) +	2.1579e + 0(1.94e + 0) =	2.6674e + 0(1.63e + 0) -	1.5078e + 0(1.06e + 0)
	ю	4.1816e - 1 (6.96e - 2) -	7.8348e - 2(6.70e - 2) +	1.0414e - 1(1.07e - 1) =	9.6664e - 2(6.68e - 2) +	9.7175e - 2(6.73e - 2) +	$1.5317e-1\left(1.93e-1 ight)$
M.T.T	5	6.6882e - 1 (9.97e - 2) -	3.0806e - 1(3.14e - 2) +	3.4948e - 1(7.95e - 2) =	5.3419e - 1 (2.89e - 1) -	5.2444e - 1(8.14e - 2) -	4.5676e - 1 (2.18e - 1)
MIAF /	8	2.6675e + 0(7.84e - 1) -	6.8468e - 1(1.15e - 1) +	7.6474e - 1(6.64e - 2) =	1.0871e + 0(2.06e - 1) -	1.6125e + 0(1.99e - 1) -	7.8131e - 1(1.58e - 1)
	10	6.8030e + 0(2.46e + 0) -	9.9676e - 1 (6.84e - 2) -	1.1721e + 0(5.49e - 2) -	1.6276e + 0(2.36e - 1) -	2.3059e + 0(4.43e - 1) -	9.3142e - 1(2.72e - 1)
	3	5.6149e - 1 (4.09e - 1) -	$4.8657e - 1 \left(2.82e - 1 ight) -$	$5.2346e - 1 \left(2.56e - 1 ight) -$	5.2134e - 1(6.28e - 1) -	5.2018e - 1 (4.12e - 1) -	2.3977e - 1(1.70e - 1)
MAFe	5	6.2559e - 1(4.62e - 1) =	6.3346e - 1(3.49e - 1) =	6.7845e - 1(7.82e - 1) =	5.7294e - 1 (2.55e - 1) -	7.5741e - 1 (4.33e - 1) -	$4.3327e - 1 \left(3.19e - 1 \right)$
INIALO	8	8.2220e - 1(5.99e - 1) =	8.4196e - 1(5.82e - 1) =	7.3656e - 1 (4.52e - 1) =	8.3142e - 1(4.83e - 1) =	8.6695e - 1(5.10e - 1) =	8.1544e - 1(5.75e - 1)
	10	1.2371e + 0(1.17e + 0) =	1.2061e + 0(1.08e + 0) =	7.3359e - 1 (3.79e - 1) =	1.2292e + 0(1.05e + 0) =	9.2725e - 1(5.66e - 1) =	1.3479e + 0(1.99e + 0)
	3	$2.8575e - 1 \left(1.54e - 1 ight) -$	1.2577e + 0(3.60e + 0) -	1.6012e - 1(6.35e - 2) -	1.1485e - 1 (6.41e - 2) =	5.5967e - 1 (4.86e - 1) -	1.0658e - 1 (1.84e - 2)
MAFO	5	$6.6227e - 1 \left(3.09e - 1 ight) -$	5.0349e - 1(2.02e - 1) -	4.9496e - 1 (2.12e - 1) -	1.9826e - 1(1.05e - 1) =	1.2197e + 0(5.94e - 1) -	1.9961e - 1 (3.51e - 2)
C. TRIAT	8	2.0724e + 1(8.54e + 0) -	4.9221e + 0(2.39e + 0) -	2.6122e + 0(1.10e + 0) -	2.1318e + 0(1.40e + 0) -	4.3658e + 0(2.23e + 0) -	$3.3167e - 1 \left(6.10e - 2 ight)$
	10	5.3959e + 1 (4.87e + 1) -	2.8632e + 1(2.53e + 1) -	3.0041e + 0(1.55e + 0) -	4.1208e - 1(1.35e - 1) +	5.2586e - 1(4.46e - 1) =	5.0146e - 1(1.00e - 1)

Problem	М	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
	3	5.7064e - 1(1.20e - 1) -	2.7099e - 1 (6.09e - 2) -	2.7469e - 1 (2.74e - 2) -	1.2060e + 0(3.03e - 1) -	1.0998e + 0(1.01e - 1) -	2.2998e - 1 (3.64e - 2)
	Ŋ	1.0715e + 0(1.25e - 1) -	7.1595e - 1 (7.65e - 2) -	6.4578e - 1 (5.68e - 2) -	1.3446e + 0(1.31e - 1) -	1.8659e + 0(6.16e - 2) -	5.6904e - 1 (3.55e - 2)
Mar 10	8	2.0195e + 0(1.77e - 1) -	1.3835e + 0(9.79e - 2) -	1.2115e + 0(7.58e - 2) -	2.0108e + 0(1.45e - 1) -	2.5132e + 0(3.07e - 2) -	1.1162e + 0(6.75e - 2)
	10	2.6047e + 0(1.56e - 1) -	2.0645e + 0(1.04e - 1) -	1.4361e + 0(5.70e - 2) +	2.1211e + 0(1.53e - 1) -	2.8953e + 0(7.25e - 2) -	1.7262e + 0(1.78e - 1)
	3	5.1383e - 1 (3.53e - 2) -	1.8392e - 1(1.13e - 2) =	$2.0367e - 1 \left(6.21e - 3 ight) -$	1.7442e - 1(1.24e - 2) +	1.7119e - 1(4.61e - 3) +	1.8646e - 1 (5.78e - 3)
	Ŋ	7.2572e - 1 (3.91e - 2) -	5.4501e - 1(2.09e - 2) =	6.4109e - 1 (1.76e - 2) -	$4.9795e - 1 \left(1.54e - 2 \right) +$	5.0175e - 1(1.12e - 2) +	5.3607e - 1 (1.61e - 2)
Marii	8	1.4217e + 0(7.48e - 2) -	1.0668e + 0(3.07e - 2) =	1.1977e + 0(5.29e - 2) -	1.0789e + 0(2.60e - 2) -	1.0284e + 0(2.32e - 2) +	$1.0533e + 0 \ (2.38e - 2)$
	10	1.5639e + 0(6.25e - 2) -	1.1611e + 0(3.94e - 2) +	1.4585e + 0(6.27e - 2) -	1.2809e + 0(9.81e - 2) -	1.2708e + 0(4.50e - 2) -	1.1982e + 0(4.57e - 2)
	б	7.6304e - 1 (2.11e - 1) -	2.2824e - 1 (2.70e - 2) -	2.1747e - 1(5.45e - 3) -	2.2244e - 1 (2.86e - 2) -	2.3242e - 1(8.27e - 3) -	2.1037e - 1 (3.40e - 3)
	Ŋ	1.7006e + 0(5.81e - 2) -	1.1269e + 0(1.25e - 2) -	1.1898e + 0(2.84e - 2) -	1.1637e + 0(3.76e - 2) -	1.2152e + 0(2.52e - 2) -	$1.0959e + 0 \ (2.34e - 2)$
Mar 12	8	$4.1857e + 0 \left(1.00e - 1\right) -$	3.1078e + 0(2.66e - 2) -	3.3918e + 0(1.42e - 1) -	3.9919e + 0(2.55e - 1) -	3.3667e + 0(4.68e - 2) -	3.0724e + 0(4.47e - 2)
	10	5.3628e + 0(8.24e - 2) -	3.9919e + 0(5.71e - 2) +	5.2472e + 0(2.33e - 1) -	5.8833e + 0(5.36e - 1) -	4.8840e + 0(5.86e - 2) -	4.2155e + 0(1.13e - 1)
	33	3.6621e - 1(8.46e - 2) -	1.3909e - 1(2.37e - 2) -	1.1348e - 1(1.07e - 2) -	9.1138e - 2(5.36e - 3) -	9.9556e - 2(1.23e - 2) -	7.8572e - 2(5.73e - 3)
Maria	Ŋ	1.7164e - 1 (2.02e - 2) -	1.9313e - 1(2.52e - 2) -	1.8053e - 1 (2.33e - 2) -	1.4469e - 1 (2.36e - 2) =	1.9105e - 1 (2.99e - 2) -	$1.4216e - 1 \; (1.45e - 2)$
VIAF 15	8	1.7984e - 1 (1.83e - 2) -	$1.9834e - 1 \left(1.96e - 2 ight) -$	2.0587e - 1(2.71e - 2) -	$1.6212e - 1 \left(1.54e - 2 ight) -$	2.323e - 1(2.57e - 2) -	1.4372e - 1 (8.87e - 3)
	10	1.4759e - 1(1.75e - 2) -	1.6676e - 1(1.74e - 2) -	2.3032e - 1(1.72e - 2) -	1.7107e - 1 (1.69e - 2) -	2.1582e - 1(2.14e - 2) -	1.2606e - 1 (1.06e - 2)
=/-/+		0/40/4	5/32/7	4/31/9	6/32/6	6/34/4	

TABLE 6: Continued.



FIGURE 7: Pareto front on WFG1 with 10 objectives: (a) k-NSGAII. (b) KnEA. (c) Pi-MOEA. (d) hpaEA. (e) VaEA. (f) KPEA.



FIGURE 8: Pareto front on WFG9 with 10 objectives: (a) k-NSGAII. (b) KnEA. (c) Pi-MOEA. (d) hpaEA. (e) VaEA. (f) KPEA.



FIGURE 9: Continued.



FIGURE 9: Pareto front on MaF1 with 3 objectives: (a) k-NSGAII. (b) KnEA. (c) Pi-MOEA. (d) hpaEA. (e) VaEA. (f) KPEA.



FIGURE 10: Continued.



FIGURE 10: Pareto front on MaF5 with 3 objectives: (a) k-NSGAII. (b) KnEA. (c) Pi-MOEA. (d) hpaEA. (e) VaEA. (f) KPEA.



FIGURE 11: Pareto front on MaF5 with 10 objectives: (a) k-NSGAII. (b) KnEA. (c) Pi-MOEA. (d) hpaEA. (e) VaEA. (f) KPEA.

Journal of Applied Mathematics



FIGURE 12: Pareto front on MaF8 with 3 objectives: (a) k-NSGAII. (b) KnEA. (c) Pi-MOEA. (d) hpaEA. (e) VaEA. (f) KPEA.

TABLE 7: The properties of test problems.

Properties	Problems
Linear Pareto fronts	MaF1
Concave Pareto fronts	MaF2
Degenerate Pareto fronts	MaF6
Distance minimization problems	MaF9
Convex Pareto fronts	WFG1
Disconnect Pareto fronts	WFG2
Scaled Pareto fronts	WFG9

the algorithms on each test instance, which helps the overall performance of each algorithm.

4.4. Parameter Settings. The several key parameters adopted in this experiment are summarized as follows:

- (i) *Variation Operators.* In this paper, the simulated binary crossover (SBX) [64] and polynomial mutation [65] are adopted by all the MaOEAs used for comparison to generate offspring solutions. The crossover probability p_c is set to 1, and the mutation probability p_m is set to 1/*D*, where *D* is the number of decision variables. Following the recommendations in literature [23], the distribution indexes η_c and η_m are set to 20 and 30, respectively. The other parameter settings for the algorithms are consistent with the settings described in their respective literature
- (ii) *Population Size.* This paper considers 3, 5, 8, and 10 objectives. The population size N of six algorithms for different numbers of objectives M is summarized in Table 2
- (iii) The Number of Runs and Termination Criterion. For each test instance, each algorithm runs 20 times independently. The termination criterion for each run is determined by the predefined maximum number of generations (maxGen). For WFG1 and WFG2, maxGen is set to 1000. For WFG3-WFG9, maxGen is set to 250. For the MaF test suite, max-Gen is set to 250. When the number of generations reaches maxGen, all algorithms are terminated

5. Experimental Results

In this section, according to the experimental design described in Section 4, KPEA is compared with five stateof-the-art algorithms on WFG and MaF. The statistical results of IGD and HV metrics obtained by the six algorithms on the test suite are listed in Tables 3–6. The best results for each problem instance are annotated with a dark gray background. Then, in Section 5.3, the sensitivity of KPEA to parameter r is discussed. Additionally, the performance comparison of KPEA and its different variants is analyzed in Section 5.4. Finally, in Section 5.5, the overall performance of six algorithms is analyzed. 5.1. Analysis of WFG Test Problems. The mean and standard deviation values of IGD and HV results obtained by five compared algorithms and KPEA on the WFG test suite are listed in Tables 3 and 4. The performance is challenged by the multiple complex features of WFG problems, which include scalability, deception, and separability.

As shown in Table 3, KPEA significantly outperforms other algorithms. Specifically, among 36 instances, KPEA outperforms all compared algorithms on 18 instances. The performance of KPEA is significantly better than the other five algorithms on the WFG1, WFG2, and WFG8 problems with scaling and concave-convex Pareto fronts. It can be attributed to the novel adaptive switching mechanism of KPEA, which balances diversity and convergence. In terms of the HV indicator, KnEA performs better on the WFG5 and WFG9 problems with deceptive and concave Pareto fronts. This is due to the fact that KnEA sets a small threshold T for the adaptive strategy to identify knee points, which helps KnEA to achieve good performance on scalable problems. Pi-MOEA performs better than KPEA when solving the WFG problems with three-objective. Furthermore, the performance of VaEA is considerably worse than that of KPEA. The reason is that VaEA emphasizes diversity too much but the importance of convergence is neglected.

In Table 4, in terms of IGD values, KPEA and VaEA achieve better results than other algorithms on most of the WFG test problems. Out of the 36 test instances, KPEA achieves the best IGD result on 20 instances, while Pi-MOEA obtains the best result on 8 instances. The reason why KPEA performs well on some WFG test problems may be that it combines the knee points with the maximum-vector-angle-first principle to guide the evolution of the population. Therefore, unlike KnEA and k-NSGAII that lack a diversity maintenance mechanism, it is not easy for KPEA to trap in the local optima on problems with biased Pareto fronts such as WFG1, WFG8, and WFG9. Pi-MOEA uses a threshold T defined based on the average ranking to control the neighborhood range of solutions, which helps the algorithm maintain a balance between convergence and diversity. It is important to note that on the WFG test suite, the performance of the k-NSGAII and hpaEA is relatively the worst in both HV and IGD metrics. For k-NSGAII, the effectiveness of the knee point-driven algorithms is heavily influenced by the approach used to detect the knee points, which is a crucial factor. However, the approach proposed by k-NSGAII only focuses on a small portion of the PF region when identifying the knee points, which results in poor diversity during the search process of k-NSGAII. For hpaEA, this is mainly due to its proposed prominent solution protection strategy, which is unable to maintain a good diversity of solutions on the linear Pareto front of the WFG test suite.

Taking WFG1 and WFG9 with 10 objectives as examples, the final solutions obtained by six algorithms in a single run are shown in Figures 7 and 8, respectively. For the WFG1 problem with 10 objectives, each algorithm encounters some difficulties in solving this problem, as shown in Figure 7. In terms of distribution and convergence, both hpaEA and VaEA perform poorly. Furthermore, the uniformity of the Pareto



FIGURE 13: Continued.



FIGURE 13: The average IGD values of KPEA on different test problems with 3, 5, and 10 objectives when r varies from 0 to 5. (a) 3 objectives. (b) 5 objectives. (c) 10 objectives.

front obtained by k-NSGAII and KnEA is worse than that of KPEA. Only Pi-MOEA and KPEA show relatively good performance, because their obtained solutions are closer to the true Pareto front. From Figure 8, the solutions obtained by KPEA and KnEA are relatively better in terms of both convergence and distribution. While the solutions obtained by k-NSGAII, Pi-MOEA, and hpaEA have all been trapped in local optima on the Pareto front. The solutions obtained by VaEA are the worst in terms of both convergence and diversity.

5.2. Analysis of MaF Test Problems. Although WFG problems are widely used to evaluate the performance of many-objective optimization algorithms, their benchmark functions are too regular and monotonous. Furthermore, WFG problems lack some validation for the complex and uncertain real-world problems. In contrast, the MaF problems that focus on evaluating diversity and convergence have higher complexity and practical significance. In this study, tests are conducted on the MaF problems to further validate the performance of KPEA.

The statistical results of the HV values obtained by six algorithms on the MaF test suite are presented in Table 5. KPEA outperforms the other five algorithms on 27 instances out of the 44 test instances. Furthermore, KPEA also demonstrates strong competitiveness on MaF4 with a convex Pareto front, MaF8, MaF11 with a disconnected Pareto front, and MaF13 with irregular mixed Pareto fronts. The IGD values obtained by the considered algorithms on the MaF test suite are presented in Table 6. Out of the 44 test instances, KPEA achieves the best IGD value on 26 instances. Additionally, Pi-MOEA also performs well, which obtains the best IGD value on 6 test instances. Conversely, the performance of k-NSGAII and KnEA is not as good as other algorithms.

To illustrate the convergence and distribution of solutions more intuitively, the final population of six algorithms on several representative problems is plotted in Figures 9-12. KPEA, Pi-MOEA, hpaEA, and VaEA maintain a uniform population distribution on MaF1 with an irregular PF shape, as shown in Figure 9. On the contrary, the distribution of solutions obtained by k-NSGAII and KnEA is relatively poor. The distribution of solutions obtained by different algorithms on the MaF5 with three objectives is shown in Figure 10. The solutions with the uniform distribution obtained by Pi-MOEA and KPEA are closest to the true Pareto front of MaF5. Compared to k-NSGAII and hpaEA, the solutions of KnEA and VaEA seem to have good convergence. However, both KnEA and VaEA struggle to maintain population diversity on MaF5. From Figure 11, the solutions obtained by KnEA, VAEA, and KPEA are closest to the true Pareto front when the number of objectives is 10. Moreover, as shown in Figures 10 and 11. unlike other algorithms, the performance of KPEA becomes more prominent as the number of objectives increases. MaF8 is a problem of multipoint distance

TABLE 8: Mean and standard deviation values of HV metric for KPEA, KPEA-N, KPEA-NR, and KPEA-NK on MaF proble	lems.
---	-------

Problem	М	KPEA-N	KPEA-NK	KPEA-NR	KPEA
	3	2.2089e - 1(4.29e - 4) +	2.2086e - 1(4.75e - 4) +	2.2107e - 1(3.37e - 4) +	2.1988e - 1(6.02e - 4)
	5	1.0157e - 2(1.37e - 4) -	1.0308e - 2(1.50e - 4) -	1.0240e - 2(1.79e - 4) -	1.0997e - 2(2.40e - 4)
MaFI	8	2.1377 <i>e</i> – 5 (2.59 <i>e</i> – 6) –	2.2644 <i>e</i> – 5 (2.76 <i>e</i> – 6) –	2.1132e - 5(2.07e - 6) -	3.1775 <i>e</i> – 5 (2.88 <i>e</i> – 6)
	10	2.6851e - 7(1.41e - 7) -	2.2828e - 7(1.73e - 7) -	2.1136 <i>e</i> – 7 (2.07 <i>e</i> – 7) –	5.5111e - 7(2.81e - 7)
	3	1.0347 <i>e</i> – 1 (2.10 <i>e</i> – 1) –	2.4796 <i>e</i> – 2 (1.06 <i>e</i> – 1) –	1.9116e – 2 (8.28e – 2) –	4.5241e - 1(1.50e - 1)
	5	9.9397 <i>e</i> - 3 (3.06 <i>e</i> - 2) -	1.1778 <i>e</i> – 3 (4.71 <i>e</i> – 3) –	0.0000e + 0(0.00e + 0) -	4.6754e - 2(3.60e - 2)
MaF4	8	2.8676e - 4(6.78e - 4) -	7.3239 <i>e</i> – 7 (3.28 <i>e</i> – 6) –	1.3434e - 5(6.01e - 5) -	8.3691e - 4(8.10e - 4)
	10	0.0000e + 0(0.00e + 0) -	4.8747e - 6(1.68e - 5) =	0.0000e + 0(0.00e + 0) -	8.1403e - 6(2.07e - 5)
	3	4.8596e - 1(1.27e - 1) =	5.5865e - 1(8.71e - 4) +	5.4846e - 1(4.75e - 2) +	5.3746e - 1(5.85e - 2)
	7.7647e - 1(2.07e - 3) =	7.7658e - 1 (1.67e - 3) =	7.7107e - 1(2.58e - 2)		
	9.1862e - 1(7.50e - 3) +	9.1141e - 1(2.02e - 3) =	9.0744e - 1(3.07e - 3) -	9.1176 <i>e</i> – 1 (8.41 <i>e</i> – 3)	
	10	9.1566 <i>e</i> – 1 (6.96 <i>e</i> – 3) –	9.4286e - 1(3.78e - 3) -	9.3816e - 1(5.22e - 3) -	9.5801e - 1(6.93e - 3)
	3	1.9824e - 1(6.07e - 4) -	1.9989e - 1(1.16e - 4) +	1.9994e - 1(6.50e - 5) +	1.9890e - 1 (2.06e - 4)
MaF6 MaF7	5	1.2791e - 1(8.01e - 4) -	1.2929e - 1(3.54e - 4) +	1.2927e - 1(4.04e - 4) +	1.2882e - 1 (2.55e - 4)
	8	3.7002e - 2(5.17e - 2) -	6.8908e - 2(5.19e - 2) =	3.7135e - 2(5.19e - 2) -	9.5296e - 2(3.26e - 2)
	10	0.0000e + 0(0.00e + 0) -	5.0156e - 3(2.24e - 2) =	0.0000e + 0(0.00e + 0) -	2.0106e - 2(4.13e - 2)
	3	2.6851e - 1(3.50e - 3) +	2.7859e - 1 (5.89e - 4) +	2.7522e - 1(1.16e - 2) +	2.6615e - 1(2.06e - 2)
M 177	5	2.1904e - 1(5.72e - 3) -	2.5066e - 1(4.89e - 3) =	2.5207e - 1(1.82e - 3) +	2.4103e - 1(1.75e - 2)
MaF/	8	1.0654e - 1(1.56e - 2) +	1.6928e - 1(8.21e - 3) +	1.6443e - 1(1.35e - 2) +	7.7005e - 2(1.87e - 2)
	10	3.0526e - 2(1.06e - 2) -	1.1904e - 1(1.07e - 2) +	7.5015e - 2(2.21e - 2) +	5.7313e - 2(9.76e - 3)
	3	1.5776e - 1(6.12e - 2) -	2.5758e - 1(1.34e - 2) +	2.6176 <i>e</i> – 1 (1.69 <i>e</i> – 2) +	2.1181e - 1(5.46e - 2)
M-T0	5	4.9496e - 2(3.86e - 2) =	1.1134e - 1(8.48e - 3) +	1.1529e - 1(4.84e - 3) +	7.0910e - 2(3.54e - 2)
MaF8	8	1.0392e - 2(8.88e - 3) =	2.7399e - 2(1.91e - 3) +	2.7955e - 2(2.03e - 3) +	1.3337e - 2(1.05e - 2)
	10	3.0468 <i>e</i> – 3 (2.96 <i>e</i> – 3)	$\begin{array}{llllllllllllllllllllllllllllllllllll$		
	3	6.4323e - 1(8.40e - 2) -	6.3004e - 1(3.74e - 2) -	6.3182e - 1(3.76e - 2) -	8.0297e - 1(1.26e - 2)
MaEO	5	1.5440e - 1(9.01e - 2) -	1.7294e - 1(6.32e - 2) -	1.6515e - 1(5.91e - 2) -	2.6738e - 1(1.58e - 2)
Mary	8	8.4353e - 4(2.58e - 3) -	1.1598e - 2(1.26e - 2) -	9.8085e - 3(9.34e - 3) -	3.3563e - 2(3.16e - 3)
	10	4.9652 <i>e</i> - 3 (3.20 <i>e</i> - 3) -	1.0867e - 2(3.02e - 3) +	8.2588e - 3(2.57e - 3) =	8.6205e - 3(1.19e - 3)
	3	3.8120e - 1(2.86e - 2) -	6.0320e - 1(2.44e - 2) -	8.3539e - 1 (2.46e - 2) -	9.1376e - 1(1.92e - 2)
MaE10	5	4.4763e - 1(4.16e - 2) -	6.9871e - 1(3.79e - 2) -	7.3122e - 1(4.63e - 2) -	9.3890e - 1(3.20e - 2)
Iviar 10	8	3.5522e - 1(4.08e - 2) -	6.0066e - 1(5.95e - 2) -	6.2230e - 1(3.63e - 2) -	8.2514e - 1(6.79e - 2)
	10	3.0918e - 1(4.95e - 2) -	5.6976e - 1(4.75e - 2) =	5.8848e - 1(5.32e - 2) +	5.6479e - 1(9.82e - 2)
	3	9.0938e - 1 (4.80e - 3) -	9.0585e - 1(9.03e - 3) -	9.2773e - 1(2.41e - 3) -	9.3479e - 1(1.03e - 3)
MaE11	5	9.4812e - 1(8.35e - 3) -	9.3286e - 1(1.74e - 2) -	9.8274e - 1(3.67e - 3) -	9.9377e - 1(9.73e - 4)
IVIAF11	8	9.5226e - 1(1.08e - 2) -	9.5909e - 1(8.12e - 3) -	9.8842e - 1(2.77e - 3) -	9.9471e - 1(1.79e - 3)
	10	9.4896e - 1(9.53e - 3) -	9.6433e - 1(9.35e - 3) -	9.8716 <i>e</i> – 1 (5.59 <i>e</i> – 3) –	9.9092e - 1(2.46e - 3)
	3	5.2372e - 1(3.56e - 3) -	5.2860e - 1(4.14e - 3) -	5.3250e - 1(3.80e - 3) -	5.3619e - 1(4.26e - 3)
MaE12	5	6.8959e - 1(1.85e - 2) -	7.1681e - 1(8.43e - 3) +	7.2182e - 1(6.58e - 3) +	7.0763e - 1(1.08e - 2)
wiar12	8	7.3725e - 1(4.89e - 2) -	8.1452e - 1(3.44e - 2) +	8.2499e - 1(1.01e - 2) +	7.9307e - 1(1.46e - 2)
	10	7.4826e - 1(3.94e - 2) -	8.3485e - 1(1.23e - 2) =	8.2515e - 1 (3.65e - 2) =	8.2902e - 1(1.37e - 2)

Problem	М	KPEA-N	KPEA-NK	KPEA-NR	KPEA
MaF13	3	4.9582e - 1(1.67e - 2) =	5.1793e - 1(8.66e - 3) +	4.9792e - 1(1.68e - 2) =	4.9983 <i>e</i> - 1 (1.41 <i>e</i> - 2)
	5	2.3591e - 1(1.76e - 2) -	2.5166e - 1(8.49e - 3) =	2.1398e - 1(2.87e - 2) -	2.5636e - 1(8.44e - 3)
	8	1.3714e - 1(8.44e - 3) -	1.4881e - 1(5.82e - 3) -	1.2333e - 1(1.85e - 2) -	1.6289e - 1(4.20e - 3)
	10	1.0115e - 1(1.65e - 2) -	1.1908e - 1(6.03e - 3) -	9.9478e - 2(1.08e - 2) -	1.3634e - 1(1.59e - 3)
+/-/=		4/35/5	15/20/9	15/25/4	

TABLE 8: Continued.

minimization [43]. From Figure 12, KPEA demonstrates the best convergence and diversity performance on MaF8. However, the distribution of k-NSGAII, Pi-MOEA, and KnEA is quite unsatisfactory. The solutions obtained by hpaEA and VaEA cannot converge to the Pareto front on MaF8. This also proves the effectiveness of the proposed KPEA.

5.3. Sensitivity of Parameter r. To investigate the sensitivity of parameter r of preprocessing in KPEA, the performance with varying r from 0 to 5 on 3, 5, and 10 objective test problems is evaluated. Seven representative problems in Table 7 are selected for testing and comparison. The average IGD values in 30 iterations for each problem are calculated.

From Figure 13, on MaF2 and WFG9 problems with concave and scaled Pareto fronts, the performance of KPEA is insensitive to r. On MaF1 with degenerate Pareto front, the analysis needs to be conducted based on the number of objectives. As shown in Figures 13(a) and 13(b), when the number of objectives is less than 10, the IGD value of KPEA slightly fluctuates as the value of *r* increases. However, when the number of objectives reaches 10, the variation of r has a significant impact on the performance of KPEA. Firstly, as r increases from 0 to 1.5, the IGD value consistently decreases. Then, as r continues to increase, the IGD value gradually increases. Moreover, on WFG1 and MaF9, the IGD value also slightly fluctuates as the r value increases. However, no matter how the r value fluctuates, the IGD values of KPEA are always the smallest at r = 1.5. On WFG2 and MaF6, the IGD values rapidly decrease until r = 0.5 and then tends to stabilize with the increase of r.

Therefore, r = 1.5 is the best choice to improve the performance of KPEA. The parameter setting of r = 1.5 can significantly enhance the applicability of the algorithm.

5.4. Performance Comparison of KPEA and Its Different Variants. The performance of KPEA is compared with its three variants in this section. The experimental results are used to validate the effectiveness of penalty, the effectiveness of preprocessing, and the necessity of knee point guidance. KPEA-N represents a variant of KPEA. During the environment selection of KPEA-N, solutions are selected based on the worse-elimination principle in VaEA, rather than on the penalty mechanism. This variant contains only preprocessing strategy and knee point guidance. KPEA-NR denotes a variant of KPEA that does not eliminate DRSs. KPEA-NK represents a variant of KPEA without a knee point-driven strategy. The IGD and HV results of KPEA, KPEA-N,

KPEA-NR, and KPEA-NK on the MaF test suite are given in Tables 8 and 9, respectively.

Specifically, in terms of HV values, KPEA outperforms the other three variants on all test instances of the MaF4 and MaF11 problems. In terms of IGD values, KPEA outperforms KPEA-N, KPEA-NR, and KPEA-NK on 29 instances out of the 44 test instances. As shown in Table 8, for the MaF8 problem with distance minimization, there is no need to use preprocessing to eliminate DRSs. For problems with nonseparable Pareto fronts, the solutions obtained by KPEA-NR are very close to the final Pareto optimal front. In other words, the preprocessing strategy of KPEA may not be beneficial for optimization on such problems.

As shown in Table 9, for the MaF7 problem with concave-convex Pareto fronts, KPEA-NK performs significantly better than KPEA. This may be due to the fact that MaF7 is a multimodal problem that contains a large number of segmented Pareto-optimal fronts, which results in the KPEA easily falling into local Pareto optimality when driven by knee points.

It is important to note that on the MaF test suite, the performance of the KPEA-N is relatively the worst in both HV and IGD metrics. It is difficult for this variant to achieve relatively better performance on any problem. The adaptive penalty mechanism selects the individuals with the best convergence one by one, which achieves comprehensive coverage of the Pareto front. Compared to the worseelimination principle in KPEA-N, the penalty mechanism in KPEA can better balance the convergence and diversity of the population.

However, both for IGD and HV values, KPEA has a significant advantage over the variant algorithms KPEA-N, KPEA-NR, and KPEA-NK in most test instances. Overall, the combination of the penalty mechanism, knee-pointdriven strategy, and elimination of DRS strategy is more suitable for most problems.

5.5. Overall Performance Analysis. In this section, two statistical tests (i.e., the Wilcoxon rank-sum test and the Friedman test) are used to validate the comprehensive performance of KPEA in terms of convergence and diversity.

The statistical results based on the Wilcoxon rank-sum test are listed in Table 10. Specifically, in terms of the IGD metric, the proportion of the test instances where KPEA outperforms k-NSGAII, KnEA, Pi-MOEA, hpaEA, and VaEA with statistical significance is 50/80, 62/80, 69/80, 77/80, and 60/80, respectively. In terms of the HV metric, the proportion of the test instances where KPEA outperforms

TABLE 9: Mean and standard deviation values of IGD metric for KPEA, KPEA-N, K	KPEA-NR, and KPEA-NK on MaF	problems.
---	-----------------------------	-----------

Problem	М	KPEA-N	KPEA-NK	KPEA-NR	KPEA
	3	4.0391e - 2(4.61e - 4) +	4.0302e - 2(5.82e - 4) +	4.0226e - 2(3.57e - 4) +	2.1988e - 1(6.02e - 4)
MaF1	5	1.2929e - 1(1.15e - 3) -	1.2824e - 1(1.34e - 3) -	1.2931e - 1 (2.00e - 3) -	1.0997e - 2(2.40e - 4)
	8	2.1941e - 1(2.94e - 3) -	2.2086e - 1(2.46e - 3) -	2.2223e - 1(2.29e - 3) -	3.1775e - 5(2.88e - 6)
	10	2.6049e - 1(3.93e - 3) -	2.6551e - 1(4.91e - 3) -	2.6649e - 1(5.24e - 3) -	5.5111e - 7(2.81e - 7)
	3	1.7742e + 1(1.21e + 1) -	2.8287e + 1(2.02e + 1) -	1.9268e + 1(8.91e + 0) -	4.5241e - 1(1.50e - 1)
	5	1.1676e + 2(9.09e + 1) -	1.5897e + 2(1.04e + 2) -	1.4215e + 2(7.28e + 1) -	4.6754e - 2(3.60e - 2)
MaF4	8	1.2242e + 2(1.27e + 2) =	1.0260e + 3(5.72e + 2) -	1.3778e + 3(8.81e + 2) -	8.3691e - 4(8.10e - 4)
MaF4 MaF5 MaF6 MaF7 MaF8	10	3.5924 <i>e</i> + 3 (2.44 <i>e</i> + 3) -	3.8548e + 3(2.18e + 3) -	5.1648e + 3(3.20e + 3) -	8.1403e - 6(2.07e - 5)
	3	$8.4227e - 1 \left(1.17e + 0 \right) =$	2.5088e - 1(3.63e - 3) +	3.4198e - 1(3.95e - 1) +	5.3746e - 1 (5.85e - 2)
Mare	5	2.4032e + 0(5.83e - 2) -	2.4570e + 0(6.17e - 2) -	2.4771e + 0(5.83e - 2) -	7.7107e - 1(2.58e - 2)
Mars	8	1.5798e + 1(1.27e + 0) =	2.0142e + 1(1.96e + 0) -	1.9247e + 1(1.09e + 0) -	9.1176e - 1(8.41e - 3)
	10	7.1782e + 1(4.37e + 0) -	7.6200e + 1(6.73e + 0) -	7.4328e + 1(6.66e + 0) -	9.5801e - 1(6.93e - 3)
	3	5.3533e - 3(5.41e - 4) +	4.4887e - 3(8.53e - 5) +	$4.5258e - 3\left(1.15e - 4\right) + \\$	$1.9890e - 1 \ (2.06e - 4)$
MaEC	5	5.7915e - 3(5.86e - 4) -	5.0733e - 3(2.54e - 4) -	5.0449e - 3(1.83e - 4) -	1.2882e - 1 (2.55e - 4)
маго	8	2.1817e + 0(2.24e + 0) -	1.3031e + 0(1.97e + 0) -	1.5309e + 0(1.79e + 0) -	9.5296e - 2(3.26e - 2)
	10	7.0339e + 0(3.08e + 0) -	2.6262e + 0(1.59e + 0) -	4.7142e + 0(5.63e + 0) -	2.0106e - 2(4.13e - 2)
	3	6.4477e - 2(3.88e - 3) +	5.6687e - 2(9.13e - 4) +	8.7337e - 2(9.25e - 2) =	2.6615e - 1(2.06e - 2)
	5	3.7782e - 1(2.74e - 2) =	3.5378e - 1(9.46e - 2) =	3.1597e - 1(7.25e - 3) +	2.4103e - 1(1.75e - 2)
MaF/	8	1.0007e + 0(2.23e - 1) -	8.6721e - 1(7.70e - 2) -	1.3021e + 0(7.10e - 1) -	7.7005e - 2(1.87e - 2)
	10	1.7514e + 0(2.95e - 1) -	1.4734e + 0(1.86e - 1) -	1.4407e + 0(3.23e - 1) -	5.7313e - 2(9.76e - 3)
	3	3.9414 <i>e</i> – 1 (2.20 <i>e</i> – 1) –	1.0829e - 1(3.35e - 2) +	9.9263e - 2(4.38e - 2) +	2.1181e - 1(5.46e - 2)
MaE9	5	6.8312e - 1(3.75e - 1) -	1.5634e - 1(4.22e - 2) +	1.3217e - 1(1.76e - 2) +	7.0910e - 2(3.54e - 2)
MaF8	8	8.5861e - 1(5.66e - 1) =	2.0235e - 1(5.04e - 2) +	1.7471e - 1(3.02e - 2) +	1.3337e - 2(1.05e - 2)
	10	9.5389e - 1(5.65e - 1) =	2.6425e - 1(8.33e - 2) +	$\begin{aligned} 6551e - 1 & (4.91e - 3) - 2.6649e - 1 & (5.24e - 3) - 5.5111e - 7 & (2.81e) \\ 8287e + 1 & (2.02e + 1) - 1.9268e + 1 & (8.91e + 0) - 4.5241e - 1 & (1.50e) \\ 5897e + 2 & (1.04e + 2) - 1.4215e + 2 & (7.28e + 1) - 4.6754e - 2 & (3.60e) \\ 0260e + 3 & (5.72e + 2) - 1.3778e + 3 & (8.81e + 2) - 8.3691e - 4 & (8.10e) \\ 8548e + 3 & (2.18e + 3) - 5.1648e + 3 & (3.20e + 3) - 8.1403e - 6 & (2.07e) \\ 5088e - 1 & (3.63e - 3) + 3.4198e - 1 & (3.95e - 1) + 5.3746e - 1 & (5.85e) \\ 4570e + 0 & (6.17e - 2) - 2.4771e + 0 & (5.83e - 2) - 7.7107e - 1 & (2.58e) \\ 0142e + 1 & (1.96e + 0) - 1.9247e + 1 & (1.09e + 0) - 9.1176e - 1 & (8.41e) \\ 6200e + 1 & (6.73e + 0) - 7.4328e + 1 & (6.66e + 0) - 9.5801e - 1 & (6.93e) \\ 4887e - 3 & (8.53e - 5) + 4.5258e - 3 & (1.15e - 4) + 1.9890e - 1 & (2.06e) \\ 0733e - 3 & (2.54e - 4) - 5.0449e - 3 & (1.83e - 4) - 1.2882e - 1 & (2.55e) \\ 3031e + 0 & (1.97e + 0) - 1.5309e + 0 & (1.79e + 0) - 9.5296e - 2 & (3.26e) \\ 6262e + 0 & (1.59e + 0) - 4.7142e + 0 & (5.63e + 0) - 2.0106e - 2 & (4.13e) \\ 6687e - 2 & (9.13e - 4) + 8.7337e - 2 & (9.25e - 2) = 2.6615e - 1 & (2.06e) \\ 5378e - 1 & (9.46e - 2) = 3.1597e - 1 & (7.25e - 3) + 2.4103e - 1 & (1.75e) \\ 6721e - 1 & (7.70e - 2) - 1.3021e + 0 & (7.10e - 1) - 7.7005e - 2 & (1.87e) \\ 4734e + 0 & (1.86e - 1) - 1.4407e + 0 & (3.23e - 1) - 5.7313e - 2 & (9.76e) \\ 0229e - 1 & (3.35e - 2) + 9.9263e - 2 & (4.38e - 2) + 2.1181e - 1 & (5.46e) \\ 5634e - 1 & (4.22e - 2) + 1.3217e - 1 & (1.76e - 2) + 7.0910e - 2 & (3.54e) \\ 0235e - 1 & (5.33e - 2) + 2.0850e - 1 & (3.35e - 2) + 4.4683e - 3 & (4.01e) \\ 7865e - 1 & (7.79e - 2) - 3.8255e - 1 & (7.26e - 2) - 8.0297e - 1 & (1.26e) \\ 5926e - 1 & (6.38e - 2) - 7.4287e - 1 & (8.78e - 2) - 9.3890e - 1 & (3.20e) \\ 5926e - 1 & (1.43e - 1) + 5.0789e - 1 & (1.47e - 1) = 8.6205e - 3 & (1.192e) \\ 9941e - 1 & (7.5e - 2) - 1.6005e + 0 & (1.10e + 0) - 3.3563e - 2 & (3.16e) \\ 5928e + 0 & (9.96e - 2) - 1.4060e + 0 & (7.32e - 2) - 8.2514e - 1 & (6.79e) \\ 5928e + 0 & (9.96e - 2) - 1.4060e + 0 & (7.32e - 2) - 8.2514e - 1 & (6.79e) \\ 5928e + 0 & (9$	4.4683e - 3(4.01e - 3)
	3	3.1576e - 1(1.23e - 1) -	3.7865e - 1(7.79e - 2) -	3.8255e - 1(7.26e - 2) -	8.0297e - 1(1.26e - 2)
MaEO	5	6.8967e - 1(5.38e - 1) -	5.6269e - 1(2.65e - 1) -	5.7609e - 1(2.64e - 1) -	2.6738e - 1(1.58e - 2)
MaF6 MaF7 MaF8 MaF9 MaF10 MaF11	8	4.4207e + 0(1.79e + 0) -	1.3037e + 0(7.46e - 1) -	1.6005e + 0(1.10e + 0) -	3.3563e - 2(3.16e - 3)
	10	1.6035e + 0(1.83e + 0) -	3.6377e - 1(1.43e - 1) +	5.0789e - 1(1.47e - 1) =	8.6205e - 3(1.19e - 3)
	3	1.1055e + 0(4.94e - 2) -	7.9941e - 1(7.35e - 2) -	2.8117e - 1(3.73e - 2) -	9.1376 <i>e</i> – 1 (1.92 <i>e</i> – 2)
MaE10	5	1.4883e + 0(1.05e - 1) -	9.7904e - 1(6.38e - 2) -	7.4287e - 1(8.78e - 2) -	9.3890e - 1(3.20e - 2)
Iviar 10	8	2.2126e + 0(1.39e - 1) -	1.5982e + 0(9.96e - 2) -	1.4060e + 0(7.32e - 2) -	8.2514e - 1(6.79e - 2)
	10	2.6663e + 0(2.06e - 1) -	1.8065e + 0(9.33e - 2) =	1.7623e + 0(1.56e - 1) =	5.6479e - 1(9.82e - 2)
	3	1.6686e - 1 (9.09e - 3) +	1.8924e - 1(1.75e - 2) =	1.5985e - 1(3.65e - 3) +	9.3479e - 1(1.03e - 3)
MaE11	5	6.1606e - 1(7.52e - 2) -	9.8898e - 1(1.11e - 1) -	4.8079e - 1 (1.10e - 2) +	9.9377e - 1(9.73e - 4)
Marii	8	1.4710e + 0(8.43e - 2) -	1.8382e + 0(9.02e - 2) -	1.1113e + 0(1.66e - 2) -	9.9471e - 1(1.79e - 3)
	10	1.4892e + 0(8.56e - 2) -	1.8557e + 0(5.22e - 2) -	1.2690e + 0(3.32e - 2) -	9.9092e - 1(2.46e - 3)
	3	2.1761e - 1(3.53e - 3) -	2.1599e - 1(2.75e - 3) -	2.1602e - 1(2.80e - 3) -	5.3619 <i>e</i> – 1 (4.26 <i>e</i> – 3)
MaE12	5	1.1590e + 0(1.08e - 2) -	1.1688e + 0(1.28e - 2) -	1.1828e + 0(1.60e - 2) -	7.0763e - 1(1.08e - 2)
war12	8	3.2493e + 0(3.19e - 2) -	3.3432e + 0(3.36e - 2) -	3.3202e + 0(4.17e - 2) -	7.9307e - 1(1.46e - 2)
	10	4.8536e + 0(5.34e - 2) -	4.9391e + 0(3.82e - 2) -	4.9869e + 0(6.84e - 2) -	8.2902e - 1(1.37e - 2)

Problem	М	KPEA-N	KPEA-NK	KPEA-NR	KPEA
MaF13	3	8.3569e - 2(7.89e - 3) +	7.5092e - 2(5.68e - 3) +	9.1016e - 2(1.34e - 2) =	4.9983e - 1(1.41e - 2)
	5	1.4420e - 1(2.90e - 2) =	1.2586e - 1(1.66e - 2) +	1.9030e - 1(3.34e - 2) -	2.5636e - 1(8.44e - 3)
	8	1.6562e - 1(1.58e - 2) -	1.4569e - 1(1.31e - 2) =	2.1786e - 1(4.53e - 2) -	1.6289e - 1 (4.20e - 3)
	10	1.7024e - 1(1.66e - 2) -	1.5308e - 1(1.14e - 2) -	2.2327e - 1(2.87e - 2) -	1.3634e - 1(1.59e - 3)
+/-/=		5/32/7	11/29/4	10/30/4	

TABLE 9: Continued.

TABLE 10: Overall performance comparison based on the Wilcoxon rank-sum test based on HV and IGD metrics on the entire benchmark problem.

	Problem	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA
Overall performance	comparison based	on the inverted genera	ational distance			
	MaF	40/4/0	32/7/5	31/9/4	32/6/6	34/4/6
$V DE A \left(\left(- 1 \right) \right)$	MaF	40/4/0	32/7/5	31/9/4	32/6/6	34/4/6
RPEA(-1=1+)	WFG	33/3/0	21/12/3	16/12/8	36/0/0	30/2/4
	Overall	73/7/0	53/19/8	47/21/12	68/6/6	64/6/10
Overall performance	comparison based	on the hypervolume				
	MaF	41/3/0	33/6/5	33/9/2	29/10/5	40/4/0
KPEA (-/=/+)	WFG	33/1/2	17/9/10	13/21/2	33/3/0	33/2/1
	Overall	74/4/2	50/15/15	46/30/4	62/13/5	73/6/1

KPEA outperforms k-NSGAII, KnEA, Pi-MOEA, hpaEA, and VaEA with statistical significance is 74/80, 50/80, 46/ 80, 62/80, and 73/80, respectively. Moreover, the superiority of KPEA in terms of convergence and diversity is visually validated by Figure 14. These bar charts clearly indicate that KPEA achieves the best results of both HV and IGD indicators in most test instances.

Furthermore, in order to quantify the overall performance of each algorithm, a more effective statistical test, namely, the Friedman test, is also employed. The Friedman test ranks based on the IGD and HV results are presented in Table 11. KPEA ranks 1st in both HV and IGD metrics of the combined test suite. In addition, the bar chart in Figure 15 also intuitively shows that KPEA performs much better than the other five state-of-the-art algorithms in terms of the Friedman test ranking.

Both the results of the Wilcoxon rank-sum test and the average performance ranks indicate that KPEA has significant advantages in terms of convergence and diversity.

6. Discussions

In KPEA, a good balance between convergence and diversity is achieved through an adaptive switching mechanism. The knee point-driven selection mechanism increases selection pressure. The convergence and diversity of KPEA are maintained alternatively by the penalty mechanism. The experimental results in Section 5 demonstrate that KPEA can effectively solve optimization problems with different PFs.

In KPEA, when the number of nondominated solutions is less than the population size, knee points are first preserved by the selection mechanism, which increases selection pressure and improves convergence. Then, the remaining solutions are selected based on the principle of maximumvector-angle-first, which maintains diversity. It is worth mentioning that both KnEA and k-NSGAII are knee pointdriven algorithms that use adaptive neighborhood strategies to identify knee points. In addition, Pi-MOEA uses the average ranking method to identify pivot solutions within the neighborhood, which is similar to the strategy of identifying knee points. However, in KnEA and Pi-MOEA, the remaining nondominated solutions are selected based on the distance of each solution to the hyperplane. Therefore, this selection strategy emphasizes convergence too much but neglects the importance of diversity. Similarly, k-NSGAII uses parameter η to control the width of the neighborhood, which leads to difficulty in maintaining satisfactory diversity.

Compared to the worse-elimination principle in VaEA, the penalty mechanism in KPEA can better balance the convergence and diversity of the population. However, the penalty mechanism uses a fixed neighborhood range to penalize solutions, which may be not suitable for addressing the problem with mixed PFs.

It is a challenging task to set an appropriate neighborhood range for the solution. Because too high or too low thresholds will affect the performance of the algorithm. Estimating the shape of the PFs or adaptively adjusting the neighborhood range may be a feasible approach.

Furthermore, recent research has pointed out that DRSs can have a significant impact on the performance of MaOEAs. Currently, scholars have proposed various strategies to eliminate DRSs, such as the direct identification of



FIGURE 14: A comparison of the Wilcoxon rank-sum test performance on benchmark problems between KPEA and state-of-the-art algorithms.

Problem	k-NSGAII	KnEA	Pi-MOEA	hpaEA	VaEA	KPEA
Friedman's rank b	ased on inverted ger	nerational distance				
MaF	5.0455	3.3636	3.2273	3.3636	4.2500	1.7500
WFG	4.7222	2.8611	1.9861	5.9444	3.6806	1.7056
Overall	4.9000	3.1375	2.6688	4.5250	3.9937	1.7750
Friedman's rank b	ased on hypervolum	e				
MaF	4.9886	3.4091	2.7955	3.3409	4.9432	1.5227
WFG	4.5139	2.4889	2.3667	5.5972	4.3194	2.0139
Overall	4.7750	2.9500	2.5125	4.3563	4.6625	1.7437

TABLE 11: Overall performance comparison based on the Friedman test ranking based on HV and IGD metrics on the entire benchmark problem.



FIGURE 15: The average ranks of the Friedman test among all considered MaOEAs on WFG and MaF problems. (a) Average ranks of all considered MaOEAs based on all IGD results. (b) Average ranks of all considered MaOEAs based on all HV results.

DRSs and the indirect elimination of DRSs by identifying boundary solutions. However, some boundary solutions of the extremely convex Pareto front (ECPF) are often mistakenly identified as DRSs. Indirect elimination of DRSs also cannot guarantee the identification of all boundary solutions. In real-world applications, such as unmanned aerial vehicle route planning [66] and software product line configuration [67], it is crucial to preserve ECPF boundary solutions. Therefore, it is necessary to accurately identify DRSs on the ECPF.

7. Conclusions

As the number of objectives increases, the performance of algorithms based on the Pareto dominance is severely affected. Recently, the angle-based selection strategies have been demonstrated to effectively address this problem. Motivated by this, this paper proposes a knee point-driven manyobjective evolutionary algorithm with adaptive switching mechanism (KPEA). KPEA incorporates the knee points and the weighted distance into its mating strategy to increase the probability of generating excellent offspring. And an interquartile range method is introduced to detect and eliminate DRSs in the population. Moreover, in the environmental selection, an adaptive switching mechanism between angle-based selection and penalty is designed to balance convergence and diversity.

The performance of KPEA is compared with five stateof-the-art algorithms, namely, k-NSGAII, KnEA, Pi-MOEA, hpaEA, and VaEA, on WFG and MaF problems. The experimental results indicate that, in terms of IGD and HV, KPEA outperforms its competitors in most test instances. Additionally, this paper also discusses the sensitivity of parameter r and provides recommendations for parameter settings.

KPEA still has some issues that require further research. First, when dealing with MaOPs with complex PFs, the neighborhood range of solutions will affect the coverage of the population on the Pareto front. Adaptively adjusting the neighborhood range during the evolutionary process is a feasible research direction. In addition, the DRS elimination strategy may face a dilemma between eliminating DRSs and preserving boundary solutions on some problems. Future research will focus on designing an effective strategy to differentiate between DRSs and boundary solutions.

Data Availability

The operational data used to support the findings of this study are included in the article.

Conflicts of Interest

The author states that this research paper does not have any known competitive economic interests or personal relationships.

Acknowledgments

This research was supported by the Innovation Project of the Engineering Research Center of Integration and Application of Digital Learning Technology (No. 1221003).

References

- M. Mehrabi, H. Taheri, and P. Taghdiri, "An improved DVhop localization algorithm based on evolutionary algorithms," *Telecommunication Systems*, vol. 64, no. 4, pp. 639–647, 2017.
- [2] A. H. Gandomi and A. R. Kashani, "Automating pseudo-static analysis of concrete cantilever retaining wall using evolutionary algorithms," *Measurement*, vol. 115, pp. 104–124, 2018.
- [3] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multiobjective workflow scheduling in cloud," *IEEE Transactions* on *Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1344– 1357, 2016.
- [4] L. Gao, R. Liu, F. Wang et al., "An advanced quantum optimization algorithm for robot path planning," *Journal of Circuits*, *Systems and Computers*, vol. 29, no. 8, article 2050122, 2020.
- [5] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms," *ACM Computing Surveys (CSUR)*, vol. 48, no. 13, pp. 1–35, 2015.
- [6] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in 2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings, pp. 233–238, New Orleans, LA, USA, 2002.
- [7] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 2, pp. 957–962, Seoul, Korea, 2001.
- [8] S. Sudeng and N. Wattanapongsakorn, "Adaptive geometric angle-based algorithm with independent objective biasing for pruning Pareto-optimal solutions," in 2013 Science and Information Conference, pp. 514–523, London, UK, 2013.
- [9] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions* on Evolutionary Computation, vol. 11, no. 6, pp. 770–784, 2007.
- [10] M. Li and X. Yao, "What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation," *Evolutionary Computation*, vol. 28, no. 2, pp. 227–253, 2020.
- [11] N. Li, L. Ma, G. Yu, B. Xue, M. Zhang, and Y. Jin, "Survey on evolutionary deep learning: principles, algorithms, applications and open issues," ACM Computing Surveys, vol. 56, no. 41, pp. 1–34, 2023.

- [12] L. Ma, N. Li, Y. Guo et al., "Learning to optimize: reference vector reinforcement learning adaption to constrained manyobjective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 12698– 12711, 2022.
- [13] L. Yuan, N. Zhu, K. Li, M. Li, J. Zheng, and K. Li, "An angle dominance criterion for evolutionary many-objective optimization," *Information Sciences*, vol. 509, pp. 376–399, 2020.
- [14] S. Yang, M. Li, X. Liu, and J. Zhang, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721– 736, 2013.
- [15] B. Khan, S. Hanoun, M. Johnstone, C. P. Lim, D. Creighton, and S. Nahavandi, "A scalarization-based dominance evolutionary algorithm for many-objective optimization," *Information Sciences*, vol. 474, pp. 236–252, 2019.
- [16] Q. Gu, H. Chen, L. Chen, X. Li, and N. N. Xiong, "A manyobjective evolutionary algorithm with reference points-based strengthened dominance relation," *Information Sciences*, vol. 554, pp. 236–255, 2021.
- [17] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 331– 345, 2019.
- [18] F. Yang, X. Liang, X. Chu, and S. Wang, "A new dominance relation based on convergence indicators and niching for many-objective optimization," *Applied Intelligence*, vol. 51, no. 8, pp. 5525–5542, 2021.
- [19] H. Li, J. Deng, Q. Zhang, and J. Jin, "Adaptive epsilon dominance in decomposition-based multiobjective evolutionary algorithm," *Swarm and Evolutionary Computation*, vol. 45, pp. 52–67, 2019.
- [20] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [21] J. Zhou, X. Yao, L. Gao et al., "A decomposition based evolutionary algorithm with direction vector adaption and selection enhancement," *Information Sciences*, vol. 501, pp. 248–271, 2019.
- [22] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [23] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
- [24] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329–346, 2017.
- [25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [26] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [27] K. Shang and H. Ishibuchi, "A new hypervolume-based evolutionary algorithm for many-objective optimization," *IEEE*

Transactions on Evolutionary Computation, vol. 24, no. 5, pp. 839–852, 2020.

- [28] Y. Sun, G. G. Yen, and Z. Yi, "IGD indicator-based evolutionary algorithm for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 173–187, 2019.
- [29] L. Zhao, Y. Ren, Y. Zeng, Z. Cui, and W. Zhang, "A knee pointdriven many-objective pigeon-inspired optimization algorithm," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 4277–4299, 2022.
- [30] K. Deb and S. Gupta, "Understanding knee points in bicriteria problems and their implications as preferred solution principles," *Engineering Optimization*, vol. 43, no. 11, pp. 1175– 1204, 2011.
- [31] G. Yu, Y. Jin, and M. Olhofer, "A method for a posteriori identification of knee points based on solution density," in 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, Rio de Janeiro, Brazil, 2018.
- [32] F. Zou, G. G. Yen, and L. Tang, "A knee-guided prediction approach for dynamic multi-objective optimization," *Information Sciences*, vol. 509, pp. 193–209, 2020.
- [33] L. Wei and E. Li, "An effective offspring generation strategy for many-objective optimization driven by knee points under variable classification," *Applied Intelligence*, vol. 52, no. 15, pp. 17508–17533, 2022.
- [34] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761– 776, 2015.
- [35] V. Palakonda, J. Kang, and H. Jung, "An adaptive neighborhood based evolutionary algorithm with pivot- solution based selection for multi- and many-objective optimization," *Information Sciences*, vol. 607, pp. 126–152, 2022.
- [36] W. Li, R. Wang, T. Zhang, M. Ming, and K. Li, "Reinvestigation of evolutionary many-objective optimization: focus on the Pareto knee front," *Information Sciences*, vol. 522, pp. 193–213, 2020.
- [37] Z. Wang, Y. S. Ong, and H. Ishibuchi, "On scalable multiobjective test problems with hardly dominated boundaries," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 217–231, 2019.
- [38] K. S. Bhattacharjee, H. K. Singh, T. Ray, and Q. Zhang, "Decomposition based evolutionary algorithm with a dual set of reference vectors," in 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 105–112, Donostia, Spain, 2017.
- [39] Y. Su, J. Wang, L. Ma et al., "A hybridized angleencouragement-based decomposition approach for manyobjective optimization problems," *Applied Soft Computing*, vol. 78, pp. 355–372, 2019.
- [40] M. Zhang and H. Li, "A reference direction and entropy based evolutionary algorithm for many-objective optimization," *Applied Soft Computing*, vol. 70, pp. 108–130, 2018.
- [41] B. Wu, W. Hu, Z. He, M. Jiang, and G. G. Yen, "A manyobjective particle swarm optimization based on virtual Pareto front," in 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, Rio de Janeiro, Brazil, 2018.
- [42] R. S. Laundy, "Multiple criteria optimisation: theory, computation and application," *Journal of the Operational Research Society*, vol. 10, no. 1, pp. 89-90, 1988.
- [43] H. Seada, M. Abouhawwash, and K. Deb, "Multiphase balance of diversity and convergence in multiobjective optimization,"

IEEE Transactions on Evolutionary Computation, vol. 23, no. 3, pp. 503–513, 2019.

- [44] Y. Xiang, Y. Zhou, X. Yang, and H. Huang, "A many-objective evolutionary algorithm with Pareto-adaptive reference points," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 99–113, 2020.
- [45] Y. Zhou, Y. Xiang, Z. Chen, J. He, and J. Wang, "A scalar projection and angle-based evolutionary algorithm for manyobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2073–2084, 2019.
- [46] X. Cai, Z. Yang, Z. Fan, and Q. Zhang, "Decomposition-basedsorting and angle-based-selection for evolutionary multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2824–2837, 2017.
- [47] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 131–152, 2017.
- [48] Z. He and G. G. Yen, "Many-objective evolutionary algorithms based on coordinated selection strategy," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 220–233, 2017.
- [49] X. Li, X. Li, K. Wang, and S. Yang, "A strength Pareto evolutionary algorithm based on adaptive reference points for solving irregular fronts," *Information Sciences*, vol. 626, pp. 658– 693, 2023.
- [50] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 183–195, 2011.
- [51] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.
- [52] H. Chen, Y. Tian, W. Pedrycz, G. Wu, R. Wang, and L. Wang, "Hyperplane assisted evolutionary algorithm for manyobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3367–3380, 2020.
- [53] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Mining*, vol. 10, no. 1, p. 35, 2017.
- [54] I. Das, "On characterizing the "knee" of the Pareto curve based on normal-boundary intersection," *Structural Optimization*, vol. 18, no. 2-3, pp. 107–115, 1999.
- [55] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2689–2702, 2017.
- [56] R. W. R. C. Purshouse and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.
- [57] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
- [58] R. Cheng, M. Li, Y. Tian et al., "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.
- [59] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [60] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms,"

IEEE Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 174–188, 2003.

- [61] Z. Wang, Y. S. Ong, J. Sun, A. Gupta, and Q. Zhang, "A generator for multiobjective test problems with difficult-to-approximate Pareto front boundaries," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 556–571, 2019.
- [62] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [63] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675– 701, 1937.
- [64] R. B. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.
- [65] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.
- [66] Q. Tan, Z. Wang, Y. S. Ong, and K. H. Low, "Evolutionary optimization-based mission planning for UAS Traffic Management (UTM)," in 2019 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 952–958, Atlanta, GA, USA, 2019.
- [67] A. Syarif, A. Abouaissa, L. Idoumghar, A. Kodar, and P. Lorenz, "Search based software engineering on evolutionary multi-objective approach," in 2016 IEEE International Conference on Communications (ICC), pp. 1–6, Kuala Lumpur, Malaysia, 2016.