Egyptian Informatics Journal 24 (2023) 100405

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Decomposition-based interval multi-objective evolutionary algorithm with adaptive adjustment of weight vectors and neighborhoods

Yaqing Jin, Zhixia Zhang, Liping Xie, Zhihua Cui *

Shanxi Key Laboratory of Big Data Analysis and Parallel Computing, Taiyuan University of Science and Technology, Taiyuan, 030024, Shanxi, China

ARTICLE INFO

Keywords:
Decomposition-based multi-objective evolutionary algorithm
Interval parameter
Weight vector
Neighborhood adjustment

ABSTRACT

Interval multi-objective optimization problems (IMOPs) are one of the most critical optimization problems in practical applications. However, compared to deterministic multi-objective optimization problems (MOPs), there are few researchs addressing IMOP. In addition, the uncertainty contained in the problem makes the distribution of the population more challenging. Therefore, this paper proposed a decomposition-based interval multi-objective evolutionary algorithm with adaptive adjustment of weight vectors and neighborhoods (IMOEA/D-AWN). Firstly, an interval sparsity level function (ISL) is constructed to measure the density of individuals, and a comprehensive ranking of interval sparsity ranking and interval uncertainty ranking is proposed. For the purpose of improving the distribution of the population while reducing its uncertainty, based on the above comprehensive ranking, a new adaptive adjustment weight vector strategy guided by interval elite population is designed. Besides, an adaptive adjustment neighborhoods strategy is designed. This strategy adjusts individuals' neighborhood size according to the number of iterations to improve the efficiency of evolution. Finally, the IMOEA/D-AWN is evaluated on 17 interval benchmark test problems and a collaborative computation offloading optimization problem, and compared with four advanced multi-objective evolutionary algorithms with interval parameters (IMOEAs). Experimental results show that this algorithm performs well in convergence, diversity, and uncertainty.

1. Introduction

In real-world applications, there exist various optimization problems with multiple conflicting objectives [1–3], which are known as MOPs [4–6]. Unlike single objective optimization problems, the purpose of solving MOPs is to find a set of optimal solutions which form its Pareto Front in the objective space [7–9]. Multi-objective evolutionary algorithms (MOEAs) can effectively solve MOPs [10–12], which are divided into three main categories: Based on Pareto [13], decomposition-based [14], and indicator-based [15]. Among them, MOEA/D is the classic decomposition-based algorithm, which decomposes MOPs into multiple single-objective subproblems through a uniformly distributed set of weight vectors and optimizes these subproblems collaboratively. At present, there are various improved MOEA/D algorithms that can solve MOPs well [16]. However, in real production, the parameters of the objective function are usually uncertain due to various factors. Uncertain parameters can be represented as fuzzy numbers, random variables, intervals, etc. However, the membership function of fuzzy numbers and the distribution function of random variables are difficult to obtain in advance, while the interval information of uncertain parameters is easier to obtain [17]. Moreover, fuzzy numbers and random variables can be transformed to interval numbers by confidence level and cut set [18], respectively, and the interval numbers have a clear operation criterion. Therefore, IMOPs are receiving increasing attention from researchers in various fields.

Currently, there are two main approaches to solve IMOPs through evolutionary algorithms (EAs). The first method mainly transforms IMOPs into deterministic MOPs using information such as the midpoint, width, and endpoints of the interval, and solves the transformed problem using MOEAs. Zhang et al. [19] transformed IMOPs into deterministic MOPs by using the midpoint and width of the interval values as new objective functions. Fu et al. [20] considered the radius and midpoint of the initial objective functions as two objectives of equal importance and solved them using MOEA. Liu et al. [21] used nonlinear interval number programming to convert each objective function containing uncertain information to a deterministic single-objective problem, and solved the nested optimization problems using two genetic algorithms. Li et al. [22] established the interval multi-objective opti-

* Corresponding author.
E-mail addresses: yaqingjin1@163.com (Y. Jin), 15634969919@163.com (Z. Zhang), 2001052@tyust.edu.cn (L. Xie), cuizhihua@tyust.edu.cn (Z. Cui).

https://doi.org/10.1016/j.eij.2023.100405
Received 30 June 2023; Received in revised form 15 September 2023; Accepted 21 September 2023
Available online 3 October 2023

mal dispatch of microgrid (MODMG) model to meet the requirements of economy, security of microgrid and power quality, and transformed the interval MODMG into MOPs by simultaneously optimizing the midpoint of each interval objective function. Guo et al. [23] designed a knowledge-induced MOEA/D to solve IMOPs by using the midpoint of the interval value as the objective. Recently, He et al. [24] used the midpoint of the interval as the objective value and reacted the robustness of the solution with the length of the interval. And designed Knee-based and Interval-based MOEA to solve portfolio optimization problem. Although the above approaches are simple and easy to implement, they can lead to the loss of interval information, resulting in some differences between the transformed problem and the original problem. Moreover, different transformation methods will produce different MOPs, and the final solution obtained can hardly reflect the uncertainty of the problem.

The second approach designs IMOEAs by defining interval Pareto dominance relation for the direct solving of IMOPs. This method can avoid losing uncertain information and obtain more accurate solutions. Therefore, this paper mainly focuses on the second approach. Limbourg et al. [25] first defined interval Pareto dominance relation (IPDR) based on interval order relation $\succ_{IP}$, and calculated interval crowding distance (ICD) by defining the degree of contribution of individuals to the hypervolume using interval upper and lower limits, but this approach has high computational complexity. Jiang et al. [26] defined the interval probability dominance relationship, subdivided the positions of two intervals into six cases, and provided specific calculations for each. Gong et al. [17] proposed an IPDR using the interval confidence level, and defined an ICD using the midpoint and volume of each interval, and overlap between the two intervals, proposing the II-MOEA algorithm. On this basis, Sun et al. [27] designed interval Pareto dominance relationship based possibility degree lower bound through confidence threshold parameter γ. The algorithm can obtain different Pareto frontiers by adjusting the value of γ, which has greater flexibility. Gong et al. [28] developed the theory of preference polyhedron for the IMOP and based on this, they designed the sorting of individuals. Zhang et al. [29] analyzed the features of μ [30] and P [31] in sorting intervals and proposed an integrated sorting assistance NSGA-II for IMOPs. Zhang et al. [32] chose interval confidence as the interval ranking method and proposed interval cooperative multiobjective artificial bee colony algorithm. Sun et al. [18] proposed the IMOMA-II by integrating the local search into the existing IMOEA. The fitness function of individuals is calculated using the contribution of individuals to the hypervolume and the uncertainty of individuals in the local search stage, so that the individuals with excellent convergence, diversity and uncertainty are more likely to enter the next generation population. Liu et al. [33] defined a new interval crowding distance by the ratio of the distance between two intervals on a certain objective and the distance between the maximum and minimum intervals on this objective, and modified the interval confidence dominance method to compare each individual in the external elite population and evolution population, and proposed the DI-μMOGA algorithm. Yi et al. [34] used parallel cell entropy to subdivide the evolutionary state to three stages, and designed different interval dominance relationships for the different stages, in addition to interval crowding distances that take into account the shape of the intervals.

In recent years, Sun et al. [35] proposed adaptive interval confidence and developed a probabilistic wind power interval prediction model. Xu et al. [36] proposed a hybrid search IMOEA that includes modified gray wolf optimization and pattern search. Zhang et al. [37] modified the calculation of interval distance in interval confidence level and proposed the InMaOEA. Chen et al. [38] proposed a modified IPDR, and ICD calculation with respect to the mixture of real values and intervals, and proposed the interval multi-objective particle swarm optimization algorithm. Xu et al. [39] designed an angle-based ICD (ICA), providing a new way to calculate ICD. They also defined interval knee point to help decision-makers make decisions, and proposed an interval elite selection strategy to generate better performance offspring. However, this algorithm is for bi-objective optimization problems only, corresponding ICA still needs to be designed when solving tri-objective optimization problems. The above algorithms all use the framework of Pareto-based IMOEAs. However, for the most IMOPs, particularly for over-concave, over-convex or discontinuous Pareto Fronts (PFs), it is hard for the above algorithms to produce the well distributed irregular PFs due to the lack of a diversity maintained mechanism for irregular PFs. To solve the above problems, Gan et al. [40] proposed a new IMOEA within the framework of MOEA/D, which explored a new way of solving IMOPs. They integrate various interval Pareto dominance relationships and propose an integrated comparison strategy. Besides, an adaptive reference vectors adjustment strategy was proposed using interval crowding distance, which significantly improves the computational efficiency, and the convergence and diversity of the population. However, the uncertain information introduced brings more challenges to the distribution of the population, and the uncertainty of the interval conflicts with diversity and convergence, so the algorithm does not significantly reduce the uncertainty of the population.

Currently, a large number of studies have focused on interval Pareto dominance relations, in contrast to few studies that have considered designing interval diversity maintenance strategies [39,41]. However, the uncertainty negatively affects population diversity [40], and, for most IMOPs, especially for problems with irregular PFs, it is difficult to generate optimal solution sets that are uniformly distributed on the true PFs because the majority of present IMOEAs lack diversity maintenance methods designed for irregular PFs. IMOEA/D [40] significantly improved the diversity of the population, however, it did not significantly reduce the uncertainty of the population as the uncertainty conflicted with the diversity and convergence. Thus, to better balance the three performances of IMOEA, i.e., to improve diversity and convergence while reducing population uncertainty, this paper proposes a decomposition-based IMOEA with adaptive adjustment of weight vectors and neighborhoods (IMOEA/D-AWN). In order to increase diversity while reducing population uncertainty, we introduced uncertainty information when adjusting the weight vectors, and designed an adaptive weight vector adjustment strategy based on the dual ranking of interval sparsity ranking and interval uncertainty ranking. In addition, in order to ensure that the population converges to the true PFs, an external archive is constructed, denoted as the interval elite population (IEP), which stores the interval Pareto non-dominant individuals. And IEP is merged with the current population prior to each adjustment of the weight vectors, picking the most appropriate solution for each of the weight vectors in order to update the current population, which indirectly guides the updating of the weight vectors. Furthermore, since the objective values of IMOPs are interval values rather than exact values, comparing the size of the interval values and deriving interval non-dominated solutions increases the computational cost, and in decomposition-based IMOEAs, a fixed-size neighborhood produces the computational resources [42]. Based on the above reasons, we design an adaptive neighborhood adjustment strategy to improve evolutionary efficiency.

The main contributions of this paper are as follows:

(1) To improve the population distribution and reduce uncertainty, a dual-ranking adaptive weight vector adjustment strategy based on interval sparsity ranking and interval uncertainty ranking is proposed. The weights of the rankings are also discussed in detail. In addition, to ensure population convergence on the true PF, an IEP is established to update the contemporary population before adjusting the weight vector.
(2) In order to improve the evolutionary efficiency and avoid wasting computational resources, an adaptive neighborhood adjustment strategy is proposed. In the early stage of evolution, individuals have large neighborhoods to accelerate global search. As the evolution progresses, the neighborhood size gradually decreases to

ensure local search. Reduced runtime with guaranteed algorithm performance.

The remainder of this paper is structured as follows: Section 2 introduces some relevant definitions. Section 3 elaborates the adaptive weight vectors adjustment strategy, the adaptive neighborhoods adjustment strategy, and the framework of the proposed algorithm. Experimental results and analysis are presented in Section 5. Finally, the main conclusions and future works of this paper are presented.

2. Preliminaries

2.1. Model of IMOPs

Generally, a minimization IMOP is described as [33]:

$$\begin{aligned} \min \quad & f(x,c) = (f_1(x,c), f_2(x,c), ..., f_M(x,c)) \\ s.t. \quad & x = [x_1, x_2, ..., x_d] \in S \subseteq R^n, \\ & c = [c_1, c_2, ..., c_d]^T, c_j = [\underline{c_j}, \overline{c_j}], j = 1, 2, ..., K \end{aligned} \tag{1}$$

where $x = [x_1, x_2, ..., x_d] \in S \subseteq R^n$ is a d-dimensional decision variable. S is a d-dimensional decision space. $f(x,c) \in \Omega \subseteq R$ is the M-dimensional objective space. $f_k(x,c) = [\underline{f_k}(x,c), \overline{f_k}(x,c)], k = 1, 2, ..., M$ is the k-th component of an IMOP, and c is a vector with K parameters and its j-th component c_j is an interval value, where $\underline{f_k}(x,c)$ and $\overline{f_k}(x,c)$ are the lower and upper bounds of this interval.

2.2. Definitions of interval

Definition 1. Interval [34]: One interval can be expressed as $A = [\underline{A}, \overline{A}]$, where $\underline{A} < \overline{A} \in R$. $\underline{A}$ and $\overline{A}$ are the lower limit and the upper limit, respectively. In particular, A can regress to a point and called a point interval when $\underline{A} = \overline{A}$.

Definition 2. Interval Arithmetic Operations [40]: For two intervals A and B

$$\begin{aligned} & A + B = [\underline{A} + \underline{B}, \overline{A} + \overline{B}], \ A - B = [\underline{A} - \overline{B}, \overline{A} - \underline{B}], \\ & \gamma A = [\gamma \underline{A}, \gamma \overline{A}], \gamma > 0; \ \gamma A = [\gamma \overline{A}, \gamma \underline{A}], \gamma < 0. \end{aligned} \tag{2}$$

Definition 3. Interval confidence level [17]: For two intervals A and B, their minimum interval is expressed as $K = [\underline{K}, \overline{K}]$, where $\underline{K} = min\{\underline{A}, \underline{B}, \overline{A}, \overline{B}\}$ and $\overline{K} = min\{\{\underline{A}, \underline{B}, \overline{A}, \overline{B}\} \backslash \underline{K}\}$, respectively. The interval confidence level of A min than B is defined as [17]:

$$P(A \le B) = \frac{d(B,K)}{d(A,K) + d(B,K)} \tag{3}$$

where $d(B,K)$ is the distance between B and K, calculated as [17]:

$$d(B,K) = \sqrt{\frac{(\underline{B} - \underline{K})^2 + (\overline{B} - \overline{K})^2}{2}} \tag{4}$$

The calculation of $d(A,K)$ is similar to Equation (4).

Definition 4. IPDR based on the lower limit of possibility degree γ [27]: $\gamma \in [0.5, 1]$ is the threshold, if $P(A \le B) \ge \gamma$, the probability that A is less than or equal to B can be considered to be no less than γ. For two solutions x_i and x_j, on any k-th objective, $k, 1 \le k \le M$, we can say that the x_i interval dominates x_j, denoted as $x_i \prec_{IN} x_j$, if all of the following conditions are met:

$$\begin{aligned} & \forall k, P(f_k(x_i, c)) < p(f_k(x_j, c)) \ge \gamma \\ & \exists k_0, P(f_k(x_i, c)) < p(f_k(x_j, c)) > \gamma \end{aligned} \tag{5}$$

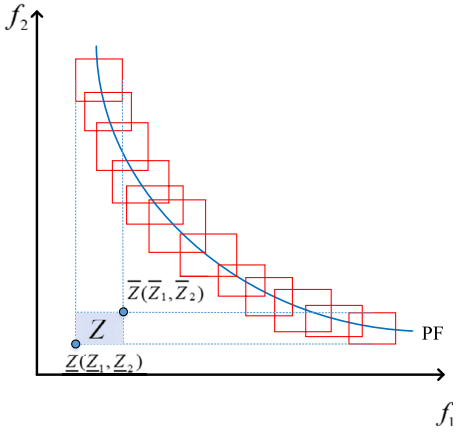

Fig. 1. Reference hypervolume schematic.

Where $k_0 \in 1, 2, ..., M$, if x_i does not dominate x_j and x_j does not dominate x_i, then x_i and x_j do not dominate each other, denoted as $x_i \parallel x_j$.

Definition 5. Interval distance [43]: The interval distance between two intervals is calculated as [43]:

$$D(A,B) = \sqrt{[M(A) - M(B)]^2 + \frac{1}{3}[w^2(A) + w^2(B)] - \frac{2}{3}[w(A \cap B)]^2} \tag{6}$$

where $M(A) = (\underline{A} + \overline{A})/2$ is the midpoint of the interval, $w(A) = (\overline{A} - \underline{A})/2$ is the width of the interval A. $w(A \cap B)$ is the intersection of the interval A and B. This definition integrates the deviation of each point in the number of two intervals and the relationship between the positions of the interval numbers.

2.3. Interval-valued Chebyshev scalar function

IMOEA/D decomposes the IMOP into multiple subproblems and then co-optimizes the solutions of these subproblems to obtain the Pareto-optimal solutions of the IMOP [40].

As the red rectangle in Fig. 1 shows, each individual is a hypercube in objective space since the value of the function for each objective in IMOPs is an interval value. The blue curve in this figure is PF, and the blue rectangle at the bottom left is the reference hypercube, it consists of the minimum upper limit and minimum lower limit values of each objective in the population, as $Z = ([\underline{Z_1}, \overline{Z_1}], [\underline{Z_2}, \overline{Z_2}])$ in Fig. 1, where $\underline{Z} = (\underline{Z_1}, \underline{Z_2})$ is the lower reference point and $\overline{Z} = (\overline{Z_1}, \overline{Z_2})$ is the upper reference point. Therefore, for the IMOPs, the reference hypercube is $Z = ([\underline{Z_1}, \overline{Z_1}], ..., [\underline{Z_M}, \overline{Z_M}])$, where $\underline{Z_k} = \min \underline{f_k}(x, c)$, $\overline{Z_k} = \min \overline{f_k}(x, c), k = 1, 2, ...M$. The minimization Interval-valued Tchebycheff Scalar Function is defined as [40]:

$$g^{itc}(x|w, Z) = \max_{1 \le k \le M} \{w_k(f_k(x, c) - Z_k)\} \tag{7}$$

where, $w_j \in WV$ is the weight vector. Z is a reference hypercube and $Z_k = [\underline{Z_k}, \overline{Z_k}]$ is its k-th component.

3. Proposed approach

In this paper, we propose an IMOEA/D-AWN to improve the convergence and diversity of the population while reducing its uncertainty. To achieve this task, we propose an adaptive adjustment of the weight vectors (WVs) strategy based on the comprehensive ranking of interval sparsity ranking and interval uncertainty ranking to improve the diversity and uncertainty performance of the population. And establish an IEP to update the current population to enhance the performance of

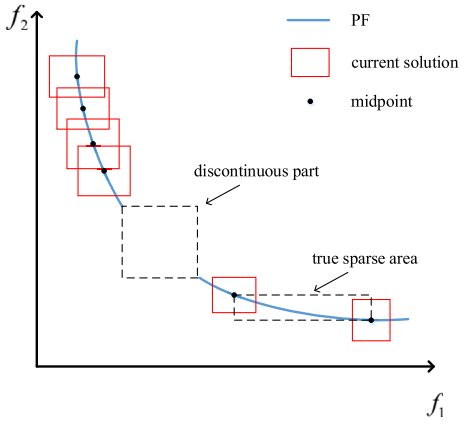

Fig. 2. The real sparse area and discontinuous part of the discontinuous PF.

the population, and to further guide the adjustment of the WVs. In addition, to improve evolutionary efficiency, an adaptive adjustment of neighborhood strategy from global to local search is proposed.

3.1. Adaptive adjustment of weight vectors strategy

Since uniformly distributed WVs do not ensure that uniformly distributed optimal solutions are produced on PFs with spikes and low tails [44], and, for irregular PFs, uniform WVs can adversely affect the population distribution [40]. Therefore, to ensure the distribution of population, the WVs need to be adjusted according to their sparsity. In addition, the introduction of uncertain information has a negative impact on the distribution of the population [40]. Therefore, individual uncertainty should also be taken into account when adjusting the weight vector. On the basis of the above considerations, we design an adaptive adjustment weight vectors strategy based on interval sparsity ranking and interval uncertainty ranking. Individuals with worse sparsity ranking indicate that their neighborhood distribution is more dense and that individual is a dense individual, and vice versa. Individuals with worse uncertainty ranking have larger interval uncertainty. Therefore, we will delete the weight vectors corresponding to those individuals with both poor sparsity ranking and uncertainty ranking, and generate new weight vectors using the weight vectors corresponding to individuals with excellent comprehensive ranking to make the population better distributed on the true PF (PF_{True}).

However, for some discontinuous PFs (as shown in Fig. 2), more computational resources are usually required to find truly sparse area rather than pseudo-sparse area, i.e., discontinuous parts [45]. In order to make the population distribute to the PF_{True}, IEP is established to update the contemporary population and thus guide the deletion and addition of weight vectors. Since there are no non-dominated solutions in the discontinuous part of PF [45], when the population converges to a certain level, the individuals in the IEP are considered as interval non-dominated solutions. This approach can ensure that the algorithm does not add too many new subproblems in discontinuous part. It also helps generate new subproblems in truly sparse regions so individuals can distribute in the PF_{True}. For regular PFs, due to the use of IEP which contains well distributed interval non-dominated solutions to update the current population, the diversity and convergence of the population can be further enhanced. The way to update the IEP is shown in Algorithm 1 (Line 1-11). Initially, the IEP is empty, so the interval non-dominated solutions among the current population P and the offspring population O are selected as the IEP. In the later stage of evolution, the IEP is not empty, so the interval non-dominated solutions among the IEP and offspring population O are selected as the updated IEP. If the size of the IEP ($|IEP|$) is larger than the predefined maximum IEP size $nIEP$, those individuals with the worst comprehensive ranking are removed from the IEP so that the $|IEP|$ is equal to $nIEP$. The way to update the contemporary population using IEP is shown in (Line 12-16). The most suitable individuals in the IEP and the current population are selected for each weight vector, i.e., if an individual y in the IEP can cause the scalar function of one weight vector to obtain a smaller interval value than an individual x in the current population P, replace x with y (Line 13-14), and subsequent operations to adjust the WVs are based on this updated population.

Algorithm 1 Adaptive adjustment of weight vectors strategy.

Input: Population P, offspring population O, weight vectors WV, interval elite populations IEP, max size of IEP: $nIEP$, the maximal number of subproblems adjusted: nus.
Output: Adjusted reference vectors WV, adjusted population P.
1: **if** is empty (IEP) **then** // Update the interval elite populations IEP
2: Merging population P with O and selecting the interval non-dominated solutions as updated IEP;
3: **else**
4: Merging IEP with O and selecting the interval non-dominated solutions as updated IEP;
5: **end if**
6: **if** $|IEP| > nIEP$ **then**
7: **for** $i = 1 : N$ **do**
8: Calculate the overall ranking $R(i)$ of individual i;
9: **end for**
10: Remove the worst ranked $|IEP| - nIEP$ solutions from the IEP;
11: **end if**
12: **for** $j = 1 : N$ **do** // Update the current population by IEP
13: **if** $g^{itc}\left(y\middle|w_j, Z\right) \leq_{IN} g^{itc}\left(x\middle|w_j, Z\right), y \in IEP,\ x \in P$ **then** //Refer to Eq. (7)
14: Set $x = y$;
15: **end if**
16: **end for**
17: **for** $i = 1 : N$ **do** //Deleting overcrowded weight vectors
18: Calculate the overall ranking $R(i)$ of individual i in updated population;
19: **end for**
20: Delete the worst ranked nus individuals from the updated population;
21: Delete the nus weight vectors from the WV corresponding to the worst ranked nus individuals;
22: **while** |Remain w| $< N$ **do** //Add new weight vectors to the sparse region.
23: **for** $r = 1 : nus$ **do**
24: Generate a new weight vector $w' = w_r\left(1 - 0.5\frac{|Remain\ w|-1}{|Remain\ w|}\right) + w_{r+1}\left(0.5\frac{|Remain\ w|-1}{|Remain\ w|}\right)$;
25: **end for**
26: $|Remain\ w| = |Remain\ w| + r$;
27: **end while**
28: Add the solutions of the new weight vectors to the current population P;
29: **end**

The adaptive adjustment strategy of the WVs based on comprehensive ranking is elaborated in the following. First, the interval sparsity level function is designed according to the interval objective function value, and for the i-th subproblem, its interval sparsity level is calculated as follows:

$$ISL(i) = \frac{\prod_{k=1}^{M} D(f_k(x_{i-1}, c), f_k(x_i, c)) + \prod_{k=1}^{M} D(f_k(x_i, c), f_k(x_{i+1}, c))}{2} \quad (8)$$

Where $D(*,*)$ is the same as Eq. (6). x_{i-1} and x_{i+1} are the two nearest individuals to individual x_i, and M is the number of objective functions. From this formula, we can know that the larger the ISL value is, the larger the interval distance between the current individual and its neighboring individuals, and the sparser the individual and the corresponding weight vector of that individual. The interval sparsity ranking R_{ISL} is obtained by sorting the ISL in descending order so that the sparse subproblems get the top ranking, while those dense ones get the bottom ranking.

Since IMOPs contain uncertain information: intervals, which leads to different solutions having overlapping parts in the objective space, it has a negative influence on the distribution of individuals [40]. Therefore, the uncertainty of individuals needs to be considered when adjusting the weight vectors. The interval uncertainty is calculated as follows:

$$I(i) = \prod_{k=1}^{M} (\overline{f_k}(x_i, c) - \underline{f_k}(x_i, c)) \quad (9)$$

The product of the interval lengths of an individual x_i on every objective is its uncertainty, and the smaller the value of $I(i)$, the smaller the uncertainty of that individual. For uncertainty problems, decision makers tend to prefer solutions that can lower the uncertainty of the algorithm. Therefore, the interval uncertainty ranking R_I is obtained by sorting I in ascending order so that those solutions with small uncertainty get the top ranking, while those individuals with larger uncertainty get the bottom ranking.

Based on the above two rankings, we can obtain a comprehensive ranking for each subproblem, which is calculated as:

$$R(i) = \alpha R_{ISL}(i) + (1-\alpha) R_I(i) \tag{10}$$

Where α is the weight, the larger α is, the more the comprehensive ranking is influenced by interval sparsity, and vice versa, the more it is influenced by interval uncertainty. The setting and effectiveness of α will be discussed in Section 4.

The larger the R value of the subproblem, the more dense and uncertain it is. The denser it is, the worse the diversity of the subproblem. The larger the uncertainty, the more negative impact on the distribution of the population. Therefore, the *nus* subproblems with the largest R values need to be deleted, and the number of remaining WVs is noted as |Remain w|, |Remain w| $= N - nus$, and new *nus* WVs are inserted in the sparse region. Since those subproblems with the smallest R values are relatively sparse and have small uncertainty, i.e., the negative impact on the population distribution is relatively small. Therefore, it is necessary to generate new weight vectors w' using these subproblems. Select the best ranked weight vectors in turn, denote the order numbers of the neighboring ranked weight vectors as r and $r+1$ respectively, and then add a new weight vector w' between w_r and w_{r+1}. Since the rank difference between them is 1, the newly inserted weight vector w' is closer to the better ranked weight vector w_r:

$$w' = w_r \left(1 - 0.5\frac{|Remain\ w| - 1}{|Remain\ w|}\right) + w_{r+1}\left(0.5\frac{|Remain\ w| - 1}{|Remain\ w|}\right) \tag{11}$$

where w_r is the weight vector with rank r, w_{r+1} is the weight vector with rank $r+1$, and w' is the newly generated weight vector. *nus* is the number of adjusted WVs. Algorithm 1 is the pseudo-code for adaptively adjusting the WVs. Lines 17-21 show the steps to delete the weight vectors, and lines 22-27 are the steps to add new weight vectors.

3.2. Adaptive adjustment of neighborhoods

Since the size of the neighborhood has an influence on the algorithm's performance [44], specifically, larger neighborhoods usually increase the diversity of the population but also increase the complexity of the algorithm. Conversely, smaller neighborhoods reduce the complexity but tend to slip into local optima. The neighborhood size of MOEA/D always remains consistent throughout the evolutionary process, which has a negative impact on the complexity and performance of the algorithm.

In order to reduce the waste of computing resources and improve evolutionary efficiency, a neighborhoods adaptive adjustment strategy is designed. At the early stage of evolution, the setting of a large neighborhood is needed to accelerate the global convergence of the population. With the progression of evolution, the size of the neighborhood needs to be gradually reduced to ensure local convergence. Such a setting can efficiently allocate computing resources and improve evolutionary efficiency. The size of the neighborhood is adjusted according to the following formula:

$$T' = T_{max} \times e^{-(\frac{G}{G_{max}})} \tag{12}$$

where T' is the adjusted neighborhood size, which is adaptively adjusted after each iteration. $T_{max} = \frac{3}{2}T$, T is the size of the neighborhood, T_{max} is the maximum neighborhood size. G is the current number of iterations, and G_{max} is the maximum number of iterations. The pseudo-code for adaptive adjustment of neighborhood size is in Algorithm 2. Since the size of neighborhoods is continuously adjusted with the number of iterations (Line 2), the neighborhoods of each individual are also changing (Line 5).

Algorithm 2 Adaptive adjustment of neighborhoods.

Input: Maximum number of iterations G_{max}, maximum neighborhood size T_{max}, neighbors of each weight vector $B(i) = i_1, i_2, \ldots, i_T$
Output: New neighborhood size T', new neighborhoods $B'(i)$.

1: **while** $G < G_{max}$ **do**
2: $T' = T_{max} \times e^{-(\frac{G}{G_{max}})}$;
3: **end while**
4: **for** $i = 1 : N$ **do**
5: $B'(i) = i_1, i_2, \ldots, i_{T'}$;
6: **end for**

3.3. Algorithm flow

Algorithm 3 IMOEA/D-AWN.

Input: Maximum number of iterations G_{max}, maximum neighborhood size T_{max}, ratio of iterations to evolve $rate_{evol}$, period of updating the weight vector wag, population size N, interval elite population $IEP = \emptyset$.
Output: Population P.

1: Initialization population P_0, uniformly distributed weight vectors $WV \Rightarrow w_1; w_2; \ldots; w_N$, the T nearest neighbors of each weight vector $B(i) = i_1, i_2, \ldots, i_T$, the reference hypercube Z;
2: $G = 1$;
3: $IEP = \emptyset$;
4: **while** $G < G_{max}$ **do**
5: **for** $i = 1 : N$ **do**
6: Perform evolutionary operation in neighborhood $B(i)$ produce offspring $O(i)$;
7: Update the reference hypercube Z;
8: **for** $x_j \in B(i)$ **do**
9: **if** $g^{itc}(O(i)|w_j, Z) \leq_{IN} g^{itc}(x_j|w_j, Z)$ **then** // Refer to Eq. (7)
10: Set $x_j = O(i)$;
11: **end if**
12: **end for**
13: **end for**
14: $G++$;
15: Adaptively update the neighborhood size T and the neighborhoods B; // Refer to Algorithm 2
16: **if** $G \geq rate_{evol} * G_{max}$
17: Update the IEP by the offspring O based on interval non-dominance relations and overall ranking R; // Refer to Algorithm 1
18: **if** $mod(G, wag) == 0$ **then**
19: Adjust the weight vector WV; // Refer to Algorithm 1
20: Update the neighborhood $B(i)$;
21: **end if**
22: **end if**
23: **end while**
24: $P \leftarrow P_{G_{max}}$;

Algorithm 3 is the pseudo-code for IMOEA/D-AWN. First, a population P_0 of size N and a set of uniformly distributed weight vectors WV are initialized, the neighborhoods of each individual $B(i)$ and the initial reference hypercube Z are determined, the initial number of iterations is set to 1, and IEP is the empty set (Line 1-3). Then, evolutionary operations are performed for each individual in the population in turn, and the newly produced offspring $O(i)$ is compared with Z to update Z (Line 6-7). Subsequently, the interval scalar function values of the offspring $O(i)$ and the neighborhoods of the current individual $B(i)$ are calculated, and the interval confidence level is used to determine the magnitude between them to update the neighborhoods of the current individual (Line 8-12). After each iteration, the neighborhood size T is adaptively updated (Line14-15). After multiple iterations, if the number of iterations G meets the pre-set ratio of iterations to evolve $rate_{evol}$, the IEP is updated: the interval non-dominated solutions in the current population and offspring are integrated into the IEP. If the size of the IEP exceeds the predetermined size, the poorly ranked individuals

are removed from the IEP using interval sparsity ranking and uncertainty ranking, so that the $|IEP|$ is equal to the predetermined size $nIEP$. Then, merge the updated IEP with the current population, select the most suitable individual for each weight vector as the new current population (Line 17). When the current number of iterations meets the adjustment period wag (Line 18), execute the adaptive WVs adjustment strategy to adjust the WVs and update their corresponding neighborhoods (Line 19-20). Finally, when the G_{max} is reached, the optimal population is output (Line 24).

3.4. Computational complexity

In the proposed IMOEA/D-AWN. The complexity of evaluating interval individuals using the interval Chebyshev scalar function is $O(MT)$, where M is the number of objectives and T is the neighborhood size; the worst case computational complexity of interval non-dominated sorting when update the interval elite populations is $O(MN^2)$, where N is the size of the population; the complexity of both removing and adding the weight vectors are $O(nusMN^2)$, where nus is the number of weight vectors to be adjusted; the complexity of updating the neighborhood is $O(TMN^2)$. To sum up the belongings, the maximum complexity of IMOEA/D-AWN is $O((nus+T)MN^2)$.

4. Experiments

In this section, the performance of the proposed IMOEA/D-AWN is validated. Four advanced IMOEAs were selected for comparison, namely, II-MOEA [17] DI-μMOGA [33], InMaOEA [37] and IMOEA/D [40]. All algorithms were evaluated on 17 interval benchmark test problems.

4.1. Test problems

WFG1-9 [46], DTLZ1-4 [47], MaF1-4 [48] test problems are used to evaluate the performance of IMOEA/D-AWN. The number of objectives in the above problems is 3, and the number of variables in WFG1-9, DTLZ2-4 and MaF1-4 is 12, that in DTLZ1 is 7.

Since the objective value of the test problem is deterministic, while the objective values of IMOPs are interval values, it is necessary to add an imprecision factor δ [49] to expand the deterministic values into interval values. For the tri-objective problem [40]:

$$\delta = \begin{bmatrix} sin(10\pi \sum_i x_i)/4 \\ sin(20\pi \sum_i x_i)/4 \\ sin(40\pi \sum_i x_i)/4 \end{bmatrix} \tag{13}$$

where x_i is the i-th variables, $i = 1, 2, ..., d$.

The original optimization problem is represented as $f_j(x), j = 1, 2, ..., M$. And the perturbed interval-valued function is denoted as $f_j(x, \delta)$:

$$\begin{cases} \underline{f_j}(\mathrm{x}, \delta_j) = \min\left\{f_j(\mathrm{x}), f_j(\mathrm{x}) + \delta_j\right\} \\ \overline{f_j}(\mathrm{x}, \delta_j) = \max\left\{f_j(\mathrm{x}), f_j(\mathrm{x}) + \delta_j\right\} \end{cases} \tag{14}$$

Corresponding interval benchmark test problems are denoted as IWFG1-9, IDTLZ1-4, and IMaF1-4.

4.2. Performance metrics

The following three performance metrics are used to evaluate the convergence, diversity, and uncertainty performance of each algorithm.

(1) IGD-metric
Pareto-optimal Front of each deterministic benchmark test problem as PF_{True} of each uncertainty problem. The IGD metric is computed as the average Euclidean distance between PF_{True} and the midpoint of the interval objective value of the approximate Pareto optimal solution set [17]:

$$\mathrm{IGD} = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{|PF_{True}|} \tag{15}$$

where d_i^2 denotes the minimum Euclidean distance from solution x_i on PF_{True} to the solution in the approximate Pareto optimal solution set, and $|PF_{True}|$ denotes the number of solutions in PF_{True}. This metric can reflect the two performances of convergence and diversity of the population. The smaller the value, the better.

(2) HV-metric
The hypervolume of the approximate Pareto solution set X of IMOP (Eq. (1)) is calculated as [49]:

$$H(X) = [\underline{H(X)}, \overline{H(X)}] = \Lambda(\bigcup_{x \in X} y \in R^n \mid f(x,c) \prec_{IN} y \prec_{IN} f(x_{ref}, c)) \tag{16}$$

where Λ means the Lebesgue measure; $\prec_{IN}$ is the interval Pareto dominance relation; and x_{ref} refers to the reference point. $\underline{H(X)}$ is the worst-case hypervolume and $\overline{H(X)}$ is the best-case hypervolume. The midpoint of $H(x)$ is referred to as the hypervolume [40].

(3) Imprecision-metric
The performance of the population in uncertainty is reflected by Imprecision, which is the sum of the interval lengths of all individuals in the Pareto optimal solution set for all objectives [49]. The smaller the Imprecision indicator, the better.

4.3. Comparison algorithms

(1) II-MOEA [17]: II-MOEA is a classic IMOEA based on Pareto dominance. Interval Pareto dominance relation based interval confidence level sorts individuals into different interval Pareto fronts. Then calculate the interval crowding distance according to the location and volume of the individual in the objective space. And select the optimal solution set based on the above two strategies.

(2) DI-μMOGA [33]: DI-μMOGA employs the interval dominance relation in [17] to compare the individuals from the evolutionary population and the external elite population. Modified the interval crowding distances to take into account the maximum and minimum interval values for the entire population at each objective when calculating the crowding distances of individuals.

(3) InMaOEA [37]: InMaOEA is a variant of II-MOEA. Modified the calculation of the interval confidence level and defined the interval Pareto dominance relation based on the modified interval confidence level. And designed an interval crowding distance strategy based on hypervolume and overlap degree.

(4) IMOEA/D [40]: An adaptive reference vector-based interval multi-objective evolutionary algorithm in the framework of MOEA/D. Decompose IMOP into multiple subproblems with interval parameters, and developed an ensemble scheme for evaluating interval individuals. Then, adaptively adjusted reference vectors based on the interval crowding distance.

4.4. Parameter setting

To make sure the experiment is fair, the population size N was set to 105, and the maximum number of iterations was set to 900 for all algorithms. The neighborhood size T was set to 10 for our algorithm and IMOEA/D based on the original paper [40], and the number of WVs is equal to N. Referring to [43], ratio of iterations to evolve $rate_{evol}$ is set to 0.8, and the adjustment period wag is set to 100, the ratio of updated weight vectors is set to 0.05, and $nus = 0.05 * N$. The crossover and mutation probabilities of II-MOEA, InMaOEA, and IMOEA/D are set to 0.9 and 0.1, respectively, according to the original papers, and the crossover and mutation probabilities of DI-μMOGA are set to 0.8 and 0.05, respectively, according to the original paper [33]. The generation for starting adjusting reference vectors of IMOEA/D is set to half of the maximum generation according to the original paper [40].

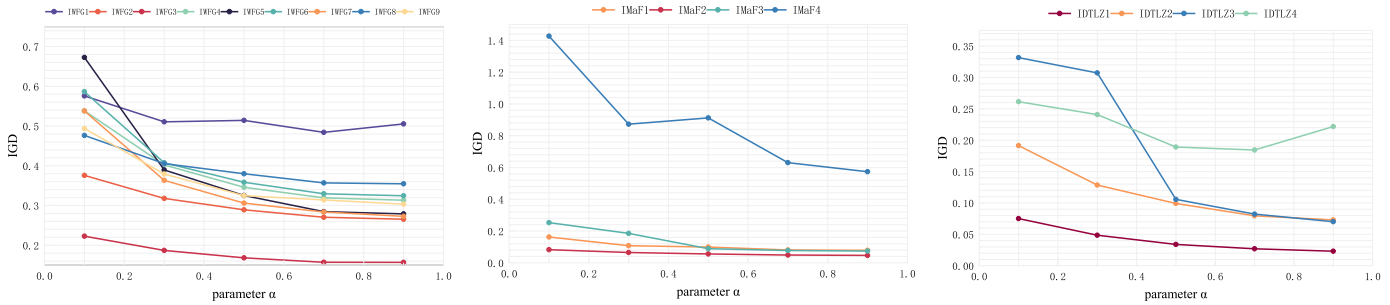

Fig. 3. Influence of parameter α to IGD metric. (a) IWFG. (b) IMaF. (c) IDTLZ.

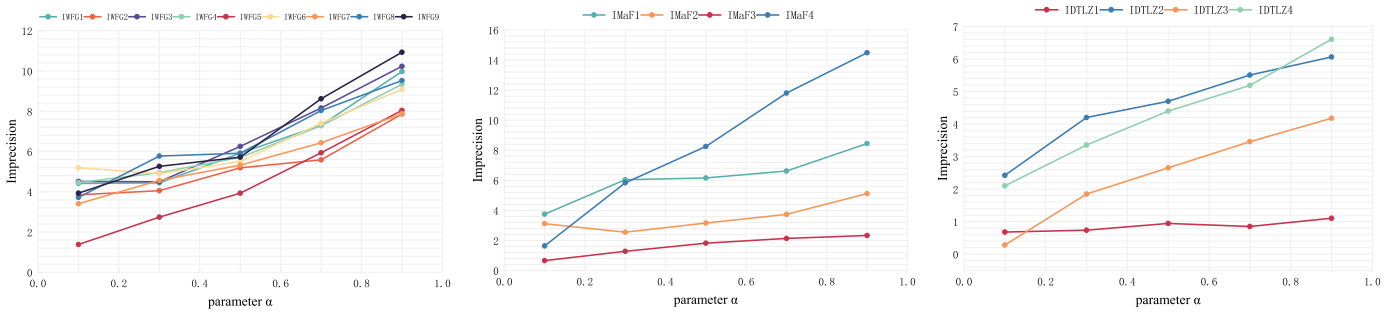

Fig. 4. Influence of Parameter α to Imprecision metric. (a) IWFG. (b) IMaF. (c) IDTLZ.

4.5. Experimental results and analysis

A. Effects of different values of α

This subsection will evaluate the parameters α in the comprehensive ranking. The α was set to 0.1-0.9, and to avoid experimental accidents, each group was run 30 times independently. The values of IGD and Imprecision obtained by IMOEA/D-AWN were recorded, and the average values were calculated. Fig. 3 and 4 show the IGD and Imprecision values obtained on each test problem when α is taken to different values, respectively.

As mentioned in Section 3, α is the weight of interval sparsity level ranking, and $(1-\alpha)$ is the weight of interval uncertainty ranking. The larger α is, the greater the impact of interval sparsity level on an individual's comprehensive ranking, and vice versa, the greater the impact of interval uncertainty. From Fig. 3, we can see that the value of IGD keeps decreasing as α gradually increases, i.e., the convergence and diversity of the algorithm gradually become better, indicating that a large value of α can improve these two performances of the algorithm. While from Fig. 4, we can see that the value of Imprecision keeps increasing as the weight of individual uncertainty ranking $(1-\alpha)$ gradually decreases, i.e., the uncertainty performance of the algorithm gradually becomes worse, indicating that a small value of α can reduce the uncertainty of the population. Since the introduced uncertain information has a negative impact on the distribution of the population, to minimize the impact of uncertainty on the distribution of the population and to reduce the uncertainty of the population at the same time, α is set to 0.8 here, i.e., the weight of interval sparsity is set to 0.8 and the weight of interval uncertainty is set to 0.2.

B. Effectiveness of Adaptive Adjustment of Weight Vectors

We designed interval uncertainty ranking R_I in the weight vectors adaptive adjustment strategy to enhance the uncertainty performance of the population. And we established an IEP to guide the update of the current population and thus guide the adjust of the WVs to enhance the convergence and diversity of the population. The following experiments were conducted to verify the effectiveness of the above strategies:

Table 1
The Imprecision index of two cases.

Problem	IMOEA/D-AWN (no R_I)	IMOEA/D-AWN
IWFG1	1.1345e+1 (3.87e+0) -	**8.3446e+0 (3.29e+0)**
IWFG2	9.3305e+0 (2.40e+0) -	**7.2199e+0 (1.31e+0)**
IWFG3	1.1393e+1 (1.83e+0) -	**8.8723e+0 (1.38e+0)**
IWFG4	9.8941e+0 (1.66e+0) -	**8.2720e+0 (1.19e+0)**
IWFG5	8.1637e+0 (1.63e+0) -	**6.9317e+0 (1.19e+0)**
IWFG6	9.4213e+0 (1.78e+0) -	**8.0609e+0 (1.59e+0)**
IWFG7	8.6821e+0 (1.03e+0) -	**7.7405e+0 (1.10e+0)**
IWFG8	1.0830e+1 (2.34e+0) -	**8.9499e+0 (1.43e+0)**
IWFG9	1.1610e+1 (1.81e+0) -	**1.0105e+1 (1.71e+0)**
IMaF1	9.0347e+0 (2.04e+0) -	**7.2441e+0 (1.02e+0)**
IMaF2	6.5240e+0 (2.13e+0) -	**4.3213e+0 (9.44e-1)**
IMaF3	3.3769e+0 (2.52e+0) -	**1.8819e+0 (8.50e-1)**
IMaF4	1.6106e+1 (4.62e+0) -	**1.3612e+1 (3.24e+0)**
IDTLZ1	2.0590e+0 (2.01e+0) -	**1.0365e+0 (9.32e-1)**
IDTLZ2	6.5894e+0 (1.38e+0) -	**5.7611e+0 (1.05e+0)**
IDTLZ3	5.1836e+0 (2.58e+0) -	**3.9413e+0 (2.17e+0)**
IDTLZ4	6.4361e+0 (3.13e+0) =	**5.8363e+0 (1.77e+0)**
+/-/=	0/16/1	

(1) Effectiveness of ranking of interval uncertainty R_I

To verify whether the introduction of R_I can reduce the uncertainty of the optimal solution, the following experiments were conducted. Table 1 shows the experimental results of IMOEA/D-AWN on Imprecision metrics in two cases: one case is the algorithm without the R_I, and the other case is the algorithm with the R_I. To avoid the randomness of the experimental results, each algorithm runs 30 times independently, the mean and standard deviation of Imprecision are listed in Table 1, where the optimal value is shown in bold. Wilcoxon signed rank test with 5% level of significance is performed, the symbols '+', '-', and '=' for the Imprecision metric indicate that IMOEA/D-AWN without R_I performs

Table 2
The IGD index of two cases.

Problem	IMOEA/D-AWN (no IEP)	IMOEA/D-AWN
IWFG1	5.8356e-1 (1.06e-1) -	**4.7870e-1 (1.21e-1)**
IWFG2	3.1761e-1 (1.95e-2) -	**2.7435e-1 (2.25e-2)**
IWFG3	2.3259e-1 (2.02e-2) -	**1.5247e-1 (1.83e-2)**
IWFG4	3.9113e-1 (2.81e-2) -	**3.1409e-1 (1.53e-2)**
IWFG5	3.2243e-1 (1.60e-2) -	**2.7962e-1 (7.86e-3)**
IWFG6	4.1622e-1 (2.94e-2) -	**3.2566e-1 (3.05e-2)**
IWFG7	3.3569e-1 (1.37e-2) -	**2.7401e-1 (9.23e-3)**
IWFG8	4.4480e-1 (2.55e-2) -	**3.5189e-1 (7.82e-3)**
IWFG9	3.3793e-1 (5.01e-2) -	**3.0266e-1 (4.84e-2)**
IMaF1	1.2399e-1 (1.07e-2) -	**7.8521e-2 (2.94e-3)**
IMaF2	7.7125e-2 (7.35e-3) -	**4.7151e-2 (1.95e-3)**
IMaF3	9.1558e-2 (1.32e-2) -	**7.4900e-2 (7.00e-3)**
IMaF4	8.0051e-1 (6.69e-1) -	**6.4061e-1 (3.28e-1)**
IDTLZ1	4.2921e-2 (6.97e-3) -	**2.4338e-2 (1.01e-3)**
IDTLZ2	1.1472e-1 (7.67e-3) -	**7.4193e-2 (2.15e-3)**
IDTLZ3	2.1602e-1 (3.93e-1) -	**7.3628e-2 (9.30e-3)**
IDTLZ4	1.9641e-1 (1.26e-1) -	**1.7981e-1 (1.81e-1)**
+/-/=	0/17/0	

Table 4
The IGD index of two cases.

Problem	IMOEA/D-AWN (no AW)	IMOEA/D-AWN
IWFG1	5.5219e-1 (1.33e-1) -	**4.7870e-1 (1.21e-1)**
IWFG2	3.1073e-1 (5.12e-2) -	**2.7435e-1 (2.25e-2)**
IWFG3	1.8600e-1 (1.05e-2) -	**1.5247e-1 (1.83e-2)**
IWFG4	3.2941e-1 (1.25e-2) -	**3.1409e-1 (1.53e-2)**
IWFG5	2.9904e-1 (4.93e-3) -	**2.7962e-1 (7.86e-3)**
IWFG6	3.3062e-1 (2.16e-2) =	**3.2566e-1 (3.05e-2)**
IWFG7	2.9302e-1 (6.64e-3) -	**2.7401e-1 (9.23e-3)**
IWFG8	3.6962e-1 (6.89e-3) -	**3.5189e-1 (7.82e-3)**
IWFG9	3.2384e-1 (5.23e-2) -	**3.0266e-1 (4.84e-2)**
IMaF1	8.0884e-2 (1.88e-3) -	**7.8521e-2 (2.94e-3)**
IMaF2	4.8827e-2 (1.64e-3) -	**4.7151e-2 (1.95e-3)**
IMaF3	8.0131e-2 (8.33e-3) -	**7.4900e-2 (7.00e-3)**
IMaF4	1.1433e+0 (2.52e+0) -	**6.4061e-1 (3.28e-1)**
IDTLZ1	2.5398e-2 (1.21e-3) -	**2.4338e-2 (1.01e-3)**
IDTLZ2	7.5944e-2 (2.74e-3) -	**7.4193e-2 (2.15e-3)**
IDTLZ3	8.3190e-2 (1.82e-2) -	**7.3628e-2 (9.30e-3)**
IDTLZ4	**1.3604e-1 (1.35e-1)** =	1.7981e-1 (1.81e-1)
+/-/=	0/15/2	

Table 3
The HV index of two cases.

Problem	IMOEA/D-AWN (no IEP)	IMOEA/D-AWN
IWFG1	6.8935e-1 (6.17e-2) =	**7.2464e-1 (6.86e-2)**
IWFG2	8.7749e-1 (9.91e-3) -	**8.9427e-1 (6.88e-3)**
IWFG3	3.2237e-1 (8.41e-3) -	**3.5880e-1 (8.40e-3)**
IWFG4	4.6218e-1 (7.53e-3) -	**4.9559e-1 (6.09e-3)**
IWFG5	4.6081e-1 (5.17e-3) -	**4.7372e-1 (6.91e-3)**
IWFG6	4.1884e-1 (2.13e-2) -	**4.5624e-1 (2.93e-2)**
IWFG7	4.8781e-1 (8.05e-3) -	**5.1167e-1 (7.11e-3)**
IWFG8	3.8972e-1 (1.04e-2) -	**4.2595e-1 (6.54e-3)**
IWFG9	4.5222e-1 (4.21e-2) -	**4.5912e-1 (4.49e-2)**
IMaF1	1.3001e-1 (9.67e-3) -	**1.7659e-1 (3.34e-3)**
IMaF2	1.9957e-1 (5.16e-3) -	**2.2139e-1 (3.12e-3)**
IMaF3	9.0789e-1 (1.06e-2) -	**9.2186e-1 (9.10e-3)**
IMaF4	4.1027e-1 (1.12e-1) -	**4.5387e-1 (8.59e-2)**
IDTLZ1	7.5894e-1 (1.79e-2) -	**7.9326e-1 (1.12e-2)**
IDTLZ2	4.6556e-1 (1.30e-2) -	**5.1666e-1 (4.35e-3)**
IDTLZ3	4.4716e-1 (1.26e-1) -	**5.2285e-1 (1.74e-2)**
IDTLZ4	4.4627e-1 (5.69e-2) -	**4.8844e-1 (8.14e-2)**
+/-/=	0/16/1	

Table 5
The HV index of two cases.

Problem	IMOEA/D-AWN (no AW)	IMOEA/D-AWN
IWFG1	6.7468e-1 (7.17e-2) -	**7.2464e-1 (6.86e-2)**
IWFG2	8.7139e-1 (2.17e-2) -	**8.9427e-1 (6.88e-3)**
IWFG3	3.4474e-1 (5.29e-3) -	**3.5880e-1 (8.40e-3)**
IWFG4	4.8535e-1 (7.89e-3) -	**4.9559e-1 (6.09e-3)**
IWFG5	4.6406e-1 (4.10e-3) -	**4.7372e-1 (6.91e-3)**
IWFG6	**4.5677e-1 (2.28e-2)** =	4.5624e-1 (2.93e-2)
IWFG7	5.0093e-1 (8.57e-3) -	**5.1167e-1 (7.11e-3)**
IWFG8	4.1040e-1 (4.75e-3) -	**4.2595e-1 (6.54e-3)**
IWFG9	4.4445e-1 (4.85e-2) -	**4.5912e-1 (4.49e-2)**
IMaF1	1.7336e-1 (2.82e-3) -	**1.7659e-1 (3.34e-3)**
IMaF2	**2.2149e-1 (2.40e-3)** =	2.2139e-1 (3.12e-3)
IMaF3	9.0554e-1 (8.72e-3) -	**9.2186e-1 (9.10e-3)**
IMaF4	4.3531e-1 (8.43e-2) -	**4.5387e-1 (8.59e-2)**
IDTLZ1	7.8440e-1 (1.17e-2) -	**7.9326e-1 (1.12e-2)**
IDTLZ2	5.1076e-1 (5.48e-3) -	**5.1666e-1 (4.35e-3)**
IDTLZ3	4.9894e-1 (3.82e-2) -	**5.2285e-1 (1.74e-2)**
IDTLZ4	**5.0223e-1 (5.89e-2)** +	4.8844e-1 (8.14e-2)
+/-/=	1/14/2	

statistically better, worse, and comparably than IMOEA/D-AWN. From this table, it can be seen that IMOEA/D-AWN achieved optimal values in all testing problems, and significantly outperformed IMOEA/D-AWN without R_I in 16 test instances. For IDTLZ4, the values of Imprecision for the two cases are not statistically significantly different, but IMOEA/D-AWN is still slightly better than IMOEA/D-AWN without R_I. Therefore, designing the uncertainty ranking R_I in the weight vectors adaptive adjustment strategy can effectively reduce the uncertainty of the optimal solution.

(2) Effectiveness of interval elite population IEP

This part will verify whether the strategy of IEP-guided weight vector adjustment can improve the convergence and diversity of the optimal solutions. Tables 2 and 3 show the experimental results of IMOEA/D-AWN on IGD and HV metrics in two cases, respectively. From Table 2, we can see that, for the IGD metric, IMOEA/D-AWN is significantly better than IMOEA/D-AWN without IEP on all 17 test instances. From Table 3, we can see that IMOEA/D-AWN is significantly superior to IMOEA/D-AWN without IEP on 16 test cases. In the IWFG1, there is no statistically significant difference between the HV values of the two cases. Nevertheless, the HV of IMOEA/D-AWN is slightly better than that of IMOEA/D-AWN without IEP in this problem. Therefore, using interval elite populations to guide the adjustment of the weight vectors can effectively improve the convergence and diversity of the optimal solutions.

(3) Effectiveness of adaptive adjustment weight vectors to solve IMOPs

In order to verify the effectiveness of the adaptive adjustment weight vector strategy for solving IMOPs, the following comparison experiments are conducted. The algorithm that does not adjust the weight vectors is denoted as IMOEA/D-AWN (no AW). Table 4 shows the IGD of IMOEA/D-AWN in two cases: one is the algorithm without AW and the other is the algorithm using AW. From this table, IMOEA/D-AWN is better than IMOEA/D-AWN without adaptive weight vector adjustment on 16 test instances. In only one problem IDTLZ4, the IGD of IMOEA/D-AWN is slightly inferior to the algorithm without weight vector adjustment, but they are not significantly different at the statistical level. Table 5 shows the HV values of two cases, IMOEA/D-AWN achieved optimal values on 14 test instances, in the IWFG6 and IMaF2 problems, IMOEA/D-AWN did not achieve an optimal value, but there was no statistically significant difference between the two cases again. On the IDTLZ4 problem, the HV of IMOEA/D-AWN is inferior to that of the algorithm without weight vector adjustment. Therefore, the weight vector adjustment strategy proposed in this paper improves the convergence and diversity of the IMOEA which deals with IMOPs. Table 6 shows the Imprecision values of two cases, IMOEA/D-AWN achieved optimal values in 10 test problems, and IMOEA/D-AWN performed slightly worse than the algorithm without adaptive weight vectors adjustment in the five problems: IWFG3, IWFG4, IWFG6, IWFG7, IWFG9, but there was no significant difference at the statistical level. For the

Table 6
The Imprecision of two cases.

Problem	IMOEA/D-AWN (no AW)	IMOEA/D-AWN
IWFG1	9.5950e+0 (2.82e+0) =	**8.3446e+0 (3.29e+0)**
IWFG2	8.8327e+0 (1.91e+0) -	**7.2199e+0 (1.31e+0)**
IWFG3	**8.6357e+0 (1.61e+0)** =	8.8723e+0 (1.38e+0)
IWFG4	**8.1088e+0 (1.77e+0)** =	8.2720e+0 (1.19e+0)
IWFG5	7.8585e+0 (2.06e+0) -	**6.9317e+0 (1.19e+0)**
IWFG6	**7.9899e+0 (1.73e+0)** =	8.0609e+0 (1.59e+0)
IWFG7	**7.7042e+0 (1.59e+0)** =	7.7405e+0 (1.10e+0)
IWFG8	**6.0812e+0 (9.46e-1)** +	8.9499e+0 (1.43e+0)
IWFG9	**9.2380e+0 (1.83e+0)** =	1.0105e+1 (1.71e+0)
IMaF1	**6.3249e+0 (1.62e+0)** +	7.2441e+0 (1.02e+0)
IMaF2	6.1774e+0 (1.26e+0) -	**4.3213e+0 (9.44e-1)**
IMaF3	2.9838e+0 (1.61e+0) -	**1.8819e+0 (8.50e-1)**
IMaF4	1.3879e+1 (4.40e+0) =	**1.3612e+1 (3.24e+0)**
IDTLZ1	1.8112e+0 (1.62e+0) -	**1.0365e+0 (9.32e-1)**
IDTLZ2	7.6957e+0 (1.58e+0) -	**5.7611e+0 (1.05e+0)**
IDTLZ3	4.8536e+0 (2.62e+0) =	**3.9413e+0 (2.17e+0)**
IDTLZ4	6.2855e+0 (2.16e+0) =	**5.8363e+0 (1.77e+0)**
+/-/=	2/6/9	

Table 7
The runtime of two cases.

Problem	IMOEA/D-AWN (no AN)	IMOEA/D-AWN
IWFG1	6.8124e+2 (2.06e+1) -	**5.4949e+2 (2.29e+1)**
IWFG2	6.7570e+2 (2.03e+1) -	**5.5211e+2 (2.04e+1)**
IWFG3	6.8237e+2 (2.03e+1) -	**5.5780e+2 (2.04e+1)**
IWFG4	6.5972e+2 (1.90e+1) -	**5.4120e+2 (2.22e+1)**
IWFG5	6.6356e+2 (2.08e+1) -	**5.4640e+2 (2.16e+1)**
IWFG6	6.5537e+2 (1.96e+1) -	**5.4475e+2 (1.97e+1)**
IWFG7	7.1945e+2 (2.14e+1) -	**5.9209e+2 (2.34e+1)**
IWFG8	7.0826e+2 (1.93e+1) -	**5.8018e+2 (1.96e+1)**
IWFG9	7.0904e+2 (1.94e+1) -	**5.8637e+2 (2.38e+1)**
IMaF1	5.9028e+2 (1.74e+1) -	**5.4291e+2 (7.48e+0)**
IMaF2	5.9831e+2 (1.75e+1) -	**5.5703e+2 (9.86e+0)**
IMaF3	6.2337e+2 (1.94e+1) -	**5.7758e+2 (9.36e+0)**
IMaF4	6.8284e+2 (3.20e+1) -	**6.6047e+2 (2.61e+1)**
IDTLZ1	6.6316e+2 (3.20e+1) -	**5.4367e+2 (2.01e+1)**
IDTLZ2	6.4399e+2 (2.93e+1) -	**5.0020e+2 (2.43e+1)**
IDTLZ3	6.5783e+2 (2.79e+1) -	**4.9045e+2 (2.63e+1)**
IDTLZ4	6.6541e+2 (3.13e+1) -	**4.7960e+2 (2.10e+1)**
+/-/=	0/17/0	

IWFG8 and IMaF1 problems, IMOEA/D-AWN performs worse than the algorithm without weight vector adjustment, which may be due to the α is uniformly set to 0.2 during the weight vector adjustment, which may be improved by adjusting the value of the α for the different problems, but as a whole, the population imprecision of IMOEA/D-AWN is better than that of the algorithm without weight vector adjustment.

C. Effectiveness of Adaptive Adjustment of Neighborhoods to solve IMOPs

This section will verify whether the adaptive adjustment neighborhood strategy (AN) can accelerate convergence and improve computational efficiency. Table 7 shows the running times of IMOEA/D-AWN in two cases: one is the algorithm without AN and the other is the algorithm using AN. From this table, we can see that the running time of IMOEA/D-AWN is significantly smaller than IMOEA/D-AWN without AN in all 17 test problems. This shows that the adaptive adjustment of the neighborhood strategy can significantly reduce the runtime of the algorithm.

Table 8 shows the number of optimal values (denoted as Rank 1) and suboptimal values (denoted as Rank 2) obtained by IMOEA/D-AWN on the IGD and Imprecision metrics in two cases. From Table 8, we can see that, for the IGD metric, IMOEA/D-AWN without AN achieves optimal values in 9 of the 17 test problems, while IMOEA/D-AWN achieves optimal values in 8 problems. For the Imprecision metric, IMOEA/D-AWN without AN achieves the optimal value in 8 test problems, and

Table 8
The number of Rank1-2 obtained by two cases.

Algorithms	IGD		Imprecision	
	Rank 1	Rank 2	Rank 1	Rank 2
IMOEA/D-AWN (no AN)	9	8	8	9
IMOEA/D-AWN	8	9	9	8

IMOEA/D-AWN achieves the optimal value in 9 problems. It shows that convergence, diversity, and uncertainty are similar in both cases. Based on the above experimental results, the adaptive adjustment neighborhood strategy can guarantee the convergence, diversity, and uncertainty of the population while shortening the running time of the algorithm. Therefore, AN can accelerate the convergence of the population and improve computational efficiency.

D. Experimental Results Compared with Other Algorithms

The performance of IMOEA/D-AWN is compared with II-MOEA [17], DI-μMOGA [33], InMaOEA [37] and IMOEA/D [40] in this subsection to verify its effectiveness. The mean and standard deviation of IGD, HV, and I indicators over 30 independent runs of IMOEA/D-AWN and compared algorithms II-MOEA, DI-μMOGA, InMaOEA, and IMOEA/D are reported in Tables 9, 10, and 11, respectively. The optimal values are shown in bold. Wilcoxon signed rank test is also performed.

Table 9 is the IGD values of different algorithms. From this table, it can be seen that IMOEA/D-AWN achieved optimal values on 13 test instances, DI-μMOGA and IMOEA/D both achieved optimal values on two test instances each, II-MOEA and InMaOEA did not achieve optimal values, and IMOEA/D-AWN clearly outperformed the other four algorithms. WFG1 has irregular PF, which features a mixed and sharp tail, and the mean IGD of IMOEA/D-AWN is slightly larger than that of DI-μMOGA and IMOEA/D in dealing with this problem, but not significantly different at the statistical level. On the IMaF3 problem, IMOEA/D-AWN performs worse than IMOEA/D on the IGD metric, but significantly better than the remaining three algorithms. MaF4 features a multi-modal and badly scaled PF, IMOEA/D-AWN handles this problem inferiorly to DI-μMOGA and IMOEA/D on IGD, but better than II-MOEA and InMaOEA. DTLZ4 has a concave and biased PF. In dealing with this problem, the IGD values of IMOEA/D-AWN are all larger than those of the other four algorithms, so this problem poses a challenge for IMOEA/D-AWN. However, IMOEA/D-AWN performs best for other problems with complex features, such as IWFG2-9, IMaF1-2, and IDTLZ1-3, where IWFG2 is a discontinuous PF, IWFG3, IMaF2 are irregular PFs, and these irregular PFs can pose challenges to the algorithm, but IMOEA/D-AWN can deal with these problems well. This means that IMOEA/D-AWN has better convergence and diversity for most test problems compared to other algorithms.

Table 10 is the HV index of different algorithms. IMOEA/D-AWN achieved the optimal value on 10 test instances, DI-μMOGA on two test instances, and IMOEA/D on five test instances. II-MOEA and InMaOEA did not achieve the optimal value. This implies that IMOEA/D-AWN has better comprehensive performance for most of the test problems compared to other algorithms. The HV values of IMOEA/D-AWN are inferior to DI-μMOGA for IWFG1, IMaF4 and IDTLZ1 problems, but IMOEA/D-AWN outperforms DI-μMOGA for the remaining 14 problems. IMOEA/D outperforms IMOEA/D-AWN on IWFG9, IMaF3-4, and IDTLZ1,4 problems, IMOEA/D performs similarly to IMOEA/D-AWN on IDTLZ3 problems, but IMOEA/D-AWN outperforms IMOEA/D on the other 12 problems. Thus, for the HV metric, IMOEA/D-AWN significantly outperforms IMOEA/D. In addition, IMOEA/D-AWN outperforms II-MOEA and InMaOEA for all problems.

Table 11 is the Imprecision index of different algorithms. From this table, we can see that IMOEA/D-AWN achieves optimal values for all the tested problems, especially for some irregular PF problems, e.g.,

Table 9
The IGD index of different algorithm.

Problem	II-MOEA	DI-μMOGA	InMaOEA	IMOEA/D	IMOEA/D-AWN
IWFG1	7.4767e-1 (1.42e-1) -	**4.2617e-1 (7.96e-2)** =	7.6747e-1 (2.48e-1) -	4.9455e-1 (8.47e-2) =	4.7870e-1 (1.21e-1)
IWFG2	4.4872e-1 (6.33e-2) -	2.7506e-1 (2.65e-2) =	4.5370e-1 (6.98e-2) -	2.8342e-1 (1.95e-2) =	**2.7435e-1 (2.25e-2)**
IWFG3	3.8578e-1 (7.45e-2) -	1.9632e-1 (2.15e-2) -	3.9216e-1 (7.81e-2) -	2.1551e-1 (2.15e-2) -	**1.5247e-1 (1.83e-2)**
IWFG4	5.3031e-1 (4.76e-2) -	3.9937e-1 (2.65e-2) -	5.1913e-1 (3.87e-2) -	3.3582e-1 (1.62e-2) -	**3.1409e-1 (1.53e-2)**
IWFG5	5.5990e-1 (7.19e-2) -	3.9196e-1 (2.13e-2) -	5.4413e-1 (5.30e-2) -	2.9379e-1 (9.81e-3) -	**2.7962e-1 (7.86e-3)**
IWFG6	6.4948e-1 (6.02e-2) -	4.2514e-1 (2.01e-2) -	6.6551e-1 (6.71e-2) -	3.5878e-1 (4.55e-2) -	**3.2566e-1 (3.05e-2)**
IWFG7	6.8494e-1 (9.85e-2) -	3.9074e-1 (2.31e-2) -	6.8515e-1 (7.14e-2) -	2.9543e-1 (1.32e-2) -	**2.7401e-1 (9.23e-3)**
IWFG8	6.0557e-1 (4.87e-2) -	4.8032e-1 (2.44e-2) -	6.1033e-1 (3.71e-2) -	3.8845e-1 (1.94e-2) -	**3.5189e-1 (7.82e-3)**
IWFG9	5.7682e-1 (6.46e-2) -	3.8677e-1 (2.52e-2) -	6.0782e-1 (6.75e-2) -	3.0597e-1 (4.48e-2) -	**3.0266e-1 (4.84e-2)**
IMaF1	1.8798e-1 (2.39e-2) -	8.1954e-2 (5.64e-3) -	1.7792e-1 (2.36e-2) -	8.6796e-2 (4.16e-3) -	**7.8521e-2 (2.94e-3)**
IMaF2	1.1618e-1 (1.15e-2) -	5.9303e-2 (4.52e-3) -	1.1629e-1 (1.37e-2) -	5.3322e-2 (3.08e-3) -	**4.7151e-2 (1.95e-3)**
IMaF3	6.6284e+4 (1.44e+5) -	1.0757e+1 (3.93e+1) -	2.2777e+5 (3.42e+5) -	**5.4750e-2 (7.89e-3)** +	7.4900e-2 (7.00e-3)
IMaF4	7.7489e-1 (9.08e-2) -	**4.4405e-1 (4.01e-2)** +	7.6081e-1 (9.45e-2) -	4.4618e-1 (2.86e-2) +	6.4061e-1 (3.28e-1)
IDTLZ1	7.1478e-1 (1.84e+0) -	3.3394e-2 (2.89e-3) -	3.8731e+0 (1.57e+1) -	2.8322e-2 (1.65e-3) -	**2.4338e-2 (1.01e-3)**
IDTLZ2	1.5226e-1 (2.04e-2) -	1.0686e-1 (6.54e-3) -	1.4732e-1 (1.82e-2) -	8.6758e-2 (5.03e-3) -	**7.4193e-2 (2.15e-3)**
IDTLZ3	3.6480e+1 (7.92e+1) -	2.6447e-1 (3.30e-1) -	2.3904e+1 (5.98e+1) -	8.3759e-2 (8.25e-3) -	**7.3628e-2 (9.30e-3)**
IDTLZ4	1.5352e-1 (1.01e-1) +	1.7739e-1 (2.00e-1) +	1.3963e-1 (7.67e-2) +	**1.3584e-1 (1.04e-1)** +	1.7981e-1 (1.81e-1)
+/-/=	1/16/0	2/13/2	1/16/0	3/12/2	

Table 10
The HV index of different algorithm.

Problem	II-MOEA	DI-μMOGA	InMaOEA	IMOEA/D	IMOEA/D-AWN
IWFG1	7.1648e-1 (6.08e-2) =	**7.7990e-1 (5.57e-2)** +	7.0749e-1 (8.36e-2) =	7.4350e-1 (5.31e-2) =	7.2464e-1 (6.86e-2)
IWFG2	8.4091e-1 (1.55e-2) -	8.8920e-1 (8.80e-3) -	8.3976e-1 (2.26e-2) -	8.9172e-1 (5.57e-3) =	**8.9427e-1 (6.88e-3)**
IWFG3	2.5560e-1 (2.37e-2) -	3.2760e-1 (1.01e-2) -	2.5436e-1 (2.58e-2) -	3.1963e-1 (1.16e-2) -	**3.5880e-1 (8.40e-3)**
IWFG4	4.3226e-1 (1.45e-2) -	4.6556e-1 (1.15e-2) -	4.3477e-1 (9.74e-3) -	4.7646e-1 (5.71e-3) -	**4.9559e-1 (6.09e-3)**
IWFG5	4.0636e-1 (1.92e-2) -	4.4755e-1 (9.73e-3) -	4.1044e-1 (1.71e-2) -	4.6726e-1 (5.60e-3) -	**4.7372e-1 (6.91e-3)**
IWFG6	3.6145e-1 (2.67e-2) -	4.3674e-1 (1.37e-2) -	3.6048e-1 (1.97e-2) -	4.3662e-1 (3.53e-2) -	**4.5624e-1 (2.93e-2)**
IWFG7	3.9442e-1 (2.65e-2) -	4.8681e-1 (1.05e-2) -	3.9224e-1 (1.57e-2) -	4.9963e-1 (5.45e-3) -	**5.1167e-1 (7.11e-3)**
IWFG8	3.3922e-1 (1.86e-2) -	3.9384e-1 (7.41e-3) -	3.3651e-1 (1.62e-2) -	4.1780e-1 (5.35e-3) -	**4.2595e-1 (6.54e-3)**
IWFG9	4.0712e-1 (1.94e-2) -	4.5524e-1 (2.23e-2) -	4.0054e-1 (2.01e-2) -	**4.6029e-1 (3.61e-2)** +	4.5912e-1 (4.49e-2)
IMaF1	1.2241e-1 (9.35e-3) -	1.7635e-1 (4.49e-3) =	1.2576e-1 (1.08e-2) -	1.6972e-1 (4.39e-3) -	**1.7659e-1 (3.34e-3)**
IMaF2	1.8460e-1 (9.11e-3) -	2.1720e-1 (3.46e-3) -	1.8449e-1 (1.01e-2) -	2.1919e-1 (3.03e-3) -	**2.2139e-1 (3.12e-3)**
IMaF3	3.6286e-1 (4.28e-1) -	5.6409e-1 (3.94e-1) -	1.9482e-1 (3.61e-1) -	**9.4197e-1 (3.98e-3)** +	9.2186e-1 (9.10e-3)
IMaF4	3.0023e-1 (3.72e-2) -	4.5903e-1 (1.94e-2) +	3.0956e-1 (3.30e-2) -	**4.9617e-1 (5.21e-3)** +	4.5387e-1 (8.59e-2)
IDTLZ1	6.2124e-1 (3.02e-1) -	**8.1133e-1 (6.81e-3)** +	6.4687e-1 (2.67e-1) -	8.0467e-1 (5.59e-3) +	7.9326e-1 (1.12e-2)
IDTLZ2	4.4013e-1 (2.47e-2) -	4.9887e-1 (8.51e-3) -	4.4336e-1 (2.78e-2) -	5.0273e-1 (6.83e-3) -	**5.1666e-1 (4.35e-3)**
IDTLZ3	2.4566e-1 (2.13e-1) -	4.0391e-1 (1.65e-1) -	2.1789e-1 (2.12e-1) -	**5.2896e-1 (7.54e-3)** =	5.2285e-1 (1.74e-2)
IDTLZ4	4.6241e-1 (4.39e-2) -	4.6516e-1 (9.60e-2) -	4.7308e-1 (3.39e-2) -	**4.9536e-1 (4.63e-2)** +	4.8844e-1 (8.14e-2)
+/-/=	0/16/1	3/13/1	0/16/1	5/9/3	

Table 11
The Imprecision index of different algorithm.

Problem	II-MOEA	DI-μMOGA	InMaOEA	IMOEA/D	IMOEA/D-AWN
IWFG1	1.7363e+1 (7.15e+0) -	2.3145e+1 (4.73e+0) -	1.6610e+1 (7.12e+0) -	1.0327e+1 (2.92e+0) -	**8.3446e+0 (3.29e+0)**
IWFG2	1.4375e+1 (1.82e+0) -	2.2071e+1 (2.18e+0) -	1.4237e+1 (1.75e+0) -	1.1316e+1 (1.38e+0) -	**7.2199e+0 (1.31e+0)**
IWFG3	1.8041e+1 (1.97e+0) -	2.3920e+1 (2.21e+0) -	1.6609e+1 (1.57e+0) -	1.4115e+1 (1.38e+0) -	**8.8723e+0 (1.38e+0)**
IWFG4	1.7497e+1 (2.18e+0) -	3.7917e+1 (3.03e+0) -	1.8420e+1 (2.23e+0) -	1.4645e+1 (1.37e+0) -	**8.2720e+0 (1.19e+0)**
IWFG5	1.6697e+1 (2.37e+0) -	3.5319e+1 (3.13e+0) -	1.6215e+1 (2.25e+0) -	1.2235e+1 (1.72e+0) -	**6.9317e+0 (1.19e+0)**
IWFG6	1.3617e+1 (2.05e+0) -	3.4605e+1 (2.70e+0) -	1.3957e+1 (2.46e+0) -	1.4636e+1 (2.45e+0) -	**8.0609e+0 (1.59e+0)**
IWFG7	1.4143e+1 (1.90e+0) -	3.5831e+1 (2.47e+0) -	1.3130e+1 (1.66e+0) -	1.3152e+1 (1.50e+0) -	**7.7405e+0 (1.10e+0)**
IWFG8	1.8158e+1 (2.29e+0) -	3.6066e+1 (2.24e+0) -	1.8015e+1 (2.55e+0) -	1.6213e+1 (1.57e+0) -	**8.9499e+0 (1.43e+0)**
IWFG9	1.7755e+1 (2.14e+0) -	3.8952e+1 (2.53e+0) -	1.6677e+1 (1.97e+0) -	1.6210e+1 (1.74e+0) -	**1.0105e+1 (1.71e+0)**
IMaF1	8.5155e+0 (1.24e+0) -	1.4900e+1 (2.24e+0) -	8.2508e+0 (1.66e+0) -	8.8920e+0 (1.15e+0) -	**7.2441e+0 (1.02e+0)**
IMaF2	7.3327e+0 (1.48e+0) -	1.1703e+1 (2.28e+0) -	7.6945e+0 (2.00e+0) -	5.9906e+0 (8.33e-1) -	**4.3213e+0 (9.44e-1)**
IMaF3	2.1692e+1 (1.52e+1) -	2.0350e+1 (1.09e+1) -	2.7141e+1 (1.37e+1) -	1.9585e+0 (5.64e-1) =	**1.8819e+0 (8.50e-1)**
IMaF4	1.8894e+1 (3.44e+0) -	3.8248e+1 (2.46e+0) -	1.8837e+1 (2.43e+0) -	1.8422e+1 (2.51e+0) -	**1.3612e+1 (3.24e+0)**
IDTLZ1	5.0550e+0 (7.93e+0) -	2.4756e+0 (1.16e+0) -	4.8134e+0 (9.70e+0) -	1.1589e+0 (3.65e-1) -	**1.0365e+0 (9.32e-1)**
IDTLZ2	8.8609e+0 (1.68e+0) -	1.4403e+1 (2.28e+0) -	9.4135e+0 (1.88e+0) -	7.6679e+0 (1.28e+0) -	**5.7611e+0 (1.05e+0)**
IDTLZ3	1.5352e+1 (1.31e+1) -	1.6171e+1 (6.08e+0) -	1.6103e+1 (1.32e+1) -	4.0645e+0 (1.29e+0) =	**3.9413e+0 (2.17e+0)**
IDTLZ4	1.0853e+1 (5.33e+0) -	1.4428e+1 (4.36e+0) -	1.0256e+1 (3.75e+0) -	7.9889e+0 (1.63e+0) -	**5.8363e+0 (1.77e+0)**
+/-/=	0/17/0	0/17/0	0/17/0	0/15/2	

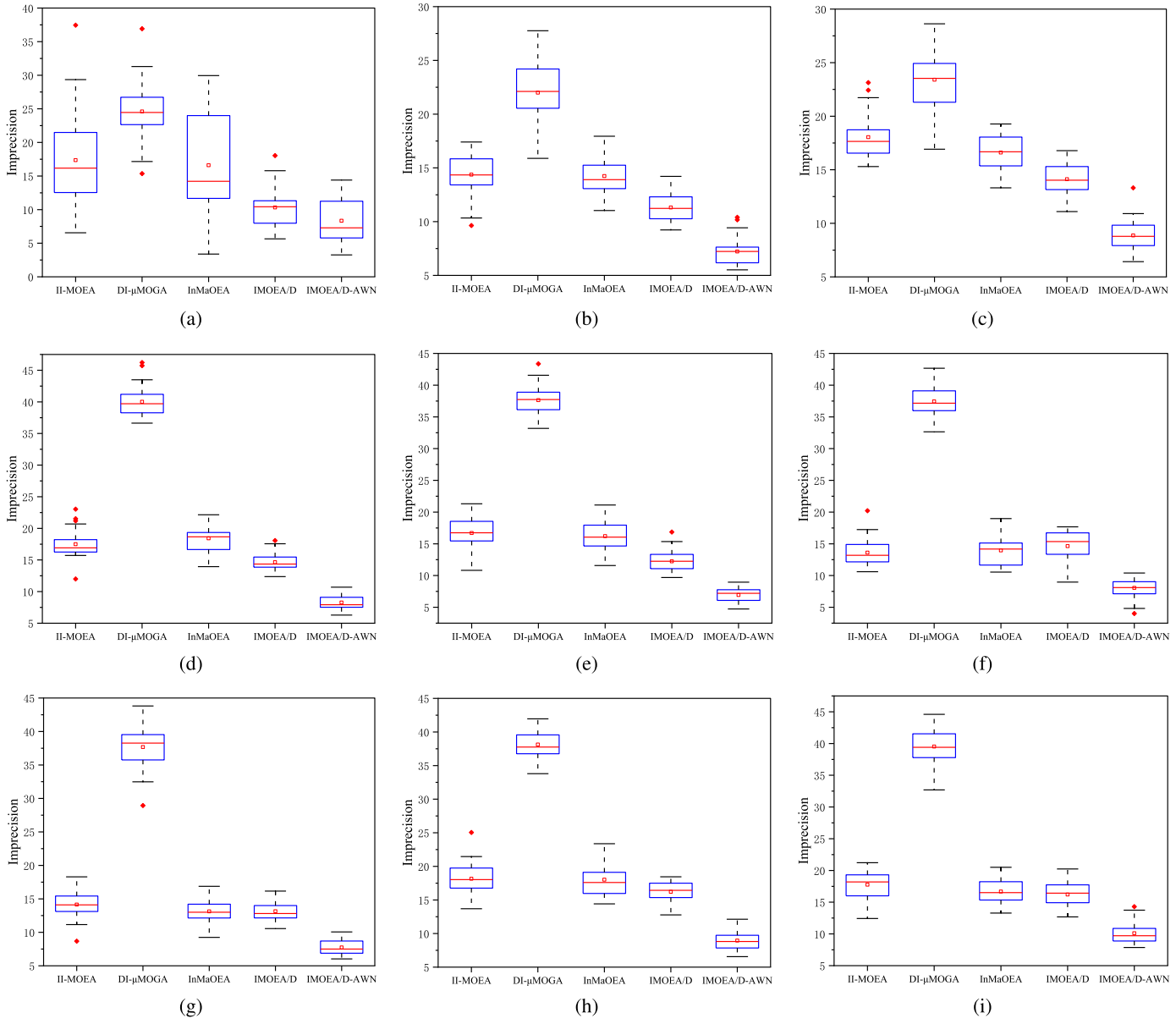

Fig. 5. The boxplot of Imprecision on WFG series benchmark optimization problems. (a) IWFG1. (b) IWFG2. (c) IWFG3. (d) IWFG4. (e) IWFG5. (f) IWFG6. (g) IWFG7. (h) IWFG8. (I) IWFG9.

WFG1 has an irregular PF features sharp tail and mixed, WFG2 has discontinuous PF [50], WFG3 has a degenerate PF, which is also irregular. MaF2 has concave irregular PF, MaF4 has badly-scaled and multi-modal PF [42]. This is due to the individual uncertainty that is taken into account when designing the weight vector adaptive adjustment strategy. Fig. 5 is the boxplot of Imprecision on IWFG series benchmark optimization problems. From Fig. 5, we can see that the Imprecision values of IMOEA/D-AWN are overall significantly smaller than the other four algorithms, and the Imprecision values of the solutions obtained by it are more aggregated, except for IWFG1. For IWFG1, although the Imprecision values obtained by IMOEA/D-AWN are not as aggregated as those of IMOEA/D, their average value is apparently smaller than its of IMOEA/D. Therefore, the uncertainty performance of IMOEA/D-AWN for the IWFG series problems is significantly superior to that of the other algorithms. Fig. 6 shows the boxplot of Imprecision on IMaF series benchmark optimization problems. From this figure, we can know that the Imprecision values of IMOEA/D-AWN are all overall significantly smaller than the other four algorithms, except for IMaF3. As can be seen in Table 11, for IMaF3, the Imprecision values obtained by IMOEA/D-AWN are not statistically significantly different from those obtained by IMOEA/D, but the average value of Imprecision achieved by IMOEA/D-AWN is still slightly better than that of IMOEA/D. Fig. 7 shows the boxplot of Imprecision on IDTLZ series benchmark optimization problems. From the figure, we can see that the Imprecision values of IMOEA/D-AWN are all overall significantly smaller than the other four algorithms except for IDTLZ3. From Table 11, we can see that for IDTLZ3, the imprecision values obtained by IMOEA/D-AWN and IMOEA/D are not statistically significantly different from each other, but the average value of Imprecision gained by IMOEA/D-AWN is still slightly better than that of IMOEA/D and significantly superior to the other three algorithms. These experimental results show the superior uncertainty performance of IMOEA/D-AWN.

E. Application to an uncertain collaborative computation offloading optimization problem

In order to further verify the performance of the proposed IMOEA/D-AWN, an uncertain collaborative computation offloading optimization problem is applied in this subsection. Uncertain collaborative compu-

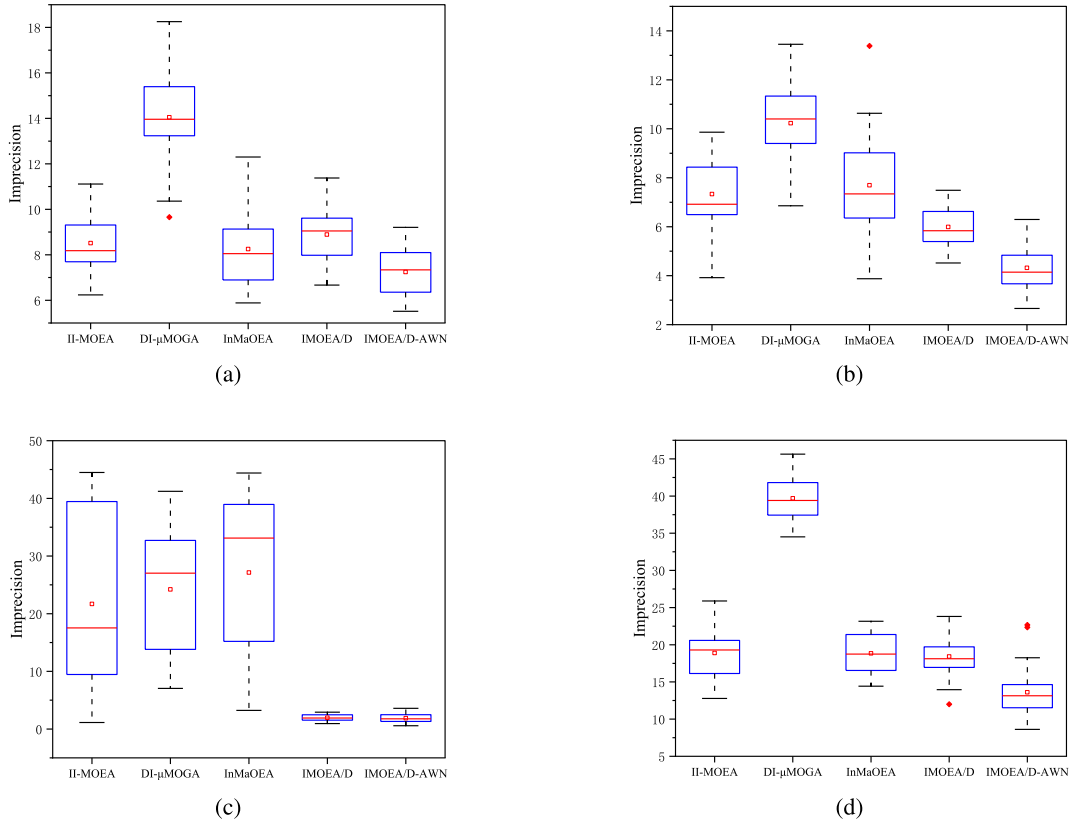

Fig. 6. The boxplot of Imprecision on MaF series benchmark optimization problems. (a) IMaF1. (b) IMaF2. (c) IMaF3. (d) IMaF4.

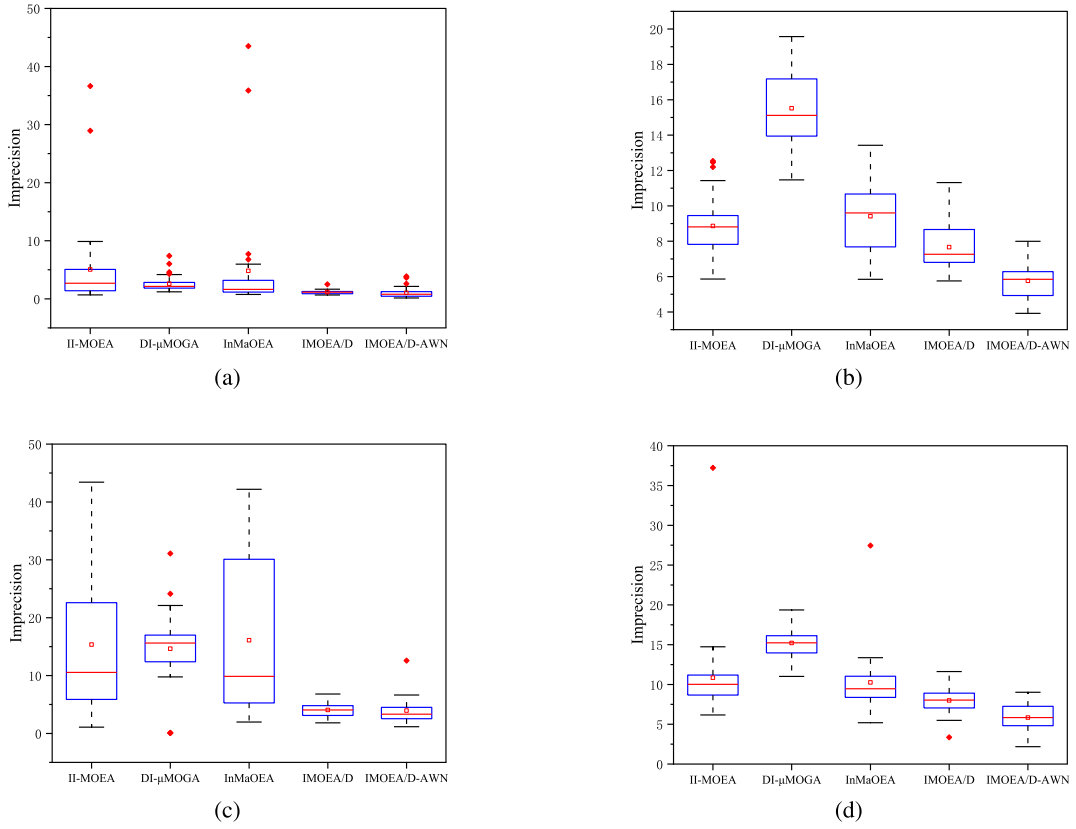

Fig. 7. The boxplot of Imprecision on DTLZ series benchmark optimization problems. (a) IDTLZ1. (b) IDTLZ2. (c) IDTLZ3. (d) IDTLZ4.

Table 12
Comparisons between different algorithms on a real application.

Metric	II-MOEA	DI-μMOGA	InMaOEA	IMOEA/D	IMOEA/D-AWN
HV	3.7288e-1 (9.87e-4) -	3.7251e-1 (9.78e-4) -	3.7304e-1 (1.10e-3) -	4.3579e-1 (3.24e-3) -	**4.9521e-1 (4.62e-2)**
Imprecision	1.0872e+5 (2.21e+4) -	1.2191e+5 (2.33e+4) -	1.1040e+5 (2.15e+4) -	3.6726e+5 (5.13e+4) -	**9.5304e+3 (5.85e+3)**

tation offloading is a two-layer edge computing framework, including the terminal layer and edge layer. The edge layer deploys edge servers to provide strong computing power support for the entire framework, while the terminal layer deploys intelligent terminal devices. Each intelligent terminal device generates tasks and has limited computing power, which can be used for data transmission and computing tasks. Collaborative computing offloading aims to fully use the remaining computing and storage capacity of various edge servers, achieve task load balancing, improve the computing response speed of terminal nodes, reduce the cost of the entire computing offloading system, and make more reasonable use of the computing resources of the entire system. Due to uncertain factors such as network bandwidth, which results in offloading delay, load balancing, and energy consumption being uncertain, so it is an IMOP.

(1) Modeling

An interval multi-objective model of collaborative computation offloading optimization problems is formulated as follows:

$$\min\{[\underline{T},\overline{T}],[\underline{L},\overline{L}],[\underline{C},\overline{C}]\} \tag{17}$$

where $T=[\underline{T},\overline{T}]$ represents the offloading delay, $L=[\underline{L},\overline{L}]$ represents the Load balancing, and $C=[\underline{C},\overline{C}]$ is the energy consumption.

Interval offloading delay function: The latency of task offloading is defined as $T=T_{Trans}+T_{EG}$, where T_{EG} is the execution time of the task, which includes the execution time of terminal equipment and edge servers. And the communication latency of task transfer is: $T_{trans}=(DS\cdot Dis)/R$, where DS is the size of the data for the task, and Dis is the distance between the terminal equipment and the unloading location. Due to uncertain factors such as bandwidth, the transmission rate R is uncertain, denoted as $R=[\underline{R},\overline{R}]$, therefore, T_{trans} is an interval. So the interval offloading delay function is $T=[\underline{T},\overline{T}]$, where

$$\underline{T}=\underline{T_{trans}}+T_{EG},\ \overline{T}=\overline{T_{trans}}+T_{EG} \tag{18}$$

Interval Load balancing function: The load balancing function is defined as $L=L_{TE}+L_{ES}$, where L_{TE} is the load balancing function of terminal equipment:

$$L_{TE}=\sqrt{\frac{\sum_{m=1}^{M}\left(TS_m-TL_m\right)^2}{M}} \tag{19}$$

where $TS_m=[\underline{TS_m},\overline{TS_m}]$ is the size of the task uploaded to the terminal equipment, $\overline{TL_m}$ is the load limit of the terminal equipment, and M is the number of terminal equipments.

L_{ES} is the load balancing function of edge servers,

$$L_{ES}=\sqrt{\frac{\sum_{i=1}^{I}\left(TS_i-TL_i\right)^2}{I}} \tag{20}$$

where $TS_i=[\underline{TS_i},\overline{TS_i}]$ is the size of the task uploaded to the edge server, $\overline{TL_i}$ is the load limit of the edge server, I is the number of edge servers.

Summarize, the interval load balancing function is $L=[\underline{L},\overline{L}]$, where

$$\underline{L}=\underline{L_{TE}}+\underline{L_{ES}},\ \overline{L}=\overline{L_{TE}}+\overline{L_{ES}} \tag{21}$$

Interval energy consumption function: The energy consumption function is defined as $C=C_{TE}+C_{ES}+C_{trans}$, where C_{TE}, C_{ES} and C_{trans} denote the energy consumption of terminal equipment, edge servers, and communication, respectively; in which $C_{trans}=T_{trans}\cdot c$, where c is the communication energy consumption, $T_{trans}=[\underline{T_{trans}},\overline{T_{trans}}]$ is the communication latency of task transfer. Therefore, the interval energy consumption function is $C=[\underline{C},\overline{C}]$, where

$$\underline{C}=C_{TE}+C_{ES}+\underline{C_{trans}},\ \overline{C}=C_{TE}+C_{ES}+\overline{C_{trans}} \tag{22}$$

(2) Results and analysis

Table 12 shows the values of HV and Imprecision that are obtained by II-MOEA, DI-μMOGA, InMaOEA, IMOEA/D, and IMOEA/D-AWN on the uncertain collaborative computation offloading optimization problem. It is clear from this table that the HV and Imprecision values of IMOEA/D-AWN are all the best. This implies that IMOEA/D-AWN outperforms the other four algorithms in terms of convergence, diversity, and uncertainty. In terms of the HV metric, IMOEA/D-AWN ranks first, and the other algorithms are ranked as follows: IMOEA/D, InMaOEA, II-MOEA, and DI-μMOGA. For the Imprecision metric, IMOEA/D-AWN also ranks first, and the other algorithms are ranked as: II-MOEA, InMaOEA, DI-μMOGA, and IMOEA/D. The ranking of some of the comparison algorithms has changed in these two metrics, while the ranking of IMOEA/D-AWN has not changed and ranks first. This implies that IMOEA/D-AWN can better balance convergence, diversity, and uncertainty. This is due to that not only interval sparsity but also imprecision is taken into account when adjusting the weight vectors, and the interval elite population is used to update the contemporary population, further improving the performance of the population. The above experimental results and analysis illustrate that IMOEA/D-AWN can effectively solve the uncertain collaborative computation offloading optimization problem.

5. Conclusions

Designing IMOEAs that can enhance population convergence and diversity while reducing population uncertainty is important for solving IMOPs. This paper proposes a decomposition-based IMOEA with adaptive adjustment of weight vectors and neighborhoods. Firstly, an interval sparsity function is designed, and interval sparsity ranking and interval uncertainty ranking are given based on interval sparsity and interval uncertainty, respectively. Based on the above comprehensive ranking, a new adaptive adjustment weight vector strategy guided by interval elite population is designed. Besides, an adaptive adjustment neighborhoods strategy is designed. This strategy focuses on global search in the early stage of evolution and gradually focuses on local search as evolution proceeds, so as to allocate computing resources reasonably and enhance evolutionary efficiency. The experimental results demonstrate that the IMOEA/D-AWN can effectively reduce the uncertainty of the population and enhance its convergence and diversity.

However, the changes of the parameter in the comprehensive ranking can have an influence on the performance of the proposed algorithm. Although we have selected its appropriate value through comparative experiments, it may not be optimal. Therefore, how to obtain its optimal value deserves further research. And the number of iterations to start adjusting the weight vectors is fixed, different number of generations to start adjusting will have an impact on the performance of the algorithm. Therefore, how to adaptively determine this value deserves further research. Besides, IMOEA/D-AWN can be applied to practical applications to provide more high-quality options for decision makers. However, IMOPs with interval constraints are not uncommon

in the real world. Therefore, it is our future research direction to design an efficient IMOEA to solve the problems mentioned above.

Declaration of competing interest

No conflict of interest exists in the submission of this manuscript, and the manuscript is approved by all authors for publication.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China [Grant NO. 61806138, NO. 61003053]; in part by the China University Industry-University-Research Collaborative Innovation Fund (Future Network Innovation Research and Application Project) [Grant 2021FNA04014]; in part by the Central Government Guides Local Science and Technology Development Funds (Grant NO. YDZJSX2021A038); in part by the Key R&D program of Shanxi Province (Grant No. 202202020101012); in part by the Taiyuan University of Science and Technology Scientific Research Initial Funding (TYUST SRIF), No. 20232087.

References

[1] Wang J, Wang L, Xiu X. A cooperative memetic algorithm for energy-aware distributed welding shop scheduling problem. Eng Appl Artif Intell 2023;120.
[2] Panigrahi GS, Patra AK, Nanda A, Kar SK. Adaptive controller design based on grasshopper optimisation technique for bg regulation in tidm patient. Int J Autom Control 2023;17.
[3] Wang F, Li Y, Li Y, Chen J. Bi-level programming model for post-disaster emergency supplies scheduling with time windows and its algorithm. Int J Autom Control 2022;16.
[4] Hu J, Wu H, Zhan R, Li Y, Menassel R. Self-adaptive wolf pack algorithm based on dynamic population updating for continuous optimisation problems. Int J Autom Control 2021;15.
[5] Yao B, Chen C, Shan W, Yu B. Artificial leaf-vein network optimisation algorithm for urban transportation network design. Int J Bio-Inspir Comput 2022;20(4):256–68.
[6] Aslan S, Arslan S. A modified artificial bee colony algorithm for classification optimisation. Int J Bio-Inspir Comput 2022;20(1).
[7] Dong T, Zhou L, Chen L, Song Y, Tang H, Qin H. A hybrid algorithm for workflow scheduling in cloud environment. Int J Bio-Inspir Comput 2023;21(1).
[8] Zhao X, Tian X, Li Z, Tan X, Zhang Q, Chen H, et al. Binary particle swarm optimisation and the extreme learning machine for diagnosing paraquat-poisoned patients. Int J Autom Control 2021;15(1).
[9] Yi L, Xu X, Tan M, Zhang Z, Xiao W, Fan L. Pid self-tuning method based on deep belief network and improved firefly algorithm. Int J Autom Control 2021;15(3).
[10] Zhang X, Zhang B, Meng L, Ren Y, Meng R, Li J. An evolutionary algorithm for a hybrid flowshop scheduling problem with consistent sublots. Int J Autom Control 2022;16.
[11] Cui Z, Xue Z, Fan T, Cai X, Zhang W. A many-objective evolutionary algorithm based on constraints for collaborative computation offloading. Swarm Evol Comput 2023;77.
[12] Cai X, Lan Y, Zhang Z, Wen J, Cui Z, Zhang W. A many-objective optimization based federal deep generation model for enhancing data processing capability in iot. IEEE Trans Ind Inform 2023;19(1):561–9.
[13] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: nsga-ii. IEEE Trans Evol Comput 2002;6(2):182–97.
[14] Zhang Q, Li H. Moea/d: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 2007;11(6):712–31.
[15] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Parallel problem solving from nature - PPSN VIII; 2004. p. 832–42.
[16] Trivedi A, Srinivasan D, Sanyal K, Ghosh A. A survey of multiobjective evolutionary algorithms based on decomposition. IEEE Trans Evol Comput 2017;21(3):440–62.
[17] Gong D, Qin N, Sun X. Evolutionary algorithms for multi-objective optimization problems with interval parameters; 2010. p. 411–20.
[18] Sun J, Miao Z, Gong D, Zeng X-J, Li J, Wang G. Interval multiobjective optimization with memetic algorithms. IEEE Trans Cybern 2020;50(8):3444–57.
[19] Zhang Y, Gong DW, Zhang QY, Jiang YQ. Deterministic interpretation for uncertain optimization problems with interval constraints. Syst Eng-Theory Pract 2009;29(2):127–33.
[20] Fu C, Liu Z, Deng J. A direct solution framework for structural optimization problems with interval uncertainties. Appl Math Model 2020;80:384–93.
[21] Liu X, Zhang Z, Yin L. A multi-objective optimization method for uncertain structures based on nonlinear interval number programming method. Mech Based Des Struct Mach 2017;45(1):25–42.
[22] Li Y, Wang P, Gooi HB, Ye J, Wu L. Multi-objective optimal dispatch of microgrid under uncertainties via interval optimization. IEEE Trans Smart Grid 2019;10(2):2046–58.
[23] Guo Y, Cheng J, Yang Z, Wang C. Knowledge-inducing moea/d for interval multi-objective optimization problems; 2016. p. 2729–35.
[24] He Q, He Z, Duan S, Zhong Y. Multi-objective interval portfolio optimization modeling and solving for margin trading. Swarm Evol Comput 2022;75:101141.
[25] Limbourg P, Aponte D. An optimization algorithm for imprecise multi-objective problem functions, vol. 1; 2005. p. 459–66.
[26] Jiang C, Han X, Liu GR, Liu GP. A nonlinear interval number programming method for uncertain optimization problems. Eur J Oper Res 2008;188(1):1–13.
[27] Jing S, Dunwei G. Solving interval multi-objective optimization problems using evolutionary algorithms with lower limit of possibility degree. Chin J Electron 2013;22(2):269–72.
[28] Gong D, Sun J, Ji X. Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems. Inf Sci 2013;233:141–61.
[29] Zhang P, Xu R, Sun X, Gong D, Zhang Y, Choi J. A synthesized ranking-assisted nsga-ii for interval multi-objective optimization. In: 2016 IEEE congress on evolutionary computation. CEC; 2016. p. 861–8.
[30] Alolyan I. Algorithm for interval linear programming involving interval constraints. In: Ifsa world congress and nafips meeting; 2013.
[31] Xiao J, Zhang Y, Fu C. Comparison between methods of interval number ranking based on possibility. J Tianjin Univ 2011;44(8):705–11.
[32] Zhang L, Wang S, Zhang K, Zhang X, Sun Z, Zhang H, et al. Cooperative artificial bee colony algorithm with multiple populations for interval multiobjective optimization problems. IEEE Trans Fuzzy Syst 2019;27(5):1052–65.
[33] Liu G, Liu S. Direct method for uncertain multi-objective optimization based on interval non-dominated sorting. Struct Multidiscip Optim 2020;62(2):729–45.
[34] Yi J, Bai J, He H, Zhou W, Yao L. A multifactorial evolutionary algorithm for multitasking under interval uncertainties. IEEE Trans Evol Comput 2020;24(5):908–22.
[35] Sun S, Wang C, Wang Y, Zhu X, Lu H. Multi-objective optimization dispatching of a micro-grid considering uncertainty in wind power forecasting. Energy Rep 2022;8:2859–74.
[36] Xu X, Xu G, Chen J, Liu Z, Chen X, Zhang Y, et al. Multi-objective design optimization using hybrid search algorithms with interval uncertainty for thin-walled structures. Thin-Walled Struct 2022;175:109218.
[37] Zhang Z, Zhao M, Wang H, Cui Z, Zhang W. An efficient interval many-objective evolutionary algorithm for cloud task scheduling problem under uncertainty. Inf Sci 2022;583:56–72.
[38] Chen Z, Wu H, Chen Y, Cheng L, Zhang B. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm. Appl Soft Comput 2022;116:108192.
[39] Xu Y, Pi D, Yang S, Chen Y, Qin S, Zio E. An angle-based bi-objective optimization algorithm for redundancy allocation in presence of interval uncertainty. IEEE Trans Autom Sci Eng 2023;20(1):271–84.
[40] Gan X, Sun J, Gong D, Jia D, Dai H, Zhong Z. An adaptive reference vector-based interval multi-objective evolutionary algorithm. IEEE Trans Evol Comput 2022. https://doi.org/10.1109/TEVC.2022.3193294.
[41] Cui Z, Jin Y, Zhang Z, Xie L, Chen J. An interval multi-objective optimization algorithm based on elite genetic strategy. Inf Sci 2023;648:119533. https://doi.org/10.1016/j.ins.2023.119533.
[42] Zhou X, Wang X, Gu X. A decomposition-based multiobjective evolutionary algorithm with weight vector adaptation. Swarm Evol Comput 2021;61:100825.
[43] Chen ZW, Chen L, Bai X, Yang Q, Zhao FL. Interactive multi-attribute decision-making nsga-ii for constrained multi-objective optimization with interval numbers. Control Decis 2015;30(5):865–70.
[44] Zhao Q, Guo Y, Yao X, Gong D. Decomposition-based multi-objective optimization algorithms with adaptively adjusting weight vectors and neighborhoods. IEEE Trans Evol Comput 2022.
[45] Qi Y, Ma X, Liu F, Jiao L, Sun J, Wu J. Moea/d with adaptive weight adjustment. Evol Comput 2014;22(2):231–64.
[46] Huband S, Hingston P, Barone L, While L. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 2006;10(5):477–506.
[47] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable test problems for evolutionary multiobjective optimization; 2005.
[48] Cheng R, Li M, Tian Y, Zhang X, Yang S, Jin Y, et al. A benchmark test suite for evolutionary many-objective optimization. Complex Intell Syst 2017;3(1):67–81.
[49] Gong D, Sun J, Miao Z. A set-based genetic algorithm for interval many-objective optimization problems. IEEE Trans Evol Comput 2018;22(1):47–60.
[50] M SSR, Mallipeddi R, Das KN. A twin-archive guided decomposition based multi/many-objective evolutionary algorithm. Swarm Evol Comput 2022;71:101082.