Swarm and Evolutionary Computation 44 (2019) 571–583

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Global genetic learning particle swarm optimization with diversity enhancement by ring topology

Anping Lin [a,b,c], Wei Sun [a,c,d,*], Hongshan Yu [a,d,**], Guohua Wu [e,f,***], Hongwei Tang [a,d]

[a] College of Electrical and Information Engineering, Hunan University, Changsha, 410082, PR China
[b] School of Information Engineering, Lingnan Normal University, Zhanjiang, 524048, PR China
[c] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha, 410082, PR China
[d] Hunan Key Laboratory of Intelligent Robot Technology in Electronic Manufacturing, Changsha, 410082, PR China
[e] School of Traffic and Transportation Engineering, Central South University, Changsha 410073, PR China
[f] College of Information Systems and Management, National University of Defense Technology, Changsha, 410082, PR China

ARTICLE INFO

Keywords:
Genetic learning
Particle swarm optimization
Exploration
Exploitation

ABSTRACT

Genetic learning particle swarm optimization (GL-PSO) improves the performance of particle swarm optimization (PSO) by breeding superior exemplars to guide the motion of particles. However, GL-PSO adopts a global topology for exemplar generation and cannot preserve sufficient diversity to enhance exploration, and therefore, its performance on complex optimization problems is unsatisfactory. To further improve GL-PSO's performance and adaptability, two modifications are incorporated into the original GL-PSO. A ring topology is adopted in exemplar generation to enhance diversity and exploration, while a global learning component (GLC) with linearly adjusted control parameters is employed to improve the algorithm's adaptability. The resultant algorithm is referred to as global genetic learning particle swarm optimization with diversity enhancement by ring topology (GGL-PSOD). To validate the effectiveness of these two modifications, they are combined with GL-PSO separately and together and further tested experimentally. The comparison results on the CEC2017 test suite show that the adoption of the ring topology in exemplar generation can enhance the diversity and exploration capability of GL-PSO, while combining GLC alone with GL-PSO cannot achieve significant improvement. Incorporating both modifications into GL-PSO, the resultant GGL-PSOD exhibits high performance and strong adaptability on different types of CEC2017 functions. It outperforms seven representative PSO variants and five non-PSO meta-heuristics, including GL-PSO, SL-PSO, HCLPSO, EPSO, ABC, CMA-ES and CS.

1. Introduction

By mimicking a flock of birds' foraging behavior, Kennedy and Eberhart [1,2] proposed particle swarm optimization (PSO) in 1995. Since its inception, PSO has attracted great interest from researchers, resulting in numerous PSO-related publications reported every year with regard to theoretic investigation and real-life applications [3–5]. Because PSO is easy to implement and can provide better solutions on different benchmark functions and engineering problems [6], it is widely applied to real-life optimization problems, such as power systems [7,8], mechanical engineering [9], control systems [10], communication networks [11], antenna design [12], image processing [13], clustering [5], composites [14–17] and so on. In recently years, Lal and Singh [18] presented a modified particle swarm optimization-based maximum power point tracking controller for single-stage utility-scale photovoltaic systems. Mistry et al. [19] proposed a micro genetic algorithm embedded particle swarm optimization feature selection approach for intelligent facial emotion recognition. Guedria [20] proposed an improved accelerated PSO algorithm for mechanical engineering optimization problems. Elsayed et al. [21] presented an improved random drift particle swarm optimization with a self-adaptive mechanism to solve the power economic dispatch problem. Shamsipour et al. [22,23] coupled an adaptive neuro-fuzzy inference system with particle swarm optimization to optimize the electromagnetic stirrer process of nanocomposite production.

The performance of PSO has been improved significantly in the past two decades, but there are still a few challenges. Similar to other

* Corresponding author. College of Electrical and Information Engineering, Hunan University, Changsha, 410082, PR China.
** Corresponding author. College of Electrical and Information Engineering, Hunan University, Changsha, 410082, PR China.
*** Corresponding author. School of Traffic and Transportation Engineering, Central South University, Changsha 410073, PR China.
E-mail addresses: david-sun@126.com (W. Sun), yuhongshancn@hotmail.com (H. Yu), guohuawu@nudt.edu.cn (G. Wu).

https://doi.org/10.1016/j.swevo.2018.07.002
Received 11 April 2018; Received in revised form 17 June 2018; Accepted 28 July 2018
Available online 1 August 2018
2210-6502/

population-based optimization algorithms, PSO suffers from diversity loss and premature convergence [24]. Furthermore, PSO is problem-dependent, and few PSO variants can perform well on a variety of problems [25].

Many PSO variants are reported to overcome the aforementioned limitations. The important PSO variants can be summarized as the following four types:

(1) Parameter tuning: Shi et al. [26,27] suggested that having the inertia weight start with a value of 0.9 and linearly decrease to 0.4 over the course of the run could greatly improve the performance of PSO. Clerc and Kennedy [28] explored how the particle swarm algorithm works from the individual particle's point of view and introduced constriction coefficients to control the dynamical characteristics of the particle swarm, including its exploration versus exploitation propensities. To efficiently control the local search and convergence to the global optimum solution, Ratnaweera et al. [29] introduced time-varying acceleration coefficients (TVAC) to the time-varying inertia weight factor in particle swarm optimization. Marco et al. [30] employed Fuzzy Logic to calculate the inertia, cognitive and social factor, and minimum and maximum velocity independently for each particle and proposed fuzzy self-turning PSO (FST-PSO). Zhan et al. [31] divided the particles into four evolutionary states and employed automatic control of inertia weight, acceleration coefficients, and other algorithmic parameters at runtime to improve the search efficiency and convergence speed.

(2) Neighborhood topology: Mendes and Kennedy [32] studied the square, ring, four clusters, and pyramid structures and proposed fully informed particle swarm (FIPS). Liang et al. [33,34] proposed dynamic multi-swarm particle swarm optimization. Lim et al. [35] presented particle swarm optimization with increasing topology connectivity (PSO-ITC). Chen et al. [36] proposed a dynamic multi-swarm differential learning particle swarm optimizer (DMSDL-PSO). Wang et al. [37] improved the performance of traditional PSO by increasing the two layers of swarms to multiple layers. Xia et al. [38] proposed a sophisticated PSO based on multi-level adaptation and purposeful detection operators. Liu et al. [39] analyzed a class of deterministic regular topologies with regard to what affects the optimality of algorithmic parameters and provided a guide to topology selection for PSO.

(3) Learning strategy: Peram et al. [40] proposed fitness-distance-ratio based particle swarm optimization (FDR-PSO). FDR-PSO moves particles towards nearby particles of higher fitness instead of attracting each particle towards just the best position discovered so far by any particle. Liang et al. [41] introduced a novel comprehensive learning strategy in which all other particles' historical best information is used to update a particle's velocity. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence. Wu et al. [42] presented superior solution-guided particle swarm optimization (SSG-PSO) with a novel individual level-based mutation strategy. In SSG-PSO, a collection of superior solutions is maintained and updated with the evolutionary process such that each particle can comprehensively learn from the recorded superior solutions. Tanweer et al. [43–45] proposed self-regulating particle swarm optimization (SR-PSO) by employing self-regulating inertia weight and self-perception of the global search direction. Cheng et al. [46] introduced a social learning mechanism into PSO and proposed social learning particle swarm optimization (SL-PSO). Wang et al. [47] proposed a novel surrogate-assisted particle swarm optimization inspired from committee-based active learning.

(4) Hybridized version:

Shieh et al. [48] proposed a modified particle swarm optimization algorithm with simulated annealing behavior (SA-PSO). Li et al. [49] proposed historical memory-based PSO (HMPSO) by using an estimation of distribution algorithm to estimate and preserve the distribution information of particles' historical promising Pbests. Ouyang et al. [50] presented Hybrid harmony search particle swarm optimization with global dimension selection (HHSPSO-GDS). Chen et al. [51] proposed a new biogeography-based learning strategy for particle swarm optimization (BLPSO). Aydilek [52] introduced a hybrid firefly and particle swarm optimization for computationally expensive numerical problems. Chen et al. [53] developed a particle swarm optimizer with two differential mutations (PSOTD). Bouyer and Hatamlou [54] proposed an efficient hybrid clustering method based on improved cuckoo search and modified particle swarm optimization algorithms. Tian and Shi [55] presented a modified particle swarm optimization with chaos-based initialization and robust update mechanisms to overcome premature convergence. Nenavath et al. [56] introduced a synergy of the sine-cosine algorithm and particle swarm optimizer.

Hybridizing PSO with other meta-heuristics or evolutionary algorithms (EAs) provides an effective approach to mitigate premature convergence. Because a genetic algorithm (GA) [57] possesses good exploration ability, combining GA with PSO to mitigate premature convergence has attracted great interest from the evolutionary community. Numerous hybrid PSO and GA algorithms for function optimization or real-life application are reported in the literature [58–63]. Gong et al. [64] pointed out that most of the existing hybrid GA and PSO algorithms are hybridized in a parallel manner, and their performances lie between those of PSO and GA in many optimization functions.

To further improve the performance of PSO, genetic learning particle swarm optimization (GL-PSO) [64] is proposed. GL-PSO adopts a two-cascading-layer structure, the first layer is for exemplar generation, and the second layer is for particle updates as per the normal PSO algorithm. GL-PSO improves the performance of PSO by generating high-quality exemplars to guide the evolution of the particles. Experiments have verified the effectiveness, efficiency, robustness and scalability of GL-PSO. However, the test results reveal that the performance of GL-PSO on complex composition functions is weaker than its performance on unimodal functions and basic multimodal functions [64]. The reason lies in the generation of exemplars. For GL-PSO employing the particle's Pbest and Gbest (global topology) for exemplar generation, this mechanism acts similar to a global version of PSO. Generating exemplars by global topology can achieve a high convergence rate, however, this strategy may cause the diversity to decrease quickly and impair the exploration ability of the algorithm.

To enhance the diversity and exploration of GL-PSO, the ring topology is employed for exemplar generation. There are different types of strategies for enhancing the diversity, such as a ring topology [32], multi-swarm [34], multi-layer [37], novel learning strategy [41], randomly selected neighbors [65], hybridization with other meta-heuristics [66] and so on. Because the ring topology can enlarge the potential search range and is easy to merge into GL-PSO for exemplar generation, it is adopted in this study. Furthermore, the global learning component (GLC) with linearly adjusted control parameters (including inertia weight and acceleration coefficients) is combined with GL-PSO to obtain robust performance.

To evaluate the respective effects of the ring topology and GLC, they are combined with GL-PSO separately and jointly. The proposed methods are tested on the latest CEC2017 [67] test suite and compared with seven selected PSO variants and five non-PSO meta-heuristics. The selected PSO variants are Particle Swarm Optimization with Constriction Factor (PSO-cf) [28], Fully Informed Particle Swarm (FIPS) [32], Comprehensive Particle Swarm Optimizer (CLPSO) [41], Social Learning Particle Swarm Optimization (SL-PSO) [46], Heterogeneous Comprehensive Learning Particle Swarm Optimization (HCLPSO) [68], Ensemble Particle Swarm Optimizer (EPSO) [69], and Genetic Learning Particle Swarm Optimization (GL-PSO) [64]. The selected non-PSO meta-heuristics are Artificial Bee Colony (ABC) [70], Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [71], Across Neighborhood Search (ANS) [72], Cuckoo Search (CS) [73], and Grey Wolf Optimizer (GWO) [74]. The Wilcoxon signed-rank test [75] and Friedman test [76] are employed to compare the performance of the involved algorithms. The test results indicate that employing the ring topology for exemplar generation can enhance the diversity and exploration capability of GL-PSO; combining GLC with GL-PSO, the resultant GGL-PSO can preserve sufficient diversity in the early stage, but the GLC cannot yield significant improvements on GL-PSO. With both the ring topology and GLC, GGL-PSOD exhibits high performance and strong robustness on the CEC 2017 test suite. It performs better than the seven representative PSO variants and five non-PSO meta-heuristics. GLC slows down the convergence speed of GGL-PSOD, but it can improve the adaptability of GGL-PSOD.

The rest of this paper is organized as follows: Section 2 reviews the related works. Section 3 introduces the methodologies. Section 4 reports the experimental results, and Section 5 discusses and concludes the paper.

2. Related works

2.1. Canonical PSO and important variants

In the canonical PSO, each particle is guided by its own personal best position (Pbest) and global best position (Gbest). The velocity and position of PSO are updated according to eqs. (1) and (2) [19].

$$v_{i,d} = \omega v_{i,d} + c_1 \cdot r_{1,d} \cdot \left(p_{i,d} - x_{i,d}\right) + c_2 \cdot r_{2,d} \cdot \left(g_d - x_{i,d}\right) \tag{1}$$

$$x_{i,d} = x_{i,d} + v_{i,d} \tag{2}$$

where $\boldsymbol{V_i} = [v_{i,1}, v_{i,2}, \ldots, v_{i,D}]$ and $\boldsymbol{X_i} = [x_{i,1}, x_{i,2}, \ldots, x_{i,D}]$ denote the velocity and position of particle i ($i = 1,2, \ldots, Ps$, where Ps is the population size), respectively. D is the dimensionality. $\boldsymbol{P_i} = [p_{i,1}, p_{i,2}, \ldots, p_{i,D}]$ denotes the Pbest of particle i, and $\boldsymbol{G} = [g_1, g_2, \ldots, g_D]$ is the Gbest of the entire swarm. ω denotes the inertia weight, and c_1 and c_2 are acceleration coefficients. $r_{1,d}$ and $r_{2,d}$ are two uniform distributed random numbers within the range [0,1].

The canonical PSO employs the Pbest and Gbest to guide the motion of the swarm to achieve a high convergence rate, but the canonical PSO suffers from premature convergence. To make good use of the information of the whole entire swarm, Mendes et al. [32] proposed the fully informed particle swarm (FIPS). The velocity of FIPS is calculated according to eqs. (3)–(5).

$$v_{i,d} = \chi\left(v_{i,d} + \varphi \cdot \left(p_{m,d} - x_{i,d}\right)\right) \tag{3}$$

$$p_{m,d} = \frac{\sum_{k \in N} w(k) \cdot \varphi_{k,d} \cdot p_{k,d}}{\sum_{k \in N} w_k \cdot \varphi_{k,d}} \tag{4}$$

$$\varphi_{k,d} = U\left[0, \frac{\varphi_{max}}{N}\right], \quad k \in N \tag{5}$$

where χ is the constriction factor and φ is the summation of the accelerate coefficients. The search of particle i converges on a point $\boldsymbol{P_m} = [p_{m,1}, p_{m,2}, \ldots, p_{m,D}]$ in the search space. The function $w(\cdot)$ may describe any aspect of the particle that is hypothesized to be relevant. U denotes a uniformly distributed random number. N is the set of neighbor particles. The ring topology FIPS is fit for solving complex multimodal problems.

To improve the performance of PSO on multimodal problems, Liang et al. [41] proposed the comprehensive learning particle swarm optimizer (CLPSO). The CLPSO introduced a novel comprehensive learning strategy whereby all other particles' Pbests are used to update a particle's velocity, and different dimensions of one particle may learn from different particles' Pbests.

The velocity of the particles in CLPSO is updated according to eq. (6).

$$v_i^d = \omega v_i^d + c^* rand_i^d \left(p_{fi(d)}^d - x_i^d\right) \tag{6}$$

where $\boldsymbol{fi} = [fi(1), fi(2), \ldots, fi(D)]$ defines which particles' ***Pbests*** the ith particle should follow. $p_{fi(d)}^d$ can be the corresponding dimension of any particle's ***Pbest*** including its own ***Pbest***, and the decision depends on the probability P_C. P_C is generated according to eq. (7) [41].

$$P_c(i) = 0.05 + 0.45^* \frac{\left(\exp\left(\frac{10(i-1)}{Ps-1}\right) - 1\right)}{\exp(10) - 1} \tag{7}$$

For each dimension of the ith particle, if the randomly generated number is greater than $P_C(i)$, it will learn from its own Pbest; otherwise, this dimension will learn from the particle with a better fitness value out of two tournament-selected particles. The test results show that CLPSO exhibits high performance on multimodal problems compared to other state-of-the-art PSO variants.

2.2. Genetic learning PSO

GL-PSO employs a two-cascading-layer structure to generate high-quality exemplars. The first layer adopts the crossover, mutation and selection operators of the genetic algorithm (GA) [57] to breed promising exemplars. The second layer employs the normal PSO algorithm to update the velocity and position of the swarm. The pseudo code for generating an exemplar for particle i according to GL-PSO [64] is presented in Algorithm 1. The velocity of GL-PSO is updated according to eq. (8)

$$v_{i,d} = \omega v_{i,d} + c \cdot r \cdot \left(e_{i,d} - x_{i,d}\right) \tag{8}$$

where $e_{i,d}$ denotes the exemplar for the dth dimension of the ith particle.

Algorithm 1
Pseudo code for breeding a GL exemplar.

```
For d = 1 to D do
  Randomly select a particle k∈{1,2, …,Ps};
  /*Exemplar update: Crossover*/
  If f(Pi)<f(Pk) Then
    oi,d = rd·pi,d+(1-rd)·gd;
  Else
    oi,d = pk,d
  End if
End for
  /*Exemplar update: Mutation*/
For d = 1 to D do
  If rand(0,1)<pm Then
    oi,d = rand(lbd,ubd);
  End if
End for
/*Exemplar update: Selection*/
Evaluate f(Oi)
If f(Oi)<f(Ei) Then
  Ei = Oi;
End if
If f(Ei) ceases improving for sg generations Then
  Select Ej by 20%Ps tournament selection
End if
```

Note: $r_d \in [0,1]$ is a uniform distributed random number. $\boldsymbol{O_i} = [o_{i,1}, o_{i,2}, \ldots, o_{i,D}]$ is a candidate exemplar for particle i.

Because GL-PSO moves the particles in a promising direction under the guidance of high-quality exemplars, it achieves high performance on the CEC2013 test suite [77]. However, when GL-PSO breeds exemplars according to a global topology, this mechanism can guarantee the quality of exemplars but may cause the diversity of exemplars to decrease quickly and impair the exploration ability of the algorithm.

3. Methodologies

To further improve the performance of GL-PSO, two modifications are employed. The first is to employ a ring topology for breeding exemplars to preserve the diversity of the generated exemplars, which is helpful in enhancing exploration. The second is to merge GLC with linearly adjusted control parameters in GL-PSO, which can improve the algorithm's adaptability by regulating the balance between exploration and exploitation of the resultant algorithm during the optimization process.

3.1. Diversity enhancement by ring topology

Diversity is very important for improving the performance of PSO. To achieve high performance on multimodal problems, GL-PSO should enhance diversity, especially in the early stage of evolution. The ring-topology PSO favors the preservation of diversity and exhibits high performance on complex multimodal problems. Therefore, adopting a ring topology instead of a global topology for exemplar generation facilitates the maintenance of diversity. The ring topology generates the positions of exemplars according to eqs. (9)–(11).

$$O_{i,d} = r_{\mathrm{d}} \cdot p_{ni1,d} + (1 - r_d) \cdot p_{ni2,d} \tag{9}$$

$$\begin{cases} n_{i1} = i - 1 & (i > 1) \\ n_{i1} = M & (i = 1) \end{cases} \tag{10}$$

$$\begin{cases} n_{i2} = i + 1 & (i < M) \\ n_{i2} = 1 & (i = M) \end{cases} \tag{11}$$

where n_{i1} and n_{i2} are indexes of two directly connected neighbor particles of particle i in the ring topology. The ring topology makes use of a couple of neighbor particles to breed diversity-enhanced exemplars, which can enhance the exploration ability of the resultant algorithm. When enhancing the diversity of GL-PSO by employing a ring topology for exemplar generation (Ref. Algorithm 2, line: 4–18), the resultant algorithm is referred to as genetic learning particle swarm optimization with diversity enhancement (GL-PSOD).

3.2. Global learning component

GL-PSO and GL-PSOD have no independent GLC, so it is inconvenient to adjust their exploration and exploitation dynamically. To improve the algorithm's adaptability via regulation of exploration and exploitation in the optimization process, the GLC is combined with GL-PSOD, and the resultant PSO algorithm is referred to as global genetic learning particle swarm optimization with diversity enhancement (GGL-PSOD). The velocity of GGL-PSOD is updated according to eq. (12).

$$v_{i,d} = \omega v_{i,d} + c_1 \cdot r_{1,d} \cdot (e_{i,d} - x_{i,d}) + c_2 \cdot r_{2,d} \cdot (g_d - x_{i,d}) \tag{12}$$

GGL-PSOD has two learning components serving two targets: the genetic learning component for preserving diversity and enhancing exploration; and the global learning component for enhancing exploitation. Ratnaweera et al. [29] pointed out that, "Generally, in population-based optimization methods, it is desirable to encourage the individuals to wander through the entire search space, without clustering around local optima, during the early stages of the optimization. On the other hand, during the latter stages, it is very important to enhance convergence toward the global optima, to find the optimum solution efficiently." According to Ratnaweera's instruction, the control parameters of GGL-PSOD are linearly adjusted during the optimization process to enhance exploration in the early stage and enhance exploitation in the latter stage according to eqs. (13)–(15).

$$\omega = \omega_i - \frac{iter}{iter_{\max}} (\omega_i - \omega_f) \tag{13}$$

$$c_1 = c_{1,i} - \frac{iter}{iter_{\max}} (c_{1,i} - c_{1,f}) \tag{14}$$

$$c_2 = c_{2,i} + \frac{iter}{iter_{\max}} (c_{2,f} - c_{2,i}) \tag{15}$$

where ω_i and ω_f denote the initial and final value of ω, respectively. *iter* and $iter_{max}$ stand for the current iteration number and the maximum iteration number, respectively. $c_{1,i}$ and $c_{1,f}$ are the initial and final value of c_1, respectively. $c_{2,i}$ and $c_{2,f}$ are the initial and final value of c_2, respectively. Ratnaweera et al. [29] suggested that ω is adjusted in a linearly decreasing manner from 0.9 to 0.4; c_1 is linearly varied from 2.5 to 0.5, while c_2 is linearly varied from 0.5 to 2.5.

The pseudo code of GGL-PSOD is presented in Algorithm 2. The main cycles of GL-PSO, GL-PSOD and GGL-PSOD are in accordance with Algorithm 2. Both GL-PSO and GL-PSOD update the velocity according to eq. (8), while GGL-PSOD updates the velocity according to eq. (12). GL-PSO employs a global topology for exemplar generation (Ref. line 5 of Algorithm 1), while GL-PSOD and GGL-PSOD adopt a ring topology for exemplar generation (Ref. line 8 of Algorithm 2). To evaluate the separate effects of diversity enhancement by ring topology and GLC, global genetic learning particle swarm optimization (GGL-PSO) is developed by combining GLC with GL-PSO. GGL-PSO follows the main cycle of Algorithm 2, employs a global topology for exemplar generation (Ref. Algorithm 1), and adopts eq. (12) to update the velocity. A basic test is carried out in Section 4.2 to compare the characteristics of GL-PSO, GL-PSOD, GGL-PSO and GGL-PSOD.

Algorithm 2
Pseudo code for GGL-PSOD.

1 Initialization
2 **Repeat**
3 **For** $i = 1$ to Ps do
4 **For** $d = 1$ to D **do**
5 Randomly select a particle $k \in \{1,2, \ldots, Ps\}$;
6 /*Exemplar update: Crossover*/
7 **If** $f(\boldsymbol{P}_i) < f(\boldsymbol{P}_k)$ **Then**
8 $o_{i,d} = r_{\mathrm{d}} \cdot p_{ni1,d} + (1 - r_d) \cdot p_{ni2,d}$
9 **Else**
10 $o_{i,d} = p_{k,d}$;
11 **End if**
12 **End for**
13 /*Exemplar update: Mutation*/
14 **For** $d = 1$ **to** D **do**
15 **If** $rand(0,1) < p_m$ **Then**
16 $o_{i,d} = rand(lb_d, ub_d)$;
17 **End if**
18 **End for**
19 /*Exemplar update: Selection*/
20 Evaluate $f(\boldsymbol{O}_i)$;
21 **If** $f(\boldsymbol{O}_i) < f(\boldsymbol{E}_i)$ **Then**
22 $\boldsymbol{E}_i = \boldsymbol{O}_i$;
23 /*Challenge Gbest by exemplar*/
24 **If** $f(\boldsymbol{O}_i) < f(\boldsymbol{G})$ **Then**
25 $\boldsymbol{G} = \boldsymbol{O}_i$;
26 **End if**
27 **End if**
28 **If** $f(\boldsymbol{E}_i)$ ceases improving for s_g generations **Then**
29 Select E_j by 20%Ps tournament selection of Pbests;
30 **End if**
30 /*Particle update*/
32 Linearly adjusting ω, c_1 and c_2 according to eqs. (13)–(15)
33 **For** $d = 1$ **to** D **do**
34 $v_{i,d} = \omega v_{i,d} + c_1 \cdot r_{1,d} \cdot (e_{i,d} - x_{i,d}) + c_2 \cdot r_{2,d} \cdot (g_d - x_{i,d})$;
35 $x_{i,d} = x_{i,d} + v_{i,d}$;
36 **End for**
37 Evaluate $f(\boldsymbol{X}_i)$ and update $\boldsymbol{P}_i$, $\boldsymbol{G}$;
38 **End for**
Until terminal condition

4. Experimental work

4.1. Experimental setup

To test the performance of the proposed GGL-PSOD, the latest CEC2017 test suite on single-objective real-parameter numerical optimization (CEC2017 test suite) [67] is employed. A summary of the CEC2017 test suite is given in Table 1. The CEC2017 test suite contains three unimodal functions ($f_1 \sim f_3$), seven simple multimodal functions

Table 1
Summary of the CEC2017 test suite.

	No.	Functions	$F_i^* = Fi(x^*)$
Unimodal functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Sum of Different Power Function	200
	3	Shifted and Rotated Zakharov Function	300
Simple Multimodal functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Rastrigin's Function	500
	6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	7	Shifted and Rotated LunacekBi_Rastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefel's Function	1000
Hybrid functions	11	Hybrid Function 1 (N = 3)	1100
	12	Hybrid Function 2 (N = 3)	1200
	13	Hybrid Function 3 (N = 3)	1300
	14	Hybrid Function 4 (N = 4)	1400
	15	Hybrid Function 5 (N = 4)	1500
	16	Hybrid Function 6 (N = 4)	1600
	17	Hybrid Function 7 (N = 5)	1700
	18	Hybrid Function 8 (N = 5)	1800
	19	Hybrid Function 9 (N = 5)	1900
	20	Hybrid Function 10 (N = 6)	2000
Composition functions	21	Composition Function 1 ($N = 3$)	2100
	22	Composition Function 2 ($N = 3$)	2200
	23	Composition Function 3 ($N = 4$)	2300
	24	Composition Function 4 ($N = 4$)	2400
	25	Composition Function 5 ($N = 5$)	2500
	26	Composition Function 6 ($N = 5$)	2600
	27	Composition Function7 ($N = 6$)	2700
	28	Composition Function 8 ($N = 6$)	2800
	29	Composition Function 9 ($N = 3$)	2900
	30	Composition Function 10 ($N = 3$)	3000
Search Range: $[-100,100]^D$			

Note: x^* stands for the global optima. $Fi(\cdot)$ is the fitness value functions.

($f_4 \sim f_{10}$), ten hybrid functions ($f_{11} \sim f_{20}$) and ten composition functions ($f_{21} \sim f_{30}$). The CEC2017 benchmark functions are developed with several novel features, such as new basic problems, composing test problems by extracting features dimension-wise from several problems, a graded level of linkages, rotated trap problems, and so on. The exact equations of the test functions are not allowed to be used. Under the guidance of the CEC2017 test suite, the search range is set as [-100, 100], and the maximum number of function evaluations (*FEs*) is 10000*D. The dimensionality is set as $D = 30$ in this study. Each algorithm is run for 51 runs, and the mean errors, Wilcoxon signed-rank test [75,76,78] and Friedman test [79] are employed to compare the performances of the involved algorithms. The test results are presented in Tables 3–6, 9, 10 and the best results are highlighted in bold.

4.2. Parameter configuration

To test the separate effectiveness of adopting a ring topology and GLC with linearly adjusted control parameters, the proposed GL-PSOD, GGL-PSO and GGL-PSOD are tested and compared with three state-of-the-art PSO variants, four recently reported PSO variants, including GL-PSO, and five representative non-PSO meta-heuristics. PSO-cf [28] is a global version of PSO with constriction coefficients to control the dynamic characteristics of the particle swarm. FIPS [32] makes all the individuals "fully informed", and the Uring version of FIPS adopts a Uring topology and performs well on complex multimodal problems. CLPSO [41] uses a novel comprehensive learning strategy whereby the entire swarm's Pbests are employed to update a particle's velocity. SL-PSO [46] imitates the social learning behavior of animals. In SL-PSO, one particle learns from any better particles in the current swarm. HCLPSO [68] is developed from CLPSO, and it employs two subgroups to enhance exploration and exploitation, respectively. EPSO [69] employs a self-adaptive scheme to ensemble [80,81] five PSO strategies. In EPSO, the suitable PSO strategy is gradually selected through a meritocracy approach to guide the particle's flying direction in the current stage of the search process. Artificial Bee Colony (ABC) [70] simulates the intelligent foraging behavior of a honeybee swarm. Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [71] employs a covariance matrix adaptation strategy to reduce the number of generations required for evolution strategy convergence to the optimum. Across Neighborhood Search (ANS) [72] employs a group of individuals who collaboratively search the solution space. Cuckoo Search [73] is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. The Grey Wolf Optimizer (GWO) [82] mimics the leadership hierarchy and hunting mechanism of grey wolves in nature.

All of the PSO variants and other meta-heuristics adopt the recommended parameter settings in the original paper. The parameter configurations of PSO variants are presented in Table 2.

4.3. Characteristic test

In GL-PSO, each particle learns from its exemplars, so the diversity of exemplars is very important to the exploration of GL-PSO. In this study, the diversity of exemplars (denoted by $diversity_{ep}$) is evaluated by the mean Euclidian distance between each exemplar and the Gbest according to eq. (16). Fig. 1(a) indicates that, on the unimodal function f_2, the diversity of GL-PSO decreases the fastest. With the ring topology to enhance diversity, GL-PSOD and GGL-PSOD can preserve slightly more diversity than the corresponding GL-PSO and GGL-PSO. By introducing GLC with linearly adjusted control parameters to GL-PSO and GL-PSOD, the resultant GGL-PSO and GGL-PSOD can preserve much more diversity than the corresponding GL-PSO and GL-PSOD in the early stage. GGL-PSOD preserves the highest diversity in this test because it employs both a ring topology and GLC with linearly adjusted control parameters. On the multimodal function f_{16}, the diversity of GL-PSO decreases quickly, while GL-PSOD preserves slightly more diversity than GL-PSO. GGL-PSO preserves much more diversity than GL-PSOD. By introducing the ring topology to generate exemplars, GGL-PSOD preserves more diversity than GGL-PSO. Because GL-PSOD inherits the mutation operator employed in GL-PSO to generate exemplars, the occasional diversity increase of GL-PSOD during the optimization process can be seen in Fig. 1(b).

Table 2
Parameter settings of PSO variants.

No.	algorithm	year	Parameter settings
1	GL-PSO	2016	$Ps = 50$, $\omega = 0.7298$, $c = 1.49618$, $p_m = 0.01$, $s_g = 7$
2	GL-PSOD	/	$Ps = 50$, $\omega = 0.7298$, $c = 1.49618$, $p_m = 0.01$, $s_g = 7$
	GGL-PSO	/	$Ps = 50$, $\omega = 0.9$–0.4, $c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5, $p_m = 0.01$, $s_g = 7$
3	GGL-PSOD	/	$Ps = 50$, $\omega = 0.9$–0.4, $c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5, $p_m = 0.01$, $s_g = 7$
4	PSO-cf	2002	$Ps = 40$, $\chi = 0.7298$, $c_1 = c_2 = 2.05$
5	FIPS	2004	$Ps = 40$, $\chi = 0.7298$, $\varphi = 4.1$
6	CLPSO	2006	$Ps = 40$, $\omega = 0.9$–0.4, $c = 1.49445$, $P_{ci} = 0.05 + 0.45 * \frac{e^{\frac{10(i-1)}{ps-1}} - 1}{e^{10} - 1}$
7	SL-PSO	2015	$M = 100$, $\alpha = 0.5$, $\beta = 0.01$, $p_s = M + \frac{n}{10}$, $\varepsilon = \beta \times \frac{n}{M}$, $P_L^i = \left(1 - \frac{i-1}{M}\right)^{\alpha \cdot log(\frac{n}{M})}$
8	HCLPSO	2015	$Ps = 40$, $\omega = 0.99$–0.2, $c_1 = 2.5$–0.5, $c_2 = 0.5$–2.5, $c = 3$–1.5, $g_1 = 15$, $g_2 = 25$
9	EPSO	2017	Ref. [69]

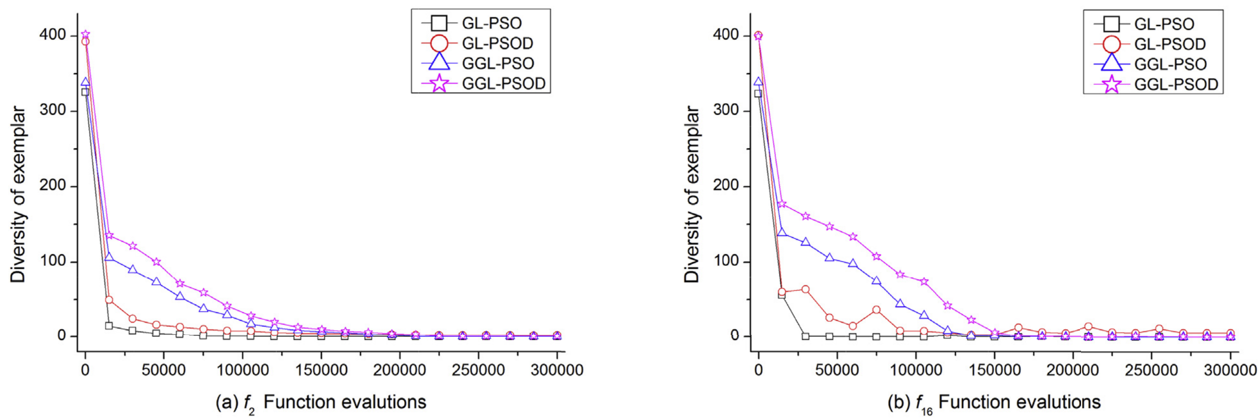

Fig. 1. Diversity curve of the exemplars.

$$diversity_{ep} = \frac{1}{Ps}\sum_{i=1}^{Ps}\sqrt{\sum_{j=1}^{D}\left(exemplar_{i,j} - Gbest_j\right)^2} \tag{16}$$

The convergence curves of four GL-PSO-based algorithms on f_2 and f_{16} are plotted in Section 4.5 Fig. 2(a) and Fig. 2(d), respectively. Fig. 2(a) indicates that, on f_2, GL-PSO converges the fastest in the early stage among the four GL-PSO-based algorithms. GL-PSOD converges relatively slowly compared to GL-PSO, but it surpasses GL-PSO in the middle stage. GGL-PSO and GGL-PSOD converge more slowly than the corresponding GL-PSO and GL-PSOD at the beginning because they employ relatively larger inertia weights to enhance diversity in the early stage. GGL-PSO and GGL-PSOD can maintain steady convergence rates in the middle stage. GGL-PSOD surpasses GL-PSO in the end. The order of performance on f_2 is GL-PSOD, GGL-PSOD, GL-PSO and GGL-PSO. Fig. 2(d) shows that, on the hybrid multimodal function f_{16}, GL-PSO and GL-PSOD converge quickly at the beginning, but they stagnate in the early stage. Though GGL-PSO and GGL-PSOD converge more slowly at the start, they surpass GL-PSO and GL-PSOD at the middle stage. The order of performance on f_{16} is GGL-PSOD, GGL-PSO, GL-PSOD and GL-PSO. The test results indicate that:

(1) Both the ring topology and GLC with linearly adjusted control parameters can enhance diversity. The former can yield a significant improvement in performance, while the latter can preserve much more diversity.
(2) The aforementioned two methods are compatible; combining both of them with GL-PSO can preserve the highest diversity and achieve the best overall performance among the four GL-PSO-based algorithms.

For a more comprehensive comparison test, refer to Section 4.4.

4.4. Numerical test

4.4.1. Comparison test among GL-PSO-based algorithms

In this section, the proposed GL-PSOD, GGL-PSO and GGL-PSOD are tested and compared with GL-PSO to demonstrate the separate degrees of effectiveness of the ring topology and GLC with linearly adjusted control parameters. The mean errors of 51 runs are presented in Table 3. The best results obtained for each function are highlighted in bold. The Wilcoxon signed-rank test results with a significance level of 0.05 [75,78] are presented in Table 3 as well. The symbols ">," " = " and "<" indicate that GGL-PSOD "performs significantly better than," "ties with" and "performs significantly worse than" the comparative algorithm, respectively. The row of "*w/t/l*" denotes the total numbers of times that GGL-PSOD "wins over," "ties with," and "loses to" the compared algorithm, respectively. The row of "*Best*" represents for the total number of times that the best performance is generated by the corresponding algorithm. The Friedman test [79] results are given in Table 4 and the best results are highlighted in bold. The row of "General rank" denotes the general performance order of the involved algorithms.

The test results in Table 3 show that, on three unimodal functions (f_1~f_3), GL-PSOD yields the best performance on f_1 and f_2, while GGL-PSOD obtains the best performance on f_3. Compared with GL-PSO, GGL-PSOD wins on two functions and loses on one function. Compared with GL-PSOD, GGL-PSOD wins on one function and loses on two functions. Compared with GGL-PSO, GGL-PSOD wins on three functions. GGL-PSOD performs better than GL-PSO and GGL-PSO but performs worse than GL-PSOD.

On seven simple multimodal functions (f_4~f_{10}), GL-PSO, GL-PSOD and GGL-PSOD exhibit the best performance on one, one and five functions, respectively. Compared with GL-PSO, GL-PSOD and GGL-PSO, GGL-PSOD wins on six, six and seven functions, respectively. GGL-PSOD performs much better than GL-PSO, GL-PSOD and GGL-PSO.

On ten hybrid functions (f_{11}~f_{20}), GL-PSO, GL-PSOD and GGL-PSOD achieve the best performance on three, three and four functions, respectively. GGL-PSOD performs better than GL-PSO, GL-PSOD and GGL-PSO on six, five and nine functions, respectively. The performance of GGL-PSOD is comparable to that of GL-PSOD and better than that of GL-PSO and GGL-PSO.

On ten composition functions (f_{21}~f_{30}), GL-PSO, GL-PSOD, GGL-PSO and GGL-PSOD exhibit the best performance on one, four, one and eight functions, respectively. GL-PSO, GL-PSOD, GGL-PSO and GGL-PSOD jointly win the best performance on f_{22}. GL-PSOD and GGL-PSOD tie for first on f_{25}. Compared with GL-PSO, GGL-PSOD wins on seven functions and ties with it on three functions. Compared with GL-PSOD, GGL-PSOD wins, ties with and loses on five, three and two functions, respectively. Compared with GGL-PSO, GGL-PSOD wins and ties with on eight and two functions, respectively. GGL-PSOD performs better than GL-PSO, GL-PSOD and GGL-PSO.

On all thirty tested functions, GL-PSO, GL-PSOD, GGL-PSO and GGL-PSOD exhibit the best performance on five, ten, one and eighteen functions, respectively. GGL-PSOD performs significantly better than, equivalently to and significantly worse than GL-PSO on twenty-one functions, three functions and six functions, respectively. Compared with GL-PSOD, GGL-PSOD performs significantly better than, equivalently to and significantly worse than GL-PSO on seventeen functions, three functions and ten functions, respectively. Compared with GGL-PSO, GGL-PSOD performs significantly better than, equivalently to and significantly worse than GL-PSO on twenty-seven functions, two functions and one function, respectively.

The Friedman test results in Table 4 indicate that GGL-PSOD obtains the best Friedman rank. The order of general performance in this test is GGL-PSOD, GL-PSOD, GGL-PSO and GL-PSO. The test results indicate that, with diversity enhancement via ring topology, GL-PSOD and GGL-

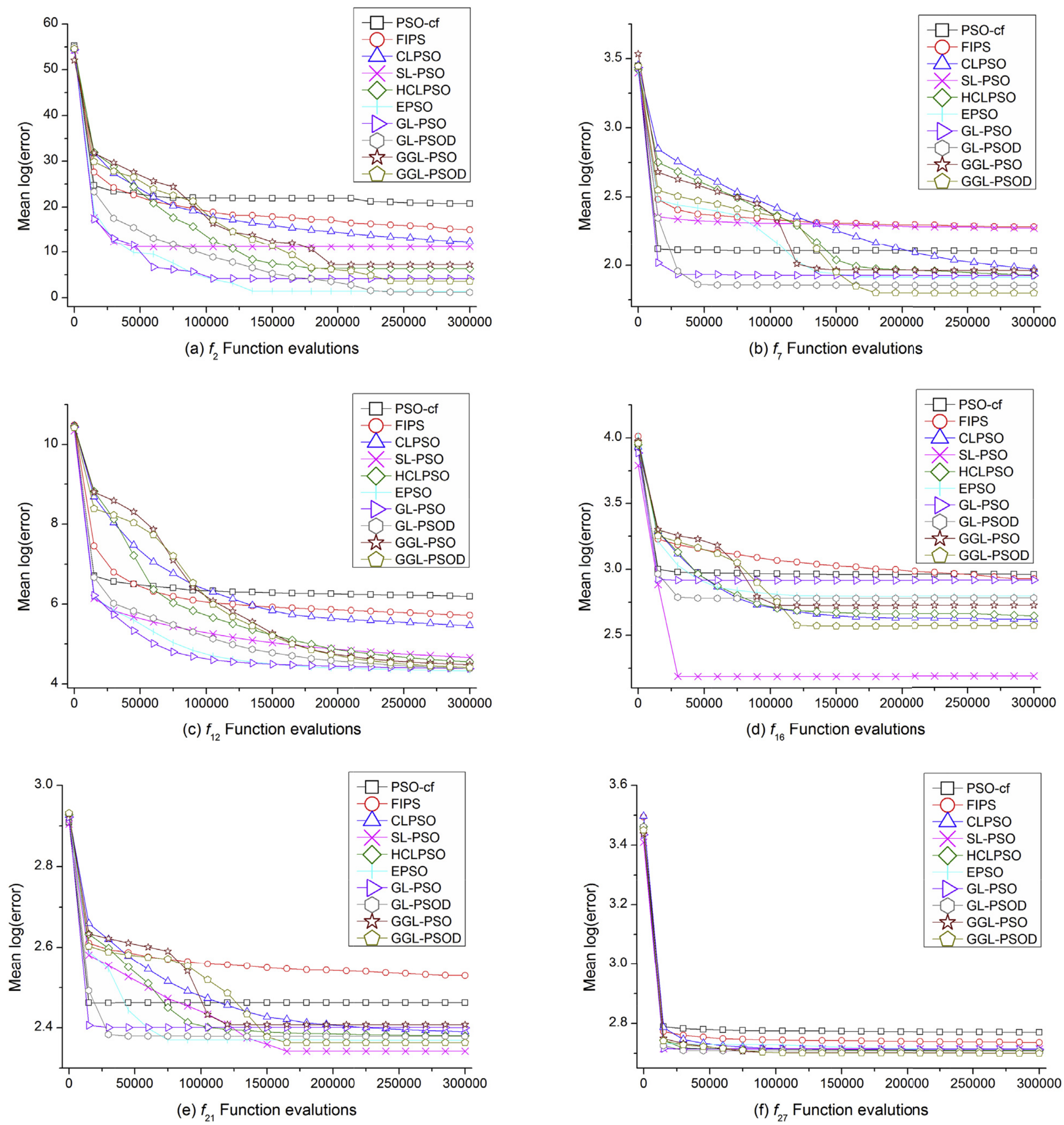

Fig. 2. Convergence curve of PSO algorithms.

PSOD perform better than the corresponding GL-PSO and GGL-PSO. When combining GLC with linearly adjusted control parameters in GL-PSO, the resultant GGL-PSO performs slightly better than GL-PSO. When combining both of the aforementioned modifications with GL-PSO, the resultant GGL-PSOD achieves the best performance and strong robustness on different types of CEC2017 functions. It outperforms the other three GL-PSO-based algorithms.

4.4.2. Comparison test with PSO variants

To show the competitiveness of the proposed GGL-PSOD, it is compared with three state-of-the-art PSO variants and three recent PSO variants. The test results in Table 5 show that, for three unimodal functions (f_1-f_3), EPSO and CLPSO exhibit the best performance on two and one functions, respectively. The Wilcoxon signed-rank test results show that, compared with PSO-cf, FIPS and CLPSO, GGL-PSOD wins on three, two and two functions, respectively. Compared with SL-PSO, HCLPSO and EPSO, GGL-PSOD wins on three, two and zero functions, respectively. GGL-PSOD performs better than the other involved peer PSO variants except for EPSO.

On seven simple multimodal functions (f_4~f_{10}), SL-PSO and GGL-PSOD exhibit the best performance on three and two functions, respectively. The Wilcoxon signed-rank test results show that, compared with PSO-cf, FIPS and CLPSO, GGL-PSOD wins on seven, five and six functions, respectively. Compared with SL-PSO, HCLPSO and EPSO, GGL-PSOD wins on three, five and seven functions, respectively. The performance of GGL-PSOD is secondary to that of SL-PSO.

On ten hybrid functions (f_{11}~f_{20}), both CLPSO and GGL-PSOD achieve the best performance on three functions. The Wilcoxon signed-rank test results show that, compared with PSO-cf, FIPS and CLPSO, GGL-PSOD performs better on ten, ten and seven functions, respectively.

Table 3
The comparison test results among GL-PSO-based algorithms on 30-*D* CEC2017 functions.

Func.	GL-PSO	GL-PSOD	GGL-PSO	GGL-PSOD
f_1	<2.93E+03	**<1.79E+03**	>5.06E+03	4.90E+03
f_2	>1.27E+04	**<1.35E+01**	>1.74E+07	3.64E+03
f_3	>1.82E+01	>2.25E+01	>5.52E-03	**3.95E-09**
f_4	**<2.53E+01**	>8.11E+01	>6.22E+01	6.09E+01
f_5	>5.19E+01	>3.98E+01	>5.07E+01	**2.69E+01**
f_6	>3.09E-04	**<4.91E-08**	>1.88E-03	1.27E-07
f_7	>8.45E+01	>7.18E+01	>9.20E+01	**6.31E+01**
f_8	>5.02E+01	>4.52E+01	>4.78E+01	**3.04E+01**
f_9	>7.63E+00	>2.40E-01	>6.41E+00	**1.42E-01**
f_{10}	>2.88E+03	>2.73E+03	>2.49E+03	**2.07E+03**
f_{11}	>6.90E+01	>5.95E+01	>7.13E+01	**3.08E+01**
f_{12}	**<2.43E+04**	<2.45E+04	>3.05E+04	2.59E+04
f_{13}	<9.86E+03	**<9.17E+03**	>1.62E+04	1.14E+04
f_{14}	**<1.72E+03**	>2.97E+03	>2.49E+03	2.20E+03
f_{15}	>3.39E+03	**<1.27E+03**	>8.02E+03	2.14E+03
F_{16}	>8.23E+02	>6.01E+02	>5.27E+02	**3.71E+02**
f_{17}	>1.88E+02	>1.67E+02	>1.36E+02	**6.13E+01**
f_{18}	**<5.44E+04**	<8.06E+04	<7.71E+04	9.04E+04
f_{19}	>7.90E+03	**<3.24E+03**	>4.87E+03	4.72E+03
f_{20}	>2.76E+02	>2.03E+02	>1.39E+02	**1.07E+02**
f_{21}	>2.51E+02	>2.39E+02	>2.56E+02	**2.31E+02**
f_{22}	**= 1.00E+02**	**= 1.00E+02**	**=1.00E+02**	**1.00E+02**
f_{23}	>4.04E+02	>3.88E+02	>4.01E+02	**3.84E+02**
f_{24}	>4.74E+02	=4.66E+02	>4.68E+02	**4.58E+02**
f_{25}	=3.90E+02	**=3.87E+02**	=3.88E+02	**3.87E+02**
f_{26}	>1.44E+03	>1.19E+03	>1.37E+03	**7.67E+02**
f_{27}	>5.17E+02	>5.10E+02	>5.03E+02	**5.02E+02**
f_{28}	=3.60E+02	**<3.52E+02**	>3.75E+02	3.59E+02
f_{29}	>6.72E+02	>5.67E+02	>5.21E+02	**4.50E+02**
f_{30}	>5.45E+03	**<2.96E+03**	>6.12E+03	3.99E+03
w/t/l	21/3/6	17/3/10	27/2/1	
Best	5	10	1	18

Table 4
The Friedman test results of GL-PSO-based algorithms on 30-*D* CEC2017 functions.

	GL-PSO	GL-PSOD	GGL-PSO	GGL-PSOD
Friedman rank	3.067	2.1	3.033	**1.567**
General rank	4	2	3	**1**

Compared with SL-PSO, HCLPSO and EPSO, GGL-PSOD wins on six, eight and five functions, respectively. GGL-PSOD and EPSO tie for first on hybrid functions among the tested PSO variants.

On ten composition functions (f_{21}~f_{30}), SL-PSO and GGL-PSOD achieve the best performance on six and five functions, respectively. On f_{22}, FIPS, SL-PSO, HCLPSO and GGL-PSOD jointly have the best performance. On f_{25}, CLPSO, SL-PSO, HCLPSO and GGL-PSOD tie for first place. The Wilcoxon signed-rank test results show that, compared with PSO-cf, FIPS and CLPSO, GGL-PSOD wins on ten, eight and seven functions, respectively. Compared with SL-PSO, HCLPSO and EPSO, GGL-PSOD wins on four, seven, and four functions, respectively. GGL-PSOD and SL-PSO tie for first on composition functions among the tested PSO variants.

On all thirty tested functions, SL-PSO, GGL-PSOD and CLPSO exhibit the best performance on ten, nine and six functions. The Wilcoxon signed-rank test results show that GGL-PSOD performs much better than the three state-of-the-art PSO algorithms. Compared with the three recently reported PSO variants, GGL-PSOD again performs better. Compared with SL-PSO, GGL-PSOD wins, ties with and loses on sixteen, two and twelve functions, respectively. Compared with HCLPSO, GGL-PSOD wins, ties with and loses on twenty-two, three and five functions, respectively. Compared with EPSO, GGL-PSOD wins, ties with and loses on sixteen, four and ten functions, respectively. GGL-PSOD wins joint first place on hybrid functions and composition functions, wins second place on unimodal functions and simple multimodal functions, and exhibits robust performance on different types of benchmark functions. The Friedman test results in Table 6 show that GGL-PSOD achieves the best Friedman rank. The general performance order is GGL-PSOD, HCLPSO, SL-PSO, EPSO, CLPSO, FIPS and PSO-cf. GGL-PSOD performs better than the six involved peer PSO algorithms.

To compare the success rates and search efficiencies of the proposed algorithms, the success rates and mean function evaluations of the involved PSO variants in 51 independent runs are reported in Tables 7 and 8. It is difficult for the algorithms to achieve zero error on CEC2017 functions, so we define the constant ε as the acceptance threshold value of a satisfactory solution (or near-optimal solution). If an algorithm obtains a function value smaller than ε in a run, this run is viewed as a success. In this study, we set $\varepsilon = 50$ for f_1~f_{20} and $\varepsilon = 500$ for f_{21}~f_{30}. The success rate on a function is defined as the number of successful runs divided by the number of total runs. The mean function evaluations is defined as the mean function evaluations consumed by one algorithm to successfully obtain the satisfactory solutions defined by the constant ε. The best results in Tables 7 and 8 are highlighted in bold.

Table 7 indicates that GGL-PSOD, SL-PSO and GL-PSOD achieve the highest success rate on twelve, twelve and eleven functions, respectively. Both GL-PSO and GGL-PSO generate the highest success rate on nine functions. CLPSO, HCLPSO, EPSO, FIPS and PSO-cf yield the highest success rate on nine, nine, nine, six and four functions, respectively. The test results in Table 7 show that GGL-PSOD, SL-PSO, GL-PSOD exhibit high success rates. With the diversity enhancement via the ring topology, both GGL-PSOD and GL-PSOD achieve higher success rates than the corresponding GGL-PSO and GL-PSO.

With regards to search efficiency, Table 8 shows that SL-PSO, PSO-cf and GL-PSO yield the lowest mean function evaluations on nine, seven and six functions, respectively, to reach the satisfactory solutions. The proposed GGL-PSOD algorithm consumes the least mean function evaluations on two functions. Because GGL-PSO and GGL-PSOD enhance exploration in the early stage, they converge relatively slowly and need more function evaluations to obtain the satisfactory solutions. GGL-PSOD increases the success rate at the cost of the convergence rate.

4.4.3. Comparison test with non-PSO meta-heuristics

In this section, the proposed GGL-PSOD is compared with five representative non-PSO meta-heuristics to further demonstrate its competitiveness. The test results in Table 9 show that, on three unimodal functions (f_1~f_3), CMA-ES achieves the best performance on three functions. The Wilcoxon signed-rank test results show that, compared with ABC, CMA-ES, CS, ANS and GWO, GGL-PSOD wins on two, zero, two, two and three functions, respectively. The performance of GGL-PSOD is dominated by that of CMA-ES, but GGL-PSOD performs better than the rest of the meta-heuristics.

On seven simple multimodal functions (f_4~f_{10}), GGL-PSOD, ABC and CMA-ES win the best performance on five, one and one functions, respectively. GGL-PSOD performs better than ABC, CMA-ES, CS, ANS and GWO on five, six, six, six and seven functions, respectively. GGL-PSOD performs better than the other involved meta-heuristics.

On ten hybrid functions (f_{11}~f_{20}), CS, GGL-PSOD and CMA-ES win the best performance on four, three and two functions, respectively. GGL-PSOD performs better than ABC, CMA-ES, CS, ANS and GWO on ten, three, four, six and ten functions, respectively. GGL-PSOD performs better than ABC, ANS and GWO but performs worse than CMA-ES and CS.

On ten composition functions (f_{21}~f_{30}), GGL-PSOD, ABC, CMA-ES and CS win the best performance on four, four, one and one functions, respectively. GGL-PSOD performs better than ABC, CMA-ES, CS, ANS and GWO on six, seven, eight, nine and ten functions, respectively. GGL-PSOD yields the best performance on composition functions.

On all thirty tested functions, GGL-PSOD, CMA-ES, ABC and CS win the best performance on twelve, seven, five and five functions, respectively. Compared with ABC, CMA-ES, CS, ANS and GWO, GGL-PSOD performs better on twenty-three, sixteen, twenty, twenty-three and thirty functions, respectively. The Friedman test results in Table 10 show that GGL-PSOD ranks first in this test. The general performance order on

Table 5
The comparison test results of PSO variants on 30-*D* CEC2017 functions.

Func.	PSO-cf	FIPS	CLPSO	SL-PSO	HCLPSO	EPSO	GGL-PSOD
f_1	>5.82E+03	<3.51E+03	**<1.44E+01**	>5.11E+03	<7.77E+01	<2.68E+03	4.90E+03
f_2	>5.32E+20	>1.11E+15	>2.19E+12	>1.52E+11	>2.14E+06	**<2.21E+01**	3.64E+03
f_3	>1.72E-08	>4.09E+03	>1.92E+04	>7.05E+03	>1.58E-03	**<8.34E-13**	3.95E-09
f_4	>1.45E+02	>1.25E+02	>7.26E+01	>7.74E+01	>6.85E+01	>6.91E+00	**6.09E+01**
f_5	>8.08E+01	>1.37E+02	>4.22E+01	**<1.84E+01**	>4.28E+01	>4.25E+01	2.69E+01
f_6	>6.67E+00	<2.60E-08	<7.09E-08	>1.68E-06	**<3.96E-13**	>3.70E-05	1.27E-07
f_7	>1.28E+02	>1.92E+02	>9.41E+01	>1.88E+02	>8.52E+01	>8.19E+01	**6.31E+01**
f_8	>9.03E+01	>1.36E+02	>4.96E+01	**<1.74E+01**	>4.37E+01	>4.73E+01	3.04E+01
f_9	>6.50E+02	**<0.00E+00**	>5.86E+01	<8.40E-02	>2.07E+01	>2.40E+01	1.42E-01
f_{10}	>3.64E+03	>6.31E+03	>2.21E+03	**<1.01E+03**	=2.07E+03	>2.15E+03	2.07E+03
f_{11}	>1.37E+02	>7.09E+01	>4.79E+01	**<2.81E+01**	>5.51E+01	>7.41E+01	3.08E+01
f_{12}	>1.55E+06	>5.29E+05	>2.93E+05	>4.59E+04	>3.59E+04	**<2.13E+04**	2.59E+04
f_{13}	>1.06E+06	>1.33E+04	**<2.80E+02**	>1.57E+04	<7.37E+02	<3.08E+03	1.14E+04
f_{14}	>4.57E+04	>6.17E+03	>2.05E+04	>1.74E+04	>3.44E+03	>2.70E+03	**2.20E+03**
f_{15}	>1.46E+04	>1.65E+04	**<1.09E+02**	<1.89E+03	>3.31E+02	<4.33E+02	2.14E+03
F_{16}	>9.09E+02	>8.41E+02	>4.14E+02	**<1.54E+02**	>4.41E+02	>6.22E+02	3.71E+02
f_{17}	>4.50E+02	>1.63E+02	>8.36E+01	>9.08E+01	>9.81E+01	>1.68E+02	**6.13E+01**
f_{18}	>1.33E+05	>3.08E+05	>1.00E+05	>1.02E+05	>9.36E+04	**<6.49E+04**	9.04E+04
f_{19}	>1.30E+04	>5.78E+03	**<6.66E+01**	<2.23E+03	<1.58E+02	<3.86E+02	4.72E+03
f_{20}	>4.71E+02	>1.93E+02	>1.19E+02	>1.33E+02	>1.27E+02	>2.00E+02	**1.07E+02**
f_{21}	>2.90E+02	>3.38E+02	>2.45E+02	**<2.20E+02**	>2.40E+02	=2.34E+02	2.31E+02
f_{22}	>2.55E+03	**= 1.00E+02**	=1.03E+02	**= 1.00E+02**	**= 1.00E+02**	=1.01E+02	**1.00E+02**
f_{23}	>4.39E+02	>4.62E+02	>3.96E+02	**<3.71E+02**	>3.94E+02	>4.00E+02	3.84E+02
f_{24}	>5.06E+02	>5.64E+02	>4.79E+02	**<4.48E+02**	>4.67E+02	=4.58E+02	4.58E+02
f_{25}	>4.17E+02	=3.92E+02	**= 3.87E+02**	**= 3.87E+02**	**= 3.87E+02**	=3.88E+02	**3.87E+02**
f_{26}	>2.25E+03	>1.94E+03	**<3.04E+02**	>1.19E+03	<4.31E+02	<7.26E+02	7.67E+02
f_{27}	>5.89E+02	>5.46E+02	>5.11E+02	>5.19E+02	>5.12E+02	>5.18E+02	**5.02E+02**
f_{28}	>4.39E+02	>4.07E+02	>4.13E+02	>3.76E+02	>3.76E+02	**<3.22E+02**	3.59E+02
f_{29}	>9.55E+02	>6.98E+02	>5.15E+02	>4.89E+02	>5.09E+02	>5.97E+02	**4.50E+02**
f_{30}	>4.61E+04	>2.81E+04	>4.70E+03	**<3.71E+03**	>4.50E+03	>4.08E+03	3.99E+03
w/t/l	30	25/2/3	22/2/6	16/2/12	22/3/5	16/4/10	
Best	0	2	6	10	2	5	9

Table 6
The Friedman test results of PSO variants on 30-*D* CEC2017 functions.

	PSO-cf	FIPS	CLPSO	SL-PSO	HCLPSO	EPSO	GGL-PSOD
Friedman rank	6.533	5.5	3.667	3.167	3.033	3.433	**2.300**
General rank	7	6	5	3	2	4	**1**

all thirty tested functions is GGL-PSOD, ANS, ABC, CS, CMA-ES, and GWO. The test results indicate that the proposed GGL-PSOD performs better than the five involved non-PSO meta-heuristics.

4.5. Convergence rate

To compare the convergence rate of the involved PSO algorithms, the convergence curves on six representative functions are presented in Fig. 2. On the unimodal function f_2, Fig. 2(a) shows that EPSO converges quickly and achieves almost the same mean error as GL-PSOD. GL-PSO also converges quickly, while in the latter stage, it is surpassed by GL-PSOD and GGL-PSOD. With diversity enhancement by ring topology, GL-PSOD converges more slowly than GL-PSO at the beginning, but it outperforms GL-PSO in the middle stage and yields the lowest mean error finally. With GLC and linearly adjusted control parameters, GGL-PSO and GGL-PSOD converge more slowly than the corresponding GL-PSO and GL-PSOD in the early stage. GGL-PSO and GGL-PSOD maintain steady convergence rates and surpass PSO-cf, FIPS, CLPSO and SL-PSO. GGL-PSOD achieves the third-lowest mean error in the end. PSO-cf, FIPS and CLPSO converge relatively slowly. With the ring topology for the enhancement of diversity, GGL-PSOD outperforms GGL-PSO.

For simple multimodal functions, Fig. 2(b) shows that, on f_7, GL-PSO and GL-PSOD converge quickly and generate very low mean errors. GL-PSOD performs better than GL-PSO. GGL-PSO and GGL-PSOD converge relatively slowly at the beginning, but they speed up in the middle stage. GGL-PSOD and GL-PSOD occupy the top two places in the end. SL-PSO and FIPS converge quickly at the start, but they become trapped in local optima very early.

For hybrid functions, Fig. 2(c) shows that, on f_{12}, GL-PSO, EPSO converge quickly, and GL-PSOD follows GL-PSO. GGL-PSO and GGL-PSOD converge much more slowly than GL-PSO in the early stage, but GGL-PSO and GGL-PSOD maintain steady convergence rates in the middle stage and achieve almost the same mean error as GL-PSO and EPSO. PSO-cf, FIPS and CLPSO fall behind. Fig. 2(d) shows that, on f_{16}, SL-PSO converges quickly and achieves the lowest mean error. Although GGL-PSOD converges relatively slowly, it surpasses most of the other PSO algorithms and achieves the second-lowest mean error. The mean performance order of the four GL-PSO-based algorithms on f_{16} is GGL-PSOD, GGL-PSO, GL-PSOD and GL-PSO. CLPSO and HCLPSO also perform well. PSO-cf, FIPS and GL-PSO fall behind.

For composition functions, Fig. 2(e) indicates that, on f_{21}, GL-PSO and GL-PSOD converge quickly at the beginning, but they experience stagnation very early. Although SL-PSO, GGL-PSO and GGL-PSOD converge relatively slowly at the start, they can maintain a steady convergence rate in the middle stage. SL-PSO and GGL-PSOD yield the lowest and the second-lowest mean errors, respectively. GGL-PSO generates a slightly higher mean error than GL-PSO in the end. FIPS and PSO-cf fall behind. On f_{27}, the convergence curves of CLPSO, SL-PSO, HCLPSO, EPSO, GL-PSO, GL-PSOD, GGL-PSO and GGL-PSOD are almost overlapped. The data in Tables 3 and 5 show that GGL-PSOD yields a slightly lower mean error than the other involved PSO algorithms. PSO-cf and FIPS fall behind.

In general, GL-PSO and GL-PSOD converge quickly, and GL-PSOD performs better than GL-PSO because its exploration is improved due to its adoption of a ring topology for exemplar generation. After combining GLC with GL-PSO and GL-PSOD, the resultant GGL-PSO and GGL-PSOD converge relatively slowly in the early stage. Both GGL-PSO and GGL-PSOD can maintain steady convergence rates in the middle

Table 7
Success rate of PSO variants on 30-*D* CEC2017 functions.

Func.	ε	PSO-cf	FIPS	CLPSO	SL-PSO	HCLPSO	EPSO	GL-PSO	GL-PSOD	GGL-PSO	GGL-PSOD
f_1	50	5.88%	0.00%	**84.31%**	11.76%	52.94%	52.94%	9.80%	3.92%	9.80%	3.92%
f_2	50	0.00%	0.00%	0.00%	37.25%	58.82%	58.82%	49.02%	**80.39%**	15.69%	**80.39%**
f_3	50	**100.00%**	0.00%	0.00%	0.00%	**100.00%**	**100.00%**	86.27%	92.16%	**100.00%**	**100.00%**
f_4	50	3.92%	0.00%	9.80%	0.00%	11.76%	11.76%	**54.90%**	1.96%	1.96%	1.96%
f_5	50	1.96%	0.00%	92.16%	**100.00%**	86.27%	86.27%	54.90%	80.39%	60.78%	80.39%
f_6	50	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_7	50	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	**3.92%**	0.00%	**3.92%**
f_8	50	0.00%	0.00%	54.90%	**100.00%**	80.39%	80.39%	50.98%	64.71%	47.06%	66.67%
f_9	50	0.00%	**100.00%**	29.41%	**100.00%**	92.16%	92.16%	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_{10}	50	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
f_{11}	50	3.92%	13.73%	66.67%	**70.59%**	47.06%	47.06%	35.29%	41.18%	27.45%	33.33%
f_{12}	50	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
f_{13}	50	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	**3.92%**	0.00%	1.96%	1.96%
f_{14}	50	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
f_{15}	50	0.00%	0.00%	3.92%	5.88%	0.00%	0.00%	5.88%	**17.65%**	1.96%	3.92%
F_{16}	50	0.00%	0.00%	0.00%	**25.49%**	0.00%	0.00%	0.00%	0.00%	5.88%	5.88%
f_{17}	50	0.00%	0.00%	1.96%	**66.67%**	13.73%	13.73%	0.00%	17.65%	19.61%	31.37%
f_{18}	50	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
f_{19}	50	0.00%	0.00%	**45.10%**	1.96%	31.37%	31.37%	1.96%	0.00%	0.00%	7.84%
f_{20}	50	0.00%	0.00%	11.76%	37.25%	17.65%	17.65%	0.00%	9.80%	25.49%	**39.22%**
f_{21}	500	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_{22}	500	35.29%	94.12%	**100.00%**	90.20%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_{23}	500	96.08%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_{24}	500	54.90%	0.00%	92.16%	**100.00%**	92.16%	92.16%	94.12%	**100.00%**	98.04%	**100.00%**
f_{25}	500	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_{26}	500	0.00%	1.96%	**94.12%**	0.00%	**94.12%**	**94.12%**	31.37%	45.10%	35.29%	47.06%
f_{27}	500	0.00%	0.00%	7.84%	0.00%	5.88%	5.88%	1.96%	3.92%	**25.49%**	1.96%
f_{28}	500	80.39%	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
f_{29}	500	0.00%	0.00%	41.18%	60.78%	**66.67%**	**66.67%**	25.49%	43.14%	47.06%	45.10%
f_{30}	500	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Best		4	6	9	**12**	9	9	9	11	9	**12**

Note: "*Best*" denotes the number of functions for which the algorithm achieved the highest success rate compared to the other PSO variants.

Table 8
Mean function evaluations of PSO variants on 30-*D* CEC2017 functions.

Func.	ε	PSO-cf	FIPS	CLPSO	SL-PSO	HCLPSO	EPSO	GL-PSO	GL-PSOD	GGL-PSO	GGL-PSOD
f_1	50	**1.62E+04**	Inf	2.44E+05	2.05E+04	2.30E+05	2.30E+05	3.24E+04	6.02E+04	1.30E+05	6.02E+04
f_2	50	Inf	Inf	Inf	**6.48E+04**	2.10E+05	2.10E+05	8.65E+04	1.87E+05	1.94E+05	1.89E+05
f_3	50	**1.42E+05**	Inf	Inf	Inf	1.78E+05	1.78E+05	2.04E+05	2.30E+05	1.80E+05	2.30E+05
f_4	50	**1.13E+05**	Inf	2.24E+05	Inf	2.36E+05	2.36E+05	1.83E+05	1.77E+05	2.47E+05	1.77E+05
f_5	50	**9.94E+03**	Inf	2.33E+05	1.31E+05	1.34E+05	1.34E+05	2.48E+04	2.85E+04	1.05E+05	3.04E+04
f_6	50	1.27E+03	2.16E+03	1.22E+04	**1.23E+03**	4.06E+03	4.06E+03	1.63E+03	2.42E+03	1.92E+03	2.41E+03
f_7	50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	**4.39E+04**	Inf	**4.39E+04**
f_8	50	Inf	Inf	2.58E+05	1.08E+05	1.30E+05	1.30E+05	**1.47E+04**	2.69E+04	1.10E+05	2.91E+04
f_9	50	Inf	3.74E+04	2.56E+05	7.17E+03	8.32E+04	8.32E+04	**1.16E+04**	1.65E+04	9.91E+04	1.82E+04
f_{10}	50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
f_{11}	50	**2.02E+04**	2.09E+05	2.21E+05	6.13E+04	1.51E+05	1.51E+05	2.69E+04	5.58E+04	1.09E+05	1.26E+05
f_{12}	50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
f_{13}	50	Inf	Inf	Inf	Inf	Inf	Inf	**3.56E+04**	Inf	1.26E+05	1.30E+05
f_{14}	50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
f_{15}	50	Inf	Inf	2.51E+05	**4.01E+04**	Inf	Inf	8.69E+04	4.84E+04	1.66E+05	1.28E+05
F_{16}	50	Inf	Inf	Inf	**1.85E+04**	Inf	Inf	Inf	Inf	8.17E+04	9.06E+04
f_{17}	50	Inf	Inf	2.79E+05	6.28E+04	1.07E+05	1.07E+05	Inf	**3.51E+04**	1.02E+05	1.09E+05
f_{18}	50	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
f_{19}	50	Inf	Inf	2.21E+05	2.76E+04	2.23E+05	2.23E+05	**1.81E+04**	Inf	Inf	1.06E+05
f_{20}	50	Inf	Inf	1.91E+05	**1.91E+04**	9.19E+04	9.19E+04	Inf	2.90E+04	8.72E+04	9.28E+04
f_{21}	500	1.12E+03	1.66E+03	6.45E+03	**1.08E+03**	2.03E+03	2.03E+03	1.31E+03	1.78E+03	1.42E+03	1.76E+03
f_{22}	500	**2.02E+03**	1.24E+04	3.27E+04	4.37E+03	2.54E+04	2.54E+04	3.62E+03	5.54E+03	4.99E+04	6.55E+03
f_{23}	500	**7.53E+03**	1.19E+05	4.06E+04	1.82E+04	2.54E+04	2.54E+04	7.56E+03	1.37E+04	8.70E+04	1.50E+04
f_{24}	500	2.38E+04	Inf	1.64E+05	3.35E+04	4.10E+04	4.10E+04	**9.60E+03**	1.84E+04	9.32E+04	1.98E+04
f_{25}	500	5.05E+03	1.30E+04	5.03E+04	**3.40E+03**	3.90E+04	3.90E+04	5.67E+03	9.03E+03	4.98E+04	1.00E+04
f_{26}	500	Inf	2.14E+05	1.42E+05	Inf	1.03E+05	1.03E+05	**9.53E+03**	1.71E+04	8.08E+04	2.02E+04
f_{27}	500	Inf	Inf	1.91E+05	Inf	1.25E+05	1.25E+05	2.75E+05	2.77E+04	9.01E+04	**2.49E+04**
f_{28}	500	2.84E+04	3.67E+04	6.30E+04	**7.35E+03**	4.14E+04	4.14E+04	9.46E+03	1.84E+04	7.30E+04	1.97E+04
f_{29}	500	Inf	Inf	2.00E+05	**2.44E+04**	1.12E+05	1.12E+05	2.60E+04	4.62E+04	1.04E+05	4.85E+04
f_{30}	500	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Best		7	0	0	**9**	0	0	6	2	0	2

Note: *"Best"* denotes the number of functions for which the algorithm achieved the lowest mean function evaluations compared to the other PSO variants.
"Inf" indicates that the algorithm failed to achieve a near-optimal solution within the maximum allowable function evaluations.

Table 9
The comparison test results with meta-heuristics on 30-*D* CEC2017 functions.

Func.	ABC	CMA-ES	CS	ANS	GWO	GGL-PSOD
f_1	<2.41E+02	**<2.84E-14**	<6.31E-07	<3.36E+01	>2.41E+09	4.90E+03
f_2	>5.91E+10	**<0.00E+00**	>6.64E+05	>2.81E+06	>8.01E+28	3.64E+03
f_3	>1.10E+05	**<1.05E-13**	>4.45E+03	>1.51E+04	>3.38E+04	3.95E-09
f_4	<2.24E+01	**<1.47E+01**	<3.57E+01	<3.57E+01	>2.02E+02	6.09E+01
f_5	>8.31E+01	>3.06E+02	>1.39E+02	>4.95E+01	>9.68E+01	**2.69E+01**
f_6	**<3.82E-13**	>6.35E+01	>2.94E+01	>3.73E-07	>8.57E+00	1.27E-07
f_7	>1.04E+02	>7.50E+02	>2.00E+02	>8.23E+01	>1.68E+02	**6.31E+01**
f_8	>9.36E+01	>2.25E+02	>1.25E+02	>5.47E+01	>8.51E+01	**3.04E+01**
f_9	>1.17E+03	>4.70E+03	>3.56E+03	>1.22E+02	>7.79E+02	**1.42E-01**
f_{10}	>2.57E+03	>4.57E+03	>3.61E+03	>2.29E+03	>3.01E+03	**2.07E+03**
f_{11}	>7.11E+02	<1.59E+02	<7.65E+01	**<2.18E+01**	>7.47E+02	3.08E+01
f_{12}	>1.01E+06	**<1.72E+03**	>2.90E+04	>3.63E+05	>5.36E+07	2.59E+04
f_{13}	>1.94E+04	<2.15E+03	**<1.57E+02**	<5.01E+02	>2.09E+06	1.14E+04
f_{14}	>1.38E+05	<1.70E+02	**<6.75E+01**	>4.69E+04	>3.55E+05	2.20E+03
f_{15}	>4.09E+03	<2.06E+02	**<5.65E+01**	<1.35E+02	>5.97E+05	2.14E+03
F_{16}	>6.13E+02	>1.35E+03	>8.74E+02	>6.25E+02	>8.16E+02	**3.71E+02**
f_{17}	>1.91E+02	>3.49E+02	>2.54E+02	>1.72E+02	>3.13E+02	**6.13E+01**
f_{18}	>2.86E+05	**<1.72E+02**	<3.15E+02	>1.23E+05	>1.39E+06	9.04E+04
f_{19}	>1.53E+04	<1.64E+02	**<2.74E+01**	<1.49E+02	>6.06E+05	4.72E+03
f_{20}	>2.10E+02	>1.37E+03	>3.14E+02	>2.16E+02	>3.32E+02	**1.07E+02**
f_{21}	**<2.11E+02**	>3.98E+02	>3.21E+02	>2.51E+02	>2.90E+02	2.31E+02
f_{22}	>1.14E+02	>5.83E+03	>1.92E+03	>6.24E+02	>2.21E+03	**1.00E+02**
f_{23}	>4.20E+02	>2.31E+03	>4.91E+02	>4.01E+02	>4.52E+02	**3.84E+02**
f_{24}	**<3.81E+02**	>5.52E+02	>5.59E+02	>5.23E+02	>5.20E+02	4.58E+02
f_{25}	**=3.84E+02**	=3.88E+02	=3.86E+02	=3.85E+02	>4.75E+02	3.87E+02
f_{26}	**<2.97E+02**	>2.27E+03	>1.61E+03	>8.07E+02	>2.18E+03	7.67E+02
f_{27}	>5.14E+02	**=5.00E+02**	>5.18E+02	>5.14E+02	>5.55E+02	5.02E+02
f_{28}	>3.99E+02	<3.51E+02	**<3.28E+02**	>3.98E+02	>6.20E+02	3.59E+02
f_{29}	>6.23E+02	>7.84E+02	>8.71E+02	>5.74E+02	>8.48E+02	**4.50E+02**
f_{30}	>1.55E+04	>5.31E+03	>4.20E+03	>4.72E+03	>7.08E+06	**3.99E+03**
w/t/l	23/1/6	16/2/12	20/1/9	23/1/6	30/0/0	
Best	5	7	5	1	0	12

Table 10
The Friedman test results with meta-heuristics on 30-*D* CEC2017 functions.

	ABC	CMA-ES	CS	ANS	GWO	GGL-PSOD
Friedman rank	3.400	4.00	3.567	2.833	5.033	**2.1667**
General rank	3	5	4	2	6	**1**

stage. GGL-PSOD yields high accuracy in the end, while GGL-PSO shows no obvious advantage compared to GL-PSO. With both the ring topology and GLC, GGL-PSOD exhibits the highest performance and strong robustness in solving different types of functions.

5. Conclusion

In this study, the ring topology and global learning component (GLC) with linearly adjusted control parameters are combined with GL-PSO to enhance its diversity, exploration and adaptability. The Wilcoxon signed-rank test and Friedman test results on the CEC2017 test suite indicate that adopting a ring topology instead of a global topology for exemplar generation can enhance the diversity and exploration of GL-PSO and GGL-PSO. The resultant GL-PSOD and GGL-PSOD perform better than the corresponding GL-PSO and GGL-PSO. With GLC, GGL-PSO and GGL-PSOD can preserve much more diversity than the corresponding GL-PSO and GGL-PSO in the early stage of optimization. However, GGL-PSOD can achieve high accuracy in the latter stage, while GGL-PSO does not show a significantly advantage over GL-PSO. With both the ring topology and GLC, GGL-PSOD exhibits high performance and strong robustness on different types of CEC2017 functions. It outperforms seven representative PSO variants and five meta-heuristics, including GL-PSO, SL-PSO, HCLPSO, EPSO, ABC, CMA-ES and CS. The convergence curves on six representative functions reveal that GGL-PSOD converges relatively slowly in the early stage, but it can maintain a steady convergence speed and achieve high accuracy in the end.

GGL-PSOD slows down the convergence rate to improve its adaptability to different types of functions. It adopts open-loop optimization similar to most of the existing PSO variants, and no feedback of the evolution state is employed to alter the structure or adjust the parameters of the algorithm. To further improve the performance of PSO, the dynamical evolution state detection, variable structure, and adaptive parameter regulation deserve further investigation. With feedback of the evolution state and a humanlike decision mechanism, more intelligent PSO variants can be developed.

Compliance with ethical standards

Conflict of interest The author declares that there is no conflict of interest.

Ethical approval The work of this article does not involve use of human participants or animals.

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments, which were helpful for improving the quality of the manuscript. The authors thank Prof. Suganthan (http://www3.ntu.edu.sg/home/epnsugan/) and Prof. Jin Yaochu (http://www.soft-computing.de/jin-pub_year.html) for code on their homepages. The authors thank the reference authors for providing their code for free. This work is supported by the Independent Research Project of State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body Grant 71765003, the National Natural Science Foundation of China under Grant 61573135, the Hunan Province Graduate Student Scientific Research Innovation Fund Grant CX2017B110, the Hunan Key Laboratory of Intelligent Robot Technology in Electronic Manufacturing Open Foundation Grant 2017TP1011, the Key Research and Development Project of Science and Technology Plan of Hunan Province Grant 2018GK2021, and the Science and Technology Plan Project of Shenzhen City Grant JCYJ20170306141557198. Please contact Dr. Anping Lin

(anping719@126.com) for source code.

References

[1] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[2] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Presented at the Proc. IEEE Int. Conf. Neural Netw., 1995.
[3] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, Math. Probl Eng. 2015 (2015) 1–38.
[4] M.R. Bonyadi, Z. Michalewicz, Particle swarm optimization for single objective continuous space problems: a review, Evol. Comput. 25 (Spr 2017) 1–54.
[5] A.A.A. Esmin, R.A. Coelho, S. Matwin, A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data, Artif. Intell. Rev. 44 (Jun 2015) 23–45.
[6] A.M. Ramanath, Computational intelligence: an introduction, Jasss J. Artif. Soc. Social Simul. 7 (Jan 2004).
[7] M.R. AlRashidi, M.E. El-Hawary, A survey of particle swarm optimization applications in electric power systems, IEEE Trans. Evol. Comput. 13 (Aug 2009) 913–918.
[8] B. Zhao, C.X. Guo, Y.J. Cao, A multiagent-based particle swarm optimization approach for optimal reactive power dispatch, IEEE Trans. Power Syst. 20 (2005) 1070–1078.
[9] M.S. Hasanoglu, M. Dolen, Feasibility enhanced particle swarm optimization for constrained mechanical design problems, Proc. IME C J. Mech. Eng. Sci. 232 (Jan 2018) 381–400.
[10] A. Moharam, M.A. El-Hosseini, H.A. Ali, Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers, Appl. Soft Comput. 38 (Jan 2016) 727–737.
[11] K.K. Soo, Y.M. Siu, W.S. Chan, L. Yang, R.S. Chen, Particle-swarm-optimization-based multiuser detector for CDMA communications, IEEE Trans. Veh. Technol. 56 (2007) 3006–3013.
[12] Y.L. Li, W. Shao, L. You, B.Z. Wang, An improved PSO algorithm and its application to UWB antenna design, IEEE Antenn. Wireless Propag. Lett. 12 (2013) 1236–1239.
[13] L. Dhanalakshmi, S. Ranjitha, H.N. Suresh, A novel method for image processing using particle swarm optimization technique, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (Iceeot), 2016, pp. 3357–3363.
[14] A.A. Tofigh, M.O. Shabani, Efficient optimum solution for high strength Al alloys matrix composites, Ceram. Int. 39 (2013) 7483–7490.
[15] A. Mazahery, M.O. Shabani, Extruded AA6061 alloy matrix composites: the performance of multi-strategies to extend the searching area of the optimization algorithm, J. Compos. Mater. 48 (2014) 1927–1937.
[16] M.O. Shabani, M.R. Rahimipour, A.A. Tofigh, P. Davami, Refined microstructure of compo cast nanocomposites: the performance of combined neuro-computing, fuzzy logic and particle swarm techniques, Neural Comput. Appl. 26 (May 2015) 899–909.
[17] A.A. Tofigh, M.R. Rahimipour, M.O. Shabani, P. Davami, Application of the combined neuro-computing, fuzzy logic and swarm intelligence for optimization of compocast nanocomposites, J. Compos. Mater. 49 (2015).
[18] V.N. Lal, N. Singh, Modified particle swarm optimisation-based maximum power point tracking controller for single-stage utility-scale photovoltaic system with reactive power injection capability, IET Renew. Power Gener. 10 (Jul 2016) 899–907.
[19] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, B. Fielding, A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition, IEEE Trans. Cybern. PP (2017) 1–14.
[20] N.B. Guedria, Improved accelerated PSO algorithm for mechanical engineering optimization problems, Appl. Soft Comput. 40 (2016) 455–467.
[21] W. Elsayed, Y. Hegazy, M. El-Bages, F. Bendary, Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem, IEEE Trans. Ind. Inform. PP (2017), pp. 1–1.
[22] M. Shamsipour, Z. Pahlevani, M.O. Shabani, A. Mazahery, Optimization of the EMS process parameters in compocasting of high-wear-resistant Al-nano-TiC composites, Appl. Phys. A 122 (2016) 457.
[23] M. Shamsipour, Z. Pahlevani, M.O. Shabani, A. Mazahery, Squeeze casting of electromagnetically stirred aluminum matrix nanocomposites in semi-solid condition using hybrid algorithm optimized parameters, Kovove Mater.-Metal. Mater. 55 (2017) 33–43.
[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, Inf. Sci. 326 (2016) 1–24.
[25] M. Hu, T. Wu, J.D. Weir, An adaptive particle swarm optimization with multiple adaptive methods, IEEE Trans. Evol. Comput. 17 (2013) 705–720.
[26] Yuhui Shi, Eberhart, C. Russell, Parameter Selection in Particle Swarm Optimization, Springer Berlin Heidelberg, 1998.
[27] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 1, 1999, pp. 320–324.
[28] M. Clerc, J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (2002) 58–73.
[29] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (2004) 240–255.
[30] M.S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, G. Pasi, Fuzzy Self-tuning PSO: a Settings-free Algorithm for Global Optimization, Swarm & Evolutionary Computation, 2017.
[31] Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybern. B Cybern. : Publ. IEEE Syst. Man Cybern. Soci. 39 (2009) 1362–1381.
[32] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (2004) 204–210.
[33] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer with local search, in: IEEE Congress on Evolutionary Computation, 2005, pp. 522–528.
[34] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: 2005 Ieee Swarm Intelligence Symposium, 2005, pp. 124–129.
[35] W.H. Lim, N.A.M. Isa, Particle swarm optimization with increasing topology connectivity, Eng. Appl. Artif. Intell. 27 (Jan 2014) 80–102.
[36] Y. Chen, L. Li, H. Peng, J. Xiao, Q.T. Wu, Dynamic multi-swarm differential learning particle swarm optimizer, Swarm Evol. Comput. 39 (2017).
[37] L. Wang, B. Yang, Y.H. Chen, Improving particle swarm optimization using multi-layer searching strategy, Inf. Sci. 274 (Aug 1 2014) 70–94.
[38] X.W. Xia, C.W. Xie, B. Wei, Z.B. Hu, B.J. Wang, C. Jin, Particle swarm optimization using multi-level adaptation and purposeful detection operators, Inf. Sci. 385 (Apr 2017) 174–195.
[39] Q. Liu, W. Wei, H. Yuan, Z.H. Zhan, Y. Li, Topology selection for particle swarm optimization ☆, Inf. Sci. 363 (2016) 154–173.
[40] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Swarm Intelligence Symp., 2003, pp. 174–181.
[41] J.J. Liang, A.K. Qin, S. Member, P.N. Suganthan, S. Member, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (2006) 281–295.
[42] G. Wu, D. Qiu, Y. Yu, W. Pedrycz, M. Ma, H. Li, Superior solution guided particle swarm optimization combined with local search techniques, Expert Syst. Appl. 41 (2014) 7536–7548.
[43] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. 294 (2015) 182–202.
[44] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, Inf. Sci. 326 (Jan 1 2016) 1–24.
[45] M.R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self-regulating particle swarm optimization algorithm, Swarm and Evol. Comput. 28 (Jun 2016) 98–116.
[46] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (2015) 43–60.
[47] H.D. Wang, Y.C. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Trans. Cybern. 47 (Sep 2017) 2664–2677.
[48] H.L. Shieh, C.C. Kuo, C.M. Chiang, Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification, Appl. Math. Comput. 218 (2011) 4365–4383.
[49] J. Li, J.Q. Zhang, C.J. Jiang, M.C. Zhou, Composite particle swarm optimizer with historical memory for function optimization, IEEE Trans. Cybern. 45 (Oct 2015) 2350–2363.
[50] H.B. Ouyang, L.Q. Gao, X.Y. Kong, S. Li, D.X. Zou, Hybrid harmony search particle swarm optimization with global dimension selection, Inf. Sci. 346–347 (2016) 318–337.
[51] X. Chen, H. Tianfield, C.L. Mei, W.L. Du, G.H. Liu, Biogeography-based learning particle swarm optimization, Soft Comput. 21 (Dec 2017) 7519–7541.
[52] I.B. Aydilek, A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, Appl. Soft Comput. 66 (May 2018) 232–249.
[53] Y.G. Chen, L.X. Li, H.P. Peng, J.H. Xiao, Y.X. Yang, Y.H. Shi, Particle swarm optimizer with two differential mutation, Appl. Soft Comput. 61 (Dec 2017) 314–330.
[54] A. Bouyer, A. Hatamlou, An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms, Appl. Soft Comput. 67 (2018) 172–182.
[55] D. Tian, Z. Shi, MPSO: modified particle swarm optimization and its applications, Swarm Evol. Comput. 41 (2018).
[56] K. Chen, F.Y. Zhou, L. Yin, S.Q. Wang, Y.G. Wang, F. Wan, A hybrid particle swarm optimizer with sine cosine acceleration coefficients, Inf. Sci. 422 (Jan 2018) 218–241.
[57] J.H. Holland, Erratum: genetic algorithms and the optimal allocation of trials, SIAM J. Comput. 2 (1973) 88–105.
[58] C.F. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, IEEE Trans. Syst. Man Cybern. Part B Cybern. Publ. IEEE Systems Man Cybern. Soci. 34 (2004) 997–1006.
[59] R.J. Kuo, Y.J. Syu, Z.Y. Chen, F.C. Tien, Integration of particle swarm optimization and genetic algorithm for dynamic clustering, Inf. Sci. 195 (2012) 124–140.
[60] P. Ghamisi, J.A. Benediktsson, Feature selection based on hybridization of genetic algorithm and particle swarm optimization, Geosci. Rem. Sens. Lett. IEEE 12 (2014) 309–313.
[61] C. Tan, S. Chang, L. Liu, Hierarchical genetic-particle swarm optimization for bistable permanent magnet actuator, Appl. Soft Comput. 61 (2017).
[62] X. Liu, H. An, L. Wang, X. Jia, An integrated approach to optimize moving average rules in the EUA futures market based on particle swarm optimization and genetic algorithms, Appl. Energy 185 (2017) 1778–1787.
[63] A. Gholami, H. Bonakdari, I. Ebtehaj, M. Mohammadian, B. Gharabaghi, S.R. Khodashenas, Uncertainty analysis of intelligent model of hybrid genetic

algorithm and particle swarm optimization with ANFIS to predict threshold bank profile shape based on digital laser approach sensing, Measurement 121 (2018).
[64] Y.J. Gong, J.J. Li, Y.C. Zhou, Y. Li, H.S.H. Chung, Y.H. Shi, et al., Genetic learning particle swarm optimization, IEEE Trans. Cybern. 46 (Oct 2016) 2277–2290.
[65] W. Sun, A. Lin, H. Yu, Q. Liang, G. Wu, All-dimension neighborhood based particle swarm optimization with randomly selected neighbors, Inf. Sci. 405 (2017) 141–156.
[66] S. Jiang, Z. Ji, Y. Shen, A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints, Int. J. Electr. Power Energy Syst. 55 (2014) 628–644.
[67] M.Z.A.N.H. Awad, 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: Available: http://www.ntu.edu.sg/home/epnsugan/.
[68] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm and Evol. Comput. 24 (2015) 11–24.
[69] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017) 533–548, 6//.
[70] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (Nov 2007) 459–471.
[71] N. Hansen, S.D. Muller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evol. Comput. 11 (Spr 2003) 1–18.
[72] G. Wu, Across neighbourhood search for numerical optimization, Inf. Sci. 329 (2016) 597–618.
[73] A.H. Gandomi, X.S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (2013) 17–35.
[74] S. Deng, B. Jing, H. Zhou, Heuristic particle swarm optimization approach for test point selection with imperfect test, J. Intell. Manuf. (2014) 1–14.
[75] M. Hollander, D.A. Wolfe, Nonparametric Statistical Methods, second ed., vol. 2, John Wiley &SonsInc, 1999.
[76] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evol. Comput. 1 (2011) 3–18.
[77] J.J. Liang, B.Y. Qu, P.N. Suganthan, A.G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-parameter Optimization, Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, 2013. Tech. Rep. 2013.
[78] J.D. Gibbons, S. Chakraborti, Nonparametric Statistical Inference, fifth ed., Chapman & Hall, 2010.
[79] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan 2006) 1–30.
[80] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (Apr 2009) 398–417.
[81] G.H. Wu, X. Shen, H.F. Li, H.K. Chen, A.P. Lin, P.N. Suganthan, Ensemble of differential evolution variants, Inf. Sci. 423 (Jan 2018) 172–186.
[82] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.

Anping Lin received the B. S, M.S. 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 towards the Ph. D in Hunan University, Changsha, China. His main research interests include evolutionary computation, multimodal optimization and computation intelligence.

Wei Sun received his B.S, M.S, and Ph.D. degrees from the Department of Automation Engineering, Hunan University, P. R. China, in 1997, 1999 and 2003, respectively. He now is working as a Professor at the College of Electrical and Information Engineering, Hunan University. His areas of interests are computer vision and robotics, neural networks, intelligent control.

Hongshan Yu is an associate Professor, College of Electrical and InformationEngineering, Hunan University. He received his B.S., M.S., and Ph.D. degreesfrom Hunan University in 2001, 2004 and 2007. His research interests includemobile robots navigation and computer vision

Guohua Wu received the B.S. degree in Information Systems and Ph.D degree in Operations Research from National University of Defense Technology, China, in 2008 and 2014, respectively. During 2012 and 2014, he was a visiting Ph.D student at University of Alberta, Edmonton, Canada, supervised by Prof. WitoldPedrycz. He is now a Lecturer at the College of Information Systems and Management, National University of Defense Technology, Changsha, China. His current research interests include planning and scheduling, evolutionary computation and machine learning. He has authored more than 20 referred papers including those published in IEEE Transactions on System Man Cybernetics: System, Information Sciences, Computers & Operations Research, Applied Soft Computing. He serves as an Associate Editor of Swarm and Evolutionary Computation Journal and an editorial board member of International Journal of Bio-Inspired Computation. He is a regular reviewer of more than 20 journals including IEEE TEVC, IEEE TCYB, Information Sciences and Applied Soft Computing.

Hongwei Tang was born in 1982, he is a Ph.D.candidate in the College of Electrical and Information Engineering, Hunan University. His research interests include intelligent control, robot vision SLAM and path planning and intelligent optimization algorithms.