A Comprehensive Competitive Swarm Optimizer for Large-Scale Multiobjective Optimization

Songbai Liu, *Member*, *IEEE*, Qiuzhen Lin, *Member*, *IEEE*, Qing Li, *Fellow*, *IEEE*, Kay Chen Tan, *Fellow*, *IEEE*

***Abstract* — Competitive swarm optimizers (CSOs) have shown very promising search efficiency in large-scale decision space. However, they face difficulties when solving large-scale multi-/many-objective optimization problems (LMOPs), as their winner particles are selected by random pairwise competition based on only a single evaluation criterion, which does not provide diverse guidance for LMOPs. To alleviate this issue, this paper proposes a comprehensive competitive learning strategy for CSOs using three competition mechanisms to guide the particle search. Specifically, environmental competition classifies winner and loser particles from the swarm, while cognitive competition and social competition select one winner particle as the cognitive component and the social component, respectively, to guide the search for loser particles. This competitive learning strategy aims to enhance the search capability of loser particles and provides diverse search directions for solving LMOPs. When compared with eight competitive optimizers, the experimental results validate the high efficiency and effectiveness of our method in solving nine LMOPs with 2 to 10 objectives and 100 to 5000 variables.**

***Index Terms*—Competitive Swarm Optimizer, Large-Scale Optimization, Multiobjective Optimization.**

I. Introduction

Large-scale multiobjective and many-objective optimization problems (LMOPs) exist in many real-world applications, e.g., capacitated arc routing [1], community detection [2], cloud workflow scheduling [3], and distribution networks [4]. Here, an unconstrained LMOP can be simply modelled as:

Manuscript received XX. XX. 2021; revised XX. XX. 2021; accepted XX. XX. 2021. This work is supported in part by the National Key Research and Development Project, Ministry of Science and Technology, China (Grant No. 2018AAA0101301), and in part by the National Natural Science Foundation of China (NSFC) under Grants 61876162, 61876110, and 61806130, and in part by the Research Grants Council of the Hong Kong SAR under Grant No. PolyU11202418 and Grant No. PolyU11209219, and in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20190808164211203. (Corresponding authors: *Qiuzhen Lin* and *Kay Chen Tan*).

S.B. Liu is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: sbliu2-c@my.cityu.edu.hk).

Q.Z. Lin is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (email: qiuzhlin@szu.edu.cn).

Q. Li and K.C. Tan are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR (e-mails: csqli@comp.polyu.edu.hk, kctan@polyu.edu.hk).

$$\text{Minimize } F(\mathbf{x}) \in \mathbf{Y}, \ \mathbf{x} \in \Omega, \tag{1}$$

where $F(\mathbf{x}) = (f_1(\mathbf{x}),..., f_m(\mathbf{x}))$ and $\mathbf{x} = (x_1, \ldots, x_n)$, respectively, denote m (often conflicting) objective functions in the objective space $\mathbf{Y}$ and n (often interacting) decision variables in the search space Ω, with $m \geq 2$ and $n \geq 100$ [5]. To optimize these m objective functions simultaneously, a swarm-based (or population-based) optimizer is often used to search a set of equally optimal solutions in Ω, called the Pareto set (PS), and the projection of PS in $\mathbf{Y}$ is called the Pareto front (PF) [6]. Thus, the main goal of solving an LMOP is to find a set of solutions that can closely and evenly approximate its true PF [7]-[8]. Generally, metaheuristic swarm optimizers (MSOs), characterized by a population-based iterative search engine, such as evolutionary algorithms [9], estimation of distribution optimizers [10], particle swarm optimization (PSO) [11], and competitive swarm optimizers (CSOs) [5], are applied to solve these LMOPs.

To date, MSOs have been validated to have high efficiency and effectiveness for solving both large-scale single-objective optimization problems (LSOPs) [12]-[13] and multiobjective and many-objective optimization problems (MOPs) [14]-[15]. Specifically, LSOPs are problems with $m=1$ and $n \geq 100$, while MOPs are problems with $m \geq 2$ and $n < 100$. When solving LSOPs, there are two main strategies available for MSOs: 1) by developing effective evolutionary strategies to enhance the search ability in the decision space Ω, e.g., the covariance matrix model is established in [16]-[17] with a way similar to principal component analysis for guiding the heuristic search with multiple directions, and the loser particles of the random competition in [18]-[19] learn from the pairwise winners to enhance their search ability; and 2) by the divide-and-conquer mechanism to divide the decision space, e.g., the cooperative coevolution strategies first divide the LSOP into multiple exclusive low-dimensional subproblems based on a variable grouping method followed by solving each subproblem separately via an existing optimizer [20]-[22], which are further improved in the self-evaluation evolution strategies by solving each subproblem with the aid of a meta-model [23]-[24].

When solving LSOPs, the MSOs mostly focus on improving the search strategies in the decision space Ω, while the MSOs designed for solving MOPs (called MOMSOs) focus on the improvement of environmental selection in the objective space $\mathbf{Y}$. In general, most MOMSOs can be classified into four main categories: 1) Pareto-based MOMSOs gradually improve the definition of Pareto dominance relationship between solutions in $\mathbf{Y}$ to enhance the selection pressure [25]-[26]; 2) decomposition-based MOMSOs with a set of weight vectors can decompose $\mathbf{Y}$ into multiple subspaces and then use a proper aggregation method in each subspace for selection [27]-[28]; 3) clustering-based MOMSOs decompose $\mathbf{Y}$ into multiple subspaces via clustering methods without the guidance of weight vectors [31]-[32]; and 4) indicator-based MOMSOs use a performance indicator, e.g., hypervolume, to directly guide the evolutionary search in Ω and environmental selection in $\mathbf{Y}$ [29]-[30]. However, due to the curse of dimensionality, the performance of these MOMSOs deteriorates significantly when handling LMOPs with $m \geq 2$ and $n \geq 100$ [5]. To alleviate the above issue, the techniques designed for solving LSOPs are directly embedded or adapted into MOMSOs in most of the existing works [33]-[36], which can be classified into the two main categories below.

The first approach is to transform the original LMOP into a set of low-dimensional subcomponents or subproblems using grouping [33]-[34], clustering, analysis, or transformation methods on decision variables for dimension reduction of the search space [38]-[43]. Specifically, in CCGDE3 [33] and MOEA/D-RDG [34], cooperative coevolution strategies with different grouping methods are embedded into two classical MOMSOs, i.e., GDE3 [35] and MOEA/D [27], while in DLS-MOEA [36], a self-evaluation evolution strategy is used to design the dual local search and is embedded into an indicator-based MOMSO, i.e., SMS-EMOA [37]. In addition, to adapt the cooperative coevolution strategy to handle LSMOPs, decision variables are classified by running interdependence analysis in DPCCMOEA [38] and MOEA/DVA [39], defining three kinds of variables (position variables, distance variables, and mixed variables). More specifically, only position and distance variables are detected by running the cosine similarity-based clustering methods in LMEA [40] and S^3-CMA-ES [41]. Furthermore, problem transformation or reformulation strategies are proposed in LSMOF [42], WOF [43], and GLEA [44], where n original variables are assigned to associate with multiple weight variables (far less than n) in different ways, aiming to reduce the search space. However, the above dimension reduction mechanisms are validated to be suitable for solving LMOPs without or with fewer interacting variables but inefficient for those LMOPs with many inseparable variables. The reason behind this is that the processing results of the decision variables in these algorithms may be inaccurate or mismatched [36], and the wrong division of two interacting variables will result in a misleading handling of these variables. Consequently, some essential information about the PS of the target LMOP will be missed, which may finally fall into a local optimum. Moreover, variable processing methods are often computationally expensive, which waste substantial computational resources for analysing variables.

The second approach is to design more powerful search strategies to explore the large-scale variables. For example, in [45], ten crossover operators are embedded into NSGA-III [46] to enhance its performance for solving LMOPs. In [47], new initialization strategies and new genetic operators were designed to address LMOPs with many sparse variables. In [48], information from previous generations is incorporated into the search process to enhance the performance of MOEA/D on

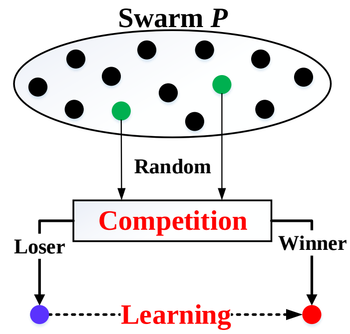


Fig. 1 Random pairwise competitive learning in CSO

LMOPs. In [49], the search efficiency of differential evolution is improved by considering the importance of variables. Recently, a stepped-up CSO and an enhanced CSO with random pairwise competitive learning strategy respectively proposed in LMOCSO [5] and S-ECSO [50] have shown very promising performance for addressing LMOPs.

Inspired by the competitive learning strategy in LMOCSO, this paper follows this research direction and designs a comprehensive competitive swarm optimizer for addressing LMOPs, termed CCSO, which can be embedded into different kinds of MOMSOs to improve their performance for solving LMOPs. The main contributions of this paper are as follows:

(1) A comprehensive competitive learning strategy that includes three different kinds of competition is designed in CCSO to enhance the learning abilities of loser particles.

(2) Environmental competition is implemented to classify the loser and winner particles in a swarm. After this competition, the loser particles can learn from the winner particles in the particle search, and the winner particles can survive to the next iteration of environmental selection.

(3) Cognitive competition and social competition are applied to select the cognitive component and social component, respectively, for guiding loser particles. In this way, each loser particle can efficiently learn from these two winners in the search space to avoid premature convergence or stagnation in the evolutionary search.

The rest of this paper is organized as follows. Section II introduces the related works about CSO and our motivation to design the proposed CCSO. Section III presents the details of CCSO. The experimental results and discussions are provided in Section IV, while the conclusions and future works are given in Section V.

II. Related Works and Motivations

In this section, some related works on CSOs are first introduced in Section II.A. Then, the motivation to design a comprehensive CSO framework for LMOPs is elaborated in Section II.B.

A. Related Works of CSOs

The CSO originally proposed in [19] is a specially designed variant of PSO for problems with large-scale variables. PSO is a typical MSO with the advantages of easy implementation and high search efficiency [11]. In traditional PSO, each particle has two basic attributes (i.e., velocity **v** and position **x**) and two historical memory attributes (i.e., the personal best position p_{best} and the global best position g_{best}). Generally, p_{best} and g_{best} are referred to as, respectively, the cognitive component and social component of a particle. Then, the particle learns from its p_{best} and g_{best} by updating its velocity **v** as follows:

$$\mathbf{v} \leftarrow \omega\mathbf{v} + c_1 r_1 (p_{best} - \mathbf{x}) + c_2 r_2 (g_{best} - \mathbf{x}), \tag{2}$$

and then updating its position **x** as follows:

$$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}, \tag{3}$$

where ω is the inertia weight, c_1 and c_2 are two learning coefficients, and r_1 and r_2 are two random vectors within $[0, 1]^n$. However, the learning strategy in PSO may lead to premature convergence when p_{best} and g_{best} are trapped in local optima [19], especially for LSOPs [51]. Without involving p_{best} and g_{best}, competitive learning strategies are proposed in CSOs to alleviate the issue of premature convergence. In this way, each particle in CSO is a potential leader to guide the search when it wins in the competition. Thus, CSO also has two basic attributes **v** and **x**. Specifically, as shown in Fig. 1, two random particles are first selected from swarm $\boldsymbol{P}$ to compete with each other based on their fitness values, and then only the loser particle with attributes ($\mathbf{v}_l$, $\mathbf{x}_l$) learns from the winner particle with attributes ($\mathbf{v}_w$, $\mathbf{x}_w$) by updating its $\mathbf{v}_l$ as:

$$\mathbf{v}_l \leftarrow r_1\mathbf{v}_l + r_2(\mathbf{x}_w - \mathbf{x}_l) + \varphi r_3(\bar{\mathbf{x}} - \mathbf{x}_l), \tag{4}$$

where r_1, r_2, and r_3 are three random vectors within $[0, 1]^n$, $\bar{\mathbf{x}}$ is the average position of the relevant particles, and φ is the corresponding control parameter. Then, $\mathbf{x}_l$ is updated according to (3). Note that the first part $r_1\mathbf{v}_l$ in (4) can be regarded as the inertia component in (2) by replacing ω with a random vector r_1, which enhances the search capacity. Without historical memory, the winner particle and $\bar{\mathbf{x}}$ are respectively treated as the cognitive component and the social component to guide the loser particle, which simulates biological behaviour in an intelligent and plausible manner [19].

Recently, owing to the efficient search capability of CSO in solving LSOPs [18]-[19], several variants of CSO for LSOPs have been proposed, including the segment-based predominant learning swarm optimizer (SPLSO) [52], level-based learning swarm optimizer (LLSO) [53], ranking-based biased learning swarm optimizer (RBLSO) [54], distributed swarm optimizer (DEGLSO) [55], and hierarchical sorting swarm optimizer (HSSO) [56]. Moreover, CSO has been extended to solve MOPs in CMOPSO [57], to solve LMOPs in LMOCSO [5], and to solve sparse LMOPs in S-ECSO [50].

B. Motivations

As introduced in Section I, most MOMSOs are designed by enhancing the environmental selection for addressing MOPs, while they still use traditional evolutionary search strategies, such as PSO, differential evolution (DE) [35], polynomial mutation (PM) [58], and simulated binary crossover (SBX) [59], which are inefficient for exploring the large-scale space Ω [5]. As experimentally validated in [52]-[57], CSO using random pairwise competitive learning is very efficient and effective in addressing LSOPs. However, few studies have adapted CSO to handle LMOPs with many objectives, i.e., $m \geq 4$ and $n \geq 100$ in (1). In one recent work (i.e., LMOCSO), the position update strategy is enhanced by embedding an accelerant component to get superior performance for solving LMOPs. Nonetheless, competition in LMOCSO still occurs between two randomly selected particles based on a single criterion, which may restrict the learning abilities of the loser particles [51]-[52], especially on LMOPs. Besides, as noted in [36], the main goal of solving LMOPs is to approximate the entire PF, which is different from the target of addressing LSOPs. Thus, when solving LMOPs, it is not effective to directly embed the techniques of CSOs into MOMSOs. Moreover, the learning abilities of particles may be restricted when using one type of competition to determine the cognitive and social components of the loser particles [5], as this approach cannot provide diverse guided directions for LMOPs.

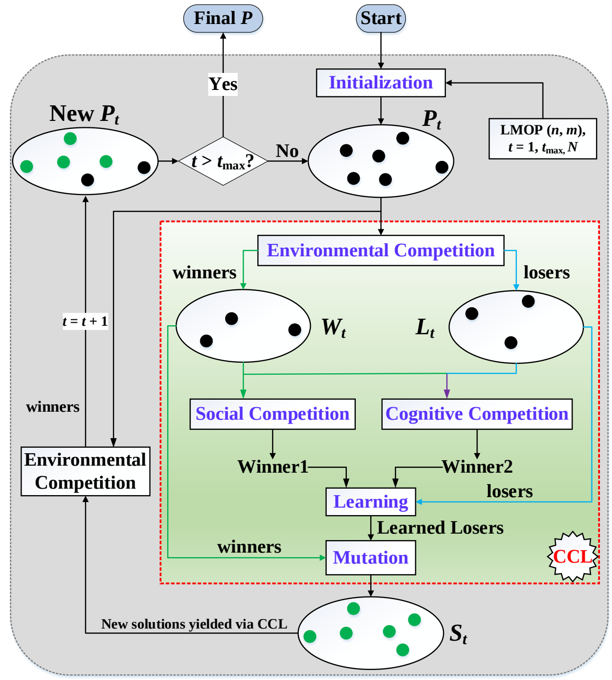


Fig. 2 The general framework of the proposed CCSO.

Based on the above discussions, one natural idea is to design a CSO with more effective competition mechanisms for enhancing the learning abilities of loser particles when solving LMOPs. To do this, a comprehensive competitive swarm optimizer, termed CCSO, is proposed to address LMOPs by applying three kinds of competition. First, environmental competition is implemented in the swarm to classify a set of winner particles and a set of loser particles. The winner particles can better approximate the true PF of the target LMOP and are used to guide the loser particles to search towards the true PF. Then, cognitive competition and social competition are used to select two winner particles as the cognitive component and the social component, respectively, to guide the search of loser particles. These three competition mechanisms constitute our comprehensive competitive learning strategy, which can be embedded into different kinds of MOMSOs to improve the learning abilities of particles in the large-scale decision space and objective space.

Algorithm 1 CCSO

Input: an LMOP with m objectives and n variables, the swarm size N, and the maximum number of iterations $t_{\max}$
Output: the final $\boldsymbol{P}_t$
1: set t=1 and initialize swarm $\boldsymbol{P}_t$ with N particles
2: **while** $t \leq t_{\max}$ **do**
3: $\boldsymbol{S}_t$ = *Comprehensive-Competitive-Learning*($\boldsymbol{P}_t$)
4: $\boldsymbol{U}_t = \boldsymbol{P}_t + \boldsymbol{S}_t$
5: $\boldsymbol{P}_{t+1}$ = *Environmental-Competition*($\boldsymbol{U}_t$)
6: $t = t+1$
7: **end while**
8: **return** $\boldsymbol{P}_t$

Algorithm 2 *Comprehensive-Competitive-Learning*($\boldsymbol{P}_t$)

1: initialize $\boldsymbol{W}_t=\varnothing$, $\boldsymbol{L}_t=\varnothing$, $\boldsymbol{S}_t=\varnothing$
2: ($\boldsymbol{W}_t$, $\boldsymbol{L}_t$) = *Environmental-Competition*($\boldsymbol{P}_t$)
3: **for** each loser ($\mathbf{v}_l$, $\mathbf{x}_l$) in $\boldsymbol{L}_t$
4: $\mathbf{x}_{wc}$ = *Cognitive-Competition*($\mathbf{x}_l$, $\boldsymbol{W}_t$)
5: $\mathbf{x}_{ws}$ = *Social-Competition*($\boldsymbol{W}_t$)
6: update $\mathbf{v}_l$ by learning from $\mathbf{x}_{wc}$ and $\mathbf{x}_{ws}$ with (5)
7: update $\mathbf{x}_l$ like (3)
8: **end for**
9: mutate all $\mathbf{x}_w$ and all $\mathbf{x}_l$ by polynomial mutation
10: $\boldsymbol{S}_t = \boldsymbol{W}_t + \boldsymbol{L}_t$
11: **return** $\boldsymbol{S}_t$

III. The Proposed CCSO

The details of our CCSO are introduced in this section. First, the general framework of CCSO is provided in Section III.A. In the proposed CCSO, a comprehensive competitive learning (CCL) strategy is developed to effectively explore the decision space, in which the environmental competition is performed in the objective space to distinguish between the elite and poor solutions. Then, the details of CCL are given in Section III.B, which introduces the three different forms of competition, i.e., environmental competition, cognitive competition, and social competition. Notably, these three forms of competition are used to guide the competitive search.

A. The General Framework of CCSO

To provide a clear overview of CCSO, its main framework is shown in Fig. 2, where t is the iteration counter and $\boldsymbol{P}_t$ is the current particle swarm. Similar to the canonical framework of an evolutionary algorithm (EA), the general framework of CCSO consists of three main components: 1) Initialization: it specifies the values of some necessary parameters (e.g., the swarm size N and the maximum number of iterations $t_{\max}$) and an initial population $\boldsymbol{P}_{t=1}$; 2) the CCL strategy: similar to the reproduction procedure in an EA, it generates the swarm ($\boldsymbol{S}_t$) with N new particles; and 3) environmental competition: this process selects N winner particles from the combined swarm of $\boldsymbol{P}_t$ and $\boldsymbol{S}_t$ to fill up the next particle swarm ($\boldsymbol{P}_{t+1}$). Particularly, the CCL strategy in Fig. 2 is different from the competitive learning strategy in Fig. 1, as CCL is equipped with three different types of competition (i.e., environmental competition, social competition, and cognitive competition). Concretely, the swarm P_t is divided into a set of winner particles ($\boldsymbol{W}_t$) and a set of loser particles ($\boldsymbol{L}_t$) via the environmental competition, and then the cognitive component and social component of each loser particle are selected from $\boldsymbol{W}_t$ respectively via cognitive competition and social competition, which can effectively guide the search of losers. Notably, the loser particles in $\boldsymbol{L}_t$ can competitively learn from two winner particles to improve their quality, and then both the winner and the learned loser particles run mutation to escape from local optima, which finally produces a new particle swarm ($\boldsymbol{S}_t$) with higher quality.

The pseudocode of CCSO is given in **Algorithm 1** with the following inputs: an LMOP with m objectives and n variables, N, and $t_{\max}$. In line 1, the iteration counter is set to t = 1 and a swarm $\boldsymbol{P}_t$ is initialized to have N randomly sampled particles in the decision space Ω for the target LMOP. Then, the CCL strategy (introduced in **Section III.B**) is applied in line 3 to obtain the updated swarm $\boldsymbol{S}_t$ via the competitive search starting from $\boldsymbol{P}_t$. Next, $\boldsymbol{P}_t$ and $\boldsymbol{S}_t$ are combined in line 4 to obtain the union swarm $\boldsymbol{U}_t$, which has $2N$ particles. At last, all particles of $\boldsymbol{U}_t$ implement environmental competition (introduced in **Section III.B.1**) in line 5 to finally keep N winner particles as $\boldsymbol{P}_{t+1}$. Then, t is increased by 1 in line 6. If t is smaller than $t_{\max}$, the above procedures in lines 2-6 are run iteratively; otherwise, the final swarm $\boldsymbol{P}_t$ is outputted in line 8 as the final result for the target LMOP.

B. Comprehensive Competitive Learning

Generally, each particle in PSO is updated by its inertia weight, cognitive component, and social component in (2). However, in our CCL strategy, only the loser particle is updated by the competitive search using the above three components. Then, the loser and winner particles are all disturbed by mutation, as shown in Fig. 2. Here, the pseudocode of CCL is given in **Algorithm 2** with the following input: swarm $\boldsymbol{P}_t$. In line 1, three empty sets, $\boldsymbol{W}_t$, $\boldsymbol{L}_t$, and $\boldsymbol{S}_t$, are initialized, where $\boldsymbol{W}_t$ and $\boldsymbol{L}_t$ are, respectively, used to preserve the winner and loser particles in environmental competition and $\boldsymbol{S}_t$ maintains the newly generated particles. First, environmental competition (**Algorithm 3**) is applied in line 2 to obtain the winner particles (stored in $\boldsymbol{W}_t$) and the loser particles (stored in $\boldsymbol{L}_t$) from $\boldsymbol{P}_t$, the details of which are introduced in Section III.B.1. Then, each loser particle in $\boldsymbol{L}_t$ with attributes ($\mathbf{v}_l$, $\mathbf{x}_l$) learns from two winner particles in $\boldsymbol{W}_t$ in lines 3-8. Specifically, one winner particle with attributes $(\mathbf{v}_{wc}, \mathbf{x}_{wc})$ is identified in line 4 using cognitive competition (**Algorithm 4**), and another winner with attributes $(\mathbf{v}_{ws}, \mathbf{x}_{ws})$ is selected in line 5 via social competition (**Algorithm 5**). Thus, the loser particle ($\mathbf{v}_l$, $\mathbf{x}_l$) updates its velocity $\mathbf{v}_l$ as follows:

$$\mathbf{v}_l \leftarrow r_1 \mathbf{v}_l + r_2(\mathbf{x}_{wc} - \mathbf{x}_l) + r_3(\mathbf{x}_{ws} - \mathbf{x}_l)\,, \tag{5}$$

and correspondingly updates its position $\mathbf{x}_l$ via (3), as shown in line 6 and line 7, respectively. Here, r_1, r_2, and r_3 are three random real-valued vectors within $[0, 1]^n$. As indicated in [5], the polynomial mutation (PM) operator should be run in line 9 to slightly mutate all winner particles in $\boldsymbol{W}_t$ and all loser particles in $\boldsymbol{L}_t$ to escape from local optima. These results are then added to $\boldsymbol{S}_t$ in line 10. Finally, the newly generated swarm $\boldsymbol{S}_t$ is returned in line 11.

The details of the three competitions used in comprehensive competitive learning are introduced below.

Algorithm 3 *Environmental-Competition*($\boldsymbol{P}_t$)

1: preset $\mathbf{r} = (\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^{|\boldsymbol{P}_t|/2})$, $\boldsymbol{W}_t = \boldsymbol{L}_t = \varnothing$
2: **for** each particle $\mathbf{x} \in \boldsymbol{P}_t$ **do**
3: compute $d_1(\mathbf{r}^i, \mathbf{x})$ by (6) for each $\mathbf{r}^i \in \mathbf{r}$
4: **end for**
5: **for** each $\mathbf{r}^i \in \mathbf{r}$ **do**
6: get the cluster S_i based on (7)
7: particles in S_i are ascending sorted by their fitness (8)
8: **end for**
9: compute $L = \max\{|S_1|, |S_2|, \cdots, |S_{|P_t|/2}|\}$
10: **for** $i = 1$ to L **do**
11: add the ith particle of each cluster (if existed) into G_i
12: **end for**
13: $\boldsymbol{W}_t = G_1 + G_2 + \ldots + G_{k\text{-}1}$ and $\boldsymbol{L}_t = G_{k+1} + G_{k+2} + \ldots + G_L$
14: randomly sort the particles in G_k, and *size* = $|\boldsymbol{P}_t|/2$ - $|\boldsymbol{W}_t|$
15: add the first *size* particles from G_k to $\boldsymbol{W}_t$ and others to $\boldsymbol{L}_t$
16: **return** ($\boldsymbol{W}_t$, $\boldsymbol{L}_t$)

1) Environmental Competition

As noted in SPLSO [52] and LLSO [53], the search ability of CSO in a large-scale decision space may be restricted, as its loser particles learn from only the winner particles that are selected in a random pairwise competition, as shown in Fig. 1. To alleviate this issue in CSO, the goal of the environmental competition in CCSO is to ensure that competition involves all the particles. Thus, two groups of particles are classified rather than two particles, i.e., particles in one group are better than those in the other group for the target problem in terms of convergence and diversity. This requirement is the same as that of environmental selection in most MOMSOs. Therefore, in this paper, environmental competition is designed using an environmental selection strategy from an MOMSO to classify all particles of $\boldsymbol{P}_t$ into two sets of equal size, i.e., $\boldsymbol{W}_t$ preserves the winner particles and $\boldsymbol{L}_t$ keeps the loser particles. Here, a reference-vector-guided strategy introduced in [60] is customized for CCSO.

The pseudocode of environmental competition is provided in **Algorithm 3** with the following input: swarm $\boldsymbol{P}_t$. In line 1, a set of uniformly distributed reference vectors $\mathbf{r} = (\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^{N/2})$ is preset to constitute the $N/2$ clustering centroids to divide the particles in $\boldsymbol{P}_t$ into $N/2$ clusters. Notably, we assume the population size N is even and allow empty clusters in this process. In lines 2-4, the distance $d_1(\mathbf{r}^i, \mathbf{x})$ between each particle $\mathbf{x} \in \boldsymbol{P}_t$ and each centroid $\mathbf{r}^i \in \mathbf{r}$ is calculated as follows:

$$d_1(\mathbf{x}, \mathbf{r}^i) = \frac{(F(\mathbf{x}) - z^*) \cdot (\mathbf{r}^i - z^*)}{\| F(\mathbf{x}) - z^* \|}, \tag{6}$$

where $z^* = (z_1^*, z_2^*, \ldots, z_m^*)$ is an ideal point that is approximated by the minimum value of each objective from all particles in $\boldsymbol{P}_t$. Then, in lines 5-8, the particles closest to the centroid $\mathbf{r}^i$ are clustered into cluster S_i, which can be formulated by

$$S_i = \{\mathbf{x} \in P_t \mid \mathbf{r}^i = \arg\min_{\mathbf{r}^k \in \mathbf{r}} d_1(\mathbf{x}, \mathbf{r}^k)\},\ i = 1, 2, \ldots, N/2. \tag{7}$$

Then, the particles in S_i are sorted in ascending order in line 7 based on their fitness values. The penalty-based boundary intersection (PBI) aggregation value [27] is adopted as the fitness value of each particle $\mathbf{x} \in S_i$, termed $g(\mathbf{x} \mid \mathbf{r}^i, z^*)$, which can be computed as

$$g(\mathbf{x} \mid \mathbf{r}^i, z^*) = d_1(\mathbf{x}, \mathbf{r}^i) + \theta d_2(\mathbf{x}, \mathbf{r}^i), \tag{8}$$

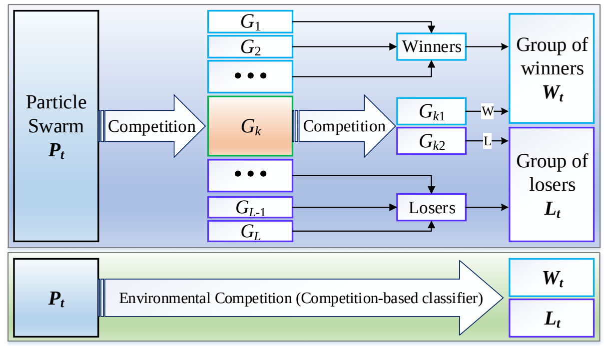


Fig. 3 Illustration of the general process of the environmental competition

with
$$d_2(\mathbf{x}, \mathbf{r}^i) = \left\| F(\mathbf{x}) - z^* - \frac{(\mathbf{r}^i - z^*) d_1(\mathbf{x}, \mathbf{r}^i)}{\|(\mathbf{r}^i - z^*)\|} \right\|, \tag{9}$$

and the parameter θ is used to control the impact of $d_1(\mathbf{x}, \mathbf{r}^i)$ and $d_2(\mathbf{x}, \mathbf{r}^i)$, as adaptively set by

$$\theta = \text{random}(0,1) + \frac{5t}{t_{\max}}, \tag{10}$$

where t and $t_{\max}$ are the current iteration and the maximum iteration, respectively, as set in **Algorithm 1**. Of course, other aggregation methods, such as Tchebycheff [27], achievement scalarizing function [61], and angle-penalized distance [62], can also be used to replace PBI fitness, which will be further studied in the supplementary file of this paper. Next, the ith particle in each cluster S_j is further added to group G_i, where $i = 1, 2, \ldots, L$, and $j = 1, 2, \ldots, N/2$, as shown in lines 10-12. Here, L, which indicates the number of particles in the cluster, has the maximum cardinality in all clusters, as computed in line 9. In this way, particles in swarm $\boldsymbol{P}_t$ are further assigned to multiple groups $\boldsymbol{G}_1$ to $\boldsymbol{G}_L$, where a smaller index of a group indicates a better quality of its particles. Suppose that the first k clusters satisfy the conditions that $|\boldsymbol{G}_1 + \boldsymbol{G}_2 + \ldots + \boldsymbol{G}_{k\text{-}1}| \le N/2$ and $|\boldsymbol{G}_1 + \boldsymbol{G}_2 + \ldots + \boldsymbol{G}_k| > N/2$. Then, particles in $\boldsymbol{G}_1, \boldsymbol{G}_2 \ldots, \boldsymbol{G}_{k\text{-}1}$ are the winners in environmental competition, while particles in $\boldsymbol{G}_{k+1}$ to $\boldsymbol{G}_L$ are the corresponding losers, which are preserved in $\boldsymbol{W}_t$ and $\boldsymbol{L}_t$, respectively, in line 13. The particles in $\boldsymbol{G}_k$ are first sorted by a random method in line 14, followed by considering the first ($N/2$-$|\boldsymbol{G}_1 + \boldsymbol{G}_2 + \ldots + \boldsymbol{G}_{k\text{-}1}|$) particles in $\boldsymbol{G}_k$ as the winners saved into $\boldsymbol{W}_t$ and the remaining particles as the losers saved into $\boldsymbol{L}_t$, as shown in line 15. Finally, the environmental competition classifies the particle swarm $\boldsymbol{P}_t$ into two sets, i.e., $\boldsymbol{W}_t$ with $N/2$ winners and $\boldsymbol{L}_t$ with $N/2$ losers, which are outputted in line 16 as the result of competition. For ease of understanding, the general process of this competition is shown briefly in Fig. 3, in which the environmental competition can be simply imaged as a bi-classifier. Thus, other environmental selection strategies (e.g., Pareto-based strategies [25], indicator-based strategies [29], and clustering-based strategies [31]) can also be used to implement the environmental competition. More discussions will be provided in **Section IV.C**.

After environmental competition, the final winner particles in $\boldsymbol{W}_t$ perform better than the loser particles in $\boldsymbol{L}_t$ in terms of both convergence and diversity. Consequently, each winner in $\boldsymbol{W}_t$ can be a potential leader for each loser in $\boldsymbol{L}_t$, which effectively guides the loser particle to search in diverse directions to

Algorithm 4 *Cognitive-Competition*($\mathbf{x}_l$, $\boldsymbol{W}_t$)

1: initialize $\mathbf{x}_{wc}$ as a random winner from $\boldsymbol{W}_t$, $A=\varnothing$
2: **for** each winner $\mathbf{x}_w \in \boldsymbol{W}_t$
3: **if** $wcc(\mathbf{x}_w, \mathbf{x}_l) = 1$ //*win*
4: add $\mathbf{x}_w$ into $\boldsymbol{A}$
5: **end if**
6: **end for**
7: **if** $|A| > 0$
8: replace $\mathbf{x}_{wc}$ with the randomly selected winner from $\boldsymbol{A}$
9: **end if**
10: **return** $\mathbf{x}_{wc}$

Algorithm 5 *Social-Competition*($\boldsymbol{W}_t$)

1: initialize $\mathbf{x}_{ws}$ as a random winner from $\boldsymbol{W}_t$
2: **for** each $\mathbf{x}_w \in \boldsymbol{W}_t$
3: compute its convergence performance $con(\mathbf{x}_w)$ by (13)
4: compute its diversity performance $div(\mathbf{x}_w)$ by (14)
5: **end for**
6: randomly select two different winners $(\mathbf{x}_w^1, \mathbf{x}_w^2)$ from $\boldsymbol{W}_t$
7: set $\mathbf{x}_{ws} = wsc(\mathbf{x}_w^1, \mathbf{x}_w^2)$ by the competition in (15)
8: **return** $\mathbf{x}_{ws}$

approximate the true PS. Therefore, ensuring an appropriate environmental competition plays the most essential role in the whole process of competitive learning for loser particles.

2) Cognitive Competition

Generally, the cognitive component of a particle in PSO is the local best position, i.e., p_{best} in (2), which attracts the particle to search towards the best position found in its neighbourhood in the decision space. In this paper, all particles are first classified into two sets by environmental competition, i.e., the winner set $\boldsymbol{W}_t$ and the loser set $\boldsymbol{L}_t$. Then, cognitive competition is used to obtain the cognitive component from $\boldsymbol{W}_t$, i.e., $\mathbf{x}_{wc} \in \boldsymbol{W}_t$ in (5), for each loser particle in $\boldsymbol{L}_t$.

The pseudocode of cognitive competition is given in **Algorithm 4** with the loser particle $\mathbf{x}_l$ and the winner set $\boldsymbol{W}_t$ as the inputs. In line 1, $\mathbf{x}_{wc}$ is first initialized as a random member selected from $\boldsymbol{W}_t$. Then, in lines 2-6, each winner particle $\mathbf{x}_w \in \boldsymbol{W}_t$ competes with the loser particle $\mathbf{x}_l$, termed $wcc(\mathbf{x}_w, \mathbf{x}_l)$ to obtain the winner in cognitive competition, as follows:

$$wcc(\mathbf{x}_w, \mathbf{x}_l) = \begin{cases} 1, & \text{if } g(\mathbf{x}_w \mid \mathbf{x}_l, z^*) < \left\| F(\mathbf{x}_l) - z^* \right\| \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

where $z^* = (z_1^*, z_2^*, ..., z_m^*)$ is an ideal point and $g(\mathbf{x}_w \mid \mathbf{x}_l, z^*)$ is similar to the PBI value in (8), as defined by

$$g(\mathbf{x}_w \mid \mathbf{x}_l, z^*) = d_1(\mathbf{x}_w, F(\mathbf{x}_l)) + \theta d_2(\mathbf{x}_w, F(\mathbf{x}_l)), \quad (12)$$

where $F(\mathbf{x}_l)$ is regarded as the reference vector $\mathbf{r}^i$ in (6) and (9) and the parameter θ is set according to (10). $wcc(\mathbf{x}_w, \mathbf{x}_l) = 1$ indicates that $\mathbf{x}_w$ wins the competition in (11). An example of calculating $d_1(\mathbf{x}_w, F(\mathbf{x}_l))$ and $d_2(\mathbf{x}_w, F(\mathbf{x}_l))$ for $\mathbf{x}_w$ is plotted in Fig. 4. Thus, the winner particle $\mathbf{x}_w$ in the competition of (11) is preserved in candidate set $\boldsymbol{A}$, as shown in line 4. Finally, in lines 7-9, $\mathbf{x}_{wc}$ is replaced by a randomly selected member from the non-empty $\boldsymbol{A}$. As shown in Fig. 4 with seven particles in $\boldsymbol{W}_t$, using the competition of (11) on $\boldsymbol{W}_t$, only three particles at the grid region can further win the loser particle, and one of them is outputted as the cognitive component. Here, the red line in Fig. 4 represents the contour curve of the PBI method, whose angle is controlled by the value of θ, which will be further

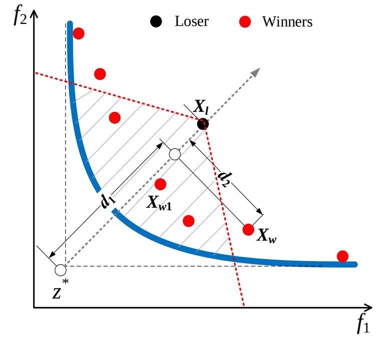


Fig. 4 Illustration of the cognitive competition in 2-objective space

studied in Section IV.H. Based on this contour curve, the objective space is divided into two subspaces, where $\mathbf{x}_w$ in one subspace (indicated by the grid region) is better than $\mathbf{x}_l$ while it is worse in the other subspace.

This cognitive competition is run among all winner particles in $\boldsymbol{W}_t$ to guide the loser particle $\mathbf{x}_l$, i.e., $wcc(\mathbf{x}_w, \mathbf{x}_l)$ in (11). This competition shrinks the range for choosing the cognitive component $\mathbf{x}_{wc}$ from $\boldsymbol{W}_t$, which is different from the experienced local best position for traditional PSO in (2). By learning from $\mathbf{x}_{wc}$ selected via cognitive competition, the search capability of $\mathbf{x}_l$ can be enhanced to move towards the optimal position. Thus, each loser particle is self-supervised to determine its cognitive component for learning. That is, $\mathbf{x}_{wc}$ is not randomly selected from all winner particles but from its own identified winner particles in (11), which can adaptively and effectively guide the local search of the loser particle.

3) Social Competition

Generally, the social component of a particle in PSO is the global best position, i.e., g_{best} in (2), which attracts the particle to search towards the best position found by the whole swarm in the decision space. Here, social competition is a random pairwise competition based on two criteria, and it is implemented to choose the social component, i.e., $\mathbf{x}_{ws} \in \boldsymbol{W}_t$ in (5), for each loser particle $\mathbf{x}_l \in \boldsymbol{L}_t$.

The pseudocode of social competition is provided in **Algorithm 5** with the following input: winner particles $\boldsymbol{W}_t$. In line 1, $\mathbf{x}_{ws}$ is first initialized as a random particle selected from $\boldsymbol{W}_t$. Then, two performance indicators of each particle $\mathbf{x}_w \in \boldsymbol{W}_t$, i.e., $con(\mathbf{x}_w)$ and $div(\mathbf{x}_w)$, are calculated in line 2 and line 5, respectively. Here, $con(\mathbf{x}_w)$ refers to the convergence performance of $\mathbf{x}_w$, which is defined by the Euclidean distance between its objective values $F(\mathbf{x}_w)$ and the ideal point z^*, as computed below:

$$con(\mathbf{x}_w) = \left\| F(\mathbf{x}_w) - z^* \right\|, \quad (13)$$

where $div(\mathbf{x}_w)$ denotes the diversity performance of $\mathbf{x}_w$, which is defined by shift-based density estimation (SDE) [63], and can be calculated as follows:

$$div(\mathbf{x}_w) = \min_{\mathbf{x} \in \boldsymbol{P} \setminus \{\mathbf{x}_w\}} \sqrt{\sum_{i=1}^{m} (\max\{0, f_i(\mathbf{x}) - f_i(\mathbf{x}_w)\})^2}. \quad (14)$$

Next, two different particles $(\mathbf{x}_w^1, \mathbf{x}_w^2)$ are randomly selected from $\boldsymbol{W}_t$, as shown in line 6. Based on the above two indicators, social competition is applied to $\mathbf{x}_w^1$ and $\mathbf{x}_w^2$ to obtain the winning particle, termed $wsc(\mathbf{x}_w^1, \mathbf{x}_w^2)$, as follows:

$$wsc(\mathbf{x}_w^1, \mathbf{x}_w^2) = \begin{cases} \mathbf{x}_w^1, \text{ if } (r \le 0.5 \,\&\, con(\mathbf{x}_w^1) < con(\mathbf{x}_w^2)) \\ \quad\quad \text{or } (r > 0.5 \,\&\, div(\mathbf{x}_w^1) > div(\mathbf{x}_w^2)) \\ \mathbf{x}_w^2, \quad\quad \text{otherwise} \end{cases} \quad (15)$$

where r is a uniformly distributed random number within [0, 1]. Finally, the winner in $(\mathbf{x}_w^1, \mathbf{x}_w^2)$ is regarded as $\mathbf{x}_{ws}$ in line 7 and is returned in line 8. Note that other convergence and diversity performance indicators can also be used to compute $con(\mathbf{x}_w)$ and $div(\mathbf{x}_w)$ in (15), respectively.

This social competition is run between two randomly selected members $(\mathbf{x}_w^1, \mathbf{x}_w^2)$ from $\boldsymbol{W}_t$ based on two criteria, i.e., $wsc(\mathbf{x}_w^1, \mathbf{x}_w^2)$ in (15), which emphasizes the performance in terms of convergence by (13) or diversity by (14) for the social component $\mathbf{x}_{ws} \in (\mathbf{x}_w^1, \mathbf{x}_w^2)$ to guide the loser particle $\mathbf{x}_l$. Here, each particle in $\boldsymbol{W}_t$ is a potential $\mathbf{x}_{ws}$ for $\mathbf{x}_l$, which is different from the experienced global best position for traditional PSO in (2). Thus, two randomly selected winner particles further compete based on two criteria (one for convergence and the other for diversity) to determine the social component $\mathbf{x}_{ws}$ for each loser $\mathbf{x}_l$, which makes $\mathbf{x}_l$ explore the search space in a global manner.

To summarize, environmental competition distinguishes loser particle $\mathbf{x}_l$, while cognitive competition and social competition select two winner particles as the cognitive and social components to guide the particle search in (5). In this way, CCSO can improve the learning ability of $\mathbf{x}_l$ in terms of both exploring and exploiting the whole search space, which may alleviate the issue of premature convergence or evolution stagnation in solving LMOPs.

C. Search Performance Analysis

Since the search patterns of PSO and CSO have been analysed in these related works [18]-[19], the search performance of our proposed CCL is studied here by comparing it with PSO and CSO. More precisely, the differences between these three distinct search paradigms can be expressed as follows:

$$\begin{cases} \text{PSO}: \mathbf{v} \leftarrow \omega\mathbf{v} + c_1 r_1 (p_{best} - \mathbf{x}) + c_2 r_2 (g_{best} - \mathbf{x}) \\ \text{CSO}: \mathbf{v}_l \leftarrow r_3 \mathbf{v}_l + r_4 (\mathbf{x}_w - \mathbf{x}_l) + \varphi r_5 (\bar{\mathbf{x}} - \mathbf{x}_l) \\ \text{CCL}: \mathbf{v}_l \leftarrow r_6 \mathbf{v}_l + r_7 (\mathbf{x}_{wc} - \mathbf{x}_l) + r_8 (\mathbf{x}_{ws} - \mathbf{x}_l) \end{cases} \quad (16)$$

To obtain a deeper and more explicit comparison, (16) can be further transformed as follows:

$$\begin{cases} \text{PSO}: \mathbf{v} \leftarrow \omega\mathbf{v} + \eta_1(\rho_1 - \mathbf{x}), \eta_1 = c_1 r_1 + c_2 r_2 \\ \text{CSO}: \mathbf{v}_l \leftarrow r_3 \mathbf{v}_l + \eta_2(\rho_2 - \mathbf{x}_l), \eta_2 = r_4 + \varphi r_5 \\ \text{CCL}: \mathbf{v}_l \leftarrow r_6 \mathbf{v}_l + \eta_3(\rho_3 - \mathbf{x}_l), \eta_3 = r_7 + r_8 \end{cases} \quad (17)$$

and

$$\begin{cases} \rho_1 = c_1 r_1 (p_{best} / \eta_1) + c_2 r_2 (g_{best} / \eta_1) \\ \rho_2 = r_4 (\mathbf{x}_w / \eta_2) + \varphi r_5 (\bar{\mathbf{x}} / \eta_2) \\ \rho_3 = r_7 (\mathbf{x}_{wc} / \eta_3) + r_8 (\mathbf{x}_{ws} / \eta_3) \end{cases} \quad (18)$$

where r_1 to r_8 are random vectors within $[0, 1]^n$, and the first part in PSO, CSO, and CCL (corresponding to $\omega\mathbf{v}$, $r_3\mathbf{v}_l$, and $r_6\mathbf{v}_l$) are basically the same, which act as the inertia term of the search. Thus, it can be noticed that the search pattern of PSO is driven mainly by ρ_1, which is a combination including two historical best positions (p_{best} and g_{best}) for one particle. As all

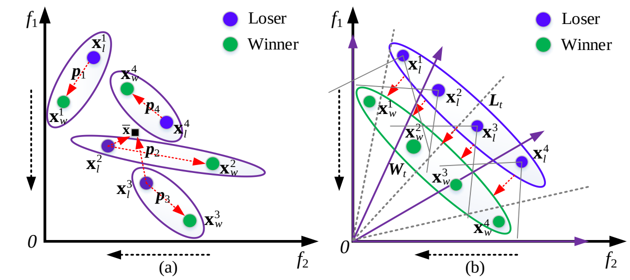


Fig. 5 Illustration of the search paradigm in (a) CSO, and (b) CCL

particles in the swarm have a high probability of memorizing the same g_{best}, while the p_{best} of each particle is very likely to approach the g_{best}, the search via PSO will easily fall into local optima once the g_{best} is a local optimality, which limits its ability to explore the whole landscape. Moreover, the particle's p_{best} and g_{best} may remain unchanged for many generations, which will slow down the convergence speed of the search process in PSO [18]. In contrast, ρ_2 and ρ_3 respectively for CSO and CCL do not require the historical best positions, but use the current superior positions obtained from competition instead, which makes the search more likely to maintain a higher degree of diversity [19]. Regarding the comparison of search performance between CSO and CCL, their difference can be classified from the following aspects based on the examples shown in Fig. 5.

At first, both CSO and CCL distinguish N particles of the current swarm $\boldsymbol{P}_t$ into N/2 losers and N/2 winners via a competition strategy. For CSO, the current particle swarm is randomly divided into N/2 exclusive pairs at first, and then two particles in each pair are compared based on a specified fitness indicator. Finally, one particle with a better fitness value is treated as the winner, while the other one is treated as the loser. As discussed in Section II, such a pair-wise competitive strategy in CSO is very effective for solving LSOPs, but its efficiency is greatly reduced when solving LMOPs, because the two mutually non-dominated solutions for an LMOP are difficult to compare by a single performance indicator. Taking Fig. 5(a) as an example, $\boldsymbol{P}_t$ has eight particles for a 2-objective LMOP, which are randomly divided into four pairs (i.e., $\boldsymbol{p}_1$ to $\boldsymbol{p}_4$), in which only two solutions in $\boldsymbol{p}_1$ can easily be compared based on their objective values, while those in other pairs are incomparable. Please note that the current competition results shown in Fig. 5(a) are based on the SDE as used in LMOCSO [5]. Especially, the incomparability of two different solutions is very common when solving LMOP with many objectives. Regarding CCL, the environmental competition is designed to effectively divide $\boldsymbol{P}_t$ into two subsets (i.e., $\boldsymbol{W}_t$ and $\boldsymbol{L}_t$), where N/2 winners in $\boldsymbol{W}_t$ together are better than N/2 losers in $\boldsymbol{L}_t$ for the LMOP in terms of convergence and diversity, as shown in Fig. 5(b). Thus, these N/2 winners in CCL can guide the search for the N/2 losers more comprehensively and effectively as they have better convergence and diversity, which can provide more diverse search directions than CSO for solving LMOPs.

Secondly, instead of learning from p_{best} and g_{best}, each loser $\mathbf{x}_l$ in CSO and CCL can correspondingly learn from its ρ_2 and ρ_3, which are the combinations including ($\mathbf{x}_w$, $\bar{\mathbf{x}}$) and ($\mathbf{x}_{wc}$,

$\mathbf{x}_{ws}$), respectively. Particularly, in CSO, $\mathbf{x}_w$ is the winner in pair ($\mathbf{x}_l$, $\mathbf{x}_w$) and $\bar{\mathbf{x}}$ is the average position of all particles in $\boldsymbol{P}_t$. As shown in Fig. 5(a) for CSO, both ($\mathbf{x}_w^2$, $\bar{\mathbf{x}}$) and ($\mathbf{x}_w^3$, $\bar{\mathbf{x}}$) will obviously mislead the search of $\mathbf{x}_l^2$ and $\mathbf{x}_l^3$, respectively, which will worsen the search efficiency for the losers and slow down the convergence speed of the overall swarm. In contrast, $\mathbf{x}_{wc}$ and $\mathbf{x}_{ws}$ of $\mathbf{x}_l$ in CCL are determined by the cognitive competition and the social competition, respectively. As shown in Fig. 5(b) for CCL, $\mathbf{x}_l^1$, $\mathbf{x}_l^2$, $\mathbf{x}_l^3$, and $\mathbf{x}_l^4$ obtain their $\mathbf{x}_{wc}$ by randomly selecting a winner from{ $\mathbf{x}_w^1$, $\mathbf{x}_w^2$ }, { $\mathbf{x}_w^1$, $\mathbf{x}_w^2$ }, { $\mathbf{x}_w^2$, $\mathbf{x}_w^3$ }, and { $\mathbf{x}_w^3$, $\mathbf{x}_w^4$ }, respectively, based on the cognitive competition in (11). Thus, in CCL, the loser $\mathbf{x}_l$ has an enhanced search ability by learning from its $\mathbf{x}_{wc}$ to exploit its neighbour space. Considering $\mathbf{x}_{ws}$, it is selected from $\boldsymbol{W}_t$ = { $\mathbf{x}_w^1$, $\mathbf{x}_w^2$, $\mathbf{x}_w^3$, $\mathbf{x}_w^4$ } for each loser $\mathbf{x}_l$ in $\boldsymbol{L}_t$ based on the social competition in (15), which balances the trade-off between convergence and diversity in the search. Therefore, the search ability of $\mathbf{x}_l$ to explore an extended space is further improved by learning from its $\mathbf{x}_{ws}$.

Finally, to validate the effectiveness of CCL in solving LMOPs, empirical studies have been carried out on its search performance by comparing CCL with three traditional evolutionary search strategies (i.e., SBX, DE, and PSO) and one state-of-the-art search strategy (i.e., CSO) in the next section.

IV. Experimental Studies

A. Benchmark Problems and Performance Metrics

The LSMOP test suite proposed in [64], which includes nine test problems, i.e., LSMOP1-LSMOP9, is used in our experimental studies. As defined in [36], these problems can be classified as diversity-type II problems that are difficult to solve. LSMOP1-LSMOP4 are characterized by linear variable linkages in their PSs and a linear unit hyperplane in their PFs, while LSMOP5-LSMOP8 have nonlinear variable linkages in their PSs and a concave unit hypersphere in their PFs. LSMOP9 has nonlinear variable linkages in its PS and disconnected segments in its PF. In this study, the number of objectives (m) in each test problem varies from 2 to 10, i.e., $m \in \{2,3,5,10\}$, and the corresponding number of decision variables (n) varies from 100 to 5000, i.e., $n \in$ {100, 300, 600, 1000, 2000, 5000}. In addition, the number of subcomponents (n_k) in each variable group is set to $n_k = 5$ for all problems. Detailed definitions of the test problems used in this paper can be found in [64].

The well-known inverted generational distance (IGD) [65] and hypervolume (HV) [29] are used as the performance indicators, as they can concurrently reflect the convergence and diversity of the final solution set obtained by the optimizers. A smaller IGD (or larger HV) value indicates a better performance for approximating the true PF. To calculate IGD, a large number of solutions are evenly sampled from the true PF of the target LMOP. In our experiments, 5000 points, 10000 points, 20000 points, and 30000 points are evenly sampled from the true PF to calculate the IGD values for the problems with 2, 3, 5, and 10 objectives, respectively. Regarding the computation of HV, the latest walking fish group

TABLE I
Summary of Significance Test between CCSO and its Six Competitors on LSMOP1-LSMOP9 Problems with IGD and HV

Metric	vs. CCSO	TDEA +/-/~	MOEA/C +/-/~	DVA +/-/~	LMEA +/-/~	WOF +/-/~	LMCSO +/-/~
IGD	$m = 2$	3/31/2	8/28/0	0/36/0	0/34/2	10/25/1	9/18/9
	$m = 3$	2/32/2	4/31/1	2/33/1	1/35/0	3/32/1	2/22/12
	$m = 5$	5/29/2	5/28/3	5/30/1	3/32/1	2/34/0	3/22/11
HV	$m = 2$	4/29/3	6/28/2	0/33/3	0/33/3	7/21/8	5/18/13
	$m = 3$	1/34/1	3/32/1	0/35/1	1/34/1	3/32/1	2/23/11
	$m = 5$	5/24/7	6/28/2	3/30/3	1/31/4	3/33/0	5/21/10

method is used, which needs to carefully set a reference point. To get a normalized HV, the objective vector of each solution $\mathbf{x}$ is first normalized by using $1.1\times(f_1^{\max},..., f_m^{\max})$ in the final particle swarm, followed by setting the reference point to (1.0, …, 1.0). Here, $f_i^{\max}$ represents the maximum value of $f_i(\mathbf{x})$ in the target LMOP's true PF and i = 1, ..., m. In the following tables, the mean IGD and HV results along with their standard deviations are collected from 20 runs for problems with $n \le$ 1000 and 10 runs for problems with n > 1000. In each run, the maximum number of function evaluations (MFE) is set as the termination condition for each algorithm, where MFE = 10000n for 2-objective problems and MFE = 15000n for 3-, 5- and 10-objective problems. Moreover, the particle swarm size or population size N for 2-, 3-, 5- and 10-objective problems is set to N=200, N=300, N=336, and N=440, respectively.

B. The Compared Algorithms and Parameter Settings

For performance comparisons, four algorithms proposed for MOPs are considered: NSGA-II [25], SMPSO [66], TDEA [60], and MOEA/C [31]. Moreover, four competitive algorithms proposed for LMOPs are also included for comparison, i.e., DVA [39], LMEA [40], WOF [43], and LMCSO [5]. The parameter settings of these algorithms, as suggested in their references, are provided in Table S1 of the supplementary file.

Note that NSGA-II, SMPSO, TDEA, MOEA/C and our CCSO are implemented by Java code on jMetal [67], while the remaining optimizers are realized by MATLAB code on PlatEMO [68]. The source code of the proposed CCSO is downloadable in: https://github.com/songbai-liu/CCSO. To ensure a statistically sound conclusion, a Wilcoxon rank sum test with a 0.05 significance level is used to show the statistically significant differences in the performance results. Specifically, the symbols "+", "–" and "~", respectively, indicate that the compared algorithms perform significantly better, worse, and similarly when compared with CCSO for solving the test LMOPs with different numbers of variables, which are provided after the IGD and HV results in the following comparison tables. Besides, the numbers summarizing the above statistical results are collected in the "+/−/~" row for different cases of m and n.

C. Comparisons between CCSO and Existing MOMSOs

Here, CCSO is compared with TDEA, MOEA/C, WOF, DVA, LMEA, and LMCSO on LSMOP1-LSMOP9 with $n \in$ {100, 300, 600, 1000} and $m \in \{2,3,5\}$. Due to space limitations, both the average IGD and HV results from 20 runs of CCSO and its six competitors on these LSMOPs are provided in Tables S3-S8 of the supplementary file, in which the best result in each case is marked in bold and the number of best-performance cases is collected in the second-to-last row of each table. Clearly, our proposed CCSO exhibits significantly better overall performance than its six competitors as it obtains the best performance in most cases of both the IGD and HV results, which validates the superior performance of the proposed CCL strategy in handling the large-scale search space of these LSMOPs. Concretely, Table I summarizes the statistical results by comparing the performance of these six MOMSOs with that of CCSO via Wilcoxon rank-sum test on their IGD and HV results, respectively. From these statistical results, CCSO still shows significant advantages over its six competitors in solving these LSMOP problems for all cases of $m \in \{2,3,5\}$. Next, the possible reasons for the above experimental results of the compared algorithms are analysed and explained from the following three aspects: i) search strategy in the decision space; ii) whether the decision space is processed; and iii) selection strategy in the objective space.

As summarized in Table S22 of the supplementary file, to search in the decision space, the SBX operator is adopted in TDEA, MOEA/C, and LMEA, the DE operator is used in DVA, the PSO strategy is applied in WOF, the CSO strategy is used in LMCSO, and the CCL strategy is employed in our proposed CCSO. As for the search ability among PSO, CSO and CCL, we have made a detailed analysis in Section III.C. Compared with CSO, CCL can not only improve its convergence speed in guiding the whole particle swarm search, but also enhance the search ability of those loser particles by providing diverse search directions. Regarding the traditional evolutionary search strategies (i.e., SBX, DE, and PSO), their search capability deteriorates significantly in the large-scale decision space, which is the reason that MOEA/C and TDEA perform well for MOPs but are particularly poor in solving LMOPs. To alleviate this issue, the search space in DVA, LMEA, and WOF is further processed via the variable grouping methods, as introduced in Section I. However, variable grouping needs to consume computational resources for analysing the correlation between variables, which makes it significantly difficult to ensure the accuracy and cost expensive. Thus, if the grouping of variables is not accurate, it may mislead the search of the whole particle swarm.

Moreover, for the environmental selection in the objective space, the diversity-first and convergence-second strategy is applied in TDEA, MOEA/C, LMCSO, and CCSO, while the opposite strategy is used in the other three competitors. Based on the IGD and HV results collected in Tables S3-S8, it can be found that the selection in the objective space also impacts the performance of an optimizer when solving LMOPs, and the diversity-first and convergence-second strategy is preferred to search the large-scale decision space, which will be further discussed in Section IV. D. Nevertheless, when solving these LSMOPs, the performance of the algorithm is more dependent on the search strategy adopted in the decision space. Although CCSO and TDEA use the similar selection strategies as introduced in Section III. B, the performance of CCSO is obviously better than that of TDEA. To summarize, CCSO performs significantly better than other six competitors, which indicates the effectiveness of the proposed CCL strategy in solving these considered LSMOPs. In addition, the dimension reduction of the search space used in DVA, LMEA, and WOF does not effectively enhance their search efficiency, as their performances are poor on most problems.

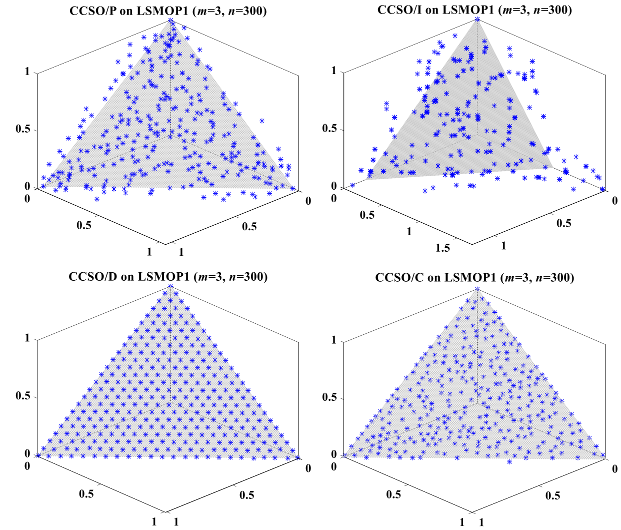


Fig. 6 The final solution sets with the median IGD values obtained by four CCSOs on LSMOP1 with (m = 3, n = 300)

TABLE II
SUMMARY OF SIGNIFICANCE TEST BETWEEN CCSO/P AND THREE ALGORITHMS ON LSMOP1-LSMOP9 PROBLEMS WITH IGD AND HV

Indicators	vs. CCSO/P	NSGA-II(SBX) +/−/~	NSGA-II (DE) +/−/~	SMPSO +/−/~
IGD	m = 2	3/31/2	4/24/8	0/32/4
	m = 3	6/23/7	2/33/1	1/32/3
HV	m = 2	4/29/3	4/23/9	0/29/7
	m = 3	7/24/5	3/29/4	1/32/3

To visually illustrate the performance, the final non-dominated solution sets obtained by the seven compared algorithms with the median IGD values from 20 runs are plotted in Figs. A1-A3 of the supplementary file, respectively, on LSMOP2 (m = 2, n = 1000) with a linear PF, on LSMOP5 (m = 3, n = 1000) with a concave PF, and on LSMOP9 (m = 3, n = 1000) with a disconnected PF. As shown in the subfigures in Fig. A1, only CCSO obtains a final solution set with both good convergence and distribution on LSMOP2, which validates the high efficiency of the proposed CCL strategy in CCSO to search the large-scale decision space. As observed in Figs. A2-A3, only LMCSO and CCSO obtain a set of solutions with good distribution in terms of convergence and diversity on both LSMOP5 and LSMOP9, which also indicates the promising performance of our CCL in solving LSMOPs

D. Influence of the Environmental Competition

As discussed in Section III.B, during the competitive learning process of particles in CCSO, environmental competition plays the most essential role, as it is used to support the competitive learning of loser particles in the search space and is also applied to run the environmental selection for all particles in the objective space. Specifically, environmental competition can distinguish winner particles and loser particles in a swarm. Thus, the impact of environmental competition in CCSO is analysed by replacing **Algorithm 3** with three other environmental selection strategies, i.e., Pareto-based strategy, indicator-based strategy, and clustering-based strategy, giving the variants of CCSO, respectively, termed CCSO/P, CCSO/I, and CCSO/C. Specifically, the strategies from NSGA-II [25], SMS-EMOA [37], and MOEA/C [31] are adopted to customize the environmental competition with the similar process as in Fig. 3 for CCSO/P, CCSO/I, and CCSO/C, respectively. Concretely, particles in CCSO/P and CCSO/I are assigned to multiple groups $\boldsymbol{G}_1$ to $\boldsymbol{G}_L$ by using non-dominated sorting. Then, particles in $\boldsymbol{G}_k$ are sorted in descending order by their crowding distance in CCSO/P and by their hypervolume value in CCSO/I, respectively. In CCSO/C, particles are first divided into multiple clusters via hierarchical clustering [31] followed by sorting the particles in each cluster with ascending order based on the sum of their objective values [26]. Next, the first particle of each cluster is assigned to group $\boldsymbol{G}_1$, the second is assigned to group $\boldsymbol{G}_2$, and so on, generating groups $\boldsymbol{G}_1$ to $\boldsymbol{G}_L$. After that, particles in $\boldsymbol{G}_k$ are randomly sorted in CCSO/C.

First, the final solution sets with the median IGD values obtained by these four CCSOs on LSMOP1 in the case of (m = 3, n = 300) are plotted in Fig. 6 (the original CCSO introduced in **Section III** is termed CCSO/D here). As observed from the subfigures in Fig. 6, CCSO/P and CCSO/I perform poorly in terms of convergence and diversity on LSMOP1, while both CCSO/D and CCSO/C obtain final solutions with good convergence and even distribution. Considering the environmental competition in CCSO/P and CCSO/I, particles are divided into half winners and half losers via a convergence-first and diversity-second rule, i.e., non-dominated sorting followed by crowding-distance-based or hypervolume-based sorting, respectively. Conversely, in CCSO/D and CCSO/C, the environmental competition follows a diversity-first and convergence-second strategy, where particles are first grouped into multiple clusters to maintain diversity and then the particles in each cluster are sorted based on the sum of objective values to ensure convergence. In particular, CCSO/D presets the evenly distributed reference vectors as the cluster centroids [58], while CCSO/C dynamically adjusts the centroids based on the clustering results [31]. Thus, from the final solution set shown in Fig. 6, it is more efficient to solve large-scale LSMOP1 by prioritizing diversity and then pursuing convergence in the environmental competition of CCSOs.

To further verify the above findings, the detailed average IGD and HV results of CCSO/P, CCSO/D, and CCSO/C on LSMOP1-LSMOP9 with 100 to 1000 variables are recorded in Tables S9-S14 for the cases of 2, 3, and 5 objectives, respectively (Tables S9-S14 are provided in the supplementary file). Here, the performance results of CCSO/I on these LSMOPs are not considered due to the extremely high computational cost. Clearly, CCSO/P performs significantly worse than CCSO/D and CCSO/C on these LSMOPs with $m \in \{2,3,5\}$ and $n \in \{100,300,600,1000\}$, as CCSO/P can only obtain the best results in 5, 1, and 0 of 36 problems for the IGD cases of 2, 3, and 5 objectives, respectively. CCSO/D slightly outperforms CCSO/C on these large-scale optimization problems, as CCSO/D performs best in more cases. This is because LSMOP1-LSMOP4 and LSMOP5-LSMOP8 problems are characterized by linear PFs and concave PFs, respectively, which can be evenly covered by the preset reference vectors in CCSO/D. Thus, the performance of CCSO/D appears to be more efficient and stable on these test problems. For LSMOP9 with a disconnected PF, the IGD/HV results of CCSO/D are better than those of CCSO/C in the case of m = 2 (as observed in Table S9 and Table S11), but worse than those of CCSO/C in the case of m = 5 (as observed in Table S13 and Table S14). Therefore, the performance of an optimizer in solving LMOPs is also dependent on the target problem's PF.

To summarize, the environmental competition in CCSOs should prefer to maintain the diversity and then supplement the convergence, which can enhance the search capability in the large-scale decision space. Moreover, the performance of environmental competition also strongly depends on the PF of the target LSMOPs in the objective space. Therefore, CCSO/D and CCSO/C are recommended to solve LSMOPs in this paper, where CCSO/D and CCSO/C are better for solving LSMOPs with regular PFs and irregular PFs, respectively.

E. Effectiveness of CCL Strategy

In Section IV.D, the performance of the proposed CCSO is studied by comparing with other MOMSOs, which validates the effectiveness of its CCL strategy. Here, to verify the effectiveness of the CCL strategy further clearly with a fair experimental setting, CCSO/P is compared with NSGA-II (SBX), NSGA-II (DE) and SMPSO. Specifically, the same environmental selection via non-dominated sorting and crowding distance is adopted in these four algorithms. For the reproduction, SBX and DE are applied to generate the offspring populations in NSGA-II (SBX) and NSGA-II (DE), respectively. The PSO strategy and the CCL strategy are used to update the particles in SMPSO and CCSO/P, respectively. Moreover, polynomial mutation is also applied in these four optimizers. In Fig. 7, the evolutionary processes of the above four algorithms on LSMOP1, LSMOP5, and LSMOP9 with m=3 and n=600 are illustrated, where the average IGD values over 20 runs are used. As observed from Fig. 7, the final average IGD value obtained by CCSO/P is significantly better than those obtained by NSGA-II (SBX), NSGA-II (DE), and SMPSO, as the competitors appear to fall into local optima. Thus, the advantages of our CCL strategy in CCSO/P are validated by solving these three LSMOPs.

Furthermore, the detailed average IGD and HV results of the above four algorithms on LSMOP1-LSMOP9 with 100 to 1000 variables are recorded in Tables S15 to S18 in the supplementary file for the 2- and 3-objective cases, respectively. In addition, a summary of significance tests between CCSO/P and the three competitors is given in Table II based on these IGD and HV results with different cases of m. From these results, we can find that CCSO/P performs significantly better than the three competitors on LSMOP1-LSMOP9 with different numbers of variables (i.e., n = 100, 300, 600, or 1000) and objectives (i.e., m = 2 or 3) due to the effective searching

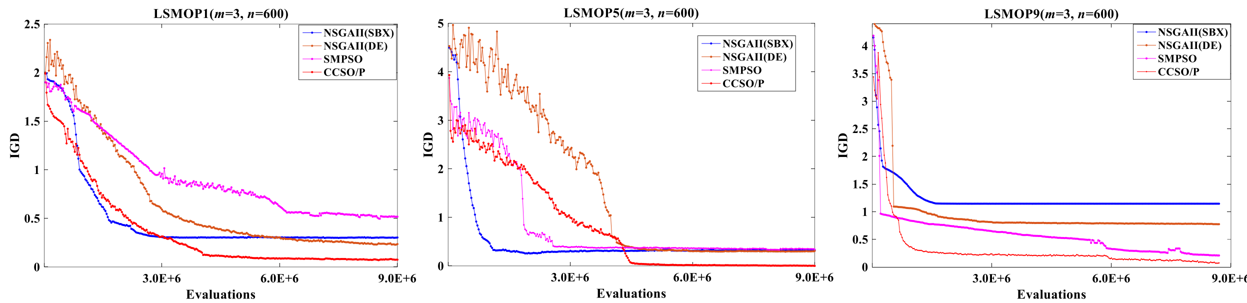


Fig. 7 Evolutionary process of CCSO/P and its three competitors on LSMOP1, LSMOP5, and LSMOP9 with $m = 3$ and $n = 600$ based on IGD values

strategy (i.e., CCL), which can handle the large-scale decision space better than the traditional genetic operators. Concretely, CCSO/P is outperformed by NSGA-II (SBX), NSGA-II (DE), and SMPSO in only 3 (4), 4 (4), and 0 (0) of 36 problems in the case $m = 2$ with IGD (HV) and in only 6 (7), 2 (3), and 1 (1) of 36 problems in the case $m = 3$, respectively. Therefore, we can conclude that the new particles yielded based on CCL are more effective than those generated based on the three traditional EA operators, i.e., SBX, DE, and PSO.

F. Further Ablation Studies

To experimentally study the effectiveness of cognitive co mpetition and social competition, three variants of CCSO are considered: CCSO-V1, CCSO-V2, and CCSO-V3. Specifically, the cognitive component of each loser particle is randomly selected from all winner particles in CCSO-V1, and the social component of each loser particle is randomly selected from all winner particles in CCSO-V2. Both the cognitive component and the social component of each loser particle are randomly selected from all winner particles in CCSO-V3. The IGD results of these three variants on LSMOP1-LSMOP9 problems with (m=2 and n=300) are provided in Table III. As observed from the results, both social competition and cognitive competition can significantly improve the performance of CCSO in solving these LSMOPs, as CCSO performs significantly better than its three variants in most cases, which validates the effectiveness of these two forms of competition in CCSO. Finally, we design another variant of CCSO, termed CCSO-V4, by randomly selecting one winner for each loser using the learning strategy proposed in LMOCSO. The IGD results of this variant on these nine LSMOPs are provided in Table III. From the results, we can observe that CCSO also performs better than CCSO-V4 in most cases, which validates that the proposed CCL strategy in CCSO is more efficient than the CSO in LMOCSO, as the proposed CCL is equipped with three competition mechanisms to effectively guide the particle search, while the original CSO is only designed by a random pair-wise competition.

G. Scalability to Higher Dimensionality

In the previous subsections, the superior performance of the proposed CCSO on LMOPs with the dimension of variables n within 100 to 1000 and the number of objectives m within 2 to 5 is validated. Here, the scalability of CCSO to solve LMOPs with higher dimensionality of variables (i.e., n = 2000, 5000) and with more objectives (i.e., m=10) is further studied.

Table S19 in the supplementary file presents the average IGD results of CCSO/D and CCSO/C on LSMOP1-LSMOP9 with $m \in \{2,3,5\}$ and $n \in \{2000, 5000\}$. The performance of CCSO/D and CCSO/C does not deteriorate on most problems with 2000 and 5000 variables when compared to the cases with fewer than 2000 variables (see Table I and Table S9-S14). The final solution sets obtained by CCSO/D and CCSO/C with the median IGD values from 10 runs on LSMOP1, LSMOP4, LSMOP8, and LSMOP9 are plotted in Fig. A4 of the supplementary file and Fig. 8, including the cases of (m =2, n = 5000) and (m = 3, n = 5000). Fig. A4 shows that the solutions obtained by CCSO/D and CCSO/C basically cover the entire true PFs of these LSMOPs with m = 2, which validates the high efficiency of our CCSO in solving these 2-objective LMOPs. Considering the case of (m = 3, n = 5000), CCSO/D is promising in solving LMOPs with regular (linear or concave here) PFs, i.e., LSMOP1, LSMOP4, and LSMOP8, while CCSO/C is better for solving LSMOP9 with a disconnected PF. Moreover, the average IGD results of MOEA/DVA, LMEA, LMOCSO, and WOF, in the case of n = 5000, are given in Table S20 of the supplementary file, where CCSO/D and CCSO/C still show significant advantages over these four competitors.

Finally, Table S21 of the supplementary file gives the IGD comparison results of CCSO/D and its six competitors on LSMOP2, LSMOP5, and LSMOP9 problems with m = 10 and n = 300. As observed from the results recorded in Table S21, CCSO/D shows the best performance on these 10-objective LSMOPs. To visually illustrate the distribution of solutions in 10-dimensional objective space, the final solution sets obtained by all considered algorithms with median IGD values from 20 runs for LSMOP2 are plotted in Fig. A5 of the supplementary file, where CCSO/D obtains the best distributions.

H. Parameter Sensitivity and Complexity Analysis

From the parameter settings for all considered algorithms in Table S1 of the supplementary file, the proposed CCSO in this paper requires a minimum of external parameters. Nevertheless, the parameter θ used in (8) and (12) are studied here. As shown in (10), θ in CCSO is incrementally changed from *rand* (0, 1) to 5+*rand* (0, 1) during the evolutionary process, which can effectively control the selection of cognitive components for loser particles. The detailed sensitivity analysis of θ is provided in the supplementary file due to page limitations. Moreover, the computational complexity analysis of CCSO is also presented in the supplementary file.

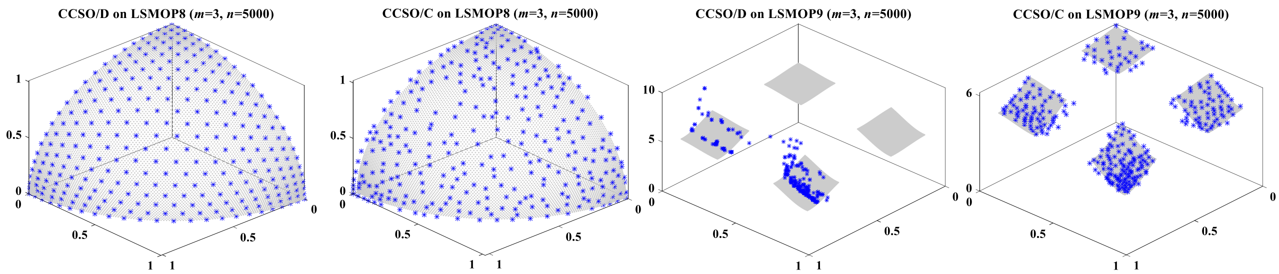


Fig. 8 The final solution sets obtained by CCSO/D and CCSO/C on LSMOP1, LSMOP4, LSMOP8, and LSMOP9 with (m = 3, n = 5000)

TABLE III
IGD RESULTS OF CCSO/D AND ITS THREE VARIANTS ON LSMOP1-LSMOP9 PROBLEMS WITH 2 OBJECTIVES AND 300 VARIABLES

Problems	(m, n)	CCSO/D-V1	CCSO/D-V2	CCSO/D-V3	CCSO/D-V4	CCSO/D
LSMOP1	(2, 300)	2.156E-03(6.98E-05)-	2.146E-03(2.64E-05)-	2.234E-03(6.33E-05)-	2.004E-03(4.87E-05)~	**1.996E-03(5.18E-05)**
LSMOP2	(2, 300)	3.894E-03(7.18E-04)-	3.542E-03(4.91E-04)-	3.954E-03(4.11E-04)-	3.344E-03(5.07E-04)~	**3.269E-03(3.25E-04)**
LSMOP3	(2, 300)	5.586E-01(7.93E-02)-	5.504E-01(7.05E-02)~	5.903E-01(9.57E-02)-	5.401E-01(6.73E-02)~	**5.384E-01(4.74E-02)**
LSMOP4	(2, 300)	5.272E-03(1.17E-04)-	5.263E-03(1.06E-04)-	5.382E-03(1.17E-04)-	5.402E-03(5.60E-04)-	**5.133E-03(1.25E-04)**
LSMOP5	(2, 300)	2.062E-03(2.39E-05)~	2.074E-03(4.81E-05)~	2.368E-03(5.46E-05)-	2.068E-03(3.24E-05)~	**2.018E-03(2.28E-05)**
LSMOP6	(2, 300)	**3.409E-01(8.70E-03)~**	3.955E-01(3.43E-02)-	4.718E-01(1.14E-02)-	3.783E-01(2.43E-02)-	3.646E-01(2.03E-02)
LSMOP7	(2, 300)	**6.981E-01(5.57E-01)+**	1.056E+00(4.32E-01)+	1.100E+00(5.87E-01)+	7.906E-01(2.12E-01)+	1.271E+00(3.44E-01)
LSMOP8	(2, 300)	2.663E-03(1.09E-04)-	2.646E-03(4.53E-04)-	6.015E-03(1.04E-03)-	3.124E-03(1.04E-03)-	**2.549E-03(5.29E-04)**
LSMOP9	(2, 300)	4.610E-03(3.41E-05)-	4.709E-03(2.87E-05)-	5.158E-03(3.01E-05)-	4.183E-03(4.17E-05)~	**4.108E-03(3.68E-05)**
+/-/~		1/6/2	1/6/2	1/8/0	1/3/5	--

V. CONCLUSIONS AND FUTURE WORK

In this paper, a comprehensive competitive swarm optimizer, called CCSO, has been proposed for LMOPs. In CCSO, a comprehensive competitive learning strategy that includes three kinds of competition is designed to enhance the search efficiency for particles. Specifically, environmental competition is used to classify the winner and loser particles from the swarm. Then, for each loser particle, two winner particles are selected to guide its particle search using cognitive competition and social competition. In this way, diversified search directions are provided to enhance the search efficiency of the loser particles. To study the impact of environmental competition in our CCSO, four classic environmental selection strategies are considered, and the experimental results show that the diversity-first and convergence-second environmental competition in CCSO is more promising. Thus, CCSO/D and CCSO/C are recommended in this paper to solve LMOPs. When compared to several algorithms designed for solving LMOPs, the experimental results also validate the superior performance of CCSO/D when solving the LSMOP1 to LSMOP9 test problems.

In our future work, the environmental competition of CCSO will be further studied by considering competition in both the decision and objective spaces to solve LMOPs with multi-modal landscapes, e.g., LSMOP3. In addition, the social competition strategy with two criteria will be further improved by considering more performance metrics in the normalized objective space. At last, the impact of setting different maximum iterations or maximum function evaluations (e.g., a limited computing resource) on the search behaviour in handling LMOPs will also be studied in our future work.

REFERENCES

[1] R. Shang, K. Dai, L. Jiao, R. Stolkin, “Improved Memetic Algorithm Based on Route Distance Grouping for Multiobjective Large Scale Capacitated Arc Routing Problems,” *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 1000-1013, 2016.

[2] C. Liu, J. Liu, Z. Jiang, “A Multiobjective Evolutionary Algorithm Based on Similarity for Community Detection from Signed Social Networks,” *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274-2287, 2014.

[3] Z. Chen, Z. Zhan, Y. Lin, Y. Gong, et al., “Multiobjective Cloud Workflow Scheduling: A Multiple Populations Ant Colony System Approach,” *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2912-2926, 2019.

[4] W. Chen, Y. Jia, F. Zhao, X. Luo, X. Jia, J. Zhang, “A Cooperative Co-Evolutionary Approach to Large-Scale Multisource Water Distribution Network Optimization,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 842-857, 2019.

[5] Y. Tian, X. Zheng, X. Zhang, Y. Jin, “Efficient Large-Scale Multiobjective Optimization Based on a Competitive Swarm Optimizer,” *IEEE Trans. Cybern.*, vol. 50, no.8, pp. 3696-3708, 2020.

[6] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, Y. S. Ong, “A New Decomposition-Based NSGA-II for Many-Objective Optimization,” *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 48, no. 7, pp. 1191-1210, 2018.

[7] Y. R. Naidu and A. K. Ojha, “Solving Multiobjective Optimization Problems Using Hybrid Cooperative Invasive Weed Optimization with Multiple Populations,” *IEEE Trans. Syst., Man, Cybern., Syst., vol. 48, no. 6*, pp. 821–832, 2018.

[8] C. He, R. Chen, D. Yazdani, “Adaptive Offspring Generation for Evolutionary Large-Scale Multiobjective Optimization,” *IEEE Trans. Syst., Man, Cybern., Syst.*, in press, 2020.

[9] J. Wang, W. Ren, Z. Zhang, H. Huang, and Y. Zhou, “A Hybrid Multiobjective Memetic Algorithm for Multiobjective Periodic Vehicle Routing Problem with Time Windows,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4732-4745, 2020.

[10] Y. Liang, Z. Ren, X. Yao, Z. Feng, A. Chen, W. Guo, “Enhancing Gaussian Estimation of Distribution Algorithm by Exploiting Evolution Direction With Archive,” *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 140-152.

[11] Q. Lin, S. Liu, 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, 2018.

[12] H. Li, M. Gong, C. Wang, Q. Miao, “Pareto Self-Paced Learning Based on Differential Evolution,” *IEEE Trans. Cybern.*, in press, doi: 10.1109/TCYB.2019.2906383, 2019.

[13] Y. Sun, M. Kirley. S. K. Halgamuge, “A Recursive Decomposition Method for Large Scale Continuous Optimization,” *IEEE Trans. Evol.*

Comput., vol. 22, no. 5, pp. 647-661, 2018.

[14] Q. Kang, X. Song, M. Zhou, L. Li, "A Collaborative Resource Allocation Strategy for Decomposition-Based Multiobjective Evolutionary Algorithms," *IEEE Trans. Syst., Man, Cybern., Syst*., vol. 49, no. 12, pp. 2416–2423, 2019.

[15] W. Qiu, et al., "Ensemble Many-Objective Optimization Algorithm Based on Voting Mechanism", *IEEE Trans. Syst., Man, Cybern*., Syst., in press, 2020.

[16] X. He, Y. Zhou, Z. Chen, J. Zhang, W. Chen, "Large-Scale Evolution Strategy Based on Search Direction Adaptation," *IEEE Trans. Cybern*., vol. 51, no. 3, pp. 1651-1665, 2021.

[17] Z. Li, Q. Zhang, "A Simple However, Efficient Evolution Strategy for Large-Scale Black-Box Optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 637-646, 2018.

[18] R. Cheng, Y. Jin, "A Social Learning Particle Swarm Optimization Algorithm for Scalable Optimization," *Information Sciences*, vol. 291, pp. 43-60, 2015.

[19] R. Cheng, Y. Jin, "A Competitive Swarm Optimizer for Large Scale Optimization," *IEEE Trans. Cybern*., vol. 45, no. 2, pp. 191-204, 2015.

[20] X. Zhang, Y. Gong, Y. Lin, J. Zhang, S. Kwong, J. Zhang, "Dynamic Cooperative Coevolution for Large Scale Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp.935-948, 2019.

[21] M. N. Omidvar, X. Li, Y. Mei, X. Yao, "Cooperative Co-Evolution With Differential Grouping for Large Scale Optimization," *IEEE Trans. Evol. Comput*., vol. 18, no. 3, pp. 378-393, 2014.

[22] X. Ma, X. Li, Q. Zhang, K. Tang, Z. Liang, W. Xie, Z. Zhu, "A Survey on Cooperative Co-Evolutionary Algorithm," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 421-441, 2019.

[23] Q. Fan, X. Yan, Y. Zhang, C. Zhu, "A Variable Search Space Strategy Based on Sequential Trust Region Determination Technique," *IEEE Trans. Cybern*., vol. 51, no. 5, pp. 2712-2724, 2021.

[24] P. Yang, K. Tang, X. Yao, "Turning High-Dimensional Optimization into Computationally Expensive Optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 143-159, 2018.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[26] Y. Xiang, Y. R. Zhou, M. Q. Li, "A Vector Angle based Evolutionary Algorithm for Unconstrained Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131-152, 2017.

[27] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evol. Comput*., vol. 11, no. 6, pp. 712–731, 2007.

[28] S. Liu, Q. Lin, K. Wong, L. Ma, C. A. Coello Coello, D. Gong, "A Novel Multi-Objective Evolutionary Algorithm with Dynamic Decomposition Strategy," *Swarm and Evol. Comput.*, vol. 48, pp. 182-200, 2019.

[29] S. Jiang, J. Zhang, "A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm," *IEEE Trans. Cybern*., vol. 45, no. 10, pp. 2202-2213, 2015.

[30] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume Based Many-Objective Optimization," *Evol. Comput.*, vol.19, no. 1, pp. 45-76, 2011.

[31] Q. Lin, S. Liu, K. Wong, M. Gong, C. A. Coello Coello, J. Chen, J. Zhang, "A Clustering-Based Evolutionary Algorithm for Many-Objective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 391-405, 2019**.**

[32] S. Liu, J. Zheng, Q. Lin, K. C. Tan. "Evolutionary Multi and Many-Objective Optimization via Clustering for Environmental Selection," Information Sciences, vol. 578, pp. 930-949, 2021.

[33] L. M. Antonio, C. A. Coello Coello, "Use of Cooperative Coevolution for Solving Large Scale Multiobjective Optimization Problems," *in Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, pp. 2758–2765, 2013.

[34] A. Song, Q. Yang, W. Chen, J. Zhang, "A Random-Based Dynamic Grouping Strategy for Large Scale Multi-Objective Optimization," *in Proc. IEEE Congr. Evol. Comput.*, pp. 468–475, 2016.

[35] S. Kukkonen, J. Lampinen, "GDE3: The Third Evolution Step of Generalized Differential Evolution," *in Proc. IEEE Congr. Evol. Comput.*, vol. 1. Edinburgh, U.K., pp. 443–450, 2005.

[36] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, X. Yao, "A Scalable Indicator-Based Evolutionary Algorithm for Large-Scale Multiobjective Optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 525-537, 2019.

[37] N. Beume, B. Naujoks, M. Emmerich, "SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume," *Eur. J. Oper. Res*., vol. 181, no. 3, pp. 1653–1669, 2007.

[38] B. Cao, J. Zhao, Z. Lv, X. Liu, "A Distributed Parallel Cooperative Coevolutionary Multiobjective Evolutionary Algorithm for Large-Scale Optimization," *IEEE Trans. Industrial Informatics*, vol. 13, no. 4, pp. 2030-2038, 2017.

[39] X. Ma, F. Liu, 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, 2016.

[40] X. Zhang, Y. Tian, R. Chen, 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, 2018.

[41] H. Chen, R. Cheng, J. Wen, H. Li, J. Weng, "Solving Large-Scale Many-Objective Optimization Problems by Covariance Matrix Adaptation Evolution Strategy with Scalable Small Subpopulations," *Information Sciences*, vol. 509, pp. 457-469, 2020.

[42] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, X. Yao, "Accelerating Large-Scale Multiobjective Optimization via Problem Reformulation," *IEEE Trans. Evol. Comput*., vol. 23, no. 6, pp. 949-961, 2019.

[43] H. Zille, H. Ishibuchi, S. Mostaghim, Y. Nojima, "A Framework for Large-Scale Multiobjective Optimization Based on Problem Transformation," *IEEE Trans. Evol. Comput*., vol. 22, no. 2, pp. 260–275, 2018.

[44] S. Liu, Q. Lin, K. Wong, Q. Li, K. C. Tan. "Evolutionary Large-Scale Multiobjective Optimization: Benchmarks and Algorithms," *IEEE Trans. Evol. Comput*., in press, 2021.

[45] J. Yi, L. Xing, et al., "Behaviour of Crossover Operators in NSGA-III for Large-Scale Optimization Problems," *Information Sciences*, vol. 509, pp. 470-487, 2020.

[46] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Non-Dominated Sorting Approach, Part I: Solving Problems with Box Constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[47] Y. Tian, X. Zhang, C. Wang, Y. Jin, "An Evolutionary Algorithm for Large-Scale Sparse Multiobjective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380-393, 2020.

[48] Y. Zhang, et al., "Enhancing MOEA/D with Information Feedback Models for Large-Scale Many-Objective Optimization," *Information Sciences*, vol. 522, pp. 1-16, 2020

[49] S.Liu, Q. Lin, Y. Tian, K. C. Tan. "A Variable Importance-Based Differential Evolution for Large-Scale Multiobjective Optimization," *IEEE Trans. Cybern*., in press, 2021

[50] X. Wang, K. Zhang, J. Wang, and Y. Jin, "An Enhanced Competitive Swarm Optimizer with Strongly Convex Sparse Operator for Large-Scale Multi-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, in press, 2021

[51] X. Li, X. Yao, "Cooperatively Coevolving Particle Swarms for Large Scale Optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210-224, 2012.

[52] Q. Yang, W. Chen, et al., "Segment-Based Predominant Learning Swarm Optimizer for Large-Scale Optimization," *IEEE Trans. Cybern*., vol. 47, no. 9, pp. 2896-2910, 2017.

[53] Q. Yang, W. Chen, et al., "A Level-Based Learning Swarm Optimizer for Large-Scale Optimization," *IEEE Trans. Evol. Comput*., vol. 22, no. 4, pp. 578-594, 2018.

[54] H. Deng, L. Peng, H. Zhang, B. Yang, Z. Chen, "Ranking-Based Biased Learning Swarm Optimizer for Large-Scale Optimization," *Information Sciences*, vol. 493, pp. 120-137, 2019.

[55] Q. Yang, W. Chen, et al., "A Distributed Swarm Optimizer With Adaptive Communication for Large-Scale Optimization," *IEEE Trans. Cybern*., vol. 50, no. 7, pp. 3393-3408, 2020.

[56] R. Lan, L. Zhang, Z. Tang, Z. Liu, X. Luo, "A Hierarchical Sorting Swarm Optimizer for Large-Scale Optimization," *IEEE Access*, vol. 7, pp. 40625-40635, 2019.

[57] X. Zhang, X. Zheng, R. Cheng, J. Qiu, Y. Jin, "A Competitive Mechanism Based Multi-Objective Particle Swarm Optimizer with Fast Convergence," *Information Sciences*, vol. 427, pp. 63-76, 2018.

[58] S. Liu, Q. Yu, Q. Lin, and K. C. Tan, "An Adaptive Clustering-based Evolutionary Algorithm for Many-objective Optimization Problems," *Information Sciences*, vol. 537, pp. 261-283, 2020.

[59] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Syst*., vol. 9, no. 4, pp. 115–148, 1995.

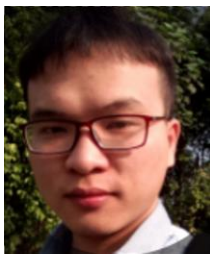
[60] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A New Dominance Relation Based Evolutionary Algorithm for Many-Objective Optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16-37, 2016.

[61] Z. N. He, G. G. Yen, "Many-Objective Evolutionary Algorithm Based on Coordinated Selection Strategy," IEEE Trans. Evol. Comput., vol. 21, no. 2, pp. 220-233, 2017.

[62] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-objective Optimization," IEEE Trans. Evol. Comput., vol. 20, no. 5, pp. 773-791, 2016.

[63] M. Li, S. Yang, and X. Liu, "Shift-Based Density Estimation for Paretobased Algorithms in Many-Objective Optimization," *IEEE Trans. Evol. Comput*., vol. 18, no. 3, pp. 348–365, 2014.

[64] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test Problems for Largescale Multiobjective and Many-objective Optimization," *IEEE Trans. Cybern*., vol. 47, no. 12, pp. 4108–4121, 2017.
[65] P. A. N. Bosman and D. Thierens, "The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms," *IEEE Trans. Evol. Comput*., vol. 7, no. 2, pp. 174–188, 2003.
[66] A. J. Nebro et al., "SMPSO: A New PSO-Based Metaheuristic for Multi-Objective Optimization," *in Proc. IEEE Symp. Comput. Intell. Multi Criteria Decis*. Making, Nashville, TN, USA, pp. 66–73, 2009.
[67] J. J. Durillo and A. J. Nebro, "jMetal: A Java Framework for Multi-objective Optimization," Adv. Eng. Softw., vol. 42, no. 10, pp. 760–771, Oct. 2011.
[68] 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.



Songbai Liu (Member IEEE) received the B.S. degree from Changsha University and the M.S. degree from Shenzhen University, China, in 2012 and 2018, respectively. He worked for ShenZhen TVT Digital Technology Co., Ltd as a software engineer from 2013 to 2015, and he worked for Shenzhen University as a research assistance from 2018 to 2019.

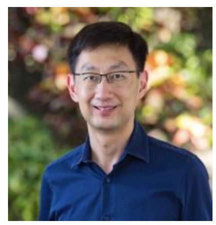
He is currently a PhD student in Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include nature-inspired computation, evolutionary transfer optimization, and evolutionary large-scale optimization.

Qiuzhen Lin (Member IEEE) received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over sixty research papers since 2008. His current research interests include artificial immune system, multi-objective optimization, and dynamic system.

Qing Li (Fellow, IEEE) received the B.Eng. degree in computer science from Hunan University, Changsha, China, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, CA, USA.

Prof. Li has been a chair professor of data science, the Hong Kong Polytechnic University since December 2018. Prior to that, he taught at the City University of Hong Kong, the Hong Kong University of Science and Technology, and Australian National University. He is a Fellow of IEEE, IET/IEE, and a distinguished member member of the CCF (Chinese Computer Federation). In addition, he is the Chairperson of the Hong Kong Web Society, Vice Chairman of the IEEE Computational Intelligence Society Hong Kong Chapter, and also served/is serving as an executive committee (EXCO) member of the IEEE-Hong Kong Computer Chapter and ACM Hong Kong Chapter.



Kay Chen Tan (Fellow, IEEE) received the B.Eng. degree (First Class Hons.) and the Ph.D. degree from the University of Glasgow, U.K., in 1994 and 1997, respectively.

He is currently a Chair Professor (Computational Intelligence) of the Department of Computing, the Hong Kong Polytechnic University. He has published over 300 refereed articles and seven books.

Prof. Tan is currently the Vice-President (Publications) of IEEE Computational Intelligence Society, USA. He served as the Editor-in-Chief of the IEEE Computational Intelligence Magazine from 2010 to 2013 and the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2015 to 2020. He currently serves as an Editorial Board Member for more than ten journals. Prof. Tan is an IEEE Distinguished Lecturer Program (DLP) Speaker and the Chief Co-Editor of Springer Book Series on Machine Learning: Foundations, Methodologies, and Applications.