

A Two-Phase Learning-Based Swarm Optimizer for Large-Scale Optimization

Rushi Lan¹, Yu Zhu, Huimin Lu², Zhenbing Liu, and Xiaonan Luo³

Abstract—In this article, a simple yet effective method, called a two-phase learning-based swarm optimizer (TPLSO), is proposed for large-scale optimization. Inspired by the cooperative learning behavior in human society, mass learning and elite learning are involved in TPLSO. In the mass learning phase, TPLSO randomly selects three particles to form a study group and then adopts a competitive mechanism to update the members of the study group. Then, we sort all of the particles in the swarm and pick out the elite particles that have better fitness values. In the elite learning phase, the elite particles learn from each other to further search for more promising areas. The theoretical analysis of TPLSO exploration and exploitation abilities is performed and compared with several popular particle swarm optimizers. Comparative experiments on two widely used large-scale benchmark datasets demonstrate that the proposed TPLSO achieves better performance on diverse large-scale problems than several state-of-the-art algorithms.

Index Terms—Competitive swarm optimizer (CSO), cooperative coevolution, large-scale optimization, swarm intelligence.

I. INTRODUCTION

THE EVOLUTIONARY algorithm (EA) is a group-oriented random search technique that simulates the evolution of organisms in nature [1]. Compared with traditional optimization algorithms, such as the calculus-based methods, EA can be free from the natural problems and effectively addresses some complicated practical problems. Therefore, an increasing number of research interests focus on this area [2], [3], [34], [39], [42].

Various EAs have been proposed to solve a wide range of optimization problems so far, such as resource

allocation [10], [41]; image processing [11]; network planning [9]; and many others [8], [15], [16]. Specifically, the particle swarm optimization (PSO), first introduced by Kennedy and Eberhart in 1995 [7], has attracted numerous attention for decades. However, researchers found that PSO easily leads to the premature convergence; therefore, it performs poorly when encountering the complicated multimodal problems [25].

Inspired by the nature and human society, many researchers have proposed diverse improvements to PSO, such as PSO with an aging leader and challengers (ALC-PSO) [4] and orthogonal learning PSO (OLPSO) [40], to improve the performance of PSO in dealing with complicated optimization problems. However, these PSO variants are only effective in the low-dimensional space. When dealing with high-dimensional optimization problems, the aforementioned algorithms dramatically deteriorate the result. This phenomenon is called “the curse of dimensionality” [23]. As the dimension size increases, the search spaces and the number of local optimal traps increase exponentially [5], [6]. These phenomena are the initiators of premature convergence. Unfortunately, in recent years, most engineering problems indicate that the number of decision variables increases exponentially [24], such as marine underwater signal processing [13], salient-object detection [12], and the training of deep-learning models [17].

Enlightened by human society, some novel learning strategies for PSO have been developed for large-scale optimization. Cheng and Jin proposed a competitive swarm optimizer (CSO) [5] and a social learning particle swarm optimizer (SL-PSO) [6]. These two methods adopt one predominant particle instead of pbest or gbest to update the particles. Similar algorithms are a level-based learning swarm optimizer (LLSO) [35] and segment-based predominant learning swarm optimizer (SPLSO) [36]. These PSO variants largely alleviate the problem of premature convergence because they can provide higher diversity for the swarm than the earlier PSO variants. However, the premature convergence is still the main challenge for large-scale optimization.

To further alleviate the problem of premature convergence, a two-phase learning-based swarm optimizer (TPLSO) is proposed in this article. In TPLSO, inspired by the learning behavior of human society, each evolution is divided into two phases, namely, a mass learning phase and an elite learning phase. In the mass learning phase, particles with different potential in exploration and exploitation are randomly selected to form a study group, and then a competition strategy is adopted among group members to yield the winner and the loser ones. After that, the winner is the leader to guide the

Manuscript received July 10, 2019; revised October 29, 2019; accepted January 6, 2020. Date of publication March 9, 2020; date of current version December 22, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61702129, Grant 61772149, Grant 61866009, and Grant U1701267, in part by the National Key Research and Development Program of China under Grant 2018AAA0100305, and in part by the Guangxi Science and Technology Project under Grant 2019GXNSFAA245014, Grant AD18281079, Grant AD18216004, Grant 2017GXNFDA198025, and Grant AA18118039. This article was recommended by Associate Editor Y. Zhang. (Corresponding author: Huimin Lu.)

Rushi Lan, Yu Zhu, and Zhenbing Liu are with the Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China.

Huimin Lu is with the Department of Mechanical and Control of Engineering, Kyushu Institute of Technology, Kitakyushu 8048550, Japan (e-mail: dr.huimin.lu@ieee.org).

Xiaonan Luo is with the National Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCYB.2020.2968400>.

Digital Object Identifier 10.1109/TCYB.2020.2968400

losers. Through this learning strategy, the high diversity can be preserved. During the elite learning phase, some elite particles with good fitness values are picked out by sorting the particles in the population to form a new swarm, and then the elite particles update their positions by learning two better particles in this swarm. It is worth noting that the two better particles mentioned above are randomly selected. In this phase, the learning among elite particles accelerates the convergence, so promising areas can be located as fast as possible. The comparative results with some state-of-the-art algorithms demonstrate the effectiveness of the proposed method.

The remainder of this article is organized as follows. Section II briefly reviews some related works on large-scale optimization. The details of TPLSO are presented in Section III. A set of experiments is conducted in Section IV to verify the performance of TPLSO. Finally, Section V concludes this article.

II. RELATED WORKS

The problem studied in this article is to minimize $f(X)$, where $X = [x^1, x^2, \dots, x^D]$, D is the dimension of the optimization problems, and x^i is the value of the i th dimension. In recent years, researchers have proposed many methods to solve the high-dimensionality problems, and these methods can be roughly divided into two categories, that is, cooperatively coevolutionary algorithms (CCEAs) and novel update strategies for classic PSOs [33], [40].

A. Cooperative Coevolutionary Algorithms

The first CCEA, the cooperative coevolutionary genetic algorithm (CCGA), which decomposes a high-dimensional problem into several lower dimensional subproblems, was proposed by Potter [27]. The performance of CCEA heavily depends on its grouping strategy; thus, researchers focus on designing the grouping methods. Yang *et al.* [37] proposed a random grouping strategy, which randomly decomposes a D -dimensional problem into m S -dimensional subproblems in each iteration. It is worth noting that S represents the dimension of subproblems and $S \ll D$. Combining the differential evolution (DE) [30] algorithm with this method, namely, DECC-G [37], performs well on some 1000-D problems. Yang *et al.* [38] dynamically changed the value of S during each evolutionary cycle and proposed a multilevel CC algorithm, namely, multilevel cooperative coevolution (MLCC). On this basis, the cooperatively coevolving particle swarm optimizer (CCPSO2) [19] was proposed.

A new grouping strategy that detects the interdependent decision variables, called differential grouping (DG), was proposed by Omidvar *et al.* [23]. DG achieves satisfactory performance when detecting interdependent variables. Based on this, its variants, such as DG2 [24], XDG [31], and GDG [21], have been further proposed to detect more complex structures in optimization problems.

B. Novel Learning Strategies for PSO Variants

In PSO, imitating the foraging behavior of a flock of birds, particles in a swarm are optimized by searching the entire

solution space to find a globally best solution. In this algorithm, $pbest_i$ and $gbest$ are utilized to guide the learning of all particles, which easily leads to premature convergence. $pbest_i$ is the personal best position of the i th particle in current generation and $gbest$ is the best position found in the particle swarm so far. In contrast to the traditional PSOs, inspired by the competition of human society, Cheng and Jin proposed a CSO [5], which adopts one predominant particle instead of $pbest_i$ and $gbest$. In detail, two particles are randomly selected from the swarm for comparison. Then, the loser is updated by learning from the winner, which goes directly to the next generation. In CSO, only the loser is updated in the following way:

$$V_l(t+1) = r_1(t)V_l(t) + r_2(t)(\bar{X}_w(t) - X_l(t)) + \psi r_3(t)(\bar{X}(t) - X_l(t)) \quad (1)$$

$$X_l(t+1) = X_l(t) + V_l(t+1) \quad (2)$$

where $X_l(t)$ and $V_l(t)$ are the position and velocity of the loser in the t -th generation. r_i ($i = 1, 2, 3$) is the random variable within $[0, 1]$, and ψ is the control parameter of $\bar{X}(t)$.

Algorithms similar to CSO are SL-PSO [6] and LLSO [35]. Observing these algorithms, we can find that the update or learning methods of these optimizers greatly increase the diversity of the swarm, thus, premature convergence may be avoided. Despite the large amount of work involved in dealing with large-scale optimization, premature convergence and falling into local optima are still the main challenges in large-scale optimization.

III. TWO-PHASE LEARNING-BASED SWARM OPTIMIZER

A. Motivation

Studying CSO, LLSO, and SL-PSO, we find that this type of optimizer does not have a good balance between diversity and convergence. CSO emphasizes the diversity and ignores the convergence. In contrast, LLSO and SL-PSO focus on convergence abilities. Therefore, these algorithms can be further improved. To optimize high-dimensional problems more effectively, we seek inspiration from nature and human society. In education, different students usually have different potentials or learning abilities, and their abilities can be quickly improved by learning from each other. In particular, the group cooperative learning model is widely used in educational practice [14], [29]. Similarly, in swarms, different particles have different exploration and exploitation abilities in traversing the objective space. Thus, they should be treated differently. Enlightened by this, a mass learning strategy is proposed in TPLSO, which randomly selects several particles from the swarm to form a group, and then a competition mechanism is adopted to update the members. In real life, the team's strength is often further enhanced by improving the ability of the team's leaders because smart leaders can better guide the team. Thus, an elite learning method is introduced to improve the fitness value of elite particles, which have better fitness values than others in the swarm. In this phase, elite particles are selected, and then the elite particles are updated by learning from any two particles better than themselves. Inspired by the above observation and phenomenon, a reinforced competition-based learning strategy for PSO is proposed, namely, the TPLSO.

B. TPLSO

In TPLSO, the entire evolution process is divided into two parts, namely, mass learning and elite learning.

1) *Mass Learning*: During this stage, assume the number of particles in the swarm is NP , and the NP particles are further divided into NP/K study groups according to K particles as a group. For each group, different particles have different abilities in exploring and exploiting the search space, and these particles update their positions through the collaboration and competitive strategies. Specifically, the particles compete with each other to determine their ranking in the group; the winner is represented by W , and the inferior and worst particles are represented by L_1 and L_2 , respectively. Then, L_1 updates its position by learning from W , as in CSO. For L_2 , both W and L_1 are used to update its state. The winner passes directly to the elite learning phase, while the losers update their position and velocity using the following strategy:

$$\begin{cases} V_{L_2} = R_1 V_{L_2} + R_2 (X_W - X_{L_2}) + \varphi R_3 (X_{L_1} - X_{L_2}) & (3a) \\ V_{L_1} = R_1 V_{L_1} + R_2 (X_W - X_{L_1}) + \varphi R_3 (\bar{X} - X_{L_1}) & (3b) \\ X_{L_i} = X_{L_i} + V_{L_i}; \quad i = 1, 2 & (3c) \end{cases}$$

where X_W , X_{L_i} , V_W , and V_{L_i} represent the position and velocity of the winner and loser of the competition in each group. It is worth noting that L_1 is the winner compared with L_2 . R_1 , R_2 , and R_3 are three random variables with a range of $[0, 1]$. φ is a parameter within $[0, 1]$ that controls the influence of X_{L_1} or \bar{X} . \bar{X} is the mean position of each group or mean position value of the swarm. For \bar{X} , a global version and a local version can be adopted in this article.

1) \bar{X}_g is the mean position of all particles in each evolution.

2) \bar{X}_i is the mean position of each group in each generation.

2) *Elite Learning*: During the elite learning phase, particles in the swarm are sorted in an ascending order of the fitness value. Afterward, the first N particles of this swarm are selected to form a new swarm, denoted by P_h , and the remaining particles are passed directly to the next generation of the swarm. In addition, the swarm size of P_h is set as $NP/2$, in which NP is the original swarm size. Observing the particles in P_h , we can see that these N particles are the elite ones, which have better fitness values than the remaining particles in the original swarm. Consequently, these particles usually possess more beneficial information to guide other particle learning and are more likely to be close to the global optimum area. In P_h , the particle (j) randomly selects two better particles (r_1 and r_2) from the current swarm, and then updates its position and velocity via

$$V_j = R_1 V_j + R_2 (X_{r_1} - X_j) + \varphi R_3 (X_{r_2} - X_j) \quad (4)$$

$$X_j = X_j + V_j \quad (5)$$

where j , r_1 , $r_2 \in [1, N]$ represent three particle indices, respectively. Specifically, r_1 and r_2 are the indices of two randomly selected exemplars. The other parameters are set as previously. Note that $r_1 < r_2 < j$ shows that j is lower than r_2 , and they both are lower than r_1 . Thus, X_j is worse than X_{r_2} , and both are worse than X_{r_1} . Mutual learning between the elite particles further improves their fitness values so that fast convergence can be achieved.

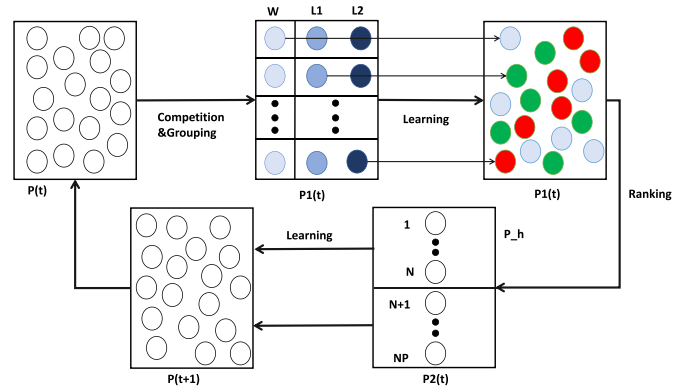


Fig. 1. General idea of TPLSO. The entire evolution process is divided into two phases: mass learning and elite learning. The upper part of this graph is the mass learning process, and the lower part is the elite learning process. In the mass learning phase, three particles are randomly selected from $P(t)$ for competitions. The losers, whose fitness is worse will be updated by learning from the winner, and the winner is directly passed to the elite learning phase. Afterward, particles in $P_1(t)$ are sorted in ascending order of fitness value. Then, in the elite learning phase, the first N particles are selected to form a new swarm P_h . In P_h , the particle updates its position by learning from two randomly selected particles with a better fitness value.

Algorithm 1 Pseudocode of TPLSO. NP and N Are the Sizes of Swarms P and P_h , Respectively. K Is Each Group Size. The Terminal Condition Is the Maximum Number of Fitness Evaluations

```

1: randomly initialize  $P(0)$ ;
2: while the terminal condition is not satisfied do
3:   //the first phase: Name the swarm at this phase as  $P_1$ 
4:   Calculate the fitness of all particles in  $P_1$ ;
5:   Divide all particles into  $NP/K$  groups;
6:   Identify winner ( $W$ ) and losers ( $L_1, L_2$ ) based on competitive mechanisms and  $f(X_W) < f(X_{L_1}) < f(X_{L_2})$ ;
7:   Update  $X_{L_1}$  according to Eq. (3b) and Eq. (3c);
8:   Update  $X_{L_2}$  using Eq. (3a) and Eq. (3c);
9:   Add  $X_W, \bar{X}_{L_1}$  and  $X_{L_2}$  into the next phase;
10:  //the second phase: Name the swarm at this phase as  $P_2$ 
11:  Sort particles in  $P_2$  in ascending order of fitness value;
12:   $N$  optimal particles in  $P_2$  are selected from  $P_1$  to form a new swarm, namely,  $P_h$ ;
13:  for  $j = 3, \dots, N$  do
14:    Select two particles from  $[1, j - 1]$  levels:  $r_1, r_2$ ;
15:    if  $r_1 > r_2$  then
16:      Swap( $r_1, r_2$ );
17:    end if
18:    Update particle  $X_j$  using Eq. (4) and Eq. (5);
19:  end for
20: end while

```

Combining the above two learning phases, the framework of TPLSO is displayed in Fig. 1. The pseudocode of TPLSO is also summarized in Algorithm 1.

C. Theoretical Analysis

Exploration and exploitation play important roles when the particles traverse the search space. A good optimizer should compromise between these two aspects. Here, we investigate the exploration and exploitation capacities of TPLSO by comparing with LLSO [35] and global PSO (GPSO) [7].

1) *Exploration Ability*: For EA, improving its exploration ability is equivalent to promoting the swarm diversity so that it can avoid falling into the local areas and finding more promising areas. Exploration is particularly important for the

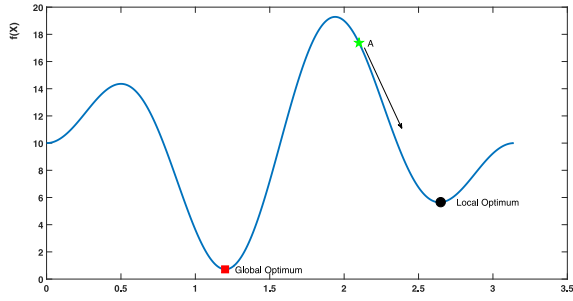


Fig. 2. Illustration of the search dynamics of LLSO on a multimodal problem. In this case, point A represents the start position of the particle, and the particles eventually fall into a local optimum.

high-dimensional and multimodal problems with a large number of local traps. The observation of (3a), (3b), and (4) indicates that the learning samples of the particles in (3a) and (4) are two different individuals, and the samples learned from the particles in (3b) are the winner and the mean value of the corresponding groups. According to the theoretical analysis in LLSO [35], regarding (3a) and (4), it is easy to conclude that TPLSO has the same excellent exploration ability as LLSO and has higher diversity than classic PSOs, which utilize $pbest_i$ and $gbest$ to guide particle learning such as GPSO. To further investigate the exploration ability of TPLSO, \bar{X} is the mean value of each group in this section, and (3b) was rewritten as follows:

$$\begin{cases} V_{L_1} = R_1 V_{L_1} + p_1 (X_W - X_{L_1}) + p_2 (X_{L_1} - X_{L_2}) & (6a) \\ p_1 = R_2 + \frac{\varphi R_3}{3}, \quad p_2 = -\frac{\varphi R_3}{3} & (6b) \end{cases}$$

where R_1 , R_2 , R_3 , and φ are positive numbers; thus, we have $p_1 > 0$ and $p_2 < 0$.

Similarly, we can also write the update formula of LLSO as follows:

$$V_{i,j} = R_1 V_{i,j} + R_2 (X_{r_{l_1,k_1}} - X_{i,j}) + \varphi R_3 (X_{r_{l_2,k_2}} - X_{i,j}) \quad (7)$$

$$X_{i,j} = X_{i,j} + V_{i,j}. \quad (8)$$

According to the derivation of LLSO [35], the first term on the right-hand side of (7) is a positive number, and the other two terms are negative numbers. Therefore, the values of $V_{i,j}$ and $f(X_{i,j})$ continue to decrease so that the particles can constantly search for more promising areas. Unlike (6a), the second term on the right-hand side of (6a) is negative, and the other two are positive, and the value of V_{L_1} may increase. Thus, particles may search for poor areas. Compared with LLSO, this update strategy may lead to worse performance, but the truth may be the opposite. Let us consider a situation where A is the start position of the particle, as illustrated in Fig. 2. According to the update method of particles in LLSO, as shown in (7), the particles move toward a more promising area in each iteration. In Fig. 2, the particles approach the local optimal area and eventually lead to premature convergence. However, in (6a), the third term on the right is a positive number, and its sum with the second term may also be positive. Then, the values of V_{L_1} and $f(X_{L_1})$ increase, indicating that the next position of the particle may be worse than the current position, as shown in Fig. 3. In Fig. 3, the particle moves from positions A to B

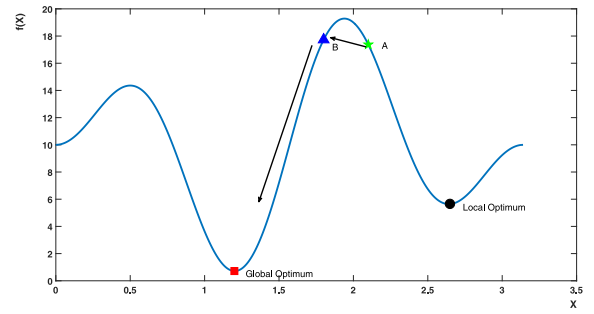


Fig. 3. Illustration of the search dynamics of TPLSO on a multimodal problem. A is the start position of the particle, B is the particle position in the next generation, and global optima can be obtained.

and finally finds the global optimal area. Therefore, we can infer that the diversity of TPLSO may be better than LLSO. Thus, TPLSO has a better ability to jump out of the local optimal areas and can find the more promising areas faster. Furthermore, if \bar{X} is the mean position of the swarm, the result of exploration ability is the same as before.

2) *Exploitation Ability*: Exploitation ability and exploration ability are equally important for EAs because the good exploitation ability of an EA can quickly locate better areas so that the optimization time can be reduced, which is important to some problems with limited computational resources and unimodal problems. To analyze the exploitation ability of TPLSO, two particles X_W and X_{L_2} are selected from one group in the mass learning phase. Then, we have

$$f(X_W) \leq f(X_{L_2}). \quad (9)$$

Combining $pbest$ and $gbest$ defined in classic PSO, the following relationship can be obtained:

$$\begin{cases} f(gbest) \leq f(pbest_W) \leq f(X_W) \\ f(gbest) \leq f(pbest_{L_2}) \leq f(X_{L_2}). \end{cases} \quad (10)$$

Let

$$\Delta F_{TPLSO} = |f(X_{L_2}) - f(X_W)| \quad (11)$$

$$\Delta F_{LLSO} = |f(X_{L_2}) - f(X_{r_{l_1,k_1}})| \quad (12)$$

where $X_{r_{l_1,k_1}}$ is selected from a higher level than X_{L_2} , and $X_{r_{l_1,k_1}}$ is better than all particles below its level. X_W is only better than X_{L_1} and X_{L_2} . Thus, we have $f(X_{r_{l_1,k_1}}) \leq f(X_W)$. Similarly, for the canonical PSO such as GPSO, it has

$$\Delta F_{GPSO} = |f(X_{L_2}) - f(gbest)|. \quad (13)$$

According to the definition of $X_{r_{l_1,k_1}}$, we achieve the following relationship:

$$f(gbest) \leq f(X_{r_{l_1,k_1}}). \quad (14)$$

Combining the above formulas, the following relationship can be derived:

$$\Delta F_{TPLSO} \leq \Delta F_{LLSO} \leq \Delta F_{GPSO}. \quad (15)$$

The above formula shows that TPLSO, compared with LLSO and GPSO, has a better ability to exploit the small gaps between two positions whose fitness values are very similar.

D. Time Complexity Study

Generally, the time complexity is calculated by analyzing the extra time in each iteration without the fitness evaluations [5]. According to LLSO [35], LLSO takes $O(NP + NP \log(NP)) + O(NP * D)$ in each generation. From Algorithm 1, we can see that it takes $O(NP) + O([(K-1) * NP]/K * D)$ in the mass learning phase, and for the elite learning phase, it takes $O(NP \log(NP)) + O(N * D)$. Therefore, the computation of TPLSO is $O(NP + NP \log(NP)) + O([(K-1) * NP]/K + N) * D$, where NP and D are the swarm size and dimension, respectively. K is the group size in the mass learning phase, and N is the swarm size of P_h . In this article, N and K are set as $(NP/2)$ and 3, respectively. Therefore, the time complexity of TPLSO is $O(NP + NP \log(NP)) + O([(7 * NP)/6] * D)$. In conclusion, compared to LLSO, TPLSO will take more time in each generation.

IV. EXPERIMENTS

In this section, a suite of experiments is conducted to study the performance of TPLSO from different perspectives. Two benchmark sets, namely, CEC'2010 [32] and CEC'2013 [18], are selected here. The CEC'2010 benchmark set consists of separable functions, partially separable functions, and non-separable functions. Based on CEC'2010, the CEC'2013 benchmark set introduced some new characteristics, such as the imbalance contribution of subcomponents and overlapping functions. Consequently, the functions in CEC'2013 are much more complicated and harder to optimize than those of the CEC'2010. In this article, unless otherwise specified, all statistical results are over 30 independent runs. The maximum of fitness evaluations for each independent run is set to $3000 \times D$, where D is the dimension size of the optimization problems [35].

A. Parameter Study

In TPLSO, three additional parameters are introduced, namely, the group size K , the swarm size NP , and the control parameter φ . First, the setting of K here has a crucial impact on the algorithm. Thus, to accurately set the study group size (K), several experiments are carried out on the above functions with $K = 2, 3, 4$, and the statistical results are shown in Table I. In addition, φ , NP , and D here are set as 0.15, 600, and 1000, respectively. From Table I, we can find that $K = 3$ is the most reasonable setting.

Second, the particles generally tend to exhibit premature convergence with a small swarm size (NP). This is because a small swarm size cannot provide high diversity for the swarm. In contrast, if NP is set to a large number, more computing resources are needed during each generation, which is impractical for computationally expensive problems. According to CSO [5], we can find that there exist a correlation between NP and φ . Thus, to properly set these two parameters, some experiments are conducted on TPLSO with NP varying from 400 to 800 and φ varying from 0.05 to 0.25.

Table II shows the statistical results of TPLSO with differential combinations of these two parameters on eight CEC'2010

TABLE I

STATISTICAL RESULT (MEAN VALUE IS ON THE FIRST LINE AND STD VALUE IS ON THE SECOND LINE) OF OPTIMIZATION ERROR OBTAINED BY TPLSO ON SEVERAL TEST FUNCTIONS OF 1000-D WITH THE SWARM SIZE $NP = 600$, $\varphi = 0.15$, AND K VARYING FROM 2 TO 4

Functions	$K = 2$	$K = 3$	$K = 4$
f_1	5.79E-19	2.87E-19	1.53E-17
	3.22E-20	1.63E-20	7.89E-19
f_2	8.42E+00	6.17E+02	8.69E+02
	5.69E+01	1.94E+01	6.27E+01
f_6	3.43E-07	1.83E-07	6.43E-07
	3.56E-08	7.04E-09	2.79E-08
f_7	1.81E-01	1.73E-04	4.44E+00
	1.72E-02	1.74e-04	2.31E+00
f_{11}	1.71E-12	1.25E-12	9.75E-12
	5.68E-14	2.68E-14	1.23E-13
f_{12}	8.73E+03	4.57E+03	7.59E+03
	7.45E+02	5.86E+02	4.50E+02
f_{16}	6.78E-01	1.61E-12	1.20E+00
	6.03E-01	1.44E-14	1.23E+00
f_{17}	8.08E+04	5.31E+04	9.15E+04
	3.67E+03	4.21E+03	4.33E+03

functions $f_1, f_2, f_6, f_7, f_{11}, f_{12}, f_{16}$, and f_{17} , where f_1 and f_2 are separable functions, and the remaining ones are partially separable functions. Moreover, f_1, f_7, f_{12} , and f_{17} are unimodal functions, while f_2, f_6, f_{11} , and f_{16} are multimodal functions. These functions contain almost all types of problems: fully and partially separable, and unimodal and multimodal. Generally, partially separable functions are more difficult to optimize than fully separable functions, and multimodal functions are more difficult to optimize compared with unimodal functions. Multimodal and partially separable functions are closer to real-world optimization problems, so more attention should be focused on these types of functions. From Table II, we can see that the unimodal functions perform better when $NP = 400$, while the advantage of the multimodal functions concentrates on larger swarm sizes, such as $NP = 600$. This is because the small swarm size cannot provide a high diversity for the swarm.

In summary, $NP = 600$, $\varphi = 0.15$, and $K = 3$ are used for TPLSO on 1000-D. Specifically, φ is set to 0.2 for non-separable functions (such as f_{19} and f_{20} in CEC'2010 function sets, and f_{13} - f_{15} in CEC'2013 function sets) for TPLSO on 1000-D because higher diversity is required when optimizing nonseparable functions.

B. Exploration and Exploitation Influence of EAs

Exploration ability is a crucial factor in addressing the multimodal problems because a good exploration ability can increase the probability of particles to jump out of the local optimum. Exploitation ability can affect the convergence speed of particles; thus, exploitation should be properly biased to seek fast convergence when dealing with unimodal problems [26], [28]. However, exploitation and exploration are often conflicting, so a good EA should balance between these two aspects. To verify that TPLSO is able to compromise these two abilities properly, a set of experiments is conducted on four CEC'2010 test functions and compared with CSO and GPSO. The used functions are f_3 (fully separable and

TABLE II
MEAN VALUES (FIRST LINE) AND STD VALUE (SECOND LINE) OF OPTIMIZATION ERRORS OBTAINED BY TPLSO ON SEVERAL FUNCTIONS IN CEC'2010 TEST FUNCTIONS OF 1000-D WITH THE SWARM SIZE NP VARYING FROM 400 TO 800, $K = 3$ AND φ FROM 0.05 TO 0.25

NP	Functions	$\varphi = 0.05$	$\varphi = 0.10$	$\varphi = 0.15$	$\varphi = 0.20$	$\varphi = 0.25$
$NP=400$	f_1	4.36E-20(3.08E-21)	1.35E-19(8.05E-21)	2.71E-19(2.49E-20)	8.05E-19(1.87E-19)	1.47E-18(1.23E-19)
	f_2	9.07E+02(6.83E+01)	9.94E+02(6.67E+01)	1.15E+03(5.61E+01)	1.60E+03(3.95E+01)	1.79E+03(9.71E+01)
	f_6	4.39E-08(2.29E-09)	9.16E-08(9.97E-09)	2.20E-01(4.90E-01)	1.85E+01(4.10E-01)	1.96E+01(1.00E-02)
	f_7	5.00E-02(3.00E-02)	1.83E-05(1.19E-05)	1.73E-07(2.03E-07)	2.91E-08(3.92E-08)	3.94E+04(2.36E+04)
	f_{11}	3.68E-01(6.38E-01)	6.50E-01(1.12E+00)	7.14E+00(6.75E+00)	1.97E+01(6.30E-01)	2.10E+01(4.40E-01)
	f_{12}	2.01E+04(9.91E+02)	2.56E+03(8.09E+01)	9.71E+02(9.02E+01)	9.94E+02(7.19E+01)	1.00E+04(2.81E+03)
	f_{16}	1.61E+00(6.71E-01)	2.21E+00(1.29E+00)	9.12E+00(4.35E+00)	3.31E+01(1.64E+01)	7.21E+01(2.33E+01)
	f_{17}	2.49E+05(1.11E+04)	4.42E+04(1.69E+03)	1.28E+04(6.95E+02)	1.19E+04(5.40E+02)	5.45E+04(9.17E+03)
$NP=600$	f_1	3.29E-19(1.01E-20)	1.40E-19(1.01E-20)	2.87E-19(1.63E-20)	5.35E-19(1.87E-20)	8.36E-19(4.63E-20)
	f_2	2.70E+03(7.57E+02)	6.00E+02(5.25E+01)	6.17E+02(1.94E+01)	1.05E+03(4.03E+01)	1.22E+03(1.98E+01)
	f_6	8.16E-08(3.46E-09)	1.10E-07(4.52E-09)	1.83E-07(7.04E-09)	2.82E-07(9.69E-09)	1.38E+00(3.40E-01)
	f_7	4.17E+01(9.66E+00)	8.60E-02(4.60E-02)	1.73E-04(1.74E-04)	9.59E-06(1.22E-06)	2.18E-05(2.20E-05)
	f_{11}	7.31E-12(4.92E-13)	8.69E-13(3.35E-14)	1.25E-12(2.68E-14)	1.63E-12(5.23E-14)	1.79E+01(2.90E-01)
	f_{12}	2.68E+05(3.40E+04)	2.13E+04(1.95E+03)	4.57E+03(5.86E+02)	2.63E+03(3.65E+02)	3.13E+03(2.95E+01)
	f_{16}	5.15E-12(3.90E-13)	1.98E-12(2.13E-14)	1.61E-12(1.44E-14)	1.87E+00(1.67E+00)	4.22E+00(3.88E+00)
	f_{17}	1.68E+05(1.30E+05)	3.67E+04(9.00E+02)	5.31E+04(4.21E+03)	2.44E+04(7.53E+02)	2.76E+04(1.86E+03)
$NP=800$	f_1	7.59E-11(9.63E-12)	2.73E-19(5.33E-21)	3.82E-19(1.98E-20)	6.19E-19(2.35E-20)	9.00E-19(3.26E-20)
	f_2	9.47E+03(1.37E+02)	4.63E+02(3.05E+01)	5.26E+02(2.52E+01)	7.66E+02(2.82E+01)	8.95E+02(6.67E+01)
	f_6	2.59E-07(2.50E-09)	1.18E-07(2.09E-09)	1.49E-07(3.72E-09)	2.13E-07(1.20E-08)	2.76E-07(3.52E-09)
	f_7	8.27E+01(2.66E+02)	6.02E+00(2.81E+00)	3.50E-02(1.20E-02)	1.40E-03(2.09E-04)	9.96E-04(2.43E-04)
	f_{11}	1.39E-07(2.02E-08)	1.11E-12(3.59E-14)	1.23E-12(3.42E-14)	1.56E-12(4.27E-14)	1.89E-12(6.41E-14)
	f_{12}	1.43E+06(1.06E+05)	7.68E+04(1.96E+03)	1.48E+04(1.54E+03)	6.39E+03(6.76E+02)	5.96E+03(5.78E+02)
	f_{16}	2.95E-07(3.28E-08)	1.86E-12(4.35E-14)	1.99E-12(4.73E-14)	2.46E-12(3.77E-14)	2.88E-12(8.12E-14)
	f_{17}	4.13E+05(3.49E+06)	1.42E+05(3.07E+03)	1.46E+05(1.77E+03)	4.99E+04(2.27E+03)	4.03E+04(2.43E+03)

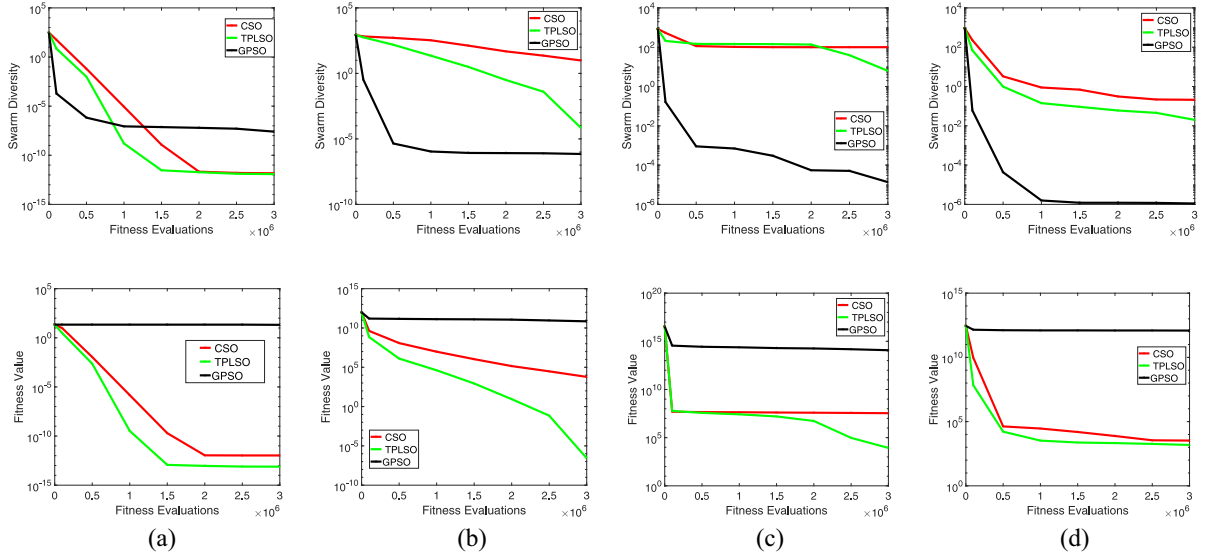


Fig. 4. Swarm diversity and fitness values of TPLSO, CSO, and GPSO on four 1000-D benchmark functions, that is, f_3 , f_7 , f_8 , and f_{18} from CEC'10. (a) Shows the fitness value and diversity of the comparative methods on f_3 , while (b), (c), and (d) are the corresponding results of f_7 , f_8 , and f_{18} , respectively.

unimodal), f_7 (partially separable and unimodal), f_8 (partially separable and multimodal), and f_{18} (partially separable and multimodal), respectively.

The diversity is used to measure particle exploration abilities, and it is defined as [5], [22]

$$D_P = \frac{1}{NP} \sum_{j=1}^{NP} \sqrt{\sum_{d=1}^D (x_j^d - \bar{x}^d)^2} \quad (16)$$

where D_P is the diversity of swarm P , $\bar{x}^d = (1/NP) \sum_{j=1}^{NP} x_j^d$ is the mean value of the d th dimension over all particles in the swarm. x_j^d is the value of the d th dimension of particle j , and NP is the swarm size.

The comparative results of the three algorithms are shown in Fig. 4. The maximum number of FEs is set at $3000 \times D$, where D is the dimension size. In addition, the swarm size NP is set to 400 for all algorithms here. We obtain the following findings from this figure.

From Fig. 4(a) and (b), for the unimodal functions, we find that the exploitation ability of TPLSO is properly biased on f_3 and f_7 ; thus, TPLSO has a better solution and converges faster than CSO and GPSO. Specifically, considering f_7 , it can be seen that after FEs $> 2.5E + 06$, the fitness value and diversity of TPLSO have changed substantially because the high diversity is maintained during the mass learning phase, which allows the particles to jump out of the local

TABLE III
OPTIMIZATION ERRORS (MEAN VALUES AND STANDARD DEVIATIONS ARE ON THE FIRST AND SECOND ROWS, RESPECTIVELY.
THE THIRD ROW SHOWS THE T -VALUES) ON 1000-D TEST FUNCTIONS (CEC'2010)

Functions	TPLSO	TPLSO-L	DLLSO	CSO	SL-PSO	DECC-DG	CCPSO2	MLCC	DMS-L-PSO
f_1	2.93 E-19	1.14 E-21	3.14 E-22	4.40 E-12	7.90 E-18	1.83 E+04	2.96 E+00	8.64 E-13	1.62 E+07
	1.81 E-20	1.08 E-22	8.05 E-23	5.94 E-13	3.30 E-18	4.66 E+04	6.68 E+00	2.97 E-12	1.41 E+06
	-	8.86 E+01	8.86 E+02	-4.05 E+01	-1.26 E+01	-2.15 E+36	-2.43 E+00	-1.59 E+00	-6.29 E+01
f_2	6.17 E+02	5.85 E+02	9.83 E+02	7.40 E+03	1.91 E+03	4.43 E+03	4.30 E+00	2.89 E+00	5.46 E+03
	3.10 E+01	2.14 E+01	4.38 E+01	2.85 E+02	1.12 E+02	1.87 E+02	1.11 E+00	1.52 E+00	5.38 E+02
	-	2.99 E-01	-2.03 E+00	-1.29 E+02	-6.09 E+01	-1.10 E+02	1.08 E+02	1.08 E+02	-4.92 E+01
f_3	9.04 E-14	2.83 E-14	2.73 E-14	2.60 E-09	1.86 E+00	1.65 E+01	4.51 E-03	2.10 E-07	1.56 E+01
	2.15 E-15	0.00 E+00	2.39 E-15	2.62 E-10	3.31 E-01	3.02 E-01	1.66 E-03	1.12 E-06	1.09 E-01
	-	1.19 E+01	1.26E+01	-5.43 E+01	-3.08 E+01	-2.99 E+02	-1.49 E+01	-1.03 E+00	-7.83 E+02
f_4	3.63 E+11	3.79 E+11	4.40 E+11	7.25 E+11	2.99 E+11	5.22 E+12	1.69 E+12	1.71 E+13	4.42 E+11
	7.11 E+10	7.66 E+10	1.10 E+11	1.23 E+11	7.17 E+10	1.89 E+12	1.02 E+12	5.46 E+12	8.03 E+10
	-	-0.22 E-01	-9.06 E-01	-1.39 E+01	3.47 E+00	-1.41 E+01	-7.11 E+00	-1.69 E+01	-4.03 E+00
f_5	9.25 E+06	5.38 E+06	1.23 E+07	2.86 E+06	3.18 E+07	1.53 E+08	4.16 E+08	4.98 E+08	9.25 E+07
	2.36 E+06	1.83 E+06	3.39 E+06	1.80 E+06	6.21 E+06	2.15 E+07	1.37 E+08	1.06 E+08	9.04 E+06
	-	2.98 E+00	-1.41 E+00	1.12 E+01	-1.86 E+01	-3.64 E+01	-1.63 E+01	-2.53 E+01	-4.87 E+01
f_6	1.71 E-07	3.55 E-09	5.20 E-01	8.21 E-07	2.08 E+01	1.64 E+01	1.68 E+07	1.77 E+07	3.78 E+01
	9.00 E-09	4.89 E-13	7.45 E-01	2.66 E-07	2.64 E+00	3.45 E-01	5.20 E+06	4.37 E+06	1.21 E+01
	-	9.78 E+01	-5.47 E+00	-1.34 E+01	-6.95 E+00	-5.47 E+00	-1.77 E+01	-2.22 E+01	-1.71 E+01
f_7	2.60 E-04	1.37 E-01	7.17 E+02	2.01 E+04	6.49 E+04	1.39 E+04	2.05 E+08	1.50 E+08	3.46 E+06
	1.71 E-04	9.29 E-02	2.58 E+03	3.87 E+03	5.60 E+04	1.27 E+04	2.03 E+08	1.45 E+08	1.15 E+05
	-	-5.47 E+00	-5.47 E+00	-2.85 E+01	-6.35 E+00	-5.99 E+00	-5.53 E+00	-5.67 E+00	-1.65 E+02
f_8	3.01 E+06	1.93 E+07	2.44 E+07	3.87 E+07	7.81 E+06	2.75 E+07	4.13 E+07	6.58 E+08	2.03 E+07
	2.60 E+05	1.84 E+05	2.46 E+05	6.81 E+04	1.54 E+06	2.62 E+07	3.82 E+07	3.41 E+07	1.88 E+06
	-	-4.62 E+00	-4.80 E+00	-7.27 E+02	-1.68 E+01	-5.12 E+00	-5.49 E+00	-1.05 E+02	-4.99 E+01
f_9	2.07 E+07	2.75 E+07	4.35 E+07	7.03 E+07	3.29 E+07	5.57 E+07	1.01 E+08	2.43 E+08	2.08 E+07
	1.12 E+06	2.31 E+06	4.27 E+06	5.73 E+06	4.45 E+06	6.43 E+06	3.28 E+07	2.15 E+07	1.59 E+06
	-	-1.35 E+00	-2.87 E+00	-4.65 E+01	-1.45 E+01	-2.94 E+01	-1.34 E+01	-5.66 E+01	-2.8 E-01
f_{10}	5.37 E+02	8.23 E+03	8.90 E+02	9.60 E+03	2.54 E+03	4.43 E+03	5.02 E+03	4.23 E+03	5.23 E+03
	2.93 E+01	1.98 E+02	3.65 E+01	7.65 E+01	2.17 E+02	1.28 E+02	7.80 E+02	1.44 E+03	4.25 E+02
	-	-5.13 E+01	-2.17 E+00	-6.05 E+02	-5.01 E+01	-1.62 E+02	-3.15 E+01	-1.40 E+01	-6.03 E+01
f_{11}	1.25 E-12	1.72 E-13	5.80 E+00	4.31 E-08	2.31 E+01	1.02 E+01	1.96 E+02	1.97 E+02	1.68 E+02
	4.53 E-14	4.05 E-15	5.40 E+00	5.13 E-09	2.10 E+00	8.71 E-01	2.11 E+00	1.13 E+00	1.91 E+00
	-	3.32 E+01	-5.47 E+00	-4.60 E+01	-6.02 E+01	-6.41 E+01	-5.09 E+02	-9.55 E+02	-4.81 E+02
f_{12}	4.45 E+03	3.19 E+04	1.25 E+04	4.37 E+05	1.75 E+04	2.81 E+03	3.35 E+04	1.03 E+05	2.38 E+01
	3.86 E+02	2.37 E+03	1.45 E+03	6.21 E+04	9.05 E+03	1.08 E+03	1.15 E+04	1.56 E+04	9.88 E+00
	-	-4.71 E+00	-3.53 E+00	-3.82 E+01	-7.89 E+00	7.83 E+00	-1.38 E+01	-3.46 E+01	6.28 E+01
f_{13}	6.19 E+02	6.38 E+02	7.34 E+02	6.29 E+02	9.59 E+02	6.28 E+03	1.33 E+03	4.23 E+03	1.06 E+04
	1.04 E+02	1.94 E+02	1.92 E+02	2.33 E+02	3.76 E+02	3.65 E+03	1.72 E+02	4.71 E+03	6.17 E+04
	-	-1.60 E-01	-8.51 E-01	-2.10 E-01	-4.77 E+00	-8.49 E+00	-1.94 E+01	-4.19 E+00	-8.80 E-01
f_{14}	6.55 E+07	1.00 E+08	1.23 E+08	2.49 E+08	8.41 E+07	3.43 E+08	3.05 E+08	5.71 E+08	1.19 E+07
	2.78 E+06	4.38 E+06	7.37 E+06	1.54 E+07	6.32 E+06	2.42 E+07	1.15 E+08	5.49 E+07	1.63 E+06
	-	-1.89 E+00	-2.56 E+00	-6.42 E+01	-1.47 E+01	-6.23 E+01	-1.14 E+01	-5.04 E+01	9.11 E+01
f_{15}	9.60 E+03	1.00 E+04	8.35 E+02	1.02 E+04	1.13 E+04	5.86 E+03	1.08 E+04	8.91 E+03	5.55 E+03
	6.15 E+01	4.45 E+01	4.30 E+01	5.22 E+01	8.66 E+01	1.05 E+02	1.36 E+03	2.07 E+03	3.47 E+02
	-	-2.21 E-01	5.73 E+01	-4.07 E+01	-8.77 E+01	1.68 E+02	-4.83 E+00	1.83 E+00	6.29 E+01
f_{16}	1.75 E-12	2.79 E-13	4.24 E+00	5.88 E-08	2.53 E+01	7.51 E-13	3.95 E+02	3.81 E+02	3.17 E+02
	3.21 E-13	3.28 E-15	2.40 E+00	5.62 E-09	1.63 E+01	6.25 E-14	5.74 E-01	5.77 E+01	2.03 E+00
	-	1.89 E+01	-5.47 E+00	-5.73 E+01	-8.50 E+01	1.67 E+01	-3.76 E+03	-3.62 E+01	-8.55 E+02
f_{17}	5.38 E+04	3.77 E+05	9.05 E+04	2.21 E+06	9.00 E+04	4.04 E+04	1.23 E+05	3.49 E+05	4.30 E+01
	2.08 E+03	2.32 E+04	3.52 E+03	1.52 E+05	1.58 E+04	2.28 E+03	5.23 E+04	3.11 E+04	1.16 E+01
	-	-4.69 E+00	-2.22 E+00	-7.77 E+01	-1.24 E+01	2.38 E+01	-7.24 E+00	-5.18 E+01	1.42 E+02
f_{18}	1.39 E+03	1.63 E+03	2.56 E+03	1.73 E+03	2.77 E+03	1.46 E+10	3.08 E+03	1.82 E+04	2.43 E+04
	2.57 E+02	4.81 E+02	8.31 E+02	5.12 E+02	8.33 E+02	2.04 E+09	2.46 E+02	9.47 E+03	1.09 E+04
	-	-7.91 E-01	-2.49 E+00	-3.25 E+00	-8.67 E+00	-3.91 E+01	-2.60 E+01	-9.72 E+00	-1.15 E+01
f_{19}	1.19 E+06	7.60 E+06	1.81 E+06	1.01 E+07	5.10 E+06	1.75 E+06	1.52 E+06	2.03 E+06	2.02 E+06
	3.91 E+04	4.51 E+05	9.96 E+04	5.64 E+05	7.05 E+05	1.10 E+05	7.10 E+04	1.43 E+05	1.40 E+05
	-	-4.64 E+00	-1.87 E+00	-8.63 E+01	-3.04 E+01	-2.65 E+01	-2.23 E+01	-3.14 E+01	-3.14 E+01
f_{20}	1.31 E+03	1.13 E+03	1.88 E+03	1.05 E+03	1.85 E+03	6.40 E+10	2.10 E+03	2.31 E+03	9.87 E+02
	7.19 E+01	6.71 E+01	1.90 E+02	1.45 E+02	2.58 E+02	6.97 E+09	1.75 E+02	1.56 E+02	1.39 E+01
	-	8.70 E-01	-1.68 E+00	8.79 E+00	-1.11 E+01	-5.03 E+01	-2.28 E+01	-3.19 E+01	2.42 E+01
$w/t/l$	-	-	12/5/3	17/1/2	19/0/1	16/0/4	19/0/1	16/3/1	14/2/4

trap and then quickly converge in the elite learning phase. The same results also can be observed from f_8 . For the multimodal functions, the ability to explore TPLSO is properly emphasized so that stagnation and premature convergence can be avoided. Therefore, TPLSO obtains more satisfactory performance compared with CSO and GPSO on multimodal functions f_8 and f_{18} . In summary, we find that TPLSO not only maintains the good exploitation and exploration abilities but also compromises these two aspects well during each evolution.

There are two ways to calculate the mean position \bar{X} in TPLSO, namely, one global version \bar{X}_g and one local version \bar{X}_l . \bar{X}_g can preserve a higher diversity compared to \bar{X}_l . Therefore, to further verify the diversity influence of EAs, several experiments are conducted on the CEC'2010 benchmark function of TPLSO with different mean position. This algorithm proposed in this article with \bar{X}_l was denoted by TPLSO-L. The corresponding results are shown in Table III, and the highlighted value indicates that TPLSO-L obtains better performance. From this comparison, we can see that

TABLE IV
OPTIMIZATION ERRORS (MEAN VALUES AND STANDARD DEVIATIONS ARE ON THE FIRST AND SECOND ROWS, RESPECTIVELY. THE THIRD ROW SHOWS THE T -VALUES) ON 1000-D TEST FUNCTIONS (CEC'2013)

Functions	TPLSO	DLLSO	CSO	SL-PSO	DECC-DG	CCPSO2	MLCC	DMS-L-PSO
f_1	3.13 E-19 1.39 E-20 -	3.40 E-22 1.33 E-22 1.26 E+02	7.76 E-12 1.30 E-12 -3.27 E+01	1.08 E-17 2.51 E-18 -2.28 E+01	6.51 E+03 1.61 E+04 -2.21 E+00	4.10 E+01 3.13 E+01 -7.17 E+00	8.61 E-10 4.39 E-09 -1.07 E+00	1.97 E+09 1.28 E+08 -8.43 E+01
f_2	7.17 E+02 2.52 E+01 -	1.15 E+03 5.78 E+01 -2.07 E+00	8.54 E+03 2.67 E+02 -1.59 E+02	2.12 E+03 1.35 E+03 -5.69 E+00	1.28 E+04 7.10 E+02 -9.32 E+01	3.60 E+01 4.85 E+00 1.45 E+02	3.82 E+00 1.74 E+00 1.54 E+02	8.65 E+03 4.89 E+02 -8.87 E+01
f_3	2.16 E+01 5.20 E-03 -	2.16 E+01 4.07 E-03 0.00 E+00	2.16 E+01 6.15 E-03 0.00 E+00	2.16 E+01 5.20 E-02 0.00 E+00	2.14 E+01 1.46 E-02 7.11 E+01	2.00 E+01 1.27 E-04 1.68 E+03	2.00 E+01 2.75 E-04 1.68 E+03	2.08 E+01 1.67 E-01 2.62 E+01
f_4	5.57 E+09 8.76 E+08 -	6.69 E+09 1.68 E+09 -9.10 E-01	1.31 E+10 2.54 E+09 -1.54 E+01	4.36 E+09 9.47 E+08 5.13 E+00	7.71 E+10 2.82 E+10 -1.39 E+01	3.50 E+10 2.17 E+10 -7.42 E+00	2.35 E+11 1.27 E+11 -9.89 E+00	2.93 E+11 7.26 E+10 -2.17 E+01
f_5	6.71 E+05 8.95 E+04 -	7.01 E+05 1.27 E+05 -2.30 E+00	5.92 E+05 1.08 E+05 3.08 E+00	8.41 E+05 1.76 E+05 -4.71 E+00	5.76 E+06 3.82 E+05 -7.10 E+01	1.40 E+07 4.81 E+06 -1.52 E+01	1.28 E+07 3.45 E+06 -1.92 E+01	3.96 E+06 5.83 E+05 -3.05 E+01
f_6	1.06 E+06 9.27 E+02 -	1.06 E+06 8.27 E+02 0.00 E+00	1.06 E+06 1.10 E+03 0.00 E+00	1.06 E+06 1.47 E+03 0.00 E+00	1.06 E+06 1.87 E+03 0.00 E+00	1.05 E+06 5.25 E+03 1.02 E+01	1.05 E+06 4.13 E+03 1.29 E+01	1.00 E+06 5.20 E+03 6.22 E+01
f_7	2.95 E+05 1.38 E+05 -	1.61 E+06 8.38 E+05 -4.46 E+00	5.86 E+06 2.56 E+06 -1.19 E+01	1.63 E+06 7.07 E+05 -1.05 E+01	4.78 E+08 1.90 E+08 -1.38 E+00	4.13 E+08 9.37 E+08 -2.41 E+00	1.44 E+09 1.08 E+09 -7.30 E+00	1.38 E+07 7.65 E+08 9.00 E-02
f_8	1.33 E+14 2.10 E+13 -	1.19 E+14 3.34 E+13 6.30 E-01	2.64 E+14 5.86 E+13 -1.15 E+01	1.05 E+14 3.63 E+13 3.65 E+00	3.75 E+15 1.86 E+15 -1.07 E+01	3.75 E+15 9.98 E+14 -5.74 E+00	1.18 E+15 6.19 E+15 -8.73 E+00	2.78 E+14 5.08 E+14 -1.56 E+00
f_9	3.98 E+07 5.35 E+06 -	1.30 E+08 3.99 E+07 -3.79 E+00	6.07 E+07 1.61 E+07 -6.75 E+00	8.26 E+07 2.03 E+07 -1.12 E+01	4.90 E+08 3.17 E+07 -7.67 E+01	3.76 E+09 1.01 E+09 -2.02 E+01	9.55 E+08 2.92 E+08 -1.72 E+01	3.61 E+08 4.60 E+07 -3.79 E+01
f_{10}	9.39 E+07 1.78 E+05 -	9.40 E+07 2.11 E+05 -5.80 E-02	9.40 E+07 1.52 E+07 -3.60 E-02	9.26 E+07 1.67 E+06 4.24 E+00	9.44 E+07 2.47 E+05 -8.99 E+00	9.30 E+07 7.02 E+05 6.80 E+00	9.28 E+07 6.08 E+05 9.51 E+00	9.18 E+07 1.05 E+06 1.08 E+01
f_{11}	1.29 E+08 3.88 E+07 -	9.29 E+11 9.51 E+09 -5.47 E+00	9.30 E+11 1.03 E+10 -4.94 E+02	9.33 E+11 1.46 E+10 -3.49 E+02	4.82 E+10 4.32 E+10 -6.09 E+00	9.36 E+11 1.54 E+10 -3.32 E+02	2.28 E+11 1.54 E+11 -8.10 E+00	1.05 E+11 7.54 E+10 -7.62 E+00
f_{12}	1.12 E+03 8.13 E+01 -	1.78 E+03 1.40 E+02 -2.03 E+00	1.07 E+03 7.78 E+01 2.43 E+00	1.78 E+03 1.73 E+02 -1.89 E+01	1.70 E+11 2.23 E+10 -4.17 E+01	2.11 E+03 1.77 E+02 -2.78 E+01	2.50 E+03 7.51 E+02 -1.00 E+01	6.98 E+04 5.53 E+04 -6.80 E+00
f_{13}	9.07 E+07 3.55 E+07 -	3.35 E+08 1.70 E+08 -3.97 E+00	6.68 E+08 2.44 E+08 -1.28 E+01	4.65 E+08 2.36 E+08 -8.59 E+00	2.06 E+10 5.35 E+09 -2.09 E+01	4.03 E+09 2.30 E+09 -9.38 E+00	1.04 E+10 3.72 E+09 -1.52 E+01	1.32 E+10 6.55 E+09 -1.09 E+01
f_{14}	2.30 E+07 4.38 E+06 -	1.72 E+08 1.39 E+08 -4.74 E+00	3.62 E+09 1.44 E+09 -1.37 E+01	3.27 E+08 5.17 E+08 -3.22 E+00	1.92 E+10 1.44 E+10 -7.29 E+00	9.10 E+10 8.53 E+10 -5.84 E+00	2.22 E+11 8.55 E+10 -1.42 E+01	2.45 E+11 1.25 E+11 -1.07 E+01
f_{15}	2.42 E+06 1.19 E+05 -	4.50 E+06 3.33 E+05 -2.53 E+00	7.87 E+07 6.50 E+06 -6.42 E+01	5.86 E+07 6.12 E+06 -5.03 E+01	9.90 E+06 2.31 E+06 -1.56 E+01	4.76 E+06 5.08 E+06 -2.52 E+00	1.61 E+07 1.91 E+06 -3.92 E+01	1.57 E+07 3.46 E+06 -2.10 E+01
$w/t/l$	-	9/4/2	10/3/2	10/2/3	13/1/1	11/0/4	10/1/4	10/2/3

TPLSO-L shows better performance on the unimodal functions compared with TPLSO.

C. Comparisons With State-of-the-Art Algorithms

To verify the feasibility of TPLSO, we compare it with a series of state-of-the-art algorithms dealing with large-scale optimization on the CEC'2010 and CEC'2013 function sets with the dimension of 1000. In particular, four popular PSO variant algorithms, including LLSO [35], CSO [5], SL-PSO [6], and dynamic multiswarm particle swarm optimizer (DMS-L-PSO) [20], and three CCEAs, namely, MLCC [38], cooperative coevolution with DG (DECC-DG) [23], and CCPSO2 [19] are selected for comparison. To provide a fair comparison, the key parameters used in each algorithm are set as the recommendations in the corresponding paper.

The statistical results on the CEC'2010 and CEC'2013 test functions are shown in Tables III and IV, respectively. The T values are listed along with the mean and standard deviation

for measuring the statistical results. The highlighted T value indicates that TPLSO is significantly better than the corresponding algorithm. Furthermore, a two-tailed T -test was used to compare two different statistical results for the significance level of $\alpha = 0.05$. In addition, $w/t/l$ in the last row indicates that TPLSO wins on w functions, ties on the t functions, and loses on l functions.

1) *Results on CEC'2010*: For the CEC'2010 functions set, Table III shows that TPLSO achieves better performance than the comparative algorithms on most of the 20 functions. Compared with the four popular PSO variants, TPLSO indicates its considerable advantages on 12, 17, 19, and 14 functions, respectively, while TPLSO fails only on 3, 2, 1, and 4 functions, respectively. In particular, in contrast with DLLSO, which gains the best performance in the current PSO variants, TPLSO fails only on two separable and unimodal functions (f_1 , f_3) and one partially separable and multimodal function (f_{15}). It is worth noting that TPLSO performs identically on five functions (f_4 , f_5 , f_{13} , f_{18} , and f_{19}) as

DLLSO, and on these five functions, TPLSO obtains better performance. Further observation shows that TPLSO has better performance in most partially separable and nonseparable functions, which indicates that TPLSO preserves the higher diversity than DLLSO. In comparison with MLCC, CCPSO2, and DECC-DG, TPLSO wins on 16, 19, and 16 functions, and it only loses 4, 1, and 1 functions, respectively.

2) *Results on CEC'2013*: The statistical results of Table IV demonstrate that TPLSO consistently outperforms the comparative algorithms on the CEC'2013 set, where the functions are more difficult to optimize than those of the CEC'2010 set. Compared with the four PSO variants, TPLSO defeats them on 9, 10, 10, and 10 functions. Compared with the three CCEAs, TPLSO shows a significant effectiveness on at least 10 functions.

Observing the results in Tables III and IV, we find that TPLSO achieves better performance in the solution quality. The superiority of TPLSO can be attributed to the mass and elite learning strategies and the exemplar method. In the mass learning phase, particle grouping in the swarm and the exemplar selection in the study group enhance the diversity of the swarm. Particles have a greater chance to jump out of the local optima and to look for more promising areas. In the elite learning phase, mutual learning between particles further enhances their fitness values, which leads to promising areas as soon as possible. In summary, these two types of compromises in exploration and exploitation make TPLSO achieve good performance.

V. CONCLUSION

In this article, we proposed a new swarm algorithm called TPLSO that is based on mass learning and elite learning strategies. The theoretical analysis shows that these two strategies are able to preserve the exploration and exploitation abilities of TPLSO well. Despite the simplicity of the proposed TPLSO algorithm, the comparative studies on 1000-D CEC'2010 and CEC'2013 test functions showed that it outperforms several state-of-the-art approaches for large-scale optimization.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valued comments and constructive suggestions that significantly improved the quality of this article.

REFERENCES

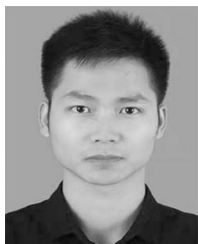
- [1] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Bristol, U.K.: CRC Press, 1997.
- [2] W.-N. Chen, D.-Z. Tan, Q. Yang, T. Gu, and J. Zhang, "Ant colony optimization for the control of pollutant spreading on social networks," *IEEE Trans. Cybern.*, early access, doi: [10.1109/TCYB.2019.2922266](https://doi.org/10.1109/TCYB.2019.2922266).
- [3] R. Lan, Y. Zhu, H. Lu, Z. Tang, Z. Liu, and X. Luo, "Large-scale optimisation via cooperatively coevolving competition swarm optimiser," *Enterprise Inf. Syst.*, early access, doi: [10.1080/17517575.2019.1681518](https://doi.org/10.1080/17517575.2019.1681518).
- [4] W.-N. Chen *et al.*, "Particle swarm optimization with an aging leader and challengers," *IEEE Trans. Evol. Comput.*, vol. 17, no. 2, pp. 241–258, Apr. 2013.
- [5] 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.
- [6] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.*, vol. 291, pp. 43–60, Jan. 2015.
- [7] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.
- [8] Y.-J. Gong, E. Chen, X. Zhang, L. M. Ni, and J. Zhang, "AntMapper: An ant colony-based map matching approach for trajectory-based applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 390–401, Feb. 2018.
- [9] Y.-J. Gong, M. Shen, J. Zhang, O. Kaynak, W.-N. Chen, and Z.-H. Zhan, "Optimizing RFID network planning by using a particle swarm optimization algorithm with redundant reader elimination," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 900–912, Nov. 2012.
- [10] Y.-J. Gong *et al.*, "An efficient resource allocation scheme using particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 6, pp. 801–816, Dec. 2012.
- [11] Y.-J. Gong and Y. Zhou, "Differential evolutionary superpixel segmentation," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1390–1404, Mar. 2018.
- [12] M. Jian, K.-M. Lam, J. Dong, and L. Shen, "Visual-patch-attention-aware saliency detection," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1575–1586, Aug. 2015.
- [13] M. Jian *et al.*, "The extended marine underwater environment database and baseline evaluations," *Appl. Soft Comput.*, vol. 80, pp. 425–437, Jul. 2019.
- [14] D. W. Johnson and R. T. Johnson, "Cooperative, competitive, and individualistic learning," *J. Res. Develop. Educ.*, vol. 12, no. 1, pp. 3–15, 1978.
- [15] R. Lan, L. Sun, Z. Liu, H. Lu, Z. Su, and X. Luo, "Cascading and enhanced residual networks for accurate single image super-resolution," *IEEE Trans. Cybern.*, early access.
- [16] R. Lan, Y. Zhou, Z. Liu, and X. Luo, "Prior knowledge-based probabilistic collaborative representation for visual recognition," *IEEE Trans. Cybern.*, early access, doi: [10.1109/TCYB.2018.2880290](https://doi.org/10.1109/TCYB.2018.2880290).
- [17] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 265–272.
- [18] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," *Gene*, vol. 7, no. 33, p. 8, 2013.
- [19] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.
- [20] J.-J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proc. IEEE Swarm Intell. Symp. (SIS)*, 2005, pp. 124–129.
- [21] Y. Mei, M. N. Omidvar, X. Li, and X. Yao, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Trans. Math. Softw.*, vol. 42, no. 2, p. 13, 2016.
- [22] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Proc. IEEE Congr. Evol. Comput. World Comput. Intell.*, 2008, pp. 1128–1134.
- [23] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, Jun. 2014.
- [24] M. N. Omidvar, M. Yang, Y. Mei, X. Li, and X. Yao, "DG2: A faster and more accurate differential grouping for large-scale black-box optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 929–942, Dec. 2017.
- [25] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 211–224, Jun. 2004.
- [26] Y. V. Pehlivanoglu, "A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 436–452, Jun. 2013.
- [27] M. A. Potter, "The design and analysis of a computational model of cooperative coevolution," Ph.D. dissertation, Dept. Comput. Sci., Univ. Texas at Austin, Austin, TX, USA, 1997.
- [28] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2238–2251, Oct. 2016.
- [29] R. E. Slavin, "Cooperative learning," *Rev. Educ. Res.*, vol. 50, no. 2, pp. 315–342, 1980.
- [30] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

- [31] Y. Sun, M. Kirley, and S. K. Halgamuge, "Extended differential grouping for large scale global optimization with direct and indirect variable interactions," in *Proc. ACM Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2015, pp. 313–320.
- [32] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise. (2009). *Benchmark Functions for the CEC'S2010 Special Session and Competition on Large Scale Global Optimization*. [Online] Available: <http://titan.csit.rmit.edu.au/~e46507/publications/lsgo-cec10.pdf>
- [33] G. Xu *et al.*, "Particle swarm optimization based on dimensional learning strategy," *Swarm Evol. Comput.*, vol. 45, pp. 33–51, Mar. 2019.
- [34] Q. Yang *et al.*, "A distributed swarm optimizer with adaptive communication for large-scale optimization," *IEEE Trans. Cybern.*, early access, doi: [10.1109/TCYB.2019.2904543](https://doi.org/10.1109/TCYB.2019.2904543).
- [35] Q. Yang, W.-N. Chen, J. Da Deng, Y. Li, T. Gu, and J. Zhang, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 578–594, Aug. 2018.
- [36] Q. Yang *et al.*, "Segment-based predominant learning swarm optimizer for large-scale optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2896–2910, Sep. 2017.
- [37] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.
- [38] Z. Yang, K. Tang, and X. Yao, "Multilevel cooperative coevolution for large scale optimization," in *Proc. IEEE Congr. Evol. Comput. Intell.*, 2008, pp. 1663–1670.
- [39] Z. Zhan, Z. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, early access.
- [40] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.
- [41] Y.-H. Zhang, Y.-J. Gong, T.-L. Gu, Y. Li, and J. Zhang, "Flexible genetic algorithm: A simple and generic approach to node placement problems," *Appl. Soft Comput.*, vol. 52, pp. 457–470, Mar. 2017.
- [42] H. Zhao *et al.*, "Local binary pattern-based adaptive differential evolution for multimodal optimization problems," *IEEE Trans. Cybern.*, early access, doi: [10.1109/TCYB.2019.2927780](https://doi.org/10.1109/TCYB.2019.2927780).



Rushi Lan received the B.S. and M.S. degrees from the Nanjing University of Information Science and Technology, Nanjing, China, and the Ph.D. degree from the University of Macau, Macau, China.

He is currently an Associate Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include image classification, image denoising, and metric learning.



Yu Zhu received the B.S. degree from the Nanchang Institute of Technology, Nanchang, China, in 2015. He is currently pursuing the M.S. degree with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China.

His current research interest includes evolutionary computation and its applications.



Huimin Lu received the M.S. degrees in electrical engineering from the Kyushu Institute of Technology, Kitakyushu, Japan, and Yangzhou University, Yangzhou, China, in 2011, and the Ph.D. degree in electrical engineering from the Kyushu Institute of Technology, in 2014.

From 2013 to 2016, he was a JSPS Research Fellow with the Kyushu Institute of Technology, where he is currently an Associate Professor. He is an Excellent Young Researcher with MEXT, Tokyo, Japan. His research interests include computer vision, robotics, artificial intelligence, and ocean observing.



Zhenbing Liu received the B.S. degree from Qufu Normal University, Qufu, China, and the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China.

He was a Visiting Scholar with the Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA, in 2015. He is currently a Professor and the Doctoral Supervisor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. His main research interests include image processing, machine learning, and pattern recognition.



Xiaonan Luo received the B.S. degree in computational mathematics from Jiangxi University, Nanchang, China, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, and the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China.

He is currently a Professor with the School of Computer and Information Security, Guilin University of Electronic Technology, Guilin, China. He received the National Science Fund for Distinguished Young Scholars granted by the National Natural Science Foundation of China, and was the Director of the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China. His current research interests include computer graphics, machine learning, and pattern recognition.