Applied Soft Computing Journal 103 (2021) 107132

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Probability-optimal leader comprehensive learning particle swarm optimization with Bayesian iteration ☆

Xing Zhang [a,b,c,*], Wei Sun [a,b,c,*], Min Xue [a,c,*], Anping Lin [c,d]

[a] College of Electrical and Information Engineering, Hunan University, Changsha, 410082, PR China
[b] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha, 410082, PR China
[c] Hunan Key Laboratory of Intelligent Robot Technology in Electronic Manufacturing, Hunan University, Changsha, 410082, PR China
[d] College of Electronic Information and Electrical Engineering, Xiangnan University, Chenzhou, 423000, PR China

ARTICLE INFO

Article history:
Received 31 January 2020
Received in revised form 15 January 2021
Accepted 18 January 2021
Available online 22 January 2021

Keywords:
Bayesian iteration method
Particle swarm optimization (PSO)
Comprehensive learning (CL)

ABSTRACT

In this paper, a novel comprehensive learning particle swarm optimization algorithm, which is based on the Bayesian iteration method and named as Bayesian comprehensive learning particle swarm optimization (BCLPSO), is proposed. In the original PSO, the flying direction of each particle is based on its own historical best position and global optimum. This updating mechanism, however, easily falls into the local optimum, and the potential optimum solution may be ignored in the iteration and update process. Therefore, the BCLPSO is designed to facilitate discovering potential solution and avoid the problem of premature convergence. In the BCLPSO algorithm, the exemplar of the swarm is not the global best position but the particle location with the largest posterior probability based on the Bayesian formula. The posterior probability is developed by historical prior information. This means that the posterior probability can inherit the historical information of particles that may be exploited. In this way, the swarm diversity can be preserved to prevent premature convergence. The BCLPSO is experimentally validated on the CEC2017 benchmark functions and compared with other state-of-the-art particle swarm optimization algorithms. The results show that BCLPSO outperforms other comparative PSO variants on the CEC 2017 test suite. Furthermore, the algorithm is applied to the quality control process of an automated welding production line for the automobile body and is found to exhibit superior performance.

1. Introduction

Particle swarm optimization (PSO) is a biological evolutionary algorithm, which originates from the study of birds or other social animals' foraging behavior. The PSO is a stochastic optimization technique, which was proposed by Eberhart and Kennedy in [1] and [2], respectively. Different from other evolutionary algorithms (e.g., genetic algorithm), the PSO does not have operators, such as selection, replication, and mutation, but achieves population evolution through competition and cooperation among individuals [3]. Its mechanism is simple and can effectively explore global solutions to some difficult problems; hence, the PSO has become one of the most popular optimization methods that is successfully applied in many engineering fields involved in assignment [4], power systems [5], and biomedical image registration [6].

In the PSO, each individual in the swarm is called a "particle", which represents a potential solution to the problem. The global optimal solution is considered as the location of the food the birds are searching for. According to the historical optima of the particle and swarm, each particle has a fitness function value and a velocity to adjust the convergence direction. As a new intelligent algorithm, the PSO rapidly converges. It is difficult, however, to maintain the balance between exploration and exploitation in the search process because the particles always track the optimal positions of individuals and the global optimal position. In the early stage, the velocity of the particle is extremely high, causing the swarm diversity to reduce rapidly and get trapped in the local optimum. When the particles get trapped in the local optimum, the PSO may lose its search capability as a result of swarm diversity. To solve the premature convergence problem, many improved research algorithms have been proposed, it can basically be classified into the following categories.

☆ This work is supported by National Natural Science Foundation of China (Grant Nos. U1813205 and 61573135), Science and Technology Plan Project of Shenzhen City, China (JCYJ20170306141557198), Hunan Key Laboratory of Intelligent Robot Technology in Electronic Manufacturing Open Foundation (2017TP1011), and Scientific Research Fund of Hunan Provincial Education Department, China (18B435).

* Corresponding author at: College of Electrical and Information Engineering, Hunan University, Changsha, 410082, PR China.

E-mail addresses: xing_zhangshare@163.com (X. Zhang), david-sun@126.com (W. Sun), xueminxhd@163.com (M. Xue).

https://doi.org/10.1016/j.asoc.2021.107132
1568-4946/

In the first category, the influence of various parameter configurations on the PSO is investigated. Shi and Eberhart [7] introduced the concept of inertia weight (ω) to balance the global search and local search abilities, which significantly improve the PSO performance. Clerc and Kennedy [8] also proposed another related parameter called the constriction coefficient (χ) to prevent premature convergence. To solve the problem of prior parameter tuning, various self-adaptive particle swarm optimization algorithms, which can adjust the control parameters adaptively throughout the execution process (e.g., PSO-AIWF [9], SRPSO [10], and UAPSO-A [11]), are proposed. Marco et al. [12] developed fuzzy logic to independently calculate the inertia weight, cognitive and social factors, and velocity threshold for each particle; the fuzzy self-turning PSO is subsequently proposed. However, considering some specific engineering problems, Serani et al. [13] proposed an efficient deterministic PSO (DPSO) formulations. If only one optimization run is needed, the deterministic PSO algorithm has advantages.

In the second category, the influence of various topology structures on the PSO algorithm is determined. Kennedy and Mendes [14] assumed that individual behaviors among humans are not usually affected by any other individuals, but by all neighbors. Based on this theory, they proposed the fully informed particle swarm. Suganthan [15] proposed a dynamically adjusted neighborhood method. In the initial stage, it has a ring structure, and as the number of iterations increases, the neighborhood of particles gradually increases until all particles are included. Peram [16] presented a novel PSO algorithm based on the fitness–distance ratio (FDR–PSO) under neighbor interaction. Lim and Isa [17] proposed a new PSO algorithm called PSO with adaptive time-varying topology connectivity that employs the adaptive time-varying topology connectivity method and a new learning strategy. Hanaf [18] presented a new PSO called hierarchical PSO, which employs a dynamic tree hierarchy based on the performance of each particle in the population to define the neighborhood structure.

The third category involves novel learning strategies. The new learning strategy refers to a new approach to update the speed or position of particles in the proposed improved PSO. Liang [19] developed a novel PSO named as comprehensive learning PSO, which utilizes a new learning strategy to maintain the swarm diversity and thereby prevent premature convergence in solving multi-modal problems. In the comprehensive learning particle swarm optimization (CLPSO), each dimension of a particle determines the learning object according to the learning probability. Sabat [20] presented an integrated learning particle swarm optimizer, which determines the deviated particles by the fitness value or Euclidean distance between the particle and the optimal position of the swarm. Lynn [21] proposed a new particle swarm optimization algorithm with enhanced exploration and exploitation based on comprehensive learning called heterogeneous comprehensive learning particle swarm optimization (HCLPSO). Lin and Sun [22] presented an adaptive comprehensive learning particle swarm optimization with cooperative archive, the cooperative archive is employed to provide additional promising information for the proposed algorithm. Tanweer and Suresh [23–25] developed the self-regulating inertia weight and self-perception of the global search direction and proposed the self-regulating particle swarm optimization. Wang and Jin [26] developed a novel surrogate-assisted particle swarm optimization inspired by committee-based active learning. To accelerate convergence rate, Kang and Chen [27] devised a non-inertial opposition-based particle swarm optimization (NOPSO), which is based on a novel kinetic equation without inertial term, and is combined with an adaptive elite mutation strategy and generalized opposition-based learning strategy.

The last category involves a hybrid algorithm, which combines other evolutionary algorithms or meta-heuristics with particle swarm optimization. For example, Løvbjerg [28] proposed a hybrid PSO based on breeding and subpopulations. Zhang [29] introduced a differential evolution factor into the traditional PSO to avoid the convergence to the local optimum. Miranda [30] used the characteristics of the evolutionary strategy to propose a PSO with self-adaptive inertia weight. Gong [31] proposed the genetic learning particle swarm optimization (GL-PSO), which adopts a two-cascading-layer structure: one layer is for the exemplar generation and the other is for particle updates according to the traditional PSO algorithm. Lin and Sun [32] proposed the global genetic learning particle swarm optimization with diversity enhancement by ring topology (GGL-PSOD) to improve the GL-PSO's performance. Lynn and Suganthan [33] accordingly assembled different optimization algorithms for solving complex problems and proposed an ensemble of different particle swarm optimization algorithms called ensemble particle swarm optimizer (EPSO). Chen and Li [34] proposed the dynamic multi-swarm differential learning particle swarm optimizer, which is a novel method for merging the differential evolution operators into each sub-swarm. Ehsan and Mahdi [35] developed a novel hybrid algorithm for solving transmission expansion planning problems in electric power networks, which is combined with shuffled frog leaping algorithm, particle swarm optimization, and teaching learning-based optimization.

The disadvantage of PSO algorithms, however, is that as the particles search for the optimal solution, and they may overlook the potential optimal solution area because of the influence of the exemplar, thereby causing it to fall into the local optimum. To overcome this deficiency, Bayesian iteration probability can be employed to select the other particle as the social learning exemplar. This is an aspect that is not investigated by other scholars.

The Bayesian theorem was proposed by British mathematician Bayes in 1763 [36]. The Bayesian inference employs subjective probability to estimate historical prior information under incomplete information. Thereafter, the Bayesian iteration update formula is used to objectively modify the event process probability. The optimal decision-making method is selected based on the modified probability, which is also the advantage of the Bayesian method. Bayesian inference is widely used in engineering fields, such as Bayesian regression analysis [37], Bayesian risk decision-making [38], neural network [39], and machine learning [40]. Some scholars have paid attention to the particle swarm optimization with Bayesian theory in the late years. For example, Di and Gao [41] proposed a novel PSO algorithms, which uses mutual information to limit particle initialization, and constructs an evolutionary model based on Bayesian network. Liu [42] uses PSO to optimize Bayesian learning network structure. After analyzing the optimal flying behaviors of some classic PSO algorithms, they put forward a new PSO-based method of learning Bayesian network structures. In these studies, PSO algorithm is applied to Bayesian networks to obtain excellent Bayesian learning networks. However, we focus on the performance improvement of PSO algorithm, and advances a novel variant of PSO algorithm, which is based on Bayesian iterative formula.

The PSO algorithm falls into the local optimum prematurely because the particle always approaches the historical optimum after updating its position, the historical optimum may be a local optimum position. Inspired by the Bayesian method, this study extends the comprehensive learning PSO. Consequently, the Bayesian iteration method (BCLPSO)-based PSO, which is a novel variant of the PSO, is developed. The swarm leader in the BCLPSO algorithm is determined by Bayesian posterior probability, which is not the simple historical optimum of the traditional PSO algorithm. The posterior probability is determined by the prior information of particle swarm. Compared with other PSO variants, the strengths of BCLPSO are as follows. (a) The idea of the Bayesian iteration method is applied to the PSO to generate more particles that can become the learning exemplar of the social learning part, thereby enhancing its self-adaptability and robustness; (b) The prior information and historical information of particles close to global optimum are fully utilized in the BCLPSO algorithm. This means that the particles will neither facilely fall into the local optimum nor miss the potential optimum solution.

This paper is organized as follows. The original PSO is introduced in Section 2, and the proposed BCLPSO is discussed in Section 3. In Section 4, the BCLPSO is experimentally validated and compared with other state-of-the-art PSO algorithms on the CEC2017 benchmark functions [43]. In Section 5, the BCLPSO algorithm is further validated against various engineering optimization problems, and the quality control of the statistical process is presented. The main conclusions and proposed topics for future research are summarized in Section 6.

2. Particle swarm optimization

In a D-dimensional target search space, a population is composed of *ps* particles, and each particle can be regarded as a point in space. The state attributes of the *i*th ($i = 1, 2, 3, \ldots, ps$) particle in the *t*th iteration are described by two vectors: position vector $\boldsymbol{x_i^t} = [x_{i,1}^t, x_{i,2}^t, x_{i,3}^t, \ldots, x_{i,D}^t]$ and velocity vector $\boldsymbol{v_i^t} = [v_{i,1}^t, v_{i,2}^t, v_{i,3}^t, \ldots, v_{i,D}^t]$. The rules of position updating in the traditional PSO algorithm are given by

$$v_{i,d}^{t+1} \leftarrow v_{i,d}^t + c_1 r_1 (pbest_{i,d}^t - x_{i,d}^t) + c_2 r_2 (gbest^t - x_{i,d}^t) \quad (1)$$

$$x_{i,d}^{t+1} \leftarrow x_{i,d}^t + v_{i,d}^{t+1} \quad (2)$$

where $\boldsymbol{pbest_i^t} = [pbest_{i,1}, pbest_{i,2}, \ldots, pbest_{i,D}]$ is the best previous position of the *i*th particle in *t* iterations; $\boldsymbol{gbest} = [gbest_1, gbest_2, \ldots, gbest_D]$ is the best position discovered by the whole particle swarm; $\boldsymbol{x_i} = [x_{i,1}, x_{i,2} \cdots, x_{i,D}]$ and $\boldsymbol{v_i} = [v_{i,1}, v_{i,2} \cdots, v_{i,D}]$ represent the position and velocity of the *i*th particle, respectively; c_1 and c_2 denote the acceleration coefficients; r_1 and r_2 are the random numbers in the range [0,1]. In order to control the particles in an effective search space, the velocity ($v_{i,d}^t$) is limited to the maximum magnitude (V_{max}). If $\left|v_{i,d}^t\right|$ exceeds $V_{\max}$, then the particle velocity is $v_{i,d}^t = V_{max}$.

In order to balance the global and local search abilities of particles, the inertia weight, w, is introduced by one of the improved PSO algorithms into the original PSO, and the velocity of the *i*th particle is renewed according to the following rule [7]:

$$v_{\boldsymbol{i,d}}^{\boldsymbol{t+1}} \leftarrow w v_{i,d}^t + c_1 r_1 (pbest_{i,d}^t - x_{i,d}^t) + c_2 r_2 (gbest^t - x_{i,d}^t). \quad (3)$$

In this study, the inertia weight (w) is also employed to balance the global and local search abilities of the proposed PSO algorithm.

3. Particle swarm optimization based on Bayesian iteration

In the original PSO, the flying direction of each particle is based on its own $pbest_{i,d}$ and global optimum $gbest_d$. This updating mechanism, however, easily falls into the local optimum, and the potential optimum solution may be ignored in the iteration and update process; accordingly, the PSO Bayesian iteration method is proposed. The exemplar of the algorithm is not the location of minimum fitness function value but the particle location ($exemplar_d$) with the largest posterior probability after iteration based on the Bayesian formula. The posterior probability is developed by historical prior information, indicating that the posterior probability can record and exploit the historical information of particles. In order to prevent premature convergence, the comprehensive learning (CL) strategy [19] is selected among simple single-population PSO algorithms [12,44–46]. In the CLPSO, the exemplar of particles is not necessarily the particle with the best fitness value; it can be any particle in the swarm of different dimensions. Moreover, the CLPSO proposes a parameter, i.e., refreshing gap (m), which is used to limit the maximum number of iterations so that a particle learns one exemplar. Section 3 is divided into three subsections: Section 3.1 provides an overview of the comprehensive learning strategy, and the proposed Bayesian CLPSO (BCLPSO) algorithm is discussed in Section 3.2. Section 3.3 presents the search mechanism of BCLPSO.

3.1. Comprehensive learning strategy

In the comprehensive learning PSO, each particle velocity is renewed by any particle's *pbest*. Moreover, different learning exemplars are selected for each dimension. Experiments demonstrate that this learning strategy can significantly prevent prematurity and maintain swarm diversity [47]. In the CLPSO, the velocity of the *i*th particle is renewed with the following equation:

$$v_{i,d}^{t+1} \leftarrow w v_{i,d}^t + cr(pbest_{fi(d)}^t - x_{i,d}^t), \quad (4)$$

where $f_i(d) = [f_i(1), f_i(2), \ldots, f_i(D)]$ determines which particle's $pbest_{i,d}$ should follow the *i*th particle for each dimension (d), and the exemplar for each dimension is decided by the learning probability, *Pc*. For each dimension of the *i*th particle, if the learning probability (Pc_i) is larger than a random number, then the corresponding dimension will learn from another particle's *pbest* [19]; otherwise, it will learn from its own*pbest*. The *Pc* of the *i*th particle is defined by the following equation:

$$Pc_i = a + b * \frac{(\exp((10(i-1))/(ps-1)) - 1)}{(\exp(10) - 1)}, \quad (5)$$

where parameters a and b are 0.05 and 0.45, respectively; *ps* represents the particle swarm size. The search range is restricted by the bound $[X_{\min}, X_{\max}]$, and if the particle is outside the bound, its fitness value and position are not renewed. Furthermore, in order to ensure that particle updating can improve its *pbest*, a certain number is defined as the refreshing gap (m). If there is no improvement after m generations, then a new $pbest_{fi(d)}$ will be generated [47].

3.2. CLPSO based on Bayesian iteration method

The Bayesian method focuses on determining the exemplar of particles based on posterior probability, which is updated after each particle uses the Bayesian theorem. Let p_i^t be the posterior probability of the *i*th particle at the *t*th iteration, and the process historical prior information is given as $H_t = [X^0, p^0, X^1, p^1, \ldots, X^{t-1}, p^{t-1}, X^t]$. The D-dimensional position information of the *i*th ($i = 1, 2, 3, \ldots, ps$) particle at the *t*th iteration is denoted by $x_i^t = [x_{i,1}^t, x_{i,2}^t, x_{i,3}^t, \ldots, x_{i,D}^t]$; $p^t = [p_1^t, p_2^t, \ldots, p_{ps}^t]^T$ represents the posterior probability of *ps* particles at the *t*th iteration. The posterior probability, $p_i^t = p(Y_i^t = 1|H_t)$, where $Y_i^t \in (0, 1)$, is the state in which the *i*th particle is chosen as an exemplar at the *t*th iteration, as follows:

$$Y_i^t = \begin{cases} 0, & \textit{the ith particle is chosen as an exemplar} \\ 1, & \textit{the ith particle isnot chosen as an exemplar} \end{cases}. \quad (6)$$

When the *i*th ($i = 1, 2, 3, \ldots, ps$) particle is in the x_i position, its fitness function value is defined as $val(i)$. It is assumed that the D-dimensional position vector, x, has a uniform distribution with the density function, $\phi(x_i)$, which is given by

$$\phi(x_i) = \begin{cases} \dfrac{1}{\overline{val} - val_{min}} & if\ \ val(i) \le \overline{val}; \\ \dfrac{1}{val_{max} - \overline{val}} & if\ \ val(i) > \overline{val}; \end{cases} \tag{7}$$

$$\overline{val} = \frac{\sum_{i=1}^{m} val(i)}{ps} \tag{8}$$

where ps is the swarm size; $\overline{val}$ denotes the mean value of the fitness function. Next, let $p_i^{t-} = p(Y_i^t = 1|H_{t-1})$ denote the prior probability of the ith particle given H_{t-1}. The posterior probability (p^t) can be obtained as follows.

$$\begin{aligned} p_i^t &= p(Y_i^t = 1|X^0, X^1, X^2, \dots, X^t) \\ &= \frac{p_i^{t-} * \phi(x_i|val(i) \le \overline{val})}{p_i^{t-} * \phi(x_i|val(i) \le \overline{val}) + (1 - p_i^{t-}) * \phi(x_i|val(i) > \overline{val})} \\ &= \frac{p_i^{t-}}{p_i^{t-} + (1 - p_i^{t-}) * \frac{val_{max} - \overline{val}}{\overline{val} - val_{min}}} \qquad for\ \ i = 1, 2, 3, \dots, ps \end{aligned} \tag{9}$$

$$\text{If } max(p^t) = p_i^t, \text{ then } x_i^t = exemplar^t. \tag{10}$$

Where the prior probability of the ith particle is

$$p_i^{t-} = p_{min} + p_i^{t-1} * \frac{val(i) - \overline{val}}{val_{min} - \overline{val}} \tag{11}$$

the next iteration posterior probability is $p_i^{t+1} = 1/ps$. The velocity of the ith particle is updated with the following equation:

$$v_{\boldsymbol{i,d}}^{\boldsymbol{t+1}} \leftarrow w v_{i,d}^t + c_1 r_1 (pbest_{fi(d)}^t - x_{i,d}^t) + c_2 r_2 (exemplar_d^t - x_{i,d}^t) \tag{12}$$

$$x_{\boldsymbol{i,d}}^{\boldsymbol{t+1}} \leftarrow x_{i,d}^t + v_{i,d}^{t+1} \tag{13}$$

According to Eq. (9), the particle with the largest posterior probability is used as an exemplar. Each dimension of the exemplar particle will attract the corresponding dimensions of all particles. In the tth iteration, the position value of the particle in a certain dimension, $x_{i,d}^t$, plus the velocity value of the corresponding dimension, $v_{i,d}^{t+1}$, is used as the position value of the particle in the next iteration, $x_{\boldsymbol{i,d}}^{\boldsymbol{t+1}}$.

The pseudo code of the BCLPSO algorithm is given in Algorithm 1, and the meaning of each symbol is listed in Table 1.

3.3. BCLPSO search mechanism

The above operations can enhance the search space, and the historical data of particles close to the global optimum are fully used in the iterative process. In the original PSO, for a certain multimodal function after t iterations, if the current optimal position is a local optimal position ($gbest_t$), then the next iteration of particles will be attracted by the local optimum, and the subsequent iterations will converge to the local optimum region. This process is presented in Fig. 1, where $gbest_t$ is the current optimum of the swarm after t iterations; $pbest_t$ is the best previous position of the particle itself; *global optimum* is the optimal value of the objective function; X_t is the current location of a certain particle (the particle may be a potential particle that can search for the optimal location); v_t^g represents the velocity component attracted by$gbest_t$ in the next iteration; v_t^p is attracted by $pbest_t$; v_t and v_{t+1} represent the particle inertial velocity and synthesis velocity after the tth iteration, respectively. As shown in Fig. 1, the potential particle leaps from the global optimum and converges with the local optimum.

In the BCLPSO, however, if the current swarm optimal position is a local optimal position $exemplar_t$ at the tth iteration for a certain particle, then the current position (X_t) is a potential particle that can search for the optimal location. Its posterior probability ($p_t = a$) is high but not the largest; $exemplar_t$ has the largest posterior probability. This is shown in Fig. 2(a), where $pbest_{fi}^t$ represents the particle's $pbest_t$ that should follow the potential particle; a detailed explanation is found in Ref. [19]. Suppose that the posterior probability of the particle increases to b ($p_{t+1} = b > a$) in the ($t + 1$)th iteration and is the highest among all particles. The particle, therefore, evolves into a swarm exemplar in the ($t + 1$)th iteration. As described in Fig. 2(b), the direction of synthesis velocity (v_{t+1}) will deviate from the local optimum and eventually escape from the local optimum region. This will considerably enhance the exploration performance of particles and expand the particle search space. The search time of this method, however, will inevitably be prolonged, conforming to the no-free-lunch theorem.

4. Numerical investigation

4.1. Experimental preparation

To test the BCLPSO performance, the CEC 2017 benchmark functions are employed to evaluate its search behavior, which is composed of all types of unimodal, multimodal, hybrid, and composition functions [43]. The global optimum values of all these benchmark functions are summarized in Table 2. Some state-of-the-art PSO algorithms are compared with the proposed algorithm to evaluate the latter's performance. The algorithm and parameter settings are listed as follows:

- PSO with inertia weight (PSO-w) [7];
- FDR-PSO [16];
- Comprehensive learning PSO (CLPSO) [19];
- Heterogeneous comprehensive learning PSO (HCLPSO) [21];
- Ensemble particle swarm optimizer [33];
- Global genetic learning particle swarm optimization PSO (GGL-PSOD) [32];
- Non-inertial opposition-based particle swarm optimization (NOPSO) [27]
- Deterministic PSO (DPSO) [13]
- BCLPSO.

The PSO inertia weight develops ω to balance the exploration and exploitation abilities [7]. In the FDR-PSO algorithm, the velocity update selects another particle's *nbest*, which has a higher fitness value and is closer to the particle being updated [16]. The CLPSO uses the learning probability curve and CL strategy to maintain swarm diversity and prevent premature convergence [19]. In the HCLPSO, all particles are divided into two subpopulations—exploration and exploitation capacities, which are assigned to each subpopulation [21]. The EPSO [33] assembles different optimization algorithms to solve complex problems. In the GGL-PSOD, the global learning component is combined with linearly adjusted control parameters (e.g., inertia weight and acceleration coefficients) and genetic learning particle swarm optimization (GL-PSO); the resultant GGL-PSOD yields a better performance [32]. It is important to choose the proper parameters for PSO, and the ones for all compared algorithms are consistent with their original literatures [27]. The DPSO develops a deterministic particle swarm optimization algorithm with fixed random factors to improve the computational efficiency [13]. Each algorithm is tested on all the 30 benchmark functions and run 30 times. The swarm particle size is set to 40, and the number of maximum function evaluations (FEs) is 300 000. The details of all BCLPSO parameters and other algorithms are summarized in Table 3. The refresh gap (m) is set to 5 in this study.

Algorithm 1 The pseudo code of BCLPSO algorithm

```
Input   The initial position of particle, X_i = [x_{i,1}, x_{i,2}, x_{i,3}, ..., x_{i,D}], and the velocity of particle
          V_i = [v_{i,1}, v_{i,2}, v_{i,3}, ..., v_{i,D}].
Output  The whole particle's best position.
Initialization X_i, Set V_i=0, p_i^0=1/ps, max_fes, max_iteration, ps, D, m;
  k=1, t=1
  For i=1 to ps do
    Update V_i and X_i according to eq.(4) , (2)
    Update p_i^t and exemplar^t according to eq.(9) , (10)
    k=k+1
  End for
  While k<=max_fes&&t<=max_iteration
    For i=1 to ps do
      Update V_i and X_i according to eq.(12) , (13)
      k=k+1;
      If fit(X_i)<Pbestval_i
        Stage(i)=0;
        Pbestval_i= fit(X_i)
        If fit(X_i)< Gbestval
          Gbestval= fit(X_i)
        End if
       Else
          Stage(i)= Stage(i)+1
          If Stage(i)>m
            Update Pc_i according to eq.(5)
            Update CL exemplar Pbest^t_fi
            Stage(i)=0
          End if
      End if
    End for
    t=t+1
    Update p_i^t and exemplar^t according to eq.(9) , (10)
  End while
```

Table 1
The meaning of each symbol.

Symbol	Meaning	Symbol	Meaning
k	Current function evolution (FEs)	ps	Population size
t	Current iteration	m	Refreshing gap
i	Particle's id counter	D	Dimension
max_fes	Maximum function evolutions	*max_iteration*	Maximum iterations
X_i	ith particle's position	V_i	ith particle's velocity
$Pbestval_i$	ith particle's best previous position	*Gbestval*	The whole particle's best position
Stag(i)	The successive iteration number without improving the fitness value of particle i		

4.2. Numerical investigation and comparisons

In this section, the proposed algorithm is compared with other state-of-the-art PSO algorithms, and the performance of every algorithm is measured and ranked according to mean error and standard deviation values in 30 runs. The final rank of each algorithm is determined according to the average rank value of the 30 benchmark functions. In order to test the statistical difference between the BCLPSO algorithm and other PSO algorithms, the non-parametric Wilcoxon signed-rank test is employed [48–51]. The symbol (+) indicates that the BCLPSO performs significantly better than the compared algorithm, whereas (-) means the converse. The symbol (≈) indicates that the BCLPSO performance and that of the compared algorithm are approximately the same. The value of p denotes the probability of rejecting the original hypothesis in the Wilcoxon test.

(a) Results on 10-dimensional problems

Table 4 lists the mean errors and variances of eight algorithms on the CEC 2017 benchmark functions with 10 dimensional problems. The best values of all eight algorithms are bold highlighted.

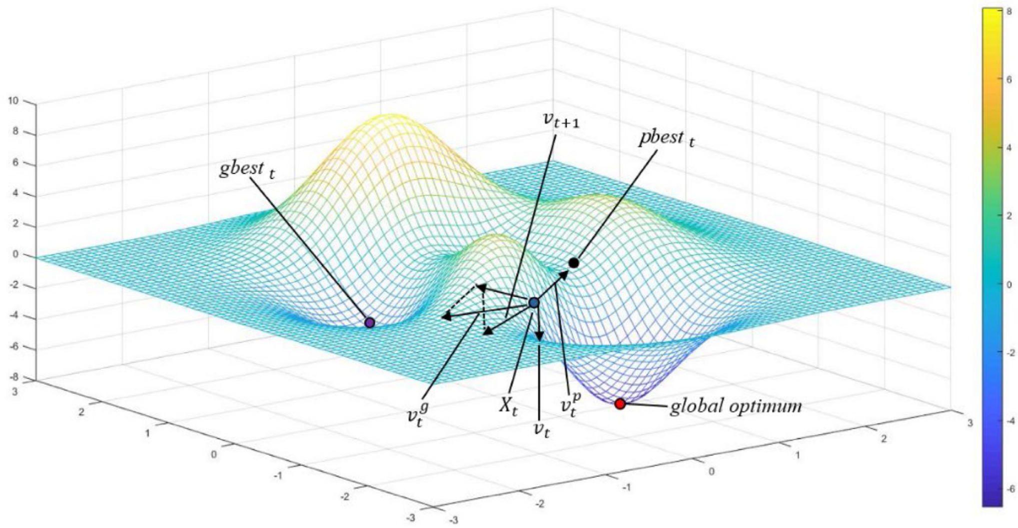

Fig. 1. The original PSO may fall into local optimum in complex multimodal problems.

Table 2
CEC 2017 test functions.

Function type	Function	$f(x^*)$
Unimodal functions	F1: Shifted and Rotated Bent Cigar Function	100
	F2: Shifted and Rotated Sum of Different Power Functions	200
	F3: Shifted and Rotated Zakharov Function	300
Multimodal functions	F4: Shifted and Rotated Rosenbrock Function	400
	F5: Shifted and Rotated Rastrigin Function	500
	F6: Shifted and Rotated Expanded Scaffer F6 Function	600
	F7: Shifted and Rotated LunacekBi_Rastrigin Function	700
	F8: Shifted and Rotated Non-Continuous Rastrigin Function	800
	F9: Shifted and Rotated Levy Function	900
	F10: Shifted and Rotated Schwefel Function	1000
Hybrid functions	F11: Hybrid Function 1 (N = 3)	1100
	F12: Hybrid Function 2 (N= 3)	1200
	F13: Hybrid Function 3 (N =3)	1300
	F14: Hybrid Function 4 (N = 4)	1400
	F15: Hybrid Function 5 (N = 4)	1500
	F16: Hybrid Function 6 (N = 4)	1600
	F17: Hybrid Function 7 (N = 5)	1700
	F18: Hybrid Function 8 (N = 5)	1800
	F19: Hybrid Function 9 (N = 5)	1900
	F20: Hybrid Function 10 (N = 6)	2000
Composition functions	F21: Composition Function 1 (N = 3)	2100
	F22: Composition Function 2 (N = 3)	2200
	F23: Composition Function 3 (N = 4)	2300
	F24: Composition Function 4 (N = 4)	2400
	F25: Composition Function 5 (N = 5)	2500
	Composition Function 6 (N = 5)	2600
	Composition Function 7 (N = 6)	2700
	Composition Function 8 (N = 6)	2800
	Composition Function 9 (N = 3)	2900
	Composition Function 10 (N = 3)	3000

Note: Search Range is $[-100,100]^D$; $f(x^*)$is the global optimum.

DPSO does not include comparison, because DPSO is a deterministic particle swarm optimization algorithm, which always provides the same result, so it is impossible to calculate the mean value and standard deviation. Based on the list in Table 4, the BCLPSO algorithm exhibits the best performances in terms of the CEC2017 test functions, especially on the unimodal, multimodal, and hybrid functions. On unimodal functions, most algorithms perform well and achieve a zero error on f_2 and f_3. The proposed BCLPSO achieves the best performance on f_1, f_2, and f_3. On the other hand, the FDR-PSO, CLPSO, HCLPSO, EPSO, and NOPSO perform well on f_2 and f_3.

On multimodal functions, the BCLPSO performs best on f_4, f_5, f_6, f_7, and f_9, and ranks second on the other functions. The CLPSO best performs on f_6 and f_9. The HCLPSO yields the best solution on f_8 and f_{10}, and EPSO best performs on f_9.

On hybrid functions, the BCLPSO performs well and achieves the best solutions on functions f_{11}, f_{13}, f_{15}, f_{16}, f_{19}, and f_{20}. On the other hand, the HCLPSO achieves the best performance on functions f_{12}, f_{14}, f_{17}, and f_{18}.

On composition functions, the BCLPSO consistently performs well and obtains the best solutions on f_{24}, f_{25}, f_{27}, and f_{30}; it ranks fourth on the other functions. The HCLPSO achieves the best performance on f_{21}, f_{22} and f_{29}. The EPSO provides the best

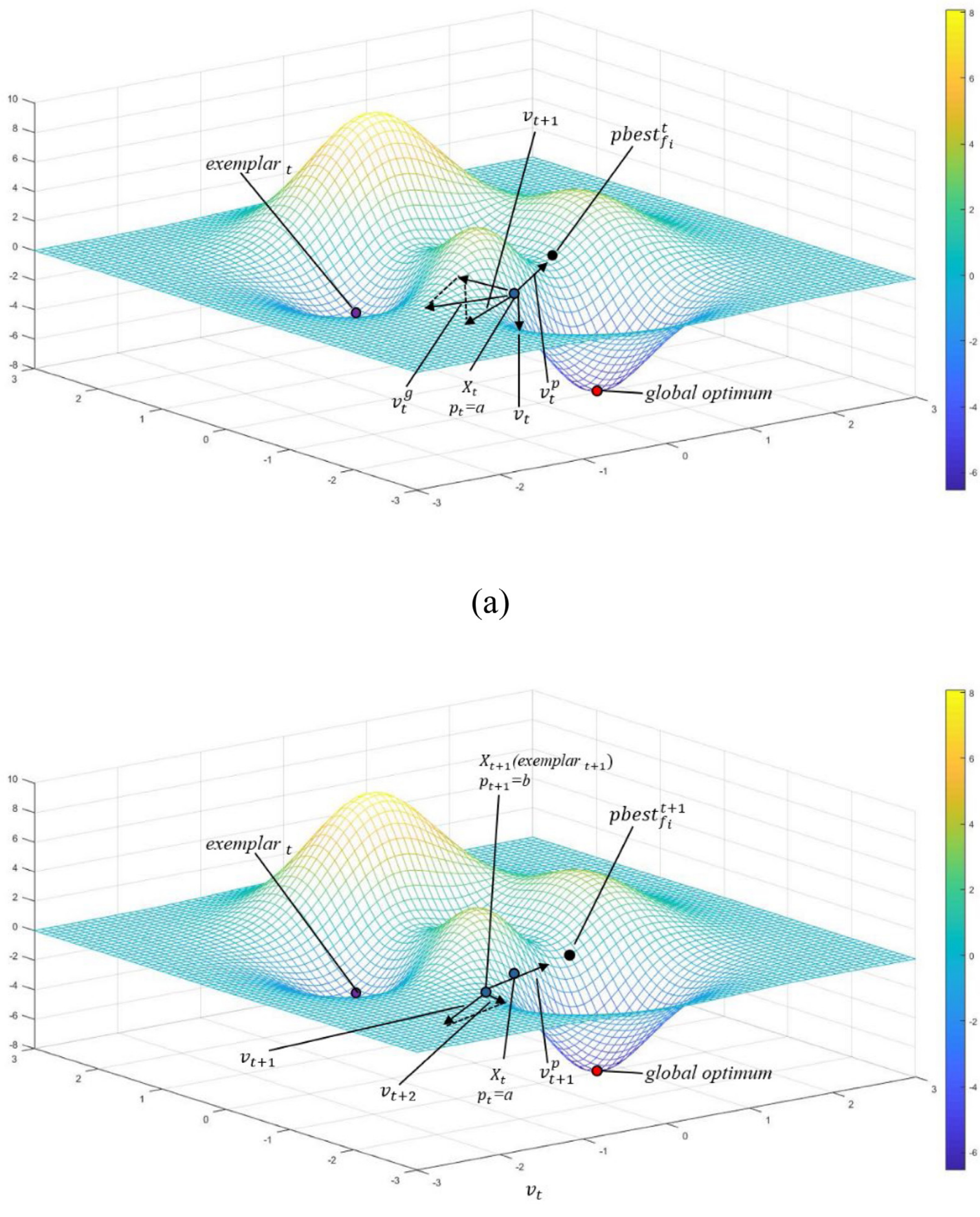

Fig. 2. The BCLPSO can escape from local optimum in complex multimodal problems. (a) at iteration t. (b) at iteration t+1.

Table 3
Parameter settings of PSO algorithms.

Parameter	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO	BCLPSO
Inertia weight w	0.729	0.9–0.5	0.9–0.4	0.9–0.2	Ref. [33]	0.9–0.4	–	0.9–0.4	0.9–0.4
Constriction factor χ	0.721	–	–	–		–	0.721	0.721	–
Acceleration coefficients c	2	1.4944	1.4944	$c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5, $c = 3$–1.5		$c_1 = 2.5$–0.5 $c_2 = 0.5$–2.5	1.496	1.655	$c_1 = 2.5$–0.5 $c_2 = 0.5$–2.5
Swarm size g	40	40	40	$g_1 = 15$, $g_2 = 25$.		40	40	40	40
Evaluations (FEs)	300000	300000	300000	300000		300000	300000	300000	300000

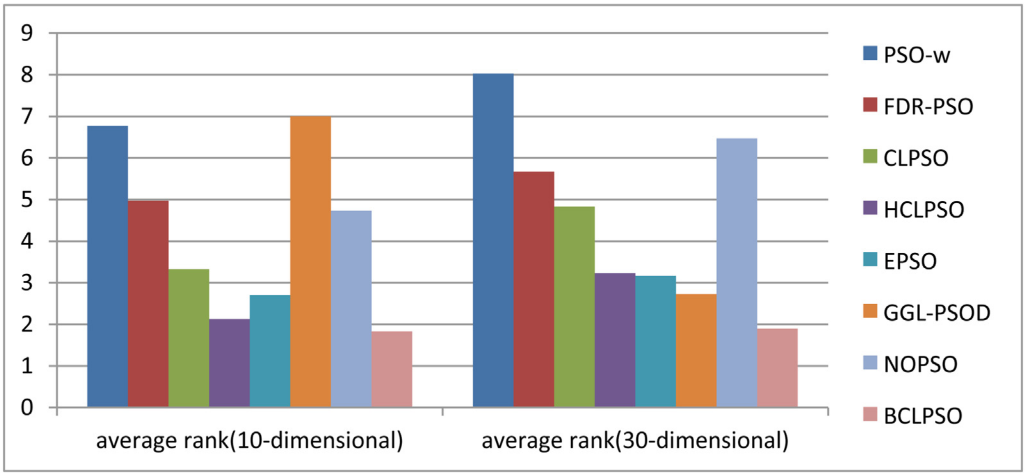

Fig. 3. The average rank of eight PSO variants on mean error.

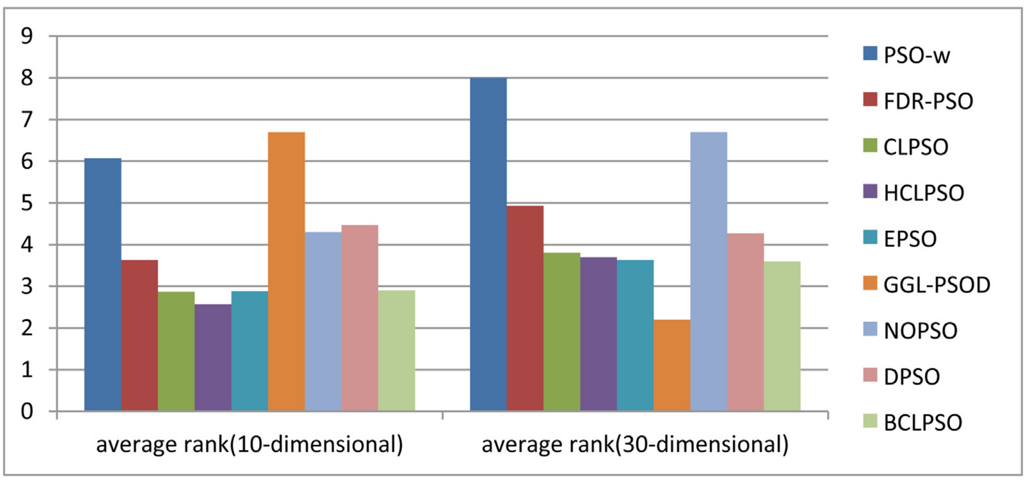

Fig. 4. The average rank of nine PSO variants on best optimum error.

solutions on f_{26} and f_{28}, and NOPSO best performs on f_{23}. As presented in the Best/2nd Best/Worst category, the proposed BCLPSO algorithm achieves the best performance for 18 times and has no "Worst" rating on any benchmark functions. Moreover, the BCLPSO consistently performs well on multimodal functions, hybrid functions, and composition functions. Therefore, it can provide an optimal approach to solve complex practical problems.

However, from an engineering point of view, if an algorithm can find the optimal value, even if it not provides the best average value, the algorithm should also have a certain application value in engineering [13]. Therefore, it is necessary to compare best optimum values of the nine algorithms. Here, Hammersley sequence sampling is used for deterministic initialization of DPSO [52]. Table 5 and Table 8 list the best optimum errors of nine algorithms on the CEC 2017 benchmark functions with 10-dimensional and 30-dimensional, respectively. As shown in the last line of Table 5, the HCLPSO achieves the best ranking, followed by CLPSO algorithm, and EPSO achieved the third. Although the BCLPSO algorithm can only achieve the fourth in the best optimum value ranking, it is similar to the above three algorithms and achieves the best rank in the mean value ranking. Compared with Table 5 and Table 8, the performance of BCLPSO has been relatively stable in both 10-dimensional and 30-dimensional problems.

The 10-dimensional Wilcoxon signed-rank test results of BCLPSO and the other eight algorithms are summarized in Table 6. The last line of the Wilcoxon table provides the number of (+/≈/−) marks obtained by the BCLPSO, indicating that its performance is evidently better than those of the other eight algorithms. Compared with the PSO-w algorithm, the BCLPSO "wins" on 28 functions, is "tied" on 2 functions, and has no "loss". Compared with the FDR-PSO, the BCLPSO "wins" on 20 functions and is "tied" on 10 functions. Compared with the CLPSO and HCLPSO (i.e., state-of-the-art PSOs based on comprehensive learning strategy), the BCLPSO "wins" on 17 and 12 functions, respectively. Compared with the recently reported EPSO, GGL-PSOD, NOPSO, and DPSO, the BCLPSO "wins" on 18 functions, 28 functions, 24 functions, and 18 functions, respectively.

(b) Results on 30-dimensional problems

As summarized in Table 7, the BCLPSO algorithm achieves the best performances on the CEC2017 test functions, particularly on multimodal, hybrid, and composition functions.

On the unimodal functions, the proposed algorithm, BCLPSO, performs well on function f_1, the EPSO achieves the best performance on f_2, and the GGL-PSOD has the best performance on f_3.

On multimodal functions, the EPSO has the best performance on f_4, and the GGL-PSOD obtains the best solution on f_5, f_8, and f_9. The proposed algorithm, BCLPSO, achieves the best performance on functions f_6, f_7, and f_{10}, ranks second on f_5, f_8, and f_9. The results demonstrate that it consistently performs well.

On hybrid test functions, the BCLPSO performs best on f_{11}, f_{13}, f_{14}, f_{15}, f_{19}, and f_{20}. The GGL-PSOD performs best on f_{16}, f_{17},

Table 4
Mean errors and variances on 10 dimensions.

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	BCLPSO
f_1	Mean	1.78E+03	1.20E+03	1.31E+03	53.15	300.04	3.08E+03	632.3	**34.73**
	Std	2.11E+03	1.46E+03	2.12E+03	102.42	244.86	3.58E+03	820.9	41.92
	Rank	7	5	6	2	3	8	4	1
f_2	Mean	5.89E+03	**0**	**0**	**0**	**0**	1.45E+04	**0**	**0**
	Std	6.69E+03	0	0	0	0	5.27E+04	0	0
	Rank	2	1	1	1	1	3	1	1
f_3	Mean	1.90	**0**	**0**	2.84E−14	**0**	1.50E−08	1.25E−13	**0**
	Std	4.44	0	0	2.89E−14	0	8.09E−08	7.37E−14	0
	Rank	5	1	1	2	1	3	4	1
f_4	Mean	14.82	6.53E−02	0.14	0.54	5.01E−02	58.48	0.42	**8.01E−02**
	Std	20.71	3.36E−02	0.04	0.53	3.75E−02	1.79	0.44	7.67E−02
	Rank	7	3	4	6	2	8	5	1
f_5	Mean	28.22	6.96	5.54	3.56	4.76	31.44	9.22	**3.22**
	Std	10.69	3.15	2.54	1.27	1.49	10.68	2.79	1.16
	Rank	7	5	4	2	3	8	6	1
f_6	Mean	4.53	3.79E−15	**0**	2.65E−14	2.65E−14	2.97E−07	**0**	**0**
	Std	6.13	2.08E−14	0	4.89E−14	4.89E−14	8.89E−07	0	0
	Rank	5	2	1	3	3	4	1	1
f_7	Mean	31.80	16.99	16.18	14.22	16.24	64.47	15.85	**12.06**
	Std	8.72	3.71	4.31	1.44	2.75	12.75	2.31	1.40
	Rank	7	6	4	2	5	8	3	1
f_8	Mean	19.58	5.27	5.31	**3.59**	5.16	32.03	5.64	3.79
	Std	8.79	2.27	2.30	1.34	2.11	8.47	2.24	1.01
	Rank	7	4	5	1	3	8	6	2
f_9	Mean	46.32	3.79E−15	**0**	1.14E−14	**0**	0.18	0.05	**0**
	Std	95.09	2.08E−14	0	3.47E−14	0	0.36	0.19	0
	Rank	6	2	1	3	1	5	4	1
f_{10}	Mean	869.73	285.78	111.27	**68.33**	131.23	1.97E+03	264.22	86.47
	Std	328.67	152.84	111.21	81.13	89.72	5.50E+02	132.39	69.70
	Rank	7	6	3	1	4	8	5	2
f_{11}	Mean	47.75	2.76	2.00	1.40	2.27	38.60	3.42	**0.28**
	Std	30.46	1.70	1.42	1.12	1.29	31.52	2.21	0.14
	Rank	8	5	3	2	4	7	6	1
f_{12}	Mean	1.27E+04	1.12E+04	7.71E+03	**5.63E+03**	9.52E+03	2.12E+04	7.20E+03	7.98E+03
	Std	1.10E+04	9.22E+03	5.84E+03	3.45E+03	6.21E+03	1.01E+04	6.86E+03	5.88E+03
	Rank	6	5	3	1	4	7	2	4
f_{13}	Mean	7.13E+03	4.05E+03	3.08E+03	172.24	175.21	1.28E+04	3.50E+03	**104.88**
	Std	6.46E+03	3.68E+03	1.96E+03	211.78	402.96	1.18E+04	2.60E+03	99.17
	Rank	7	6	4	2	3	8	5	1
f_{14}	Mean	769.28	26.89	16.82	**10.51**	15.52	3.42E+03	358.11	25.70
	Std	381.35	12.22	10.85	9.97	10.19	4.74E+03	478.43	18.29
	Rank	7	5	3	1	2	8	6	4
f_{15}	Mean	1.12E+03	7.97	4.70	7.18	6.82	4.02E+03	177.41	**3.71**
	Std	1.37E+03	7.57	3.43	5.49	6.25	8.64E+03	307.09	3.76
	Rank	7	5	2	4	3	8	6	1
f_{16}	Mean	207.68	105.79	0.47	0.44	0.46	377.90	39.75	**0.42**
	Std	124.84	117.90	0.25	0.15	0.30	261.69	66.03	0.14
	Rank	7	5	4	2	3	8	6	1
f_{17}	Mean	81.90	32.91	8.67	**1.01**	4.32	59.86	20.58	2.02
	Std	59.03	36.25	9.56	0.65	3.86	49.75	11.66	4.23
	Rank	8	6	4	1	3	7	5	2
f_{18}	Mean	4.65E+03	2.01E+03	2.79E+03	**502.52**	820.43	6.99E+04	4.09E+03	1.11E+03
	Std	6.86E+03	2.52E+03	3.16E+03	378.63	650.77	5.69E+04	3.32E+03	8.54E+02
	Rank	7	4	5	1	2	8	6	3
f_{19}	Mean	3.18E+03	6.44	3.41	4.23	3.47	6.04E+03	257.14	**2.30**
	Std	3.06E+03	4.52	2.49	6.25	2.02	5.60E+03	517.32	1.61
	Rank	7	5	2	4	3	8	6	1
f_{20}	Mean	99.53	52.64	6.37	6.63E−02	3.12E−02	101.31	12.69	**2.08E−02**
	Std	57.73	60.33	9.19	2.52E−01	9.53E−02	59.14	10.01	7.92E−02
	Rank	7	6	4	3	2	8	5	1
f_{21}	Mean	200.49	198.04	125.12	**107.16**	107.32	233.23	151.84	111.32
	Std	52.28	33.34	46.13	26.97	27.64	8.34	56.43	32.20
	Rank	7	6	4	1	2	8	5	3
f_{22}	Mean	103.08	98.87	93.57	**60.05**	62.26	100	99.19	72.76
	Std	3.03	16.51	24.31	41.85	38.46	1.15E−13	14.15	32.68
	Rank	8	5	4	1	2	7	6	3

(*continued on next page*)

Table 4 (*continued*).

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	BCLPSO
f_{23}	Mean	344.99	308.00	296.60	294.53	298.63	382.32	**291.35**	305.71
	Std	21.84	4.20	56.08	55.65	37.14	12.35	79.32	2.22
	Rank	7	6	3	2	4	8	1	5
f_{24}	Mean	352.32	317.57	194.95	125.81	141.83	453.77	219.50	**121.98**
	Std	79.16	64.27	118.29	64.60	85.05	13.07	118.92	54.84
	Rank	7	6	4	2	3	8	5	1
f_{25}	Mean	416.22	415.57	405.76	379.04	372.92	387.22	417.37	**370.71**
	Std	64.25	22.61	17.55	72.41	77.37	0.12	23.09	71.29
	Rank	7	6	5	3	2	4	8	1
f_{26}	Mean	332.25	414.33	300	196.65	**190.67**	946.21	283.25	201.46
	Std	244.06	317.51	1.19E−13	104.32	106.29	516.95	202.76	79.57
	Rank	6	7	5	2	1	8	4	3
f_{27}	Mean	429.66	402.34	391.45	390.15	390.43	503.20	401.75	**389.27**
	Std	33.59	23.89	2.62	1.95	1.79	6.80	11.04	2.10
	Rank	7	6	3	2	4	8	5	1
f_{28}	Mean	520.15	439.90	300	290	**278.66**	349.13	338.84	282.39
	Std	110.82	152.42	1.46E−13	54.77	112.91	62.03	150.56	67.04
	Rank	8	7	4	3	1	6	5	2
f_{29}	Mean	353.66	280.30	245.63	**241.66**	247.76	467.74	264.08	249.05
	Std	74.14	42.85	9.56	5.98	10.91	35.56	15.08	7.93
	Rank	7	6	2	1	3	8	5	4
f_{30}	Mean	6.47E+05	3.31E+05	1.68E+03	2.17E+03	2.30E+03	3.56E+03	1.17E+05	**1.10E+03**
	Std	1.48E+06	4.90E+05	8.14E+02	7.99E+02	2.01E+03	1.41E+03	3.56E+05	9.39E+02
	Rank	8	7	2	3	4	5	6	1
Average rank		6.77	4.97	3.33	2.13	2.70	7.00	4.73	1.83
Best/2nd Best/Worst		0/1/8	2/2/0	4/4/0	10/11/0	5/7/0	0/0/21	3/1/1	18/4/0
Algorithms		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	BCLPSO

and f_{18}. The CLPSO has the best performance on f_{12}. The BCLPSO consistently performs well and is consistently in the top 3 on all 10 functions.

On composition functions, the BCLPSO performs best on $f_{21}, f_{23}, f_{25}, f_{27}, f_{28}, f_{29}$, and f_{30}. The GGL-PSOD achieves the best solution on f_{22}. The EPSO and HCLPSO have the best performances on f_{24} and f_{26}, respectively.

As presented in the Best/2nd Best/Worst category, the BCLPSO ranks first as it performs best on 17 out of 30 problems. It also consistently performs well on other problems. As shown in the last line of Table 8, the GGL-PSOD achieves the best ranking, followed by BCLPSO algorithm, and EPSO achieved the third. However, the PSO-w algorithm gets the worst ranking in both 10-dimensional and 30-dimensional optimal error ranking. In addition, the last row in the Wilcoxon signed-rank test, Table 9, indicates that the proposed BCLPSO algorithm performs best among all algorithms. However, the GG-PSOD and DPSO algorithms also achieve good performance.

The average rank of all PSO algorithms on mean error and best optimum error are presented in Fig. 3 and 4, respectively. The Fig. 3 shows the proposed BCLPSO algorithm is ranked the best in both 10 and 30 dimensions. HCLPSO and EPSO algorithms also have good performance in 10 and 30 dimensions. The GGL-PSOD achieves good performance in high dimension, but poor performance in low dimension. The Fig. 4 shows BCLPSO algorithm ranks fourth and second on 10 and 30 dimensions respectively, while HCLPSO and GGL-PSOD rank first respectively, but there was no significant difference between BCLPSO and the best.

(c) Comparison of the methods using sign test and Friedman test

The statistical comparison of the methods using sign test is given in Table 10. As summarized in Table, the proposed BCLPSO algorithm achieves the best performance among all algorithms. DPSO achieves good performance in 30 dimensions, in addition, GGL-PSOD also has good performance.

Considering that the Wilcoxon signed test is only a pairwise comparison method, and the Friedman test should be used to simultaneously compare all datasets and all algorithms for providing more solid evidence. Referring to [53], all algorithms are compared by Friedman test. Friedman test results in Table 11 show that the BCLPSO obtains best Friedman rank, and the p-value is smaller than 0.05, means the performance of BCLPSO is significant different than nine peer algorithms.

(d) Comparison of mean error, diversity and robustness

The swarm diversity is employed to determine whether a population is being explored or exploited [54]. When the swarm has higher diversity, the search space is larger, though the swarm has a strong global search ability, the search time will prolong; on the contrary, when the swarm has low diversity, the search space is small, the swarm has strong local search ability, but it easily falls into local optimum.

$$Diversity(t) = \frac{1}{ps}\sum_{i=1}^{ps}\sqrt{\sum_{d=1}^{D}(x_i^d(t) - \overline{x^d(t)})^2} \tag{14}$$

$$\overline{x^d(t)} = \frac{\sum_{i=1}^{ps} x_i^d(t)}{ps}, \tag{15}$$

where ps is the swarm size; D is the dimensionality of the problem; $\overline{x^d(t)}$ denotes the dth dimension of the mean position.

Without the loss of generality, in this study, the diversity of the proposed PSO algorithm is compared with other state-of-the-art PSO algorithms. It is investigated in one hybrid problem (f_{11}: Hybrid Function 1 (N = 3)) and one composition problem (f_{30}: Composition Function 10 (N = 3)) on 30 dimensions. The functions are derived from the CEC2017 benchmark functions. The maximum number of *FEs* is set to 300 000, and every function is run 30 times. The mean error and diversity are shown in Figs. 5 and 6, respectively. Error is the optimal value found by the algorithm minus the global optimum of the function.

Fig. 5((a) and (b)) shows the convergence curve of hybrid function 1 (N = 3) and composition function 10 (N = 3), respectively. The convergence rate of the proposed BCLPSO algorithm is generally in the early evolutionary stage; however, its rate is better in the later evolutionary stage of the search. By maintaining a

Table 5
The best optimum errors on 10 dimensions (run 30 times).

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO	BCLPSO
f_1	Optimum	2.33	2.58	0.01	6.36E−03	0.05	1.39	3.50	0.91	9.08E−04
	Rank	7	8	3	2	4	6	9	5	1
f_2	Optimum	3	0	0	0	0	0	0	1.13	0
	Rank	3	1	1	1	1	1	1	2	1
f_3	Optimum	5.68E−14	0	0	0	0	2.16E−12	0	3.93E−15	0
	Rank	3	1	1	1	1	4	1	2	1
f_4	Optimum	0.11	0.02	0.08	0.13	0.02	50.64	6.36E−05	0.18	9.15E−03
	Rank	6	4	5	7	3	9	1	8	2
f_5	Optimum	10.94	0.99	1.99	0.99	1.99	16.91	1.99	1.58	0.02
	Rank	6	2	4	2	4	7	5	3	1
f_6	Optimum	2.59E−03	0	0	0	0	1.14E−13	0	0.88	0
	Rank	3	1	1	1	1	2	1	4	1
f_7	Optimum	18.85	7.51	3.60	11.12	11.38	47.91	11.52	2.23	9.78
	Rank	8	3	2	5	6	9	7	1	4
f_8	Optimum	2.98	0.99	0.99	0.99	1.99	11.94	1.00	0.30	2.00
	Rank	6	2	2	2	4	7	3	1	5
f_9	Optimum	4.13E−10	0	0	0	0	0	0	0.28	0
	Rank	2	1	1	1	1	1	1	3	1
f_{10}	Optimum	2.77E+02	6.89	0.12	0.31	11.89	9.48E+02	3.54	58.77	0.44
	Rank	8	5	1	2	6	9	4	7	3
f_{11}	Optimum	8.17	2.06E−07	7.38E−06	1.12E−03	4.32E−08	10.14	4.58E−03	1.71	9.06E−03
	Rank	8	2	3	4	1	9	5	7	6
f_{12}	Optimum	1.44E+03	8.21	2.19E+02	3.67E+02	7.31E+02	6.96E+03	6.64E+02	1.50E+02	1.05E+03
	Rank	8	1	3	4	6	9	5	2	7
f_{13}	Optimum	1.43E+02	80.84	15.84	6.65	13.24	21.08	83.45	92.72	9.84
	Rank	9	6	4	1	3	5	7	8	2
f_{14}	Optimum	45.91	6.26	0	2.48	3.05	2.77E+02	9.95	2.64	0.05
	Rank	8	6	1	3	5	9	7	4	2
f_{15}	Optimum	72.62	1.20	1.03	1.65	1.15	48.29	7.10	9.04	0.26
	Rank	9	4	2	5	3	8	6	7	1
f_{16}	Optimum	1.83	0.02	0.02	9.27E−03	5.63E−03	10.52	0.73	5.41	0.16
	Rank	7	3	4	2	1	9	6	8	5
f_{17}	Optimum	25.88	1.33	0.33	0.02	0.34	9.78	2.42	5.08	0.03
	Rank	9	5	3	1	4	8	6	7	2
f_{18}	Optimum	1.43E+02	17.38	1.01E+02	21.67	28.87	6.12E+03	3.64E+02	41.24	1.18E+02
	Rank	7	1	5	2	3	9	8	4	6
f_{19}	Optimum	8.48	0.85	0.73	0.48	0.15	17.50	2.08	0.91	0.17
	Rank	8	5	4	3	1	9	7	6	2
f_{20}	Optimum	22.30	0	0	0	2.27E−13	11.06	0	5.41	0
	Rank	5	1	1	1	2	4	1	3	1
f_{21}	Optimum	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	2.18E+02	1.00E+02	10.37	92
	Rank	4	3	3	3	3	5	3	1	2
f_{22}	Optimum	100.28	11.56	4.55E−13	4.55E−13	9.09E−13	1.00E+02	24.39	5.94	19.57
	Rank	8	4	1	1	2	7	6	3	5
f_{23}	Optimum	3.08E+02	3.03E+02	4.55E−13	9.09E−13	1.02E+02	3.59E+02	0	32.19	3.00E+02
	Rank	8	7	2	3	5	9	1	4	6
f_{24}	Optimum	1.00E+02	1.00E+02	1.00E+02	1.00E+02	23.21	4.30E+02	1.00E+02	10.26	1.00E+02
	Rank	3	3	3	3	2	4	3	1	3
f_{25}	Optimum	1.00E+02	3.98E+02	3.98E+02	1.13E+02	1.20E+02	3.87E+02	3.98E+02	3.99E+02	1.00E+02
	Rank	1	6	5	2	3	4	7	8	1
f_{26}	Optimum	4.55E−13	2.00E+02	3.00E+02	0.30	4.55E−13	2.00E+02	4.55E−13	4.49	9.43
	Rank	1	5	6	2	1	5	1	3	4
f_{27}	Optimum	3.98E+02	3.89E+02	3.87E+02	3.87E+02	3.87E+02	4.86E+02	3.90E+02	3.99E+02	3.86E+02
	Rank	6	4	2	3	2	8	5	7	1
f_{28}	Optimum	3.00E+02	3.00E+02	3.00E+02	8.41E−09	4.55E−13	3.00E+02	0	31.93	30.11
	Rank	6	6	6	3	2	7	1	5	4
f_{29}	Optimum	2.71E+02	2.37E+02	2.31E+02	2.30E+02	2.32E+02	4.20E+02	2.34E+02	2.84E+02	2.38E+02
	Rank	7	5	2	1	3	9	4	8	6
f_{30}	Optimum	1.86E+03	7.02E+02	7.11E+02	1.04E+03	6.90E+02	2.06E+03	1.37E+03	1.09E+02	1.02E+02
	Rank	8	4	5	6	3	9	7	2	1
Average rank		6.07	3.63	2.87	2.57	2.88	6.70	4.30	4.47	2.90

convergence trend in the later stage, it does not facilely fall into the local optimum. This is because the BCLPSO algorithm determines the social learning exemplar, which exploits the historical

Table 6
Wilcoxon test results on 10 dimensions (significance level: $\alpha = 0.05$).

Function		‘Pairwise comparison BCLPSO versus							
		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO
$f(1)$	Symbol	+	+	+	≈	+	+	+	+
	p	5.46E−09	3.32E−06	1.25E−07	0.60	2.03E−07	5.97E−09	3.01E−07	1.61E−02
$f(2)$	Symbol	+	≈	≈	≈	≈	+	+	+
	p	1.21E−12	NaN	NaN	NaN	NaN	4.77E−08	1.87E−09	1.86E−09
$f(3)$	Symbol	+	+	+	+	+	+	≈	≈
	p	2.47E−10	5.74E−09	5.74E−09	5.31E−05	5.74E−09	8.12E−11	0.11	0.86
$f(4)$	Symbol	+	≈	+	+	≈	+	+	+
	p	1.78E−10	0.40	5.26E−04	1.78E−10	0.76	3.02E−11	6.77E−05	1.86E−09
$f(5)$	Symbol	+	+	+	+	+	+	+	+
	p	3.02E−11	1.40E−08	8.68E−07	1.98E−04	1.34E−07	3.02E−11	1.77E−10	3.25E−09
$f(6)$	Symbol	+	≈	≈	+	+	+	≈	+
	p	1.21E−12	0.33	NaN	5.47E−03	5.47E−03	7.77E−13	NaN	1.86E−09
$f(7)$	Symbol	+	+	+	+	+	+	+	−
	p	3.02E−11	2.60E−08	1.36E−07	1.86E−06	5.00E−09	3.02E−11	4.57E−09	1.86E−09
$f(8)$	Symbol	+	≈	+	≈	+	+	+	−
	p	3.81E−10	6.32E−02	3.36E−02	0.16	4.83E−02	3.01E−11	3.84E−03	1.61E−02
$f(9)$	Symbol	+	≈	≈	≈	≈	+	+	+
	p	1.21E−12	0.33	NaN	8.14E−02	NaN	3.84E−08	3.76E−09	1.86E−09
$f(10)$	Symbol	+	+	≈	≈	+	+	+	+
	p	3.02E−11	7.04E−07	0.81	0.19	3.03E−02	3.02E−11	4.12E−06	5.20E−03
$f(11)$	Symbol	+	+	+	+	+	+	+	+
	p	3.02E−11	8.48E−09	8.48E−09	4.22E−04	6.12E−10	3.02E−11	6.52E−09	1.86E−09
$f(12)$	Symbol	≈	≈	≈	≈	≈	+	≈	≈
	p	0.12	0.26	0.83	0.13	0.26	1.73E−07	0.38	0.86
$f(13)$	Symbol	+	+	+	≈	≈	+	+	+
	p	8.99E−11	3.16E−10	1.86E−09	0.63	0.78	1.69E−09	2.03E−09	8.68E−07
$f(14)$	Symbol	+	≈	≈	−	+	+	+	−
	p	6.07E−11	0.64	5.75E−02	4.71E−04	3.03E−02	3.02E−11	6.36E−05	5.20E−03
$f(15)$	Symbol	+	+	≈	+	+	+	+	+
	p	3.02E−11	2.38E−03	6.79E−02	3.18E−03	2.89E−03	3.02E−11	8.15E−11	1.86E−09
$f(16)$	Symbol	+	+	≈	≈	≈	+	+	+
	p	3.02E−11	2.53E−04	0.30	0.54	0.59	3.02E−11	3.02E−11	1.86E−09
$f(17)$	Symbol	+	+	+	≈	+	+	+	+
	p	3.02E−11	1.69E−09	8.65E−05	0.64	3.59E−05	4.50E−11	2.61E−10	8.68E−07
$f(18)$	Symbol	≈	≈	≈	−	≈	+	+	+
	p	0.07	0.31	6.35E−02	2.50E−03	0.18	3.02E−11	4.94E−05	0.58
$f(19)$	Symbol	+	+	≈	≈	+	+	+	≈
	p	3.02E−11	5.86E−06	0.18	0.26	4.21E−02	3.02E−11	1.55E−09	8.43E−06
$f(20)$	Symbol	+	+	+	+	+	+	+	+
	p	2.35E−11	3.35E−10	8.82E−04	4.10E−05	2.67E−05	2.35E−11	2.42E−09	1.86E−09
$f(21)$	Symbol	+	+	+	−	+	+	+	−
	p	2.92E−09	3.15E−10	5.68E−05	6.51E−04	4.71E−04	3.02E−11	1.60E−06	1.86E−09
$f(22)$	Symbol	+	+	+	≈	≈	≈	+	−
	p	1.69E−09	5.97E−09	2.43E−05	0.22	0.84	0.65	3.08E−08	5.77E−08
$f(23)$	Symbol	+	≈	≈	−	≈	+	−	−
	p	4.98E−11	5.94E−02	0.22	7.96E−03	0.27	3.02E−11	1.19E−06	1.86E−09
$f(24)$	Symbol	+	+	≈	+	+	+	≈	−
	p	1.10E−08	3.35E−08	0.14	4.14E−06	6.92E−04	3.02E−11	0.79	1.86E−09
$f(25)$	Symbol	+	+	+	+	+	+	+	−
	p	5.54E−10	2.95E−11	2.94E−11	5.91E−09	6.48E−08	1.06E−07	2.95E−11	1.86E−09
$f(26)$	Symbol	+	+	+	≈	≈	+	≈	−
	p	7.87E−03	8.84E−06	5.51E−07	0.86	9.11E−02	3.11E−05	0.13	5.95E−05
$f(27)$	Symbol	+	+	+	≈	+	+	+	+
	p	3.02E−11	2.37E−08	5.23E−04	5.19E−02	1.12E−02	3.02E−11	4.56E−09	1.86E−09
$f(28)$	Symbol	+	≈	+	+	−	≈	+	+
	p	4.61E−10	0.98	7.47E−10	2.50E−09	2.13E−07	0.98	6.28E−04	5.77E−08
$f(29)$	Symbol	+	+	≈	−	≈	+	+	+
	p	3.02E−11	7.20E−05	8.50E−02	6.91E−04	0.35	3.02E−11	3.59E−05	1.86E−09
$f(30)$	Symbol	+	+	+	+	+	+	+	+
	p	3.34E−11	8.87E−06	4.86E−03	1.36E−07	2.16E−03	3.02E−11	2.67E−09	8.68E−07
+/≈/−		28/2/0	20/10/0	17/13/0	12/13/5	18/11/1	28/2/0	24/5/1	18/3/9

Table 7
Mean errors and variances on 30 dimensions.

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	BCLPSO
f_1	Mean	7.14E+08	3.75E+03	4.31E+03	72.47	3.21E+03	3.84E+03	2.40E+03	**30.46**
	Std	7.60E+08	4.68E+03	4.42E+03	96.45	4.30E+03	4.34E+03	2.51E+03	74.40
	Rank	8	5	7	2	4	6	3	1
f_2	Mean	4.52E+29	8.48E+09	2.58E+05	1.75E+04	**6.13**	1.82E+04	1.41E+17	3.18E+04
	Std	1.36E+30	3.96E+10	1.12E+06	9.56E+04	13.53	9.56E+04	4.25E+17	7.16E+03
	Rank	8	6	5	2	1	3	7	4
f_3	Mean	4.64E+04	1.12E−06	1.40E−04	3.77E−03	8.07E−08	**9.42E−09**	8.20E+03	3.31E+03
	Std	1.75E+04	2.86E−06	5.10E−04	1.10E−02	1.80E−07	4.94E−08	3.33E+03	1.47E+03
	Rank	8	3	4	5	2	1	7	6
f_4	Mean	278.89	46.36	75.55	66.50	**14.26**	57.72	82.41	69.73
	Std	180.06	31.85	20.72	22.02	21.47	4.08	16.18	15.71
	Rank	8	2	6	4	1	3	7	5
f_5	Mean	163.40	52.20	43.81	39.05	46.55	**28.20**	84.35	33.28
	Std	34.78	12.06	12.28	7.70	13.86	9.59	19.07	4.85
	Rank	8	6	4	3	5	1	7	2
f_6	Mean	41.46	6.59E−02	1.20E−04	3.64E−13	1.32E−05	4.43E−07	8.23	**2.08E−13**
	Std	10.76	0.11	2.65E−04	1.21E−13	7.10E−05	9.47E−07	3.71	4.31E−14
	Rank	8	6	5	2	4	3	7	1
f_7	Mean	264.49	103.77	93.23	85.52	83.95	62.36	98.78	**61.99**
	Std	63.52	18.91	19.45	12.41	27.95	8.71	14.35	5.94
	Rank	8	7	5	4	3	2	6	1
f_8	Mean	142.09	54.16	42.37	42.25	48.12	**27.65**	68.52	36.12
	Std	33.47	10.45	10.03	8.56	13.60	6.84	14.59	6.03
	Rank	8	6	4	3	5	1	7	2
f_9	Mean	3.35E+03	17.41	14.34	13.09	8.91	**0.22**	612.52	2.53
	Std	8.66E+02	16.50	17.02	15.51	8.04	0.46	342.62	2.69
	Rank	8	6	5	4	3	1	7	2
f_{10}	Mean	4.01E+03	3.07E+03	2.75E+03	2.11E+03	2.43E+03	1.94E+03	2.95E+03	**1.93E+03**
	Std	7.26E+02	7.25E+02	6.00E+02	2.79E+02	4.01E+02	5.17E+02	6.55E+02	2.91E+02
	Rank	8	7	5	3	4	2	6	1
f_{11}	Mean	329.85	87.47	66.70	60.33	52.21	44.16	94.65	**23.22**
	Std	158.35	39.67	33.52	25.55	21.99	30.38	28.15	7.08
	Rank	8	6	5	4	3	2	7	1
f_{12}	Mean	1.33E+07	2.34E+04	**2.10E+04**	3.33E+04	2.28E+04	2.50E+04	1.46E+05	2.32E+04
	Std	1.34E+07	1.12E+04	1.20E+04	1.70E+04	8.22E+03	1.36E+04	1.60E+05	1.22E+04
	Rank	8	4	1	6	2	5	7	3
f_{13}	Mean	4.95E+04	1.41E+04	1.28E+04	537.37	4.21E+03	1.36E+04	8.48E+03	**181.75**
	Std	2.12E+04	1.42E+04	1.28E+04	310.88	3.90E+03	1.27E+04	6.40E+03	114.82
	Rank	8	7	5	2	3	6	4	1
f_{14}	Mean	3.12E+04	4.48E+03	4.19E+03	3.92E+03	2.90E+03	2.91E+03	1.03E+04	**1.29E+03**
	Std	3.04E+04	2.87E+03	3.12E+03	3.24E+03	3.09E+03	2.79E+03	1.19E+04	1.15E+03
	Rank	8	6	5	4	2	3	7	1
f_{15}	Mean	2.24E+04	5.26E+03	4.25E+03	475.78	541.17	2.57E+03	3.32E+03	**96.33**
	Std	1.60E+04	7.08E+03	6.58E+03	890.54	657.68	2.34E+03	4.36E+03	49.79
	Rank	8	7	6	2	3	4	5	1
f_{16}	Mean	1.34E+03	700.34	546.02	435.84	538.72	**305.44**	810.85	413.05
	Std	3.83E+02	278.74	234.39	209.88	126.91	277.71	192.46	120.65
	Rank	8	6	5	3	4	1	7	2
f_{17}	Mean	610.65	219.00	129.94	81.94	148.15	**53.19**	195.04	79.29
	Std	249.64	106.35	67.30	41.03	74.26	40.40	87.24	39.79
	Rank	8	7	4	3	5	1	6	2
f_{18}	Mean	4.67E+05	8.18E+04	1.05E+05	9.52E+04	9.98E+04	**7.61E+04**	1.22E+05	9.37E+04
	Std	9.34E+05	4.64E+04	8.00E+04	6.16E+04	9.00E+04	5.70E+04	7.74E+04	4.60E+04
	Rank	8	2	6	4	5	1	7	3
f_{19}	Mean	3.67E+04	5.55E+03	5.23E+03	214.60	274.91	5.05E+03	4.71E+03	**60.07**
	Std	6.73E+04	6.51E+03	6.17E+03	274.97	266.11	5.81E+03	6.26E+03	60.07
	Rank	8	7	6	2	3	5	4	1
f_{20}	Mean	629.77	207.46	152.82	140.88	203.21	108.17	279.22	**91.84**
	Std	229.94	65.33	46.35	66.74	61.59	58.40	109.05	51.07
	Rank	8	6	4	3	5	2	7	1
f_{21}	Mean	355.16	253.42	246.47	243.49	229.33	232.56	286.98	**224.67**
	Std	48.53	14.50	15.74	11.21	52.61	9.95	21.32	44.14
	Rank	8	6	5	4	2	3	7	1
f_{22}	Mean	2.30E+03	1.09E+03	199.60	100.27	100.67	**100**	416.12	100.19
	Std	2.27E+03	1.57E+03	544.61	0.88	1.26	0	966.92	1.22
	Rank	8	7	5	3	4	1	6	2

(*continued on next page*)

Table 7 (*continued*).

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	BCLPSO
f_{23}	Mean	698.64	414.37	395.10	388.80	396.05	385.30	483.05	**385.24**
	Std	74.60	16.71	19.76	11.12	50.44	10.69	32.55	7.30
	Rank	8	6	4	3	5	2	7	1
f_{24}	Mean	756.00	479.80	474.59	469.14	**439.69**	458.00	548.44	471.94
	Std	98.94	24.88	15.16	12.64	89.94	10.98	37.11	10.98
	Rank	8	6	5	3	1	2	7	4
f_{25}	Mean	524.44	387.56	387.09	386.77	386.51	387.22	411.52	**386.34**
	Std	52.58	2.61	1.25	1.20	1.58	0.26	18.50	0.76
	Rank	8	6	4	3	2	5	7	1
f_{26}	Mean	3.64E+03	1.05E+03	1.08E+03	**325.73**	340.23	818.66	1.40E+03	380.49
	Std	9.57E+02	6.37E+02	6.49E+02	276.17	330.24	528.68	1.46E+03	247.13
	Rank	8	5	6	1	2	4	7	3
f_{27}	Mean	739.62	525.61	518.91	512.82	511.68	503.71	584.03	**502.81**
	Std	112.28	12.94	10.03	7.39	5.04	4.95	28.41	5.30
	Rank	8	6	5	4	3	2	7	1
f_{28}	Mean	650.30	350.58	366.67	369.39	327.77	333.07	398.26	**322.30**
	Std	103.58	60.38	61.73	51.05	46.93	56.74	25.54	9.43
	Rank	8	4	5	6	2	3	7	1
f_{29}	Mean	1.41E+03	614.53	538.34	490.83	516.15	466.69	688.49	**465.18**
	Std	3.32E+02	98.91	94.92	50.08	70.42	52.52	154.16	44.98
	Rank	8	6	5	3	4	2	7	1
f_{30}	Mean	7.00E+05	5.23E+03	4.15E+03	3.82E+03	4.02E+03	4.65E+03	5.77E+03	**3.52E+03**
	Std	6.47E+05	2.68E+03	1.57E+03	1.11E+03	1.11E+03	2.61E+03	2.17E+03	431.64
	Rank	8	6	4	2	3	5	7	1
Average rank		8.03	5.67	4.83	3.23	3.17	2.73	6.47	1.90
Best/2nd Best/Worst		0/0/30	0/2/0	1/0/0	1/7/0	3/7/0	8/8/0	0/0/0	17/6/0
Algorithms		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	BCLPSO

Table 8
The best optimum errors on 30 dimensions (run 30 times).

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO	BCLPSO
f_1	Optimum	1.04E+08	1.64E+02	5.49	0.05	0.05	12.02	45.30	6.75E+02	1.77E−03
	Rank	9	7	4	2	3	5	6	8	1
f_2	Optimum	1.653E+18	110	0	0	0	0	2.48E+11	1.25E+19	8.57
	Rank	5	3	1	1	1	1	4	6	2
f_3	Optimum	1.65E+04	3.88E−10	2.16E−09	1.65E−05	2.15E−10	5.68E−13	2.68E+03	3.27E+03	1.50E+03
	Rank	9	3	4	5	2	1	7	8	6
f_4	Optimum	1.35E+02	0.09	0.65	0.62	1.15E−03	43.13	64.14	44.23	15.14
	Rank	9	2	4	3	1	6	8	7	5
f_5	Optimum	1.06E+02	29.85	16.91	23.88	27.86	15.92	41.79	21.02	21.11
	Rank	9	7	2	5	6	1	8	3	4
f_6	Optimum	26.99	3.19E−06	1.37E−07	2.27E−13	2.27E−13	1.14E−13	1.74	3.29	1.14E−13
	Rank	7	4	3	2	2	1	5	6	1
f_7	Optimum	1.57E+02	67.23	63.82	65.71	56.03	48.31	69.07	30.58	49.19
	Rank	9	7	5	6	4	2	8	1	3
f_8	Optimum	73.84	33.83	28.85	23.88	25.87	13.93	37.81	19.10	23.44
	Rank	9	7	6	4	5	1	8	2	3
f_9	Optimum	1.78E+03	2.54	0.91	0.18	0.72	0	1.25E+02	1.52E+02	0.03
	Rank	9	6	5	3	4	1	7	8	2
f_{10}	Optimum	2.14E+03	1.28E+03	1.73E+03	1.43E+03	1.83E+03	1.01E+03	1.90E+03	5.84E+02	1.35E+03
	Rank	9	3	6	5	7	2	8	1	4
f_{11}	Optimum	1.20E+02	30.89	21.00	13.48	20.47	5.18	43.93	60.94	10.73
	Rank	9	6	5	3	4	1	7	8	2
f_{12}	Optimum	8.32E+05	4.58E+03	3.91E+03	8.95E+03	6.19E+03	2.77E+03	9.83E+03	1.00E+07	5.04E+03
	Rank	8	3	2	6	5	1	7	9	4
f_{13}	Optimum	1.64E+04	85.21	30.06	52.62	51.32	20.60	2.02E+02	1.82E+03	48.01
	Rank	9	6	2	5	4	1	7	8	3
f_{14}	Optimum	1.92E+03	1.93E+02	3.42E+02	1.63E+02	1.39E+02	1.88E+02	3.05E+02	1.94E+02	3.14E+02
	Rank	9	4	8	2	1	3	6	5	7
f_{15}	Optimum	3.74E+03	52.88	12.51	33.14	34.09	56.75	1.58E+02	4.13E+02	31.40
	Rank	9	5	1	3	4	6	7	8	2
f_{16}	Optimum	6.33E+02	2.49E+02	1.25E+02	10.91	2.55E+02	7.05	5.41E+02	1.23E+02	9.60

(continued on next page)

Table 8 (*continued*).

Function	Criteria	PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO	BCLPSO
	Rank	9	6	5	3	7	1	8	4	2
f_{17}	Optimum Rank	89.78 9	51.60 6	32.83 3	40.17 5	56.89 7	10.52 1	57.49 8	32.54 2	36.53 4
f_{18}	Optimum Rank	2.50E+04 6	2.52E+04 7	1.66E+04 5	1.11E+04 3	1.61E+04 4	1.06E+04 2	3.07E+04 8	5.07E+03 1	3.33E+04 9
f_{19}	Optimum Rank	4.70E+02 8	86.18 6	39.40 4	28.01 3	49.32 5	9.02 1	1.01E+02 7	6.55E+03 9	17.26 2
f_{20}	Optimum Rank	2.74E+02 9	52.98 7	32.36 2	39.54 5	43.51 6	3.01 1	94.06 8	39.09 4	36.70 3
f_{21}	Optimum Rank	2.84E+02 9	2.29E+02 7	2.17E+02 5	2.25E+02 6	100 2	2.13E+02 4	2.45E+02 8	38.68 1	1.19E+02 3
f_{22}	Optimum Rank	1.82E+02 3	1.00E+02 2	1.00E+02 2	1.00E+02 2	1.00E+02 2	1.00E+02 2	1.00E+02 2	99.23 1	1.00E+02 2
f_{23}	Optimum Rank	5.65E+02 9	3.87E+02 7	3.62E+02 4	3.56E+02 3	1.38E+02 2	3.68E+02 6	4.01E+02 8	76.24 1	3.65E+02 5
f_{24}	Optimum Rank	6.25E+02 9	4.49E+02 6	4.43E+02 4	4.46E+02 5	2.06E+02 2	4.41E+02 3	4.85E+02 8	77.27 1	4.51E+02 7
f_{25}	Optimum Rank	4.44E+02 5	3.84E+02 2	3.84E+02 2	3.84E+02 2	3.84E+02 2	3.87E+02 3	3.88E+02 4	55.30 1	3.84E+02 6
f_{26}	Optimum Rank	1.30E+03 3	2.00E+02 1	2.00E+02 1	2.00E+02 1	2.00E+02 1	2.00E+02 1	2.00E+02 1	3.28E+02 4	2.20E+02 2
f_{27}	Optimum Rank	5.55E+02 9	5.01E+02 6	5.04E+02 7	4.99E+02 4	4.99E+02 5	4.94E+02 2	5.42E+02 8	80.58 1	4.94E+02 3
f_{28}	Optimum Rank	5.27E+02 6	3.00E+02 3	3.00E+02 3	3.00E+02 4	3.00E+02 3	3.00E+02 3	3.22E+02 5	87.15 1	2.74E+02 2
f_{29}	Optimum Rank	8.61E+02 9	4.66E+02 7	4.10E+02 6	3.96E+02 5	3.84E+02 4	3.43E+02 2	4.81E+02 8	1.47E+02 1	3.73E+02 3
f_{30}	Optimum Rank	5.39E+04 9	2.29E+03 2	2.30E+03 3	2.50E+03 5	2.36E+03 4	2.09E+03 1	3.17E+03 7	3.14E+04 8	2.65E+03 6
Average rank		8.00	4.93	3.80	3.70	3.63	2.20	6.70	4.27	3.60

Table 9
Wilcoxon test results on 30 dimensions (significance level: $\alpha = 0.05$).

Function		Pairwise comparison BCLPSO versus							
		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO
$f(1)$	Symbol p	+ 3.02E−11	+ 5.49E−11	+ 1.96E−10	+ 1.68E−03	+ 4.18E−09	+ 2.61E−10	+ 4.98E−11	+ 1.86E−09
$f(2)$	Symbol p	+ 3.02E−11	+ 1.16E−07	+ 2.25E−04	− 2.12E−05	− 9.71E−11	− 6.75E−05	+ 3.02E−11	+ 1.86E−09
$f(3)$	Symbol p	+ 3.02E−11	− 3.02E−11	− 3.02E−11	− 3.02E−11	− 3.02E−11	− 3.02E−11	+ 1.43E−08	+ 3.25E−04
$f(4)$	Symbol p	+ 3.02E−11	− 5.61E−05	≈ 0.23	≈ 0.86	− 6.72E−10	− 1.07E−07	≈ 0.22	− 8.68E−07
$f(5)$	Symbol p	+ 3.02E−11	+ 2.01E−08	+ 6.77E−05	+ 8.56E−04	+ 1.17E−04	− 9.02E−04	+ 3.02E−11	− 4.28E−02
$f(6)$	Symbol p	+ 5.14E−12	+ 5.14E−12	+ 5.10E−12	+ 3.13E−08	+ 8.74E−10	≈ 0.97	+ 5.14E−12	+ 1.86E−09
$f(7)$	Symbol p	+ 3.02E−11	+ 4.98E−11	+ 4.62E−10	+ 1.96E−10	+ 2.13E−05	≈ 0.58	+ 3.69E−11	− 8.43E−06
$f(8)$	Symbol p	+ 3.02E−11	+ 1.56E−08	+ 1.17E−02	+ 1.95E−03	+ 5.26E−04	+ 2.43E−05	+ 1.61E−10	− 5.77E−08
$f(9)$	Symbol p	+ 3.02E−11	+ 1.29E−09	+ 8.14E−05	+ 1.87E−05	+ 2.15E−06	− 1.18E−08	+ 3.02E−11	+ 1.86E−09
$f(10)$	Symbol p	+ 7.38E−11	+ 9.26E−09	+ 1.86E−06	+ 2.61E−02	+ 1.87E−05	≈ 0.85	+ 8.48E−09	+ 1.86E−09
$f(11)$	Symbol p	+ 3.02E−11	+ 5.49E−11	+ 6.12E−10	+ 1.69E−09	+ 3.96E−08	≈ 7.48E−02	+ 3.02E−11	+ 1.86E−09
$f(12)$	Symbol p	+ 3.02E−11	≈ 0.78	≈ 0.47	+ 1.50E−02	≈ 0.54	≈ 0.65	+ 4.74E−06	+ 1.86E−09
$f(13)$	Symbol p	+ 3.02E−11	+ 4.20E−10	+ 1.01E−08	+ 8.35E−08	+ 6.72E−10	+ 4.57E−09	+ 1.09E−10	+ 1.86E−09

(*continued on next page*)

Table 9 (*continued*).

Function		Pairwise comparison BCLPSO versus							
		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO
$f(14)$	Symbol p	+ 1.09E−10	+ 1.75E−05	+ 1.02E−05	+ 1.58E−04	+ 2.07E−02	+ 1.99E−02	+ 2.00E−06	+ 1.61E−02
$f(15)$	Symbol p	+ 3.02E−11	+ 1.10E−08	+ 2.78E−07	+ 1.78E−04	+ 5.09E−08	+ 1.41E−09	+ 8.99E−11	+ 1.86E−09
$f(16)$	Symbol p	+ 3.02E−11	+ 4.12E−06	+ 1.99E−02	≈ 0.47	+ 2.39E−04	− 2.32E−02	+ 6.70E−11	− 1
$f(17)$	Symbol p	+ 6.07E−11	+ 3.08E−08	+ 2.62E−03	≈ 0.86	+ 5.09E−06	− 1.17E−04	+ 7.69E−08	≈ 0.58
$f(18)$	Symbol p	+ 3.85E−03	≈ 0.21	≈ 0.86	≈ 0.53	≈ 0.20	≈ 7.01E−02	≈ 0.30	≈ 1.86E−09
$f(19)$	Symbol p	+ 3.02E−11	+ 1.78E−10	+ 9.76E−10	+ 4.94E−05	+ 9.83E−08	+ 4.11E−07	+ 1.33E−10	− 0.36
$f(20)$	Symbol p	+ 3.02E−11	+ 3.82E−09	+ 5.56E−04	+ 2.16E−03	+ 1.20E−08	≈ 0.83	+ 8.99E−11	≈ 1.86E−09
$f(21)$	Symbol p	+ 3.02E−11	+ 1.77E−03	≈ 0.35	≈ 0.70	≈ 0.23	+ 1.12E−02	+ 2.87E−10	− 1.86E−09
$f(22)$	Symbol p	+ 3.02E−11	≈ 0.07	+ 3.28E−08	+ 4.48E−08	+ 1.36E−04	+ 1.21E−12	+ 4.92E−03	+ 5.77E−08
$f(23)$	Symbol p	+ 3.02E−11	+ 1.29E−09	+ 2.92E−02	+ 4.84E−02	+ 2.78E−07	≈ 0.92	+ 3.02E−11	+ 1.86E−09
$f(24)$	Symbol p	+ 3.02E−11	≈ 0.46	≈ 0.54	≈ 0.33	≈ 0.66	− 5.61E−05	+ 5.49E−11	+ 1.86E−09
$f(25)$	Symbol p	+ 3.02E−11	+ 8.15E−05	+ 1.17E−03	≈ 7.73E−02	≈ 9.63E−02	+ 8.56E−04	+ 3.02E−11	− 1.86E−09
$f(26)$	Symbol p	+ 4.08E−11	+ 8.27E−03	+ 5.54E−03	− 8.55E−04	− 8.97E−04	≈ 0.20	≈ 0.44	+ 1.40E−03
$f(27)$	Symbol p	+ 3.02E−11	+ 2.44E−09	+ 4.57E−09	+ 3.81E−07	+ 1.16E−07	≈ 0.62	≈ 3.02E−11	− 1.86E−09
$f(28)$	Symbol p	+ 3.02E−11	≈ 0.53	≈ 0.28	+ 1.84E−02	+ 5.48E−03	+ 5.47E−03	+ 2.87E−10	− 1.86E−09
$f(29)$	Symbol p	+ 3.02E−11	+ 4.57E−09	+ 2.01E−04	+ 2.51E−02	+ 1.11E−03	≈ 0.45	+ 2.37E−10	+ 1.86E−09
$f(30)$	Symbol p	+ 3.02E−11	+ 9.88E−03	≈ 0.59	≈ 0.86	≈ 0.13	≈ 0.62	+ 1.43E−08	+ 1.86E−09
+/≈/−		30/0/0	23/5/2	22/7/1	19/8/3	20/6/4	10/12/8	26/4/0	17/3/10

Table 10
The statistical comparison of the methods using sign test (for 30-dimensional).

Function		Pairwise comparison BCLPSO versus							
		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO
$f(1)$	Symbol p	+ 1.86E−09	+ 1.86E−09	+ 5.77E−08	+ 5.22E−03	+ 5.77E−08	+ 5.77E−08	+ 1.86E−09	+ 1.86E−09
$f(2)$	Symbol p	+ 1.86E−09	+ 8.43E−06	+ 1.61E−02	− 5.95E−05	− 1.86E−09	− 1.61E−02	+ 1.86E−09	+ 1.86E−09
$f(3)$	Symbol p	+ 1.86E−09	− 1.86E−09	− 1.86E−09	− 1.86E−09	− 1.86E−09	− 1.86E−09	+ 8.43E−06	≈ 0.86
$f(4)$	Symbol p	+ 1.86E−09	− 1.43E−03	≈ 0.20	≈ 0.86	− 1.86E−09	− 8.43E−06	≈ 0.20	− 8.68E−07
$f(5)$	Symbol p	+ 1.86E−09	+ 8.68E−07	+ 1.61E−02	+ 1.61E−02	+ 1.43E−03	− 4.28E−02	+ 1.86E−09	+ 1.86E−09
$f(6)$	Symbol p	+ 1.86E−09	+ 1.86E−09	+ 1.86E−09	+ 2.38E−07	+ 5.96E−08	≈ 1	+ 1.86E−09	+ 1.86E−09
$f(7)$	Symbol p	+ 1.86E−09	+ 5.77E−08	+ 5.77E−08	+ 1.86E−09	+ 3.25E−04	≈ 0.58	+ 1.86E−09	− 1.86E−09
$f(8)$	Symbol p	+ 1.86E−09	+ 5.95E−05	+ 1.61E−02	+ 5.22E−03	+ 1.43E−03	+ 5.22E−03	+ 5.77E−08	+ 1.86E−09
$f(9)$	Symbol p	+ 1.86E−09	+ 1.86E−09	+ 3.25E−04	+ 1.43E−03	+ 8.68E−07	− 5.77E−08	+ 1.86E−09	+ 1.86E−09
$f(10)$	Symbol p	+ 1.86E−09	+ 8.68E−07	+ 3.25E−04	≈ 0.20	+ 3.25E−04	≈ 0.58	+ 5.77E−08	− 1.86E−09

(*continued on next page*)

Table 10 (*continued*).

Function		Pairwise comparison BCLPSO versus							
		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO
$f(11)$	Symbol	+	+	+	+	+	≈	+	+
	p	1.86E−09	1.86E−091	5.77E−08	5.77E−08	5.77E−08	0.58	1.86E−09	1.86E−09
$f(12)$	Symbol	+	≈	≈	≈	≈	≈	+	+
	p	1.86E−09	0.58	0.58	9.87E−02	0.58	0.86	1.43E−03	1.86E−09
$f(13)$	Symbol	+	+	+	+	+	+	+	+
	p	1.86E−09	5.77E−08	8.68E−07	8.43E−06	1.86E−09	8.68E−07	5.77E−08	1.86E−09
$f(14)$	Symbol	+	+	+	+	≈	≈	+	−
	p	1.86E−09	1.43E−03	8.43E−06	5.22E−03	0.20	9.87E−02	1.43E−03	1.86E−09
$f(15)$	Symbol	+	+	+	+	+	+	+	+
	p	1.86E−09	8.43E−06	8.68E−07	5.22E−03	8.43E−06	8.68E−07	5.77E−08	1.86E−09
$f(16)$	Symbol	+	+	≈	≈	+	≈	+	+
	p	1.86E−09	5.95E−05	9.87E−02	0.58	3.25E−04	3.62E−01	5.77E−08	5.77E−08
$f(17)$	Symbol	+	+	+	≈	+	−	+	+
	p	1.86E−09	8.43E−06	4.28E−02	1	5.95E−05	4.28E−02	5.77E−08	1.86E−09
$f(18)$	Symbol	+	≈	≈	≈	≈	≈	≈	+
	p	1.86E−09	0.58	0.86	0.58	0.36	0.20	0.36	1.86E−09
$f(19)$	Symbol	+	+	+	+	+	+	+	+
	p	1.86E−09	5.77E−08	5.77E−08	3.25E−04	8.68E−07	8.43E−06	5.77E−08	1.86E−09
$f(20)$	Symbol	+	+	+	+	+	≈	+	+
	p	1.86E−09	8.68E−07	5.22E−03	4.28E−02	5.77E−08	0.36	1.86E−09	5.77E−08
$f(21)$	Symbol	+	+	≈	≈	≈	+	+	−
	p	1.86E−09	5.22E−03	0.86	0.86	0.36	1.61E−02	5.77E−08	1.86E−09
$f(22)$	Symbol	+	≈	+	+	+	+	+	−
	p	1.86E−09	0.20	8.43E−06	8.68E−07	1.43E−03	1.86E−09	4.28E−02	1.86E−09
$f(23)$	Symbol	+	+	≈	≈	+	≈	+	−
	p	1.86E−09	5.77E−08	0.36	0.20	5.95E−05	0.86	1.86E−09	1.86E−09
$f(24)$	Symbol	+	≈	≈	≈	≈	−	+	−
	p	1.86E−09	0.36	0.86	0.58	0.58	5.22E−03	1.86E−09	1.86E−09
$f(25)$	Symbol	+	+	≈	≈	≈	+	+	−
	p	1.86E−09	5.95E−05	9.87E−02	0.36	0.20	3.25E−04	1.86E−09	1.86E−09
$f(26)$	Symbol	+	+	+	−	−	≈	≈	≈
	p	1.86E−09	0.04	4.28E−02	5.22E−03	5.22E−03	0.58	0.86	0.20
$f(27)$	Symbol	+	+	+	+	+	≈	+	
	p	1.86E−09	8.68E−07	5.77E−08	3.25E−04	8.68E−07	0.58	1.86E−09	1.86E−09
$f(28)$	Symbol	+	≈	≈	+	+	+	+	−
	p	1.86E−09	0.86	0.36	4.28E−02	4.28E−02	4.28E−02	5.77E−08	1.86E−09
$f(29)$	Symbol	+	+	+	+	+	≈	+	−
	p	1.86E−09	5.77E−08	5.95E−05	4.28E−02	3.25E−04	0.36	1.86E−09	1.86E−09
$f(30)$	Symbol	+	≈	≈	≈	+	≈	+	+
	p	1.86E−09	0.20	0.86	1	4.28E−02	0.86	8.68E−07	1.86E−09
+/≈/−		30/0/0	22/6/2	19/10/1	16/11/3	20/6/4	9/14/7	27/3/0	17/2/11

Table 11
The Friedman test results of PSO variants on CEC2017 functions.

		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO	BCLPSO	p-value	Friedman value
10 dimensions	Friedman	8.13	5.90	4	2.78	3.31	8.3	5.67	4.27	2.63	3.56E−28	47.27
30 dimensions	rank	8.80	6.23	5.33	3.67	3.57	3.13	7.07	4.70	2.50	1.61E−25	37.92

prior information of particles. The BCLPSO algorithm therefore consistently performs well on hybrid and composition functions.

As shown in Fig. 6, the BCLPSO, HCLPSO, and EPSO maintain good diversity in the evolutionary process. The CLPSO, FDR-PSO, and GGL-PSOD maintain good diversity in the early evolutionary stage but rapidly loses its diversity in the later stage as the convergence rate accelerates. The three algorithms can therefore perform better explorations in the early stage and then fall into the local optimum in the later stage. The diversity curve of PSO-w show that the traditional PSO algorithm can facilely fall into the local optimum and prematurely converge.

The robustness is to measure the stability of the algorithm for parameter changes. We refer to paper [55] to analyze the robustness of BCLPSO algorithm on different dimensions. It is investigated in one unimodal problem (f_3: Shifted and Rotated Zakharov Function) and one multimodal problem (f_{10}: Shifted and Rotated Schwefel Function) on different dimensions. Fig. 6((a) and (b)) shows the sensitivity curve of f_3 and f_{10}, respectively.

$$\Delta t=\sqrt{\frac{\Delta x+\Delta f}{2}}\ where\ \Delta x=\sqrt{\frac{1}{N}\sum_{d=1}^{D}(\frac{g_d-x^*_{d,min}}{Z_d})^2}\ and$$
$$\Delta f=\frac{f(g)-f^*_{\min}}{f^*_{\max}-f^*_{\min}} \tag{16}$$

where the Δt is a normalized Euclidean distance between the global optimum found by the algorithm and a desired minimizer x^*, where $Z_d=|u_d-l_d|$ is the range of the d-th variable. $f_{\min}$ is the

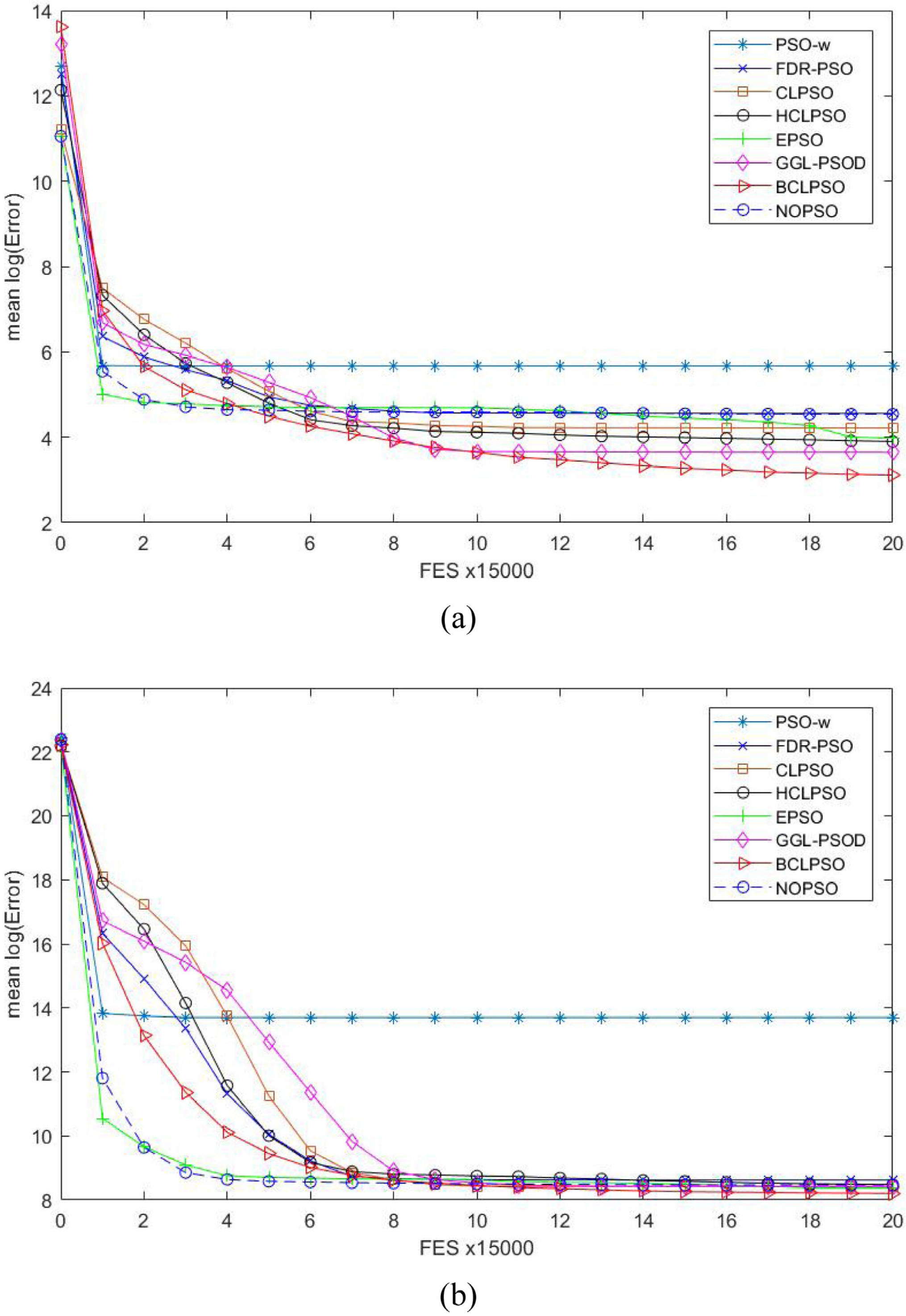

Fig. 5. Mean error comparisons between PSO-w, FDR-PSO, CLPSO, HCLPSO, EPSO, GGL-PSOD, NOPSO, and BCLPSO on f_{11} in (a) and on f_{30} in (b).

minimum found by the algorithm, $f^*_{\min}$ is the analytical minimum, and $f^*_{\max}$ is the analytical maximum of the function $f(\mathbf{x})$ [55].

It can be seen from Fig. 7 that the robustness of BCLPSO is relatively stable in both unimodal and multimodal problems with the change of dimensions. This means that the robustness of BCLPSO algorithm is consistently stable in different dimensions, especially on high-dimensional problems. However, the BCLPSO achieves relatively weak performance on the low-dimensional of multimodal problems. According to no free lunch theorem, all algorithms cannot perform well on every problem, so this is acceptable.

5. Application of BCLPSO on quality process control of automated welding production line for automobile body

In order to verify the performance of the proposed BCLPSO algorithm on real-life problems, focus is set on the quality process control of an automated welding production line for automobile

Table 12
ECT of multivariate Bayesian VSI chart for automated welding production line.

Function evolutions		PSO-w	FDR-PSO	CLPSO	HCLPSO	EPSO	GGL-PSOD	NOPSO	DPSO	BCLPSO
1000	ECT	28.70	26.97	29.53	28.66	26.03	28.47	28.70	28.79	26.83
	Std	3.43	1.00	1.34	1.83	1.11	0.77	3.43	–	0.85
	Penalty	6.52%	0.52%	9.14%	6.40%	3.07%	2.33%	1.40%	6.81%	–
10000	ECT	25.98	24.62	23.55	23.29	23.70	24.14	24.83	24.13	23.46
	Std	1.76	1.23	0.78	0.79	0.86	1.55	1.05	–	0.62
	Penalty	9.70%	4.71%	0.38%	−0.75%	1.02%	2.80%	5.52%	2.78%	–
100000	ECT	24.87	23.23	22.55	22.93	21.80	24.69	23.70	22.88	21.65
	Std	1.72	0.82	0.78	0.42	0.73	1.03	0.90	–	0.74
	Penalty	12.95%	6.81%	3.99%	5.57%	0.67%	8.59%	8.67%	5.34%	–
300000	ECT	23.92	22.14	21.92	21.69	22.32	23.71	22.63	22.25	20.18
	Std	0.35	0.58	0.53	0.64	0.35	0.78	0.49	–	0.47
	Penalty	15.64%	8.85%	7.94%	6.96%	9.86%	14.89%	10.83%	9.30%	–

bodies in Geely Cars. The relative dimensions of the left and right mounting holes of the frame in the engine compartment are the quality control parameters, as shown in Fig. 8.

In the company's current situation, the quality control of the welding production line is designed by a simple $\overline{X}$ control chart to manage the coordinate dimensions of the hole center, X, Y, and Z. However, considering the three shortcomings of the $\overline{X}$ control chart, one is that the control chart does not consider the economic factors in quality control; the other is that the $\overline{X}$ control chart can only control a one-dimensional independent variable; the third is that the control line setting of the control chart is too simple and the statistical performance is not good. To achieve the best long-run average quality control cost, the multivariate Bayesian VSI control chart [56] is designed, and a quality control model is presented in (Eq. (17)). The economic parameters of the control chart can be calculated based on empirical data, but the statistical parameters such as control line setting can only be solved by particle swarm optimization algorithm. In the process of searching for optimal statistical parameters, the Monte Carlo simulation method is used for our objective function, in which the simulation cycle is 2000.

The related parameters are as follows: fixed cost per sample (b); variable sampling cost per unit (c); sample size (n); sampling period (h); cost per unit time of operation in the out-of-control state (M); the cost of search ($A > 0$); cost of additional repairs ($R \geq 0$). The expected investigation time is denoted as T_1, and the expected time to eliminate the assignable cause is denoted as T_1; π_{0p} is the state probability, which indicates that the process is in control; π_{1p} is the steady state probability, which indicates that the process is out-of-control; $\pi_{0p} = \pi_{1p} = 0.5$; p_{S_i} and p_{R_i} ($i = 1{,}2$) denote the control and alarm limits, respectively. It is assumed that the randomly occurring assignable cause has a time, τ, between occurrences; it is exponentially distributed with a mean $1/\theta$.

$$
\begin{aligned}
ECT &= \frac{\textit{Expected Cost}}{\textit{Expected Time}} \\
&= \sum_{i=1,2}\Bigg\{ b + \sum_{p\le p_{R_i}} \pi_{0p}\left[M\left[h_1-\left(1-e^{-\theta h_1}\right)/\theta\right]+cn\right] \\
&\quad + \sum_{p_{R_i}<p\le p_{S_i}} \pi_{0p}\left[M\left[h_2-\left(1-e^{-\theta h_2}\right)/\theta\right]+cn\right] \\
&\quad + \sum_{p>p_{S_i}} \pi_{0p}\left[M\left[h_1-\left(1-e^{-\theta h_1}\right)/\theta\right]+cn+A\right] \\
&\quad + \sum_{p\le p_{R_i}} \pi_{1p}\left(Mh_1+cn\right) \\
&\quad + \sum_{p_{R_i}<p\le p_{S_i}} \pi_{1p}\left(Mh_2+cn\right) \\
&\quad + \sum_{p>p_{S_i}} \pi_{1p}\left[M\left[h_1-\left(1-e^{-\theta h_1}\right)/\theta\right]+cn+R\right]\Bigg\} \div \\
&\quad \sum_{i=1,2}\Bigg\{ \sum_{p\le p_{R_i}} \left(\pi_{0p}+\pi_{1p}\right)h_1 + \sum_{p_{R_i}<p\le p_{S_i}} \left(\pi_{0p}+\pi_{1p}\right)h_2 \\
&\quad + \sum_{p>p_{S_i}} \pi_{0p}\left(h_1+T_0\right) + \sum_{p>p_{S_i}} \pi_{1p}\left(h_1+T_0+T_1\right)\Bigg\}
\end{aligned}
\tag{17}
$$

The optimization model is described as follows:

$$
\begin{aligned}
&\min \; ECT(h_1, h_2, n, p_{S1}, p_{S2}) = \frac{\textit{Expected Cost}}{\textit{Expected Time}} \\
&s.t. \quad h_1 \ge h_2, p_{S1} \ge p_{S2}; \\
&\qquad\quad n \in N^*.
\end{aligned}
\tag{18}
$$

After two months of investigation in the Geely automobile body automatic welding production line, relevant parameters are calculated. The in-control mean is $\boldsymbol{\mu}_0^T = (0.08, 908.15, 0.05)$, and the out-of-control mean is $\boldsymbol{\mu}_1^T = (-1.47, 909.54, 1.67)$. The covariance matrix is $\hat{\sum} = \begin{bmatrix} 0.54 & 0.04 & 0 \\ 0.04 & 0.37 & -0.02 \\ 0 & -0.02 & 0.33 \end{bmatrix}$. $A = 115.5$ *yuan*, $R = 349.5$ *yuan*, $M = 111$ *yuan* per hour, $\theta = 0.013$, $b = 4.9$ *yuan* per unit time, and $c = 0.6$ *yuan*.

There are five variables in this model: h_1, h_2, n, p_{S1}, and p_{S2}. The proposed BCLPSO algorithm is employed to solve this model, and the results are compared with the other eight algorithms. The comparison results are summarized in Table 12.

The penalty is $\frac{\textit{other algorithms}-\textit{BCLPSO}}{\textit{other algorithms}}(\%)$. The optimal result of each algorithm decrease with the increase of function evolutions

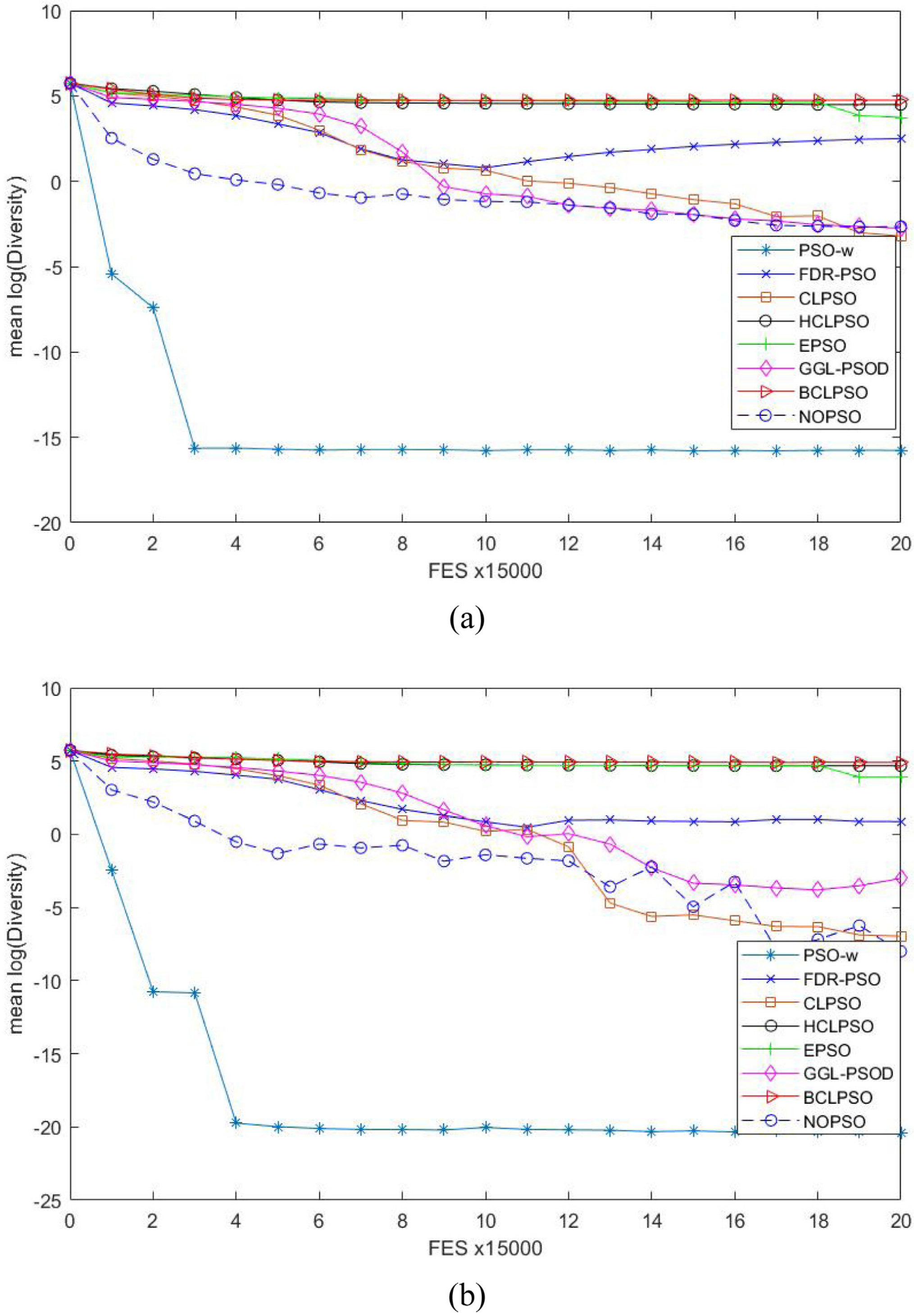

Fig. 6. Diversity comparisons between PSO-w, FDR-PSO, CLPSO, HCLPSO, EPSO, GGL-PSOD, NOPSO, and BCLPSO on f_{11} in (a) and on f_{30} in (b).

(FEs), which means that the reduction of FEs will greatly weaken the accuracy of the algorithm. Selecting the appropriate FES can not only improve the algorithm accuracy, but also reduce the running time. Therefore, the FEs is set as 300000. In addition, when FES ranges from 1000 to 10000, the advantage of BCLPSO is not obvious. It is because BCLPSO algorithm is similar to other algorithms in the early stage of evolution, but in the later stage, it can avoid falling into local optimum and has a wider exploration space.

As listed in the last line of Table 12, the BCLPSO provides the best performance on this practical problem, and it can considerably improve the economy than the other algorithms. The maximum and minimum values increase by 15.64% and 6.96%, respectively. The proposed BCLPSO is therefore applicable to practical problems.

6. Conclusions and future works

In this study, the Bayesian iteration method is merged with the comprehensive learning strategy; accordingly, the BCLPSO

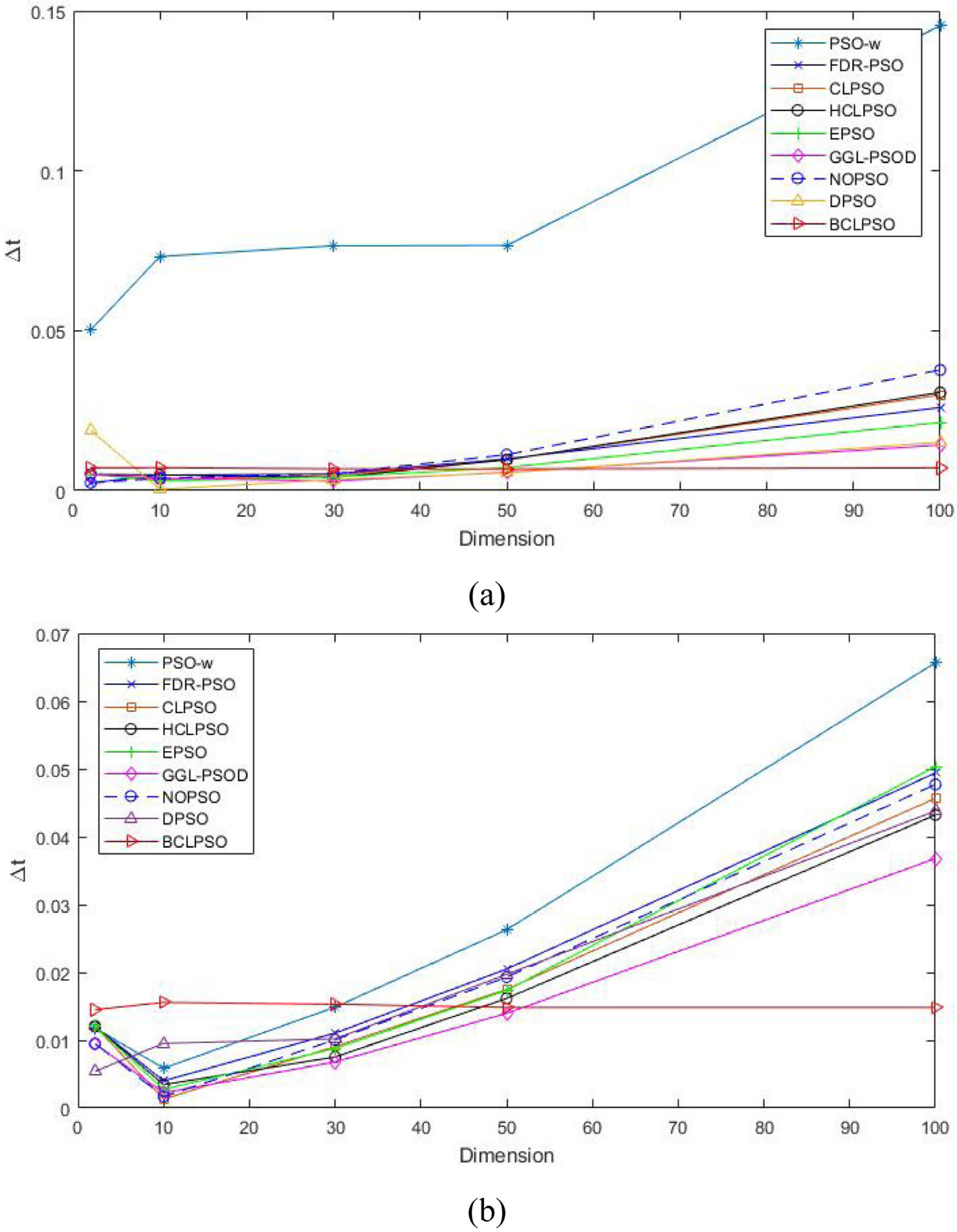

Fig. 7. Sensitivity analysis on f_3 in (a) and on f_{10} in (b).

algorithm is proposed. Overall, the BCLPSO outperforms other state-of-the-art PSO algorithms, particularly on multimodal and hybrid functions. The comparison results show that the BCLPSO algorithm can fully utilize the historical prior information of particles, indicating that the particles will neither easily fall into the local optimum nor miss the potential optimum solution. This is because the swarm leader of the BCLPSO algorithm is not the location of the minimum fitness function value (Gbest) but the particle location with the largest posterior probability after iteration based on the Bayesian formula. The posterior probability is developed by historical prior information.

In order to verify the performance of the proposed BCLPSO algorithm on real-life problems, the quality process control of an automated welding production line for automobile bodies is investigated. A process quality control model is constructed and employed to test the BCLPSO and other similar PSO algorithms. The test results show that the BCLPSO algorithm outperforms other state-of-the-art PSO algorithms. It is evident that the BCLPSO can be successfully applied to certain complex practical engineering problems.

Currently, the parameters of BCLPSO algorithm is not self-regulated, therefore, for future research, more intelligent adaptive mechanism deserves further investigation. The adaptive mechanism should consider more information such as the population distribution and the velocity of the particles, the success rate of the search behavior and so on. Furthermore, new learning strategies need be studied to efficiently improve the ability of exploration and exploitation.

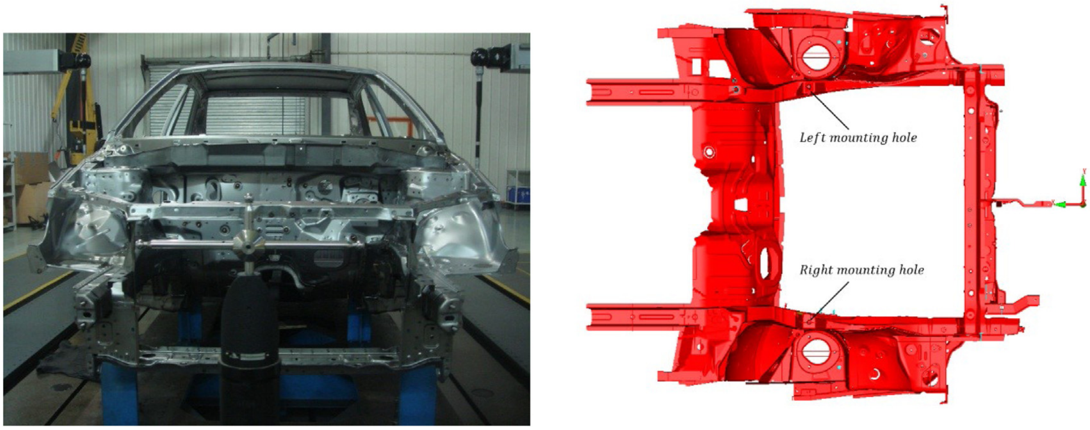

Fig. 8. Left and right mounting holes of the frame in the engine compartment.

CRediT authorship contribution statement

Xing Zhang: Conceptualization, Methodology, Software, and Writing - original draft. **Wei Sun:** Project administration, Supervision, Reviewing, Editing. **Min Xue:** Data curation, Visualization, Investigation. **Anping Lin:** Software, Validation, Reviewing, Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.
[2] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Australia, 1995, pp. 1942–1948.
[3] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufman Publishers, San Francisco, 2001.
[4] S.Y. Ho, H.S. Lin, W.H. Liauh, S.J. Ho, OPSO: Orthogonal particle swarm optimization and its application to task assignment problems, J. IEEE Trans. Syst. Man Cybern. B 38 (2) (2008) 288–289.
[5] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R.G. Harley, Particle swarm optimization: Basic concepts, variants and applications in power system, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195.
[6] M.P. Wachowiak, R. Smolikova, Y. Zheng, J.M. Zurada, A.S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, J. IEEE Trans. Evol. Comput. 8 (3) (2004) 289–301.
[7] Y.H. Shi, R. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Conference on Evolutionary Computation, IEEE Press, Piscataway, 1998, pp. 69–73.
[8] M. Clerc, J. Kennedy, The particle swarm – explosion, stability, and convergence in a multidimensional complex space, J. IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[9] B. Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang, Improved particle swarm optimization combined with chaos, Chaos Solitons Fractals 25 (5) (2005) 1261–1271.
[10] M.R. Tanweer, S. Suresh, N. Sundararajan, Self-regulating particle swarm optimization algorithm, J. Inform. Sci. 294 (2015) 182–202.
[11] A.B. Hashemi, M.R. Meybodi, A note on the learning automata based algorithms for adaptive parameter selection in PSO, J. Appl. Soft Comput. 11 (1) (2011) 689–705.
[12] M.S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, G. Pasi, Fuzzy selftuning PSO: a settings-free algorithm for global optimization, J. Swarm Evol. Comput. 39 (2018) 70–85.
[13] A. Serani, M. Diez, Are random coefficients needed in particle swarm optimization for simulation-based ship design, in: Proceedings of the 7th International Conference on Computational Methods in Marine Engineering, Nantes, France, 2017, pp. 1–12.
[14] J.R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: Simpler, maybe better, J. IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.
[15] P.N. Suganthan, Particle swarm optimizer with neighborhood operator, in: Proceedings of the Congress on Evolutionary Computation, Washington, D.C. USA, 1999, pp. 1958–1962.
[16] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Proceedings of the 2003 IEEE swarm intelligence symposium, Indianapolis, Indiana, USA, 2003, pp. 174–181.
[17] W.H. Lim, N.A.M. Isa, Adaptive division of labor particle swarm optimization, J. Expert Syst. Appl. 42 (14) (2015) 5887–5903.
[18] I. Hanaf, F.M. Cabrerab, F. Dimanea, J.T. Manzanaresb, Application of particle swarm optimization for optimizing the process parameters in turning of peek cf30 composites, in: Proceedings of the 9th International Conference on Interdisciplinarity in Engineering, Romania, 22, 2016, pp. 195–202.
[19] J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, J. IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[20] S.L. Sabat, L. Ali, S.K. Udgata, Integrated learning particle swarm optimizer for global optimization, J. Appl. Soft Comput. 11 (1) (2011) 574–584.
[21] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, J. Swarm Evol. Comput. 24 (2015) 11–24.
[22] A.P. Lin, W. Sun, H.S. Yu, G.H. Wu, H.W. Tang, Adaptive comprehensive learning particle swarm optimization with cooperative archive, J. Appl. Soft Comput. 77 (2019) 533–546.
[23] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, J. Inform. Sci. 294 (2015) 182–202.
[24] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, J. Inform. Sci. 326 (2016) 1–24.
[25] M.R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self-regulating particle swarm optimization algorithm, J. Swarm Evol. Comput. 28 (2016) 98–116.
[26] H.D. Wang, Y.C. Jin, J. Doherty, Committee-based active learning for surrogate assisted particle swarm optimization of expensive problems, J. IEEE Trans. Cybern. 47 (9) (2017) 2664–2677.
[27] L.L. Kang, R.S. Chen, W.L. Cao, Y.C. Chen, Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications, J. Appl. Soft Comput. 88 (2020) 106038.
[28] M. Løvbjerg, T.K. Rasmussen, T. Krink, Hybrid particle swarm optimizer with breeding and subpopulations, in: Proceedings of International Conference on Genetic and Evolutionary Computation, San Francisco, USA, 2001, pp. 469–476.
[29] W.J. Zhang, X.F. Xie, DEPSO: Hybrid particle swarm with differential evolution operator, in: Proceedings of the IEEE International Conference on System, Man and Cybernetics, 4, 2003, pp. 3816–3821.
[30] V. Miranda, N. Fonseca, EPSO-best-of-two-worlds meta-heuristic applied to power system problems, in: Proceedings of the IEEE Congress on Evolutionary Computation, Honolulu, Hawaii, USA, vol. 2, 2002, pp. 1080–1085.
[31] Y.J. Gong, J.J. Li, Y.C. Zhou, Y. Li, H.S.H. Chung, Y.H. Shi, J. Zhang, Genetic learning particle swarm optimization, J. IEEE Trans. Cybern. 46 (2016) 2277–2290.
[32] A.P. Lin, W. Sun, H.S. Yu, G.H. Wu, H.W. Tang, Global genetic learning particle swarm optimization with diversity enhancement by ring topology, J. Swarm Evol. Comput. 44 (2019) 571–583.

[33] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, J. Appl. Soft Comput. 55 (2017) 533–548.
[34] Y.G. Chen, L.X. Li, H.P. Peng, J.H. Xiao, Q.T. Wu, Dynamic multi-swarm differential learning particle swarm optimizer, J. Swarm Evol. Comput. 39 (2018) 209–221.
[35] E. Naderi, M. Pourakbari-Kasmaei, M. Lehtonen, Transmission expansion planning integrated with wind farms: A review, comparative study, and a novel profound search approach, J. Int. J. Electr. Power Energy Syst. 115 (2020) 105460.
[36] J.M. Bernardo, A.F.M. Smith, Bayesian Theory, John Wiley & Sons, 2009.
[37] M.H. Pesaran, R.P. Smith, A Bayesian analysis of linear regression models with highly collinear regressors, J. Econom. Stat. 11 (2019) 1–21.
[38] T.G. Tan, S. Jang, A. Yamaguchi, A novel method for risk-informed decision-making under non-ideal instrumentation and control conditions through the application of Bayes' theorem, J. Reliab. Eng. Syst. Safety 188 (2019) 463–472.
[39] S. Pfreundschuh, P. Eriksson, D. Duncan, A neural network approach to estimate posterior distributions of Bayesian remote sensing retrievals, in: Proceedings of the 20th the European Geosciences Union, Vienna, Austria, 2018, pp. 13988.
[40] M.E. Tipping, Bayesian inference: An introduction to principles and practice in machine learning, in: Proceedings of the Summer School on Machine Learning, Berlin, Heidelberg, 2004, pp. 41–62.
[41] R. Di, X. Gao, Bayesian network structure learning based on restricted particle swarm optimization, J. Syst. Eng. Electron. 33 (11) (2011) 2423–2427.
[42] X. Liu, X. Liu, Structure learning of Bayesian networks by continuous particle swarm optimization algorithms, J. Stat. Comput. Simul. 88 (8) (2018) 1528–1556.
[43] N.H. Awad, M.Z. Ali, J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization, 2016, Available: http://www.ntu.edu.sg/home/epnsugan/.
[44] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 1999, pp. 1945–1950.
[45] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, J. IEEE Trans. Evol. Comput. 8 (3) (2004) 240–255.
[46] K.E. Parsopoulos, M.N. Vrahatis, UPSO: A Unified Particle Swarm Optimization Scheme, in: Lecture Series on Computational Sciences, 2004, pp. 868–873.
[47] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Particle swarm optimization algorithms with novel learning strategies, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Netherlands, Oct. 2004, pp. 3659–3664.
[48] T. Bartz-Beielstein, Experimental Research in Evolutionary Computation, Springer, New York, 2006, pp. 3001–3020.
[49] M. Hollander, D. Wolfe, E. Chicken, Nonparametric Statistical Methods, third ed., John Wiley & Sons, Inc., 2013.
[50] J.D. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, fifth ed., Chapman & Hall, 2010.
[51] J. Derrac, S. Garcia, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, J. Swarm Evol. Comput. 1 (1) (2011) 3–18.
[52] T. Zhang, C. Yang, H. Chen, L. Sun, K.J. Deng, Multi-objective optimization operation of the green energy island based on hammersley sequence sampling, J. Energy Conv. Manag. 204 (2020) 112316.
[53] S. Garcia, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617–644.
[54] O. Olorunda, A. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, 2008, pp. 1128–1134.
[55] A. Serani, C. Leotardi, U. Iemma, E.F. Campana, G. Fasano, M. Diez, Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems, J. Appl. Soft Comput. 49 (2016) 313–334.
[56] X. Zhang, J. Liu, C. Tan, Economic design based on a multivariate Bayesian VSI chart with a dual control limit, J. Commun. Stat. 14 (46) (2017) 6808–6822.

Xing Zhang received B. Eng. degree from Hunan University of Science and Technology, and the M.E.degree in mechanical engineering from Hunan University, Changsha, China, in 2013 and 2016, respectively. He is currently working towards the Ph.D. degree at the college of Electrical and Information Engineering, Hunan University. His research interests include evolutionary computation, swarm intelligence, and pattern recognition.

Wei Sun received the B.S., M.S., and Ph.D. degrees from Hunan University, Changsha, China, in 1997, 1999, and 2003, respectively. He is currently working as a Professor with the College of Electrical and Information Engineering, Hunan University. His research interests include computer vision and robotics, neural networks, and intelligent control.

Min Xue received the M.S. degree in control science and engineering from Hunan University, Changsha, China, in 2017, where she is currently working toward the Ph.D. degree in control science and engineering. Her current research interests include localization, navigation and intelligent robot.

Anping Lin received the B.S. and M.S. degrees in marine engineering from Dalian Maritime University, Dalian, China, in 2004 and 2007, respectively. From 2007 to 2014, he was teaching in Guangdong Ocean University, Zhanjiang, China. He is currently working toward the Ph.D. degree with Hunan University, Changsha, China. His main research interests include evolutionary computation, multimodal optimization, and computation intelligence.