COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 03/06 pp 291−307
Volume 1, Number 4, December 2021
DOI: 10.23919/CSMS.2021.0023

Novel PIO Algorithm with Multiple Selection Strategies for Many-Objective Optimization Problems

Zhihua Cui, Lihong Zhao, Youqian Zeng, Yeqing Ren, Wensheng Zhang*, and Xiao-Zhi Gao

Abstract: With the increase of problem dimensions, most solutions of existing many-objective optimization algorithms are non-dominant. Therefore, the selection of individuals and the retention of elite individuals are important. Existing algorithms cannot provide sufficient solution precision and guarantee the diversity and convergence of solution sets when solving practical many-objective industrial problems. Thus, this work proposes an improved many-objective pigeon-inspired optimization (ImMAPIO) algorithm with multiple selection strategies to solve many-objective optimization problems. Multiple selection strategies integrating hypervolume, knee point, and vector angles are utilized to increase selection pressure to the true Pareto Front. Thus, the accuracy, convergence, and diversity of solutions are improved. ImMAPIO is applied to the DTLZ and WFG test functions with four to fifteen objectives and compared against NSGA-III, GrEA, MOEA/D, RVEA, and many-objective Pigeon-inspired optimization algorithm. Experimental results indicate the superiority of ImMAPIO on these test functions.

Key words: pigeon-inspired optimization algorithm; many-objective optimization problem; multiple selection strategy; elite individual retention

1 Introduction

Swarm intelligence algorithms are optimization algorithms inspired by the behavior of some insects and animals in nature. These algorithms are widely common in the field of computational intelligence. For example, particle swarm optimization (PSO)[1, 2] simulates the predatory behavior of birds, which moves towards their closest neighbors and to the best state experienced. The bat algorithm (BA)[3, 4] simulates bats' use of sonar to detect prey and avoid obstacles in nature. Other swarm intelligence optimization algorithms, include the ant colony optimization (ACO) algorithm[5], cuckoo search algorithm (CS)[6, 7], firefly algorithm (FA)[8, 9], and artificial bee colony algorithm (ABC)[10]. New evolutionary algorithms have also been proposed to solve practical problems in various fields[11, 12].

The pigeon-inspired optimization (PIO) algorithm was proposed by Duan and Qiao[13] in 2014 as a swarm intelligence optimization algorithm for the air robot path planning problem.

PIO simulates pigeons' spontaneous homing behavior. However, the standard PIO simply solves single-objective problems. Given the challenge of selecting individuals according to the values calculated by functions, a new mechanism should be added to

• Zhihua Cui, Lihong Zhao, and Youqian Zeng are with the School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China. E-mail: cuizhihua@tyust.edu.cn; zhaolihong0312@163.com; youqianzeng997@163.com.
• Yeqing Ren is with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: 18435155956@163.com.
• Wensheng Zhang is with the State Key Laboratory of Intelligent Control and Management of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: wensheng.zhang@ia.ac.cn.
• Xiao-Zhi Gao is with the School of Computing, University of Eastern Finland, Kuopio, FI-70211, Finland. E-mail: xiao-zhi.gao@uef.fi.
∗ To whom correspondence should be addressed.

Manuscript received: 2021-08-06; revised: 2021-09-04; accepted: 2021-09-13

increase the selection pressure. With the increase of industrial problem dimensions, many multi-objective algorithms have been developed[14, 15]. And PIO algorithms for multi-objective optimization problems have also been proposed in various fields[16−18]. The proposal of multi-objective pigeon-inspired optimization (MPIO) approach enriches the application of PIO algorithms in real life, but the non-dominance solutions have become common in many-objective optimization problems. Moreover, the Pareto dominance mechanism has become inadequate in the selection of individuals, and it cannot guarantee convergence; thus, the PIO algorithm for many-objective optimization problem (MAPIO)[19] was proposed.

Several many-objective optimization algorithms are applied in various fields[20−22]. The MAPIO[19] extends the pigeon-inspired optimization algorithm to higher dimensions, but it only considers 4, 6, 8, and 10 objectives. At the same time, the solution accuracy is insufficient. The solutions are non-dominant especially when the number of objectives is more than 10. Hence, choosing the best solutions and balancing convergence and diversity are extremely important. To solve these problems, this study proposes the improved pigeon-inspired optimization algorithm for many-objective optimization problems with multiple selection strategies (ImMAPIO), which can preserve elite individuals and extend the influence of the initial population and elite individuals on velocity and position update.

The main contributions of this work can be summarized as follows:

(1) An improved many-objective pigeon-inspired optimization algorithm is proposed to solve many-objective optimization problems. The proposed algorithm can retain the elite individuals in the population and improve the solution accuracy. It also provides a new scheme for swarm intelligence algorithms to solve high-dimensional problems.

(2) A novel selection strategy integrating the advantage of hypervolume, knee point, and vector angles is suggested to improve the accuracy of selection and provide additional selection pressure to the true Pareto Front (PF). Through this strategy, the elite individuals are retained in the external archive set to accelerate convergence in the population iteration.

The rest of this paper is organized as follows. In Section 2, related work of the PIO algorithm is described. Details of the proposed algorithm are given in Section 3. In Section 4, results are discussed, along with the comparison of the proposed algorithm with five of the most advanced algorithms on WFG and DTLZ test functions. Finally, the conclusion and future work are summarized in Section 5.

2 Related Work

Pigeon-inspired optimization algorithm[13] simulates the behavior of pigeons' spontaneous homing. Standard PIO is composed of two independent iterative loops.

In these iterative loops, different cycle stages of flight are simulated using different navigation tools. These tools include landmark operator and map and compass operator. The process of pigeons using landmarks in navigate is simulated by the landmark operator model. The map and compass operator model imitates the effect of the height of the sun and the magnetic field of the earth on pigeons. To solve the problem of the standard PIO algorithm easily falling into the local optimum, Li and Duan[23] proposed the Simulated Annealing Pigeon-Inspired Optimization (SAPIO) algorithm with the goal of completing the target detection approach for unmanned aerial vehicles (UAVs) by using the simulated annealing mechanism and the and the edge potential function (EPF). The simulated annealing mechanism can form new individuals and avoid local optimum. The EPF can provide an attractive pattern for the target by calculating its value from the original image' edge map. Duan et al.[24] proposed the collaborative control method with predation and escape pigeon-inspired algorithm for UAV tight formation; the method uses the inner and outer ring controller to solve the tight formation cooperative control system problem. The predation and escape mechanism can avoid the tendency of the PIO algorithm to easily fall into the local optimum and improve the overall performance of PIO. Zhang and Duan[25] put forth the predator prey PIO in a dynamic environment used in areas of uninhabited combat aerial vehicles (UCAV). The algorithm can solve the problems of 3D path planning, improve global best properties with the use of the prey predator concept, and enhance the speed of convergence. By using the sequence of nodes to define the path of UACVs, all considerations of the ideal path are abstracted as the cost functions. Therefore, the establishment of the path planner becomes the optimization of the cost functions. Many scholars have applied PIO to multiple fields[26, 27].

With the increasing of scale of problems, the difficulty of solving the problems also increases. Hence, MPIO has been proposed accordingly. Qiu and Duan[28] proposed a variant of PIO called multi-objective PIO, which is used in the parameter design of brushless direct current motors. This method uses a Pareto sorting scheme[29] and consolidation operator to enhance the selection pressure of the individuals in a population to the true PF. Duan et al.[30] proposed a novel MPIO that uses the limit cycle-based mutant mechanism to produce new solutions and search directions. This method achieves good performance in terms of the diversity and accuracy of solutions. Liang et al.[31] proposed the MPIO algorithm with a self-organizing multimodal feature, which can improve the solution efficiency for multimodal problems; the algorithm uses a self-organizing map to transform the space dimension of a solution set and improves the space division of the solution set by using a special crowding distance. The efficiency of solving multimodal problems is thus improved.

The existing MPIO suffers from insufficient solution accuracy and algorithm convergence. Thus, the PIO algorithm for many-objective optimization problems is proposed. Cui et al.[19] proposed the MAPIO, which uses the balanceable fitness estimation (BFE) approach[32] and the external archive approach[32] to improve the selection capacity of individuals. The BFE approach[32] can measure the overall performance of a population's individuals. The external archive approach[32] can retain the elite individuals in the population.

In sum, the selection pressure and calculation accuracy of many-objective optimization algorithms need to be improved. Therefore, ImMAPIO is proposed. Details are presented in Section 3.

3 ImMAPIO

The growing scale of problems increases the difficulty of solution selection, which emphasizes convergence in the early stage of evolution and distribution in the late stage of evolution. Thus, to measure the performance of distribution and convergence, this study proposes the ImMAPIO with multiple selection strategies.

3.1 Multiple selection strategies for elite individual retention

3.1.1 Elite individual retention strategy

Most evolutionary algorithms mainly adopt corresponding evolutionary strategies for selected elite individuals to update target populations. When the complexity of a problem increases, these evolution strategies lead to the loss of some elite individuals in the process of evolution, resulting in the poor performance of solution sets. In recent years, some algorithms based on the elite individual mechanism have been proposed[33]. Inspired by the advantages of multiple selection strategies, this study proposes an elite individual retention strategy to improve the performance of solution sets in terms of convergence, diversity, uniformity, and solution precision.

As shown in Fig. 1, three different selection strategies are used to generate individuals from the initial population P and form new offsprings P_1, P_2, P_3. Each of the offsprings has N individuals. The selection strategy pool with $3N$ individuals is constructed by the offspring individuals produced by different strategies. Then, the best N of them is selected to update the external archive A. Finally, the retention of elite individuals is realized.

The selection strategy pool of the ImMAPIO includes the hypervolume (HV) based selection strategy[34], which is based on the HV indicator, which can comprehensively consider the convergence and diversity of solutions. In the present work, this strategy is mainly used to improve the selection accuracy of the algorithm and help it obtain satisfactory solution sets in solving industrial problems. The knee point based selection strategy[35] can be approximately regarded as a preference for a large HV and is thus efficient in accelerating the convergence of solutions. The knee points of a non-dominated front of the current population are a subset of the Pareto optimal solution. Therefore, the selection of knee points can ensure the good convergence of the solution set. The vector angle-based selection strategy[36] adopts the maximum vector

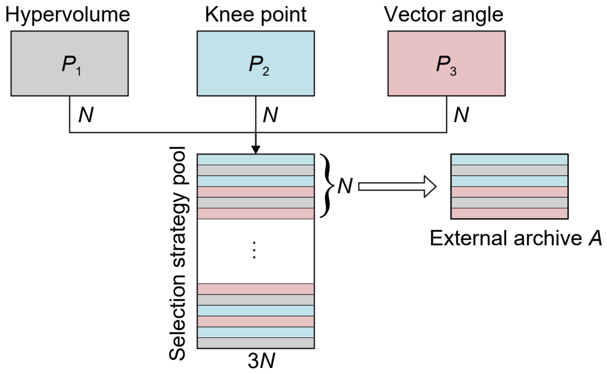


Fig. 1 Main procedure of elite individual retention.

angle first principle to ensure the diversity and uniformity of solution sets. The diversity and uniformity of solution sets are guaranteed by finding the solutions with the largest vector angle and extreme solutions.

The framework of the elite individual retention strategy is described in Algorithm 1. P_1, P_2, and P_3 are the offsprings generated by different selection strategies. In Lines 2−5, the sub-offsprings are obtained by different operators and form a new solution set S. In Lines 7−14, the Pareto dominance relationships of the solutions between A and S are compared. If A_j dominates S_i, it returns 1. When S_i dominates A_j, it returns −1. In Lines 16−22, the external archive A is updated according to the fitness values calculated by BFE method[32].

Algorithm 1 Elite individual retention strategy

```
Input: The population P, the population size N, the reference
       set R, the objectives number M, the set of knee points K,
       the external archive A;
1:  $P_1$ = Hypervolume_based ($P$);
2:  $P_2$ = Knee points_based ($P$);
3:  $P_3$ = Vector Angle_based ($P$);
4:  $S = P_1 \cup P_2 \cup P_3$;
5:  for $i = 1$ to $|S|$
6:      for $j = 1$ to $|A|$
7:          teb = CheckDominance$(S_i, A_j)$;
8:          if teb == 1
9:              $A_j$ is tagged as the dominated solution;
10:         else if teb == −1
11:             $S_i$ is tagged as the dominated solution;
12:         end if
13:     end for
14:     remove all of the tagged solutions from the archive $A$;
15:     if $S_i$ is not tagged
16:         add individual $S_i$ to $A$;
17:         if $|A| > N$
18:             calculate the function values;
19:             remove the worst individuals according to the fitness;
20:         end if
21:     end if
22: end for
Output: Archive $A$
```

3.1.2 Hypervolume-based selection strategy

The HV-based selection strategy[34] uses the rankings of HV values to assign fitness values to individuals and choose the individuals with the best performance. The HV indicator is strictly monotonic under Pareto dominance. When one Pareto set is approximately completely better than another, the value of the HV indicator is better than that of the other. Thus, this strategy is used in this work to improve the selection accuracy of the algorithm. In the high-dimensional objective space (objective number $r > 3$), the problem is recursively transformed into that in the low-dimensional objective space through continuous mapping to the $r - 1$ object space to reduce complexity. The recursive process is carried out until the number of objects is reduced to 3. The detailed calculation procedure of the HV-based selection strategy for a three-dimensional case is available in Ref. [37].

The Hypervolume calculation in the two-dimensional case is shown in Eq. (1):

$$\text{HV}(S) = \sum_{i=1}^{n} \left|\text{obj}_2(p_i) - \text{obj}_2(\text{Ref})\right| \cdot \left|\text{obj}_1(p_i) - \text{obj}_1(p_{i-1})\right| \tag{1}$$

where S is the solution set. $\text{obj}_i(p_j)$ represent the objective value of the j-th individual p_j on the i-th dimensional. $\text{obj}_2(\text{Ref})$ is the initial value of objective $\text{obj}_1(p_0)$. By comparing the HV values of individuals in the population, the individuals with relatively larger values are selected to guarantee the solving accuracy of the algorithm.

3.1.3 Knee point based selection strategy

The knee point based selection strategy[35] uses three championship selection strategies, namely, dominance relationship, knee point criterion, and weighted distance measure. Knee points are the subset of Pareto optimal solutions and are defined of bias of large Hypervolume. Thus, they can be used as the main criterion to measure population convergence. At the same time, the adaptive recognition of knee points in a small field is conducted without the knowledge of the number of knee points. The local knee points are located by the adaptive strategy so as to accelerate the convergence.

The dominant relationship is used initially to select the solution. In the case in which the solutions are not dominated by each other, it is considered whether the solutions are knee points, the knee points are given priority as the first choice. If neither is the case, then the weight distance between the solutions is judged. If the weight distance is the same, then a solution is selected randomly.

The determination of knee points is shown in Fig. 2, where B', G', and E' are the knee points. But if the number of the neighbor is 1, only E' is the knee point. Details of the knee point strategy are available in Ref. [35].

The weight distance (population size is p, nearest neighbors is k) is calculated as follows:

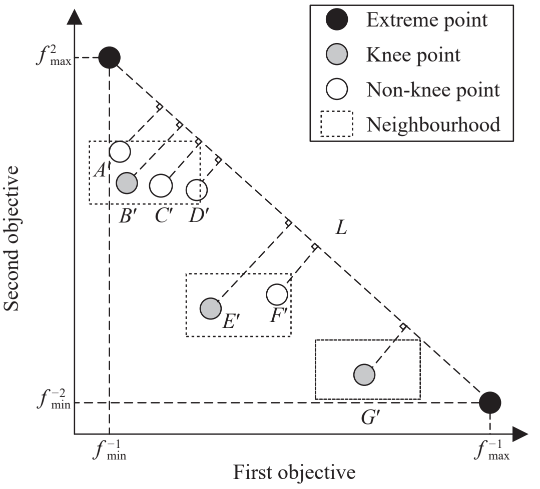


Fig. 2 Determining knee points for two objectives minimization problem.

$$\mathrm{DW}(p)=\sum_{i=1}^{k} w_{p_i}\mathrm{dis}_{pp_i} \tag{2}$$

$$w_{p_i}=\frac{r_{p_i}}{\sum_{i=1}^{k} r_{p_i}} \tag{3}$$

$$r_{p_i}=\frac{1}{\left|\mathrm{dis}_{pp_i}-\frac{1}{k}\sum_{i=1}^{k}\mathrm{dis}_{pp_i}\right|} \tag{4}$$

where p_i is the i-th nearest neighbor of individual p, w_{p_i} is the weight of p_i, dis_{pp_i} is the Euclidean distance calculated by the individual p and the nearest neighbor p_i, and r_{p_i} denotes the sorting of the distance dis_{pp_i}.

3.1.4 Vector angles based selection strategy

In the vector angle based selection strategy[36], the maximum vector angle first principle is adopted to ensure the distribution of the solution sets. After the nondominant sort, the norm of each solution in the normalized objective space is calculated. Then, the vector angles between two solutions in a normalized vector space are also calculated. The vector angles between individuals are used to select elite individuals. The diversity and uniformity of the solution set are guaranteed by finding the solutions with the largest vector angles.

x_j is the solution in the objective space. The norm of x_j is defined as Eq. (5), and the vector angle between two solutions is defined as Eq. (6):

$$\mathrm{norm}\left(x_j\right)\triangleq\sqrt{\sum_{i=1}^{m} f_i'\left(x_j\right)^2} \tag{5}$$

$$\mathrm{angle}\left(x_j,y_k\right)\triangleq\arccos\left|\frac{F'\left(x_j\right)\cdot F'\left(y_k\right)}{\mathrm{norm}\left(x_j\right)\cdot\mathrm{norm}\left(y_k\right)}\right| \tag{6}$$

where $f_i'\left(x_j\right)$ represents the i-th normalized value of objective vector x_j and $F'\left(x_j\right)\cdot F'\left(y_k\right)$ is the inner product between normalized objective vector $F'\left(x_j\right)$ and $F'\left(y_k\right)$, it can be calculated by Eq. (7):

$$F'\left(x_j\right)\cdot F'\left(y_k\right)=\sum_{i=1}^{m} f_i'\left(x_j\right)\cdot f_i'\left(y_k\right) \tag{7}$$

3.2 Velocity and position update

The ImMAPIO uses the velocity and position update equation proposed by Cui et al. [19] to provide a new search direction for individual pigeons, namely, the evolution direction from the center position point to the global optimal position direction. The proposed update equation is as follows:

$$\begin{aligned} V_i(t_{\mathrm{now}})={}&\mathrm{e}^{-Rt}\cdot V_i(t_{\mathrm{now}}-1)+r_1\cdot r_2\cdot\mathrm{tr}\cdot\left(1-\log_T^{t_{\mathrm{now}}}\right)\cdot\\ &\left(X_{\mathrm{glo}}-X_i(t_{\mathrm{now}})\right)+r_3\cdot r_4\cdot\mathrm{tr}\cdot\log_T^{t_{\mathrm{now}}}\cdot\\ &(X_{\mathrm{cen}}-X_i(t_{\mathrm{now}}))+r_5\cdot r_6\cdot\left(X_{\mathrm{glo}}-X_{\mathrm{cen}}\right)\end{aligned} \tag{8}$$

$$X_i(t_{\mathrm{now}})=X_i(t_{\mathrm{now}}-1)+V_i(t_{\mathrm{now}}) \tag{9}$$

where t_{now} represents the number of current iteration, T represents the largest number of iteration, R denotes the map and compass operator, tr is the migration factor used to ensure the smooth transition of the two operators in the algorithm, X_{glo} is position information about the global best in all pigeons, and X_{cen} is the center position information of some individuals in the current iteration. X_{cen} can be calculated according to Eq. (10).

$$X_{\mathrm{cen}}=\frac{\sum_{j=1}^{n_1^x} S_{1,j}^X}{n_1^x} S_1^X n_1^X \tag{10}$$

where $\sum_{j=1}^{n_1^x} S_{1,j}^X$ represents the sum of all solutions of individuals in the non-dominated set S_1^X and n_1^X is the number of the solutions in the set S_1^X.

The parameters r_1, r_3, and r_5 represent three random distribution numbers to provide extra random updating of different degree within [0, 1]. Meanwhile, r_2, r_4, and r_6 represent three learning factors and are defined according to the following Eq. (11):

$$r_i = \begin{cases} 0, & 0 < \text{rand}() \leqslant \dfrac{1}{M}; \\ 1, & \dfrac{1}{M} < \text{rand}() \leqslant 1 \end{cases} \tag{11}$$

where rand() is a random value between [0, 1] and M presents the number of the objectives. If the value of r_i is 0, it presents $X_i(t)$ will not update this part to the new velocity. If the value of r_i is 1, then the velocity updates this part dynamically with the change of parameter M.

3.3 Additional strategies

In ImMAPIO, an external archive[32] is adopted to retain the elitist solutions, which can provide the appropriate selection to the true PF. Meanwhile, by using the BFE method[32], elite individuals are retained in the external archive for the next generation of iterative optimization. External archive A preserves elite individuals. The new solution set S emerges from the population after the evolution process. By comparing the Pareto dominance relationships between S and A, the external archive A is updated until it reaches the ter minal condition.

Inspired by the work in Ref. [38], ImMAPIO uses an evolutionary search pattern that includes simulated binary crossover (SBX) and polynomial mutation (PM)[38] to further improve the selection of non-dominant solutions. By using the BFE method[32] and the evolutionary strategies in Ref. [38], the solution accuracy is improved, and elite individuals are retained.

3.4 Framework of ImMAPIO

This study investigates the role of different selection strategies in the process of choosing solutions and analyzes the characteristics of each strategy. This work screens different offspring generation solutions from the perspective of multiple selection strategies. The selection strategy pool is constructed to improve the algorithm performance and find relatively satisfactory solutions.

The ImMAPIO algorithm framework is detailed in Algorithm 2. The initialization process proceeds in the parent population P with N individuals, iteration number, and the external archive A. External archive A is null in initialized procedure. The position X_i and velocity V_i of pigeon individual p_i and the individual best positon are initialized in Lines 2−6. Then, the non-dominant solution in P is added to external archive A, and the fitness values of the solutions are calculated. In Lines 10−17, the velocity and position of pigeon individuals are updated by Eqs. (7) and (8), respectively. In Line 18, the new external archive A is composed of elite individuals by using multiple selection strategies. The external archive set A is then updated, and the new swarm S is generated by processing the population on the basis of evolutionary strategies SBX and PM. The function fitness of the new solutions in S is also calculated. Subsequently, the external archive A is updated again. This evolutionary procedure is repeated until the iteration reaches the maximum number.

Algorithm 2 Framework of ImMAPIO

```
1:  Initialization: The population $P$, individuals number $N$, the
    external archive $A$, the maximum number of iteration $T_{\max}$;
2:  for $i = 1$ to $N$
3:    position $X_i$ is randomly initialized, and $V_i = 0$ for $p_i$;
4:    calculate the function values of individual $p_i$;
5:    set $p_i^{\text{best}} = p_i$ as the individual best position to $p_i$;
6:  end for
7:  the non-dominance solutions in $P$ are added into archive $A$;
8:  calculate the fitness of the individuals in the archive $A$;
9:  while $T \leqslant T_{\max}$
10:   for $i = 1$ to $N$
11:         update the velocity $V_i$ of $p_i$ by Eq. (7);
12:         update the position $X_i$ of $p_i$ by Eq. (8);
13:         evaluate the objective fitness for $p_i$;
14:         if $p_i^{\text{best}}$ cannot dominate $p_i$
15:             set $p_i^{\text{best}} = p_i$;
16:         end if
17:   end for
18:   Selection strategy pool is generated by multiple selection
        strategies in Section 3.1;
19:   Choose the elite individuals into the archive $A$;
20:   $A$ = Update_Archive $(A, P)$;
21:   executing evolutionary strategies on $A$ to get a new swarm $S$;
22:   calculate the objectives fitness of new solutions in $S$;
23:   $A$ = Update_Archive $(A, S)$;
24: end while
25: output $A$;
```

4 Experimental Results and Discussion

In this section, the performance of ImMAPIO in standard benchmark problems, such as DTLZ[39] and WFG[40], is compared with that of NSGA-III[41], GrEA[42], MOEA/D[43], RVEA[44], and MAPIO[19]. Meanwhile, the results of the comparative experiment are subjected to the Wilcoxon signed-rank test to prove the superiority of the proposed algorithm.

4.1 Benchmark problem

The experiments were conducted on standard benchmark problems, such as DTLZ[27] and WFG[28]. DTLZ1–DTLZ8 are selected as the test examples for the DTLZ problems, and WFG1–WFG9 are chosen for the WFG problems. For these problems in the experiment, the cases with objective numbers of 4, 6, 8, 10, and 15 should be considered.

The DTLZ[39] and WFG[40] test problems have their own characteristics, such as linear, multi-modal, disconnected, mixed convex/concave, and deceptive characteristics. Table 1 lists the characteristics of the different DTLZ and WFG test problems. Table 2 presents the parameters of the different test functions, where M is the number of objectives, k is a fixed position-related parameter, and l is set to 20 and is the fixed distance-related parameter.

The simulated binary crossover SBX[41] and polynomial mutation PM[41] have been adopted in some algorithms[41, 44] to produce offspring; the distribution indexes of crossover and mutation are set to $n_{\text{c}} = 20$ and $n_{\text{m}} = 20$, respectively. And the probability of crossover and mutation are set to $p_{\text{c}} = 1.0$ and $p_{\text{m}} = 1/D$, where D represents the number of decision variables. For GrEA[42], the setting of the parameter *div* which denotes the number of divisions in each dimension is taken from Ref. [42]. The parameter pertaining to the range of neighborhood in MOEA/D is set to $N/10$ for all of the test problems; the other relevant parameters are available in Ref. [43]. The parameter pertaining to MAPIO is set to $\text{tr} = 1$, which indicates that the transition factor is 1.

Table 1 Characterization of test instances DTLZ1−DTLZ7 and WFG1−WFG9.

Problem	Feature
DTLZ1	Linear, multi-modal
DTLZ2	Concave
DTLZ3	Concave, multi-modal
DTLZ4	Concave, biased
DTLZ5	Concave, degenerate
DTLZ6	Concave, degenerate, biased
DTLZ7	Mixed, disconnected, multi-modal, scaled
WFG1	Mixed, biased, scaled
WFG2	Convex, disconnected, multi-modal, non-separable, scaled
WFG3	Linear, degenerate, non-separable, scaled
WFG4	Concave, multi-modal, scaled
WFG5	Concave, deceptive, scaled
WFG6	Concave, non-separable, scaled
WFG7	Concave, biased, scaled
WFG8	Concave, biased, non-separable, scaled
WFG9	Concave, biased, multi-modal, deceptive, non-separable, scaled

Table 2 Parameter settings for the test functions.

Test function	Number of decision variables (n)	Parameter	Maxgen
DTLZ1	$M-1+k$	$k=5$	700
DTLZ2	$M-1+k$	$k=10$	250
DTLZ3	$M-1+k$	$k=10$	1000
DTLZ4−DTLZ6	$M-1+k$	$k=10$	250
DTLZ7, DTLZ8	$M-1+k$	$k=20$	250
WFG1	$M-1+k$	$k=M-1,\ l=20$	1000
WFG2	$M-1+k$	$k=M-1,\ l=20$	700
WFG3−WFG9	$M-1+k$	$k=M-1,\ l=20$	250

The population size changes with the number of objectives. The settings of population size are 120, 132, 156, 275, and 135, with the number of objectives are 4, 6, 8, 10, and 15. The running time is set to 30 times with each algorithm for each test function.

4.2 Performance metric

In the evaluation of the performance of many-objective evolutionary algorithms, the convergence, homogeneity, and universality of the solution set are usually evaluated. At the same time, the indicators of diversity include evenness and spread. Therefore, two indicators, namely, inverted generational distance (IGD)[45] and HV[46], which can simultaneously measure convergence and diversity, are used in the performance evaluation.

(1) Inverted generational distance (IGD)

The IGD[45] indicates the average distance from each reference point to the nearest solution. The smaller the IGD value is, the better the comprehensive performance of the algorithm is. The IGD can be calculated using Eq. (12):

$$\text{IGD}(Q,\ Q^*) = \frac{\sum\limits_{x\in Q^*} \min\limits_{y\in Q} \text{dis}(x,y)}{|Q^*|} \tag{12}$$

where Q is the approximation solution set obtained by the algorithm. Q^* represents a reference points set that is uniformly distributed and sampled from true PF. $\text{dis}(x,y)$ denotes the Euclidean distance between point x in reference points set Q^* and point y belongs to approximation solution set Q. During the calculation of IGD value, the computation is efficient and measures the convergence and multiformity of the algorithm.

(2) Hypervolume (HV)

HV[46] indicates the volume of a region in the objective space bounded by the non-dominant solution set PF and the reference point obtained by the algorithm. The higher the HV value is, the better the comprehensive performance of the algorithm is. The value of HV can be calculated using Eq. (13):

$$\mathrm{HV}=\delta\left(\bigcup_{i=1}^{|S|} v_i\right) \tag{13}$$

where δ represents the Lebesgue measure of the volume, $|S|$ is the number of individuals in the non-dominance solution set PF, v_i is the hypervolume of the reference point and the i-th solution in the PF set. During the calculation of HV value, the non-dominated solution set PF and reference set P^* need not be known, but the selection of reference points determines the accuracy of the HV indicator to a certain degree.

4.3 Experiment results and analysis

Tables 3 − 5 reveal the details of the comparison results. Table 3 shows the performance of the IGD in the DTLZ test functions relative to the six algorithms. Tables 4 and 5 show the properties of the IGD and HV values of the WFG test functions on the six algorithms. In these tables, the bold parts represent the best results. "+", "–", and "=" in Tables 3, 4, and 5 denote that the results from other algorithms are higher than, lower than, or equal to the results from the proposed ImMAPIO, respectively. On the basis of the numerical analysis of the experimental results, the superiority of ImMAPIO is analyzed.

Table 3 IGD values of six algorithms for different objectives in the DTLZ test problems.

Problem	M	IGD value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
DTLZ1	4	2.3888×10^{1} (5.83×10^{0}) −	2.4071×10^{1} (7.48×10^{0}) −	**7.2175×10^{0}** **(3.10×10^{0})** +	1.9722×10^{1} (5.57×10^{0}) =	1.6976×10^{1} (5.99×10^{0}) =	1.9585×10^{1} (5.62×10^{0})
	6	2.0401×10^{1} (7.24×10^{0}) =	2.4501×10^{1} (7.27×10^{0}) =	**5.8150×10^{0}** **(2.53×10^{0})** +	1.6796×10^{1} (4.97×10^{0}) +	1.8277×10^{1} (4.61×10^{0}) +	2.2822×10^{1} (7.40×10^{0})
	8	2.5337×10^{1} (7.80×10^{0}) =	2.3780×10^{1} (7.10×10^{0}) =	**6.2681×10^{0}** **(2.98×10^{0})** +	1.7858×10^{1} (4.56×10^{0}) +	1.7862×10^{1} (5.85×10^{0}) +	2.3125×10^{1} (6.09×10^{0})
	10	2.7201×10^{1} (8.48×10^{0}) −	2.6984×10^{1} (6.26×10^{0}) −	**7.3240×10^{0}** **(3.16×10^{0})** +	2.3775×10^{1} (7.71×10^{0}) =	1.9401×10^{1} (5.50×10^{0}) =	2.0198×10^{1} (5.22×10^{0})
	15	2.2261×10^{1} (7.88×10^{0}) =	2.3769×10^{1} (7.69×10^{0}) =	**7.3268×10^{0}** **(3.23×10^{0})** +	1.2768×10^{1} (3.39×10^{0}) +	1.6969×10^{1} (4.65×10^{0}) +	2.2674×10^{1} (6.23×10^{0})
DTLZ2	4	5.0397×10^{-1} (3.18×10^{-2}) −	4.9442×10^{-1} (3.47×10^{-2}) −	4.3144×10^{-1} (4.43×10^{-2}) −	5.2941×10^{-1} (4.44×10^{-2}) −	3.5691×10^{-1} (2.30×10^{-2}) −	**3.2650×10^{-1}** **(2.14×10^{-2})**
	6	7.3475×10^{-1} (3.40×10^{-2}) −	7.1729×10^{-1} (3.87×10^{-2}) −	7.1453×10^{-1} (8.24×10^{-2}) −	7.5866×10^{-1} (4.37×10^{-2}) −	5.5658×10^{-1} (2.88×10^{-2}) −	**5.0884×10^{-1}** **(2.66×10^{-2})**
	8	9.0014×10^{-1} (4.31×10^{-2}) −	8.6907×10^{-1} (4.56×10^{-2}) −	8.2620×10^{-1} (9.30×10^{-2}) −	9.3744×10^{-1} (3.85×10^{-2}) −	7.3065×10^{-1} (3.08×10^{-2}) −	**6.7541×10^{-1}** **(3.50×10^{-2})**
	10	9.9208×10^{-1} (4.50×10^{-2}) =	9.9122×10^{-1} (3.73×10^{-2}) =	1.0046×10^{0} (3.35×10^{-2}) =	9.9227×10^{-1} (2.87×10^{-2}) =	**9.8746×10^{-1}** **(3.82×10^{-2}) =**	9.9872×10^{-1} (2.80×10^{-2})
	15	1.2018×10^{0} (3.46×10^{-2}) −	1.1669×10^{0} (3.58×10^{-2}) −	1.1952×10^{0} (7.96×10^{-2}) −	1.2790×10^{0} (5.74×10^{-2}) −	1.0439×10^{0} (2.58×10^{-2}) −	**1.0212×10^{0}** **(2.77×10^{-2})**
DTLZ3	4	2.6493×10^{2} (5.65×10^{1}) =	2.4989×10^{2} (3.76×10^{1}) +	**7.5101×10^{1}** **(1.75×10^{1})** +	2.3674×10^{2} (3.36×10^{1}) +	1.8886×10^{2} (3.75×10^{1}) +	2.8646×10^{2} (5.91×10^{1})
	6	2.9861×10^{2} (4.81×10^{1}) =	2.9476×10^{2} (5.29×10^{1}) =	**6.1431×10^{1}** **(1.38×10^{1})** +	2.3572×10^{2} (5.58×10^{1}) +	2.3187×10^{2} (4.00×10^{1}) +	2.7578×10^{2} (5.85×10^{1})
	8	3.0952×10^{2} (6.44×10^{1}) −	3.6041×10^{2} (5.95×10^{1}) −	**7.0422×10^{1}** **(2.00×10^{1})** +	2.2531×10^{2} (4.34×10^{1}) +	2.4397×10^{2} (4.01×10^{1}) +	2.6897×10^{2} (3.95×10^{1})
	10	3.7231×10^{2} (5.99×10^{1}) −	3.6206×10^{2} (5.61×10^{1}) −	**7.0760×10^{1}** **(2.39×10^{1})** +	3.2356×10^{2} (5.69×10^{1}) =	2.4865×10^{2} (3.56×10^{1}) +	2.9963×10^{2} (5.72×10^{1})
	15	3.5092×10^{2} (7.94×10^{1}) −	3.6350×10^{2} (7.67×10^{1}) −	**7.9687×10^{1}** **(1.82×10^{1})** +	2.0457×10^{2} (5.01×10^{1}) +	2.2626×10^{2} (3.74×10^{1}) +	2.9846×10^{2} (5.63×10^{1})
DTLZ4	4	8.7855×10^{-1} (8.38×10^{-2}) −	8.0189×10^{-1} (9.52×10^{-2}) −	8.8553×10^{-1} (2.03×10^{-1}) −	7.4879×10^{-1} (7.54×10^{-2}) =	7.5976×10^{-1} (9.77×10^{-2}) =	**7.4544×10^{-1}** **(1.04×10^{-1})**
	6	1.0763×10^{0} (7.93×10^{-2}) −	1.0064×10^{0} (8.25×10^{-2}) −	1.1349×10^{0} (1.00×10^{-1}) −	1.0685×10^{0} (8.44×10^{-2}) −	8.8936×10^{-1} (6.75×10^{-2}) =	**8.8272×10^{-1}** **(8.88×10^{-2})**
	8	1.1102×10^{0} (7.31×10^{-2}) −	1.0409×10^{0} (5.48×10^{-2}) −	1.1979×10^{0} (1.07×10^{-1}) −	1.1272×10^{0} (7.53×10^{-2}) −	9.4858×10^{-1} (6.68×10^{-2}) =	**9.1424×10^{-1}** **(6.48×10^{-2})**
	10	1.1442×10^{0} (6.09×10^{-2}) =	**1.1291×10^{0}** **(6.08×10^{-2}) =**	1.1370×10^{0} (5.25×10^{-2}) =	1.1403×10^{0} (5.81×10^{-2}) =	1.1425×10^{0} (3.92×10^{-2}) =	1.1444×10^{0} (5.00×10^{-2})
	15	1.2471×10^{0} (5.36×10^{-2}) −	1.1985×10^{0} (5.23×10^{-2}) −	1.3014×10^{0} (7.10×10^{-2}) −	1.2929×10^{0} (5.56×10^{-2}) −	**1.0667×10^{0}** **(4.07×10^{-2}) =**	1.0749×10^{0} (5.46×10^{-2})

(To be continued)

Table 3 IGD values of six algorithms for different objectives in the DTLZ test problems. (Continued)

Problem	M	IGD value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
DTLZ5	4	3.4967×10^{-1} (5.29×10^{-2}) −	3.4604×10^{-1} (3.51×10^{-2}) −	2.3627×10^{-1} (6.94×10^{-2}) −	4.2477×10^{-1} (5.39×10^{-2}) −	**1.4400×10^{-1}** **(5.83×10^{-2}) +**	1.7401×10^{-1} (2.20×10^{-2})
	6	3.8873×10^{-1} (3.61×10^{-2}) −	3.8807×10^{-1} (4.20×10^{-2}) −	3.0491×10^{-1} (6.37×10^{-2}) −	5.0149×10^{-1} (5.89×10^{-2}) −	1.8638×10^{-1} (3.78×10^{-2}) =	**1.8392×10^{-1}** **(2.50×10^{-2})**
	8	3.6485×10^{-1} (3.96×10^{-2}) −	3.8458×10^{-1} (4.21×10^{-2}) −	2.7851×10^{-1} (8.50×10^{-2}) −	5.4130×10^{-1} (6.42×10^{-2}) −	**1.8872×10^{-1}** **(2.78×10^{-2}) =**	1.9267×10^{-1} (2.47×10^{-2})
	10	3.6992×10^{-1} (3.74×10^{-2}) =	3.6597×10^{-1} (3.67×10^{-2}) =	3.8113×10^{-1} (3.00×10^{-2}) =	**3.5890×10^{-1}** **(3.77×10^{-2}) =**	3.7866×10^{-1} (3.49×10^{-2}) =	3.7017×10^{-1} (3.42×10^{-2})
	15	3.9242×10^{-1} (4.33×10^{-2}) −	3.8907×10^{-1} (5.12×10^{-2}) −	3.2872×10^{-1} (8.10×10^{-2}) −	5.6115×10^{-1} (7.91×10^{-2}) −	2.0885×10^{-1} (2.98×10^{-2}) =	**2.0679×10^{-1}** **(3.60×10^{-2})**
DTLZ6	4	8.0477×10^{0} (2.14×10^{-1}) −	7.9284×10^{0} (2.17×10^{-1}) −	7.8025×10^{0} (4.67×10^{-1}) −	8.0799×10^{0} (3.18×10^{-1}) −	**6.4606×10^{0}** **(7.11×10^{-1}) =**	6.6087×10^{0} (5.38×10^{-1})
	6	8.2582×10^{0} (1.91×10^{-1}) −	8.1236×10^{0} (1.45×10^{-1}) −	7.7522×10^{0} (5.25×10^{-1}) −	8.3874×10^{0} (2.73×10^{-1}) −	6.6396×10^{0} (6.90×10^{-1}) =	**6.6123×10^{0}** **(5.19×10^{-1})**
	8	8.1601×10^{0} (1.80×10^{-1}) −	8.1142×10^{0} (1.87×10^{-1}) −	7.4586×10^{0} (6.07×10^{-1}) −	8.4815×10^{0} (2.30×10^{-1}) −	**6.3743×10^{0}** **(6.18×10^{-1}) =**	6.3994×10^{0} (6.41×10^{-1})
	10	8.3975×10^{0} (1.54×10^{-1}) =	8.3974×10^{0} (9.50×10^{-2}) =	8.3814×10^{0} (1.32×10^{-1}) =	**8.3201×10^{0}** **(1.68×10^{-1}) =**	8.3756×10^{0} (1.09×10^{-1}) =	8.3867×10^{0} (1.13×10^{-1})
	15	8.2145×10^{0} (2.42×10^{-1}) −	8.1845×10^{0} (2.33×10^{-1}) −	7.1694×10^{0} (5.63×10^{-1}) −	8.5950×10^{0} (4.14×10^{-1}) −	6.7692×10^{0} (6.39×10^{-1}) −	**6.3664×10^{0}** **(7.91×10^{-1})**
DTLZ7	4	1.0721×10^{1} (9.39×10^{-1}) =	1.0523×10^{1} (8.97×10^{-1}) =	**6.8686×10^{0}** **(1.51×10^{0}) +**	9.0019×10^{0} (8.53×10^{-1}) +	1.1170×10^{1} (1.12×10^{0}) −	1.0333×10^{1} (7.84×10^{-1})
	6	1.7522×10^{1} (1.41×10^{0}) −	1.7168×10^{1} (1.25×10^{0}) −	**1.2474×10^{1}** **(1.92×10^{0}) +**	1.6993×10^{1} (1.70×10^{0}) −	1.6776×10^{1} (1.79×10^{0}) −	1.5615×10^{1} (1.70×10^{0})
	8	2.3403×10^{1} (1.63×10^{0}) −	2.3105×10^{1} (2.46×10^{0}) −	**1.6554×10^{1}** **(3.01×10^{0}) +**	2.2673×10^{1} (1.94×10^{0}) −	2.2713×10^{1} (1.78×10^{0}) =	2.1635×10^{1} (2.22×10^{0})
	10	2.8954×10^{1} (1.83×10^{0}) =	**2.8632×10^{1}** **(2.23×10^{0}) =**	2.9084×10^{1} (2.09×10^{0}) =	2.9295×10^{1} (1.82×10^{0}) =	2.9114×10^{1} (2.52×10^{0}) =	2.8991×10^{1} (2.84×10^{0})
	15	4.8157×10^{1} (3.39×10^{0}) −	4.7771×10^{1} (3.24×10^{0}) −	**3.7765×10^{1}** **(5.13×10^{0}) +**	4.6075×10^{1} (3.23×10^{0}) =	4.8246×10^{1} (3.13×10^{0}) −	4.6251×10^{1} (2.95×10^{0})
DTLZ8	4	2.6816×10^{-1} (1.20×10^{-2}) −	2.5290×10^{-1} (1.66×10^{-2}) −	2.9764×10^{-1} (3.31×10^{-2}) −	2.5702×10^{-1} (1.79×10^{-2}) −	2.5491×10^{-1} (2.05×10^{-2}) −	**2.3448×10^{-1}** **(1.91×10^{-2})**
	6	3.2956×10^{-1} (1.36×10^{-2}) −	3.2875×10^{-1} (1.60×10^{-2}) −	3.7389×10^{-1} (3.52×10^{-2}) −	3.5488×10^{-1} (2.08×10^{-2}) −	3.1313×10^{-1} (1.76×10^{-2}) −	**2.9977×10^{-1}** **(1.69×10^{-2})**
	8	3.8214×10^{-1} (1.36×10^{-2}) −	3.7347×10^{-1} (1.79×10^{-2}) −	4.3741×10^{-1} (4.44×10^{-2}) −	4.1455×10^{-1} (1.95×10^{-2}) −	3.5998×10^{-1} (1.64×10^{-2}) −	**3.4724×10^{-1}** **(1.28×10^{-2})**
	10	**4.0439×10^{-1}** **(1.03×10^{-2}) =**	4.0760×10^{-1} (1.28×10^{-2}) =	4.0825×10^{-1} (1.33×10^{-2}) =	4.1142×10^{-1} (1.06×10^{-2}) =	4.1032×10^{-1} (1.25×10^{-2}) =	4.0898×10^{-1} (1.14×10^{-2})
	15	5.0885×10^{-1} (1.47×10^{-2}) −	5.1728×10^{-1} (1.20×10^{-2}) −	6.0916×10^{-1} (4.75×10^{-2}) −	5.6736×10^{-1} (2.08×10^{-2}) −	4.9992×10^{-1} (1.52×10^{-2}) =	**4.9466×10^{-1}** **(1.88×10^{-2})**
+/−/=		0/28/12	1/28/11	14/20/6	8/21/11	9/11/20	

Note: "+", "−", and "=" denote that the results from other algorithms are higher than, lower than, or equal to the results from the proposed ImMAPIO, respectively. The bold parts represent the best results. The value before the brackets represents the mean value of the indicator used to measure the algorithm, and the value in the brackets refers to the standard deviation of the indicator.

Table 4 IGD values of six algorithms for different objectives in the WFG test problems.

Problem	M	IGD value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
WFG1	4	2.2679×10^{0} (5.83×10^{-2}) +	2.0836×10^{0} (6.07×10^{-2}) +	**1.9482×10^{0}** **(9.49×10^{-2}) +**	2.1434×10^{0} (1.60×10^{-1}) +	2.1668×10^{0} (5.32×10^{-2}) +	2.3239×10^{0} (4.25×10^{-2})
	6	2.6632×10^{0} (3.62×10^{-2}) =	2.5207×10^{0} (4.50×10^{-2}) +	**2.3985×10^{0}** **(3.50×10^{-2}) +**	2.7377×10^{0} (1.51×10^{-1}) =	2.5629×10^{0} (2.97×10^{-2}) +	2.6873×10^{0} (4.78×10^{-2})
	8	3.0568×10^{0} (3.73×10^{-2}) =	2.9148×10^{0} (4.07×10^{-2}) +	**2.8456×10^{0}** **(3.58×10^{-2}) +**	3.0935×10^{0} (1.22×10^{-1}) =	2.9695×10^{0} (5.04×10^{-2}) +	3.0501×10^{0} (5.50×10^{-2})
	10	3.3744×10^{0} (2.58×10^{-2}) =	3.3030×10^{0} (3.10×10^{-2}) +	**3.2350×10^{0}** **(3.25×10^{-2}) +**	3.4821×10^{0} (8.31×10^{-2}) −	3.3225×10^{0} (2.93×10^{-2}) +	3.3813×10^{0} (3.10×10^{-2})
	15	4.3397×10^{0} (6.01×10^{-2}) =	4.2716×10^{0} (5.94×10^{-2}) +	**4.2636×10^{0}** **(3.39×10^{-2}) +**	4.3775×10^{0} (6.75×10^{-2}) −	4.2776×10^{0} (5.32×10^{-2}) +	4.3155×10^{0} (3.48×10^{-2})

(To be continued)

Table 4 IGD values of six algorithms for different objectives in the WFG test problems. (Continued)

Problem	M	IGD value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
WFG2	4	8.8248×10^{-1} (6.15×10^{-2}) −	8.1748×10^{-1} (1.00×10^{-1}) −	9.7502×10^{-1} (1.55×10^{-1}) −	8.4418×10^{-1} (1.13×10^{-1}) −	**5.7956×10^{-1}** **(5.57×10^{-2}) +**	7.3098×10^{-1} (7.18×10^{-2})
	6	1.4167×10^{0} (2.03×10^{-1}) =	1.1910×10^{0} (1.14×10^{-1}) +	1.5752×10^{0} (2.99×10^{-1}) =	1.2032×10^{0} (1.56×10^{-1}) +	**1.1057×10^{0}** **(2.22×10^{-1}) +**	1.4936×10^{0} (2.46×10^{-1})
	8	2.1908×10^{0} (3.82×10^{-1}) =	1.7794×10^{0} (2.27×10^{-1}) +	2.4856×10^{0} (5.20×10^{-1}) −	**1.7235×10^{0}** **(1.70×10^{-1}) +**	1.7806×10^{0} (2.64×10^{-1}) +	2.1205×10^{0} (3.58×10^{-1})
	10	3.3382×10^{0} (4.79×10^{-1}) −	2.9570×10^{0} (5.79×10^{-1}) =	2.6705×10^{0} (6.97×10^{-1}) =	2.7563×10^{0} (4.26×10^{-1}) =	**2.5374×10^{0}** **(3.46×10^{-1}) =**	2.6705×10^{0} (3.92×10^{-1})
	15	4.7815×10^{0} (1.14×10^{0}) =	3.8352×10^{0} (8.18×10^{-1}) +	4.4897×10^{0} (1.49×10^{0}) +	**3.8040×10^{0}** **(6.68×10^{-1}) +**	4.4664×10^{0} (7.12×10^{-1}) +	5.0460×10^{0} (8.25×10^{-1})
WFG3	4	7.6595×10^{-1} (3.04×10^{-2}) −	7.5258×10^{-1} (5.13×10^{-2}) −	1.2244×10^{0} (2.43×10^{-1}) −	8.7538×10^{-1} (4.80×10^{-2}) −	6.1213×10^{-1} (3.97×10^{-2}) −	**5.7753×10^{-1}** **(3.57×10^{-2})**
	6	1.1042×10^{0} (4.48×10^{-2}) −	1.0629×10^{0} (5.37×10^{-2}) −	1.8104×10^{0} (3.36×10^{-1}) −	1.5754×10^{0} (9.76×10^{-2}) −	8.1342×10^{-1} (3.78×10^{-2}) −	**7.7909×10^{-1}** **(3.96×10^{-2})**
	8	1.3299×10^{0} (6.67×10^{-2}) −	1.3116×10^{0} (6.40×10^{-2}) −	2.4667×10^{0} (3.43×10^{-1}) −	2.2462×10^{0} (1.56×10^{-1}) −	9.5307×10^{-1} (5.42×10^{-2}) =	**9.3275×10^{-1}** **(5.69×10^{-2})**
	10	1.5305×10^{0} (6.61×10^{-2}) =	**1.5074×10^{0}** **(6.31×10^{-2}) =**	1.5220×10^{0} (6.56×10^{-2}) =	1.5149×10^{0} (6.00×10^{-2}) =	1.5104×10^{0} (5.34×10^{-2}) =	1.5172×10^{0} (6.61×10^{-2})
	15	2.3159×10^{0} (1.12×10^{-1}) −	2.1354×10^{0} (8.42×10^{-2}) −	5.3943×10^{0} (1.40×10^{0}) −	4.9843×10^{0} (2.88×10^{-1}) −	1.5755×10^{0} (9.29×10^{-2}) =	**1.5687×10^{0}** **(1.44×10^{-1})**
WFG4	4	1.2495×10^{0} (9.74×10^{-2}) −	1.0623×10^{0} (9.81×10^{-2}) −	1.3239×10^{0} (1.53×10^{-1}) −	1.1744×10^{0} (1.18×10^{-1}) −	1.0430×10^{0} (1.03×10^{-1}) −	**9.5073×10^{-1}** **(7.55×10^{-2})**
	6	3.5989×10^{0} (2.80×10^{-1}) −	3.4923×10^{0} (2.35×10^{-1}) −	3.6606×10^{0} (4.57×10^{-1}) −	3.9492×10^{0} (3.52×10^{-1}) −	2.9753×10^{0} (2.94×10^{-1}) −	**2.7154×10^{0}** **(2.41×10^{-1})**
	8	6.5429×10^{0} (3.66×10^{-1}) −	6.3809×10^{0} (3.94×10^{-1}) −	6.0238×10^{0} (4.31×10^{-1}) −	7.1237×10^{0} (4.26×10^{-1}) −	5.4316×10^{0} (4.31×10^{-1}) =	**5.3036×10^{0}** **(4.05×10^{-1})**
	10	**9.3553×10^{0}** **(4.51×10^{-1}) =**	9.5364×10^{0} (3.67×10^{-1}) =	9.4802×10^{0} (3.87×10^{-1}) =	9.5797×10^{0} (4.14×10^{-1}) =	9.4157×10^{0} (4.38×10^{-1}) =	9.5146×10^{0} (5.16×10^{-1})
	15	1.8761×10^{1} (7.33×10^{-1}) −	1.7837×10^{1} (7.92×10^{-1}) −	1.8129×10^{1} (1.12×10^{0}) −	1.9520×10^{1} (9.30×10^{-1}) −	1.7050×10^{1} (1.12×10^{0}) =	**1.6813×10^{1}** **(9.02×10^{-1})**
WFG5	4	1.1252×10^{0} (4.47×10^{-2}) −	1.0532×10^{0} (4.12×10^{-2}) −	1.5054×10^{0} (2.21×10^{-1}) −	1.1365×10^{0} (3.27×10^{-2}) −	1.0562×10^{0} (3.54×10^{-2}) −	**9.8621×10^{-1}** **(3.16×10^{-2})**
	6	2.7896×10^{0} (1.11×10^{-1}) −	2.6278×10^{0} (1.29×10^{-1}) −	3.2877×10^{0} (2.31×10^{-1}) −	2.8435×10^{0} (1.41×10^{-1}) −	2.3348×10^{0} (8.44×10^{-2}) −	**2.2062×10^{0}** **(6.07×10^{-2})**
	8	5.0017×10^{0} (2.60×10^{-1}) −	4.7340×10^{0} (2.40×10^{-1}) −	5.5699×10^{0} (3.87×10^{-1}) −	5.2512×10^{0} (2.32×10^{-1}) −	4.1651×10^{0} (1.11×10^{-1}) −	**4.0178×10^{0}** **(1.10×10^{-1})**
	10	7.2526×10^{0} (2.88×10^{-1}) =	**7.1248×10^{0}** **(3.24×10^{-1}) =**	7.2314×10^{0} (3.42×10^{-1}) =	7.2017×10^{0} (2.96×10^{-1}) =	7.1677×10^{0} (3.12×10^{-1}) =	7.2006×10^{0} (2.65×10^{-1})
	15	1.4940×10^{1} (6.03×10^{-1}) −	1.5364×10^{1} (6.09×10^{-1}) −	1.5001×10^{1} (1.06×10^{0}) −	1.6095×10^{1} (8.23×10^{-1}) −	1.3166×10^{1} (5.92×10^{-1}) −	**1.2696×10^{1}** **(5.08×10^{-1})**
WFG6	4	1.2505×10^{0} (4.23×10^{-2}) −	1.2093×10^{0} (4.29×10^{-2}) −	1.6584×10^{0} (2.31×10^{-1}) −	1.3003×10^{0} (6.09×10^{-2}) −	1.1608×10^{0} (4.65×10^{-2}) −	**1.0867×10^{0}** **(4.38×10^{-2})**
	6	2.9531×10^{0} (1.75×10^{-1}) −	2.8090×10^{0} (1.03×10^{-1}) −	3.4805×10^{0} (3.13×10^{-1}) −	3.1188×10^{0} (1.47×10^{-1}) −	2.6019×10^{0} (1.29×10^{-1}) −	**2.4118×10^{0}** **(9.83×10^{-2})**
	8	5.2762×10^{0} (2.94×10^{-1}) −	5.0758×10^{0} (2.36×10^{-1}) −	5.6672×10^{0} (2.72×10^{-1}) −	5.6237×10^{0} (3.36×10^{-1}) −	4.7682×10^{0} (3.18×10^{-1}) −	**4.4482×10^{0}** **(1.99×10^{-1})**
	10	7.4959×10^{0} (2.30×10^{-1}) =	7.5062×10^{0} (2.62×10^{-1}) =	7.5112×10^{0} (3.37×10^{-1}) =	7.6682×10^{0} (3.84×10^{-1}) −	7.5592×10^{0} (3.05×10^{-1}) =	**7.4866×10^{0}** **(3.07×10^{-1})**
	15	1.5541×10^{1} (8.42×10^{-1}) −	1.5833×10^{1} (6.47×10^{-1}) −	1.5868×10^{1} (7.99×10^{-1}) −	1.6171×10^{1} (7.16×10^{-1}) −	1.4975×10^{1} (7.09×10^{-1}) −	**1.4041×10^{1}** **(5.77×10^{-1})**
WFG7	4	1.1391×10^{0} (6.44×10^{-2}) −	1.0519×10^{0} (6.12×10^{-2}) −	1.4513×10^{0} (2.11×10^{-1}) −	1.1253×10^{0} (5.14×10^{-2}) −	1.0057×10^{0} (7.45×10^{-2}) −	**9.3171×10^{-1}** **(4.16×10^{-2})**
	6	3.0335×10^{0} (2.01×10^{-1}) −	2.8505×10^{0} (1.58×10^{-1}) −	3.4183×10^{0} (2.34×10^{-1}) −	3.2046×10^{0} (2.32×10^{-1}) −	2.6170×10^{0} (1.81×10^{-1}) −	**2.3913×10^{0}** **(1.49×10^{-1})**
	8	5.4840×10^{0} (3.35×10^{-1}) −	5.2294×10^{0} (3.19×10^{-1}) −	5.8570×10^{0} (3.46×10^{-1}) −	5.8238×10^{0} (4.26×10^{-1}) −	5.0230×10^{0} (3.23×10^{-1}) −	**4.6235×10^{0}** **(2.15×10^{-1})**
	10	8.0380×10^{0} (3.84×10^{-1}) =	8.0994×10^{0} (3.17×10^{-1}) =	8.1133×10^{0} (4.20×10^{-1}) =	8.1266×10^{0} (3.52×10^{-1}) =	8.0862×10^{0} (3.99×10^{-1}) =	**7.9913×10^{0}** **(3.60×10^{-1})**
	15	1.6222×10^{1} (7.83×10^{-1}) −	1.6131×10^{1} (6.31×10^{-1}) −	1.6466×10^{1} (9.98×10^{-1}) −	1.7083×10^{1} (6.68×10^{-1}) −	1.5289×10^{1} (7.77×10^{-1}) −	**1.4755×10^{1}** **(6.68×10^{-1})**

(To be continued)

Table 4 IGD values of six algorithms for different objectives in the WFG test problems. (Continued)

Problem	M	IGD value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
WFG8	4	1.3602×10^{0} (4.34×10^{-2}) −	1.3191×10^{0} (4.73×10^{-2}) −	1.6783×10^{0} (1.20×10^{-1}) −	1.4033×10^{0} (6.55×10^{-2}) −	1.3062×10^{0} (5.37×10^{-2}) −	**1.2182×10^{0}** **(5.06×10^{-2})**
	6	3.1347×10^{0} (1.80×10^{-1}) −	3.0351×10^{0} (1.31×10^{-1}) −	3.5691×10^{0} (2.42×10^{-1}) −	3.3104×10^{0} (1.51×10^{-1}) −	2.8526×10^{0} (1.67×10^{-1}) −	**2.6735×10^{0}** **(1.10×10^{-1})**
	8	5.4695×10^{0} (2.57×10^{-1}) −	5.2240×10^{0} (1.88×10^{-1}) −	5.9009×10^{0} (3.95×10^{-1}) −	5.8838×10^{0} (3.13×10^{-1}) −	5.1078×10^{0} (2.78×10^{-1}) −	**4.7267×10^{0}** **(1.94×10^{-1})**
	10	7.7556×10^{0} (2.50×10^{-1}) =	7.7605×10^{0} (3.37×10^{-1}) =	7.9148×10^{0} (3.35×10^{-1}) =	**7.7397×10^{0}** **(3.23×10^{-1}) =**	7.7631×10^{0} (3.05×10^{-1}) =	7.7501×10^{0} (3.06×10^{-1})
	15	1.5751×10^{1} (6.14×10^{-1}) −	1.5860×10^{1} (5.12×10^{-1}) −	1.6241×10^{1} (8.39×10^{-1}) −	1.6671×10^{1} (8.24×10^{-1}) −	1.5230×10^{1} (8.01×10^{-1}) −	**1.4330×10^{1}** **(6.44×10^{-1})**
WFG9	4	1.2771×10^{0} (6.50×10^{-2}) −	1.2225×10^{0} (6.66×10^{-2}) −	1.5791×10^{0} (1.05×10^{-1}) −	1.3426×10^{0} (7.02×10^{-2}) −	1.1065×10^{0} (7.48×10^{-2}) −	**1.0018×10^{0}** **(4.89×10^{-2})**
	6	3.1110×10^{0} (1.82×10^{-1}) −	3.0923×10^{0} (2.10×10^{-1}) −	3.5141×10^{0} (3.75×10^{-1}) −	3.3101×10^{0} (1.79×10^{-1}) −	2.5172×10^{0} (1.51×10^{-1}) −	**2.2930×10^{0}** **(1.06×10^{-1})**
	8	5.5229×10^{0} (3.33×10^{-1}) −	5.3238×10^{0} (2.83×10^{-1}) −	5.6756×10^{0} (3.51×10^{-1}) −	5.9342×10^{0} (3.48×10^{-1}) −	4.4527×10^{0} (2.51×10^{-1}) −	**4.2249×10^{0}** **(1.89×10^{-1})**
	10	8.0942×10^{0} (2.79×10^{-1}) =	8.0332×10^{0} (3.66×10^{-1}) =	8.0114×10^{0} (3.93×10^{-1}) =	8.0706×10^{0} (3.09×10^{-1}) =	7.9967×10^{0} (4.35×10^{-1}) =	**7.9617×10^{0}** **(4.08×10^{-1})**
	15	1.6040×10^{1} (6.28×10^{-1}) −	1.5619×10^{1} (8.42×10^{-1}) −	1.5887×10^{1} (1.09×10^{0}) −	1.6794×10^{1} (6.36×10^{-1}) −	1.3647×10^{1} (7.31×10^{-1}) −	**1.2918×10^{1}** **(5.12×10^{-1})**
+/−/=		1/30/14	8/29/8	6/30/9	4/32/9	9/24/12	

Note: "+", "−", and "=" denote that the results from other algorithms are higher than, lower than, or equal to the results from the proposed ImMAPIO, respectively. The bold parts represent the best results. The value before the brackets represents the mean value of the indicator used to measure the algorithm, and the value in the brackets refers to the standard deviation of the indicator.

Table 5 HV values of these algorithms for different objectives in the WFG test problems.

Problem	M	HV value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
WFG1	4	3.5784×10^{1} (1.85×10^{1}) +	9.0046×10^{1} (2.00×10^{1}) +	9.2829×10^{1} (2.28×10^{1}) +	**1.1726×10^{2}** **(1.70×10^{1}) +**	6.3304×10^{1} (1.74×10^{1}) +	2.0385×10^{1} (1.26×10^{1})
	6	1.1416×10^{4} (1.86×10^{3}) +	**1.6261×10^{4}** **(1.76×10^{3}) +**	1.3574×10^{4} (1.35×10^{3}) +	1.4804×10^{4} (2.12×10^{3}) +	1.4192×10^{4} (1.52×10^{3}) +	8.2635×10^{3} (2.81×10^{3})
	8	3.6132×10^{6} (3.59×10^{5}) +	**4.5193×10^{6}** **(1.58×10^{5}) +**	3.5048×10^{6} (2.91×10^{5}) +	4.0264×10^{6} (4.85×10^{5}) +	4.0921×10^{6} (3.18×10^{5}) +	2.9339×10^{6} (9.62×10^{5})
	10	1.4806×10^{9} (1.06×10^{8}) +	1.6884×10^{9} (6.38×10^{7}) +	1.3266×10^{9} (1.38×10^{8}) +	1.3983×10^{9} (1.70×10^{8}) +	**1.7054×10^{9}** **(5.03×10^{7}) +**	9.0170×10^{8} (4.47×10^{8})
	15	2.4364×10^{16} (6.79×10^{14}) +	**2.5234×10^{16}** **(2.39×10^{14}) +**	1.8951×10^{16} (2.23×10^{15}) +	2.3302×10^{16} (2.09×10^{15}) +	2.0558×10^{16} (5.80×10^{15}) +	9.9738×10^{15} (3.37×10^{15})
WFG2	4	3.8408×10^{2} (1.20×10^{1}) −	3.9749×10^{2} (1.17×10^{1}) −	3.4158×10^{2} (2.62×10^{1}) −	3.8459×10^{2} (1.84×10^{1}) −	**4.3437×10^{2}** **(1.16×10^{1}) +**	4.1143×10^{2} (1.02×10^{1})
	6	5.3681×10^{4} (2.12×10^{3}) −	5.6442×10^{4} (1.72×10^{3}) =	4.8562×10^{4} (3.52×10^{3}) −	5.3773×10^{4} (2.71×10^{3}) −	**6.2267×10^{4}** **(2.38×10^{3}) +**	5.5325×10^{4} (2.36×10^{3})
	8	1.3448×10^{7} (5.10×10^{5}) −	1.4286×10^{7} (6.13×10^{5}) =	1.2110×10^{7} (1.31×10^{6}) −	1.3357×10^{7} (7.98×10^{5}) −	**1.5963×10^{7}** **(4.86×10^{5}) +**	1.4333×10^{7} (4.39×10^{5})
	10	5.2616×10^{9} (1.65×10^{8}) −	5.4782×10^{9} (2.41×10^{8}) −	5.0647×10^{9} (5.88×10^{8}) −	5.3277×10^{9} (3.26×10^{8}) −	**6.5392×10^{9}** **(1.61×10^{8}) +**	6.1226×10^{9} (1.49×10^{8})
	15	8.3501×10^{16} (4.60×10^{15}) −	8.9301×10^{16} (5.56×10^{15}) =	7.6236×10^{16} (8.10×10^{15}) −	8.3628×10^{16} (5.65×10^{15}) −	**1.0295×10^{17}** **(3.56×10^{15}) +**	8.9246×10^{16} (3.00×10^{15})
WFG3	4	6.7255×10^{-1} (2.21×10^{-1}) −	6.2814×10^{-1} (2.41×10^{-1}) −	2.6620×10^{-2} (5.10×10^{-2}) −	4.0052×10^{-1} (1.89×10^{-1}) −	9.8456×10^{-1} (2.56×10^{-1}) −	**1.1281×10^{0}** **(2.64×10^{-1})**
	6	**0.0000×10^{0}** **(0.00×10^{0}) =**	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0})
	8	**0.0000×10^{0}** **(0.00×10^{0}) =**	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0})
	10	**0.0000×10^{0}** **(0.00×10^{0}) =**	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0})
	15	**0.0000×10^{0}** **(0.00×10^{0}) =**	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0}) =	0.0000×10^{0} (0.00×10^{0})

(To be continued)

Table 5 HV values of these algorithms for different objectives in the WFG test problems. (Continued)

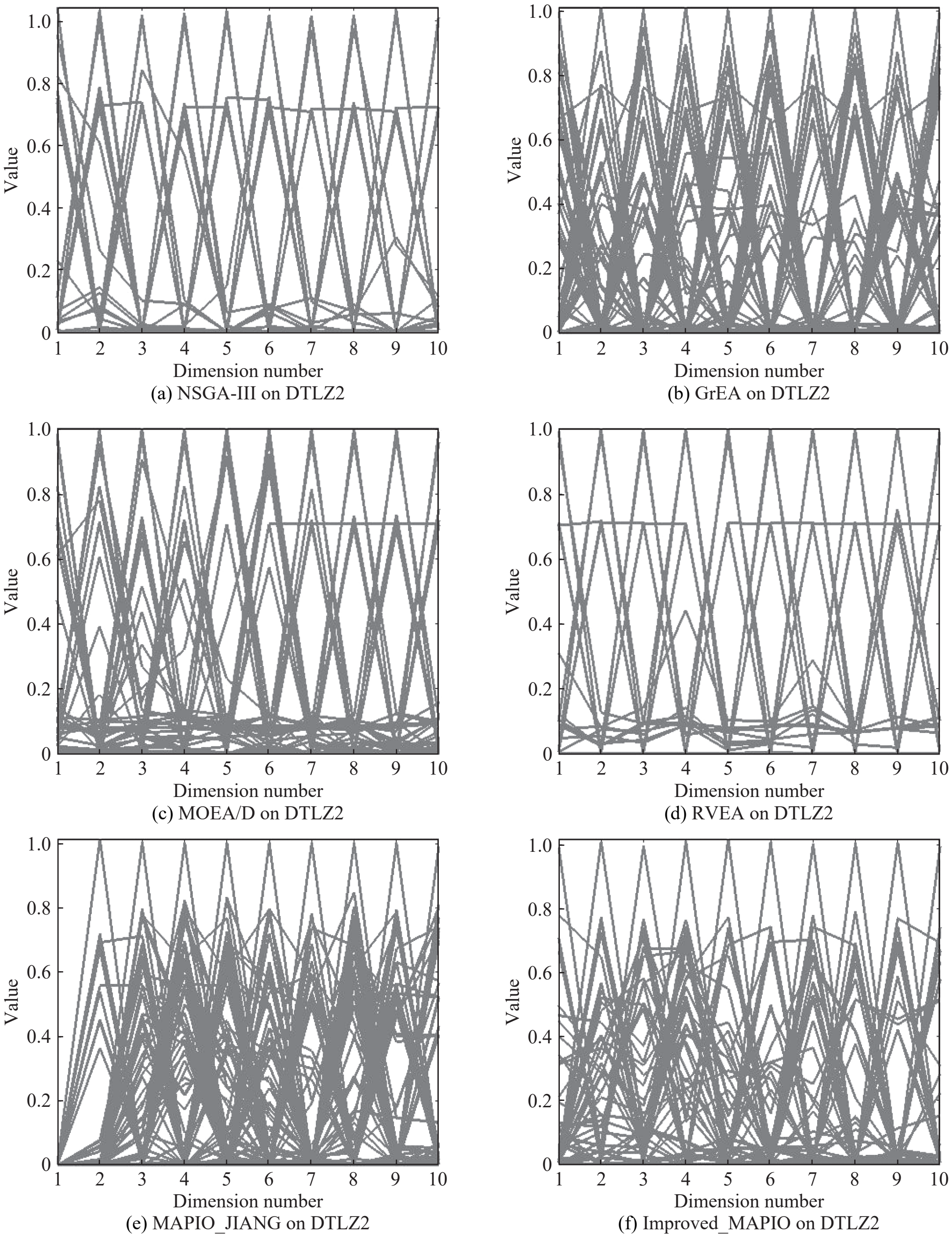
Problem	M	HV value					
		NSGA-III	GrEA	MOEA/D	RVEA	MAPIO	ImMAPIO
WFG4	4	2.0090×10^{2} (7.17×10^{0}) −	2.1494×10^{2} (8.23×10^{0}) −	1.6090×10^{2} (1.96×10^{1}) −	1.9753×10^{2} (1.07×10^{1}) −	2.3180×10^{2} (1.10×10^{1}) −	**2.4395×10^{2}** **(8.49×10^{0})**
	6	2.8430×10^{4} (1.41×10^{3}) −	2.9478×10^{4} (1.13×10^{3}) −	2.1901×10^{4} (1.72×10^{3}) −	2.2975×10^{4} (1.45×10^{3}) −	3.5850×10^{4} (1.77×10^{3}) −	**3.7948×10^{4}** **(1.74×10^{3})**
	8	7.8951×10^{6} (2.75×10^{5}) −	8.2707×10^{6} (3.00×10^{5}) −	6.0156×10^{6} (5.60×10^{5}) −	6.0796×10^{6} (2.62×10^{5}) −	1.0200×10^{7} (4.00×10^{5}) −	**1.0723×10^{7}** **(4.02×10^{5})**
	10	**3.4974×10^{9}** **(1.07×10^{8}) =**	3.4541×10^{9} (1.07×10^{8}) =	3.4326×10^{9} (1.16×10^{8}) =	3.4557×10^{9} (8.69×10^{7}) =	3.4728×10^{9} (9.23×10^{7}) =	3.4472×10^{9} (1.26×10^{8})
	15	6.0675×10^{16} (3.05×10^{15}) −	6.4184×10^{16} (2.64×10^{15}) −	4.6306×10^{16} (5.62×10^{15}) −	4.3322×10^{16} (3.82×10^{15}) −	7.7883×10^{16} (3.94×10^{15}) −	**8.0952×10^{16}** **(3.10×10^{15})**
WFG5	4	1.6439×10^{2} (6.53×10^{0}) −	1.7568×10^{2} (6.79×10^{0}) −	1.0502×10^{2} (1.86×10^{1}) −	1.5588×10^{2} (7.28×10^{0}) −	1.7808×10^{2} (6.93×10^{0}) −	**1.9221×10^{2}** **(5.20×10^{0})**
	6	2.5761×10^{4} (8.79×10^{2}) −	2.6921×10^{4} (8.49×10^{2}) −	1.4994×10^{4} (2.97×10^{3}) −	2.1958×10^{4} (7.74×10^{2}) −	2.9153×10^{4} (1.05×10^{3}) −	**3.0877×10^{4}** **(1.16×10^{3})**
	8	7.2302×10^{6} (2.71×10^{5}) −	7.6177×10^{6} (1.66×10^{5}) −	4.1603×10^{6} (9.29×10^{5}) −	5.8004×10^{6} (2.75×10^{5}) −	8.2373×10^{6} (3.02×10^{5}) −	**8.6090×10^{6}** **(2.83×10^{5})**
	10	3.2628×10^{9} (6.43×10^{7}) −	3.2956×10^{9} (7.80×10^{7}) =	3.2963×10^{9} (8.58×10^{7}) =	3.2922×10^{9} (6.50×10^{7}) =	3.3015×10^{9} (6.51×10^{7}) =	**3.3026×10^{9}** **(7.18×10^{7})**
	15	5.4309×10^{16} (1.58×10^{15}) −	5.6930×10^{16} (2.02×10^{15}) −	3.0704×10^{16} (6.49×10^{15}) −	3.7635×10^{16} (2.52×10^{15}) −	6.3712×10^{16} (2.51×10^{15}) −	**6.6755×10^{16}** **(2.17×10^{15})**
WFG6	4	1.4221×10^{2} (6.04×10^{0}) −	1.4936×10^{2} (6.11×10^{0}) −	8.2764×10^{1} (2.10×10^{1}) −	1.2882×10^{2} (8.17×10^{0}) −	1.6224×10^{2} (7.44×10^{0}) −	**1.7343×10^{2}** **(6.17×10^{0})**
	6	2.2630×10^{4} (1.01×10^{3}) −	2.3855×10^{4} (6.79×10^{2}) −	1.2621×10^{4} (3.04×10^{3}) −	1.8782×10^{4} (1.10×10^{3}) −	2.6545×10^{4} (8.94×10^{2}) −	**2.8445×10^{4}** **(8.91×10^{2})**
	8	6.3329×10^{6} (2.32×10^{5}) −	6.7261×10^{6} (2.09×10^{5}) −	3.7067×10^{6} (6.93×10^{5}) −	5.0147×10^{6} (3.09×10^{5}) −	7.5557×10^{6} (2.38×10^{5}) −	**7.9697×10^{6}** **(2.67×10^{5})**
	10	2.8954×10^{9} (1.04×10^{8}) =	2.8886×10^{9} (6.31×10^{7}) =	2.8932×10^{9} (9.69×10^{7}) =	2.8775×10^{9} (9.96×10^{7}) =	2.8897×10^{9} (9.88×10^{7}) =	**2.9020×10^{9}** **(1.02×10^{8})**
	15	4.7881×10^{16} (1.77×10^{15}) −	4.9365×10^{16} (1.98×10^{15}) −	2.7847×10^{16} (3.99×10^{15}) −	3.3242×10^{16} (2.58×10^{15}) −	5.7045×10^{16} (2.36×10^{15}) −	**6.1368×10^{16}** **(2.21×10^{15})**
WFG7	4	1.8153×10^{2} (5.66×10^{0}) −	1.9349×10^{2} (7.23×10^{0}) −	1.2537×10^{2} (2.25×10^{1}) −	1.7471×10^{2} (6.68×10^{0}) −	2.1006×10^{2} (6.05×10^{0}) −	**2.2167×10^{2}** **(4.70×10^{0})**
	6	2.7572×10^{4} (1.28×10^{3}) −	2.8581×10^{4} (9.34×10^{2}) −	1.7851×10^{4} (2.80×10^{3}) −	2.2908×10^{4} (9.57×10^{2}) −	3.2718×10^{4} (1.20×10^{3}) −	**3.4878×10^{4}** **(1.10×10^{3})**
	8	7.7367×10^{6} (2.74×10^{5}) −	8.0545×10^{6} (2.48×10^{5}) −	4.6446×10^{6} (8.49×10^{5}) −	6.1752×10^{6} (3.61×10^{5}) −	9.0819×10^{6} (3.77×10^{5}) −	**9.6743×10^{6}** **(3.31×10^{5})**
	10	3.3529×10^{9} (1.26×10^{8}) =	3.3313×10^{9} (9.17×10^{7}) −	3.3678×10^{9} (1.07×10^{8}) =	3.3659×10^{9} (9.66×10^{7}) =	3.3725×10^{9} (1.02×10^{8}) =	**3.3945×10^{9}** **(8.94×10^{7})**
	15	5.8204×10^{16} (2.31×10^{15}) −	6.1402×10^{16} (2.61×10^{15}) −	3.8224×10^{16} (7.41×10^{15}) −	4.2571×10^{16} (2.34×10^{15}) −	7.1340×10^{16} (3.73×10^{15}) −	**7.3846×10^{16}** **(2.51×10^{15})**
WFG8	4	1.4275×10^{2} (4.75×10^{0}) −	1.5113×10^{2} (5.78×10^{0}) −	8.1614×10^{1} (1.51×10^{1}) −	1.3064×10^{2} (7.23×10^{0}) −	1.5802×10^{2} (7.83×10^{0}) −	**1.6932×10^{2}** **(6.04×10^{0})**
	6	2.3171×10^{4} (8.35×10^{2}) −	2.4173×10^{4} (9.54×10^{2}) −	1.2178×10^{4} (3.21×10^{3}) −	1.9003×10^{4} (1.33×10^{3}) −	2.6918×10^{4} (1.05×10^{3}) −	**2.8651×10^{4}** **(9.73×10^{2})**
	8	6.7731×10^{6} (2.14×10^{5}) −	7.0675×10^{6} (2.12×10^{5}) −	3.4656×10^{6} (7.54×10^{5}) −	5.1390×10^{6} (3.03×10^{5}) −	7.8044×10^{6} (3.40×10^{5}) −	**8.3515×10^{6}** **(2.46×10^{5})**
	10	**3.1409×10^{9}** **(8.31×10^{7}) =**	3.1283×10^{9} (8.30×10^{7}) =	3.1125×10^{9} (9.62×10^{7}) =	3.1360×10^{9} (9.03×10^{7}) =	3.1194×10^{9} (7.38×10^{7}) =	3.1146×10^{9} (9.06×10^{7})
	15	5.2570×10^{16} (2.44×10^{15}) −	5.5063×10^{16} (2.76×10^{15}) −	2.8691×10^{16} (5.63×10^{15}) −	3.5071×10^{16} (3.14×10^{15}) −	6.3728×10^{16} (2.62×10^{15}) −	**6.7598×10^{16}** **(2.00×10^{15})**
WFG9	4	1.5209×10^{2} (8.27×10^{0}) −	1.5941×10^{2} (1.05×10^{1}) −	8.9853×10^{1} (1.76×10^{1}) −	1.3817×10^{2} (1.13×10^{1}) −	1.8435×10^{2} (1.12×10^{1}) −	**2.0232×10^{2}** **(8.42×10^{0})**
	6	2.2504×10^{4} (1.27×10^{3}) −	2.2896×10^{4} (1.51×10^{3}) −	1.2910×10^{4} (2.91×10^{3}) −	1.8568×10^{4} (1.65×10^{3}) −	2.8914×10^{4} (1.92×10^{3}) −	**3.1961×10^{4}** **(1.79×10^{3})**
	8	6.4729×10^{6} (3.35×10^{5}) −	6.7877×10^{6} (3.83×10^{5}) −	3.4978×10^{6} (9.81×10^{5}) −	4.8800×10^{6} (4.10×10^{5}) −	8.2628×10^{6} (4.38×10^{5}) −	**8.9906×10^{6}** **(3.56×10^{5})**
	10	2.8154×10^{9} (1.42×10^{8}) =	2.7936×10^{9} (1.28×10^{8}) =	2.7844×10^{9} (1.06×10^{8}) =	2.8074×10^{9} (1.24×10^{8}) =	2.8082×10^{9} (1.15×10^{8}) =	**2.8361×10^{9}** **(1.13×10^{8})**
	15	4.8525×10^{16} (2.74×10^{15}) −	5.1927×10^{16} (3.02×10^{15}) −	2.9578×10^{16} (5.41×10^{15}) −	3.1632×10^{16} (2.68×10^{15}) −	6.2604×10^{16} (3.23×10^{15}) −	**6.9644×10^{16}** **(2.61×10^{15})**
+/−/=		5/31/9	5/28/12	5/30/10	5/30/10	10/25/10	

Note: “+”, “−”, and “=” denote that the results from other algorithms are higher than, lower than, or equal to the results from the proposed ImMAPIO, respectively. The bold parts represent the best results. The value before the brackets represents the mean value of the indicator used to measure the algorithm, and the value in the brackets refers to the standard deviation of the indicator.

4.3.1 Results on DTLZ1−DTLZ7

To test the property of ImMAPIO, this study compares the proposed algorithm with the other five many-objective optimization algorithms for the DTLZ[39] test function. The comparison results are evaluated using the IGD indicator. The PFs obtained by these algorithms with ten objectives on the DTLZ2 test function are shown in Fig. 3. The PFs obtained by MOEA/D, NSGA-III, and RVEA have poor distribution. The distributions obtained by GrEA and MAPIO are relatively superior, which indicates the comprehensive performance of these algorithms in terms of diversity and convergence. The distribution performance of ImMAPIO falls between those of GrEA and MAPIO.

Table 3 reveals the results of the comparison of ImMAPIO with five state-of-the-art algorithms, namely, GrEA, NSGA-III, RVEA, MOEA/D, and MAPIO, on the DTLZ test functions using the IGD as the indicator[45]. As observed from the last row in



(a) NSGA-III on DTLZ2

(b) GrEA on DTLZ2

(c) MOEA/D on DTLZ2

(d) RVEA on DTLZ2

(e) MAPIO_JIANG on DTLZ2

(f) Improved_MAPIO on DTLZ2

Fig. 3 Pareto front obtained by different algorithms on the 10-objective DTLZ2 test function.

Table 3, relative to NSGA-III and GrEA, ImMAPIO has a significant advantage on the 40 test functions. The performance of NSGA-III is weaker than that of the others on any function, and GrEA only has one best result relative to the proposed algorithm. In the comparison of MOEA/D and RVEA, 14 items of MOEA/D and 8 items of RVEA are better than those of the proposed algorithm.

MOEA/D performs better than the other algorithms on the DTLZ1, DTLZ3, and DTLZ7 test functions because of the multiple modal features of the test functions and the difficulty of obtaining convergence solutions. Relative to MAPIO, the proposed algorithm has a slight advantage as 9 of its items are worse than those of MAPIO; however, MAPIO has 11 items that are better than those of the proposed algorithm. The highlighted region shows the superior performance of the ImMAPIO on DTLZ2, DTLZ4, and DTLZ8. The PF of DTLZ2 is concave. These comparison results confirm the effectiveness of ImMAPIO.

4.3.2 Results on WFG1−WFG9

To test the property of the proposed algorithm, this study compares the ImMAPIO algorithm with five many-objective optimization algorithms on the WFG[40] test function. The comparison results are evaluated by IGD[45] and HV[46].

Table 4 reveals the comparison results for ImMAPIO and five of the most advanced algorithms, namely, NSGA-III, MOEA/D, RVEA, GrEA, and MAPIO, on the basis of the WFG test functions using the indicator of IGD. As noticed from the last row of Table 4, relative to NSGA-III, ImMAPIO has a significant advantage over the 45 test functions. Meanwhile, NSGA-III only has one best result relative to the proposed algorithm. Specifically, the use of the Pareto dominance relationship and reference point strategy lack selection pressure and can thus lead to poor convergence. The comparison of the proposed ImMAPIO with other methods shows that 8, 6, and 4 items of GrEA, MOEA/D, and RVEA, respectively, exceed those of ImMAPIO. The proposed algorithm has a slight advantage because the proposed algorithm and MAPIO adopt the BFE strategy, which has an excellent performance in terms of the convergence and diversity of the population.

Although the proposed algorithm's performance does not exceed that of other algorithms for the WFG1 and WFG2 test problems, it performs better than the other algorithms on the whole test functions, especially on the WFG6, WFG7, and WFG9 test functions. The HV-based selection strategy in ImMAPIO can measure multiformity and diversity and improve the selection accuracy. The knee point based selection strategy can improve the convergence of the population. The vector angle based selection strategy can ensure wideness and uniformity.

Table 5 displays the comparison of the ImMAPIO algorithm with the other five algorithms on the WFG test functions with 4, 6, 8, 10, and 15 objectives on the basis of the indicator HV[46]. By using the same analysis method, NSGA-III, GrEA, MOEA/D, RVEA, and MAPIO are compared with the ImMAPIO algorithm, which shows obvious significant advantages. The property of GrEA exceeds those of the other algorithms on WFG1 because the grid dominance in GrEA can provide additional selection. The performance of MAPIO on the WFG2 test function is obviously excellent.

Meanwhile, the performance of the proposed algorithm is superior to that of the other algorithms on the WFG4–WFG9 test functions. The numerical analysis shows that for the 45 comparison results, 31, 28, 30, and 25 items of ImMAPIO are superior to those of NSGA-III, GrEA, RVEA, MOEA/D, and MAPIO, respectively. The highlighted region shows the superior property of the proposed algorithm on WFG4–WFG9. These comparison results confirm the superiority of the proposed algorithm on the WFG test functions.

4.3.3 Wilcoxon signed-rank test of the comparison results

The Wilcoxon signed-rank test, also known as the signed-rank sum test, is based on the assumption that the two treatment effects of a pairing are the same. If the value of the significance difference is less than 0.05, then the null hypothesis is rejected, and a significant difference between the matched samples is considered. The Wilcoxon signed-rank test is used in this work to inspect the differences in the IGD indicators between the proposed ImMAPIO algorithm and the other five algorithms for the WFG test functions.

As shown in Table 6, the value of the significant difference is significantly less than 0.5, and some values are even 0. In the case in which the comparison results of the proposed algorithm are better than those of the other algorithms, significant differences are noted in the compared data. Hence, the proposed algorithm is significantly better than the other algorithms.

Table 6 Inspection results about the comparison results between the ImMAPIO and five algorithms.

Algorithm	Progressive significance (double tail)
ImMAPIO-NSGA-III	0.000
ImMAPIO-GrEA	0.000
ImMAPIO-MOEA/D	0.000
ImMAPIO-RVEA	0.000
ImMAPIO-MAPIO	0.005

Note: Wilcoxon signed-rank test, also known as the signed-rank sum test, is based on the assumption that the two treatment effects of a pairing are the same. It is used to judge whether there is significant difference between matched samples.

5 Conclusion

ImMAPIO is proposed to solve many-objective optimization problems. Then, an integration-based selection strategy that integrates three strategies and their advantages is proposed. The advantages of each strategy are combined while the convergence and multiformity of solution sets are guaranteed. By using the DTLZ and WFG test functions, the performance of ImMAPIO is verified and compared with that of five many-objective optimization evolutionary algorithms. The comparison and the Wilcoxon signed-rank test results show the advantages of the ImMAPIO algorithm. In our future work, we will solve practical engineering problems by using the proposed algorithm.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2018YFC1604000), the National Natural Science Foundation of China (Nos. 61806138, 61772478, U1636220, 61961160707, and 61976212), the Key R&D Program of Shanxi Province (High Technology) (No. 201903D121119), the Key R&D Program of Shanxi Province (International Cooperation) (No. 201903D421048), and the Key R&D Program (International Science and Technology Cooperation Project) of Shanxi Province, China (No. 201903D421003).

References

[1] R. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in *Proc. the 6th Int. Symp. on Micro Machine and Human Science*, Nagoya, Japan, 2002, pp. 39–43.

[2] Z. H. Cui, J. J. Zhang, D. Wu, X. J. Cai, H. Wang, W. S. Zhang, and J. J. Chen, Hybrid many-objective particle swarm optimization algorithm for green coal production problem, *Informat. Sci.*, vol. 518, pp. 256–271, 2020.

[3] X. S. Yang, A new metaheuristic bat-inspired algorithm, in *Nature Inspired Cooperative Strategies for Optimization* (*NICSO 2010*), J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor, eds. Berlin, Germany: Springer, vol. 284, pp. 65–74, 2010.

[4] H. T. Rauf, M. Hadi, and A. Rehman, Bat algorithm with Weibull walk for solving global optimisation and classification problems, *Int. J. Bio-Inspir. Comput.*, vol. 15, no. 3, pp. 159–170, 2020.

[5] M. Dorigo, V. Maniezzo, and A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, 1996.

[6] X. S. Yang and S. Deb, Cuckoo search via Lévy flights, presented at 2009 World Congress on Nature&Biologically Inspired Computing (NaBIC), Coimbatore, India, 2010, pp. 210–214.

[7] Z. H. Cui, M. Q. Zhang, H. Wang, X. J. Cai, W. S. Zhang, and J. J. Chen, Hybrid many-objective cuckoo search algorithm with Lévy and exponential distributions, *Memetic Comput.*, vol. 12, no. 3, pp. 251–265, 2020.

[8] X. S. Yang, Firefly algorithm, in *Engineering Optimization: An Introduction with Metaheuristic Applications*, X. S. Yang, ed. Hoboken, NJ, USA: John Wiley&Sons, Inc., 2010, pp. 221–230.

[9] J. Zhao, J. J. Tang, A. Y. Shi, T. H. Fan, and L. Z. Xu, Improved density peaks clustering based on firefly algorithm, *Int. J. Bio-Inspir. Comput.*, vol. 15, no. 1, pp. 24–42, 2020.

[10] D. Karaboga, Artificial bee colony algorithm, *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010.

[11] Y. N. Guo, H. Yang, M. R. Chen, J. Cheng, and D. W. Gong, Ensemble prediction-based dynamic robust multi-objective optimization methods, *Swarm Evol. Comput.*, vol. 48, pp. 156–171, 2019.

[12] Y. N. Guo, J. Cheng, S. Luo, D. W. Gong, and Y. Xue, Robust dynamic multi-objective vehicle routing optimization method, *IEEE-ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1891–1903, 2018.

[13] H. B. Duan and P. X. Qiao, Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning, *Int. J. Intell. Comput. Cybern.*, vol. 7, no. 1, pp. 24–37, 2014.

[14] Y. N. Guo, X. Zhang, D. W. Gong, Z. Zhang, and J. J. Yang, Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks, *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 750–764, 2020.

[15] Y. N. Guo, P. Zhang, J. Cheng, C. Wang, and D. W. Gong, Interval multi-objective quantum-inspired cultural algorithms, *Neural Comput. Appl.*, vol. 30, no. 3, pp. 709–722, 2018.

[16] Z. H. Cui, Y. R. Zhao, Y. Cao, X. J. Cai, W. S. Zhang, and J. J. Chen, Malicious code detection under 5G HetNets based on a multi-objective RBM model, *IEEE Netw.*, vol. 35, no. 2, pp. 82–87, 2021.

[17] H. B. Duan, J. X. Zhao, Y. M. Deng, Y. H. Shi, and X. L. Ding, Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning, *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 1, pp. 706–720, 2021.

[18] H. X. Qiu and H. B. Duan, A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles, *Inform. Sci.*, vol. 509, pp. 515–529, 2020.

[19] Z. H. Cui, J. J. Zhang, Y. C. Wang, Y. Cao, X. J. Cai, W. S. Zhang, and J. J. Chen, A pigeon-inspired optimization algorithm for many-objective optimization problems, (in Chinese), *Sci. China Inf. Sci.*, vol. 62, no. 7, p. 70212, 2019.

[20] Z. H. Cui, X. H. Xu, F. Xue, X. J. Cai, Y. Cao, W. S. Zhang, and J. J. Chen, Personalized recommendation system based on collaborative filtering for IoT scenarios, *IEEE Trans. Serv. Comput.*, vol. 13, no. 4, pp. 685–695, 2020.

[21] Z. H. Cui, F. Xue, S. Q. Zhang, X. J. Cai, Y. Cao, W. S. Zhang, and J. J. Chen, A hybrid Block Chain-based identity authentication scheme for Multi-WSN, *IEEE Trans. Serv. Comput.*, vol. 13, no. 2, pp. 241–251, 2020.

[22] X. J. Cai, S. J. Geng, D. Wu, J. H. Cai, and J. J. Chen, A multicloud-model-based many-objective intelligent algorithm for efficient task scheduling in internet of things, *IEEE Int. Things J.*, vol. 8, no. 12, pp. 9645–9653, 2021.

[23] C. Li and H. B. Duan, Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function, *Aerosp. Sci. Technol.*, vol. 39, pp. 352–360, 2014.

[24] H. B. Duan, H. X. Qiu, and Y. M. Fan, Unmanned aerial vehicle close formation cooperative control based on predatory escaping pigeon-inspired optimization, (in Chinese), *Sci. Sin. Tech.*, vol. 45, no. 6, pp. 559–572, 2015.

[25] B. Zhang and H. B. Duan, Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 97–107, 2017.

[26] H. Arshad, S. Batool, Z. Amjad, M. Ali, S. Aimal, and N. Javaid, Pigeon inspired optimization and enhanced differential evolution using time of use tariff in smart grid, presented at the 9th Int. Conf. Intelligent Networking and Collaborative Systems (INCoS 2017), Tronto, Canada, 2017, pp. 563–575.

[27] X. X. Sun, J. S. Pan, S. C. Chu, P. Hu, and A. Q. Tian, A novel pigeon-inspired optimization with QUasi-Affine TRansformation evolutionary algorithm for DV-Hop in wireless sensor networks, *Int. J. Distrib. Sens. Netw.* vol. 16, no. 6, p. 155014772093274, 2020.

[28] H. X. Qiu and H. B. Duan, Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design, (in Chinese), *Sci. China Tech. Sci.*, vol. 58, no. 11, pp. 1915–1923, 2015.

[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[30] H. B. Duan, M. Z. Huo, and Y. H. Shi, Limit-cycle-based mutant multiobjective pigeon-inspired optimization, *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 948–959, 2020.

[31] Y. Hu, J. Wang, J. Liang, K. J. Yu, H. Song, Q. Q. Guo, C. T. Yue, and Y. L. Wang, A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm, (in Chinese), *Sci. China Inf. Sci.*, vol. 62, no. 7, p. 70206, 2019.

[32] Q. Z. Lin, S. B. Liu, Q. L. Zhu, C. Y. Tang, R. Z. Song, J. Y. Chen, C. A. C. Coello, K. C. Wong, and J. Zhang, Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems, *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, 2018.

[33] M. A. Dulebenets, Archived elitism in evolutionary computation: Towards improving solution quality and population diversity, *Int. J. Bio-Inspir. Comput.*, vol. 15, no. 3, pp. 135–146, 2020.

[34] J. Bader and E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization, *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[35] X. Y. Zhang, Y. Tian, and Y. C. Jin, A knee point-driven evolutionary algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, 2015.

[36] Y. Xiang, Y. R. Zhou, M. Q. Li, and Z. F. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, 2017.

[37] K. Li, J. H. Zheng, M. Q. Li, C. Zhou, and H. Lv, A novel algorithm for non-dominated hypervolume-based multiobjective optimization, presented at 2009 IEEE Int. Conf. Systems, Man and Cybernetics (SMC'2009), San Antonio, TX, USA, 2009, pp. 5220–5226.

[38] Q. Z. Lin, J. Q. Li, Z. H. Du, J. Y. Chen and Z. Ming, A novel multi-objective particle swarm optimization with multiple search strategies, *Eur. J. Oper. Res.*, vol. 247, no. 3, pp. 732–744, 2015.

[39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable multi-objective optimization test problems, in *Proc.* 2002 *Congress on Evolutionary Computation*, Honolulu, HI, USA, 2002, pp. 825–830.

[40] S. Huband, P. Hingston, L. Barone, and L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.

[41] 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 Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[42] S. X. Yang, M. Q. Li, X. H. Liu, and J. H. Zheng, A grid-based evolutionary algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, 2013.

[43] Q. F. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[44] R. Cheng, Y. C. Jin, M. Olhofer, and B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.

[45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.

[46] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.

Zhihua Cui received the PhD degree in control theory and engineering from Xi’an Jiaotong University, Xi’an, China, in 2008. He is a professor of the School of Computer Science and Technology, and the director of Complex System and Computational Intelligence Laboratory at Taiyuan University of Science and Technology, Taiyuan, China. He is the editor-in-chief of *International Journal of Bio-Inspired Computation*. His research interest includes computational intelligence, stochastic algorithm, and combinatorial optimization.

Lihong Zhao received the BS degree in engineering from Taiyuan University of Science and Technology, Taiyuan, China, in 2017. She is currently pursuing the MS degree at the School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China. Her main research includes computational intelligence and combinatorial optimization.

Xiao-Zhi Gao received the BS and MS degrees from Harbin Institute of Technology, Harbin, China, in 1993 and 1996, respectively. He obtained the DSc. (Tech.) degree from Helsinki University of Technology (now Aalto University), Finland, in 1999. He has been working as a professor at the University of Eastern Finland, Kuopio, Finland since 2018. He has published more than 400 technical papers in refereed journals and international conferences. His current Google Scholar H-index is 34. His research interests are nature-inspired computing methods with their applications in optimization, data mining, machine learning, control, signal processing, and industrial electronics.

Yeqing Ren received the MS degree from Taiyuan University of Science and Technology, Taiyuan, China, in 2019. She is currently pursuing the PhD degree at Beijing University of Posts and Telecommunications, Beijing, China. Her research interest includes computational intelligence, image recognition, and network security.

Wensheng Zhang received the PhD degree in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, in 2000. He joined the Institute of Software, CAS, Beijing, China, in 2001. He is a professor of machine learning and data mining and the director of the Research and Development Department, Institute of Automation, CAS, Beijing, China. His research interests include computer vision, pattern recognition, artificial intelligence, and computer human interaction.

Youqian Zeng received the BS degree in engineering from Taiyuan University of Science and Technology, Taiyuan, China, in 2018. He is currently pursuing the MS degree at the School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China. His main research includes computational intelligence and combinatorial optimization.