A Large-Scale Multiobjective Particle Swarm Optimizer With Enhanced Balance of Convergence and Diversity

Dongyang Li, *Student Member, IEEE*, Lei Wang, *Member, IEEE*, Li Li, *Member, IEEE*, Weian Guo, *Member, IEEE*, Qidi Wu, *Senior Member, IEEE*, and Alexander Lerch, *Member, IEEE*

***Abstract*—Large-scale multiobjective optimization problems (LSMOPs) continue to be challenging for existing multiobjective evolutionary algorithms (MOEAs). The main difficulties are that: 1) the diversity preservation in both the objective space and the decision space needs to be taken into account when solving LSMOPs and 2) the existing learning structures in current MOEAs usually make the learning operators only coincidentally serve convergence and diversity, leading to difficulties in balancing these two factors. Therefore, balancing convergence and diversity in current MOEAs is difficult. To address these issues, this article proposes a multiobjective particle swarm optimizer with enhanced balance of convergence and diversity (MPSO-EBCD). In MPSO-EBCD, a novel velocity update structure for multiobjective particle swarm optimization is put forward, dividing the convergence, and diversity preservation operations into independent components. Following the proposed update structure, a weighted convergence factor is introduced to serve the convergence strategy, whilst a diversity preservation strategy is built to uniformly distribute the particles in the searched space based on a proposed multidimensional local sparseness degree indicator. By this means, MPSO-EBCD is able to balance convergence and diversity with specific parameters in independent operators. Experimental results on LSMOP benchmarks and a voltage transformer optimization problem demonstrate the competitiveness of the proposed algorithm compared to several state-of-the-art MOEAs.**

***Index Terms*—Convergence, diversity, large-scale multiobjective optimization, multidimensional local sparseness, particle swarm optimization (PSO), weighted convergence factor (WCF).**

I. Introduction

LARGE-SCALE multiobjective optimization problems (LSMOPs), very common in real-world applications [1], involve at least two or three conflicting optimization objectives and more than 100 decision variables [2]. These problems create challenges in both the objective space and the decision space. On the one hand, a set of well-converged and uniformly distributed solutions in the objective space rather than a single one has to be obtained [3], [4]. On the other hand, the increasing dimensionality sharply expands the search space around local optimal solutions and results in complex relationships between decision variables [3], [5]. As a result, LSMOPs bring challenges to multiobjective evolutionary algorithms (MOEAs) with respect to balancing convergence and diversity in both objective space and decision space. In response to this problem, considerable effort has been put forward to either alleviate the difficulties of the balance between convergence and diversity for MOEAs or to improve the algorithms' ability in balancing these two factors.

Current methods can be grouped into two broad categories: 1) solving LSMOPs in a low-dimensional decision space [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] and 2) improving the convergence and diversity preservation ability of MOEAs [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39]. Methods in the first category commonly adopt divide-and-conquer strategies and problem transformation techniques, which aim at dividing the LSMOP into a set of subproblems or transforming the LSMOP into a lower dimensional problem, respectively. These include random grouping strategies [8], analysis-based grouping schemes [9], weighted optimization frameworks [16], and decomposition-based strategies [21]. However, due to the considerable complexity of the optimization problems, the constructed subproblems potentially remain LSMOPs thus still posing challenges to the optimizers with respect to balancing convergence and diversity. Approaches in the second category focus on optimizing all decision variables by designing strategies to enhance the convergence or diversity for MOEAs,

Manuscript received 15 June 2022; revised 30 August 2022, 31 October 2022, and 13 November 2022; accepted 25 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62273263, Grant 72171172, and Grant 71771176; in part by the Natural Science Foundation of Shanghai, China, under Grant 19ZR1479000 and Grant 20692191200; in part by the Shanghai Municipal Science and Technology Major Project under Grant 2022-5-YB-09 and Grant 2021SHZDZX0100; in part by the Fundamental Research Funds for the Central Universities; and in part by the Education Research and Reform Project of Tongji University: Research on Talent Training Program and Model of Sino-German International Factory. This article was recommended by Associate Editor Y. S. Y. S. Ong. *(Corresponding author: Weian Guo.)*

Dongyang Li and Weian Guo are with the Sino-German College of Applied Sciences, Tongji University, Shanghai 200092, China (e-mail: lidongyang0412@163.com; guoweian@163.com).

Lei Wang, Li Li, and Qidi Wu are with the Department of Electronics and Information Engineering, Tongji University, Shanghai 201804, China (e-mail: wanglei@tongji.edu.cn; lili@tongji.edu.cn; qidi@tongji.edu.cn).

Alexander Lerch is with the Music Informatics Group, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: alexander.lerch@gatech.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2022.3225341.

Digital Object Identifier 10.1109/TCYB.2022.3225341

such as the reference vector-based diversity enhancement [24], the machine-learning-based convergence strategy [38], and the indicator-based learning strategy [31]. Nevertheless, the existing learning structures of current MOEAs fail at clarifying the targets of the learning strategies in serving convergence and diversity. Consequently, the learning strategies are either ambiguous in function or asynchronous in implementation, leading to difficulties for MOEAs in balancing convergence and diversity.

To summarize, balancing convergence and diversity for MOEAs in LSMOPs remains an unsolved challenge since the aforementioned methods cannot explicitly, independently, and simultaneously balance these two factors. To address this problem, this article proposes two indicators and puts forward a multiobjective particle swarm optimizer with enhanced balance of convergence and diversity (MPSO-EBCD), aiming at explicitly, independently, and simultaneously conducting convergence and adjusting diversity. The main contributions of this article are listed as follows.

1) A novel velocity update structure is proposed for multiobjective particle swarm optimization (PSO) to solve LSMOPs, formulating the convergence and diversity preservation strategies into two different operators. In this way, an explicit, independent, and simultaneous balance of convergence and diversity can be achieved by adjusting the parameters in the corresponding operators. To our best knowledge, this is the first attempt to decouple the learning strategies of convergence and diversity preservation in the context of the velocity update for multiobjective PSO to solve LSMOPs.
2) A novel weighted convergence factor (WCF) is proposed to measure the convergence quality of solutions, based on which a velocity update strategy for convergence (VUSC) is introduced; furthermore, a multidimensional local sparseness degree (MLSD) indicator is proposed to detect the local diversity information for particles. Finally, a velocity update strategy for diversity preservation is designed based on MLSD, resulting in MPSO-EBCD.

The remainder of this article is organized as follows. Section II presents a comprehensive review on the current mainstream of MOEAs in LSMOPs. Section III introduces the details of the proposed indicators and learning strategies. Section IV experimentally validates the performance of the proposed algorithm. Finally, we conclude this article and provide the future work directions in Section V.

II. Related Work

A. Selected Current Work

The divide-and-conquer technique is widely adopted to divide LSMOPs into a set of lower-dimensional problems to be simultaneously optimized by specific optimizers [6], [7]. By this means, the original LSMOP can be solved in multiple smaller decision spaces. One of the first such approaches was proposed by Antonio and Coello [8], who randomly divide the entire decision vector into several subsegments. However, the random variable grouping strategy causes adverse impacts on the convergence of MOEAs if there exist interactions between variables. Targeting this issue, different analysis-based variable grouping methods have been proposed. On the one hand, researchers classify the variables into different groups according to the interactions between variables. Random-based dynamic grouping strategies, for instance, dynamically select both different subgroup size settings in the optimization process [10], [12] and the differential grouping strategy [13]. On the other hand, classifying the variables according to their properties is also an intensely researched topic, as evidenced by decision variable analyses [9], [22], [23] and decision variable clustering [11]. These methods distinguish solutions for convergence-related variables and diversity-related variables so that different sets of variables can be optimized with different strategies for convergence and diversity, respectively.

Problem transformation, aiming to directly simplify the LSMOP by designing transformation functions to reduce the problems' dimensionality, has attracted considerable attention. For example, Schütze et al. [14] proposed to generate a collection of subdecision spaces to cover the entire set of Pareto points. Zille et al. [16], [17] developed a weighted optimization framework (WOF) which adopts variable grouping and weighting to transform the original optimization problems into low-dimensional problems. Taken both the decision space and objective space into consideration, He et al. [18] proposed a large-scale multiobjective optimization framework (LSMOF) by reformulating the problem, reconstructing the decision space with the weight variables and reducing the complexity of the objective space based on an indicator function. However, such methods usually require users to predefine parameters for the problem transformation. Addressing this issue, machine learning is becoming an attractive tool in problem transformation due to its powerful learning ability. Examples for machine-learning approaches are random embedding-based intrinsic dimensionality detection [15], neural-network-based dimensionality compacting [19], and pattern mining-based dimensionality reduction [20].

In recent years, significant effort has been made toward designing various strategies to improve MOEAs' ability to balance convergence and diversity [31]. First, reference vector and decomposition-based strategies are usually adopted to balance convergence and diversity in the objective space by properly selecting offspring and optimization directions [25], [27]. Second, various exemplar selection schemes and update strategies have been proposed to enhance the convergence and diversity preservation abilities based on different indicators [2], [25], [26], [28], [29], [30], [31], [32], [36], [37], [40]. Among these methods, the multiobjective particle swarm optimizer with multiple search strategies (MMOPSOs) [25], the novel multiobjective PSO with balanceable fitness estimation (NMPSO) [26] and the large-scale multiobjective optimizer based on competitive swarm optimization (LMOCSO) [31] are three representative approaches. MMOPSO adopts different exemplar selection strategies to help balance convergence and diversity; NMPSO introduces a composite solution evaluation indicator which uses two parameters to help adjusting the importance of the convergence and diversity information when evaluating particles; LMOCSO designs a dual-step solution

update strategy to improve global search efficiency. Third, machine-learning techniques are adopted to assist the MOEAs with respect to convergence and diversity [24], [33], [34], [35], [38]. The inverse modeling-based MOEA (IMMOEA), for instance, adopts a Gaussian model to generate offspring to enhance the diversity preservation ability of the algorithm [24], the generative adversarial network-based large-scale multiobjective evolutionary framework (GAN-LMEF) adopts a strategy based on generative adversarial networks to conduct manifold interpolation for diversity preservation in the early optimization stage [35], and the accelerated evolutionary search (AES) trains a neural network to learn gradient-descent-like vectors to help update solutions for convergence [38].

B. Discussion

Although the prementioned methods have shown good performance across a wide range of benchmarks and applications, the following problems remain unresolved.

First, the methods with the divide-and-conquer technique face the issues of high computational cost and inaccurate variable grouping. Furthermore, the subproblems can potentially also be LSMOPs if there exists a large set of variables that strongly interacts with each other.

Second, for the methods with problem transformation, due to the high complexity of the optimization problems, a large decision space is still potentially required to construct the subproblems such that the original Pareto front can be accurately represented. Therefore, such methods still cannot effectively address the problem of balancing convergence and diversity of MOEAs for LSMOPs.

Third, the existing learning structures cannot provide separate guidance to design convergence operator and diversity preservation operator, respectively. For instance, the existing methods commonly evaluate solutions with a compromise between convergence and diversity, such as the shift-based density estimation (SDE) [31] and the balanceable fitness estimation (BFE) [26], resulting in ambiguous functions of the learning strategies and thus making it difficult for these methods to control convergence and diversity with parameters. Furthermore, the specific operations will cause extra parameter tuning tasks and lead to difficulties in balancing convergence and diversity [41].

In summary, balancing of convergence and diversity in current MOEAs remains a challenge. More specifically, the aforementioned methods fail at explicitly, independently, and simultaneously balancing these two factors. To solve this issue, this article proposes to design a solution update strategy which is able to explicitly, independently, and simultaneously conduct the convergence and adjust diversity.

III. Proposed Method

A. Proposed Velocity Update Structure

This article suggests a novel velocity update structure for multiobjective PSO, which divides the convergence and diversity preservation strategies into different operators. It enables decoupled control of convergence and diversity by providing separate adjustments with the corresponding operators. The proposed velocity update structure is

$$v_i^d(t+1) = \mathcal{I}\left(v_i^d(t)\right) + \mathcal{C}\left(p_i^d(t)\right) + \mathcal{D}\left(p_i^d(t)\right) \tag{1}$$

where $v^d(t)$ represents the dth dimension of the ith particle's velocity; $\mathcal{I}(\cdot)$ is the inertia operator which is the same with that of the basic PSO; $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$ denote the operators focusing on guiding particles to converge toward the Pareto front and enhance diversity, respectively.

Consequently, the parameters in $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$ can be used to adjust the convergence and diversity preservation abilities, respectively.

B. Convergence Strategy

Convergence, which aims at guiding particles to find a set of well-converged solutions, highly depends on the exemplar selection strategy. In order to explicitly evaluate the convergence property for solutions and construct the convergence learning strategy, this article proposes a WCF to qualify particles. Afterwards, a VUSC is designed based on *WCF*. The details are presented in the following.

1) Weighted Convergence Factor: To ensure that the designed convergence measurement is able to evaluate the quality of solutions and be inspired by the relative nondominance matrix *RNM* [4], *WCF* is proposed as follows:

$$WCF(i) = \sum_{p_j \in P/p_i} C(p_i, p_j) D(p_i, p_j) \tag{2}$$

$$D(p_i, p_j) = \sum_{m=1}^{M} \max\left(0, f_m(p_i) - f_m(p_j)\right) \tag{3}$$

$$C(p_i, p_j) = c/M \tag{4}$$

where p_i and p_j denote the ith particle and the jth particle, respectively; $f_m(\cdot)$ denotes the fitness of the corresponding particle on the mth objective; M is number of the objectives; c is the number of times that p_i performs worse than p_j out of the M objectives.

Specifically, $D(p_i, p_j)$ is the first order relative nondominance distance from p_i to p_j. Fig. 1 illustrates a bi-objective case to briefly show $D(p_i, p_j)$. Without loss of generality, the optimization problem is regarded as a minimization problem. As shown in Fig. 1, p_i and p_j are nondominated with each other. In spite of this, by moving p_i along $L_1 = 2$ (i.e., $D(p_i, p_j) = 2$) the solution C, which dominates p_j, can be reached. Similarly, p_j will dominate p_i if it moves along $L_2 = 3.4$ (i.e., $D(p_j, p_i) = 3.4$). It is obvious that p_i is expected to reach solution C faster than p_j in terms of the moving distance, indicating p_i is relatively better than p_j as reflected by the corresponding $D(\cdot)$ although they are nondominated with each other.

$C(p_i, p_j)$ is designed to weight the relative nondominance distance based on the following reasoning. To solve problems with more than two objectives, three potential cases require special consideration: 1) $D(p_i, p_j)$ is equal to $D(p_j, p_i)$; 2) $D(p_i, p_j)$ is larger than $D(p_j, p_i)$ but p_i outperforms p_j on more objectives; and 3) the scalar of some objective is larger than that of others. The first case can happen, for instance,

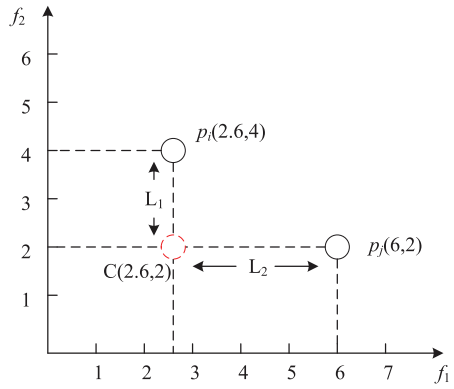

Fig. 1. Illustration of the computation of the first order relative nondominance distance.

if the two particles are $p_i(0, 0, 1)$ and $p_j(0.5, 0.5, 0)$, resulting in both $D(p_i, p_j)$ and $D(p_j, p_i)$ equaling 1. An example for the second case is $p_i(0, 0, 1.1)$ and $p_j(0.5, 0.5, 0)$, leading to $D(p_i, p_j)$ and $D(p_j, p_i)$ being 1.1 and 1, respectively. However, p_i outperforms p_j on two objectives and is only slightly worse than p_j with respect to $D(\cdot)$. If the two particles are $p_i(0, 0, 8)$ and $p_j(0.6, 0.6, 6)$, for example, the landscape of the third objective is [0, 10] and the landscape of the first two objectives is [0, 1]. This is the third case, and similar to the second case, although $D(p_i, p_j)$ is larger than $D(p_j, p_i)$, it is potentially more suitable to guide particles with p_i since p_i outperforms p_j on two objectives out of three.

In all three cases, $D(\cdot)$ can either not distinguish between the quality of p_i and p_j (first case), or estimates p_i to be worse than p_j (cases two and three). The definition of $C(\cdot)$ aims at alleviating this: in the first case, $C(p_i, p_j)D(p_i, p_j)$ and $C(p_j, p_i)D(p_j, p_i)$ are 1/3 and 2/3, respectively, the second case results in $C(p_i, p_j)D(p_i, p_j) = 11/30$ and $C(p_j, p_i)D(p_j, p_i) = 2/3$, and $C(p_i, p_j)D(p_i, p_j) = 2/3$ and $C(p_j, p_i)D(p_j, p_i) = 4/5$ for the third case. As can be seen, p_i is better than p_j in all three cases. This is a desirable outcome because p_i is better than p_j on two objectives and has the same relative nondominance distance with p_j in the first case, indicating that learning from p_i can potentially benefit to improve more objectives for other particles; in the last two cases, p_i outperforms p_j on more objectives and is acceptably worse than p_j in terms of the relative nondominance distance. Therefore, it should be also helpful in improving more objectives for a particle by guiding it with p_i.

Finally, *WCF* is introduced to measure the convergence quality of each particle by accumulating the weighted nondominance distances between a particle and the remaining individuals. Note that, unlike *WCF*, the *RNM* [4], which builds on the second-order relative nondominance distance, only solves the first challenge mentioned above but cannot address the second and third cases as the square operation fails to account for the characteristics among the particles.

2) Velocity Update Strategy for Convergence: To provide diverse directions for particles to explore the Pareto front, the VUSC is proposed as follows. First, the *WCF* is computed for the swarm and used to qualify the particles; second, the competitive mechanism [42] is adopted to build a particle identification strategy (PIS) which divides the entire swarm into the loser group (referred to as P_{updated}, denoting the particles to be updated) and the winner group (referred to as P_{con}, denoting the corresponding convergence exemplars). The pseudo-code of the PIS is shown in Algorithm 1. Finally, the losers update their velocity by learning from the corresponding winners stored in P_{con} for convergence according to

$$v_i^d(t+1) = \psi r\Big(p_{i,c}^d(t) - p_i^d(t)\Big) \tag{5}$$

where t is the generation number; v_i^d and p_i^d is the dth dimension of the velocity and position of the ith particle in P_{updated}; $p_{i,c}^d$ is the dth dimension of the position of the corresponding convergence exemplar for the ith particle in P_{con}; and r is a random number generated within (0, 1). ψ is defined to control the convergence. Such design allows the particles to learn from the superior individuals for convergence and provides diverse convergence directions to explore the entire Pareto front.

Algorithm 1 Particles Identification Strategy

Input: Swarm $P(t)$, swarm size N;
Output: The particles should be updated: $P_{updated}$ at tth generation and the corresponding convergence exemplars P_{con};

```
Compute the WCF according to Eqs. (2) to (4);
Equally Divide P(t) into N/2 sub-groups;
P_updated, P_con ← ∅;
for i = 1 to N/2 do
    In the ith sub-group;
    if the WCF(p_{i,1}) <= WCF(p_{i,2}) then
        P_updated ← P_updated ⋃ p_{i,2};
        P_con ← P_con ⋃ p_{i,1};
    else
        P_updated ← P_updated ⋃ p_{i,1};
        P_con ← P_con ⋃ p_{i,2};
    end if
end for
return P_updated, P_con;
```

C. Diversity Preservation Strategy

Diversity preservation is also crucial to MOEAs in solving LSMOPs. Normally, it is desired that the decision variables can maintain a good diversity and the obtained solutions can uniformly spread over the Pareto front [31], [43]. Motivated by the fact that the phenotype diversity is capable of reflecting the diversity in both objective space and decision space [5], [44], this article proposes an MLSD indicator to measure the local diversity information for particles based on the phenotype diversity. Afterwards, a velocity update strategy for diversity preservation (VUSD) is proposed based on MLSD for diversity preservation.

1) Multidimensional Local Sparseness Degree: Fig. 2 shows two potential swarms in a bi-objective optimization case. Obviously, the diversity of Swarm 2 is worse than that of Swarm 1. Particles in Swarm 2 need to move toward the space between p_3 and p_4 to increase swarm diversity. Consequently, an indicator is needed to help measuring the local diversity information and determining the particles that can guide others for a more uniform swarm distribution. To this end, we

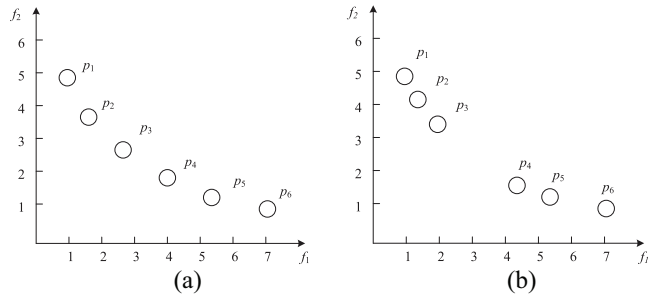

Fig. 2. Illustration of a potential swarm distribution. (a) Swarm 1. (b) Swarm 2.

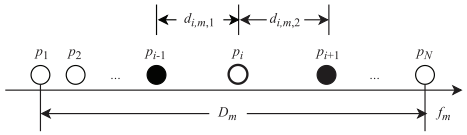

Fig. 3. Illustration of the computation of the mth component of MLSD.

propose the MLSD, which is formulated as follows:

$$\mathrm{MLSD}(i) = \sum_{m=1}^{M} \frac{\mathrm{dis}_m(i)}{\max(\mathrm{dis}_m)} \frac{\mathrm{con}_m(i)}{\max(\mathrm{con}_m)} \tag{6}$$

$$\mathrm{dis}_m(i) = \frac{\min\left(d_{i,m,1}, d_{i,m,2}\right)}{\max\left(d_{i,m,1}, d_{i,m,2}\right)} \tag{7}$$

$$\mathrm{con}_m(i) = \left(d_{i,m,1} + d_{i,m,2}\right)/D_m \tag{8}$$

where $\mathrm{MLSD}(i)$ is the proposed MLSD of the ith particle in fitness landscape; M denotes the number of objectives; and $\mathrm{con}_m(i)$ and $\mathrm{dis}_m(i)$ indicate the congestion and distribution of the ith particle in the mth objective space, respectively. Specifically, in the mth objective space, the swarm is first sorted according to the fitness in the mth objective space as shown in Fig. 3. Here, the two particles which are nearest at the two sides of the ith particle are defined as its neighbors, that is, p_{i-1} and p_{i+1} marked in black in Fig. 3; $d_{i,m,1} = |f_m(p_{i-1}) - f_m(p_i)|$; $d_{i,m,2} = |f_m(p_i) - f_m(p_{i+1})|$; and $D_m = |f_m(p_1) - f_m(p_N)|$. Afterwards, $\mathrm{dis}_m(i)$ and $\mathrm{con}_m(i)$ can be obtained using (7) and (8). Finally, $\mathrm{MLSD}(i)$ is calculated by integrating $\mathrm{dis}_m(i)$ and $\mathrm{con}_m(i)$ in all objective spaces as shown in (6).

Obviously, $\mathrm{MLSD}(i)$ will increase with the growing size of the neighborhood spaces around the ith particle with the design of $\mathrm{con}_m(i)$. On the other hand, a more uniform distribution of the ith particle and its adjacent neighbors leads to a larger value of $\mathrm{MLSD}(i)$. Such design is motivated by the fact that particles should uniformly in the fitness landscape. This is beneficial to the case shown in Fig. 4, where the neighborhood spaces around p_i and p_j are identical with respect to both objectives. However, we expect particles moving toward p_i rather than moving toward p_j, since it is beneficial to obtain a more uniform distribution of the swarm. Note that the $\mathrm{dis}_m(i)$ and $\mathrm{con}_m(i)$ are set to 0 if the ith particle is the best or the worst individual on the mth objective. The reasons for this are that: 1) the $\mathrm{dis}_m(i)$ and $\mathrm{con}_m(i)$ cannot be computed with only one neighbor; 2) learning from them risks worse performance on the corresponding objective if they are the worst on the mth objective; and 3) these particles should potentially be selected as the convergence exemplar if they perform quite well on other objectives.

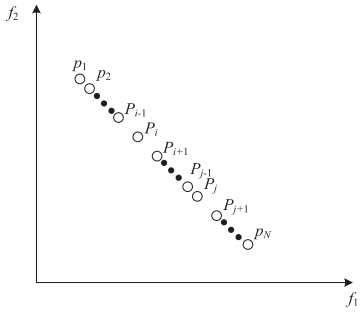

Fig. 4. Potential fitness landscape.

Algorithm 2 Diversity Exemplars Selection Strategy

Input: Swarm $P(t)$, the fitness vector WCF, the particles should be updated $P_{updated}$;
Output: The particles should be updated: $P_{updated}$ at tth generation and the corresponding diversity exemplars P_{div};

```
P_div ← ∅;
Sort the swarm according to WCF in ascending order;
Compute the MLSD according to Eqs. (6) to (8)
for p_i in P_updated do
    P_d ← Particles with larger MLSD than p_i in P(t);
    p_k ← A particle randomly selected from P_d;
    P_div ← P_div ⋃ p_k;
end for
return P_div;
```

2) Velocity Update Strategy for Diversity Preservation: The VUSD, based on MLSD, is proposed for diversity preservation. The main idea is to guide particles moving to the sparse areas of the searched spaces from the perspective of fitness landscape. The proposed velocity update operator can be formulated as follows:

$$v_i^d(t+1) = \phi r\left(p_j^d(t) - p_i^d(t)\right) \tag{9}$$

where v_i^d and p_i^d are the dth dimension of the velocity and position of the ith particle; p_j^d is the dth dimension of the position of a randomly selected particle, where $\mathrm{MLSD}(j)$ is larger than $\mathrm{MLSD}(i)$; ϕ is a parameter set by users for adjusting the diversity. In summary, VUSD is capable of reaching the following two goals: 1) an approximately uniform distribution of the particles in the fitness landscape as the phenotype diversity is adopted to compute MLSD, which benefits the diversity preservation of the searched nondominated solutions [31], [44] and 2) diversity preservation in the decision space through the widely-used phenotype diversity [5]. As a result, the proposed learning strategy is also beneficial to the diversity preservation in the decision space.

Similar to the proposed PIS in Section III-B2, a diversity exemplars selection strategy is designed as shown in Algorithm 2.

D. MPSO-EBCD

By embedding the proposed VUSC and VUSD into the proposed velocity update structure as stated in (1), a novel

learning strategy for particles to be updated is put forward as follows:

$$\begin{aligned} v_i^d(t+1) &= \mathcal{I}\left(v_i^d(t)\right) + \mathcal{C}\left(p_i^d(t)\right) + \mathcal{D}\left(p_i^d(t)\right) \\ &= \mathcal{I}\left(v_i^d(t)\right) + \mathrm{VUSC}\left(p_i^d(t)\right) + \mathrm{VUSD}\left(p_i^d(t)\right) \\ &= r_1 v_i^d(t) + \psi r_2\left(p_1^d(t) - p_i(t)\right) \\ &\quad + \phi r_3\left(p_2^d(t) - p_i^d(t)\right) \end{aligned} \tag{10}$$

$$\begin{aligned} p_i^d(t+1) &= p_i^d(t) + v_i^d(t+1) \\ &\quad + r_1\left(v_i^d(t+1) - v_i^d(t)\right) \end{aligned} \tag{11}$$

where the three parts of the right side in (10) are the inertia component for ensuring the stability of the proposed algorithm [42], the components for convergence and diversity, respectively; p_i^d is the dth dimension of the position of the ith particle; p_1^d and p_2^d are the dth dimension of the position of the convergence and diversity exemplars for the ith particle, which are obtained by Algorithms 1 and 2, respectively; r_1, r_2, and r_3 are three random numbers generated within (0, 1); v_i^d is the dth dimension of the velocity of the ith particle and t is the generation number; ψ and ϕ are the parameters set by users to control convergence and diversity, respectively; ψ is set to 1 in this article to reduce the parameter tuning tasks and ϕ can be adjusted to balance convergence and diversity. Note that the position's update strategy proposed in [31] is adopted here due to its effectiveness in improving update efficiency. Thanks to the characteristics of the proposed indicators, VUSC and VUSD are designed for convergence and diversity preservation, respectively. As a result, ψ and ϕ have the explicit meaning of adjusting the convergence and diversity preservation ability of MPSO-EBCD. Consequently, MPSO-EBCD is able to explicitly and independently adjust convergence and diversity at the level of the velocity update, which will be experimentally validated in Section IV-C. Furthermore, keeping one parameter constant, the other one can be adjusted to balance convergence and diversity, resulting in the explicit, independent, and simultaneous balance of convergence and diversity with the proposed velocity update strategy.

Note that, although the proposed VUSD is beneficial to the diversity preservation, it fails to extend the search space since it is mainly used to uniformly distribute the swarm in the searched spaces. Therefore, polynomial mutation [31] is adopted to update the solutions in the winners group to help extending the search space, which can be shown with (14)–(16), in the supplementary material, (see Appendix).

Finally, the reference vector-based selection strategy proposed in [45] is employed to select $P(t+1)$ from $P(t)$ and the updated particles at each generation, leading to MPSO-EBCD. Here, $P(t)$ is the swarm at generation t and the method proposed in [46] is adopted to generate the uniformly distributed reference vectors. Thanks to the proposed learning strategies, MPSO-EBCD is able to balance convergence and diversity by only adjusting ϕ: the particles will put more effort on convergence with a small ϕ, while increasing ϕ puts more attention on diversity preservation. The pseudo-code of MPSO-EBCD is shown in Algorithm 3.[1]

Algorithm 3 MPSO-EBCD

Input: Swarm size N, parameter ϕ;
Output: P: the obtained non-dominated solutions;

```
Initialize a swarm P(t);
t = 1;
while terminal condition is not met do
    P_offspring ← ∅;
    P_updated, P_con, WCF ← Run Algorithm. 1;
    P_div ← Run Algorithm. 2;
    for p_i in P_updated do
        p_child,1 ← Executing Eqs. (10) and (11)
        p_child,2 ← Executing mutation on P_con
        P_offspring ← P_offspring ∪ {p_child,1, p_child,2}
    end for
    P(t) ← P(t) ∪ P_offspring
    P(t) ← Execute the environment selection based on P(t)
    t = t + 1
end while
return P(t);
```

E. Computational Complexity

Given that an algorithm is applied to solve a problem with M objectives and D decision variables, both the computational cost per generation and the memory cost are taken into consideration to evaluate the algorithm's complexity. We compare the presented approach to LMOCSO with a swarm size of N.

First, MPSO-EBCD takes $O(3MN^2)$ to compute WCF while LMOCSO costs $O((3M+1)N^2)$ to compute the fitness. Second, MPSO-EBCD inherits the structure of basic PSO, consisting of three components while LMOCSO only includes two components. Therefore, MPSO-EBCD costs $O(3N/2)$ to update particles' velocity, while LMOCSO costs $O(N)$. Third, MPSO-EBCD costs $O(N\log_2(N)+5MN+N/2)$ to compute the MLSD and select the diversity exemplars. Furthermore, MPSO-EBCD costs $O(N/2)$ on mutation while LMOCSO costs $O(N)$ for mutation. To summarize, the computational complexity of MPSO-EBCD and LMOCSO are $O(3MN^2+N\log_2(N)+5MN+5N/2)$ and $O((3M+1)N^2+2N)$, respectively.

Both MPSO-EBCD and LMOCSO do not need memory to store the personal best positions, since they adopt the current position to update particles. Therefore, the memory cost of MPSO-EBCD and LMOCSO should be comparable.

In short, it can be concluded that MPSO-EBCD theoretically has higher computational efficiency than LMOCSO.

IV. Experimental Study

To evaluate the performance of MPSO-EBCD, first, MPSO-EBCD is compared with several state-of-the-art MOEAs; second, experiments are conducted to test the influences of the proposed velocity update strategy on the diversity preservation in the decision space; third, a parameter sensitivity analysis is presented; forth, the scalability of MPSO-EBCD

[1]The source code of the proposed algorithm can be found at https://github.com/YiBenjing456/MPSOEBCD.

TABLE I
IGD VALUES OF LMEA, LMOCSO, MMOPSO, WOF-NSGAII, IMMOEA, AND MPSO-EBCD ON BI-OBJECTIVE AND THREE-OBJECTIVE LSMOP_1–LSMOP_9 WITH 1000 DIMENSIONALITY, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS SHOWN IN A GRAY BACKGROUND

Problem	M	LMEA	LMOCSO	MMOPSO	WOF-NSGAII	IMMOEA	MPSO-EBCD
LSMOP1	2	3.7102e-2 (1.40e-4) -	1.2248e-3 (6.89e-7) -	1.7714e-2 (9.19e-5) -	2.8571e-1 (9.90e-3) -	1.8090e-2 (3.83e-4) -	1.1209e-3 (3.18e-7)
	3	6.4375e-2 (3.79e-3) -	1.9215e-2 (7.07e-4) -	2.4857e-1 (2.24e-2) -	1.6896e-1 (1.18e-2) -	1.7591e-1 (8.36e-3) -	1.8212e-2 (9.16e-4)
LSMOP2	2	2.0559e-2 (1.40e-5) -	1.3933e-2 (3.08e-5) -	2.2404e-2 (4.97e-4) -	9.3976e-3 (3.06e-4) +	6.1750e-3 (9.82e-5) +	1.2877e-2 (6.73e-5)
	3	2.6883e-2 (1.05e-4) -	1.9789e-2 (1.06e-4) -	3.8401e-2 (4.75e-4) -	4.1514e-2 (9.72e-4) -	3.7708e-2 (9.65e-4) -	1.9129e-2 (2.47e-4)
LSMOP3	2	2.8540e+0 (1.24e-1) -	7.0711e-1 (4.04e-8) =	1.5678e+0 (1.60e-2) -	7.6664e-1 (1.00e-1) =	5.0824e-1 (7.16e-4) +	7.0705e-1 (9.13e-8)
	3	1.1608e+0 (7.04e-2) -	4.7667e-1 (1.65e-1) -	5.8074e-1 (1.23e-2) -	4.5418e-1 (3.67e-2) -	6.6080e-1 (5.36e-2) -	3.8637e-1 (5.66e-2)
LSMOP4	2	2.2102e-2 (5.28e-4) -	1.3542e-3 (6.55e-6) -	2.7918e-2 (1.05e-3) -	2.4768e-2 (8.28e-5) -	7.0515e-3 (8.95e-5) -	1.2710e-3 (1.75e-5)
	3	3.5743e-2 (5.22e-4) -	2.7245e-2 (1.51e-3) -	1.0305e-1 (1.73e-3) -	1.0829e-1 (2.27e-3) -	6.4681e-2 (9.64e-4) -	2.0509e-2 (1.42e-3)
LSMOP5	2	4.1388e-1 (7.31e-3) -	1.8775e-1 (1.15e-5) -	6.2515e-3 (1.87e-3) -	3.9436e-2 (3.40e-3) -	1.9692e-2 (4.92e-3) -	1.4226e-3 (7.36e-5)
	3	4.5964e+0 (5.12e+0) -	2.2855e-2 (8.77e-5) -	2.3803e-1 (8.62e-3) -	3.8733e-1 (4.08e-2) -	2.0323e-1 (1.12e-2) -	2.1936e-2 (2.38e-5)
LSMOP6	2	6.2769e-1 (1.41e-1) -	5.0468e-1 (4.14e-2) =	1.8583e-1 (2.17e-2) +	2.2927e-1 (7.94e-2) +	2.0283e-1 (2.86e-2) +	5.6855e-1 (1.86e-1)
	3	1.5525e+0 (2.03e-1) -	8.6827e-1 (4.21e-1) +	6.6286e-1 (6.91e-2) +	1.2881e+0 (4.42e-4) =	1.2257e+0 (1.78e-1) +	1.2895e+0 (5.20e-1)
LSMOP7	2	1.1345e+0 (3.63e-1) =	1.2318e+0 (5.51e-1) -	1.5371e+0 (1.59e-1) =	9.0469e-1 (1.00e-1) =	1.3371e+0 (1.68e-1) =	1.1368e+0 (5.07e-1)
	3	6.9822e-1 (4.03e-2) +	9.3364e-1 (2.50e-2) +	7.6591e-1 (1.90e-1) +	8.4372e-1 (9.24e-4) +	5.8203e-1 (1.74e-2) +	9.4593e-1 (2.27e-7)
LSMOP8	2	4.1288e-2 (5.34e-4) -	3.3662e-3 (3.16e-4) -	2.3808e-2 (3.16e-5) -	3.7261e-2 (1.88e-3) -	2.6212e-2 (3.15e-3) -	2.4363e-3 (6.52e-4)
	3	9.1499e-2 (6.13e-3) -	3.0227e-2 (1.23e-3) +	6.5218e-2 (1.31e-3) -	1.2531e-1 (1.22e-2) -	9.5228e-2 (8.14e-3) -	3.1181e-2 (5.69e-4)
LSMOP9	2	4.8886e-1 (3.50e-3) -	2.3475e-3 (1.63e-6) -	9.3326e-2 (8.74e-4) -	8.0495e-1 (1.90e-3) -	1.3130e-2 (1.89e-3) -	2.1012e-3 (2.67e-7)
	3	4.2933e-1 (1.12e-2) -	2.5278e-1 (1.56e-1) -	6.1569e-1 (1.83e-3) -	1.1444e+0 (7.24e-4) -	5.3397e-1 (1.70e-2) -	2.0271e-1 (8.98e-2)
+/-/=		1/16/1	3/13/2	3/14/1	3/12/3	5/12/1	-

is tested; fifth, a run-time comparison is presented; and finally, the proposed algorithm is applied to solve a real-world optimization problem. All the experiments are run in PlatEMO [47] with MATLAB 2018a on a PC with Intel Core i7-9700k CPU and Microsoft Windows 10 Enterprise 64 bit. Each algorithm is run for 30 times on each benchmark.

A. Empirical Comparisons

1) Compared Algorithms and Benchmarks: The following algorithms are selected for the numerical comparison experiment as recommended in [31]: NSGAII with weighted optimization framework (WOF-NSGAII) [17], the decision variable clustering-based large-scale MOEA (LMEA) [11], the MMOPSO [25], the IMMOEA [24], and the large-scale multiobjective optimizer based on competitive swarm optimizer (LMOCSO) [31].

All the experiments are conducted based on the benchmark test suite proposed in [48]. Of the nine LSMOPs in this suite, LSMOP_1 to LSMOP_4 include the linear variable linkage while LSMOP_5 to LSMOP_9 involve nonlinear variable linkage on the Pareto set, respectively. A linear Pareto front, a concave Pareto front, and a disconnected Pareto front can be found in LSMOP_1 to LSMOP_4, LSMOP_5 to LSMOP_8, and LSMOP_9, respectively. In our experiments, the dimensionality of LSMOPs is set to 1000 and the number of objectives is set to 2 and 3 for each benchmark. To illustrate the algorithm characteristics with respect to diversity and convergence, LSMOP_1, LSMOP_5, and LSMOP_9 are selected with a dimensionality of 1000.

2) Experimental Settings: As recommended by [31], all the algorithms adopt the same population size of 300 and 496 for bi-objective benchmarks and three-objective benchmarks, respectively; the maximum number of fitness evaluations ($15000D$) is adopted as the termination criterion. For the parameter settings, ϕ in MPSO-EBCD is set to 0.5 according to a parameter sensitivity analysis; as recommended by [31], the parameter settings for the remaining algorithms are as listed in the following. In LMOCSO, the penalty parameter α in APD is set to 2. In WOF-NSGA-II, the number of evaluations for each optimization of original problem t_1 and the number of evaluations for each optimization of transformed problem t_2 are both set to 1000; the number of chosen solutions q, the number of groups γ and the ratio of evaluations for optimization of both original and transformed problems δ are set to 2, 3, and 0.7, respectively. In LMEA, the number of selected solutions for decision variable clustering and the number of perturbations on each solution for decision variable clustering are both set to 5; the number of selected solutions for decision variable interaction analysis is set to 6. In IMMOEA, the number of reference vectors and the model group size are set to 10 and 3, respectively. Furthermore, WOF-NSGA-II, LMEA, and MMOPSO adopt the simulated binary crossover and polynomial mutation, where the crossover probability is set to 1, the mutation probability is set to $1/D$ (D is the dimensionality of the benchmarks) and the distribution index of them is set to 20.

The performance of the MOEAs is evaluated with the inverted generational distance (IGD) and hypervolume (HV), where the IGD is calculated with roughly 10 000 reference points sampled on the Pareto front of benchmarks and HV is computed with the method proposed in [31].

3) Results: In Tables I and II, the mean and standard deviation of the corresponding results are reported. The Wilcoxon rank sum test with a significance level of 0.05 is adopted for the statistical analysis between the peer algorithms and MPSO-EBCD. The symbols "+," "−," and "=" indicate that the results of the peer algorithms are significantly better, significantly worse, and statistically similar to the results obtained by MPSO-EBCD, respectively. The best results of the average performance are highlighted with a gray background.

The IGD results in Table I clearly show that the proposed MPSO-EBCD performs overall better than the five peer MOEAs. Specifically, MPSO-EBCD outperforms its peers 11 times out of the 18 benchmark LSMOPs. Statistical analysis shows that the ratios by which MPSO-EBCD significantly outperforms LMEA, LMOCSO, MMOPSO, WOF-NSGAII, and IMMOEA are 16/18, 13/18, 14/18, 12/18, and 12/18, respectively. Furthermore, MPSO-EBCD exhibits competitive performance on all kinds problems: for LSMOP_1–LSMOP_4

TABLE II
HV VALUES OF LMEA, LMOCSO, MMOPSO, WOF-NSGAII, IMMOEA, AND MPSO-EBCD ON BI-OBJECTIVE AND THREE-OBJECTIVE $LSMOP_1$–$LSMOP_9$ WITH 1000 DIMENSIONALITY, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS SHOWN IN A GRAY BACKGROUND

Problem	M	LMEA	LMOCSO	MMOPSO	WOF-NSGAII	IMMOEA	MPSO-EBCD
LSMOP1	2	5.3230e-1 (7.96e-5) -	5.8515e-1 (2.46e-6) -	5.6441e-1 (1.17e-4) -	2.4344e-1 (1.12e-2) -	5.6224e-1 (4.22e-4) -	5.8630e-1 (1.33e-6)
	3	7.8007e-1 (8.31e-3) -	8.5621e-1 (8.21e-4) -	5.3287e-1 (3.27e-2) -	6.6677e-1 (1.78e-2) -	6.7800e-1 (1.22e-2) -	8.5776e-1(1.36e-3)
LSMOP2	2	5.5936e-1 (1.76e-5) -	5.6851e-1 (4.52e-5) -	5.5760e-1 (6.55e-4) -	5.7413e-1 (3.82e-4) +	5.7747e-1 (1.44e-4) +	5.6942e-1 (1.32e-4)
	3	8.4586e-1 (3.39e-4) -	8.5519e-1 (1.72e-4) -	8.3413e-1 (7.58e-4) -	8.3082e-1 (1.30e-3) -	8.3483e-1 (8.11e-4) -	8.5615e-1 (2.71e-4)
LSMOP3	2	0.0000e+0 (0.00e+0) -	9.0908e-2 (5.20e-8) -	0.0000e+0 (0.00e+0) -	4.5455e-2 (4.62e-2) =	9.1446e-2 (5.49e-5) =	9.1901e-2 (2.16e-7)
	3	0.0000e+0 (0.00e+0) -	1.8182e-2 (4.07e-2) -	3.1987e-1 (1.28e-2) =	3.6526e-1 (7.16e-2) =	9.1388e-2 (6.41e-2) -	3.9068e-1 (9.08e-2)
LSMOP4	2	5.5583e-1 (6.68e-4) -	5.8484e-1 (1.36e-5) -	5.5031e-1 (1.61e-3) -	5.5504e-1 (1.52e-6) -	5.7634e-1 (1.10e-4) -	5.9781e-1 (3.12e-5)
	3	8.3472e-1 (6.65e-4) -	8.4677e-1 (1.69e-3) -	7.6368e-1 (2.40e-3) -	7.6165e-1 (2.81e-3) -	8.0590e-1 (1.41e-3) -	8.5433e-1 (1.91e-3)
LSMOP5	2	7.9217e-2 (1.36e-3) -	3.2452e-1 (2.82e-4) -	3.4221e-1 (2.75e-3) -	2.9453e-1 (4.88e-3) -	3.2593e-1 (6.14e-3) -	3.5001e-1 (1.39e-4)
	3	9.6231e-2 (1.40e-1) -	5.8057e-1 (3.53e-4) -	4.0575e-1 (5.70e-4) -	3.5801e-1 (1.79e-2) -	3.6313e-1 (5.72e-2) -	5.8570e-1 (1.73e-4)
LSMOP6	2	4.6349e-2 (2.89e-3) -	9.0367e-2 (5.69e-5) =	1.2413e-1 (2.02e-2) +	1.5087e-1 (1.46e-2) +	1.3258e-1 (2.48e-3) +	7.9974e-2 (1.31e-3)
	3	0.0000e+0 (0.00e+0) =	7.0932e-2 (6.52e-2) +	8.7845e-3 (6.82e-3) =	0.0000e+0 (0.00e+0) =	0.0000e+0 (0.00e+0) =	1.5469e-2 (4.89e-2)
LSMOP7	2	0.0000e+0 (0.00e+0) -	3.6703e-2 (4.27e-2) =	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	2.7464e-3 (1.50e-2) -	4.1612e-2 (4.34e-2)
	3	0.0000e+0 (0.00e+0) -	9.0909e-2 (6.69e-8) =	1.0912e-2 (1.02e-2) -	1.4718e-2 (3.01e-3) -	9.0820e-2 (1.09e-4) -	9.0908e-2 (5.06e-7)
LSMOP8	2	2.9229e-1 (4.40e-4) -	3.4548e-1 (4.69e-4) -	3.1954e-1 (2.19e-5) -	2.9894e-1 (2.67e-3) -	3.2115e-1 (2.26e-3) -	3.5421e-1 (1.14e-3)
	3	4.6486e-1 (8.57e-3) -	5.6254e-1 (2.62e-3) =	5.0380e-1 (2.57e-3) -	4.5521e-1 (7.24e-3) -	4.8400e-1 (2.01e-2) -	5.6064e-1 (1.05e-3)
LSMOP9	2	1.6151e-1 (3.34e-4) -	2.4314e-1 (1.78e-6) -	1.9779e-1 (2.77e-4) -	9.1676e-2 (3.08e-4) -	2.3916e-1 (6.04e-4) -	2.6328e-1 (2.12e-6)
	3	2.0510e-1 (2.41e-3) -	2.0557e-1 (2.84e-2) -	1.4889e-1 (1.42e-3) -	1.4774e-1 (1.29e-5) -	1.9556e-1 (3.66e-3) -	2.6051e-1 (1.85e-2)
+/-/=		0/17/1	1/13/4	1/15/2	2/13/3	2/14/2	-

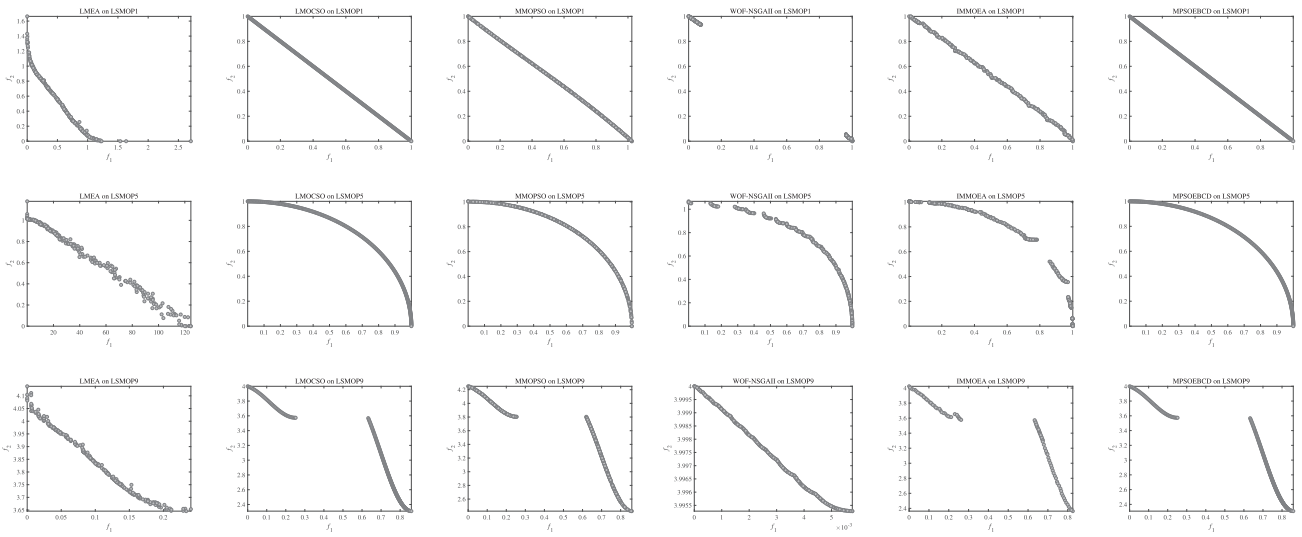

Fig. 5. Nondominated solutions with the median IGD values obtained by LMEA, LMOCSO, MMOPSO, WOF-NSGAII, IMMOEA, and MPSO-EBCD on bi-objective $LSMOP_1$, $LSMOP_5$, and $LSMOP_9$ with 1000 decision variables.

that include linear variable linkage, the proposed algorithm wins six times out of the eight instances; for the remaining five problems with nonlinear variable linkage, MPSO-EBCD wins for five times out of the ten instances; for the problems $LSMOP_5$–$LSMOP_8$ with concave Pareto front, the proposed algorithm wins for three times out of eight instances, followed by MMOPSO which wins on two instances; for $LSMOP_9$ which includes the disconnected Pareto front, the proposed algorithm wins on both the instances with two and three objectives.

The results of the six algorithms on HV, presented in Table II, show similar trends. The proposed algorithm exhibits competitive performance in comparison to the other five MOEAs, outperforming others for 13 times out of the 18 instances. Statistical analysis in Table II also shows the competitiveness of MPSO-EBCD: it significantly outperforms LMEA, LMOCSO, MMOPSO, WOF-NSGAII, and IMMOEA 17, 13, 15, 13, and 14 times out of the 18 instances, respectively. Furthermore, MPSO-EBCD shows competitive performance on all kinds of benchmarks, similar to the observations in Table I.

Figs. 5 and 6 show the nondominated solutions with the median IGD values obtained by different algorithms on benchmarks with two and three objectives, respectively. Figs. A.1 and A.2, in the supplementary material, depict the parallel coordinates of the nondominated fronts obtained by different algorithms. $LSMOP_1$, $LSMOP_5$, and $LSMOP_9$ are selected as the baseline test instances as they cover a wide variety of test suite characteristics, namely, linear variable linkage in $LSMOP_1$, nonlinear variable linkage and a concave Pareto front in $LSMOP_5$, and a disconnected Pareto front in $LSMOP_9$, respectively. The results show that LMEA and WOF-NSGAII cannot find well-converged solutions on all instances. Although IMMOEA is able to obtain a set of converged solutions on instances with two objectives, it completely fails on the instances with three objectives. Similar observations can be made for the results of MMOPSO. While the recently proposed LMOCSO performs well on the three benchmarks with two objectives and is capable of obtaining a set of converged solutions on $LSMOP_1$, $LSMOP_5$, and $LSMOP_9$ with three objectives, the diversity of the

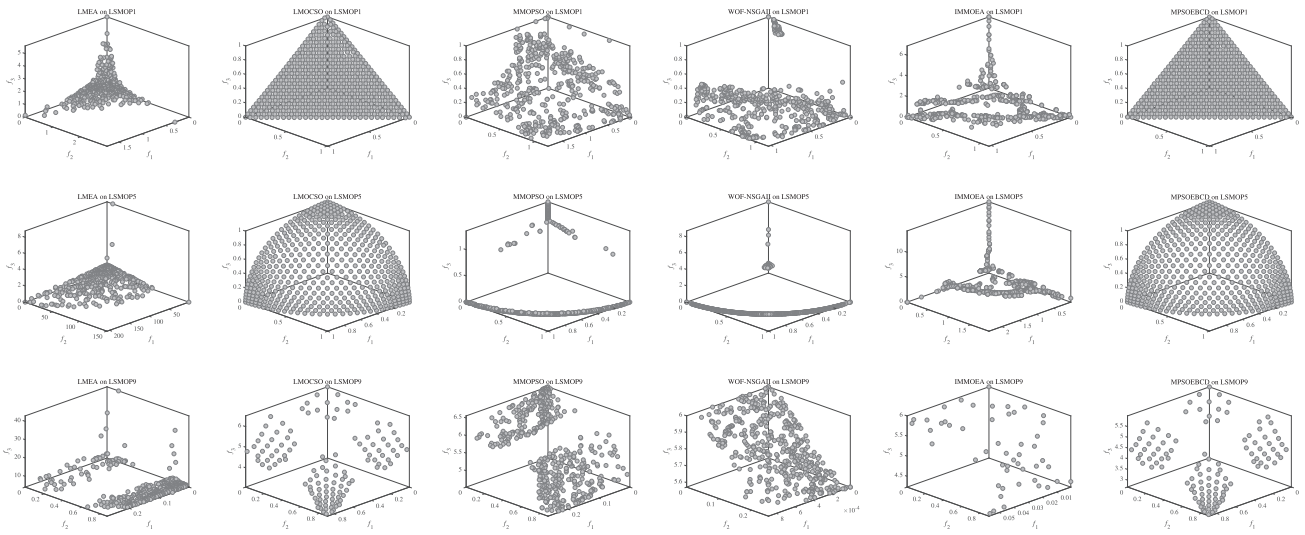

Fig. 6. Nondominated solutions with the median IGD values obtained by LMEA, LMOCSO, MMOPSO, WOF-NSGAII, IMMOEA, and MPSO-EBCD on three-objective LSMOP_1, LSMOP_5, and LSMOP_9 with 1000 decision variables.

solutions is worse than the solutions obtained by the proposed MPSO-EBCD. Finally, MPSO-EBCD obtains well-converged solutions on all of the instances. It is worth noting that the proposed algorithm can obtain better-converged solutions on different benchmarks by adjusting ϕ according to our experimental results, which indicates the effectiveness of the proposed balancing strategy for diversity and convergence.

For further investigation, the IGD convergence curves of the evaluated algorithms are depicted in Figs. A.3 and A.4, in the supplementary material, (see Appendix). One can find that, overall, MPSO-EBCD has a good convergence in comparison to the other five algorithms. LMOCSO also shows competitive convergence in the middle and late stage in some cases as seen in Figs. A.3(a), (d), (h), and (i), and A.4(b) and (h), in the supplementary material. However, MPSO-EBCD exhibits faster convergence than LMOCSO in most of these cases. This can be explained by the fact that the proposed indicators are able to independently provide the evaluation of the convergence and diversity information for particles. This enables the proposed algorithm to explicitly adjust the diversity preservation ability, resulting in a better balance of these two factors in comparison to LMOCSO.

To summarize, the above results demonstrate the competitiveness of the proposed MPSO-EBCD for LSMOPs with respect to both the solution quality and convergence speed. The proposed *WCF* and MLSD are capable of measuring the convergence quality and local diversity information for particles and thus allow the proposed velocity update strategy to explicitly conduct convergence and control the diversity. MPSO-EBCD is able to enhance its diversity preservation ability by adjusting ϕ, resulting in a balance between convergence and diversity.

B. Parameter Sensitivity Analysis

A parameter sensitivity analysis is presented to show the influences of ϕ in (10) on the final performance of MPSO-EBCD. The experiments are conducted based on LSMOP_1–LSMOP_9 with three objectives. The problem dimensionality and the swarm size are 1000 and 496, respectively; ϕ varies from 0 to 0.6 with a step size of 0.1.

Fig. A.5, in the supplementary material, shows the average results obtained by MPSO-EBCD with different settings of ϕ. Overall, one can see that MPSO-EBCD performs better with $\phi = \{0.4, 0.5, 0.6\}$. Table A.1, in the supplementary material, allows a more detailed comparison by listing the average performance with different ϕ as well as the corresponding Friedman ranking test results (the smaller, the better). The results show that increasing ϕ (when $\phi \leq 0.5$) leads to an improved performance. However, the performance deteriorates when $\phi = 0.6$. This means that increasing ϕ can improve the exploration of MPSO-EBCD, however, over-emphasizing diversity preservation potentially results in adverse impacts on convergence. Based on these results, we suggest to set $\phi = 0.5$ for solving LSMOPs with a dimensionality of 1000.

C. Investigation of the Learning Strategies

1) VUSC: This section presents the experimental results with respect to the influence of ψ on the convergence, where $\phi = 0.5$ and ψ is set to $\{0.4, 0.6, 1\}$, respectively. LSMOP_1, LSMOP_5, and LSMOP_9 are selected as representative benchmarks as stated in Section IV-A1.

The results are shown in Figs. A.6 and A.7, in the supplementary material. One can find that increasing values of ψ help the proposed algorithm obtain a faster convergence speed in all cases. This demonstrates that the proposed VUSC is effective in adjusting the convergence pressure for MPSO-EBCD by ψ.

2) VUSD: To test the influence of ϕ on diversity, an experiment is conducted to record the swarm diversity in the decision space during the optimization process, where $\psi = 1$ and ϕ is set to $\{0, 0.2, 0.4\}$, respectively. LSMOP_1, LSMOP_5, and LSMOP_9 are selected as representative benchmarks as stated in Section IV-A1. The swarm diversity is computed based on the updated offspring that have not passed the environment

selection at each generation according to

$$\mathrm{Std}(P) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{d=1}^{D}\left(p_i^d - \bar{p}_i^d\right)^2} \tag{12}$$

$$\bar{p}^d = \frac{1}{N}\sum_{i=1}^{N} p_i^d \tag{13}$$

where $\mathrm{Std}(P)$ indicates the diversity of swarm P; p_i^d is the dth dimension of the ith particle; D is the dimensionality of solution; and N is the swarm size.

The results are shown in Figs. A.8 and A.9, in the supplementary material. Considering of the results in Table A.1, in the supplementary material, we can make the following observations: ϕ is helpful in increasing the diversity in the decision space and leads to better well-converged solutions in overall. Furthermore, as shown by the results of Fig. A.9, in the supplementary material, VUSD evidently influences the diversity on three-objective benchmarks: larger ϕ generally bring higher diversity, especially in the early optimization stage. This finding indicates that larger ϕ is able to enhance MPSO-EBCD in diversity preservation in both the decision and the objective space and improve MPSO-EBCD in exploration.

In summary, the results in Secttion IV-C suggest that VUSC and VUSD are capable of serving convergence and diversity preservation for the proposed algorithm, respectively, leading to the explicit, independent, and simultaneous balance of convergence and diversity.

D. Scalability Test

In this section, the scalability of MPSO-EBCD is tested. The dimensionality of the benchmarks is reduced from 1000 to $\{100, 200, 500\}$ and ϕ is set to 0.3. The results are shown in Tables A.II–A.VII, in the supplementary material.

The results show that MPSO-EBCD exhibits better convergence performance on IGD in comparison to other algorithms in overall. Furthermore, the comparisons on HV demonstrate that MPSO-EBCD is effective in both convergence and diversity of the obtained Pareto solutions. These observations prove that MPSO-EBCD is able to balance convergence and diversity by only slightly adjusting the value of ϕ. This means one can directly balance convergence and diversity by changing ϕ according to the dimensionality. For problems with smaller dimensionality, for example, lower values of ϕ are appropriate.

E. Run Time

An empirical study of the actual run-time complementing the theoretical analysis of computational complexity presented in Section III-E is presented in this section. The results are shown in Table A.VIII, in the supplementary material for a benchmark dimensionality of 1000. The results clearly demonstrate the competitiveness of MPSO-EBCD with respect to run time: MPSO-EBCD wins for 11 times out of 18 benchmarks, followed by MMOPSO which wins for seven times. In summary, the proposed algorithm is efficient in both theory and implementation.

F. Application Study

Finally, MPSO-EBCD and the peer algorithms are applied to solve a real-world multiobjective optimization problem, the ratio error estimation of voltage transformers, a task crucial to the stability of the power system. The optimization problem TREE2 formulated by [49] is selected as it contains the largest number of voltage transformer sets [49]. The detailed definition of the problem can be found in [49].

The following parameter settings were used: the dimensionality of the optimization is set to 1998, the maximum number of fitness evaluations is set to $15000D$, the population size for all the algorithms is set to 300. The peer algorithms adopt the same settings used for the numerical comparisons, while ϕ in MPSO-EBCD is set to 0.4 according to the previous experimental results. Each algorithm is run 30 times on the problem.

The mean HV values and the nondominated solutions obtained with different algorithms are shown in Fig. A.10, in the supplementary material. It can be observed that, first, the proposed MPSO-EBCD outperforms the peer algorithms with respect to the average HV values, and second, the proposed MPSO-EBCD is able to obtain better well-converged nondominated solutions in comparison to the peer algorithms. In summary, the results demonstrate that the proposed MPSO-EBCD is competitive in solving real-world applications.

V. Conclusion and Future Work

In order to balance convergence and diversity of MOEAs for LSMOPs, this article proposes a novel multiobjective particle swarm optimizer. In the proposed algorithm, two novel indicators are introduced to measure the convergence and local diversity information, respectively. Unlike existing PSO based MOEAs, the proposed algorithm adopts two independent operators to explicitly and simultaneously conduct the convergence and control the diversity with the proposed two indicators. The experimental results show that: 1) the proposed algorithm is competitive in solving both benchmarks and real-world applications in comparison to several state-of-the-art MOEAs; 2) the proposed velocity update strategy for diversity preservation is effective in favoring diversity preservation; 3) the proposed algorithm shows satisfying scalability characteristics; and 4) the proposed algorithm is efficient in implementation.

However, the numerical results indicate that the proposed algorithm can be further improved for solving problems with nonlinear variable linkage on the Pareto set, which is a topic we want to address in future work. Furthermore, investigating the proposed algorithm for many-objective optimization, designing adaptive adjustments for diversity preservation, and combining the proposed indicators with other algorithms are promising directions of future research.

References

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

[2] B. Cao, S. Fan, J. Zhao, P. Yang, K. Muhammad, and M. Tanveer, "Quantum-enhanced multi-objective large-scale optimization via parallelism," *Swarm Evol. Comput.*, vol. 57, Sep. 2020, Art. no. 100697.

[3] W.-J. Hong, P. Yang, and K. Tang, "Evolutionary computation for large-scale multi-objective optimization: A decade of progresses," *Int. J. Autom. Comput.*, vol. 18, pp. 155–169, Jan. 2021.

[4] M. Zhang et al., "Many-objective evolutionary algorithm based on relative non-dominance matrix," *Inf. Sci.*, vol. 547, pp. 963–983, Feb. 2021.

[5] D. Li, W. Guo, A. Lerch, Y. Li, L. Wang, and Q. Wu, "An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100789.

[6] L. M. Antonio and C. A. C. Coello, "Coevolutionary multiobjective evolutionary algorithms: Survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 851–865, Dec. 2018.

[7] X. Lu, S. Menzel, K. Tang, and X. Yao, "Cooperative co-evolution-based design optimization: A concurrent engineering perspective," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 173–188, Apr. 2018.

[8] 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.*, 2013, pp. 2758–2765.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] B. Cao, J. Zhao, Y. Gu, Y. Ling, and X. Ma, "Applying graph-based differential grouping for multiobjective large-scale optimization," *Swarm Evol. Comput.*, vol. 53, Mar. 2020, Art. no. 100626.

[14] O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich, "Covering Pareto sets by multilevel evolutionary subdivision techniques," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2003, pp. 118–132.

[15] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas, "Bayesian optimization in high dimensions via random embeddings," in *Proc. IJCAI*, 2013, pp. 1778–1784.

[16] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "Weighted optimization framework for large-scale multi-objective optimization," in *Proc. Genet. Evol. Comput. Conf. Companion*, 2016, pp. 83–84.

[17] 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.

[18] 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.

[19] 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.

[20] Y. Tian, C. Lu, X. Zhang, F. Cheng, and Y. Jin, "A pattern mining-based evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6784–6797, Jul. 2022.

[21] J. Ji, J. Zhao, Q. Lin, and K. C. Tan, "Competitive decomposition-based multiobjective architecture search for the dendritic neural model," *IEEE Trans. Cybern.*, early access, Apr. 27, 2022, doi: 10.1109/TCYB.2022.3165374.

[22] Y. Xu et al., "A multi-population multi-objective evolutionary algorithm based on the contribution of decision variables to objectives for large-scale multi/many-objective optimization," *IEEE Trans. Cybern.*, early access, Jun. 23, 2022, doi: 10.1109/TCYB.2022.3180214.

[23] 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.

[24] 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.

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

[26] Q. Lin et al., "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.

[27] Y. Qi, L. Bao, X. Ma, Q. Miao, and X. Li, "Self-adaptive multi-objective evolutionary algorithm based on decomposition for large-scale problems: A case study on reservoir flood control operation," *Inf. Sci.*, vols. 367–368, pp. 529–549, Nov. 2016.

[28] 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.

[29] 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.

[30] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, Apr. 2020.

[31] 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.

[32] H. Chen, R. Cheng, J. Wen, H. Li, and J. Weng, "Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations," *Inf. Sci.*, vol. 509, pp. 457–469, Jan. 2020.

[33] C. He, S. Huang, R. Cheng, K. C. Tan, and Y. Jin, "Evolutionary multiobjective optimization driven by generative adversarial networks (GANs)," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3129–3142, Jun. 2021.

[34] 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.

[35] Z. Wang, H. Hong, K. Ye, G.-E. Zhang, M. Jiang, and K. C. Tan, "Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 29, 2021, doi: 10.1109/TNNLS.2021.3113158.

[36] 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.

[37] Y. Tian et al., "A large-scale combinatorial many-objective evolutionary algorithm for intensity-modulated radiotherapy planning," *IEEE Trans. Evol. Comput.*, vol. 26, no. 6, pp. 1511–1525, Dec. 2022.

[38] 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.*, early access, Mar. 1, 2022, doi: 10.1109/TEVC.2022.3155593.

[39] K. Zhang, C. Shen, and G. G. Yen, "Multipopulation-based differential evolution for large-scale many-objective optimization," *IEEE Trans. Cybern.*, early access, Jun. 22, 2020, doi: 10.1109/TCYB.2022.3178929.

[40] 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. 6, pp. 859–871, Oct. 2022.

[41] Q. Yang, W.-N. Chen, T. Gu, H. Jin, W. Mao, and J. Zhang, "An adaptive stochastic dominant learning swarm Optimizer for high-dimensional optimization," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1960–1976, Mar. 2022.

[42] 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.

[43] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1510–1522, Jun. 2017.

[44] 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.

[45] 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.

[46] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.

[47] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[48] 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.

[49] 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.

Dongyang Li (Student Member, IEEE) received the M.S. and Ph.D. degrees from the School of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2017 and 2022, respectively.

From 2019 to 2021, he was sponsored by China Scholarship Council to carry on his research at Georgia Institute of Technology. He is currently an Engineer with the Sino-German College of Applied Science, Tongji University. His research interest includes computational intelligence, deep learning and their applications.

Lei Wang (Member, IEEE) received the B.S. degree in electrical technology from the Jiangsu Institute of Technology, Changzhou, China, in 1992, and the M.S. and Ph.D. degrees in automation from Tongji University, Shanghai, China, in 1995 and 1998, respectively.

He is currently a Professor with the Department of Control Science and Engineering, Tongji University. His research interest covers intelligent automation, computational intelligence, heuristic algorithm, and system engineering.

Li Li (Member, IEEE) received the B.S. and M.S. degrees in electrical automation from Shengyang Agriculture University, Shengyang, China, in 1996 and 1999, respectively, and the Ph.D. degree in mechatronics engineering from the Shenyang Institute of Automation, Chinese Academy of Science, Shenyang, in 2003.

She joined Tongji University, Shanghai, China, in 2003, where she is currently a Professor of Control Science and Engineering. She has over 50 publications, including four books, over 30 journal papers, and two book chapters. Her current research interests include production planning and scheduling, computational intelligence, data-driven modeling and optimization, semiconductor manufacturing, and energy systems.

Weian Guo (Member, IEEE) received the M.Eng. degree in navigation, guidance, and control from Northeastern University, Shenyang, China, in 2009, and the Doctor of Engineering degree from Tongji University, Shanghai, China, in 2014.

From 2011 to 2013, he was sponsored by China Scholarship Council to carry on his research with the Social Robotics Laboratory, National University of Singapore, Singapore. He is currently an Associate Professor with the Sino-German College of Applied Science, Tongji University. His interests include computational intelligence, optimization, artificial intelligence, and control theory.

Qidi Wu (Senior Member, IEEE) received the B.S. degree in radio technology and the M.S. degree in automatic control from Tsinghua University, Beijing, China, in 1970 and 1981, respectively, and the Ph.D. degree in automation from ETH Zürich, Zürich, Switzerland, in 1986.

She is currently a Professor with the Department of Electronics and Information Engineering, Tongji University, Shanghai, China. Her research interest covers control theory and engineering, intelligent automation, heuristic algorithm, complex systems scheduling and optimization, system engineering and management engineering, and smart scheduling of home energy management system.

Alexander Lerch (Member, IEEE) received the "Diplom-Ingenieur" degree in EE and the Ph.D. degree in audio communications from TU Berlin, Berlin, Germany, in 2000 and 2008, respectively.

He is an Associate Professor with the Center for Music Technology, Georgia Institute of Technology, Atlanta, GA, USA. Before he joined Georgia Tech, he was the Co-Founder and the Head of Research at his company zplane.development, Berlin, an industry leader in music technology licensing. He has authored more than 50 peer-reviewed journal and conference papers, as well as the text book titled *An Introduction to Audio Content Analysis* (IEEE/Wiley, 2012). His research interests focus on signal processing and machine learning for music information retrieval and audio content analysis.