



A competitive swarm optimizer with probabilistic criteria for many-objective optimization problems

Chao He¹ · Ming Li^{1,2} · Congxuan Zhang³ · Hao Chen² · Xin Li¹ · Junhua Li²

Received: 6 April 2021 / Accepted: 5 March 2022 / Published online: 18 April 2022
© The Author(s) 2022

Abstract

Although multiobjective particle swarm optimizers (MOPSOs) have performed well on multiobjective optimization problems (MOPs) in recent years, there are still several noticeable challenges. For example, the traditional particle swarm optimizers are incapable of correctly discriminating between the personal and global best particles in MOPs, possibly leading to the MOPSOs lacking sufficient selection pressure toward the true Pareto front (PF). In addition, some particles will be far from the PF after updating, this may lead to invalid search and weaken the convergence efficiency. To address the abovementioned issues, we propose a competitive swarm optimizer with probabilistic criteria for many-objective optimization problems (MaOPs). First, we exploit a probability estimation method to select the leaders via the probability space, which ensures the search direction to be correct. Second, we design a novel competition mechanism that uses winner pool instead of the global and personal best particles to guide the entire population toward the true PF. Third, we construct an environment selection scheme with the mixed probability criterion to maintain population diversity. Finally, we present a swarm update strategy to ensure that the next generation particles are valid and the invalid search is avoided. We employ various benchmark problems with 3–15 objectives to conduct a comprehensive comparison between the presented method and several state-of-the-art approaches. The comparison results demonstrate that the proposed method performs well in terms of searching efficiency and population diversity, and especially shows promising potential for large-scale multiobjective optimization problems.

Keywords Many-objective optimization problems (MaOPs) · Competitive swarm optimizer (CSO) · Probability estimation method · Swarm update strategy

Introduction

In many real-world industrial applications, we often face complex decision-making problems that need to optimize several (often conflicting) objectives simultaneously. These problems are called multiobjective optimization problems (MOPs). Let Ω denote the decision space (which refers to a feasible search space), the notation $x = (x_1, x_2, \dots, x_n)$ indicate the decision vector, and f represent the objective

vector. An unconstrained minimization MOP can be defined as follows:

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ &\text{subject to } x \in \Omega, \end{aligned} \quad (1)$$

where $\Omega F : \Omega \rightarrow R^m$ represents the objective function, and m is the number of objectives. Being different from the single-objective problem (SOP), the natures of the multiple objectives are conflicting. Thus, the MOP usually obtains a set of optimal solutions called the Pareto optimal set (PS), and the objective vectors corresponding to PS are named the Pareto front (PF). When MOPs have more than three objectives, they are often called many-objective optimization problems (MaOPs).

MOPs widely exist in many practical applications, such as short-term wind forecast [1], autonomous control of unmanned aerial vehicles (UAVs) [2], aero-engine calibration [3], and optimization of deep neural networks [4].

✉ Ming Li
liming@nchu.edu.cn

¹ College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

² School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China

³ Key Laboratory of Nondestructive Testing, Ministry of Education, Nanchang Hangkong University, Nanchang 330063, China

It is difficult to solve MOPs using conventional mathematical tools, but due to good parallelism, evolutionary algorithms are very suitable for MOPs [5]. As a classic heuristic technique, evolutionary algorithms (EAs) have been demonstrated as a powerful framework for solving MOPs and MaOPs, which have been studied intensively for many years. According to different calculation strategies, they can be roughly divided into three categories. The first category is the multiobjective evolutionary algorithms (MOEAs) based on modified Pareto-dominance, such as the evolutionary algorithm based on grid dominance [6], preference order ranking [7], and other new dominance relations [8]. These improved dominance ranking methods significantly increase the selection pressure in non-dominated solutions and improve the efficiency of searching the true Pareto front. The second category is the indicator-based MOEAs which replaces the Pareto-dominance relation by performance indicators of solution quality measurement. SMS-EMOA [9] and IBEA [10] are two representative algorithms in this category. The third category is the decomposition-based MOEAs. The original MOP is decomposed into multiple subproblems, and these subproblems are solved in a collaborative manner through a population-based search [11]. Representatives of this type include the EA based on a localized weighted sum [12], the constrained decomposition approach with grids for the MOEA [13], the reference vector-guided EA (RVEA) [14] and the MOEA/D with an Ensemble of Neighborhood Sizes (ENS-MOEA/D) [15]. The existing MOEA model is very effective for solving MOPs with two or three objectives. However, it has been found in practice that the performance of these MOEAs is severely declined when solving MaOPs. The main reason of these poor performances is that most of the generated solutions are mutually nondominated as the number of objectives increases, leading to the MOEAs' lack of selection pressure toward the true PF [16]–[18].

As another branch of heuristic technique, the swarm intelligence algorithms also have been designed for solving MOPs and MaOPs. Some literatures have shown that particle swarm optimization (PSO), inspired by the social behavior of birds, is a potential competitor of the genetic algorithm (GA) [19]. Although it cannot be concluded that the performance of PSO on MOPs is better than GA, PSO has the advantages of easier implementation, efficient storage and effective maintenance of solution diversity [20]–[22]. PSO has the characteristic of fast convergence in single-objective optimization problems (SOP), which is based on the premise that the personal- and global-best particles can be clearly confirmed. For example, AWPSO is a novel PSO algorithm, which efficiently solves the SOPs by a sigmoid function and clearly confirmed optimal particles [23]. However, multiobjective particle swarm optimizers (MOPSOs) do not have any particles on MaOPs that can perform best on all objectives and they are usually

replaced by a set of tradeoff solutions. Most of the generated particles are mutually nondominated on MaOPs, which makes it difficult to choose the personal- and global-best particles. Since the personal- and global-best particles are used to guide the search direction of the particle swarm, they have a considerable impact on the performance of the PSO algorithm. Therefore, how to define the personal- and global-best particles has become the most important issue that MOPSOs need to solve. Many PSO methods try to solve the above problem. The first method uses the Pareto-based ranking scheme to define the personal- and global-best particles. CPSO [24], OMOPSO [25] and SMPSO [26] are three typical algorithms. They usually choose the less crowded solutions in the nondominated solutions as the global optimal particle. In addition, MOPSOs based on enhanced ranking schemes have been proposed, such as global marginal ranking [27] and preference order ranking [28]. In these algorithms, the particles in the external archive are first sorted according to the corresponding criteria, and then some elite particles can be selected as candidates for personal- and global-best particles according to their rankings. The second strategy is decomposition-based MOPSOs, where an original MOP is decomposed into a number of single-objective optimization problems where the single-objective PSO algorithms can be directly applied. For example, in AgMOPSO [29], a novel decomposition approach is used to select the personal-best particle and global-best particle during the evolutionary search process. In HMOPSO-ARA, the position information of local best particle is introduced by the decomposition method to improve search efficiency of PSO [30].

Recently, the competitive swarm optimizer (CSO) have been increasingly popular due to its benefits of high search efficiency in solving MOPs. CSO is a variant of PSO and adopts a pairwise competition strategy to update the population. Cheng et al. were the first to introduce a competition mechanism into PSO and applied it to solve SOPs [31]. In their method, the dynamic system is driven by a random competition mechanism, where any particle could be a potential leader. Afterwards, Zhang proposed a competitive mechanism based multi-objective PSO algorithm (CMOPSO), which effectively enhance swarm diversity for tackling MOPs with multimodal landscapes [32]. Moreover, the large-scale multiobjective optimization based on CSO (LMOCSSO) is proposed for solving large-scale multiobjective optimization problems (large-scale MOPs). Different from the existing algorithms that focus on the modification of updating velocity, LMOCSSO adopts a two-stage strategy to update the position of the particles, which can significantly improve the search efficiency [20]. In addition, some new particle update strategies recently proposed also provide research ideas for CSO [33].

The PSOs and CSOs have good performances on different types of benchmark MOPs and MaOPs, but several noticeable problems still remain. First, due to the fact that there is no such a particle that can perform best on all objectives, the MOPSOs are incapable of clearly determining the swarm leaders in solving MaOPs [34]. Second, PSO has the characteristic of fast convergence in SOP, which is based on the premise that search direction can be clearly confirmed [21]. However, in MOPSOs, several objectives need to be considered. Third, after the positions of the particles are updated, some of them are performing invalid searches. Moreover, similar to the problems that MOPSOs face, traditional CSOs also perform poorly on MaOPs, and the performance of CSO degrades dramatically with the increase in the number of objectives in the MaOPs.

To address the abovementioned issues, we propose a competitive swarm optimizer with probabilistic criteria termed MOC SOP. The main new contributions of this paper are summarized as follows.

To clearly define the leader of swarm in solving MaOPs, we propose a probability criterion to estimate the quality of each particle in the population and select the leaders according to the value of joint probability. The proposed probability estimation method shows good robustness as its performance is not affected by the number and range of objectives.

To address the issue of low convergence efficiency of the MOPSOs on some MaOPs, we design a competitive mechanism with winner pool to update the particles' position. Compared with the velocity and updated position strategy of the existing MOPSOs, the learning strategy, based on the competitive mechanism with winner pool, achieves a better performance of convergence on MaOPs.

To enable the swarm to be evenly distributed at the PF, we construct an environment selection scheme with the mixed probability criterion. The proposed mixed probability criterion based on diversity mechanism not only effectively develops the diversity of the population, but also strengthens the selection pressure to some extent in the early stage. After the position updating of the particles, a number of invalid particles may be generated. Some particles even move away from the PF, which will decrease the convergence efficiency. To address the abovementioned issue, we propose a swarm update strategy by using the particles in the external elite archive to update the current particle swarm. This ensures that all of the particles entering the next generation are valid, which effectively improves the convergence of the algorithm.

The rest of this article is organized as follows. The second section introduces the relevant background of MOPSO and the motivation of this article. Details of the MOC SOP are

given in Sect. “Proposed algorithm”. In Sect. “Experimental results and analysis”, some experimental studies are carried out to elaborate on the performance of the MOC SOP algorithm in detail. Finally, Sect. 5 provides our conclusions and some possible approaches for future work.

Related background and motivations

PSO has been widely used in SOPs [31] and other applications [4]. Recent reports show that PSO is a powerful potential competitor of GA in solving MOPs, and many MOPSOs have been successfully applied to MOPs. Despite the fact that MOPSO is very effective in solving MOPs with two or three objectives [32, 26], most of the existing MOPSOs still perform poorly on MaOPs. There are several significant challenges that restrict the performance of MOPSOs.

(1) Swarm leader.

MOPSOs do not have any particles in MOPs that can perform best on all objectives and are usually replaced by a set of tradeoff solutions. This makes it difficult to choose the swarm leader [34]. Since swarm leader particles are used to guide the search direction of the particle swarm, they have a considerable impact on the performance of the PSO algorithm. Especially when solving high-dimensional MaOPs, particles will oscillate repeatedly in the objective space, which will affect the convergence speed [20]. Therefore, how to define swarm leader has become the most important issue that MOPSOs need to solve. In the current studies, some novel leader selection strategies have been proposed [22, 24, 34, 35]. However, these methods have a complicated selection procedure and can't completely solve the leader selection problem, which means the MOPSOs still lacks sufficient selection pressure toward the true PFs.

(2) Convergence efficiency in solving MaOPs.

Being different from SOPs, due to the conflicting nature between the multiple objectives, there does not exist such a search direction that can be clearly confirmed. Therefore, MOPSOs do not show a good convergence advantage compared with MOEAs. As shown in Fig. 1, we selected five MOPSO algorithms [32, 26, 19, 36, 22] for comparison with MOEA/D [11], each with a population size of 105 and 30,000 function evaluations (FEs). It is obvious from the figure that the convergence of MOEA/D on three-objective DTLZ3 is significantly better than other state-of-the-art MOPSOs. As a consequence, the convergence efficiency of PSO is not high enough to find a set of Pareto optimal solutions within a limited number of generations.

(3) Invalid search.

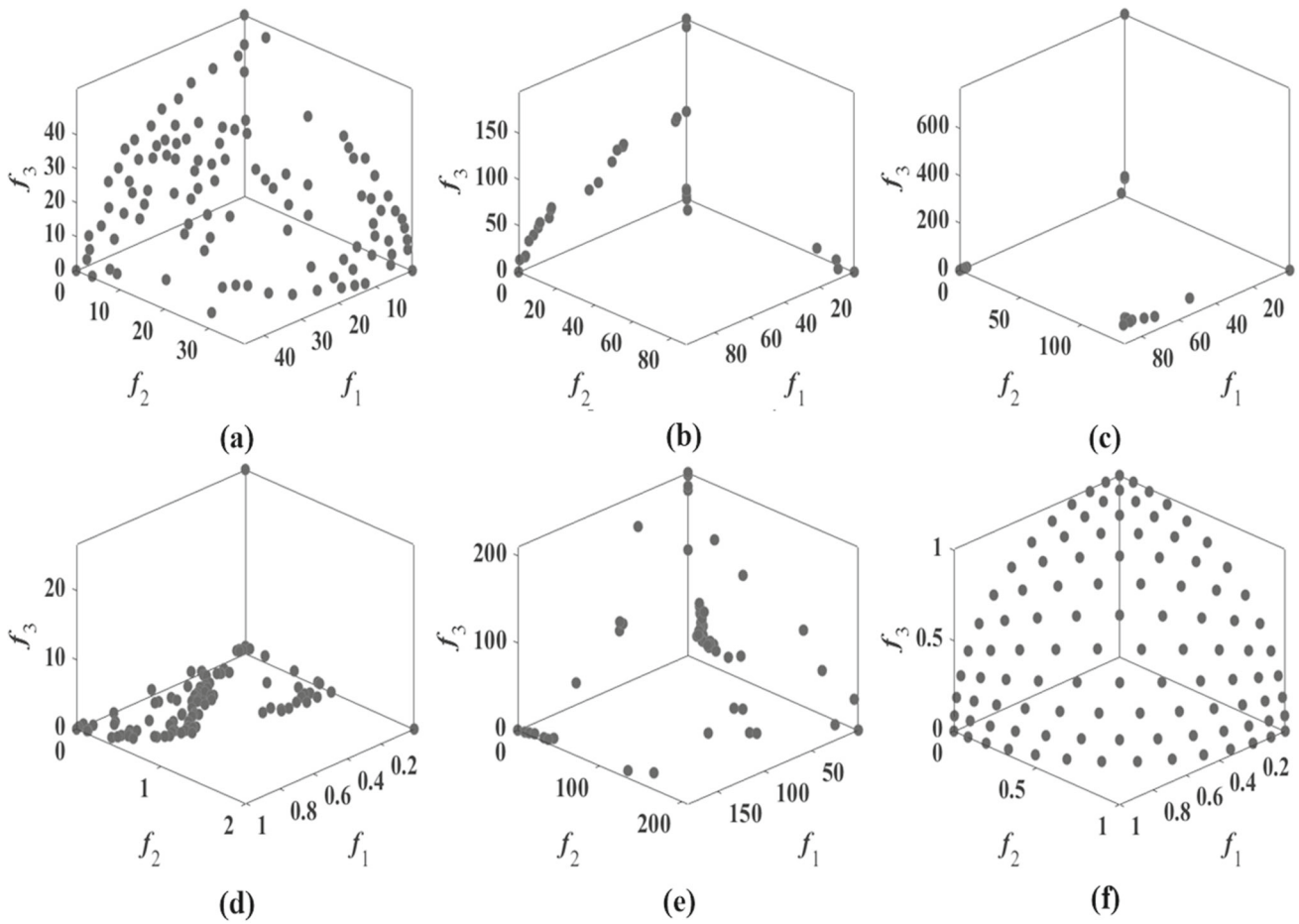


Fig. 1 Nondominated solution set obtained by (a) CMOPSO, (b) SMPSO, (c) IDMOPSO, (d) MMOPSO, (e) MaOPSO/vPF and (f) MOEA/D on DTLZ3 with three objectives

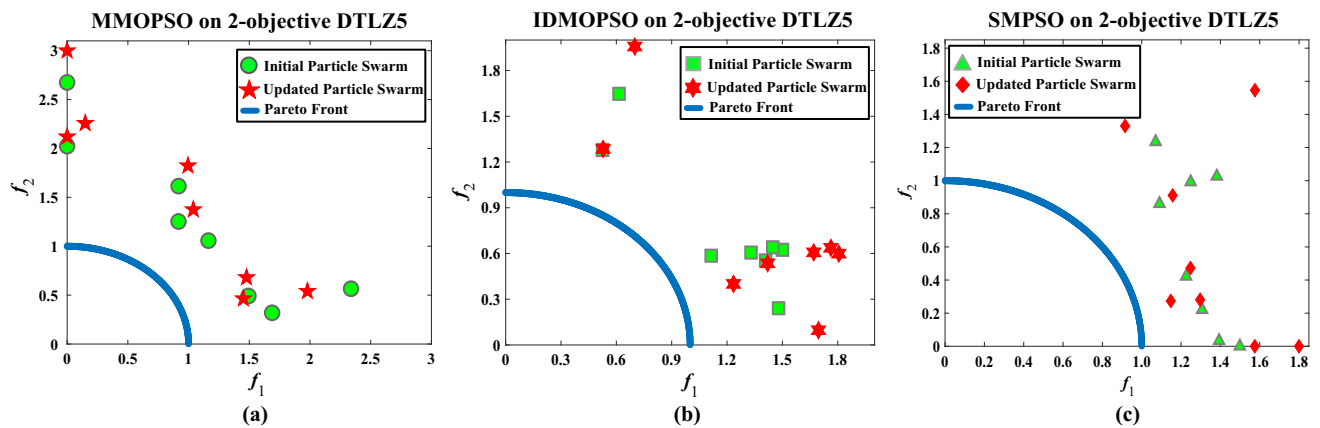


Fig. 2 a MMOPSO on 2-objective DTLZ5. b IDMOPSO on 2-objective DTLZ5. c SMPSO on 2-objective DTLZ5

Although PSO has been applied to solve MaOPs and other real-life applications, little work has been reported to consider the invalid search of particles in the objective space. In most existing MOPSOs, the velocities and the positions of the particles are usually updated using the positional information of the personal- and global-best particles. After all the particles are updated, the updated particles will directly pass to the next generation population. However, not all updated particles are valid particles. This may cause insufficient selection pressure for the population to approach the true PFs. To illustrate this fact, Fig. 2 shows the positions of eight particles updated by MMOPSO, IDMOPSO, and SMPSO strategies on 2-objective DTLZ5, respectively. As shown in Fig. 2, the updated particles are not always towards the PF. Specifically, some updated particles even move away from the PF, which will affect the search efficiency.

To solve the abovementioned issues, we propose a competitive swarm optimizer with probabilistic criteria for MaOPs, termed MOCSOP. On the one hand, MOCSOP guarantees convergence efficiency of the algorithm through the winner pool and particle swarm update strategy. On the other hand, to produce well-distributed Pareto fronts, we use the environment selection scheme with the mixed probability criterion to select particles which will enter the external archive. The specific content of MOCSOP will be described in detail in Sect. “Proposed algorithm”.

Proposed algorithm

Probability estimation method

MOPSOs guide the search direction of particles in the swarm through the appropriate swarm leaders, so choosing the swarm leaders is very important. It directly affects the performance of the MOPSO algorithms, especially when solving MaOPs, and an inappropriate swarm leader selection method will increase the invalid exploration of particles in the objective space.

In view of the above, we use probability estimation methods to find the swarm leader particle among the current swarm to form the winner pool. We first compute the probability values of the particles in each objective on the space of probabilities.

$$P_k(x_i) = \frac{D_k(x_i)}{|S| - 1} \tag{2}$$

Probability theory is used to define $P_k(x_i)$ as the probability that x_i wins the comparison on the k -th objective. In other words, $P_k(x_i)$ is the probability that $x_i \in S$ wins a comparison, according to k -th objective ($f_k, k = 1, 2, \dots, m, m$ is the number of objectives), against another randomly selected

solution from S . If $P_k(x_i) > P_k(x_j)$, it means that the probability of x_i winning the comparison on the k -th objective is higher than that of x_j . We can also say x_i performs better than x_j on the k -th objective. Where S represents the finite set of feasible solutions under consideration, $||$ represents the $L1$ -norm, $|S|$ represents the size of the population, and $D_k(x_i)$, which is calculated using the competition strategy, represents the number of times that x_i has won the competition with other particles in the population on the k -th objective. In the minimization problem, the comparison rules are as follows:

$$com(j) = \begin{cases} 1 & \text{if } f_k(x_i) < f_k(x_j) \\ 0 & \text{if } f_k(x_i) \geq f_k(x_j) \end{cases} \tag{3}$$

$$D_k(x_i) = \sum_{j=1, j \neq i}^{|S|} com(j) \tag{4}$$

For example, consider a problem involving the minimization of three objectives, f_1, f_2 and f_3 . The population contains four particles, a, b, c and d. Assume that the corresponding values of (f_1, f_2, f_3) for different particles are $a \equiv (0.5, 1, 1), b \equiv (4, 4, 3), c \equiv (3, 3, 1.5)$ and $d \equiv (1, 1.5, 2)$, respectively. Then we construct a probability matrix, M , which has dimensions $||S|| \times ||m||$, where $||$ indicates the set cardinality, S represents the population, and m is the number of objectives. Each row of the matrix M corresponds to one individual and each column of the matrix M corresponds to one objective vectors. In this example, a 4×3 matrix is constituted as shown in Fig. 3a. Then, as shown in Fig. 3b, the number of times each particle wins the competition on k -th objective is calculated by Eq. (4). And finally as shown in Fig. 3c, we calculate the probability of each particle winning the comparison on the k -th objective according to Eq. (2).

However, $P_k(x_i)$ only reflects the probability that x_i wins a comparison on the k -th objective. To estimate the quality of particles on all objectives in the population, we use the joint probability representation:

$$P(x_i) = P_1(x_i) * P_2(x_i) * P_3(x_i) \dots \dots \dots * P_m(x_i) \tag{5}$$

where $P(x_i)$ is the probability that x_i wins a comparison on all objectives, against another solution randomly selected from the current population S . The joint probability represents the probability of the particle x_i wins the competitions on all objectives. As shown in Fig. 3d, if $P(x_i) = 1$, it means that x_i is the best for all the objective functions and can dominate all other particles in the swarm. If $P(x_i) = 0$, then the convergence of x_i is worse than that of other particles. The joint probability reflects the quality of particles in the population, but the calculation of joint probability also suffers from the “curse of dimensionality”. For instance, consider a problem involving the minimization of five objectives, f_1, f_2, f_3, f_4 , and f_5 . Assume that the corresponding values of ($P_1(y)$,

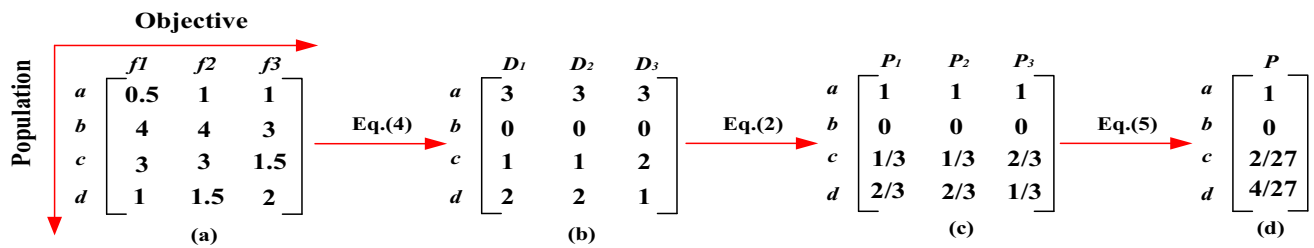


Fig. 3 Illustrative example to show the process of probability estimation

$P_2(y)$, $P_3(y)$, $P_4(y)$, and $P_5(y)$ for y are (0.1, 0.2, 0.5, 0.9, 0.4). The joint probability of y can be calculated by the formula (5): $P(y) = 0.1 * 0.2 * 0.5 * 0.9 * 0.4 = 3.6 \times 10^{-4}$. when the number of objectives increases significantly, $P(y)$ may underflow. In addition, once a particle performs poorly on one objective, the joint probability of value will directly go to 0. To obtain an easy-to-calculate equivalent formula instead of joint probability, we transform the product into a summation form through a logarithmic operation, which is often used in machine learning [37]. The approximate values of the joint probability can be computed as follows:

$$PV(x_i) = -\log(P_1(x_i)) - \log(P_2(x_i)) \dots - \log(P_m(x_i)). \tag{6}$$

It is worth noting that when $P_k(x_i) = 0$, we set $P_k(x_i) = 10^{-6}$. A smaller $PV(x_i)$ indicates that x_i has a higher probability of winning the comparison on all objectives. Algorithm 1 gives the entire process of probability estimation. Each particle is assigned a value of joint probability through the probability space to reflect the particle’s quality in the swarm.

Algorithm 1 Probability Estimation (S)

Input: $S = (x_1, x_2, \dots, x_N)$ (finite set of feasible solutions under consideration), M (number of objectives), N (size of population)

Output: PV (value of Joint Probability)

- 1: **for** $i=1$ to N **do**
- 2: **for** $j=1$ to M **do**
- 3: Compute $P_j(x_i)$ of x_i using (2);
- 4: **end for**
- 5: Compute $PV(x_i)$ of x_i by (6);
- 6: **end for**
- 7: **return** PV ;

The competition mechanism with winner pool

Recent literature reports that the competitive swarm optimizer (CSO), compared with the traditional PSO in solving MOPs and MaOPs, improves the swarm diversity to avoid premature convergence [20, 32]. Specifically, in the competitive swarm optimizer, two particles are randomly selected at a

time. The velocity of the particles with poor fitness is updated according to the position of the particles with good fitness, and the winner is directly passed to the next generation of the swarm. In our method, we also use the strategy of the competitive swarm optimizer to guide the entire swarm by learning from the winner, but we have three new contributions. First, we clearly define the swarm leaders by probability criterion. Second, we form the winner pool by selecting the particles with the best value of joint probability from the current population instead of using the random competition mechanism to select the winner. Third, MOCSOP does not use the personal- and global-best particles as mentioned in [19, 36, 38], and we use the particles of winner pool directly to guide all the particles to approach the true PFs of MaOPs. In summary, the particle velocity is updated in the proposed MOCSOP by:

$$v_{i,j}(t + 1) = \omega v_{i,j}(t) + c_1 r_1 (x_{w,j}(t) - x_{i,j}(t)), \tag{7}$$

where each particle has an n -dimensional position, $x_i(t) = (x_{i,1}(t), x_{i,2}(t), \dots, x_{i,n}(t))$, t is the iteration number, ω is the inertial weight, c_1 is the learning factor, r_1 is a random number generated uniformly in the range $[0, 1]$, the position of the winner for $x_i(t)$ is denoted as x_w and the velocity of $x_i(t)$ is an n -dimensional velocity vector $v_i(t) = (v_{i,1}(t), v_{i,2}(t), \dots, v_{i,n}(t))$.

It is worth noting that the winner pool is formed from the top 10% of the particles with better values of joint probability in the current swarm. where the winner of x_i is randomly selected from the winner pool, and then the position of x_i can be updated on the basis of the new velocity:

$$x_i(t + 1) = x_i(t) + v_i(t + 1). \tag{8}$$

Furthermore, similar to most existing MOPSOs, MOCSOP also executes polynomial mutation [39].

For further observing the position of the particles of winner pool, Fig. 4 presents an example to illustrate. It is interesting to find that the position of leaders in the population. In the early stage of the evolution, the position of leaders in the population is closer to the PF. These particles have better quality on convergence, and are regarded as swarm leaders to guide the CSO-based search. With the process of evolution, most of the generated solutions are mutually nondominated.

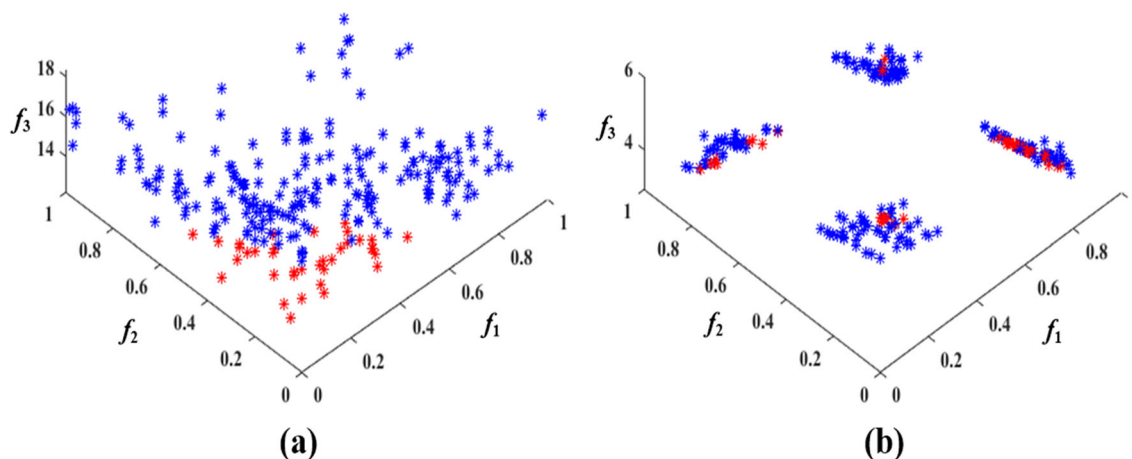


Fig. 4 Example to illustrate how to identify the position of leaders in MOCSOP for DTLZ7 with three-objectives, where the population size is set to 275 (The red points represent the leaders and blue points represent the other particles in the swarm). **a** solution set obtained by

MOCSOP after 759 FEs. **b** solution set obtained by MOCSOP after 12,903 FEs

Leaders are distributed at various positions on the PF, which enhances the diversity search of algorithm to some extent.

Algorithm 2 Environmental Selection (S, A, Z, N)

```

Input:  $S$  (finite set of feasible solutions under consideration),
 $Z$  (set of reference points),  $A$  (original archive),  $N$  (size of
population)
Output:  $A'$  (new archive)
1:  $S_t = \emptyset, i=1, A' = \emptyset;$ 
2:  $R_t = S \cup A;$ 
3:  $(F_1, F_2, \dots, F_l) = \text{Nondominatedsort}(R_t);$ 
4: repeat
5:    $S_t = S_t \cup F_i$  and  $i = i + 1;$ 
6: until  $|S_t| \geq N$ 
7:  $PV = \text{Probability Estimation}(S_t);$ 
8: Last front to be included:  $F_l = F_i;$ 
9: if  $|S_t| = N$  then
10:   $A' = S_t,$ 
11:  return  $A';$ 
12: else
13:   $A' = \bigcup_{i=1}^{l-1} F_i;$ 
14:  Objective Space Normalization ( $S_t$ );
15:  Association and Niche Preservation
    Operation ( $S_t, A', Z, PV, N$ );
16: end if
17: return  $A'$ 
    
```

Environmental selection

Similar to the existing MOEAs [39], MOCSOP also uses a set of predefined reference points to ensure the diversity of the obtained solutions. As presented in Algorithm 2, the combined population R_t is divided into different layers ($F_1,$

$F_2,$ and so on) by a nondominated sorting procedure, where F_j is the j -th Pareto nondomination level of R_t , and the last layer F_l is determined. The critical front is S_t , if $|S_t| = N$, then return $A' = S_t$. Otherwise, when $|S_t| \geq N$, we first estimate the joint probability of the particles in S_t . Then, the remaining ($K = N - |A'|$) swarm members are chosen from the last front F_l by using the association and niching operation with the mixed probability criterion (line15). In what follows, we will describe them in more detail in the following subsections.

Objective space normalization

In general, different objectives have different ranges, which can directly affect the diversity estimation of the population. Therefore, we need to perform an adaptive normalization procedure on the critical front S_t . Several normalization methods have been proposed [40, 41], and we utilize the adaptive normalization method proposed in [39]. Specifically, the normalization of the objective functions can be computed using the following equation:

$$f'_i(x_i) = \frac{f_i(x_i) - z_i^{\min}}{b_i}, \tag{9}$$

where the ideal point $z^{\min} = (z_1^{\min}, z_2^{\min}, \dots, z_m^{\min})$ is constructed from the minimum value of each objective function f_i and b_i is the intercept of the i -th objective axis.

1. Association and Niche-Preservation Operation
 The proposed MOCSOP has a similar association and niching operation as [39], except that the probability criterion is added to the niche-preservation operation procedure. In

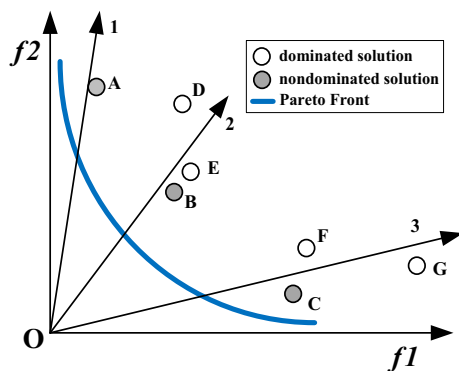


Fig. 5 Illustrative example to show the differences between archive updated by our method and NSGA-III

our proposed method, when a reference vector already has one member associated with it that exists in S_i/F_i , the particles with the best value of joint probability are preferentially selected to pass the archive. A simple example is displayed for illustration as shown in Fig. 5, where A, B, and C are non-dominated solutions, D, E, F and G are dominated solutions. Assume that five out of the seven candidate solutions need to be selected for the next archive. Considering that A, B and C are in the first layer, they are preferentially selected to enter the archive. In this case, all the reference vectors have a particle associated with it in the first layer. Then, we still need to select two particles from last layer to enter the archive. For the reference vector 2, NSGA-III randomly chooses a particle from D and E to enter the archive. However, randomly selected particles have uncertainty, and some particles with good quality may be missed. In MOCSOP, the E is passed to the archive, because of the fact that the joint probability value of E is better than D according to Eq. 6. Similar operation, we choose F to enter the archive.

There are two main reasons prompting that we add the probability criterion to the association and niching operation. On the one hand, evolutionary search and swarm update strategy are applied to the external archive (presented in Sect. “Evolutionary search on the external archive” and “Swarm update strategy”), therefore more particles with better joint probability value in the archive can effectively improve the search efficiency, especially in the early stage. On the other hand, during the experiments, we find that the proposed method is beneficial for solving large-scale MOPs. Refer to Sect. “Further discussion” for more details.

Evolutionary search on the external archive

To further enhance the solution quality in the external archive and to repair the potential insufficiency of CSO search on some MaOPs, we use the evolutionary search to further explore the archives. Recently, [36, 38] have shown that this

hybrid scheme not only effectively improves the search ability of MOPSOs, but it also enhances the robustness of the algorithm to tackle various complex PFs. In this paper, the evolutionary search framework is the same as in NMPSO [38], and we also use the simulated binary crossover (SBX) and polynomial mutation (PM) [39] to extend the search capabilities of the CSO. Due to space limitations, the specific details of using the evolutionary search to assist CSO can be found in [36, 38].

Swarm update strategy

After the velocity and position of the particle swarm are updated, the swarm has a large number of invalid particles. These particles are mostly concentrated in crowded areas or far from the Pareto front. In response to this problem, this article proposes a simple and efficient particle swarm update strategy to ensure that the particles can search the Pareto front efficiently while avoiding the repeated search of invalid areas that affect the convergence and equilibrium of the algorithm. The specific details are shown in Algorithm 3. After the environmental selection, if a_i comes from the updated particle swarm S (in line 14 of Algorithm 4) and still survive, then a_i will inherit the updated velocity (in line 10 of Algorithm 4); otherwise, the velocity of a_i will set to be 0. The external archive is directly used as the next-generation particle swarm to ensure the effectiveness of the particles in the swarm. The reason for the above is that the archive, after environmental selection, retains some elite individuals of this generation, and these individuals are obviously valid particles. Second, the external archive saves all of the elite individuals who have been searched so far. These individuals, as the next generation of particles, ensure the effectiveness of the entire population and avoid the invalid search of particles.

Algorithm 3 Swarm Update Strategy (S, A)

Input: $S = (x_1, x_2, \dots, x_N)$ (updated particle swarm), $A = (a_1, a_2, \dots, a_N)$ (archive)

Output: Q (particle swarm for next generation)

```

1:  $Q = \emptyset$ ;
2: for  $i=1$  to  $N$  do
3:   if  $a_i \in S$  then
4:      $a_i$  inherits the updated velocity and  $Q \cup a_i$ ;
5:   else
6:      $v_{a_i} = 0$  and  $Q \cup a_i$ ;
7:   end if
8: end for
9: return  $Q$ 

```

Figure 6 illustrates an example that shows the advantage of the swarm update strategy over the traditional PSO. The traditional PSO usually chooses the updated particles entering the next generation of the swarm [24, 38]. As shown in

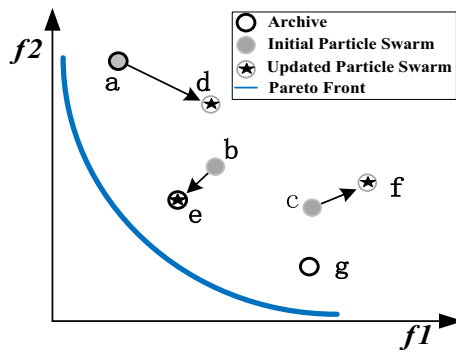


Fig. 6 Illustration of the procedure of swarm update strategy on a bi-objective problem

Fig. 6, the particle swarm $\{d, e, f\}$ is the next generation of the swarm. We can find that the positions and motion directions of d and f are far away from the PF; this situation will increase the invalid search of the PSO in the objective space and it will affect convergence. By contrast, the particle swarm $\{a, e, g\}$ is selected by the proposed swarm update strategy. They are the best candidate particles in terms of convergence and diversity. In addition, e retains the direction of velocity, which ensures that the particles always move toward the PF.

Algorithm 4 Complete Algorithm of MOCSOP

Input: N (size of population)
Output: A (final archive)

- 1: $S \leftarrow \text{Random Initialization}(N)$;
- 2: $Z \leftarrow \text{Uniform ReferencePoint}(N)$;
- 3: add the non-dominated solutions from S into A ;
- 4: **while** termination criterion not fulfilled **do**
- 5: $PV \leftarrow \text{Probability Estimation}(S)$;
- 6: create WP (winner pool) according to PV ;
- 7: $S = S \setminus WP$;
- 8: **for** each particle $x_i \in S$ **do**
- 9: randomly select a particle from WP as x_{win} ;
- 10: update the velocity using Eq. (7);
- 11: update the position using Eq. (8);
- 12: **end for**
- 13: $S = S \cup WP$;
- 14: $S \leftarrow \text{Polynomial Mutation}(S)$;
- 15: update Z^{min} ;
- 16: $A \leftarrow \text{Environmental Selection}(S, A, Z, N)$;
- 17: evolutionary search strategy is applied on A to get a new swarm S' ;
- 18: update Z^{min} ;
- 19: $A \leftarrow \text{Environmental Selection}(S', A, Z, N)$;
- 20: $S \leftarrow \text{Swarm Update Strategy}(S, A)$;
- 21: **end while**
- 22: **Output** A

Complete algorithm of MOCSOP

Similar to most existing MOPSOs, the proposed MOCSOP has a very simple framework. To describe the complete algorithm of MOCSOP in detail, Algorithm 4 presents the pseudocode of its complete framework, and the main framework of MOCSOP consists of the following steps. It begins with the initialization of a random population S and a set of uniformly distributed reference vectors Z . For each particle in S , its positional information is randomly generated, and its velocity is set to 0. Furthermore, we use Das and Dennis’s [42] method to generate uniformly distributed reference points. In line 3, the external archive A is initialized and all nondominated solutions in S are distinguished and added into A . During the evolutionary phase, we first use the probability estimation method to find the swarm leader particles among the S , and then select the particles with better values of joint probability in the current swarm to create a winner pool. After that, for each particle in S , the particle velocity and position are updated by using Eqs. (7) and (8) in lines 10–11. Then, to enhance the search ability of the CSO, the polynomial mutation is also performed. In line 16, we update the archive A by executing environmental selection. Then, in line 17, the evolutionary search strategy is applied on A to obtain new solutions. For the new population S' , we execute the environmental selection to update archive A again. Finally, swarm update strategy procedures are performed to ensure that the next generation particles are all valid particles. The main loop will repeat until a termination criterion is reached, and the archive A is reported as the final approximated PF.

Computational complexity analysis

The computational complexity of MOCSOP is mainly related to the operations of probability estimation and environmental selection. For a population size N and a M -objective problem, the computational complexity of probability estimation is $O(MN^2)$. In the population update stage, all particles in the population are updated in the worst-case scenario, and this requires $O(N)$ calculations. For the evolutionary search strategy, two operations, the simulated binary crossover (SBX) and polynomial mutation are calculated, which requires a runtime of $O(M \frac{N}{2})$. In the environmental selection, we require $O(MN^2)$ computations for nondominated sorting and niching operation. After archive updating, the swarm update strategy executed requires $O(N)$ in the worst case. In summary, the worst-case time complexity of one generation in MOCSOP is $O(MN^2)$.

Compared with the existing MOPSOs and MOEAs, our MOCSOP method is computationally efficient in solving MaOPs. In the experimental section, we will compare the average runtimes of MOCSOP with that of the various evaluated approaches for MaOPs.

Experimental results and analysis

In this section, to prove the effectiveness of our algorithm model, we first compare our method with five typical MOPSOs, namely, CMOPSO [32], NMPSO [38], IDMOPSO [19], MMOPSO [36] and MaOPSO/vPF [22]. Where CMOPSO is a recently proposed competitive mechanism-based PSO, MMOPSO is an improved version of MOPSO with multiple search strategies, NMPSO, IDMOPSO and MaOPSO/vPF are three novel PSO algorithms designed for solving MaOPs. These comparable methods have shown excellent performance on both MOPs and MaOPs with various types of Pareto fronts. Then, we compared our approach with five state-of-the-art MaOEAAs including MaOEA/IGD [43], NSGA-II/SDR [8], VaEA [44], MOEA/D-CMA [45] and A-NSGA-III [46]. They have shown a good balance between diversity and convergence on MOPs and MaOPs.

In the experiment, we selected 16 test problems, including DTLZ1-DTLZ7 [47] and WFG1-WFG9 [48], which were widely used to evaluate the performance of the algorithm. Based on the different types of Pareto fronts, these test problems can be roughly divided into three groups. The first group is DTLZ1, which includes a linear PF. The second group consists of DTLZ2-DTLZ4 and WFG4-WFG9, which have a concave PF. The problem with concave PF may have a great number of local optima, which imposes a great challenge for algorithms to push the population toward the PF. The third group consists of DTLZ5, DTLZ6, and WFG1-WFG3. These instances have discontinuous (DTLZ7 and WFG2), degenerated (DTLZ5, DTLZ6 and WFG3) and other complex PFs (WFG1), which brings the challenge to maintain the diversity of population. In this paper, we use DTLZ and WFG with a number of objectives that range from 3 to 15, because of the fact that they can scale any number of objectives and decision variables. The number of decision variables for DTLZ test suites is set to $n = k + m - 1$, where m is the number of objectives and n is the number of decision variables. As recommended in [43], we set $k = 5$ for DTLZ1, $k = 10$ for DTLZ2 to DTLZ6 and $k = 20$ for DTLZ7. For WFG1-WFG9 test instances, the number of decision variables is set to $n = k + l$ as suggested in [44], where k is set to $m - 1$, and the distance-related variable $l = 10$.

Experimental settings

Reference points and population size:

NSGA-III, MOEA/D-CMA, MaOEA/IGD, VaEA, IDMOPSO, MaOPSO/vPF and MOCOSOP were all used Das and Dennis's [42] approach with two layers to generate uniformly distributed reference points. According to the suggestion of [16], for the test suites of 3, 5, 6, 8, 10 and 15 objectives, we set the number of weight vectors to 105, 126, 132, 156, 275, 135 respectively. In addition, for quantitative

Table 1 Parameters Settings of all the Algorithms Compared

Algorithm	Parameters settings
NMPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2, c_3 \in [1.5, 2.5]$, $r_1, r_2, r_3 \in [0, 1]$, $p_m = 1/D$, $\eta_m = 20$
CMOPSO	$\omega \in [0, 1]$, $r_1 \in [0, 1]$, $p_m = 1/D$, $\eta_m = 20$
MMOPSO	$\omega \in [0.1, 0.5]$, $c_1, c_2 \in [1.5, 2.5]$, $r_1, r_2 \in [0, 1]$, $\eta_c = \eta_m = 20$, $p_m = 1/D$, $p_c = 1$
IDMOPSO	$\omega \in [0.125, 0.495]$, $c_1, c_2 = 1.494$, $r_1, r_2 \in [0, 1]$, $k = 0.005$
MaOPSO/vPF	$\omega \in [0.1, 0.5]$, $c_1, c_2 \in [1.5, 2.5]$, $r_1, r_2 \in [0, 1]$
MOCOSOP	$\omega \in [0.1, 0.5]$, $c_1 \in [1.5, 2.5]$, $r_1 \in [0, 1]$, $\eta_c = \eta_m = 20$, $p_c = 1$, $p_m = 1/D$
A-NSGA-III	$\eta_c = \eta_m = 20$, $p_c = 1$, $p_m = 1/D$
MaOEA/IGD	$p_c = 0.9$, $\eta_c = \eta_m = 20$, $p_m = 1/D$
VaEA	$p_c = 1$, $\eta_m = 20$, $p_m = 1/D$
NSGAII/SDR	$p_c = 1$, $\eta_m = 20$, $p_m = 1/D$
MOEA/D-CMA	$T = 0.1 * N$, $K = 5$

comparisons, the population size of each comparison method is set to the same value as the number of reference points.

Experimental settings of all compared algorithms:

For fair comparisons, the parameters of all comparison methods were set according to their references. Table 1 lists the related parameters used in the experiments for each algorithm, where D is the dimension of the decision space, ω is the inertial weight, c_1 and c_2 are two learning factors, r_1 and r_2 are two uniformly distributed random numbers, and η_c and η_m are the distribution indexes of SBX and PM, respectively. p_c and p_m are the crossover and mutation probabilities used in evolutionary operators, respectively. Regarding MOEA/D-CMA, the number K of Gaussian models is set to 5. In addition, for MOEA/D-CMA, T is the neighborhood size. In IDMOPSO, k is set to 0.005 according to [45]. For MOCOSOP, no additional parameters are needed to be specified.

Performance metrics:

To demonstrate the capability of our method in convergence and diversity quality, we utilized the inverted generational distance (IGD) [49] and hypervolume (HV) [50] to evaluate the performance of various approaches on MOPs and MaOPs. Specifically, the IGD and HV can measure the convergence and diversity between nondominated solutions generated by the algorithm and true PFs. In the calculation of IGD and HV, the sampling of the reference points was adopted from the suggestion of [51]. Moreover, for a comprehensive evaluation, Wilcoxon rank was further employed to test the performance of various evaluated models [52]. In the experiment, the symbols “+”, “−”, “,” and “≈” indicate that the results obtained by other comparison algorithms

are significantly better than, worse than, and similar to that obtained by MOCSOP, respectively. In this paper, the number of evaluations is adopted as the termination criterion. For DTLZ1–DTLZ7 and WFG1–WFG9, the maximal number of evaluations is set to $M \times 30,000$. In addition, all the experiments performed 20 independent runs for each algorithm on each test instance by utilizing a PC with an Intel Core I7-8750H CPU and an Nvidia GeForce GTX 1060 GPU.

Comparisons of MOCSOP with five competitive MOPSOs for solving MOPs and MaOPs

We first discuss the convergence of MOCSOP. To investigate the convergence of the proposed approach in the search process, we utilize three test functions DTLZ1, DTLZ3, and WFG1 to conduct a comparative experiment. For a quantitative evaluation, all comparison models are set to use the same initial population and each test suite is run for 20 times. The convergence profiles of IGD values obtained by MOCSOP and compared methods are plotted in Fig. 7. As shown in Fig. 7a, for DTLZ1 with linear Pareto front, MOCSOP converges to PF significantly faster than other evaluation approaches. Especially at the beginning of optimization, MOCSOP has converged rapidly, which means that the proposed approach is effective in solving problem with linear PF. It is known that PSO encounters great challenges when tackling DTLZ3 [36]. This is mainly because the DTLZ3 contains a large number of local optima that will pose challenges to existing MOPSOs in obtaining nondominated solutions. Figure 7b shows the IGD curves of all the compared algorithms on DTLZ3. We can find that CMOPSO, IDMOPSO, MaOPSO/vPF have performed poorly on DTLZ3, one possible reason for the poor convergence of the above algorithms is that maybe the PSO-based search lose its efficiency in solving problem with a great number of local optima. Although NMPSO and MMOPSO reach the best IGD values, their convergence speed is significantly slower than MOCSOP during the whole evolutionary process. Thus, it is unsurprising that the convergence of the proposed MOCSOP is better than NMPSO and MMOPSO on DTLZ3. To further observe the convergence of MOCSOP for complex PF, Fig. 7c depicts the evolutionary trends of the compared models on WFG1. WFG1 instance includes an irregular PF, which imposes a great challenge for MOPSOs to push the population toward the PF. As can be seen from the figure, MMOPSO, CMOPSO, IDMOPSO and MaOPSO/vPF have trouble in convergence. As for MOCSOP and NMPSO, they obtain similar IGD values in the early stage. However, the IGD values of NMPSO increases after 70,000 FEs. This phenomenon may be due to the fact that NMPSO only reaches the local optimum and does not toward the true PF. In contrast, as the iteration proceeds, the solutions obtained by MOCSOP get closer and closer to the true PF. By comparing the convergence of MOCSOP

with those of traditional PSO algorithms, we conclude that the proposed method has the gratifying capacity of convergence.

To make a visual comparison, Fig. 8 shows the non-dominated set obtained by MOCSOP and other competitive MOPSOs on three-objective WFG2. For WFG2, it includes a disconnected PF, which brings the challenge to maintain the diversity of population. As shown in Fig. 8, we can find that all compared methods exhibit the good performances in terms of convergence on WFG2 and have successfully converged to the true PF. However, these algorithms perform poorly in maintaining the diversity of population. NMPSO has obtained the sparse nondominated set on the Pareto front, which indicates that the balanceable fitness estimation method is not suitable for solving the problem with disconnected PF. Although CMOPSO, MMOPSO, IDMOPSO and MaOPSO/vPF have dense population, their solutions are not uniformly distributed on the disconnected parts. In contrast, the nondominated solution set obtained by our method has a significant improvement compared with those of other evaluation approaches, as evidenced by our solution set, which is closer to the shape of the true Pareto front.

To conduct a comprehensive comparison between the various methods, Table 2 summarizes the median IGD comparison results of MOCSOP with respect to five current MOPSOs on DTLZ1–DTLZ7 and WFG1–WFG9 with 3–15 objectives. As can be seen from Table 2, the proposed MOCSOP wins 46 out of the 80 comparisons, demonstrating its efficiency in handling general MaOPs. Specifically, MOCSOP achieves the best IGD values on DTLZ1 with 3 to 15 objectives. It is demonstrated that the proposed MOCSOP achieved promising performance on the problem with a linear PF. As for the instances with a concave Pareto front such as DTLZ2–DTLZ4, MOCSOP also produces competitive results compared with those of the other state-of-the-art approaches, especially on the high objectives. It is worth noting that the proposed MOCSOP performs worse than NMPSO on DTLZ5–DTLZ7 regarding IGD. This finding is unsurprising and is mainly because the reference points in MOCSOP has poor distributions on those irregular PFs, which may mislead the search efforts of the algorithm. Furthermore, CMOPSO obtains the worst IGD values on WFG3. This is because the conventional particle swarm update strategy lacks sufficient selection pressure to approach the true PF of problem with a disconnected PF. This phenomenon also exists in MMPSO and MaOPSO/vPF. Although the performance of MOCSOP is slightly worse than that of NMPSO, it performs competitively on WFG3. As far as the IGD is concerned, the performance measures obtained by the proposed MOCSOP on WFG3 with 3 to 15 objectives are better than those of MMPSO and MaOPSO/vPF. This means that the nondominant solutions obtained by MOCSOP is closer to the true PFs than those obtained by MMOPSO and

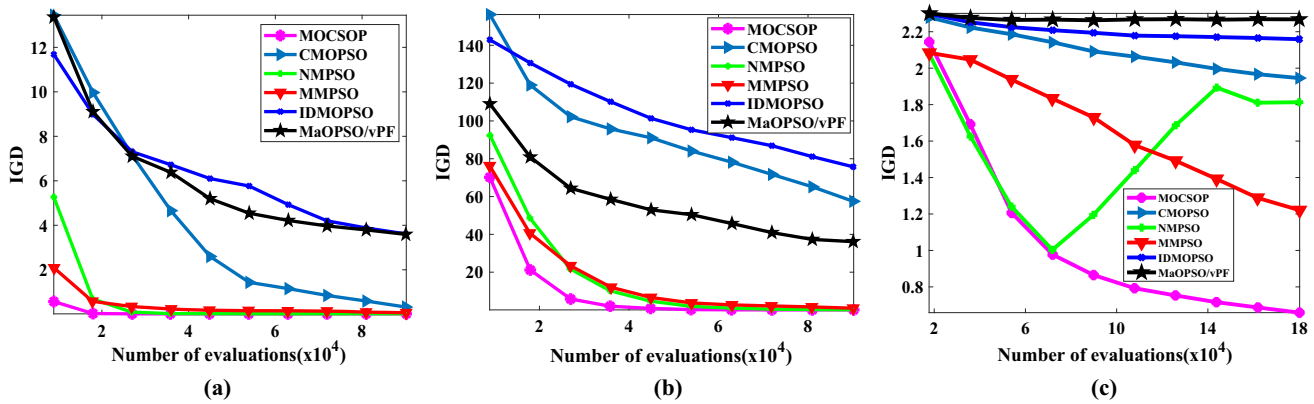


Fig. 7 Evolutionary trajectories of IGD for MOCSOP, CMOPSO, NMPSO, MMOPSO, IDMOPSO and MaOPSO/vPF on a DTLZ1 with three objectives. **b** DTLZ3 with three-objectives. **c** WFG1 with six objectives

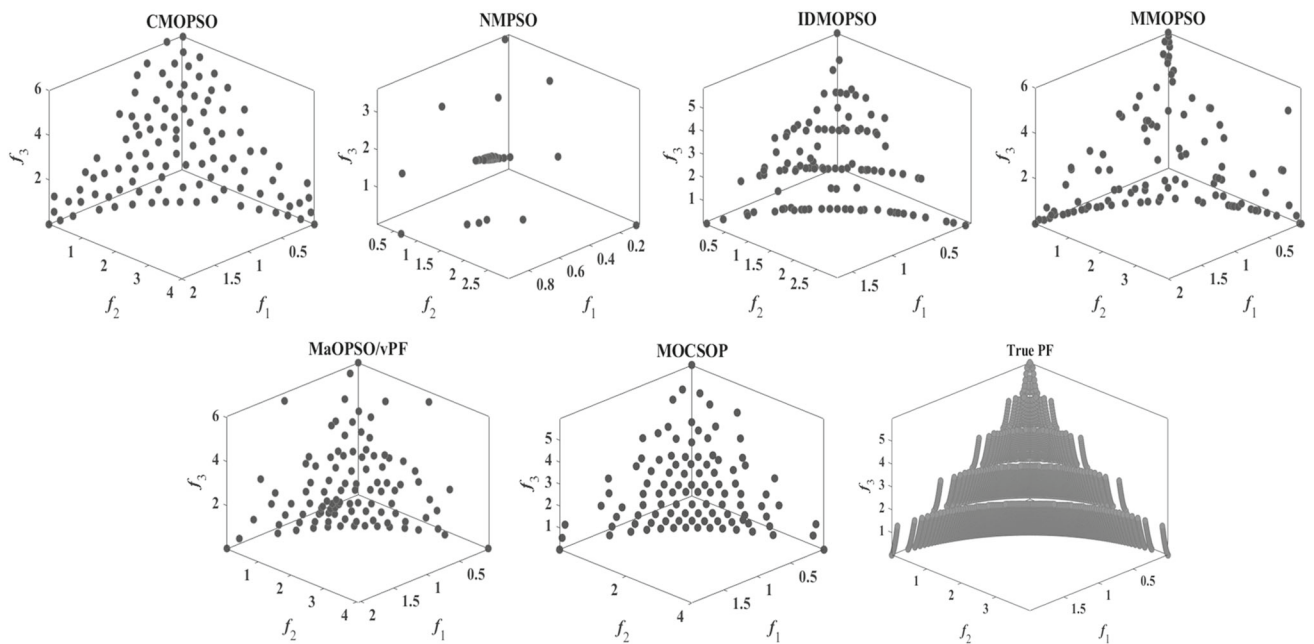


Fig. 8 Pareto nondominant solutions with the median IGD values among 20 runs obtained by CMOPSO, NMPSO, IDMOPSO, MMOPSO, MaOPSO/vPF and MOCSOP on three-objective WFG2

MaOPSO/vPF. For the other test instances with irregular PF, the proposed MOCSOP also achieves good performance in terms of both convergence and uniformity, such as WFG2 and WFG3. Therefore, the comparison results in Table 2 demonstrate that MOCSOP has good ability of solving MaOPs with various types of PFs.

Comparisons of MOCSOP with other state-of-the-art MaOEAs

To verify the capability of MOCSOP for handling MaOPs, we compare our approach with some state-of-the-art approaches, including A-NSGA-III, MOEA/D-CMA, MaOEA/IGD, NSGA-II/SDR and VaEA. These are typical methods from

different categories that use different techniques. NSGA-II/SDR is a variation of NSGAI to tackle the MaOPs, which uses a new dominance relation. ANSGA-III is constructed by applying an adaptive reference point scheme to the original NSGA-III approach. The MaOEA/IGD is an IGD indicator-based evolutionary algorithm for solving MaOPs. Finally, VaEA is a new vector angle-based evolutionary algorithm, which has the significant advantage of diversity and convergence. The above algorithms are very competitive in addressing MaOPs, making the comparisons more comprehensive.

Table 3 summarizes the HV values obtained by MOCSOP and five state-of-the-art MaOEAs on DTLZ1–DTLZ7 and WFG1–WFG9 with 3, 5, 8, 10 and 15 objectives. As shown in

Table 2 Median IGD values among 20 independent runs obtained by CMOPSO, NMPSO, IDMOPSO, MMOPSO, MaOPSO/vPF and MOC SOP on DTLZ1–7 AND WFG1–9 with 3, 5, 8, 10 and 15 Objectives

Obj	CMOPSO	NMPSO	IDMOPSO	MMOPSO	MaOPSO/vPF	MOC SOP	
DTLZ1	3	1.1681e + 0 (1.37e + 0) –	2.2553e – 2 (1.21e – 3) –	3.6530e + 0 (3.03e + 0)	6.7873e – 2 (1.37e – 1) –	3.1687e + 0 (1.91e + 0)	1.8976e – 2 (1.04e – 6)
	5	8.0777e + 0 (4.89e + 0) –	6.4901e – 2 (2.61e – 3) –	3.5472e + 0 (3.36e + 0)	1.2456e + 0 (1.33e + 0)	1.7511e + 0 (1.15e + 0)	6.3319e – 2 (3.41e – 5)
	8	5.6135e + 1 (2.96e + 1) –	1.5257e – 1 (1.22e – 2) –	2.2151e + 0 (2.87e + 0)	3.6704e + 0 (3.55e + 0)	3.0600e + 0 (4.35e + 0)	1.0029e – 1 (1.43e – 2)
	10	9.8241e + 1 (2.96e + 1) –	1.6598e – 1 (1.91e – 2) –	8.3410e – 1 (4.90e – 1) –	5.9088e + 0 (4.31e + 0)	2.0900e + 0 (3.44e + 0)	1.1392e – 1 (1.16e – 2)
	15	1.6518e + 2 (2.56e + 1) –	1.7809e + 1 (2.01e + 1)	1.4960e + 0 (1.33e + 0)	5.9513e – 1 (6.02 e –) –	5.2003 e – (2.55e – 1) –	1.7722e – 1 (8.26e – 3)
DTLZ2	3	5.6273e – 2 (8.98e – 4) –	7.3917e – 2 (3.23e – 3) –	7.6657e – 2 (3.53e – 3) –	6.6585e – 2 (2.43e – 3) –	7.5970e – 2 (6.09e – 3) –	5.0304e – 2 (2.71e – 6)
	5	4.3676e – 1 (4.10e – 2) –	2.1590e – 1 (2.00e – 3) –	2.4575e – 1 (5.36e – 3) –	2.4838e – 1 (6.14e – 3) –	4.4177e – 1 (2.10e – 2) –	2.1114e – 1 (7.27e – 2)
	8	2.2971e + 0 (2.70e – 2) –	3.5711e – 1 (1.54e – 3) +	4.0813e – 1 (5.10e – 3) +	6.9489e – 1 (6.36e – 2) –	6.3071e – 1 (2.35e – 2) –	4.8208e – 1 (8.37e – 2)
	10	2.3789e + 0 (3.00e – 2) –	4.1057e – 1 (1.73e – 3) +	4.7650e – 1 (3.26e – 2) +	9.6714e – 1 (8.95e – 2) –	6.8163e – 1 (4.31e – 2) –	5.1069e – 1 (4.12e – 2)
	15	2.4904e + 0 (2.50e – 2) –	8.0369e – 1 (6.72e – 2) –	6.8792e – 1 (3.23e – 2) –	1.2102e + 0 (8.22e – 2) –	8.0123e – 1 (7.99e – 2) –	6.4524e – 1 (1.96e – 2)
DTLZ3	3	5.2182e + 1 (3.92e + 1) –	7.4757e – 2 (1.99e – 3) –	7.2044e + 1 (2.02e + 1)	8.9578e – 1 (1.57e + 0) –	5.9077e + 1 (3.02e + 1)	5.0657e – 2 (5.82e – 4)
	5	1.3099e + 2 (2.61e + 1) –	2.3590 e – (6.53e – 2) –	4.5645e + 1 (2.31e + 1)	3.2622e + 1 (3.20e + 1)	4.3933e + 1 (3.19e + 1)	1.9533 e – (5.29e – 4)
	8	6.4989e + 2 (1.28e + 2) –	5.3689e – 1 (1.93 e –) –	2.2768e + 1 (1.24e + 1)	1.6179e + 2 (2.73e + 1)	3.3918e + 1 (1.69e + 1)	3.8983e – 1 (1.14e – 1)
	10	6.2169e + 2 (9.91e + 1) –	7.3790e – 1 (6.46e – 1) =	1.7963e + 1 (8.58e + 0)	1.7210e + 2 (2.25e + 1)	4.0304e + 1 (1.98e + 1)	6.2234e – 1 (3.97e – 1)
	15	1.1665e + 3 (1.49e + 2) –	1.5301e + 0 (7.13e – 1) –	4.9439e + 0 (3.06e + 0)	1.7207e + 2 (4.98e + 1)	2.8225e + 1 (2.30e + 1)	6.8662e – 1 (3.54e – 2)
DTLZ4	3	1.0369e – 1 (1.98e – 1) =	9.8760e – 2 (1.04e – 1) =	2.6042e – 1 (2.09e – 1) =	6.5424e – 2 (1.85e – 3) +	1.2451 e – (8.12e – 2) +	4.7498e – 1 (3.26e – 1)
	5	4.5828e – 1 (8.13e – 2) –	2.2701e – 1 (4.74e – 2) –	3.0842e – 1 (5.27e – 2) –	2.4310e – 1 (7.27e – 3) –	3.2311 e – (6.06e – 2) –	2.1809e – 1 (7.13e – 2)
	8	1.7192e + 0 (3.87 e –) –	4.0589e – 1 (7.30e – 2) –	4.2694e – 1 (1.66e – 2) =	6.5501e – 1 (5.20e – 2) –	4.7179e – 1 (2.71e – 2) –	3.8015e – 1 (9.27e – 2)
	10	1.5825e + 0 (3.09e – 1) –	4.2226e – 1 (1.74e – 2) –	4.8762e – 1 (1.11e – 2) –	1.0496e + 0 (1.62e – 1) –	5.0751e – 1 (2.10e – 2) –	4.2068e – 1 (1.31e – 3)
	15	2.2824e + 0 (2.92e – 1) –	1.0421e + 0 (1.96e – 1) –	6.4065e – 1 (4.50e – 3) –	1.3724e + 0 (1.60e – 1) –	7.2560e – 1 (2.75e – 2) –	6.3024e – 1 (1.40e – 2)

Table 2 (continued)

	Obj	CMOPSO	NMPSO	IDMOPSO	MMOPSO	MaOPSO/vPF	MOC SOP
DTLZ5	3	5.7123e-3 (6.94e-4) +	1.4475e-2 (2.58e-3) +	1.1501e-2 (1.31e-3) +	5.4706e-3 (3.55e-4) +	7.1844e-2 (2.66e-2) -	6.1033e-2 (6.79e-2) -
	5	6.0635 e- (2.49e-1) -	3.7836e-2 (3.47e-3) +	1.1036e-1 (2.56e-2) =	7.9544e-2 (3.22e-2) +	1.6098e-1 (5.69e-2) =	3.0379e-1 (1.97e-1) -
	8	2.0669e + 0 (4.58e-1) -	6.6669e-1 (1.63e-1) -	1.6892e-1 (6.39e-2) -	1.7748e-1 (8.63e-2) -	2.4137e-1 (6.44e-2) -	1.1999e-1 (3.35 e-2)
	10	1.9709e + 0 (5.48 e-1) -	7.5405 e-1 (9.75e-3) -	1.2245 e-1 (2.27e-2) =	1.4869 e-1 (6.97e-2) =	2.2810 e-1 (5.34 e-2) -	1.2670e-1 (3.67e-2) -
	15	2.3474e + 0 (2.36e-1) -	7.4220 e-1 (1.30 e-4) -	2.0089 e-1 (7.45 e-2) =	2.8238 e-1 (7.08 e-2) -	2.8464e-1 (7.38 e-2) -	1.9355 e-1 (5.33 e-2)
DTLZ6	3	4.0006 e-3 (4.37 e-5) +	1.3679 e-2 (2.20 e-3) +	1.1319 e-2 (1.41 e-3) +	6.0875 e-3 (1.13 e-3) +	3.1539e-2 (9.28 e-4) -	1.8845e-2 (1.71e-3) -
	5	3.9269e + 0 (1.49e + 0) -	4.4372e-2 (4.80e-3) +	9.8094e-2 (3.13e-2) +	1.9211e-1 (6.92e-2) -	1.8232e-1 (1.16e-1) =	1.4059e-1 (5.17e-2) -
	8	9.7165e + 0 (1.92e-1) -	7.4079e-1 (5.81e-3) -	2.4355e-1 (1.35e-1) =	4.3009e-1 (1.88e-1) -	1.9268e-1 (8.95e-2) =	1.8708e-1 (7.60e-2)
	10	9.6189e + 0 (3.10e-1) -	7.4209e-1 (2.28e-16) -	2.2976e-1 (9.77e-2) =	5.8912e-1 (2.08e-1) -	2.2390e-1 (9.59e-2) =	1.9475e-1 (7.07e-2)
	15	9.8849e + 0 (1.32e-1) -	7.1498e-1 (1.21e-1) -	2.3220e-1 (1.03e e-1) =	7.4097e-1 (5.00e-3) -	4.7676e-1 (2.39e-1) -	3.0260e-1 (1.08e-1) -
DTLZ7	3	1.5167e-1 (2.31e-1) +	6.6499e-2 (3.42e-3) +	8.2018e-1 (1.04e-1) -	9.2911e-2 (6.00e-2) =	1.3739e-1 (5.99e-3) =	2.6545e-1 (2.73e-1) -
	5	5.8291e-1 (4.78e-2) -	2.7899e-1 (8.58e-3) +	1.7812e + 0 (2.04e-1) -	3.5874e-1 (1.29e-2) =	6.5915e-1 (3.08e-2) -	5.5907e-1 (3.96e-1) -
	8	9.3883e + 0 (3.77e + 0) -	7.2136e-1 (1.02e-1) +	3.3519e + 0 (2.35e-1) -	7.8212e-1 (1.74e-2) +	1.2294e + 0 (2.65e-1) -	8.8760e-1 (1.80e-1) -
	10	1.4010e + 1 (5.76e + 0) -	8.6288e-1 (6.68e-2) +	4.0221e + 0 (5.29e-1) -	1.3643e + 0 (1.80e-1) -	1.6234e + 0 (2.79e-1) -	1.0421e + 0 (9.23e-2) -
	15	7.3420e + 1 (5.47e + 0) -	2.5634e + 0 (3.49e-1) +	9.3639e + 0 (4.49e-1) -	2.3337e + 0 (2.77e-1) +	6.7710e + 0 (7.66e-1) -	5.3225e + 0 (6.49e-1) -
WFG1	3	1.4675e + 0 (4.06e-2) -	6.6188e-1 (2.57e-1) -	1.4219e + 0 (3.30e-2) -	2.6434e-1 (2.28e-2) -	1.7001e + 0 (5.13e-2) -	1.9788e-1 (3.52e-2)
	5	2.0253e + 0 (3.44e-2) -	1.3552e + 0 (3.34e-1) -	1.9537e + 0 (3.67e-2) -	9.3878e-1 (5.05e-2) -	2.0769e + 0 (3.47e-2) -	4.4659e-1 (9.64e-3)
	8	2.4569e + 0 (1.01e-1) -	2.3562e + 0 (6.63e-1) -	2.5812e + 0 (5.52e-2) -	1.4892e + 0 (8.16e-2) -	2.8194e + 0 (6.00e-2) -	8.6276e-1 (3.60e-2)
	10	2.6362e + 0 (1.85e-1) -	4.4680e + 0 (1.50e + 0) -	2.8805e + 0 (8.48e-2) -	1.5853e + 0 (1.00e-1) -	3.1636e + 0 (5.25e-2) -	1.0433e + 0 (1.05e-1)
WFG2	15	3.4521e + 0 (2.52e-1) -	6.4781e + 0 (3.35e + 0) -	3.8207e + 0 (2.32e-1) -	2.4012e + 0 (1.02e-1) -	4.0465e + 0 (1.60e-1) -	1.8071e + 0 (1.75e-1)
	3	1.7763e-1 (4.86e-3) -	4.6350e-1 (5.69e-2) -	2.4868e-1 (1.61e-2) -	2.1280e-1 (1.13e-2) -	1.7107e-1 (5.82e-3) -	1.5005e-1 (1.30e-3)
	5	6.8453e-1 (2.90e-2) -	1.0627e + 0 (2.09e-1) -	5.8197e-1 (2.80e-2) -	7.8187e-1 (5.38e-2) -	4.7736e-1 (1.18e-2) -	4.6756e-1 (2.54e-3)

Table 2 (continued)

	Obj	CMOPSO	NMPSO	IDMOPSO	MMOPSO	MaOPSO/vPF	MOC SOP
	8	1.3218e + 0 (5.12e-2) =	1.9494e + 0 (2.90e-1) -	1.1585e + 0 (2.63e-2) =	1.3878e + 0 (4.70e-2) -	9.1589e-1 (1.91e-2) +	1.2056e + 0 (2.33e-1)
	10	1.5169e + 0 (4.01e-2) -	1.9216e + 0 (1.74e-1) -	1.2168e + 0 (2.74e-2) -	1.5886e + 0 (6.81e-2) -	1.0067e + 0 (3.97e-2) +	1.1238e + 0 (1.32e-1)
	15	2.4822e + 0 (1.69e-1) -	3.9262e + 0 (6.36e-1) -	1.9797e + 0 (2.39e-1) -	2.3471e + 0 (9.71e-2) -	3.3576e + 0 (7.34e-1) -	1.8050e + 0 (7.69e-2)
WFG3	3	1.4543e-1 (1.60e-2) -	4.0784e-2 (3.18e-3) +	9.9465e-2 (8.71e-3) -	8.1171e-2 (1.28e-2) -	3.6698e-1 (4.15e-2) -	8.0203e-2 (4.59e-2)
	5	8.3484e-1 (8.80e-2) +	2.7006e-1 (5.25e-2) +	6.3889e-1 (1.10e-1) +	4.3874e-1 (1.03e-1) +	6.3197e-1 (6.21e-2) +	3.1590e + 0 (8.43e-1)
	8	1.9682e + 0 (2.35e-1) -	6.5808e-1 (9.74e-2) =	8.5420e-1 (1.28e-1) -	1.0502e + 0 (1.84e-1) -	1.7454e + 0 (1.21e-1) -	7.3860e-1 (1.90e-1)
	10	2.7500e + 0 (2.68e-1) -	9.6980e-1 (2.15e-1) -	9.2472e-1 (1.28e-1) -	1.3064e + 0 (2.78e-1) -	2.3299e + 0 (1.11e-1) -	7.4511e-1 (1.62e-1)
	15	6.3037e + 0 (7.68e-1) -	1.1752e + 0 (2.44e-1) +	9.0567e-1 (1.89e-1) +	3.6874e + 0 (6.36e-1) -	4.6949e + 0 (2.32e-1) -	1.5926e + 0 (3.04e-1)
WFG4	3	2.5651e-1 (5.03e-3) -	2.9883e-1 (1.10e-2) -	3.2281e-1 (1.27e-2) -	2.7738e-1 (1.06e-2) -	2.5349e-1 (4.15e-3) -	2.0457e-1 (3.41e-4)
	5	1.1315e + 0 (2.43e-2) +	1.2671e + 0 (2.03e-2) -	1.2028e + 0 (2.66e-2) -	1.2088e + 0 (1.91e-2) -	1.1802e + 0 (1.07e-2) =	1.1755e + 0 (1.01e-3)
	8	3.0303e + 0 (4.88e-2) -	3.1247e + 0 (2.85e-2) -	3.1055e + 0 (3.36e-2) -	3.4159e + 0 (4.51e-2) -	3.0051e + 0 (2.47e-2) -	2.9628e + 0 (2.64e-3)
	10	4.2136e + 0 (4.69e-2) +	4.0918e + 0 (2.34e-2) +	4.1741e + 0 (2.26e-2) +	4.5833e + 0 (5.14e-2) -	4.3330e + 0 (4.91e-2) +	4.5249e + 0 (1.51e-2)
	15	8.9792e + 0 (1.13e-1)	8.5108e + 0 (1.74e-1) +	8.9335e + 0 (1.78e-1) +	8.9227e + 0 (1.17e-1) +	9.0470e + 0 (7.93e-2) +	9.3465e + 0 (2.06e-2)
WFG5	3	2.4571e-1 (7.62e-3) -	2.9087e-1 (1.24e-2) -	2.7089e-1 (8.68e-3) -	2.7034e-1 (1.01e-2) -	2.1542e-1 (6.80e-4) -	2.1451e-1 (1.00e-4)
	5	1.0884e + 0 (9.56e-3) +	1.1920e + 0 (1.83e-2) -	1.1330e + 0 (1.03e-2) +	1.2294e + 0 (2.97e-2) -	1.1257e + 0 (5.39e-3) +	1.1640e + 0 (3.93e-4)
	8	3.0275e + 0 (2.43e-2) -	3.0246e + 0 (2.67e-2) -	3.2189e + 0 (4.10e-2) -	3.4864e + 0 (8.88e-2) -	3.1405e + 0 (1.59e-1) -	2.9920e + 0 (1.46e-1)
	10	4.0454e + 0 (3.30e-2) +	4.0350e + 0 (2.87e-2) +	4.2490e + 0 (2.70e-2) +	4.6961e + 0 (7.08e-2) -	4.6259e + 0 (1.46e-1) -	4.5522e + 0 (1.25e-1)
	15	8.3698e + 0 (1.12e-1) +	9.1482e + 0 (7.13e-1) =	9.4949e + 0 (3.69e-1) =	8.9510e + 0 (1.31e-1) +	9.1975e + 0 (1.49e-1) =	9.2551e + 0 (6.00e-2)
WFG6	3	2.3064e-1 (3.75e-3) -	4.0333e-1 (9.61e-3) -	3.1478e-1 (1.31e-2) -	2.9179e-1 (4.39e-2) -	2.6190e-1 (3.42e-2) -	2.1935e-1 (3.73e-2)
	5	1.1717e + 0 (2.07e-2) -	1.2789e + 0 (1.81e-2) -	1.3220e + 0 (3.45e-2) -	1.3251e + 0 (8.39e-2) -	1.1836e + 0 (7.71e-3) -	1.1678e + 0 (2.75e-3)

Table 2 (continued)

Obj	CMOPSO	NMPSO	IDMOPSO	MMOPSO	MaOPSO/vPF	MOC SOP
8	3.2624e + 0 (5.73e-2) -	3.1393e + 0 (4.69e-2) -	3.3022e + 0 (3.46e-2) -	3.5864e + 0 (5.86e-2) -	3.0022e + 0 (1.08e-2) -	2.9855e + 0 (1.71e-1)
10	4.2649e + 0 (5.05e-2) +	4.1940e + 0 (6.59e-2) +	4.4575e + 0 (6.41e-2) +	4.6799e + 0 (4.95e-2) -	4.3679e + 0 (5.68e-2) +	4.5499e + 0 (6.99e-2)
15	8.7435e + 0 (1.33e-1) +	8.8590e + 0 (1.74e-1) +	9.9404e + 0 (8.50e-1) -	8.9388e + 0 (1.01e-1) +	9.0349e + 0 (1.51e-1) +	9.2770e + 0 (3.32e-2)
WFG7	2.2617e-1 (4.48e-3) -	3.0246e-1 (1.04e-2) -	3.0884e-1 (1.26e-2) -	2.6698e-1 (9.67e-3) -	2.2449e-1 (1.60e-3) -	2.0453e-1 (1.52e-4)
5	1.1770e + 0 (1.02e-2) =	1.2883e + 0 (2.03e-2) -	1.3473e + 0 (2.81e-2) -	1.2276e + 0 (2.87e-2) -	1.2010e + 0 (1.18e-2) -	1.1766e + 0 (4.23e-4)
8	3.1399e + 0 (1.04e-1) -	3.1111e + 0 (3.90e-2) -	3.2127e + 0 (3.22e-2) -	3.5792e + 0 (5.34e-2) -	3.0737e + 0 (5.38e-2) -	2.9593e + 0 (5.15e-3)
10	4.2730e + 0 (1.03e-1) +	4.1368e + 0 (3.45e-2) +	4.2679e + 0 (8.60e-2) +	4.6331e + 0 (5.77e-2) -	4.5244e + 0 (1.06e-1) =	4.5394e + 0 (9.49e-3)
15	8.5341e + 0 (8.45e-2) +	8.9105e + 0 (5.17e-1) +	9.9192e + 0 (8.39e-1) -	8.9799e + 0 (9.37e-2) +	9.3429e + 0 (4.05e-1) =	9.2826e + 0 (8.76e-2)
WFG8	3.2503e-1 (5.21e-3) -	3.4288e-1 (1.01e-2) -	3.7146e-1 (1.16e-2) -	3.5520e-1 (9.12e-3) -	3.1540e-1 (8.47e-3) -	2.7042e-1 (4.67e-3)
5	1.3976e + 0 (5.00e-2) -	1.2453e + 0 (1.21e-2) -	1.3473e + 0 (3.52e-2) -	1.4140e + 0 (4.67e-2) -	1.3696e + 0 (4.22e-2) -	1.1493e + 0 (4.14e-3)
8	3.6575e + 0 (4.84e-2) =	3.2209e + 0 (2.90e-2) +	3.5169e + 0 (4.34e-2) =	3.7626e + 0 (4.00e-2) =	3.4455e + 0 (7.71e-2) +	3.6591e + 0 (4.35e-1)
10	4.7229e + 0 (4.47e-2) +	4.4435e + 0 (6.16e-2) +	4.7605e + 0 (6.22e-2) +	4.8632e + 0 (4.83e-2) +	4.7527e + 0 (6.99e-2) +	5.0267e + 0 (2.83e-1)
15	9.0199e + 0 (1.39e-1) +	9.8996e + 0 (7.24e-1) =	1.1235e + 1 (4.99e-1) -	9.0352e + 0 (1.07e-1) +	9.5256e + 0 (3.27e-1) +	9.8038e + 0 (1.94e-1)
WFG9	2.1414e-1 (3.09e-3) -	3.3899e-1 (5.43e-2) -	2.6741e-1 (1.25e-2) -	2.7577e-1 (2.63e-2) -	2.1303e-1 (3.38e-3) -	2.0638e-1 (7.30e-4)
5	1.1370e + 0 (3.52e-2) =	1.1863e + 0 (2.20e-2) -	1.1442e + 0 (1.43e-2) -	1.2615e + 0 (2.29e-2) -	1.1393e + 0 (8.62e-3) =	1.1337e + 0 (7.12e-3)
8	3.2607e + 0 (6.69e-2) -	3.0329e + 0 (2.68e-2) -	3.1490e + 0 (2.86e-2) -	3.6983e + 0 (7.85e-2) -	3.1855e + 0 (2.08e-1) -	2.9550e + 0 (1.36e-1)
10	4.3422e + 0 (3.92e-2) +	4.0263e + 0 (2.44e-2) +	4.1829e + 0 (2.81e-2) +	4.8205e + 0 (8.25e-2) -	4.6716e + 0 (2.08e-1) -	4.4917e + 0 (1.71e-1)
15	8.7970e + 0 (9.14e-2) -	8.7483e + 0 (4.37e-1) =	9.2188e + 0 (3.44e-1) -	9.1382e + 0 (1.06e-1) -	9.1744e + 0 (2.52e-1) -	8.7224e + 0 (1.18e-1)
+/-≈	17/58/5	24/50/6	15/54/11	13/63/4	12/58/10	

The best result in each row are highlighted in bold

Table 3 Quantitative comparison of median HV values among 20 independent runs of various models ON DTLZ1- 7 AND WFG1- 9 WITH 3, 5, 8, 10 AND 15 objectives

Problem	Obj	Adaptive reference based	Decomposition based	Indicator based	New dominance based	Convergence and Diversity	Our Method
		A-NSGA-III	MOEA/D-CMA	MaOEA/IGD	NSGAI/SDR	VaEA	MOCSOP
DTLZ1	3	8.3515e-1 (1.01e-2) -	8.4295e-1 (4.56e-4) -	8.5314e-2 (1.41e-1) -	8.0606e-1 (1.73e-2) -	8.0844e-1 (3.35e-2) -	8.4438e-1 (1.30e-5)
	5	9.7469e-1 (5.84e-4) -	9.6740e-1 (1.10e-2) -	2.5824e-1 (3.69e-1) -	9.1046e-1 (1.96e-2) -	8.7054e-1 (3.31e-2) -	9.7498e-1 (1.47e-4)
	8	9.6783e-1 (8.82e-2) -	9.8828e-1 (1.07e-2) -	5.7979e-1 (4.40e-1) -	9.0843e-1 (3.38e-2) -	9.0425e-1 (3.87e-2) -	9.9662e-1 (4.43e-3)
	10	9.8942e-1 (2.57e-2) -	9.8179e-1 (6.12e-2) -	8.5141e-1 (2.48e-1) -	9.3979e-1 (2.45e-2) -	9.4119e-1 (9.39e-2) -	9.9936e-1 (6.71e-4)
	15	9.9975e-1 (7.73e-4) =	8.2832e-1 (9.40e-2) -	5.6103e-1 (4.02e-1) -	8.3002e-1 (1.04e-1) -	9.7124e-1 (1.41e-2) -	9.9992e-1 (7.17e-6)
DTLZ2	3	5.6065e-1 (3.12e-3) -	5.5955e-1 (4.33e-4) -	5.4585e-1 (1.17e-4) -	2.6201e-1 (2.94e-2) -	5.5804e-1 (1.25e-3) -	5.6301e-1 (2.25e-5)
	5	7.9468e-1 (4.44e-4) =	7.7429e-1 (1.78e-3) -	7.9378e-1 (1.14e-3) -	4.8680e-1 (1.66e-1) -	7.7640e-1 (2.37e-3) -	7.9482e-1 (4.77e-4)
	8	8.8624e-1 (3.63e-2) +	9.0677e-1 (5.66e-3) +	9.1934e-1 (2.81e-3) +	6.7048e-1 (1.76e-1) -	9.0779e-1 (2.91e-3) +	8.5430e-1 (3.72e-2)
	10	9.4565e-1 (1.74e-2) =	9.4535e-1 (9.34e-3) +	9.6882e-1 (8.46e-4) +	9.6257e-1 (2.63e-3) +	9.4672e-1 (2.32e-3) +	9.3819e-1 (1.48e-2)
	15	9.8143e-1 (7.98e-3) =	9.1961e-1 (1.48e-2) -	5.2330e-1 (1.38e-1) -	8.6792e-1 (1.36e-1) -	9.5773e-1 (1.96e-3) -	9.8260e-1 (7.31e-3)
DTLZ3	3	5.4659e-1 (8.45e-3) -	3.2216e-1 (2.71e-1) -	0.0000e+0 (0.00e+0) -	2.5851e-1 (1.45e-2) -	5.5142e-1 (8.29e-3) -	5.6302e-1 (1.99e-5)
	5	7.8615e-1 (7.68e-3) -	2.9371e-1 (3.41e-1) -	0.0000e+0 (0.00e+0) -	6.4314e-1 (1.83e-1) -	5.8602e-1 (6.79e-2) -	7.9001e-1 (3.79e-3)
	8	7.3380e-1 (8.36e-2) -	6.0547e-1 (3.30e-1) -	0.0000e+0 (0.00e+0) -	7.9533e-1 (1.76e-1) =	4.9001e-1 (3.17e-1) -	8.5769e-1 (9.40e-2)
	10	8.5520e-1 (4.73e-2) =	5.1071e-1 (3.34e-1) -	0.0000e+0 (0.00e+0) -	9.6322e-1 (2.63e-3) +	2.0843e-4 (9.32e-4) -	7.8817e-1 (2.82e-1)
	15	9.3167e-1 (3.96e-2) =	1.4400e-1 (2.13e-1) -	0.0000e+0 (0.00e+0) -	9.0331e-1 (1.03e-1) =	6.9345e-2 (1.12e-1) -	9.3463e-1 (3.47e-2)
DTLZ4	3	5.4008e-1 (6.82e-2) +	5.4623e-1 (2.56e-2) +	4.7509e-1 (9.89e-2) +	3.0415e-1 (6.98e-2) -	5.5727e-1 (1.04e-3) +	3.5787e-1 (1.68e-1)
	5	7.9454e-1 (5.63e-4) +	7.5227e-1 (1.17e-2) =	7.7230e-1 (4.06e-2) +	3.1919e-1 (5.63e-2) -	7.7225e-1 (2.69e-3) =	7.2349e-1 (8.26e-2)
	8	9.2005e-1 (1.59e-2) +	9.0996e-1 (4.95e-3) =	9.2211e-1 (8.86e-3) +	4.3051e-1 (5.84e-2) -	9.0023e-1 (4.36e-3) =	9.0363e-1 (3.07e-2)
	10	9.5808e-1 (1.84e-2) =	9.6269e-1 (1.83e-3) -	9.6920e-1 (3.51e-3) -	5.4597e-1 (5.44e-2) -	9.4093e-1 (5.33e-3) -	9.6950e-1 (2.19e-4)
	15	9.7785e-1 (9.12e-3) -	9.8593e-1 (1.45e-3) -	9.8422e-1 (6.50e-3) =	7.5782e-1 (4.96e-2) -	9.6403e-1 (2.82e-3) -	9.8639e-1 (8.01e-3)
DTLZ5	3	1.9580e-1 (9.58e-4) +	1.9258e-1 (2.27e-5) =	7.0358e-2 (4.18e-2) -	1.8386e-1 (2.24e-3) -	1.9966e-1 (1.50e-4) +	1.8859e-1 (6.69e-3)

Table 3 (continued)

Problem	Obj	Adaptive reference based	Decomposition based	Indicator based	New dominance based	Convergence and Diversity	Our Method
DTLZ6	5	1.0680e−1 (7.92e−3) −	9.2697e−2 (3.29e−4) −	9.9195e−2 (1.25e−4) −	1.0984e−1 (3.67e−3) −	9.7948e−2 (5.10e−3) −	1.1378e−1 (4.65e−3)
	8	9.4843e−2 (2.55e−3) −	9.9523e−2 (1.65e−4) +	8.7573e−2 (2.06e−2) −	9.2440e−2 (9.64e−4) −	9.0822e−2 (2.83e−4) −	9.7419e−2 (1.55e−3)
	10	8.9958e−2 (2.11e−3) −	9.6406e−2 (2.10e−4) +	9.1602e−2 (1.98e−4) −	9.0943e−2 (2.68e−4) −	9.0768e−2 (1.07e−4) −	9.5220e−2 (1.28e−3)
	15	9.1226e−2 (5.22e−4) −	9.2011e−2 (1.94e−4) =	9.1221e−2 (1.18e−4) −	9.1033e−2 (3.23e−4) −	9.0886e−2 (7.34e−5) −	9.1898e−2 (6.38e−4)
	3	1.9501e−1 (9.57e−4) +	1.9268e−1 (8.73e−6) +	6.2820e−2 (4.12e−2) −	1.7434e−1 (6.68e−3) −	1.9996e−1 (7.73e−5) +	1.9161e−1 (1.03e−3)
	5	9.3766e−2 (4.56e−3) −	9.2791e−2 (3.22e−4) −	6.1963e−2 (4.68e−2) −	1.0962e−1 (3.79e−3) =	9.1402e−2 (2.00e−3) −	1.0998e−1 (6.29e−3)
	8	8.6378e−2 (2.03e−2) −	9.9596e−2 (2.02e−4) +	6.5763e−2 (3.82e−2) −	9.1163e−2 (5.81e−4) −	6.3700e−2 (4.28e−2) −	9.3860e−2 (2.88e−3)
	10	8.1772e−2 (2.80e−2) −	9.6457e−2 (2.22e−4) +	8.2523e−2 (2.82e−2) =	9.1084e−2 (3.28e−4) −	2.2726e−2 (4.04e−2) −	9.2042e−2 (1.08e−3)
	15	9.0965e−2 (3.77e−4) −	9.2039e−2 (1.91e−4) +	8.6676e−2 (2.04e−2) =	9.0948e−2 (2.62e−4) −	9.0886e−2 (3.20e−4) −	9.1378e−2 (5.67e−4)
	3	2.7304e−1 (1.93e−3) +	2.6337e−1 (8.09e−4) =	1.3775e−1 (6.17e−2) −	2.7168e−1 (1.18e−3) =	2.7593e−1 (8.18e−3) +	2.5009e−1 (2.96e−2)
	5	2.3847e−1 (7.18e−3) −	5.3815e−2 (3.68e−2) −	1.5157e−1 (4.14e−2) −	2.4586e−1 (3.46e−3) −	2.3516e−1 (4.55e−3) −	2.4847e−1 (1.78e−2)
	8	2.0261e−1 (3.59e−3) =	3.1171e−2 (4.89e−2) −	1.1187e−1 (1.20e−2) −	1.7437e−1 (3.25e−2) −	1.6855e−1 (5.66e−3) −	2.0316e−1 (3.50e−3)
	10	1.8510e−1 (6.57e−3) =	1.6310e−2 (3.74e−2) −	5.8181e−2 (2.75e−2) −	1.5803e−1 (3.62e−2) −	1.3905e−1 (4.65e−3) −	1.8699e−1 (2.96e−3)
	15	1.1831e−1 (1.35e−2) −	1.0398e−1 (4.62e−4) −	2.7614e−2 (1.45e−2) −	1.1448e−3 (3.60e−3) −	1.0071e−1 (3.16e−3) −	1.2625e−1 (2.93e−3)
	3	9.4141e−1 (2.74e−3) +	7.2111e−1 (5.87e−2) −	1.6908e−1 (4.76e−2) −	9.2271e−1 (5.84e−3) +	9.3870e−1 (1.43e−3) +	9.0818e−1 (1.69e−2)
5	9.9633e−1 (5.33e−3) +	9.7139e−1 (9.27e−3) −	2.1356e−1 (4.52e−2) −	9.7419e−1 (9.44e−3) −	9.7940e−1 (2.37e−2) =	9.8095e−1 (1.83e−2)	
8	9.9928e−1 (3.00e−4) −	7.0391e−1 (9.51e−2) −	2.4387e−1 (6.80e−2) −	9.8495e−1 (1.05e−2) −	9.9415e−1 (1.64e−2) −	9.9989e−1 (9.55e−5)	
10	9.9925e−1 (2.60e−4) −	5.7805e−1 (8.86e−2) −	3.2109e−1 (5.81e−2) −	9.8902e−1 (1.45e−2) −	8.5733e−1 (5.13e−2) −	9.9980e−1 (2.09e−4)	
15	1.0000e + 0 (9.31e−6) =	9.9527e−1 (9.71e−3) −	3.0483e−1 (7.09e−2) −	9.8495e−1 (1.30e−2) −	1.0000e + 0 (2.37e−6) =	9.9997e−1 (9.49e−5)	
3	9.3010e−1 (1.42e−3) −	9.0239e−1 (1.20e−2) −	5.5834e−1 (9.54e−2) −	9.1477e−1 (6.03e−3) −	9.2663e−1 (1.62e−3) −	9.3304e−1 (8.36e−4)	
5	9.9580e−1 (9.16e−4) +	9.8793e−1 (8.09e−3) −	8.8827e−1 (3.71e−2) −	9.6384e−1 (5.91e−3) −	9.9103e−1 (1.26e−3) −	9.9418e−1 (3.80e−3)	
8	9.9714e−1 (1.83e−3) −	9.7749e−1 (1.07e−2) −	9.5425e−1 (6.30e−2) −	9.7671e−1 (5.47e−3) −	9.9635e−1 (1.02e−3) −	9.9840e−1 (1.45e−3)	
10	9.9711e−1 (1.48e−3) −	9.7956e−1 (8.51e−3) −	9.8174e−1 (4.43e−2) −	9.8602e−1 (3.88e−3) −	9.9550e−1 (1.08e−3) −	9.9962e−1 (1.92e−4)	
15	9.9877e−1 (9.70e−4) −	9.6767e−1 (2.43e−2) −	9.1372e−1 (8.69e−2) −	9.8453e−1 (6.20e−3) −	9.9819e−1 (6.93e−4) −	9.9999e−1 (1.21e−5)	
3	3.9425e−1 (5.49e−3) −	3.7899e−1 (1.75e−3) −	8.2905e−2 (2.40e−2) −	3.8630e−1 (1.47e−2) −	3.7306e−1 (4.69e−3) −	4.0469e−1 (4.87e−3)	

Table 3 (continued)

Problem	Obj	Adaptive reference based	Decomposition based	Indicator based	New dominance based	Convergence and Diversity	Our Method	
WFG4	5	1.5119e−1 (3.07e−2) −	9.1183e−2 (3.20e−4) −	8.0990e−2 (7.92e−3) −	2.0529e−1 (1.55e−2) −	1.2926e−1 (1.84e−2) −	2.2276e−1 (2.92e−2)	
	8	7.4832e−2 (1.15e−2) −	8.8073e−2 (7.50e−3) −	1.9227e−2 (2.01e−2) −	7.1934e−2 (1.60e−2) −	8.3947e−2 (6.13e−3) −	1.5248e−1 (1.74e−2)	
	10	8.2630e−3 (1.38e−2) −	8.3533e−2 (8.35e−3) −	1.6455e−2 (2.15e−2) −	4.7022e−3 (7.17e−3) −	3.6423e−2 (1.84e−2) −	1.2782e−1 (1.28e−2)	
	15	0.0000e + 0 (0.00e + 0) −	5.0802e−2 (3.54e−2) =	0.0000e + 0 (0.00e + 0) −	0.0000e + 0 (0.00e + 0) −	1.7160e−3 (4.93e−3) −	6.3708e−2 (3.43e−2)	
	3	5.3259e−1 (3.34e−3) −	5.0547e−1 (1.66e−2) −	1.0712e−1 (3.38e−2) −	5.5077e−1 (1.77e−3) −	5.5176e−1 (2.48e−3) −	5.5629e−1 (9.72e−3)	
	5	7.8819e−1 (2.61e−3) =	5.6231e−1 (4.57e−2) −	9.8550e−2 (3.53e−2) −	7.6996e−1 (3.58e−3) −	7.6056e−1 (3.44e−3) −	7.8736e−1 (1.14e−3)	
	8	8.5585e−1 (7.49e−3) −	7.1269e−1 (3.15e−2) −	1.1143e−1 (3.65e−2) −	9.1105e−1 (3.30e−3) −	8.9272e−1 (5.82e−3) −	9.1470e−1 (1.34e−3)	
	10	9.1632e−1 (1.73e−2) −	7.7755e−1 (1.63e−2) −	1.1528e−1 (3.82e−2) −	9.5829e−1 (1.53e−3) +	9.2094e−1 (5.25e−3) −	9.5215e−1 (1.85e−3)	
	15	9.8774e−1 (4.34e−3) −	7.7479e−1 (3.40e−2) −	1.3055e−1 (5.97e−2) −	9.7838e−1 (2.41e−3) −	9.5416e−1 (4.59e−3) −	9.8808e−1 (7.37e−4)	
	WFG5	3	5.0688e−1 (2.06e−3) −	4.7383e−1 (4.17e−3) −	3.0985e−1 (1.83e−1) −	5.1544e−1 (1.44e−3) −	5.1746e−1 (1.28e−3) −	5.2168e−1 (1.18e−4)
		5	7.4008e−1 (2.40e−3) −	5.5026e−1 (3.71e−2) −	9.3956e−2 (3.39e−2) −	7.2731e−1 (3.69e−3) −	7.2239e−1 (3.23e−3) −	7.4348e−1 (3.69e−4)
		8	8.2354e−1 (6.33e−3) −	6.2343e−1 (3.01e−2) −	9.2631e−2 (2.52e−2) −	8.5300e−1 (5.61e−3) −	8.4021e−1 (2.33e−3) −	8.5369e−1 (2.26e−2)
		10	8.6882e−1 (3.29e−3) −	6.9082e−1 (2.52e−2) −	1.8628e−1 (2.32e−1) −	8.9771e−1 (1.85e−3) −	8.7069e−1 (3.85e−3) −	8.9859e−1 (1.29e−2)
		15	9.1678e−1 (3.63e−3) −	6.6081e−1 (3.32e−2) −	1.0111e−1 (3.70e−2) −	9.0941e−1 (2.52e−3) −	8.7803e−1 (3.11e−3) −	9.1729e−1 (1.31e−4)
	WFG6	3	4.9520e−1 (9.98e−3) −	4.9281e−1 (3.67e−2) −	2.0183e−1 (1.37e−1) −	5.0422e−1 (1.77e−2) −	5.0164e−1 (8.83e−3) −	5.2845e−1 (3.83e−2)
5		7.2235e−1 (1.59e−2) =	5.9705e−1 (1.24e−1) −	1.8175e−1 (1.04e−1) −	7.1314e−1 (1.68e−2) =	7.0423e−1 (1.46e−2) =	6.6252e−1 (6.93e−2)	
8		7.9300e−1 (3.11e−2) =	6.7922e−1 (7.35e−2) −	2.5249e−1 (1.75e−1) −	8.4355e−1 (2.33e−2) +	8.2976e−1 (1.27e−2) =	7.7303e−1 (7.78e−2)	
10		8.4905e−1 (1.59e−2) +	7.1369e−1 (7.33e−2) −	3.0262e−1 (2.27e−1) −	8.8343e−1 (1.20e−2) +	8.5517e−1 (1.58e−2) +	7.9172e−1 (7.93e−2)	
15		8.9087e−1 (2.62e−2) =	6.6379e−1 (6.69e−2) −	3.7987e−1 (2.34e−1) −	7.9825e−1 (1.84e−1) =	8.7404e−1 (1.97e−2) =	8.5001e−1 (9.17e−2)	
WFG7	3	5.3832e−1 (3.51e−3) −	5.3415e−1 (1.56e−2) −	1.8221e−1 (7.31e−2) −	5.5477e−1 (1.43e−3) −	5.4970e−1 (2.13e−3) −	5.5990e−1 (3.21e−4)	

Table 3, the proposed method yields superior performance on most of the test instances. Although the proposed algorithm produces the suboptimal results on some concave problems, it performs well on the test instances containing complex, degenerate and disconnected Pareto fronts. For instance, our method generates the best HV values on DTLZ1, WFG5 and WFG7 and produces competitive results on DTLZ2, DTLZ3

and WFG2. It is widely recognized that the PSO operators do not perform well on WFG1 [36]. However, MOCSOP obtains the competitive HV values on WFG1, which means that the proposed method ensures the diversity of the generated solutions and avoids to local optima. Moreover, the MOCSOP performs a constant good performance couple with the increasing of the number of objectives in MaOPs.

Table 3 (continued)

Problem	Obj	Adaptive reference based	Decomposition based	Indicator based	New dominance based	Convergence and Diversity	Our Method	
WFG8	5	7.8646e−1 (2.69e−3) −	7.4730e−1 (1.90e−2) −	2.1400e−1 (5.33e−2) −	7.7702e−1 (2.79e−3) −	7.7015e−1 (3.43e−3) −	7.9035e−1 (8.05e−4)	
	8	8.6850e−1 (7.89e−3) −	7.2689e−1 (2.56e−2) −	2.1328e−1 (5.80e−2) −	9.0740e−1 (6.59e−3) −	9.0490e−1 (2.88e−3) −	9.1897e−1 (9.24e−4)	
	10	9.1748e−1 (8.71e−3) −	7.8972e−1 (2.23e−2) −	2.0461e−1 (7.43e−2) −	9.6146e−1 (2.31e−3) =	9.4563e−1 (2.30e−3) −	9.6225e−1 (1.54e−3)	
	15	9.8162e−1 (6.64e−3) =	7.6172e−1 (3.39e−2) −	1.6754e−1 (4.78e−2) −	9.5275e−1 (8.97e−2) −	9.6337e−1 (2.89e−3) −	9.8591e−1 (5.06e−3)	
	3	4.5299e−1 (6.26e−3) −	4.2727e−1 (2.91e−3) −	2.7114e−2 (5.07e−2) −	4.6840e−1 (1.87e−3) −	4.6536e−1 (3.13e−3) −	4.7479e−1 (2.69e−3)	
	5	6.3239e−1 (1.36e−2) −	2.5882e−1 (1.43e−2) −	4.8700e−2 (5.08e−2) −	6.5411e−1 (6.27e−3) −	6.3129e−1 (9.01e−3) −	6.8349e−1 (1.64e−3)	
	8	7.6734e−1 (2.01e−2) −	5.4002e−1 (2.83e−2) −	1.7448e−1 (5.29e−2) −	8.1175e−1 (3.17e−2) +	7.2140e−1 (1.35e−2) −	7.8165e−1 (1.41e−2)	
	10	8.5147e−1 (1.94e−2) −	6.0872e−1 (2.34e−2) −	2.1895e−1 (6.04e−2) −	9.2902e−1 (2.56e−2) +	7.9615e−1 (2.10e−2) −	8.6663e−1 (6.80e−3)	
	15	9.3018e−1 (2.14e−2) +	5.2797e−1 (3.72e−2) −	1.7331e−1 (6.03e−2) −	8.2870e−1 (1.01e−1) =	8.3167e−1 (1.54e−2) −	9.1593e−1 (5.67e−3)	
	WFG9	3	5.0308e−1 (3.80e−2) −	4.8777e−1 (5.92e−3) −	2.0248e−1 (4.63e−2) −	5.3992e−1 (2.92e−3) =	5.2550e−1 (2.75e−2) −	5.4006e−1 (1.71e−3)
		5	7.2880e−1 (3.35e−2) −	4.2003e−1 (7.11e−2) −	1.8381e−1 (4.42e−2) −	7.4414e−1 (5.26e−3) +	6.9484e−1 (5.56e−2) −	7.4126e−1 (3.14e−2)
		8	7.8446e−1 (4.80e−2) −	5.9512e−1 (3.24e−2) −	2.4657e−1 (1.20e−1) −	8.6871e−1 (4.68e−3) +	8.0052e−1 (5.16e−2) −	8.2808e−1 (6.57e−2)
		10	8.5827e−1 (3.62e−2) −	6.5132e−1 (3.66e−2) −	2.2908e−1 (1.41e−1) −	9.1609e−1 (4.61e−3) +	8.4365e−1 (3.36e−2) −	8.8849e−1 (3.64e−2)
		15	9.3329e−1 (1.36e−2) +	5.7310e−1 (6.17e−2) −	1.9430e−1 (1.27e−1) −	9.0208e−1 (7.15e−2) +	8.0339e−1 (7.12e−2) −	8.5970e−1 (9.93e−2)
	+/-/≈	13/52/15	9/65/6	5/72/3	12/59/9	8/65/7		

The best result in each row are highlighted in bold

This indicates that MOCSOP is a competitive model for handling MaOPs. For a visual comparison of the evaluated approaches, Fig. 9 plots the final nondominated solutions with the median HV value among 20 independent runs of various models on the 10-objective DTLZ1 instance by parallel coordinates. As shown in Fig. 9, both MOEA/D-CMA and VaEA show worse performance in terms of convergence. Due to the effective reference point adaptation strategy, ANSGA-III shows very competitive diversity performance, but it does not converge well the entire PF. Although solutions of NSGA-II/SDR reach the PF region, they fail to cover all the objectives due to modified Pareto-dominance that may have a negative effect on guidance. Furthermore, the solution sets obtained by MaOEA/IGD have shown good distribution on DTLZ1, but Table 3 shows that they have smaller HV value than MOCSOP, which indicate that they may not actually reach the true PF. As shown in Fig. 9 and Table 3, MOC-

SOP is the only algorithm that achieves good diversity and impartial convergence to the Pareto front on DTLZ1.

To further analyze the robustness of each algorithm, we introduce performance scores [53] to evaluate the overall performance of the compared algorithms. Specifically, the performance score shows how many other algorithms are significantly better than the selected algorithm on the considered problem instance. Figure 10 summarizes the average performance scores for the different numbers of objectives and the different test problems. A smaller value means better performance of the algorithm. As shown in Fig. 10, MOCSOP performs best overall in 10 out of 16 test problems, which demonstrates that the MOCSOP has excellent performance in all test instances. For further observations, the line chart of MOCSOP does not show huge fluctuations compared to other

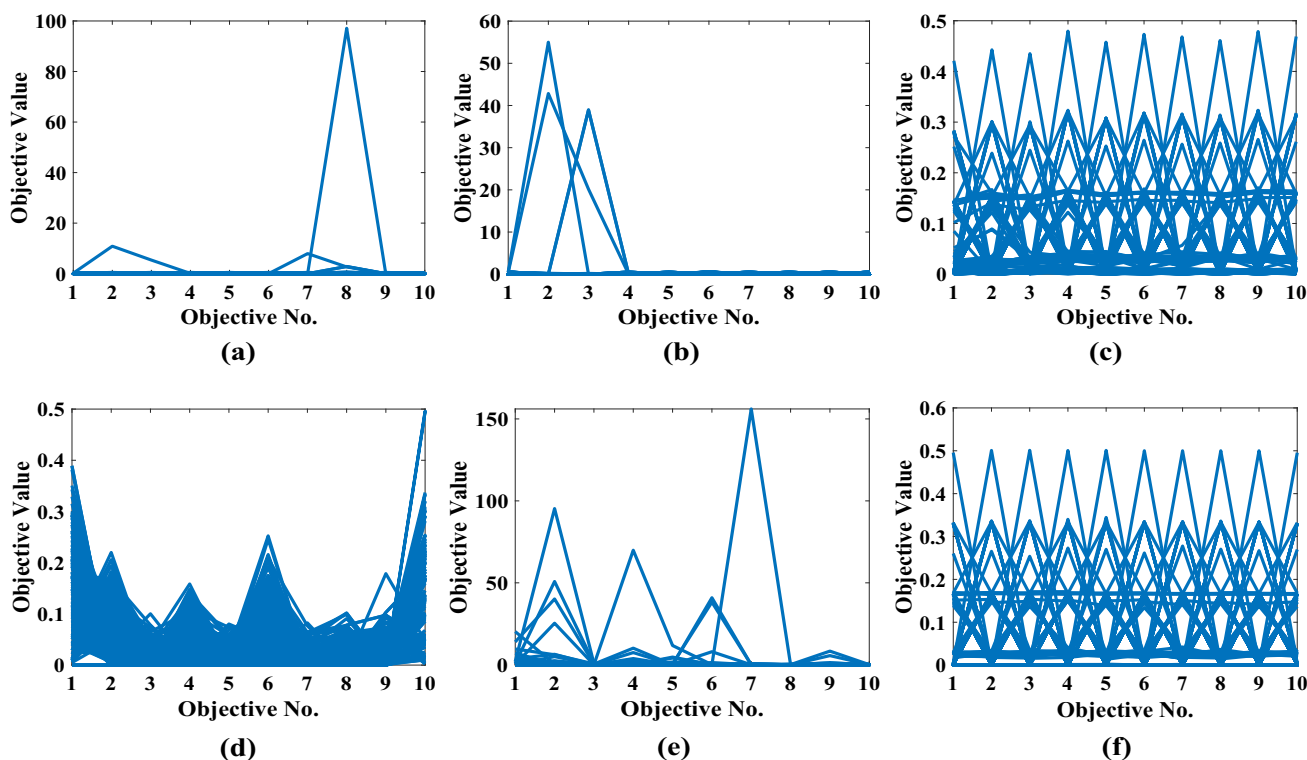


Fig. 9 Nondominated solution set with the average HV values among 20 runs on the 10-objective DTLZ1, shown by parallel coordinates. **a** A-NSGAIII, **b** MOEA/D-CMA, **c** MaOEA/IGD, **d** NSGA-II/SDR, **e** VaEA, **f** MOCSOP

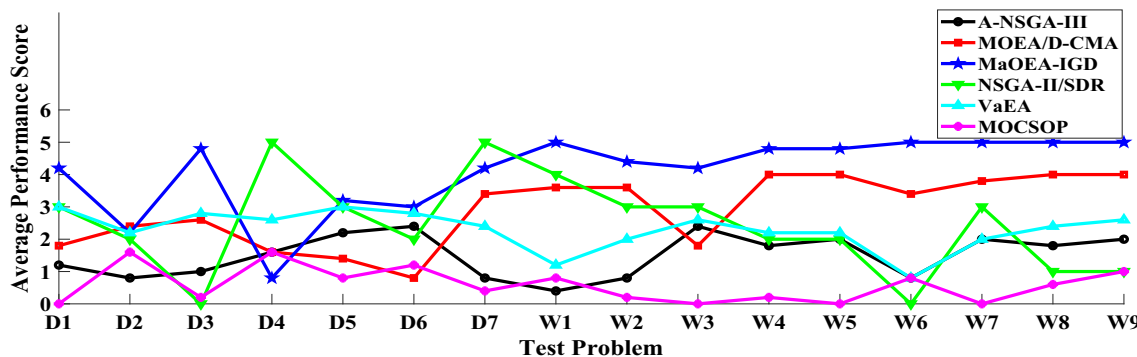


Fig. 10 Average performance score over all dimensions for different normalized test problems, namely DTLZ (Dx) and WFG (Wx). The smaller the score, the better the PF approximation in terms of HV metric

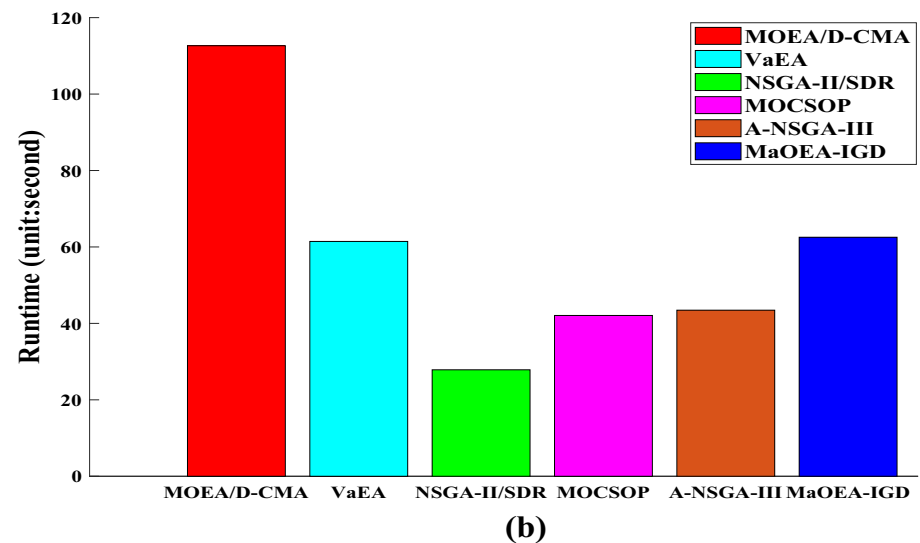
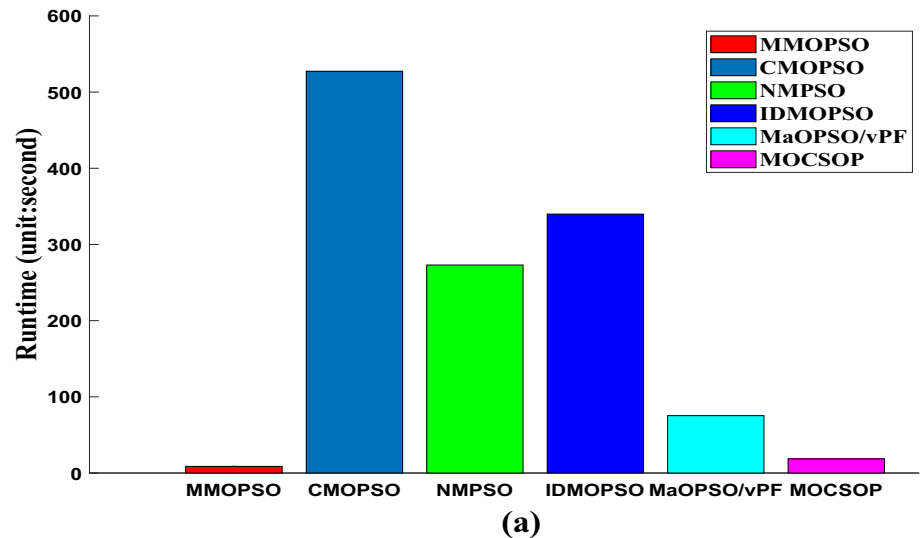
methods, which shows that our model has good robustness and strong search ability on different PF shapes.

Runtimes

To investigate the computational efficiency of MOCSOP, we record the actual running time of those nine compared algorithms on DTLZ1-DTLZ7 and WFG1-WFG9. To make a comprehensive comparison, all algorithms were implemented in an identical running platform (Matlab2019a). Figure 11a shows the average runtimes of the evaluated MOPSOs tested on all instances with 8 objectives.

In this figure, we can find that the MMOPSO achieves the best performance in terms of computational efficiency because the simple swarm leader selection strategy has a significant benefit in real-time computation. However, the proposed MOCSOP produces the second best result and performs better on the metrics of IGD value compared with MMOPSO. It is worth noting that, although both CMOPSO and MOCSOP use the competition mechanism, MOCSOP performs significantly better in terms of computational efficiency relative to CMOPSO. This is due to the fact that, in MOCSOP, the worst computational complexity of the proposed probability estimation method is

Fig. 11 Average runtimes of the various evaluated approaches tested on the all the benchmark instances. **a** 8-objective. **b** 15-objective



$O(MN^2)$. Additionally, the implementation of competition mechanism with winner pool is also simple, with negligible computational cost. By contrast, CMOCSO requires $O(MN^2)$ and $O(MN \log N)$ to calculate the nondominated sorting and crowding distance, respectively. For competition mechanism, two operations, selection of leader and the angle between the vectors are calculated, costing $O(MN)$. As a result, the competition mechanism of CMOCSOP requires a maximum total runtime of $O(MN^2) + O(MN \log N) + O(MN)$. As can be further observed from Fig. 11b, proposed MOCSOP presents a competitive performance in terms of computational efficiency with the other state-of-the-art MOEAs.

Discussions on probability estimation method

To further observe the differences between leaders obtained by our method and other fitness assignment methods, we use

two common fitness estimation methods to select leaders. The first fitness assignment method considers the L2-norm of objective value [16], which can be formulated as follows:

$$Fitness(x) = \sqrt{\sum_{i=1}^m f_i(x)^2}. \quad (10)$$

The second fitness estimator is calculated by the sum of all the objectives [24] as:

$$Fitness(x) = \sum_{i=1}^m f_i(x), \quad (11)$$

where $f_k(x)$ denotes the k -th objective value of x . Figure 12 presents an example to show the position of leaders by forming winner pool with the above fitness estimator. As shown in Fig. 12a, in the early stages, the leaders are scattered in the

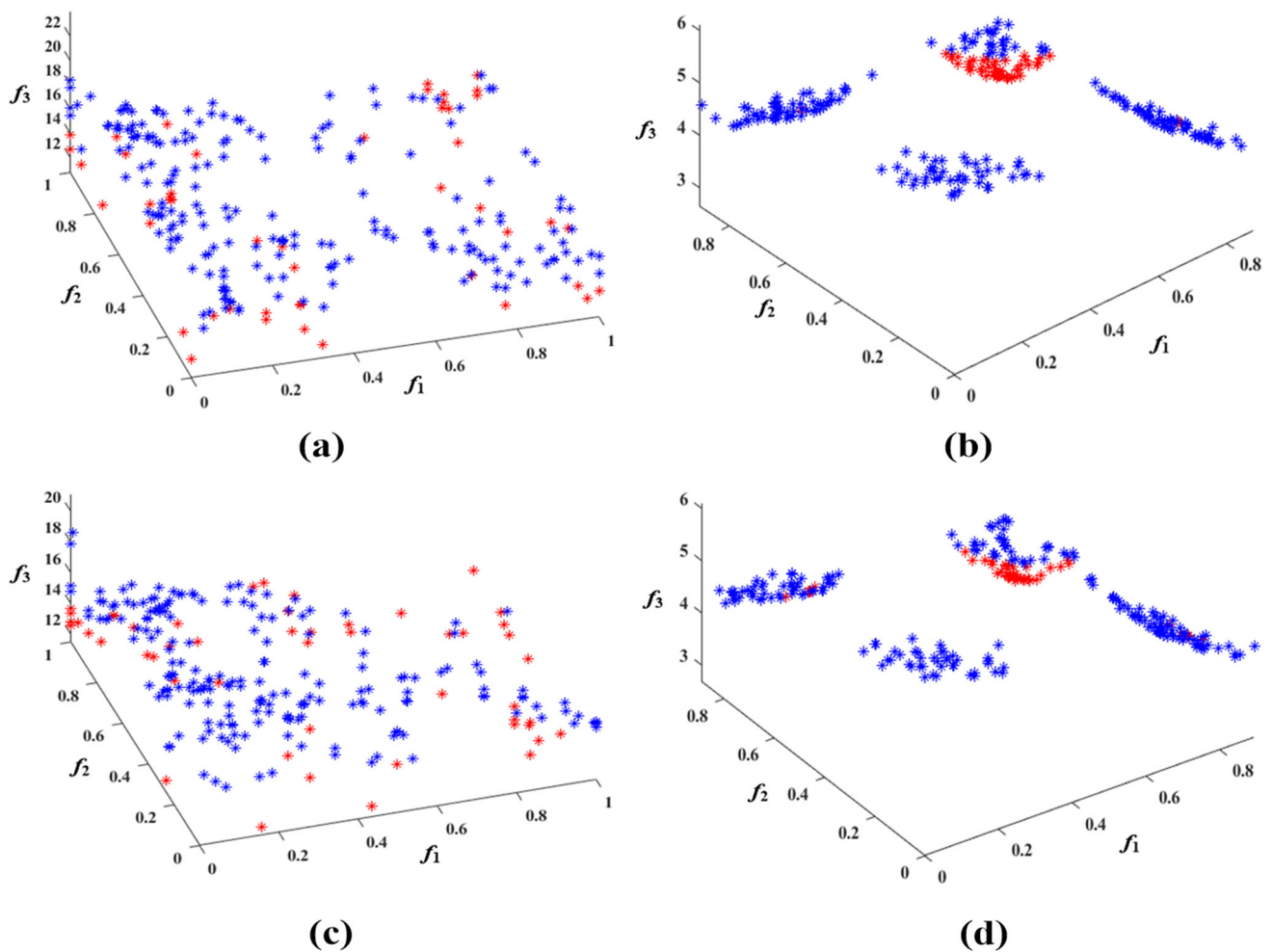


Fig. 12 Illustrative example to show the position of leaders for DTLZ7 with three-objectives, where the population size is set to 275 (The red points represent the leaders and blue points represent the other particles in the swarm). **a** Solution set obtained by the first fitness estimator

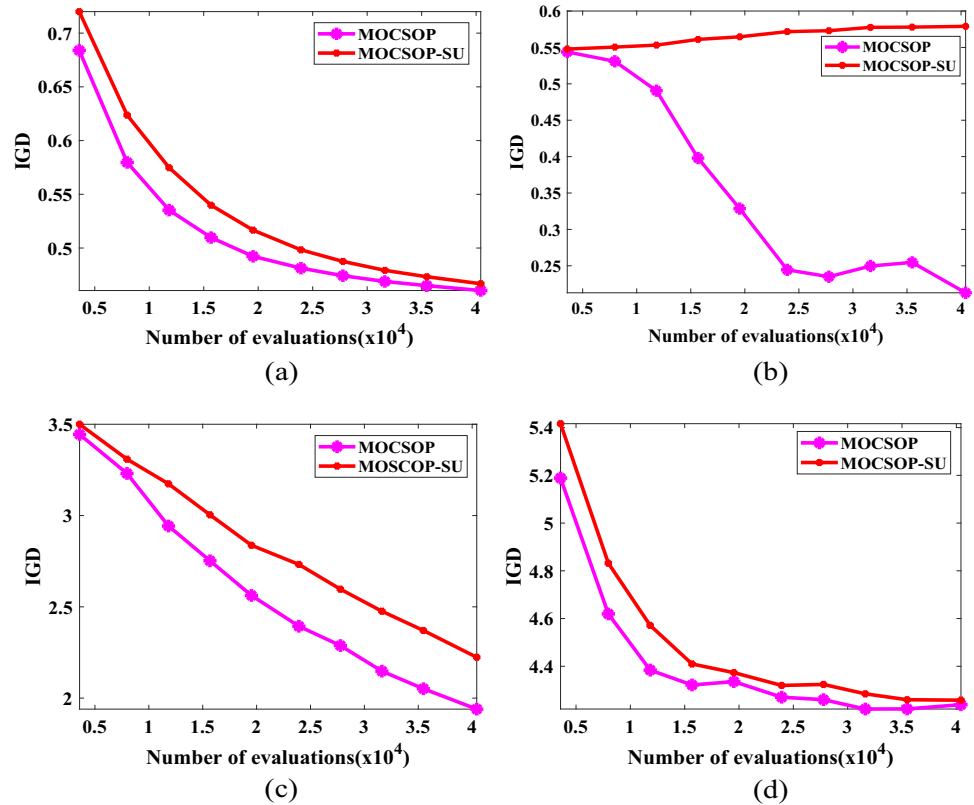
after 759 FEs. **b** solution set obtained by the first fitness estimator after 12,903 FEs. **c** Solution set obtained by the second fitness estimator after 759 FEs. **d** Solution set obtained by the second fitness estimator after 12,903 FEs

objective space. However, the position of leaders obtained by the proposed probability estimation method is closer to the ideal point, as illustrated in Fig. 4a. This means that the leaders obtained by the proposed method can guide the entire population towards the true PF in the early stages. As can be further observed from Fig. 12b, with the nondominated solutions increase, the leaders only focus on local regions of the PF instead of entire PF. On the contrary, the leaders are distributed over the entire PF in the Fig. 4b. Similarly, the second fitness estimator encounters the same problem. In summary, the proposed probability estimation method can adaptively adjust the position of leaders at different stages and guide the entire population towards the Pareto front.

Ablation experiment

To demonstrate the benefits of the proposed swarm update strategy in convergence efficiency, we utilize four 10-objective instances to conduct an ablation experiment. Figure 13 plots the evolutionary trajectories of IGD values obtained by MOCSOP and MOCSOP-SU averaged over 20 runs. In Fig. 13, the MOCSOP-SU model denotes the MOCSOP method without the proposed swarm update strategy. As shown in Fig. 13, for problems with concave PF (DTLZ4), both MOCSOP and MOCSOP-SU can guide the entire particle swarm to converge to the PF quickly, and they finally obtain similar IGD values. However, for DTLZ6 with degenerate PF, using swarm update strategy improves the speed of convergence significantly. The comparison results between MOCSOP and MOCSOP-SU indicate that the proposed swarm update strategy is beneficial for improving the

Fig. 13 Evolutionary trajectories of IGD values obtained by MOCSOP and its variant on four 10-objective test problems. (a) DTLZ4. (b) DTLZ6. (c) WFG1. (d) WFG9



performance of convergence speed, especially in MaOPs with complex PF.

Discussion of free parameters

In the proposed MOCSOP, the size of the winner pool has certain influence on the performance, because we use the winner pool to guide the particle swarm toward true PF. To achieve the desired nondominated solution set, the size of the winner pool needs to be given reasonable values. In the evolutionary process, we expect that most of the particles in the population are to be updated. If the size of the winner pool is too large, the number of the updated particles in the particle swarm will be too small. This may decrease the performance of the proposed method. Therefore, we suggest that the size of the winner pool should be set small. We choose 5%, 10%, 15%, 20% as the candidate values for experimental comparative analysis. To investigate the influence of the choice of different values on the performance of the algorithm, we use the DTLZ as the benchmark test suites. Table 4 shows four results of MOCSOP that correspond to different choices of the size of the winner pool on DTLZ1-ZDT7 with six objectives. Among them, MOCSOP-1, MOCSOP, MOCSOP-2 and MOCSOP-3 represent the size of the winner pools as 5, 10, 15 and 20 percent of the population size, respectively. To allow a fair comparison, the other parameters are remained unchanged.

As seen in Table 4, we find that MOCSOP significantly outperforms MOCSOP-1, MOCSOP-2 and MOCSOP-3 in 5 out of 7 instances. In particular, MOCSOP has achieved relatively good results on DTLZ5. This is mainly because the inappropriate candidate values increase the randomness of the competition, which affects the performance of the algorithm inevitably on irregular problems. This phenomenon is also observed on DTLZ3. Therefore, based on the statistical results in terms of the obtained IGD values on the DTLZ test suite, we recommend setting the size of winner pool to 10 percent of the population size, even though the selected values of the free parameters may not be the best choices for other evaluation datasets.

Further discussion

During the experiment, we find that MOCSOP has shown promising potential in large-scale MOPs. Table 5 exhibits the median IGD values obtained by MOCSOP and other state-of-the-art MOEAs on three-objective LSMOP1-LSMOP9 [20] with 300 decision variables, where the population size is set to 496. For fair comparisons, all compared algorithms are individually executed 20 independent times with 4,500,000 FEs. It is worth noting that the LMEA [54] and S^3 -CMA-ES [55] are two typical MOEAs for large-scale MOPs, which have shown good performance on large-scale MOPs. As shown

Table 4 The median values of IGD obtained by different sizes of winner pools on DTLZ1- DTLZ7 with 6 objectives

Problem	Obj	MOCSOP-1	MOCSOP	MOCSOP-2	MOCSOP-3
DTLZ1	6	8.3902e−2 (8.70e−3)	8.1109e−2 (5.38e−5)	8.1097e−2 (7.97e−5)	8.1122e−2 (5.32e−5)
DTLZ2	6	3.3216e−1 (1.02e−1)	3.2999e−1 (9.75e−2)	3.4635e−1 (1.10e−1)	3.4997e−1 (1.50e−1)
DTLZ3	6	3.5508e−1 (4.35e−1)	2.5729e−1 (2.51e−3)	2.7088e−1 (5.98e−2)	2.9007e−1 (1.48e−1)
DTLZ4	6	3.7955e−1 (1.25e−1)	2.7523e−1 (5.91e−2)	3.5368e−1 (1.03e−1)	3.5819e−1 (1.22e−1)
DTLZ5	6	9.9070e−2 (2.04e−2)	9.1844e−2 (2.54e−2)	9.7820e−2 (2.36e−2)	1.1346e−1 (3.72e−2)
DTLZ6	6	2.5947e−1 (7.83e−2)	2.0725e−1 (6.62e−2)	2.2787e−1 (7.41e−2)	2.2481e−1 (7.50e−2)
DTLZ7	6	9.3145e−1 (4.00e−1)	7.8771e−1 (1.45e−1)	7.6855e−1 (1.55e−1)	7.8917e−1 (7.55e−2)

The best result in each row are highlighted in bold

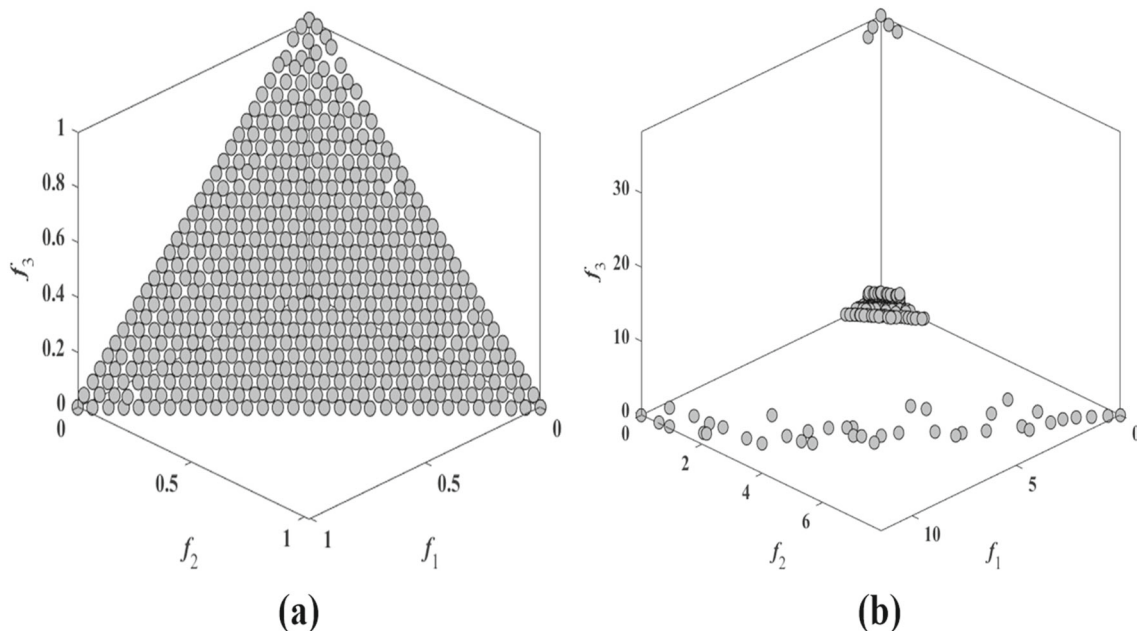


Fig. 14 Nondominated solutions obtained by MOCSOP and its variant on three-objective LSMOP1 with 300 decision variables, where the number of evaluations is set to 4,500,000 and the population size is set to 496. (a) MOCSOP. (b) MOCSOP(NSGA-III)

in Table 5, MOCSOP obtains the best IGD results on 7 out of the 9 test instances. The comparison results demonstrate that our method performs competitively on the large-scale MOPs. It is interesting to find that the potential of MOCSOP for solving large-scale MOPs is related to our proposed environmental selection strategy. The proposed MOCSOP has a similar environmental selection strategy as NSGA-III, except that the probability criterion is added. In MOCSOP, when a reference vector has several associated particles, the particles with the best value of joint probability are preferentially selected to enter the archive. This procedure is beneficial for solving large-scale MOPs, mainly due to the fact that

large-scale MOPs has a large number of decision variables that require greater selection pressure than general MOPs. Figure 4a shows that particles with better joint probability values are closer to the ideal point in the early stages. In MOCSOP, evolutionary search and swarm update strategy are applied to the external archive, therefore the more particles with better joint probability value in the archive can effectively improve the search efficiency, especially in the early stage. In general, the proposed environmental selection scheme indirectly strengthens the selection pressure to some extent and improves the performance of MOCSOP for solving large-scale MOPs. To verify the above hypothesis, we

Fig. 15 Nondominated solutions obtained by MOCSOP and LMOCSO on three-objective DTLZ7 with different numbers of decision variables

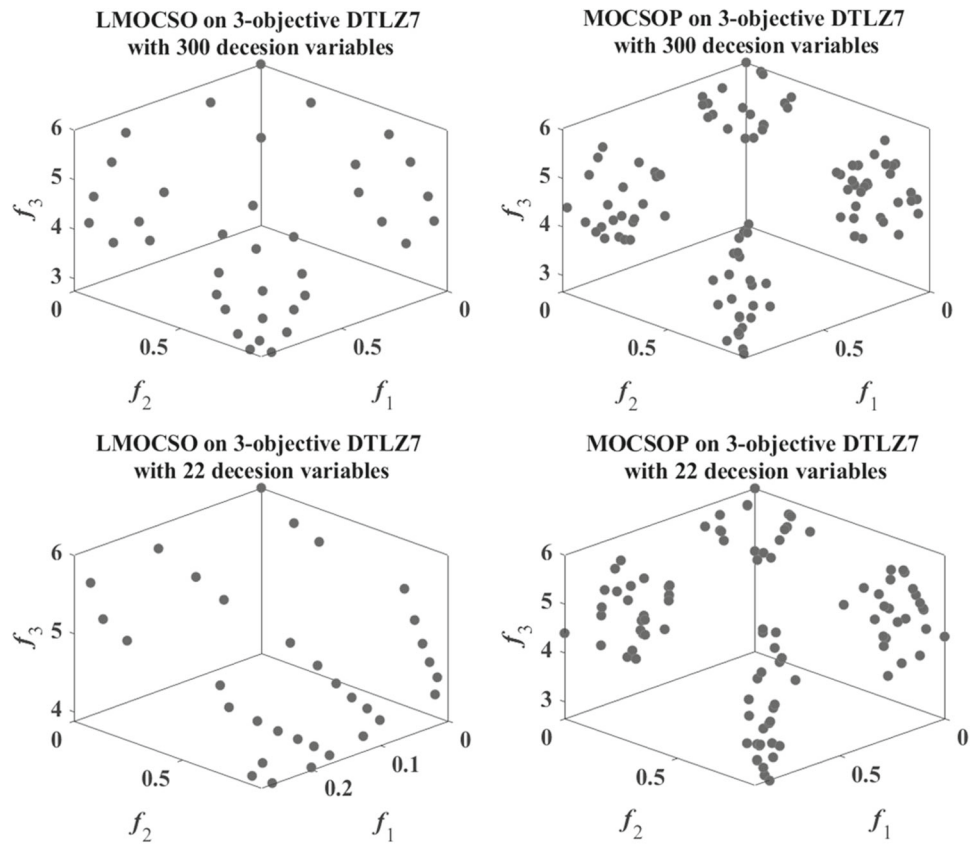


Table 5 The median values of IGD obtained by LMEA, S³-CMA-ES, AND MOCSOP on three-objective LSMOP1- LSMOP9 with 300 decision variables

Problem	Dec	Obj	LMEA	S ³ -CMA-ES	MOCSOP
LSMOP1	300	3	1.6363e-1 (1.66e-1) -	2.6539e-1 (2.28e-2) -	2.7128e-2 (3.43e-2)
LSMOP2	300	3	7.7888e-2 (4.77e-2) -	2.5764e-2 (2.77e-3) +	4.1071e-2 (4.02e-4)
LSMOP3	300	3	1.0608e + 0 (5.59e-1) =	4.3582e + 0 (3.76e-1) -	8.7867e-1 (2.52e-1)
LSMOP4	300	3	1.0093e-1 (6.71e-2) -	1.9768e-1 (7.76e-3) -	9.4951e-2 (6.40e-3)
LSMOP5	300	3	4.7119e + 0 (3.65e + 0) -	9.4596e-1 (3.62e-6) -	8.6403e-2 (4.51e-2)
LSMOP6	300	3	5.6708e + 2 (1.78e + 3) -	1.6622e + 0 (3.25e-1) -	6.1006e-1 (5.58e-2)
LSMOP7	300	3	1.3049e + 0 (3.80e-2) -	9.4735e-1 (7.50e-5) -	6.9049e-1 (1.90e-1)
LSMOP8	300	3	1.2684e-1 (1.43e-2) -	9.4603e-1 (2.76e-4) -	8.6589e-2 (1.06e-3)
LSMOP9	300	3	7.3443e-1 (6.37e-1) +	7.3294e-1 (1.13e-1) +	1.3153e + 0 (6.33e-2)
+/-/≈			1/7/1	2/7/0	

The best result in each row are highlighted in bold

Table 6 The median values of IGD obtained BY LMOCSO, AND MOCSOP on three-objective DTLZ1- DTLZ7 WITH 300 decision variables, where the number of evaluations is set to 4,500,000 and the population size is set to105

Problem	Dec	Obj	LMOCSO	MOCSOP
DTLZ1	300	3	4.5219e + 2 (1.80e + 2) –	5.0219e + 1 (4.48e + 1)
DTLZ2	300	3	5.0327e–2 (5.15e–6) –	5.0301e–2 (2.20e–8)
DTLZ3	300	3	7.5266e + 2 (2.85e + 2) –	7.6585e + 1 (9.82e + 1)
DTLZ4	300	3	2.7841e–1 (2.82e–1) =	4.0400e–1 (2.42e–1)
DTLZ5	300	3	3.0352e–2 (6.02e–4) +	1.2438e–1 (7.46e–2)
DTLZ6	300	3	3.1016e–2 (5.22e–4) –	1.8591e–2 (2.89e–3)
DTLZ 7	300	3	1.5751e–1 (9.07e–2) –	7.5740e–2 (7.32e–3)
+/-/≈			1/5/1	

The best result in each row are highlighted in bold

compared the proposed method with its variant, in which the proposed environmental selection strategy is replaced by the environmental selection of NSGA-III. Figure 14 displays the experimental result, we can find that MOCSOP can obtain a solution set with good convergence and diversity on three-objective LSMOP1 with 300 decision variables, whereas the solution sets obtained by variant is not satisfactory. The comparison results demonstrate that the proposed environmental selection strategy is beneficial for improving the performance of MOCSOP on large-scale MOPs.

In addition, to further examine the capability of the proposed MOCSOP in dealing with large-scale MOPs, we compare it with LMOCSO [20] on DTLZ test problems. Specifically, the LMOCSO is a competitive swarm optimizer (CSO)-based efficient searching method, and it shows good performance on solving large-scale MOPs. Table 6 lists the IGD values of the MOCSOP and LMOCSO evaluated on three-objective DTLZ1–DTLZ7 with 300 decision variables. The comparison results indicate that the proposed MOCSOP performs better than the LMOCSO on most of the test functions. For a visual comparison, Fig. 15 illustrates the nondominated solution set obtained by MOCSOP and LMOCSO on DTLZ7 with different numbers of decision variables. As shown in Fig. 15, the MOCSOP achieves a competitive performance on three-objective DTLZ7 with 300 decision variables compared with LMOCSO. It is noticeable that LMOCSO yields uniform distributions on DTLZ7 with 300 decision variables and it fails to maintain population diversity on DTLZ7 with 22 decision variables. The comparison results indicate that LMOCSO performs poor versatility with respect to problems containing low-dimensional decision variables. By contrast, our method has good versatility regarding different numbers of decision variables. The comparison results demonstrate that the proposed MOCSOP is effective for dealing with large-scale MOPs.

Conclusions

In this paper, we proposed a competitive swarm optimizer with probabilistic criteria to tackle MaOPs, termed MOCSOP. First, we estimated the joint probability of the particles in the population and selected some of the swarm leaders, according to the value of joint probability. Second, we utilized a competition mechanism with winner pool to update position, which can improve the efficiency of searching the true Pareto front. Then, we exploited a diversity mechanism with the mixed probability criterion to ensure the diversity of the swarm. Finally, we designed a swarm update strategy using the particles in the external elite archive to update the current particle swarm, which can effectively improve the convergence of the algorithm. The experimental results on the DTLZ1-7 and WFG1-9 test instances demonstrated that the proposed method presents robust and superior performance compared to other MOPSOs and MaOEAs for tackling MOPs and MaOPs. Furthermore, the comparison results between MOCSOP and other state-of-the-art large-scale MOEAs indicated that the MOCSOP has promising potential in large-scale MOPs.

In the future, we will further investigate the performance of MOCSOP for large-scale many-objective optimization problems (large-scale MaOPs) and apply it to some real-world problems.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable suggestions on improving this paper.

Funding This work was supported in part by the National Key Research and Development Program of China (2020YFC2003800), the National Natural Science Foundation of China (61866026, 61772255, 61866025 and 62066031), the Natural Science Foundation of Jiangxi Province (20181BAB202025), the Advantage Subject Team Project of Jiangxi Province (20165BCB19007), the Aeronautical Science Foundation of China (2018ZC56008), the Outstanding Young Scientist Project of Jiangxi Province (20192BCB23011).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Zhou Q, Wang C, Zhang G (2020) A combined forecasting system based on modified multi-objective optimization and sub-model selection strategy for short-term wind speed. *Appl Soft Comput* 94:1–21
- Barlow GJ, Oh CK, Grant E Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming. In: *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, vol. 2, Singapore, Dec, 2004, pp. 689–694.
- Liu J, Zhang Q, Pei J, Tong H, Feng X, Wu F fSDE: efficient evolutionary optimisation for many-objective aero-engine calibration, *Complex & Intelligent Systems*, 2021.
- Ma A, Wan Y, Zhong Y, Wang J, Zhang L (2021) SceneNet: Remote sensing scene classification deep learning network using multi-objective neural evolution architecture search. *ISPRS J Photogramm Remote Sens* 172:171–188
- Ma W, Wang R, Gu Y, Meng Q, Huang H, Deng S, Wu Y (2021) Multi-objective microservice deployment optimization via a knowledge-driven evolutionary algorithm. *Complex Intell Syst* 7:1153–1171
- Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 17(5):721–736
- Kim JH, Han JH, Kim YH (Feb. 2012) Preference-based solution selection algorithm for evolutionary multiobjective optimization. *IEEE Trans Evol Comput* 16(1):20–34
- Tian Y, Cheng R, Zhang X, Su Y, Jin Y (Apr. 2019) A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans Evol Comput* 23(2):331–345
- Beume N, Naujoks B, Emmerich M (Feb. 2007) SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur J Oper Res* 181(3):1653–1669
- Zitzler E, Künzli S Indicator-based selection in multiobjective search. In: *Proc. 8th Int. Conf. Parallel Problem Solving Nat., Birmingham, U.K., 2004*, pp. 832–842.
- Zhang Q, Li H (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
- Wang R, Zhou Z, Ishibuchi H (2018) Localized Weighted Sum Method for Many-Objective Optimization. *IEEE Trans Evol Comput* 22(1):3–18
- Cai X, Mei Z, Fan Z (2017) A constrained decomposition approach with grids for evolutionary multiobjective optimization. *IEEE Trans Evol Comput* 22(4):564–577
- Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
- Zhao SZ, Suganthan PN, Zhang Q (2012) Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes. *IEEE Trans Evol Comput* 16(3):442–446
- Yuan J, Liu H, Gu F, Zhang Q, He Z (2021) Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions. *IEEE Trans Evol Comput* 25(1):75–86
- Xiang Y, Zhou Y, Yang X, Huang H (2020) A many-objective evolutionary algorithm with pareto-adaptive reference points. *IEEE Trans Evol Comput* 24(1):99–113
- Yang L, Hu X, Li K (2021) A vector angles-based many-objective particle swarm optimization algorithm using archive. *Appl Soft Comput* 106:1–16
- Luo J, Huang X, Yang Y, Wang XLZ, Feng J (2020) A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. *Inform Sci* 514: 166–202
- Tian Y, Zheng X, Zhang X (2020) Efficient large-scale multiobjective optimization based on a competitive swarm optimizer. *IEEE Trans. Cybern.* 50(8):3696–3708
- Li L, Chang L, Gu T, Sheng W, Wang W (2021) On the Norm of Dominant Difference for Many-Objective Particle Swarm Optimization. *IEEE Trans Cybern* 51(4):2055–2067
- Wu B, Hu W, He Z, Jiang M, Yen GG (2018) A Many-Objective Particle Swarm Optimization Based On Virtual Pareto Front. In: *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil pp. 1–8.
- Liu W, Wang Z, Yuan Y, Zeng N, Hone K, Liu X (2021) A novel sigmoid-function-based adaptive weighted particle swarm optimizer. *IEEE Trans Cybern* 51(2):1085–1093
- Liu X, Zhan Z, Gao Y, Zhang J, Kwong S, Zhang J (2019) Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Trans Evol Comput* 23(4):587–602
- Sierra MR, Coello Coello CA Improving PSO-based multiobjective optimization using crowding, mutation and e-dominance. In: *Proc. EMO, LNCS 3410*, 2005, pp. 505–519.
- Nebro AJ et al. (2009) SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In: *Proc. IEEE Symp. Comput. Intell. Multi Criteria Decis. Making, Nashville, TN, USA*, pp 66–73.
- Li L, Wang W, Xu X (2015) Multi-objective particle swarm optimization based on global margin ranking. *Inf Sci* 375(1):30–47
- Wang Y, Yang Y (2009) Particle swarm optimization with preference order ranking for multi-objective optimization. *Inf Sci* 179(12):1944–1959
- Zhu Q, Lin Q, Chen W, Wong K, Coello C, Li J, Che J (2017) An external archive-guided multiobjective particle swarm optimization algorithm. *IEEE Trans Cybern* 47(7):2794–2808
- Li L, Chen S, Gong Z, Lin Q, Ming Z (2019) A novel hybrid multi-objective particle swarm optimization algorithm with an adaptive resource allocation strategy. *IEEE ACCESS* 7:177082–177100
- Cheng R, Jin Y (2015) A competitive swarm optimizer for large scale optimization. *IEEE Trans Cybern* 45(2):191–204
- Zhang X, Zheng X, Cheng R, Qiu J, Jin Y (2018) A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Inf Sci* 427:63–76
- Deng L, Song L, Sun G A competitive particle swarm algorithm based on vector angles for multi-objective optimization. *IEEE ACCESS*, 2021.

34. Xiang Y, Zhou Y, Chen Z, Zhang J (2020) A Many-Objective Particle Swarm Optimizer With Leaders Selected From Historical Solutions by Using Scalar Projections. *IEEE Trans Cybern* 50(5):2209–2222.
35. Han H, Lu W, Zhang L, Qiao J (2018) Adaptive Gradient Multiobjective Particle Swarm Optimization. *IEEE Trans Cybern* 48(11):3067–3079
36. Lin Q, Li J, Du Z, Chen J, Ming Z (2015) A novel multi-objective particle swarm optimization with multiple search strategies. *Eur J Oper Res* 247(3):732–744
37. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
38. Lin Q, Liu S, Zhu Q, Tang C, Song R, Chen J, Coello C, Wong K, Zhang J (2018) Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans Evol Comput* 22(1):32–46
39. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints evolutionary computation. *IEEE Trans Evol Comput* 18(4):577–601
40. Ishibuchi H, Doi K, Nojima Y (2017) On the effect of normalization in MOEA/D for multi-objective and many-objective optimization. *Complex Intell Syst* 3(4):279–294
41. Asafuddoula M, Ray T, Sarker R (2015) A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans Evol Comput* 19(3):445–460
42. Das I, Dennis JE (1998) Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J Optim* 8(3):631–657
43. Sun Y, Yen GG, Yi Z (2019) IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. *IEEE Trans Evol Comput* 23(2):173–187
44. Sun Y, Xue B, Zhang M, Yen GG (2017) A Vector Angle-Based Evolutionary Algorithm for Unconstrained Many-Objective Optimization. *IEEE Trans Evol Comput* 23(5):131–152
45. Xiang Y, Zhou Y, Li M, Chen Z (2017) Biased multiobjective optimization and decomposition algorithm. *IEEE Trans Cybern* 21(1):52–66
46. Jain H, Deb K (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Trans Evol Comput* 18(4):602–622
47. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R (eds) *Evolutionary multiobjective optimization (advanced information and knowledge processing)*. Springer, London, U.K., pp 105–145
48. S. Huband, L. Barone, R. While, and P. Hingston, “A scalable multi-objective test problem toolkit,” in *Proc. 3rd Conf. Evol. Multi Criterion Optim.*, Guanajuato, Mexico, 2005, pp. 280–295.
49. Zhou A, Jin Y, Zhang Q, Sendhoff B, Tsang E Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In: *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 892–899.
50. While L, Hingston P, Barone L, Huband S (2006) A faster algorithm for calculating hypervolume. *IEEE Trans Evol Comput* 10(1):29–38
51. Tian Y, Xiang X, Zhang X, Cheng R, Jin Y (2018) Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems. In: *Proc. IEEE Congr. Evol. Comput.*, Rio de Janeiro, Brazil, pp. 1–6.
52. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biometrics* 1(6):80–83
53. Yuan Y, Xu H, Wang B, Zhang B, Yao X (2016) Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans Evol Comput* 20(2):180–197
54. Zhang X, Tian Y, Cheng R, Jin Y (2018) A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Trans Evol Comput* 22(1):97–112
55. Chen H, Cheng R, Wen J, Li H, Weng J (Jan. 2020) Solving Large-Scale Many-Objective Optimization Problems by Covariance Matrix Adaptation Evolution Strategy with Scalable Small Subpopulations. *Inf Sci* 509:457–469

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.