A Flexible Ranking-Based Competitive Swarm Optimizer for Large-Scale Continuous Multiobjective Optimization

Xiangzhou Gao, Shenmin Song, Hu Zhang, and Zhenkun Wang, *Member, IEEE*

***Abstract*—Due to the curse of dimensionality, the search efficiency of existing operators in large-scale decision space deteriorates dramatically. The competitive swarm optimizer (CSO)-based framework has great potential in tackling large-scale single-objective optimization problems. However, the existing CSOs only focus on the loser to winner learning paradigm and neglect the significance of the winner determination mechanism for large-scale search, which makes the algorithm difficult to escape from local optima. To remedy this issue, a flexible ranking-based CSO has been tailored for handling large-scale multiobjective optimization problems (MOPs). Concretely, a novel winner determination strategy is introduced to broadly identify high-quality individuals in the population to enhance diversity maintenance. In addition, a special competitive mechanism is adopted to guide the search direction, which is capable of efficiently increasing search space utilization. The simulation results validate that the proposed algorithm can significantly enhance the exploration and exploitation ability of the conventional CSO, and outperforms several state-of-the-art large-scale multiobjective optimization algorithms on both large-scale benchmark MOPs and application examples.**

***Index Terms*—Competitive swarm optimizer (CSO), large-scale multiobjective optimization, search paradigm.**

I. Introduction

Many complex optimization problems with multiple conflicting objectives in real-world applications are expected to be simultaneously solved, which are collectively known as multiobjective optimization problems (MOPs) [1], [2], [3], [4]. The decision maker often requires as many trade-off solutions as possible to integrate any preference, especially when there is not enough prior knowledge. Although general MOPs have been extensively studied in the past few decades and have achieved dazzling accomplishments [5], [6], its scalability in terms of the number of decision variables remains inadequately explored [7]. In fact, many scientific and engineering fields, such as artificial intelligence [8], software engineering [9], and economics [10], involve MOPs that are not only generally NP-hard landscapes but also contain a large number of decision variables. Without loss of generality, in this article, the following large-scale continuous MOP, termed LSMOP, is mathematically defined as:

$$\begin{aligned} \text{minimize} \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T \\ \text{subject to} \quad & \mathbf{x} \in \prod_{i=1}^{D} [a_i, b_i] \\ & h_i(\mathbf{x}) \leq 0, i = 1, \ldots, n \end{aligned} \tag{1}$$

where $-\infty < a_i < b_i < +\infty$ for all $i \in [1, D]$. M is the number of objective functions in the LSMOP being handled. $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ denotes the decision vector containing D real variables. $\prod_{i=1}^{D} [a_i, b_i] \subseteq \mathbb{R}^D$ defines the decision space and $h_i(\mathbf{x})$ represents n inequality constraints. $F : \prod_{i=1}^{D} [a_i, b_i] \to \mathbb{R}^M$ denotes the continuous mapping from decision space to objective space with $M \geq 2$ and $D \geq 100$ [11].

Generally speaking, the same principle as tackling general MOPs ($M \geq 2$ and $D < 100$), properly tackling the LSMOP needs to find the Pareto front (PF) in the objective space which is mapped by a set of uniformly distributed Pareto-optimal solutions (PS) in the decision space or solution space [12], [13]. However, the exclusive challenge caused by the linear increase in the number of decision variables makes the treatment of such LSMOPs nontrivial [14], [15]. More specifically, as the curse of dimensionality explicitly gives rise to the size of the search space to increase exponentially, most existing sophisticated multiobjective evolutionary algorithms (MOEAs) have been empirically verified that they are difficult to represent the best possible compromises between global search and local search [16], which is critical to achieving good results.

Intuitively, the inherent multiobjective evolutionary process can be regarded as a diversity loss process, which means that it is impractical to essentially improve the overall performance of large-scale MOEAs (LSMOEAs) by only ameliorating the environment selection strategy. Moreover, a high-level exploiting of the search space is capable of avoiding the

Manuscript received 17 July 2023; revised 22 October 2023 and 18 December 2023; accepted 12 January 2024. Date of publication 17 January 2024; date of current version 31 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 61690210 and Grant 6191101340, and in part by the Defense Industrial Technology Development Program. This article was approved by Associate Editor S. Mostaghim. *(Corresponding author: Xiangzhou Gao.)*

Xiangzhou Gao and Shenmin Song are with the Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: xiangzhougao@126.com; songshenmin@hit.edu.cn).

Hu Zhang is with the Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China (e-mail: jxzhanghu@126.com).

Zhenkun Wang is with the School of System Design and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: wangzhenkun90@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TEVC.2024.3355221, provided by the authors.

Digital Object Identifier 10.1109/TEVC.2024.3355221

algorithm trapping into prematurely converging or stagnating in a suboptimal region [17]. As a result, in order to solve LSMOPs, it may be a natural way to transplant large-scale single-objective (LSO) optimization techniques into MOEA paradigm, such as cooperative coevolution [18], random embedding [19], dynamic grouping [20], and dimensionality reduction [21]. Although these optimizers might struggle to achieve high efficiency and effectiveness when used directly for a wide range of LSMOPs [7], [12], researchers have recently been inspired to develop a variety of MOEAs tailored for handling LSMOPs, which can be roughly divided into the following three recognized categories on the basis of different technologies for handling decision variables.

The first mainstream category is decision variable grouping-based MOEAs. As the name implies, these algorithms follow the divide-and-conquer idea, that is, randomly or heuristically divide the decision variables into groups and then optimize alternately. A classical attempt [18] is to embed the random grouping and cooperative co-evolutionary techniques into the generalized differential evolution algorithm framework [22], in which the decision variables are randomly allocated into several subpopulations with equal size to increase the chance of optimizing some interacting variables together. Since random grouping neglects the interactions between decision variables, the differential grouping technique [23] is additionally employed, which is effective in preventing the population from trapping into local optima. In order to simultaneously maintain the convergence and diversity performance during the evolutionary process, decision variables in [11] are divided into position variables, distance variables, and mixed variables by analyzing their control properties. Furthermore, decision variable clustering methods [24], [25] are used for more robust search performance. In [26], a bilevel variable grouping-based framework has been proposed, which divides the LSMOP into combination variable cells and decomposition variable units, and then uses multipopulation strategy to optimize the problems, respectively. An importance-based variable grouping method has been proposed in [27], which accelerates convergence by constraining the search space during the process of offspring generation. In [28], a decision variable analysis-based method has been proposed which dynamically estimates the variable interaction relationship during the evolution to effectively lead the population toward the optima.

The second category is decision space reduction-based MOEAs. One of this category is to optimize the weights of the solutions explicitly by weighted optimization framework (WOF) [29], [30]. Note that the weight vector dimension is much shorter than the decision variable dimension, and hence this framework can significantly reduce the decision space. In [31], a combination of WOF, dynamic random grouping, and a multiobjective particle swarm optimization algorithm with multiple search strategies has been proposed to facilitate the diversity. The other is inspired by dimensionality reduction techniques (DRTs), which are widely used in machine learning, such as random embedding [21], principal component analysis [32], and unsupervised neural networks [33]. Notably, most of DRTs cannot be directly applied to tackling LSMOPs as the vectors shortened by them are not recoverable. In [34], a discriminative reconstruction network model trained by the obtained nondominated solutions has been proposed to transfer the solutions of source LSMOPs to the target LSMOP to assist its optimization and learn a dimensional-reduced Pareto-optimal subspace of the target LSMOP. Recently, a self-guided problem transformation-based optimizer has been proposed in [35], which transfers the original search space to a low-dimensional weighted space for search to improve convergence performance.

Incidentally, the first two types can be roughly summarized as the *divide-and-conquer mechanism*, which may encounter difficulties in effectively finding the global optimums. On the one hand, as two interacting variables are optimized separately, some essential information related to the global optimums will be missed, which may finally lead the population to falling into a local optimum [7]. On the other hand, the weights for each group of decision variables in WOF is always the same, which is not conducive to the global optimums. Although the aim of variable analysis [11], [24], [25] is to detect the variable interactions and maintain diversity, accuracy comes at the cost of high complexity and expensive computation [12].

The third category customizes novel reproduction strategies, including probability models and learning-based reproduction operators. Different from the types aforementioned, this category regards the decision variables as a whole for evolving with simple procedures and balanced performance. Fundamentally inspired by the regularity property on the basis of Karush–Kuhn–Tucker condition [36], IM-MOEA [37] and its variant IM-MOEA/D [38] construct Gaussian process-based inverse models to map the candidate solutions from the objective space to the decision space, and create offsprings by sampling the objective space. To accelerate the convergence speed, an evolutionary search strategy based on multilayer perceptron has been proposed in [39] to guide the search direction of the offspring toward the promising region.

Furthermore, LMOCSO [40], an innovation of the competitive swarm optimizer (CSO) [41], proposed an potential update strategy for solving LSMOPs. In this strategy, loser learns from the winner in terms of particle update velocity and acceleration, respectively.

Although various LSMOEAs have been proposed, how to design and implement the search strategy relating to effectiveness and efficiency is still a challenging task, which is mainly due to threefold reasons.

1) Since the exponential increase of the search space, the conventional LSMOEAs lack an tailored operator with a simple and efficient search paradigm for the LSMOP while all the decision variables are optimized together, resulting in a low search space utilization;
2) In the evolutionary process, the potential of high-quality individuals is not fully exploited although they may also provide diverse guided directions. For example, in CSO, the rough pair-wise learning mechanism cannot select enough diverse individuals as winners which gives rise to the population into a local optimum, although they may have so-called higher fitness values;
3) The improvement of the overall quality of the population requires an effective diversity maintenance mechanism

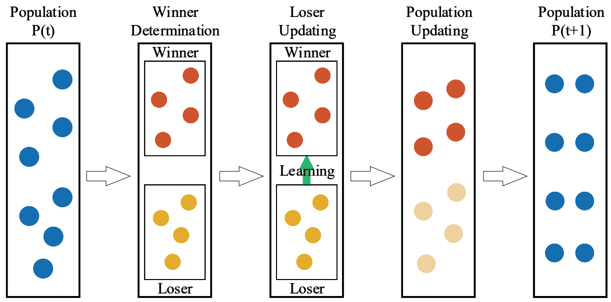

Fig. 1. Principal diagram of the CSO. The main structure diagram of the CSO is composed of three main components, which are the WDS used to divide winners and losers, the loser update strategy that the loser learns from the winner to update its position and velocity, and the population update strategy implemented by the environment selection operator.

to jump out of the local optimal region. However, how to generate plenty of potential candidate solutions that are uniformly scattered around the true PF as the search goes on is a tall order.

To address the above issues, we propose an LSMOEA enlightened by CSO for handling LSMOPs. The main contributions of this article can be summarized as follows.

1) Unlike the CSO and its variants which only focus on the loser updating mechanism to improve the performance, a flexible ranking strategy is proposed to determine the ranking of particles. This ranking can flexibly identify convergence-related nondominated particles and diversity-related high-quality particles in a single run, improving the overall quality of winner particles.
2) A flexible ranking-based CSO called FRCSO is developed. The proposed FRCSO implements a new competition mechanism, in which the loser performs different update strategies for different search stages to efficiently utilize the leading direction of the winner. It can accelerate the current population escape from the local optimum region and avoid premature convergence.
3) Compared with four state-of-the-art LSMOEAs, the experimental results show that the promising performance and scalability to the FRCSO. The proposed strategy can effectively enhance the diversity maintenance and efficiency of existing CSOs and achieve a good approximation along the PF.

The remainder of this article is organized as follows. First, Section II introduces some preliminaries. Section III presents the proposed algorithm. Section IV presents a controlled systematic comparison study of the proposed algorithm. Section V discusses the experimental results, concludes this article and look ahead to the future research avenues.

II. Related Works and Motivations

A. CSO and Its Variants

Fundamentally inspired by PSO [42], CSO was originally introduced for tackling LSO problems in 2015 [41], which replaced the global optimal position and local optimal position in PSO with a pairwise competition mechanism, and no longer recorded the information about the optimal position. Fig. 1 shows the diagram of the CSO, which consists of five main components. In CSO, the winner can be preserved, while the loser learns from the winner and satisfies the following update policy:

$$\begin{aligned}\vec{v}_l(t+1) &= r_0\vec{v}_l(t) + r_1(\vec{x}_w(t) - \vec{x}_l(t)) \\ &\quad + \varphi r_2\left(\bar{\vec{x}}(t) - \vec{x}_l(t)\right) \\ \vec{x}_l(t+1) &= \vec{x}_l(t) + \vec{v}_l(t+1)\end{aligned} \tag{2}$$

where $\vec{v}_l(t)$ is the velocity vector of the loser to be updated, $\vec{x}_w(t)$ and $\vec{x}_l(t)$ are the positions of the winner and the loser in the tth iteration, respectively, $\bar{\vec{x}}(t)$ is the mean position of the relevant particles, φ is the control parameter of $\bar{\vec{x}}(t)$, and r_0, r_1, and r_2 are the uniformly randomly distributed values in [0, 1].

Since the strong potential shown by CSO in solving problems with large-scale variables, researchers have been motivated to further investigate its extensibility and effectiveness. Generally, extant variants of CSO can be roughly divided into two categories. The first category is to directly integrate the CSO into different frameworks to handle specific problems, such as surrogate model construction [43], feature selection [44], economic dispatch problems [45], and multimodal MOPs [46]. The second category aims to enhance the particle updating strategy of CSO. In order to achieve higher convergence, the tri-population breakup mechanism [47] has been introduced into CSO so as to depict a novel tri-competition mechanism [48], [49], [50], in which two-thirds of the population are treated as losers and the rest (as winners) are directly extended to the next generation. In [51], a straightforward particle self-exploratory strategy has been proposed to enhance the exploration and exploitation of losers in order to guide search direction of the population to the promising regions. In [40], instead of directly utilizing the loser to learn from the winner, the previous velocity of each particle is first used to preupdate the position once, and then its position can be updated twice by learning from the winner. Experimental results have illustrated that this strategy is very efficient in maintaining a high degree of convergence. Subsequently, an enhanced CSO with a new loser particle update strategy [52] has been proposed by embedding covariance matrix adaptation evolutionary strategy into CSO, which considers the structural characteristics of the problem as well as the correlation between variables, integrates the evolution direction information into the learned Gaussian model to address premature convergence. To provide diverse guidance for LSMOPs, a comprehensive competitive learning strategy has been proposed for CSOs by using three competition mechanisms to enhance the search capability of loser particles and provide diverse search directions [53]. Coincidentally, an enhanced competitive swarm optimization algorithm assisted by a strongly convex sparse operator has been proposed in [50] to better balance exploration and exploitation by introducing a three-competition mechanism into the competitive swarm optimization algorithm. In this algorithm, the strongly convex sparse operator has been embedded into the position update of the particles to produce sparse solutions. In MPSO-EBCD [54], the convergence and

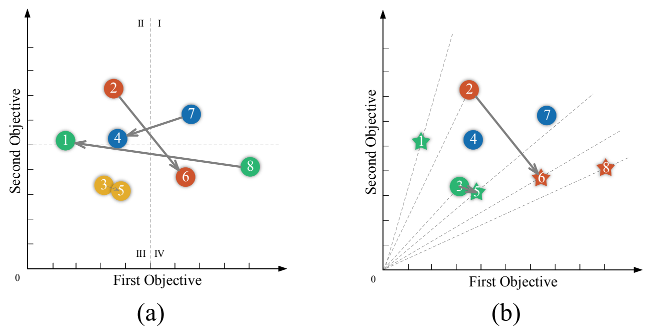

Fig. 2. Illustration of the WDS in (a) CSO and (b) CMOPSO, where the main idea of (a) is to randomly select two individuals and select the one with high fitness as the winner, while the main idea of (b) is to randomly select the other two individuals for each individual, and the one with smaller angle to the selected individual is regarded as the winner.

diversity preservation operations have been divided into two independent components by the proposed velocity update strategy, which has been demonstrated to be competitive in tackling LSMOPs.

B. Motivation of This Article

As can be seen in the discussions above, the effectiveness of CSO, could be considered as a winner oriented optimizer, has been recognized in the field of evolutionary computation, and that two crucial components, namely, winner determination and loser updating, are mainly responsible for its good performance. It should be noted that just because the CSO is formally suitable in the learning formalism, it does not necessarily mean it can perform well for the LSMOP. Suitability only means that the CSO is capable of generating convergent solutions, but gives no guarantees that this search strategy will be either efficient. However, the appropriate winner determination strategy (WDS) to enhance the search ability of CSOs, especially for solving LSMOPs, has hardly been reported so far. Tracing it to its cause, one of the challenges to maximize the search efficiency of CSO is how to make full use of the guiding role of the winner. However, most of the existing winner determination principles randomly select two particles for comparison, which are crude and univocal. In [46], the winner is not the result of random comparisons in the population, but in a neighbor determined by ring topology and special crowding distance, which means that extending this strategy for tackling LSMOPs is not recommended since failure of neighborhood topology as the variables larger than 200 [55]. In addition, an angle-based WDS has been reported in CMOPSO [56]. For each particle to be updated in the population, two randomly selected competitors from limited elite particle set are selected and the angle is calculated, respectively, in which the competitor with the smaller angle is taken as the winner. To clearly discuss the WDS, we consider an LSMOP with two objectives and eight particles as shown in Fig. 2.

Fig. 2(a) illustrates the basic principle used in the WDS, which is similar to most of the existing CSO and its variants. Specifically, the eight particles are randomly divided into four pairs, each of the same color. The arrow points to the winner, where the competition results are from LMOCSO [40]. Accordingly, it can be observed that although particles 1, 3, 5, 6, and 8 are nondominated with reasonable diversity in the current population, particle 8 still needs to give up its evolution potential in region IV as well as drive the search toward the region II, and particle 7 learns from particle 4 in region II. Such the WDS will hinder the diversity search of the population when there are fewer high-quality particles in a certain region. Moreover, in the constructed competition relation, only particle 4 and particle 7 can be compared based on the objective value, while those in other pairs are incomparable, which means that a particle may be updated due to its poor performance in several dimensions even if it has discovered the potential local optimum regions. Coincidentally, in Fig. 2(b), the WDS is derived from CMOPSO, where the stars represent the particles in the set Γ, and the set Γ consists of γ particles from the population according to the front index and the crowding distance of each particle, in which the γ is the number of elite particles to be selected. Clearly, particles 2, 3, 4, and 7 represent particles to be updated, and particles 1, 5, 6, and 8 represent elite particles in Γ. For particle 3, it is assigned the same color as particle 1 and particle 5, which means that one of the elite particles 1 and 5 will be selected as the learning object of particle 3. Since the angle between particle 3 and particle 5 is smaller, particle 3 will learn from particle 5. Similarly, when updating particle 2, elite particles 6 and 8 are selected as its potential learning objects, and these three particles are assigned the same hue. In the competition between elite particles 6 and 8, particle 2 needs to be updated on the basis of particle 6 due to the smaller angle between them. When the pairwise competition is performed among the particles 1 and 5, particles 3, 4, 6, 7, and 8 will learn from particle 5, which may obviously accelerate the population to trap into premature convergence or stagnation, especially when handling LSMOPs, which makes it challenging for the population to achieve a good spread along the PF. Last but not least, as a metaheuristic swarm optimizer inspired by the animal swarm behaviors, it may be biologically more plausible for the individual in finding the leader from a neighbor swarm rather than a distant swarm and learn it. Since the competitive relationship is not fixed during evolution, learning over a large area may lead to a convergence loss, especially in the early stages of the search.

From the above example, we can see that whether to select truly high-quality individuals in the population, which have more potential in terms of convergence or diversity, as winners, may not only raise the efficiency of the loser updating strategy in CSO, but also enable a better tradeoff between convergence and diversity in evolutionary search. Inspired by the above phenomenon and observation, FRCSO implements a novel WDS, which first discovers the nondominated individuals from the population, and then performs a secondary selection criterion on the remaining individuals to flexibly retain the potential individuals. Different with the WDS used in CMOPSO and LMOCSO, which may select the better of the two particles from a very small search region as the winner, the proposed WDS can avoid the competition and learning between two particles that are too close and exploit the potential of those particles with worse-convergence but well-diversity. Although

Algorithm 1 FRCSO Framework

Input: N (population size), $G_{\max}$ (maximum number of iterations), M (number of objectives), D (number of variables)
Output: P (final population)
1: $P \leftarrow \mathit{RandomInitialization}(N)$
2: $G \leftarrow 1$
3: **while** $G < G_{\max}$ **do**
4: $\quad [\mathit{WinSet}_1, \mathit{LosSet}] \leftarrow \mathit{FlexibleRanking}(P)$
5: $\quad \mathit{WinSet}_2 \leftarrow \mathit{LoserUpdating}(\mathit{LosSet})$
6: $\quad \mathit{Offspring} \leftarrow \mathit{Mutation}(\mathit{WinSet}_1 \cup \mathit{WinSet}_2)$
7: $\quad P \leftarrow \mathit{EnvironmentalSelection}(P \cup \mathit{Offspring})$
8: $\quad G \leftarrow G{+}{+}$
9: **end while**
10: **return** P

Algorithm 2 Flexible Ranking

Input: P (current population), N (population size)
Output: WinSet_1 (first winner set), LosSet (loser set)
1: $(F_1, \ldots, F_l) \leftarrow \mathit{NondominatedSort}(P)$
2: **if** $(|\, F_1 \,| < N)$ **then**
3: $\quad \mathit{ArcSet} \leftarrow P \backslash F_1$
4: $\quad \mathit{DisDegree} \leftarrow \mathit{CalculateCrowdingDistance}(\mathit{ArcSet})$
5: $\quad \mathit{LocalDegree} \leftarrow \mathit{CalculateLocalDegree}(\mathit{ArcSet})$
6: $\quad \mathit{ArcSort} \leftarrow \mathit{PacketSequencing}(\mathit{DisDegree}, \mathit{LocalDegree})$
7: $\quad \mathit{WinSet}_1 \leftarrow \mathit{TruncateSelection}(F_1, \mathit{ArcSort})$
8: **else**
9: $\quad \mathit{WinSet}_1 \leftarrow \mathit{RandomlySelection}(P/2)$
10: **end if**
11: $\mathit{LosSet} \leftarrow P \backslash \mathit{WinSet}_1$
12: **return** WinSet_1, LosSet

these potential individuals may not the highest-ranking, when they are arranged as winners, they can not only guide their followers to avoid the population from trapping into premature convergence but also effectively improve the utilization of the search space, which is crucial for addressing the LSMOPs.

III. Proposed FRCSO

A. General Framework of the Proposed Algorithm

The general framework of FRCSO is presented in Algorithm 1, which consists of the following parts: population initialization, winner determination, loser updating, and population updating, similar to most CSOs. Notably, after randomly generating an initial parent population of size N with M objectives and D variables, the proposed flexible ranking strategy occurs over the neighborhood corresponding to each individual to filter out high-quality individuals from the current population that are regarded as winners, as shown in line 4. Next, to take full advantage of the potential of the winners in the convergence search direction and diversity search direction, different learning strategies are implemented for loser updating, and then the mutation method is used to update all winners, as described in lines 5 and 6. Finally, the environment selection was carried out to update the population, so that N individuals could be selected into the next generation in line 7. Specifically, we select SPEA2 [57] for environmental selection since it not only has excellent performance but also can select N individuals into the next generation in each iteration. This procedure repeats until the termination criterion is met. Our proposed offspring generation method can be embedded in many other MOEAs.

B. Flexible Ranking Strategy

Most existing CSOs focus more on how the winner guides the loser to update its position vector $\vec{x}_l$ and velocity vector $\vec{v}_l$, which naturally makes the identification of the winner more important. Intuitively, since the loser update direction and convergence efficiency are determined by the winner particle, when dealing with LSMOPs, once the corresponding local optimal solutions around each peak are not fully explored, the deterioration of winner diversity will make it difficult for the optimizer to jump out of the premature region.

To address this deficiency, we propose a novel WDS termed flexible ranking strategy as presented in Algorithm 2, which mainly includes two parts. The first part uses the triple screening mechanism to jointly determine the ranking of particles. The second part is to filter out the winner using the truncated selection strategy. Specifically, nondominated sorting as reported in [5] is first performed on the current population (line 1) to screen out convergence-related nondominated particles. Next, the particles in F_1 (the subset consisting of the particles in the first front) are removed from the current population and the remaining particles are placed in *ArcSet* (line 3). Then, the crowding distance between particles in *ArcSet* is stored in *DisDegree* (line 4) and the local degree of each particle is reserved in *LocalDegree* (line 5) to extract diversity-related high-quality particles, the details of which are introduced in Section III-B1. Next, crowding distance and local degree are utilized to sort the population (line 6). After that, the winners are determined by truncate selection and stored in WinSet_1 (line 7). Finally, the remaining particles in the current population are treated as losers (line 11). With the change of the number of nondominated individuals, the proposed WDS can flexibly rank the individuals in the population to ensure that the winner is composed of a sufficient number of convergence-related nondominated particles and diversity-related high-quality particles.

1) Local Degree Calculation: The key to tackling the LSMOPs lies in judiciously allocating limited computing power to regions with potential high-quality solutions. On the one hand, during evolutionary process, it may not be imperative to learn between closely related particles, particularly in the initial search stage where capturing potential diversity-related regions takes precedence. On the other hand, a swarm-based biological characterization indicates that the learning process of an animal always occurs naturally in its specific environment and is bottom-up [58]. Inspired by these, it may be rational to guide the current particle to learn its neighbor superior particles. In our design, when collecting the local degree of the particle at generation t, the neighborhood radius of the particle should be obtained first, which is defined by a hyper cube of size $R_1, R_2, \ldots, R_m$ and can be calculated

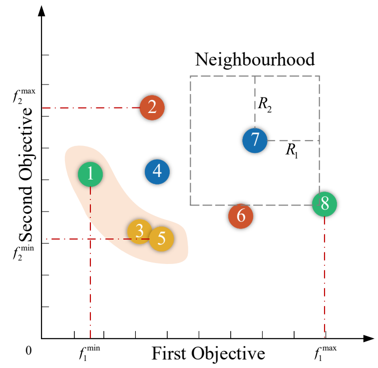

Fig. 3. Illustration of the local degree in FRCSO for a bi-objective minimization problem, in which R_1 and R_2 are the neighborhood radius of particle 7 at the ith generation, particles 1, 3, and 5 are nondominated.

Algorithm 3 Truncate Selection

Input: F_1, *ArcSort*
Output: *WinSet*$_1$
1: **if** $(|F_1| < N/2)$ **then**
2: *WinSet*$_1$ $\leftarrow$ Select first $(N/2 - |F_1|)$ particles from *ArcSort* and particles in F_1
3: **else if** $(|F_1| < 3N/4)$ **then**
4: *ArcF*$_1$ $\leftarrow$ Randomly select $(N/4)$ particles from F_1
5: *WinSet*$_1$ $\leftarrow$ Select first $(N/4)$ particles from *ArcSort* and particles in *ArcF*$_1$
6: **else**
7: *WinSet*$_1$ $\leftarrow$ Randomly select $(N/2)$ particles from F_1
8: **end if**
9: **return** *WinSet*$_1$

as follows:

$$R_m(t) = \delta \cdot \left(f_m^{\max}(t) - f_m^{\min}(t)\right) \tag{3}$$

where $f_m^{\max}(t)$ and $f_m^{\min}(t)$ represent the maximal and the minimal values of the mth objective at the tth generation, and δ denotes the adaptive neighborhood control parameter used to control the neighbor size, which is defined as

$$\delta = \alpha^{(|F_1(t)|/|N|)-1} \tag{4}$$

where α is suggested as 3 in this article. Second, for the particle $p_i \in ArcSet$, its local degree $V(i)$ in the neighborhood C_i can be calculated by the following:

$$V(i) = \frac{\sum_{j=1}^{|C_i|} B_{i,j}}{|C_i|} \tag{5}$$

where

$$B_{i,j} = \begin{cases} 1, \text{ if } p_i \text{ is dominated by } p_j \\ 0, \text{ else} \end{cases}.$$

Specifically, for each individual in the *ArcSet*, its neighborhood is first determined by (3), where the neighborhood control parameters δ can be calculated by experience formula (4), in which the base number is set as 3 in this article. The ablation study related to this choice of number will be described in Section IV-B. When the neighborhood is determined, the local degree of each individuals in the neighborhood can be obtained by (5). For each neighborhood, the individual with the largest V is considered as a diversity-related particle. Once all the diversity-related particles are determined, then they are sorted according to the magnitude of V and placed in the *ArcSort* as well as the *DisDegree*.

An illustrative example demonstrating the calculation method of our proposed local degree is presented in Fig. 3. For particle 7, its neighbors include particles 6 and 8. Therefore, $C_i = 2$. It can be seen that particle 7 is dominated by particle 6, which means that $B_{7,j} = [1, 0]$. As a result, the local degree of particle 7 is $V(7) = 0.5$. In the same way, the local degree of particles 2, 4, 6, and 8 are 0.5, 0.5, 1, and 0, respectively. Therefore, particle 8 is the most diversity-related within these five particles. It can be observed by comparing Figs. 2 and 3, after the flexible ranking strategy is implemented in a single run, particles 1, 3, 5, and 8 will be regarded as winners, where particles 1, 3, and 5 are considered as convergence-related particles, and particle 8 is considered as diversity-related particle. However, it is difficult for the mechanism showed in Fig. 2 to find the most potential particles in the current population in a single run. Although particle 8 is not nondominated, it can be retained due to its largest local degree value. It can be seen that the selection of particle 8 as the winner is beneficial to guide its followers to diversify the exploitation of the search space.

2) Truncate Selection: The purpose of truncate selection is to screen superior particles from the sorted particles, the details of which are presented in Algorithm 3. In the first case, if $|F_1| < N/2$ ($|F_1|$ represents the number of particles contained in the F_1), the particles from *ArcSort* are successively supplemented to the *WinSet*$_1$ in order of rank until $|WinSet_1| = N/2$ (line 2). In the second case, if $|F_1| < 3N/4$, $N/4$ particles are randomly selected from F_1 instead of all as winners, and the other $N/4$ particles are from the top quartile of *ArcSort* (line 4 and 5). In the third case, when $3N/4 \le |F_1| < N$, it is assumed that the current population exhibits a high-level of quality. In this case, $N/2$ particles are randomly selected from F_1 as the winners (line 7). It is worth noting that, as Algorithm 2 specifies, not only nondominated particles but also diversity-well particles can be chosen as winners. As losers continue to learn from the winners, these diverse particles can generate numerous high-quality offsprings, which are then adopted into the F_1. Therefore, this virtuous cycle can effectively guide the search direction of other individuals to achieve a good spread along the PF.

C. Loser Updating Strategy

The loser performs a suitable update strategy which is capable of reinforcing their learning of the identified convergence-/diversity-related particles. Notably, exploratory learning is particularly advantageous for rapidly capturing potential regions where nondominated solutions may exist as well as exploitative learning can facilitate diversity maintenance. While the overall quality of winners improves as the search goes on, the comprehensive ability may be restricted as the loser learn from only one winner. To take more effective

Algorithm 4 Loser Updating

Input: F_1, $LosSet$
Output: $WinSet_2$

$WinSet_2 \leftarrow \emptyset$
if ($|F_1| < N/2$) **then**
  for each particle x in $LosSet$ **do**
    $x \leftarrow$ update x by learning from x_l by (8)
  end for
else
  for each particle x in $LosSet$ **do**
    $x \leftarrow$ update x by learning from x_l by (10)
  end for
  $WinSet_2 \leftarrow WinSet_2 \cup \{x\}$
end if

advantage of the leading direction of the winner, a new competition mechanism within FRCSO is designed to adapt to the proposed WDS, in which the loser performs different update strategies for different search stages, the details of which are presented in Algorithm 4. Consequently, as $|F_1| < N/2$, the loser updating strategy proposed in LMOCSO [40] is adopted, which has been verified that the particle can move toward the leader with a fast convergence speed. Whereafter, as $|F_1| \geq N/2$, the loser is updated according to (9), which is modified from (2), aiming to enhance the diversity maintenance.

Concretely, in order to obtain a more explicit difference between the learning patterns in FRCSO and canonical CSO, (2) can be transformed as follows:

$$\begin{cases} \vec{v}_l(t+1) = r_0\vec{v}_l(t) + \eta_1(\rho_1(t) - \vec{x}_l(t)) \\ \vec{x}_l(t+1) = \vec{x}_l(t) + \vec{v}_l(t+1) \end{cases} \tag{6}$$

where

$$\begin{cases} \eta_1 = r_1 + \varphi r_2 \\ \rho_1(t) = r_1(\vec{x}_w(t)/\eta_1) + \varphi r_2\left(\vec{\bar{x}}(t)/\eta_1\right) \end{cases}. \tag{7}$$

Similarly, the loser updating strategy in LMOCSO can be written as

$$\begin{cases} \vec{v}_l(t+1) = r_3\vec{v}_l(t) + \eta_2(\rho_2(t) - \vec{x}_l(t)) \\ \vec{x}_l(t+1) = \vec{x}_l(t) + \vec{v}_l(t+1) + r_3(\vec{v}_l(t+1) - \vec{v}_l(t)) \end{cases} \tag{8}$$

where

$$\begin{cases} \eta_2 = r_4 \\ \rho_2(t) = \vec{x}_w(t) \end{cases}. \tag{9}$$

Obviously, compared to the canonical CSO, it can be seen from (6) and (8) that $\vec{\bar{x}}(t)$ is not utilized in the speed update rule of LMOCSO. On the one hand, it is argued that the loser learning toward the mean position value $\vec{\bar{x}}(t)$ of the current population degrade the search space utilization as tackling LSMOPs. On the other hand, it can lift the burden of setting parameters related to $\vec{\bar{x}}(t)$. To improve the diversity and search efficiency of particles, losers can conditionally learn from different winners in the same competition, in which the loser updating paradigm can be expressed as follows:

$$\begin{cases} \vec{v}_l(t+1) = r_5\vec{v}_l(t) + \eta_3(\rho_3(t) - \vec{x}_l(t)) \\ \vec{x}_l(t+1) = \vec{x}_l(t) + \vec{v}_l(t+1) \end{cases} \tag{10}$$

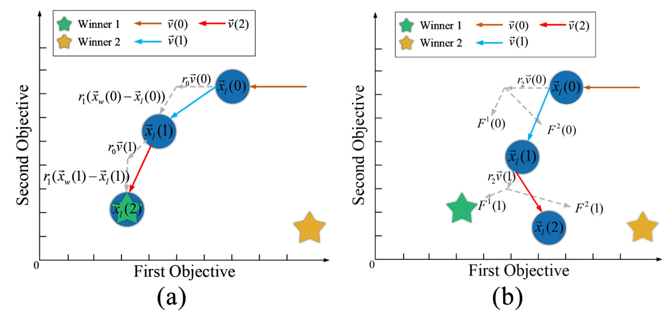

Fig. 4. Trajectory of a particle after learning twice from the fixed leader by the proposed loser updating strategy given in (8) and (10), respectively.

where

$$\begin{cases} \eta_3 = r_6 + r_7 \\ \rho_3(t) = r_6\left(\vec{x}_w^{\,i}(t)/\eta_3\right) + r_7\left(\vec{x}_w^{\,j}(t)/\eta_3\right), i \neq j \end{cases}. \tag{11}$$

Taking a closer look at Fig. 4, it plots the trajectory of one particle after learning from the winner by the proposed loser updating strategy for two times. When $|F_1| < N/2$, the loser updating strategy is the same as that used in LMOCSO, as shown in Fig. 4(a), which has been proven to accelerate the efficiency of the loser learning from the winner. This learning strategy is performed in the early stages of the search, and it can be seen that the loser learns from only one winner to accelerate exploitation. When $|F_1| \geq N/2$, the proposed loser updating strategy is executed as shown in Fig. 4(b), where $F^1(t)$ and $F^2(t)$ represent the velocity components learned by the loser from the Winner 1 and the Winner 2 at the tth generation, respectively. For clarity, the velocity update equation can be obtained by taking (11) into (10)

$$\begin{aligned} \vec{v}_l(t+1) &= r_5\vec{v}_l(t) + F^i(t) + F^j(t) \\ F^i(t) &= r_6 x_w^i(t) - \eta_3\vec{x}_l(t)/2 \\ F^j(t) &= r_7 x_w^j(t) - \eta_3\vec{x}_l(t)/2. \end{aligned} \tag{12}$$

At this stage, the loser can learn from two winners to enhance diversity maintenance, which is beneficial for the current population to jump out of the local optimal region. Distinctly, the proposed particle updating strategy can ensure that the algorithm explores new regions while also capitalizing on its previous knowledge to optimize the search process.

IV. Empirical Studies

In this section, several experiments are conducted to investigate the performance of the proposed algorithm in solving LSMOPs. First, two sets of ablation experiments are conducted to verify the effectiveness of the WDS and loser updating strategy in FRCSO. Next, FRCSO is compared with several state-of-the-art LMOEAs on benchmark LSMOPs. Then, the scalability of FRCSO is tested on LSMOPs with decision variables ranging from 100 to 5000. Finally, we apply these algorithms to tackle the time-varying ratio error estimation (TREE) problem of voltage transformers.

Six state-of-the-art LSMOEAs are involved for comparative experiments, namely, DGEA [59], IM-MOEA/D [38], LMOEA-DS [15], AGE-MOEA-II [60], WOF [29], and LMOCSO [40], which are introduced as follows.

1) DGEA utilizes the preselection strategy to generate promising offspring in the decision space, which has been verified to have good convergence and diversity when dealing with LSMOPs.
2) IM-MOEA/D combines the k-means scheme and selection criterion in the decomposition-based MOEA framework to select the most appropriate reference vector from the population to solve LSMOPs.
3) LMOEA-DS employs the directed sampling strategy to guide the search direction at each generation of the evolutionary search, which effectively improves the overall performance of the algorithm when handling LSMOPs.
4) AGE-MOEA-II is a most recent algorithm to solve LSMOPs, implementing the PF modeling and distance measurement method in the adaptive geometry estimation-based MOEA framework.
5) WOF is a decision variable grouping and problem transformation-based LSMOEA for tackling LSMOPs. In this article, we adopt the enhanced version of WOF as introduced in [29], namely, WOF-SMPSO, in which the SMPSO is used as an optimizer to optimize the weights and variables.
6) LMOCSO uses an innovation update strategy in CSO framework, which has been verified to be capable of tackling LSMOPs with $D < 5000$.

The LSMOP test suite [61] is adopted as the test problems, which is extensible and generalized and follows the basic principles defined in [62]. Among the nine problems LSMOP1–LSMOP9, the PSs in LSMOP1–LSMOP4 are characterized by linear variable linkages and the PFs of which are linear unit hyperplanes. There exist nonlinear variable linkages in the PSs of LSMOP5–LSMOP9, in which LSMOP5–LSMOP8 have concave unit hypersphere in their PFs as well as LSMOP9 has disconnected PF.

A. Experimental Settings

1) Algorithms: To allow a fair comparison, the related parameters of all the compared algorithms are set according to their original references. Concretely, DGEA uses RVEA [63] as its selection operator and the number of direction vectors is set to 10. The number of teams K and model group size L in IM-MOEA/D are set to 10 and 3, respectively. In LMOEA-DS, the threshold, the number of clusters, and the number of random samples along each guiding direction are set to $2N/3$, 10, and 30, respectively. WOF uses SMPSO [64] as the optimizer, and the number of evaluations per optimization for the original problem t_1, the number of evaluations per optimization for the transformed problem t_2, the number of selected solutions q, the number of groups γ, and the evaluation ratios δ for the original and transformed problems are set to 1000, 500, 2, 4, and 0.5, respectively. In LMOCSO, the penalty parameter α of angle-penalized distance is set to 2. LMOEA-DS and AGE-MOEA-II employ simulated binary crossover and polynomial mutation to generate offspring, where the crossover probability is set to 1, the mutation probability is set to $1/D$, in which D is the number of decision variables, and their distribution exponent is set to 20. All algorithms are implemented on a server with a 2.10 GHz (2 processors) Intel Xeon Gold 5218 CPU, 512-GB RAM, Windows 10 64-bit operating system.

2) Population Size: For all the compared LSMOEAs, the size of population N is set to $N = 200$ and $N = 300$ for bi- and tri-objective problems, respectively.

3) Problems: In the experiments, the number of objectives M is set to 2 and 3, and the number of decision variables D is considered to vary from 100 to 5000. In addition, the number of subcomponents n_k for each variable group of each problem is uniformly set to $n_k = 5$, in which its detailed definition can be seen in [61].

4) Termination Criterion: To ensure sufficient convergence of all algorithms, the maximum number of function evaluations is adopted as the termination criterion, which is set to $15\,000 \times D$.

5) Performance Metric: To quantify the performance of LSMOEAs, the inverse generational distance (IGD) [65] and hypervolume (HV) [66] are adopted as the performance indicators. The smaller IGD as well as the larger HV correspond to stronger convergence and diversity of approximate PFs obtained by the algorithm. All experiments are performed for 30 times independently, and the mean and standard deviation of the IGD values are recorded. In addition, Wilcoxon rank-sum (Mann–Whitney) test [67] with a significance level of 0.05 is conducted to assess statistically significant differences between FRCSO and other LSMOEAs. In the statistical experimental results, the symbols "†," "§," and "≈" indicate that the result by another LSMOEAs is significantly better, worse, and similar to the proposed FRCSO, respectively.

B. Ablation Study

In this section, two sets of ablation studies are conducted to verify the effectiveness of the WDS and the loser updating strategy proposed in FRCSO, respectively.

1) Effectiveness of the WDS in FRCSO: As mentioned in Section II, angle-based WDS has been executed in CMOPSO while random comparison-based WDS has been executed in LMOCSO. To verify the effectiveness of proposed WDS in FRCSO, it is compared with two of its variants, called FRCSO-A and FRCSO-B, where FRCSO-A uses the WDS in CMOPSO and FRCSO-B uses the WDS in LMOCSO. Fig. 5 plots the convergent tendency of $\lg(IGD)$ values obtained by FRCSO and its two variants on bi-objective and tri-objective LSMOP2, LSMOP6, and LSMOP8 with $D = 100$ and $D = 500$, respectively, averaged over 30 independent runs. As shown in Fig. 5, when handling LSMOPs, the proposed WDS has a better performance, which benefits from its higher search space utilization. When solving LSMOP2 with the simple landscape, the proposed WDS is capable of driving the population to converge to the PF faster. For LSMOP6 and LSMOP8 with complex landscape, WDS in FRCSO can reliably identify convergence-related nondominated particles and diversity-related high-quality particles to guide the search direction of the population, and the convergence results are

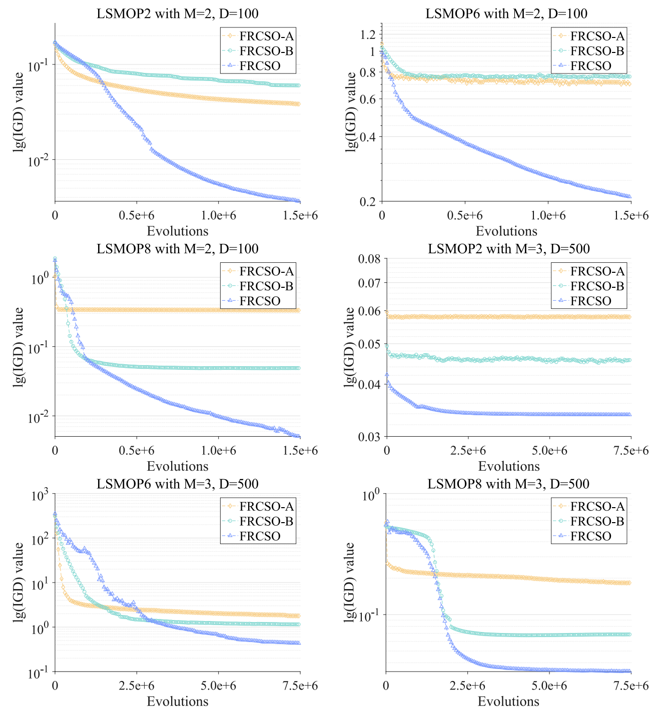

Fig. 5. Convergence profiles of IGD values obtained by FRCSO-A (using WDS in CMOPSO), FRCSO-B (using WDS in LMOCSO), and FRCSO on bi-/tri-objective LSMOP2, LSMOP6, and LSMOP8 with 100/500 decision variables.

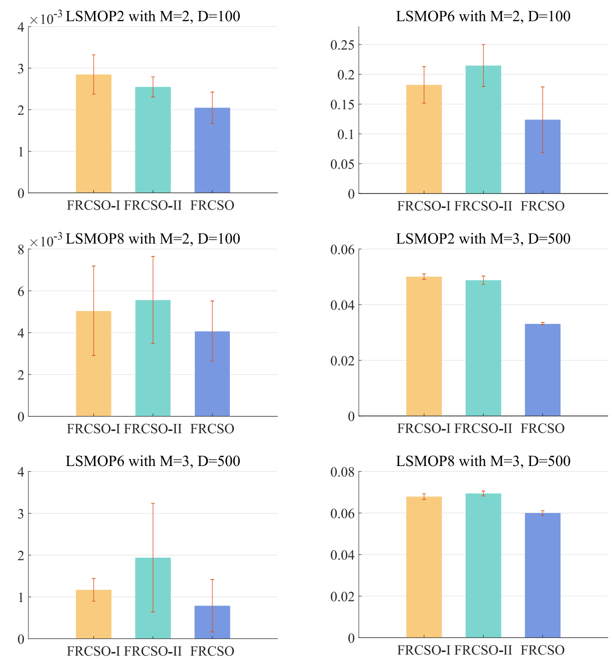

Fig. 6. Convergence profiles of IGD values obtained by FRCSO-I [updating velocities of all losers on the basis of (8)], variants FRCSO-II [updating velocities of all losers on the basis of (10)], and FRCSO on bi-/tri-objective LSMOP2, LSMOP6, and LSMOP8 with 100/500 decision variables.

significantly better than the two variants. Moreover, except in tackling LSMOP6 ($M = 2, D = 100$) and LSMOP2 ($M = 3, D = 500$), the proposed WDS does not have the fastest convergence rate in the early stage of the search, which is due to the large number of diversity-related particles rather than nondominated particles being selected as winners to guide the search. As the search proceeds, the WDS in FRCSO can discover plenty of potential particles to accelerate the convergence of the algorithm. In addition, the WDS in CMOPSO is replaced with the WDS proposed in FRCSO and compared with its original algorithm, and the WDS in LMOCSO is replaced with the WDS proposed in FRCSO and compared with its original algorithm. Due to page limitations, the comparison results are presented in the supplementary material. It can be seen that all algorithms embedded with the WDS proposed in FRCSO achieve great success on almost all test instances, which indicates that the WDS proposed in FRCSO can effectively extract potential individuals as winners to guide the search direction.

2) Effectiveness of the Particle Updating Strategy in FRCSO: To analyze and test the performance of the loser update strategy proposed in FRCSO, it is compared with two of its variants that only use (8) or (10) as the loser update strategies, respectively, which are referred to as FRCSO-I and FRCSO-II, respectively. The final IGD values and error values of FRCSO and its two variants are illustrated as Fig. 6 which are obtained on bi-objective and tri-objective LSMOP2, LSMOP6, and LSMOP8 with $D = 100$ and $D = 500$, respectively, averaged over 30 independent runs. The results show that the proposed loser update strategy has more balanced convergence and diversity than the variant algorithm, and has better superiority in solving LSMOPs. It can be observed from Fig. 6 that the proposed loser update strategy in FRCSO can better balance exploration and exploitation, leading to the best overall performance as well as better stability.

3) Sensitivity to the Base Number α in the Neighborhood Control Parameter δ: To test the effect of the choice of the α in (4), different $\alpha \in \{2, 3, 4, 5, 6\}$ are tested with the rest parameters fixed. Based on the average IGD results from 30 runs collected in the supplementary material, it can been that when $\alpha = 3$, FRCSO has significantly superior performance. In addition, when $M = 2$, FRCSO with $\alpha = 2$ achieves the best results on LSMOP5 and LSMOP8. When $M = 3$, FRCSO with $\alpha = 2$ achieves the smallest IGD value on LSMOP8. FRCSO is not sensitive to α on LSMOP1. Specifically, when the value of α is small, the initial neighborhood size (e.g., the number of $|F_1(t)|$ is small) of each particle is large, which will not only waste search resources but also result in some potential particles to be discarded. Conversely, a large α will correspond to a small initial neighborhood size which will make the inferior particles be regarded as potential particles. Therefore, it is suggested that α is set to 3.

C. Comparisons on Benchmark Problems

Table I exhibits the statistics of the IGD values calculated by DGEA, IM-MOEA/D, LMOEA-DS, AGE-MOEA-II, WOF, and LMOCSO on bi-objective LSMOP1–LSMOP9 with 100, 300, and 500 decision variables, averaging over 30 independent runs. Taking all test problems into consideration, it is observed that the algorithms with the best to worst

TABLE I
STATISTICAL RESULTS (MEAN(STD. DEV.)[RANK]) OF THE SEVEN ALGORITHMS OVER 30 INDEPENDENT RUNS ON BI-OBJECTIVE IN TERMS OF THE IGD METRICS, WHERE THE BEST-RANKED RESULT IS SHOWN IN BOLDED WITH A PROMINENT BACKGROUND

Instance	D	DGEA	IM-MOEA/D	LMOEA-DS	AGE-MOEA-II	WOF	LMOCSO	FRCSO
LSMOP1	100	2.677E-03§[2](1.09E-04)	4.371E-03§[4](2.54E-04)	1.195E-02§[5](2.01E-03)	3.448E-01§[7](1.16E-01)	4.937E-02§[6](3.42E-02)	2.709E-03§[3](1.34E-04)	**2.065E-03**[1](3.37E-05)
	300	4.406E-03§[3](3.81E-04)	8.433E-03§[4](2.52E-03)	9.741E-02§[6](5.31E-03)	3.470E-01§[7](8.95E-02)	3.443E-02§[5](2.02E-02)	1.783E-03§[2](7.68E-07)	**1.385E-03**[1](4.28E-05)
	500	6.500E-03§[4](9.89E-04)	1.580E-02§[5](7.23E-03)	3.265E-03§[3](3.80E-03)	3.468E-01§[6](8.92E-02)	4.813E-01§[7](5.63E-02)	1.783E-03§[2](6.60E-07)	**1.674E-03**[1](6.42E-05)
LSMOP2	100	3.515E-02§[5](3.79E-02)	8.621E-03§[2](4.12E-04)	1.943E-02§[4](1.97E-03)	3.585E-02§[6](2.45E-03)	1.587E-02§[3](3.55E-03)	4.771E-02§[7](1.79E-02)	**2.648E-03**[1](3.76E-04)
	300	1.171E-02§[4](6.86E-04)	5.712E-03§[7](2.96E-04)	1.380E-02§[5](2.82E-04)	2.424E-02§[6](9.58E-04)	1.148E-02§[3](2.03E-03)	4.252E-03§[2](4.95E-04)	**3.128E-03**[1](1.82E-03)
	500	8.210E-03§[3](3.16E-04)	4.439E-03§[2](1.93E-04)	9.982E-03§[4](3.75E-04)	2.378E-02§[6](8.80E-04)	5.764E-01§[7](1.63E-03)	1.062E-02§[5](5.97E-03)	**3.485E-03**[1](2.78E-03)
LSMOP3	100	1.279E+00§[7](6.86E-01)	4.833E-01§[2](3.63E-02)	8.660E-01§[5](1.25E-01)	1.029E+00§[6](2.05E-01)	6.242E-01§[3](7.02E-03)	7.071E-01§[4](5.88E-05)	**3.091E-01**[1](6.68E-02)
	300	1.247E+00§[6](1.43E-01)	**5.015E-01**†[1](1.01E-03)	1.397E+00§[7](1.04E-01)	7.679E-01§[5](1.12E-01)	7.289E-01§[4](2.11E-01)	7.151E-01§[3](4.34E-05)	6.187E-01[2](1.80E-02)
	500	1.247E+00§[6](2.20E-01)	5.011E-01†[2](7.92E-04)	1.515E+00§[7](1.84E-02)	7.732E-01§[5](1.43E-02)	**1.125E-02**†[1](4.36E-02)	7.471E-01§[4](1.24E-04)	5.595E-01[3](2.03E-01)
LSMOP4	100	1.430E-02§[2](3.94E-03)	1.514E-02§[3](6.51E-04)	1.921E-02§[4](7.35E-04)	5.147E-02§[7](2.12E-03)	5.004E-02§[6](2.07E-02)	2.598E-02§[5](1.50E-03)	**2.875E-03**[1](4.44E-04)
	300	7.861E-03§[2](3.41E-04)	9.274E-03§[3](5.56E-04)	1.040E-02§[4](4.61E-04)	3.317E-02§[6](1.16E-03)	3.559E-02§[7](9.15E-03)	1.146E-02§[5](5.22E-05)	**4.107E-03**[1](9.91E-04)
	500	6.360E-03§[2](2.64E-04)	7.168E-03§[3](4.28E-04)	1.084E-02§[5](1.38E-03)	2.363E-02§[6](6.83E-04)	5.482E-01§[7](5.31E-03)	8.690E-03§[4](4.68E-04)	**5.088E-03**[1](1.15E-04)
LSMOP5	100	3.460E-03§[2](5.48E-04)	3.246E-02§[4](2.21E-02)	1.867E-01§[6](1.74E-03)	3.424E-01§[7](6.82E-07)	5.582E-02§[5](3.44E-02)	3.495E-03§[3](2.00E-04)	**2.128E-03**[1](3.70E-05)
	300	8.161E-03§[3](1.57E-03)	5.343E-02§[4](6.93E-03)	1.765E-01§[6](3.53E-02)	3.615E-01§[7](3.45E-02)	1.129E-01§[5](1.05E-01)	2.911E-03§[2](7.79E-06)	**2.238E-03**[1](2.96E-05)
	500	1.189E-02§[3](1.27E-03)	5.525E-02§[4](4.59E-03)	1.854E-01§[5](1.24E-03)	3.719E-01§[7](9.00E-02)	2.278E-01§[6](5.19E-02)	2.720E-03§[2](8.72E-06)	**2.323E-03**[1](3.56E-05)
LSMOP6	100	3.743E-01§[4](2.73E-01)	4.807E-01§[5](9.87E-02)	1.808E-01§[2](1.93E-02)	7.331E-01§[6](8.98E-02)	3.342E-01§[3](1.28E-01)	7.536E-01§[7](8.74E-03)	**1.039E-01**[1](5.50E-02)
	300	1.806E-01§[2](1.11E-01)	5.757E-01§[7](1.29E-01)	1.874E-01§[3](1.46E-02)	5.219E-01§[6](3.61E-02)	3.247E-01§[5](5.74E-02)	2.871E-01§[4](4.06E-02)	**1.436E-01**[1](4.80E-02)
	500	1.184E-01§[2](8.20E-02)	6.406E-01§[7](1.18E-01)	1.918E-01§[4](1.25E-02)	4.558E-01§[5](2.50E-02)	1.538E-01§[3](6.10E-03)	5.669E-01§[6](1.56E-01)	**1.126E-01**[1](4.90E-02)
LSMOP7	100	1.078E+00§[6](2.88E-01)	8.920E-01§[4](3.78E-01)	9.047E-01§[5](2.11E-01)	7.780E-01§[2](2.03E-01)	8.277E-01§[3](3.69E-02)	1.099E+00§[7](2.92E-01)	**7.089E-01**[1](2.75E-01)
	300	7.953E-01†[2](5.15E-01)	1.289E+00§[6](4.31E-01)	1.410E+00§[7](1.11E-01)	**7.870E-01**†[1](2.55E-02)	9.888E-01§[5](1.19E-01)	9.837E-01§[4](5.70E-01)	8.938E-01[3](4.85E-01)
	500	1.240E+00§[5](4.13E-01)	9.777E-01§[4](5.57E-01)	1.490E+00§[7](8.27E-03)	7.839E-01§[2](3.93E-02)	1.418E+00§[6](1.12E-01)	8.497E-01§[3](6.38E-01)	**6.953E-01**[1](5.59E-01)
LSMOP8	100	4.685E-02§[6](4.26E-03)	1.151E-02§[2](5.97E-04)	3.064E-02§[3](1.61E-02)	3.469E-01§[4](1.54E-03)	7.329E-02§[7](2.24E-02)	3.532E-02§[5](7.15E-03)	**4.076E-03**[1](1.44E-03)
	300	1.921E-02§[4](6.47E-03)	1.247E-02§[2](1.07E-03)	1.856E-02§[3](9.90E-04)	3.429E-01§[7](4.07E-04)	4.628E-02§[6](1.76E-02)	3.266E-02§[5](1.97E-03)	**8.768E-03**[1](2.22E-03)
	500	1.354E-02†[2](2.71E-03)	**9.752E-03**†[1](8.90E-04)	1.829E-02§[4](9.86E-04)	3.426E-01§[7](2.75E-04)	3.135E-01§[6](9.05E-03)	2.550E-02§[5](4.36E-04)	1.555E-02[3](4.09E-03)
LSMOP9	100	3.757E-01§[4](3.97E-01)	2.029E-01§[3](2.33E-01)	8.100E-01§[7](1.17E-06)	8.100E-01§[6](1.89E-08)	7.688E-01§[5](1.60E-01)	3.271E-02§[2](2.01E-03)	**1.679E-03**[1](1.40E-03)
	300	6.761E-02§[4](2.86E-02)	8.768E-03§[2](5.16E-04)	5.546E-01§[5](1.71E-01)	8.103E-01§[7](3.00E-05)	5.843E-01§[6](2.19E-01)	3.293E-02§[3](1.14E-02)	**2.688E-03**[1](3.28E-05)
	500	1.607E-01§[4](9.18E-02)	8.058E-03§[2](6.36E-04)	2.613E-01§[6](5.59E-02)	8.095E-01§[7](5.75E-05)	1.970E-01§[5](3.02E-02)	4.126E-02§[3](1.60E-04)	**2.903E-03**[1](5.36E-05)
Mean Rank		3.667	3.519	4.889	5.704	5.000	3.963	1.259
†/§/≈		2/25/0	3/24/0	0/27/0	1/26/0	1/26/0	0/27/0	

"†", "§", and "≈" indicate that the statistical result is significantly better, significantly worse, and significantly similar to that of FRCSO, respectively.

performance are FRCSO, IM-MOEA/D, DGEA, LMOCSO, LMOEA-DS, WOF, and AGE-MOEA-II according to the ranking. Specifically, FRCSO outperformed other comparison algorithms on 23 target test questions and had the best overall performance. Among these LSMOEAs, IM-MOEA/D and LMOCSO perform very closely on most of the test problems, while FRCSO is more robustness than the other algorithms. Nonetheless, the performance of FRCSO on handling LSMOP3 with multimodal landscape needs to be further improved. Furthermore, according to the Mann–Whitney test, in the comparisons, FRCSO achieves 25, 24, 27, 26, 26, and 27 better, 2, 3, 0, 1, 1, and 0 worse, and no similar IGD values than DGEA, IM-MOEA/D, LMOEA-DS, AGE-MOEA-II, WOF, and LMOCSO, respectively. Table II lists the IGD values of the seven compared LSMOEAs on the tri-objective LSMOP1–LSMOP9 with 100, 300, and 500 decision variables, respectively. It can be seen that FRCSO achieves the best in the test problems, and the final score is clearly the best in the evaluation indicators. Furthermore, the proportion of test problems in which FRCSO performs significantly better than DGEA, IM-MOEA/D, LMOEA-DS, AGE-MOEA-II, WOF, and LMOCSO is 23/27, 21/27, 24/27, 24/27, 26/27, and 26/27, respectively. The experimental results verify the effectiveness of FRCSO in solving LSMOPs.

For further observation, the three algorithms with the best performance in the comprehensive ranking are selected for visual display, namely, DGEA, IM-MOEA/D, and FRCSO, and depict the nondominated solutions with the median IGD values obtained on bi- and tri-objective LSMOP1, LSMOP5, and LSMOP9 with 500 decision variables, as shown in Fig. 7. It can be seen that when $M = 2$, for LSMOP1 with a simple landscape, the three algorithms have better diversity. It can be seen that although the final population obtained by DGEA is very close to PF, it still does not converge to PF and maintains a certain distance. However, the final population produced by FRCSO almost completely covers the PF. For LSMOP5, IM-MOEA/D cannot obtain a set of solutions with good convergence and diversity, while DGEA can maintain good diversity but it is difficult to approximate to the PF. When solving LSMOP9 with the discontinuous PF, the convergence and diversity of the solution sets obtained by the two comparison algorithms are not satisfactory. When $M = 3$, the performance of FRCSO in solving the three LSMOPs is significantly better than that of the two comparison algorithms, which is due to the fact that the proposed WDS can search enough high-quality individuals, and then the update strategy is capable of giving full play to the advantages of these winners and iteratively guiding the losers to accelerate the approximation to the PF.

D. Scalability to Higher Dimensionality

In this section, the scalability of FRCSO in handling LSMOPs with respect to the number of decision variables is further investigated. Fig. 8 shows the averaged IGD values of FRCSO and other compared LSMOEAs on tri-objectives LSMOP1, LSMOP5, and LSMOP9 over 30 runs, in which the number of decision variables of LSMOP1–LSMOP9 is set to $D = \{100, 300, 500, 800, 1000, 1500, 2500, 5000\}$, respectively. The number of evaluations is all set to $15\,000 \times D$.

TABLE II
STATISTICAL RESULTS (MEAN(STD. DEV.)[RANK]) OF THE SEVEN ALGORITHMS OVER 30 INDEPENDENT RUNS ON TRI-OBJECTIVE IN TERMS OF THE IGD METRICS, WHERE THE BEST-RANKED RESULT IS SHOWN IN BOLDED WITH A PROMINENT BACKGROUND

Instance	D	DGEA	IM-MOEA/D	LMOEA-DS	AGE-MOEA-II	WOF	LMOCSO	FRCSO
LSMOP1	100	1.709E-01§[7](5.36E-02)	9.677E-02§[4](9.96E-03)	1.690E-01§[6](3.81E-02)	1.525E-01§[5](4.11E-02)	5.516E-02§[2](1.03E-02)	5.874E-02§[3](1.29E-02)	**1.177E-02**[1](7.04E-04)
	300	2.829E-01§[7](1.04E-02)	1.744E-01§[5](3.48E-03)	2.567E-01§[6](3.87E-02)	1.624E-01§[4](1.70E-02)	5.993E-02§[2](7.41E-03)	8.773E-02§[3](1.11E-02)	**3.520E-02**[1](8.10E-04)
	500	3.006E-01§[5](6.79E-03)	2.076E-01§[4](9.63E-03)	3.436E-01§[7](1.33E-02)	3.324E-01§[6](9.00E-02)	6.721E-02§[2](8.72E-03)	9.189E-02§[3](8.40E-03)	**4.617E-02**[1](1.12E-02)
LSMOP2	100	8.930E-02§[6](1.81E-02)	5.341E-02§[2](6.34E-04)	7.279E-02§[5](1.72E-03)	6.475E-02§[3](3.45E-03)	9.063E-02§[7](7.70E-03)	6.496E-02§[4](1.91E-03)	**4.543E-02**[1](4.35E-03)
	300	3.553E-02§[2](1.40E-03)	4.017E-02§[4](2.67E-04)	4.208E-02§[5](7.28E-04)	3.910E-02§[3](9.44E-04)	4.645E-02§[7](2.47E-03)	4.429E-02§[6](7.76E-04)	**3.183E-02**[1](1.34E-03)
	500	3.892E-02§[5](1.26E-03)	3.631E-02§[4](3.15E-04)	4.796E-02§[6](7.59E-04)	4.948E-02§[7](3.62E-04)	3.364E-02§[2](1.81E-03)	3.385E-02§[3](4.34E-04)	**3.106E-02**[1](1.43E-03)
LSMOP3	100	5.583E-01†[3](3.54E-02)	4.386E-01†[2](4.06E-02)	6.191E-01$^{\approx}$[4](1.14E-01)	**4.277E-01**†[1](1.18E-01)	7.921E-01§[7](1.22E-01)	7.494E-01§[6](6.01E-02)	6.518E-01[5](1.45E-01)
	300	5.452E-01§[4](2.80E-02)	**4.906E-01**$^{\approx}$[1](9.36E-03)	6.708E-01§[6](1.29E-01)	5.692E-01§[5](2.83E-02)	5.314E-01§[3](2.12E-01)	7.908E-01§[7](6.43E-02)	5.041E-01[2](3.20E-01)
	500	**5.121E-01**$^{\approx}$[1](2.41E-02)	5.195E-01$^{\approx}$[3](8.26E-03)	8.277E-01§[6](3.77E-02)	7.862E-01§[5](1.68E-01)	5.427E-01§[4](2.14E-01)	8.415E-01§[7](3.38E-02)	5.166E-01[2](2.09E-01)
LSMOP4	100	1.595E-01§[6](1.77E-02)	1.128E-01§[5](5.54E-03)	1.958E-01§[7](1.88E-02)	9.153E-02§[2](1.68E-02)	1.017E-01§[3](6.36E-03)	1.107E-01§[4](1.53E-02)	**5.799E-02**[1](8.80E-03)
	300	1.151E-01§[7](3.65E-03)	**6.129E-02**†[1](1.58E-03)	9.818E-02§[6](5.13E-03)	8.738E-02§[4](7.40E-03)	9.560E-02§[5](4.18E-03)	8.152E-02§[3](6.98E-03)	6.481E-02[2](6.28E-03)
	500	7.912E-02§[5](2.20E-03)	6.726E-02$^{\approx}$[3](8.21E-04)	1.275E-01§[7](1.74E-03)	1.050E-01§[6](4.60E-03)	7.070E-02§[4](3.11E-03)	**6.493E-02**$^{\approx}$[1](4.34E-03)	6.705E-02[2](3.11E-03)
LSMOP5	100	2.520E-01§[4](1.05E-01)	1.847E-01§[3](2.31E-02)	2.629E-01§[5](1.44E-02)	3.522E-01§[7](1.70E-02)	3.373E-01§[6](6.30E-02)	7.593E-02§[2](4.28E-02)	**3.639E-02**[1](1.45E-03)
	300	2.273E-01§[3](3.08E-03)	2.102E-01§[2](1.83E-02)	2.806E-01§[5](1.03E-02)	3.713E-01§[7](5.90E-02)	2.562E-01§[4](1.27E-01)	2.885E-01§[6](8.92E-02)	**3.832E-02**[1](1.69E-03)
	500	2.291E-01§[3](3.62E-03)	2.166E-01§[2](1.07E-02)	5.745E-01§[7](1.74E-02)	5.221E-01§[6](2.14E-01)	2.616E-01§[4](1.18E-01)	4.225E-01§[5](1.41E-01)	**4.108E-02**[1](1.85E-03)
LSMOP6	100	6.997E-01$^{\approx}$[3](2.21E-01)	7.012E-01$^{\approx}$[4](1.48E-01)	**5.420E-01**†[1](5.96E-02)	9.394E-01§[7](1.79E-01)	7.526E-01§[5](5.55E-02)	8.383E-01§[6](9.11E-02)	6.727E-01[2](1.06E-01)
	300	6.449E-01§[2](1.88E-01)	9.347E-01§[5](2.95E-01)	7.628E-01§[3](1.41E-01)	8.217E-01§[4](1.07E-01)	1.132E+00§[7](2.02E-02)	1.186E+00§[6](1.90E-01)	**4.567E-01**[1](2.79E-01)
	500	6.069E-01§[2](1.56E-01)	1.211E+00§[6](4.06E-01)	8.032E-01§[4](2.21E-01)	7.672E-01§[3](5.25E-02)	1.214E+00§[7](9.28E-03)	1.003E+00§[5](3.71E-01)	**5.562E-01**[1](6.28E-01)
LSMOP7	100	7.845E-01§[5](6.35E-02)	7.084E-01§[3](7.61E-02)	6.944E-01$^{\approx}$[2](3.53E-02)	8.386E-01§[6](1.23E-01)	7.189E-01§[4](3.45E-02)	9.457E-01§[7](8.17E-04)	**6.663E-01**[1](1.88E-01)
	300	7.970E-01§[4](5.67E-02)	6.310E-01$^{\approx}$[2](1.15E-01)	8.459E-01§[6](1.30E-02)	6.467E-01$^{\approx}$[3](2.53E-02)	8.018E-01§[5](5.70E-03)	9.459E-01§[7](3.74E-07)	**6.191E-01**[1](4.34E-01)
	500	7.707E-01§[4](1.00E-01)	7.352E-01§[3](1.91E-01)	8.513E-01§[6](3.56E-02)	6.221E-01$^{\approx}$[2](2.14E-01)	8.160E-01§[5](3.25E-03)	9.455E-01§[7](1.87E-03)	**6.194E-01**[1](1.70E-01)
LSMOP8	100	1.528E-01§[4](4.01E-02)	9.990E-02§[2](1.95E-02)	3.645E-01§[7](8.72E-02)	3.639E-01§[6](9.30E-04)	1.602E-01§[5](1.21E-01)	1.009E-01§[3](1.00E-02)	**6.420E-02**[1](5.46E-03)
	300	7.086E-02$^{\approx}$[3](1.47E-02)	1.140E-01§[5](1.23E-02)	1.430E-01§[6](8.08E-02)	3.605E-01§[7](8.31E-03)	6.722E-02$^{\approx}$[2](1.91E-02)	8.685E-02§[4](3.51E-03)	**6.819E-02**[1](1.71E-03)
	500	6.633E-02§[3](1.74E-03)	1.891E-01§[6](9.22E-03)	8.880E-02§[5](6.99E-03)	3.090E-01§[7](9.16E-02)	6.482E-02§[2](1.54E-02)	6.854E-02§[4](1.87E-03)	**5.848E-02**[1](1.10E-03)
LSMOP9	100	6.356E-01§[5](1.11E-01)	2.695E-01§[3](1.24E-01)	5.873E-01§[4](7.58E-04)	1.088E+00§[6](1.81E-01)	1.148E+00§[7](1.08E-02)	1.328E-01§[2](7.83E-02)	**1.105E-01**[1](1.66E-01)
	300	5.924E-01§[5](3.77E-02)	1.705E-01§[3](9.88E-02)	5.736E-01§[4](2.37E-03)	1.145E+00§[7](1.32E-04)	1.019E+00§[6](2.39E-01)	1.619E-01§[2](5.07E-02)	**1.405E-01**[1](1.35E-01)
	500	6.124E-01§[5](1.23E-01)	1.863E-01§[2](9.16E-02)	5.733E-01§[4](1.06E-03)	1.091E+00§[7](1.70E-01)	1.094E+00§[7](1.64E-01)	4.344E-01§[3](4.91E-02)	**1.624E-01**[1](1.65E-01)
Mean Rank		3.852	3.296	5.185	5.037	4.593	4.407	1.333
†/§/≈		1/23/3	1/21/5	1/24/2	1/24/2	0/26/1	0/26/1	

"†", "§", and "≈" indicate that the statistical result is significantly better, significantly worse, and significantly similar to that of FRCSO, respectively.

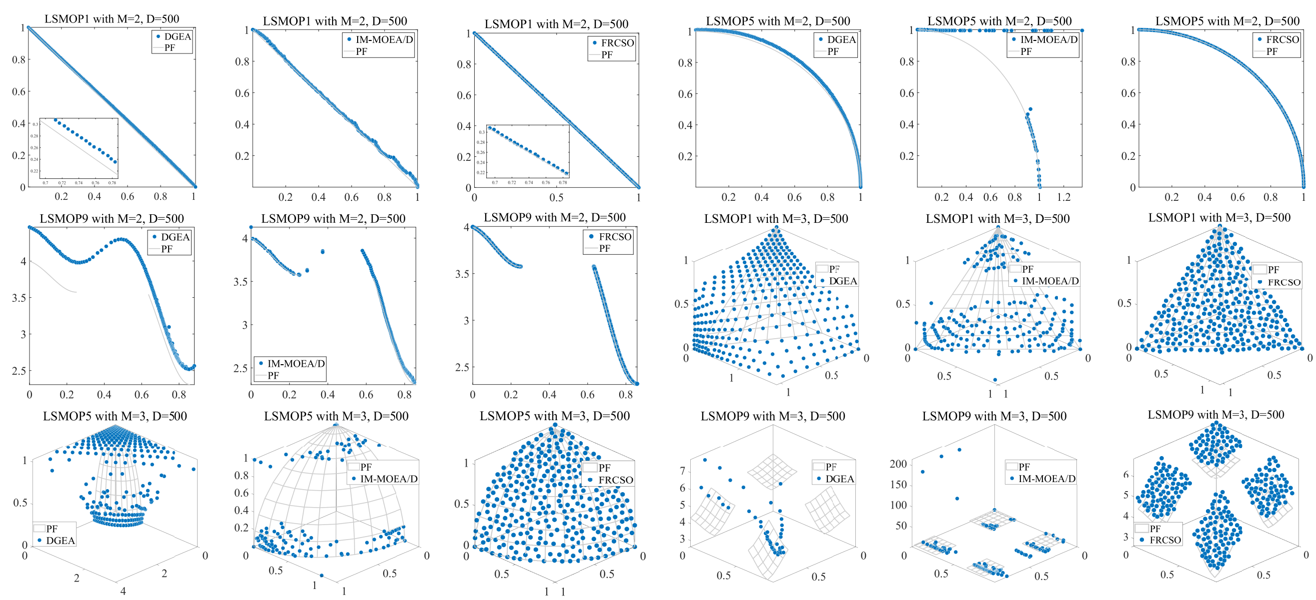

Fig. 7. Nondominated solutions with the median IGD values obtained by DGEA, IM-MOEA/D, and FRCSO on bi-/tri-objective LSMOP1, LSMOP5, and LSMOP9 with 500 decision variables.

For LSMOP1 with simple landscape, the IGD values obtained by FRCSO are better than those obtained by compared LSMOEAs with different number of decision variables. As for LSMOP5 with nonlinear variable linkage Pareto set and concave PF, FRCSO still shows significant advantages over these six competitors with decision variables ranges from 100 to 5000. When the number of decision variables ranges from 100 to 500, FRCSO has shown to perform fairly well for LSMOP9 with disconnected PF, and its performance is acceptable as $D = \{1000, 2500, 5000\}$. However, as decision $D = \{800, 1500\}$, the performance of FRCSO is mediocre. Due to page limitations, the experimental results on all problems with up to 5000 dimensions are provided in S-Table III and S-Table IV in the supplementary material. In summary, FRCSO represents similar performance on the same LSMOPs with different numbers of decision variables, and outperforms

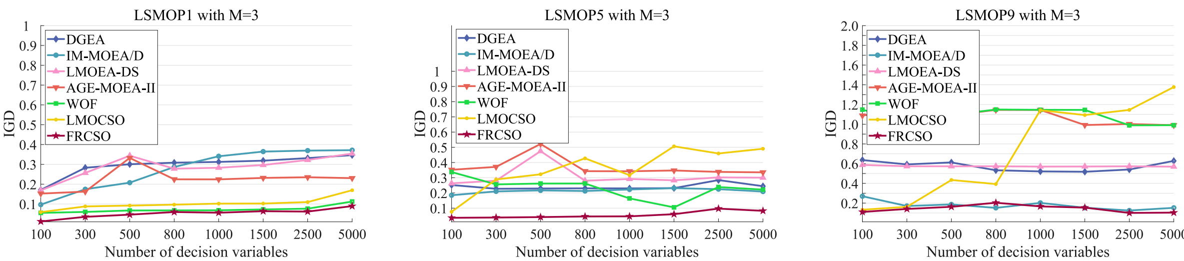

Fig. 8. IGD values obtained by DGEA, IM-MOEA/D, LMOEA-DS, AGE-MOEA-II, WOF, LMOCSO, and FRCSO on three-objective LSMOP1, LSMOP5, and LSMOP9 with the number of decision variables ranging from 100 to 5000.

the comparison algorithms on most problems. FRCSO has a stronger performance when handling the LSMOPs with $M = 2$, while the performance of FRCSO deteriorates on LSMOP3 with $M = 3$. As for LSMOP4, the IGD values calculated by FRCSO are relatively small when the number of decision variables varies from 1500 to 5000, and when the number of decision variables varies from 300 to 1000, the performance of FRCSO is slightly inferior. Hence, it can be confirmed that FRCSO has a good scalability with respect to the number of decision variables on most test instances when the number of evaluations is linearly related to the number of decision variables.

E. Comparisons With State-of-the-Art CSO-Based Algorithms

FRCSO is proposed based on single-objective CSO, which shows great potential in solving LSMOPs. To further verify the effectiveness of the proposed FRCSO, it is compared with two state-of-the-art CSO-based algorithms, namely, S-ECSO [50] and MPSO-EBCD [54]. Both comparison algorithms adopt the parameter values recommended in the literature, whose specific settings are provided in the supplementary material. The average IGD values on bi-/tri-objective LSMOP1–LSMOP9 over 30 runs are presented in S-Table V and S-Table VI in the supplementary material, where the decision variables are set to $D = \{100, 300, 500, 800, 1000, 1500, 2500, 5000\}$, respectively.

As summarized in S-Table V and S-Table VI in the supplementary material, FRCSO obtains 139/112 better, 4/20 worse, and 1/2 similar IGD values than S-ECSO and MPSO-EBCD, respectively. Obviously, our proposed FRCSO exhibits significantly better overall performance than its two competitors as it obtains the best performance in most cases of the IGD results. Concretely, when the number of decision variables varies from 500 to 5000, the IGD values obtained by FRCSO are relatively small. The reason is that when the decision space exponentially increases, the proposed WDS can still extract high-quality particles to guide the search direction. For LSMOPs with complex landscapes, such as LSMOP5, LSMOP6, LSMOP7, and LSMOP9, FRCSO performs better than the two comparison algorithms as the proposed WDS and loser update strategy can jointly promote the diversity maintenance of the population. However, the performance of FRCSO deteriorates on LSMOP3 with $M = 2$, $D = \{2500, 5000\}$ and $M = 3$, $D = \{1000, 1500, 2500, 5000\}$. In addition, FRCSO shows a slightly inferior scalability when

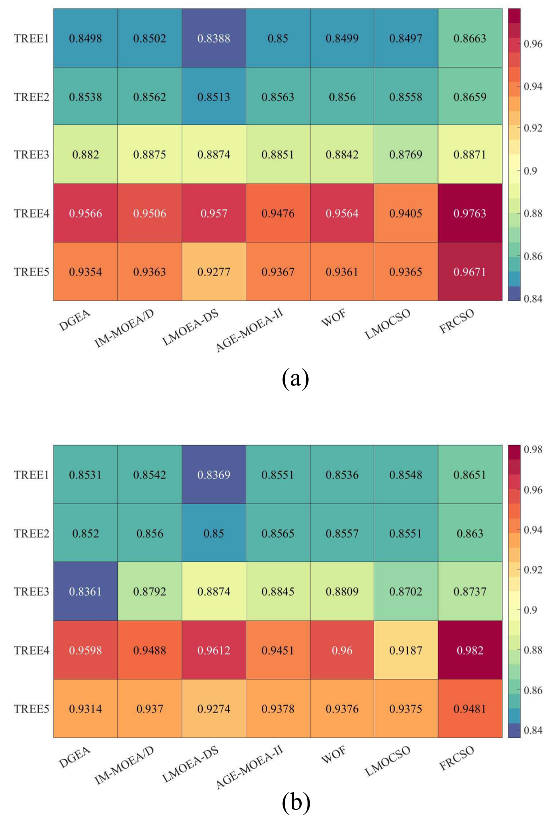

Fig. 9. Median HV value obtained by DGEA, IM-MOEA/D, LMOEA-DS, AGE-MOEA-II, WOF, LMOCSO, and FRCSO on TREE problems with $D = \{1500, 3000\}$ among 30 runs. (a) $D = 1500$. (b) $D = 3000$.

handling LSMOP8 with $M = 2$. Overall, the statistical results validate the superior performance of the proposed FRCSO in tackling the large-scale search space of the LSMOPs.

F. Applications to the Real-World Problems

To further validate the performance of the proposed FRCSO and six compared LSMOEAs in tackling real-world problems, they are employed to optimize the TREE problem [68] contains two objectives with three or four constraints. The multiobjective and inequality constraints in the TREE are expressed by statistical and physical rules extracted from the power delivery systems aiming to minimize the total ratio error and the ratio error variation, and to satisfy the topological, sequence, and phase constraints. This section considers the number of decision variables for TREE as $D = 1500$ and $D = 3000$.

Fig. 9 illustrates the heat maps of the median HV value obtained by the comparison algorithms and FRCSO on the TREE problems with 1500 and 3000 decision variables, respectively, in which the population size is set 200 and the maximum number of evaluations is set to $15\,000 \times D$. As can be seen in Fig. 9(a), when $D = 1500$, FRCSO exhibits better overall performance than the other LSMOEAs in dealing with the TREE problems, obtaining the best median HV values on four of the five problems. When $D = 3000$ as observed in Fig. 9(b), FRCSO performs significantly better than the comparison algorithms on problems TREE1, TREE2, TREE4, and TREE5. The performance obtained on the TREE3 with different numbers of decision variables ranks third and fifth, respectively. The experimental results indicate that the existing LSMOEAs are not very competitive in handling real-world LSMOPs, while FRCSO is capable of performing higher effectiveness and efficiency.

V. Conclusion

In this article, we have proposed a flexible ranking-based CSO, termed FRCSO. Different from traditional CSO and its variants, which only focus on loser updating strategy to improve the search performance, the proposed algorithm modifies and applies a novel WDS, which can reliably identify convergence-related nondominated particles and diversity-related high-quality particles in a single run to avoid trapping into local optimum. Furthermore, FRCSO implements different loser updating strategies in different search stages to enhance search efficiency and diversity maintenance. Simulation results on both large-scale benchmark MOPs and application examples demonstrate the superiority of the FRCSO over several state-of-the-art LSMOEAs. Nonetheless, it can be seen that the performance of FRCSO in tackling the LSMOP with multimodal landscapes needs to be further improved.

In our future work, the development of a more powerful WDS is conducive to the reasonable allocation of limited computability to the region of potential high-quality solutions, further increasing the search space utilization. In addition, it is desirable to design novel CSOs with superior diversity for solving real-world LSMOPs, and the powerful CSO-based environment selection strategy may be one of the promising research directions.

References

[1] K. Weinert, A. Zabel, P. Kersting, T. Michelitsch, and T. Wagner, "On the use of problem-specific candidate generators for the hybrid optimization of multi-objective production engineering problems," *Evol. Comput.*, vol. 17, no. 4, pp. 527–544, 2009.

[2] J. Yi, D. Huang, S. Fu, H. He, and T. Li, "Multi-objective bacterial foraging optimization algorithm based on parallel cell entropy for aluminum electrolysis production process," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2488–2500, Apr. 2016.

[3] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.

[4] A. Ponsich, A. L. Jaimes, and C. A. Coello Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, Jun. 2013.

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

[6] T. Erps et al., "Accelerated discovery of 3-D printing materials using data-driven multiobjective optimization," *Sci. Adv.*, vol. 7, no. 42, 2021, Art. no. eabf7435.

[7] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, and X. Yao, "A scalable indicator-based evolutionary algorithm for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 525–537, Jun. 2019.

[8] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 397–415, May 2008.

[9] Y. Xiang, Y. Zhou, Z. Zheng, and M. Li, "Configuring software product lines by combining many-objective optimization and SAT solvers," *ACM Trans. Softw. Eng. Methodol.*, vol. 26, no. 4, pp. 1–46, 2018.

[10] J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck, "Portfolio optimization with an envelope-based multi-objective evolutionary algorithm," *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 684–693, 2009.

[11] X. Ma et al., "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, Apr. 2016.

[12] Y. Tian et al., "Evolutionary large-scale multi-objective optimization: A survey," *ACM Comput. Surveys*, vol. 54, no. 8, pp. 1–34, 2021.

[13] S. Liu, Q. Lin, K.-C. Wong, Q. Li, and K. C. Tan, "Evolutionary large-scale multiobjective optimization: Benchmarks and algorithms," *IEEE Trans. Evol. Comput.*, vol. 27, no. 3, pp. 401–415, Jun. 2023.

[14] Q. Deng, Q. Kang, L. Zhang, M. C. Zhou, and J. An, "Objective space-based population generation to accelerate evolutionary algorithms for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 326–340, Apr. 2023.

[15] S. Qin, C. Sun, Y. Jin, Y. Tan, and J. Fieldsend, "Large-scale evolutionary multiobjective optimization assisted by directed sampling," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 724–738, Aug. 2021.

[16] Y. Tian, H. Chen, H. Ma, X. Zhang, K. C. Tan, and Y. Jin, "Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 10, pp. 1801–1817, Oct. 2022.

[17] Q. Yang, W.-N. Chen, J. Da Deng, Y. Li, T. Gu, and J. Zhang, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 578–594, Aug. 2018.

[18] L. M. Antonio and C. A. Coello Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2013, pp. 2758–2765.

[19] A. Song, Q. Yang, W.-N. Chen, and J. Zhang, "A random-based dynamic grouping strategy for large scale multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 468–475.

[20] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. De Feitas, "Bayesian optimization in a billion dimensions via random embeddings," *J. Artif. Intell. Res.*, vol. 55, pp. 361–387, Feb. 2016.

[21] H. Qian and Y. Yu, "Solving high-dimensional multi-objective optimization problems with low effective dimensions," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 875–881.

[22] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 1, 2005, pp. 443–450.

[23] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, Jun. 2014.

[24] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, Feb. 2018.

[25] H. Chen, X. Zhu, W. Pedrycz, S. Yin, G. Wu, and H. Yan, "PEA: Parallel evolutionary algorithm by separating convergence and diversity for large-scale multi-objective optimization," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 223–232.

[26] H. Bai, R. Cheng, D. Yazdani, K. C. Tan, and Y. Jin, "Evolutionary large-scale dynamic optimization using bilevel variable grouping," *IEEE Trans. Cybern.*, vol. 53, no. 11, pp. 6937–6950, Nov. 2023.

[27] S. Liu, Q. Lin, Y. Tian, and K. C. Tan, "A variable importance-based differential evolution for large-scale multiobjective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13048–13062, Dec. 2022.

[28] C. He, R. Cheng, L. Li, K. C. Tan, and Y. Jin, "Large-scale multiobjective optimization via reformulated decision variable analysis," *IEEE Trans. Evol. Comput.*, early access, Oct. 10, 2022, doi: 10.1109/TEVC.2022.3213006.

[29] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 260–275, Apr. 2018.

[30] C. He et al., "Accelerating large-scale multiobjective optimization via problem reformulation," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 949–961, Dec. 2019.

[31] R. Liu, J. Liu, Y. Li, and J. Liu, "A random dynamic grouping based weight optimization framework for large-scale multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 55, Jun. 2020, Art. no. 100684.

[32] R. Liu, R. Ren, J. Liu, and J. Liu, "A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106120.

[33] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3115–3128, Jun. 2021.

[34] S. Liu, Q. Lin, L. Feng, K.-C. Wong, and K. C. Tan, "Evolutionary multitasking for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 863–877, Aug. 2023.

[35] S. Liu, M. Jiang, Q. Lin, and K. C. Tan, "Evolutionary large-scale multiobjective optimization via self-guided problem transformation," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2022, pp. 1–8.

[36] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12. New York, NY, USA: Springer, 2012.

[37] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.

[38] L. R. Farias and A. F. Araújo, "IM-MOEA/D: An inverse modeling multi-objective evolutionary algorithm based on decomposition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, 2021, pp. 462–467.

[39] S. Liu, J. Li, Q. Lin, Y. Tian, and K. C. Tan, "Learning to accelerate evolutionary search for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 1, pp. 67–81, Feb. 2023.

[40] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multiobjective optimization based on a competitive swarm optimizer," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3696–3708, Aug. 2020.

[41] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.

[42] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[43] C. Sun, J. Ding, J. Zeng, and Y. Jin, "A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems," *Memet. Comput.*, vol. 10, no. 2, pp. 123–134, 2018.

[44] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.

[45] G. Xiong and D. Shi, "Orthogonal learning competitive swarm optimizer for economic dispatch problems," *Appl. Soft Comput.*, vol. 66, pp. 134–148, May 2018.

[46] Y. Wang, Z. Yang, Y. Guo, J. Zhu, and X. Zhu, "A novel multi-objective competitive swarm optimization algorithm for multi-modal multi-objective problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2019, pp. 271–278.

[47] K. N. Das and R. P. Parouha, "An ideal tri-population approach for unconstrained optimization and applications," *Appl. Math. Comput.*, vol. 256, pp. 666–701, Apr. 2015.

[48] J. Zhou, W. Fang, X. Wu, J. Sun, and S. Cheng, "An opposition-based learning competitive particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2016, pp. 515–521.

[49] P. Mohapatra, K. N. Das, and S. Roy, "A modified competitive swarm optimizer for large scale optimization problems," *Appl. Soft Comput.*, vol. 59, pp. 340–362, Oct. 2017.

[50] X. Wang, K. Zhang, J. Wang, and Y. Jin, "An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 859–871, Oct. 2022.

[51] S. Qi, J. Zou, S. Yang, Y. Jin, J. Zheng, and X. Yang, "A self-exploratory competitive swarm optimization algorithm for large-scale multiobjective optimization," *Inf. Sci.*, vol. 609, pp. 1601–1620, Sep. 2022.

[52] W. Li, Z. Lei, J. Yuan, H. Luo, and Q. Xu, "Enhancing the competitive swarm optimizer with covariance matrix adaptation for large scale optimization," *Appl. Intell.*, vol. 51, no. 7, pp. 4984–5006, 2021.

[53] S. Liu, Q. Lin, Q. Li, and K. C. Tan, "A comprehensive competitive swarm optimizer for large-scale multiobjective optimization," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 9, pp. 5829–5842, Sep. 2022.

[54] D. Li, L. Wang, L. Li, W. Guo, Q. Wu, and A. Lerch, "A large-scale multiobjective particle swarm optimizer with enhanced balance of convergence and diversity," *IEEE Trans. Cybern.*, early access, Dec. 19, 2022, doi: 10.1109/TCYB.2022.3225341.

[55] L. C. Cagnina, S. C. Esquivel, and C. A. Coello Coello, "Solving constrained optimization problems with a hybrid particle swarm optimization algorithm," *Eng. Optim.*, vol. 43, no. 8, pp. 843–866, 2011.

[56] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Inf. Sci.*, vol. 427, pp. 63–76, Feb. 2018.

[57] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Comput. Eng. Netw. Lab., ETH Zurich, Zürich, Switzerland, TIK Rep. 103, 2001.

[58] C. Kaspar, B. Ravoo, W. G. van der Wiel, S. Wegner, and W. Pernice, "The rise of intelligent matter," *Nature*, vol. 594, no. 7863, pp. 345–355, 2021.

[59] C. He, R. Cheng, and D. Yazdani, "Adaptive offspring generation for evolutionary large-scale multiobjective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 786–798, Feb. 2022.

[60] A. Panichella, "An improved Pareto front modeling algorithm for large-scale many-objective optimization," in *Proc. Genet. Evol. Comput. Conf.*, 2022, pp. 565–573.

[61] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4108–4121, Dec. 2017.

[62] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2005.

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

[64] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, "SMPSO: A new PSO-based metaheuristic for multi-objective optimization," in *Proc. IEEE Symp. Comput. Intell. Multi-Criteria Decis.-Making (MCDM)*, 2009, pp. 66–73.

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

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

[67] M. P. Fay and M. A. Proschan, "Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules," *Stat. Surveys*, vol. 4, p. 1, Apr. 2010.

[68] C. He, R. Cheng, C. Zhang, Y. Tian, Q. Chen, and X. Yao, "Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 868–881, Oct. 2020.

Xiangzhou Gao received the M.S. degree in control science and engineering from the School of Information and Control Engineering, Liaoning Petrochemical University, Fushun, China, in 2020. He is currently pursuing the Ph.D. degree in control science and engineering with the Center for Control Theory and Guidance Technology Research, School of Astronautics, Harbin Institute of Technology, Harbin, China.

His research interests include military intelligence, evolutionary computation, and their applications.

Shenmin Song received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology (HIT), Harbin, China, in 1990, 1993, and 1996, respectively.

From 2000 to 2002, he was a Postdoctoral Researcher with the Faculty of Engineering, The University of Tokyo, Tokyo, Japan. He is currently a Professor with the Center for Control Theory and Guidance Technology Research, School of Astronautics, HIT. He has published over 130 academic papers in national and international journals and conference proceedings, and two academic monographs. His main research interests include guidance, navigation, and control, system effective evaluation, and evolutionary optimization.

Hu Zhang received the M.Sc. degree in mechanical design and theory from China Three Gorges University, Yichang, Hubei, China, in 2012, and the Ph.D. degree from the Center for Control Theory and Guidance Technology, Harbin Institute of Technology, Harbin, China.

He is currently a Senior Engineer with Beijing Electro-Mechanical Engineering Institute, Beijing, China. His current research interests include evolutionary computation, military intelligence, optimal design of system, and operation assessment.

Zhenkun Wang (Member, IEEE) received the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2016.

From 2017 to 2020, he was a Postdoctoral Research Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, and with the Department of Computer Science, City University of Hong Kong, Hong Kong. He is currently an Assistant Professor with the School of System Design and Intelligent Manufacturing and the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include evolutionary computation, optimization, machine learning, and their applications.

Dr. Wang is an Associate Editor of the *Swarm and Evolutionary Computation*.