Swarm and Evolutionary Computation 45 (2019) 33–51

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Particle swarm optimization based on dimensional learning strategy

Guiping Xu [a], Quanlong Cui [a], Xiaohu Shi [a,b], Hongwei Ge [c], Zhi-Hui Zhan [d], Heow Pueh Lee [e], Yanchun Liang [a,b], Ran Tai [a], Chunguo Wu [a,b,*]

[a] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun, 130012, PR China

[b] Zhuhai Laboratory of Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, School of Computer, Zhuhai College of Jilin University, Zhuhai, 519041, PR China

[c] College of Computer Science and Technology, Dalian University of Technology, Dalian, 116024, PR China

[d] Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, PR China

[e] Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, Singapore

ARTICLE INFO

Index Terms:
Dimensional learning
Comprehensive learning
Particle swarm optimization (PSO)

ABSTRACT

In traditional particle swarm optimization (PSO) algorithm, each particle updates its velocity and position with a learning mechanism based on its personal best experience and the population best experience. The learning mechanism in traditional PSO is simple and easy to implement, but it suffers some potential problems, such as the phenomena of "oscillation" and "two steps forward, one step back". Therefore, designing an effective learning strategy to avoid these two phenomena and to improve the search efficiency is an urgent issue for PSO research. This paper proposes a dimensional learning strategy (DLS) for discovering and integrating the promising information of the population best solution according to the personal best experience of each particle. Thereafter, a two-swarm learning PSO (TSLPSO) algorithm based on different learning strategies is proposed. One of the subpopulations constructs the learning exemplars by DLS to guide the local search of the particles, and the other subpopulation constructs the learning exemplars by the comprehensive learning strategy to guide the global search. 16 classic benchmark functions, 30 CEC2014 test functions, and 1 real-world optimization problem are used to test the proposed algorithm against with 5 typical PSO algorithms and 1 state-of-the-art differential evolution (DE) algorithm. The experimental results show that TSLPSO is statistically and significantly better than the compared algorithms for most of the test problems. Moreover, the convergence speed and convergence accuracy of TSLPSO are also significantly improved.

1. Introduction

In recent years, many complex optimization problems that are difficult to solve by traditional optimization algorithms, such as nonlinear process control, structural design, and text mining, have emerged in engineering applications [1]. Swarm intelligence optimization algorithms have gradually become the preferred methods for solving such complex problems. Swarm intelligence optimization belongs to random optimization algorithms, including the classical particle swarm optimization algorithm (PSO), ant colony optimization algorithm (ACO) [2,3], culture algorithm (CA), artificial fish swarm algorithm (AFSA), and free search algorithm (FS), which simulate biological behaviors in nature. Among these algorithms, PSO has become the focus of research in recent years because of its advantages, such as simple principle, few parameters, and fast convergence speed [4,5].

PSO is a random search algorithm that was proposed by Kennedy and Eberhart [6] in 1995 by simulating the foraging and flocking behavior of birds in nature. PSO attracted the attention of researchers when it was first proposed, and various theories and applications of PSO had continuously emerged [7–9], which had greatly promoted the study of PSO. Research on PSO mainly focuses on the following four aspects: parameter setting, selection of neighborhood topology, improvement of learning strategy, hybridization of PSO with other algorithms [1]. The following briefly introduces these four aspects.

* Corresponding author. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun, 130012, PR China.

E-mail address: wucg@jlu.edu.cn (C. Wu).

https://doi.org/10.1016/j.swevo.2018.12.009
Received 6 June 2018; Received in revised form 24 December 2018; Accepted 27 December 2018
Available online 4 January 2019

To balance exploration and exploitation, Shi and Eberhart [10] first proposed the inertia weight, and Clerc and Kennedy [11] proposed the contraction coefficient. Both methods could balance exploration and exploitation abilities by controlling the convergence tendency of the algorithm. Self-organizing hierarchical particle swarm optimization (HPSO-TVAC) [12] adopted time-varying acceleration coefficients. Zhan et al. [13] proposed an adaptive particle swarm optimization algorithm (APSO) that would divide the state of the population into exploration, exploitation, convergence and jumping out through evolutionary state estimation; the inertia weights and acceleration coefficients were then adaptively adjusted according to these four states.

PSOs based on neighborhood topology was used to control exploration and exploitation according to different information-sharing mechanisms among the particles [14–16]. Mendes and Kennedy [17] proposed a new information flow mechanism to update the position of each particle called fully informed particle swarm optimization (FIPS). FIPS utilized the weighted average of the personal best positions of all neighbors to update the position of the specified particle. Parsopoulos and Vrahatis [18] proposed a unified particle swarm optimization (UPSO), which used the best experience from local neighbors and global neighbors to control exploration and exploitation. Nasir et al. [19] proposed a dynamic neighbor learning particle swarm optimization (DNLPSO), in which an exemplar particle was selected from the best position of the neighbors, including itself. Therefore, the velocity of a particle might be affected by the neighbors' or its own historical experience.

In addition to the adoption of different neighborhood topologies to control exploration and exploitation, research on learning strategies has attracted considerable attention. Liang et al. [20] proposed a comprehensive learning particle swarm optimization (CLPSO) to solve complex multimodal problems. CLPSO used the personal best experience of all other particles to update the specified particle velocity, which enhanced the diversity of the population and improves the global search ability. Some scholars had proposed different improvement strategies based on CLPSO [21–27] to balance exploration and exploitation. Among them, the HCLPSO algorithm [26] proposed heterogeneous subpopulations based on CLPSO. HCLPSO divided the swarm into one exploration subpopulation and one exploitation subpopulation, and both subpopulations adopted a comprehensive learning strategy. Li et al. [28] proposed self-learning particle swarm optimization (SLPSO). SLPSO adopted four different learning strategies based on an adaptive learning mechanism, i.e., learning from (1) its own *pbest*, (2) the *pbest* of a random particle, (3) the *gbest* of the whole population, and (4) the position generated randomly. Each strategy corresponds to a different operation, allowing each particle to independently address different situations. Wang et al. [29] proposed an enhanced particle swarm optimization using the generalized opposition-based learning (GOPSO). GOPSO used generalized opposition-based learning and Cauchy mutation to prevent the population from being trapped in local optima. Lim et al. [30] proposed the adaptive two layer particle swarm optimization algorithm (ATL-PSO-ELS), which adopted an elitist learning strategy and orthogonal learning strategy to help the algorithm escape local optima. Ouyang et al. [31] proposed the improved global-best-guided particle swarm optimization algorithm (IGPSO), which divided the population into the current population, historical best population and global best population. Each population was assigned a different search strategy. Zhan et al. [32] proposed the orthogonal learning particle swarm optimization (OLPSO), in which an orthogonal learning strategy would use the orthogonal experiment design (OED) to combine the personal best experience and population best experience to construct a learning exemplar for each particle. The learning exemplars could effectively guide the particle search and improve the search efficiency.

The use of the genetic algorithm (GA), artificial bee colony algorithm (ABC) and other different evolutionary algorithms to improve the performance of PSO is another focus of researchers. Gong et al. [33] proposed the genetic learning particle swarm optimization (GL-PSO), which introduced selection, mutation and crossover into the PSO. Each particle would produce a promising exemplar according to the genetic operation and then learn from the exemplar in the same way as that in the traditional PSO. PSO is easy to fall into local optima because of the low efficiency of the global search, whereas ABC suffers from converging slowly due to the lack of local exploitation ability. Therefore, some researchers combined ABC with PSO [34–37]. Among them, Li et al. [37] proposed the hybrid algorithm PS-ABC, which adopted a different search strategy based on the activity of each particle. This hybrid algorithm not only maintains good diversity, but also accelerates the convergence speed, effectively balancing the exploration and exploitation.

Classical PSO algorithms have been shown to perform well on low dimensional problems, however often poorly on high dimensional cases. There are also some researchers who have proposed divide-and-conquer strategies to tackle large-scale optimization problems. An early work on a cooperative co-evolutionary algorithm (CCEA) by Potter and Jong [38] provides a promising approach for decomposing high-dimensional problems and tackling its sub-problems individually. By cooperatively coevolving multiple EA subpopulations, we can obtain an overall solution derived from combinations of sub-solutions evolved from individual subpopulations. An early attempt to apply Potter's CC model to PSO was made by Van den Bergh and Engelbrecht [39], where two cooperative PSO models, CPSO-S_k and CPSO-H_k, were developed. However, these two models were only tested on functions of up to 30 dimensions. Based on CPSO-S_k, an improve cooperative coevolving PSO (CCPSO) [40] algorithm integrating the random grouping and adaptive weighting schemes was developed, and demonstrated great promise in scaling up PSO on high-dimensional and non-separable problems. The CCPSO outperformed the previously proposed CPSO-H_k [39] on both separable and non-separable 30D functions. CCPSO was also shown to perform reasonably well on functions of up to 1000 dimensions. An improved CCPSO, i.e., CCPSO2 [41] adopts a new PSO position update rule that relies on Cauchy and Gaussian distributions to sample new points in the search space, and a scheme to dynamically determine the coevolving subcomponent sizes of the variables. CCPSO2 is more reliable and robust to use—a user does not have to specify the subcomponent size, since it is adaptively chosen from a set. The improved CCPSO2 performs well on complex multimodal functions of up to 2000 dimensions.

According to the various improvement strategies mentioned above, the main common aim of the PSO variants is to maintain a balance between exploration and exploitation and to improve the convergence speed and accuracy while maintaining population diversity. However, the unimodal and multimodal problems have different requests to population diversity. Unimodal problems require small population diversity, which allows the algorithm to converge quickly along the optimal gradient. In contrast, multimodal problems require great population diversity to help the algorithm jump out of local optima. It is difficult for a single improvement strategy to achieve high convergence accuracy for both unimodal and multimodal problems. The existing literature have shown that adopting multiple improvement strategies can balance the exploration and exploitation of the algorithm to achieve high convergence accuracy for both unimodal and multimodal problems. HCLPSO, ATLPSO-ELS, and IGPSO propose different learning strategies for exploration and exploitation to allow the algorithm effectively solving unimodal and multimodal problems. GL-PSO and PS-ABC incorporate the advantages of other evolutionary algorithms into PSO to effectively overcome the shortcomings of premature convergence in PSO. It is necessary to design different learning strategies according to the search features of the algorithm to effectively avoid premature convergence and help the algorithm converge quickly to the global optimum [42].

To effectively discover and protect the promising information of the population best solution, inspired by CLPSO and OLPSO, we propose the dimensional learning PSO (DLPSO) algorithm. DLPSO adopts a dimensional learning strategy (DLS) to construct a learning exemplar for each particle. The learning exemplar is constructed through each dimension of the particles' personal best solution learning from corresponding dimension of the population best solution. Therefore, the learning exemplar combines the excellent information of the personal best experience and population best experience. In addition, to balance exploration and exploitation, inspired by OLPSO and HCLPSO, this paper proposes a two-swarm learning PSO (TSLPSO) algorithm based on different learning strategies. These two subpopulations use learning exemplars constructed by the DLS and the comprehensive learning strategy (CLS), respectively, to guide the particle search.

The remainder of this paper is organized as follows. The related work is introduced in Section 2. The proposed DLS, DLPSO, and TSLPSO are presented in Section 3. In Section 4, several groups of test problems are used to test the proposed DLPSO, TSLPSO and to compare them with 5 other typical PSO variants and 1 CEC2014 winner algorithm. Finally, we end the paper with conclusions and future work in Section 5.

2. Related work

2.1. PSO

Assume that the population has m particles in the n-dimensional search space and that each particle represents a solution in the search space. Particle i consists of an n-dimensional position vector $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})^T$ and velocity vector $\boldsymbol{v}_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,n})^T$. Each particle updates its velocity and position by learning from its personal best experience and population best experience, and the update formulas are as follows:

$$v_{i,j} = v_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{pbest} - x_{i,j}\right) + c_2 r_{2,j}\left(x_j^{gbest} - x_{i,j}\right) \tag{1}$$

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

where $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, m and n are the population size and the dimension of the search space, respectively. $v_{i,j}$ and $x_{i,j}$ denote the velocity and position of the jth dimension of particle i, respectively. $\boldsymbol{x}_i^{pbest} = (x_{i,1}^{pbest}, x_{i,2}^{pbest}, \ldots, x_{i,n}^{pbest})^T$ denotes the historical best position of particle i; $\boldsymbol{x}^{gbest} = (x_1^{gbest}, x_2^{gbest}, \ldots, x_n^{gbest})^T$ denotes the population historical best position. c_1 and c_2 are the acceleration coefficients, and $r_{1,j}$ and $r_{2,j}$ are two uniformly distributed random numbers independently generated within [0, 1] for the jth dimension. The maximum velocity v_j^{max} is used to control each particle velocity within a reasonable range. If $|v_{i,j}| > v_j^{max}$, then $v_{i,j} = sign(v_{i,j}) * v_j^{max}$. To balance the exploration and exploitation of the population, Shi and Eberhart [10] proposed the inertial weight w to control the velocity, and Eq. (1) is modified as the following Eq. (3):

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{pbest} - x_{i,j}\right) + c_2 r_{2,j}\left(x_j^{gbest} - x_{i,j}\right). \tag{3}$$

In Eq. (3), w is usually set to decrease linearly from 0.9 to 0.4 over time. In the search process, the reduction of the inertia weight ensures strong global exploration ability in the early stage and strong local exploitation ability in the later stage. However, the classical PSO could not deal with the complex multimodal problems effectively. Researchers proposed various exemplar learning strategies to enhance and balance the exploitation and exploration abilities of PSO variants to conquer multimodal problems. The two typical exemplar learning strategies with orthogonal experimental design and comprehensive learning strategy are briefly introduced as follows.

2.2. PSO based on OED

Ho et al. [43] proposed the orthogonal particle swarm optimization (OPSO). OPSO uses an "intelligent move mechanism" (IMM) to generate two temporary positions, $\boldsymbol{h} = (h_1, h_2, \ldots, h_n)^T$ and $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)^T$, for each particle:

$$h_j = x_{i,j} + wv_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{pbest} - x_{i,j}\right) \tag{4}$$

$$r_j = x_{i,j} + wv_{i,j} + c_2 r_{2,j}\left(x_j^{gbest} - x_{i,j}\right). \tag{5}$$

$\boldsymbol{h}$ and $\boldsymbol{r}$ correspond to the self-cognitive and social learning components, respectively. OED is performed on $\boldsymbol{h}$ and $\boldsymbol{r}$ to obtain the next position $\boldsymbol{x}^*$, and then the particle velocity is obtained by calculating the difference between the new position $\boldsymbol{x}^*$ and the current position $\boldsymbol{x}$. The IMM effectively combines the excellent information in $\boldsymbol{h}$ and $\boldsymbol{r}$ to generate the next position for the particles. In the search process, each particle uses an IMM to generate a new position and velocity in each generation.

To discover and protect the particles' personal best solution and population best solution, Zhan et al. [32] proposed the orthogonal learning particle swarm optimization. OLPSO adopts an orthogonal learning strategy (OLS) that uses OED combined with $\boldsymbol{x}_i^{pbest}$ and $\boldsymbol{x}^{gbest}$ to construct a learning exemplar $\boldsymbol{x}_i^{ol}$ for each particle, and $\boldsymbol{x}_i^{ol}$ is then used to substitute both $\boldsymbol{x}_i^{pbest}$ and $\boldsymbol{x}^{gbest}$ to guide the particle search. The particle velocity update formula is modified as follows:

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{ol} - x_{i,j}\right), \tag{6}$$

where $\boldsymbol{x}_i^{ol}$ contains the promising information in $\boldsymbol{x}_i^{pbest}$ and $\boldsymbol{x}^{gbest}$. In contrast to OPSO, to avoid oscillation resulting from the learning exemplar changing its direction too frequently, $\boldsymbol{x}_i^{ol}$ is used successively as the exemplar for a certain number of generations until it cannot lead the particle to a better position. A new learning exemplar is then regenerated. The stagnation generation k_i is set for this purpose. If the personal best position of a particle is not updated, then k_i is increased by 1. When $k_i > K$, OLS is used to reconstruct a new learning exemplar. The learning exemplars constructed by OLS can effectively avoid the "oscillation" phenomenon and improve the search efficiency.

2.3. PSO based on CLS

Liang et al. [20] proposed the CLPSO to solve complex multimodal problems. In CLPSO, each dimension of a particle learns from its personal best position with probability η and learns from the corresponding dimensions of the other particles' personal best positions selected by tournament selection with probability $(1 - \eta)$ independently. The velocity update formula is as follows:

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{cl} - x_{i,j}\right), \tag{7}$$

where x_i^{cl} is constructed by the comprehensive learning strategy (CLS). CLPSO uses the personal best experience of all other particles to update the specified particle velocity. Therefore, each dimension of a particle learns from different particles. This CLS enhances the population diversity and has strong global exploration ability. Therefore, when a particle falls into a local optimum, CLS could help the particle jump out of the local optimum by learning from the other particles and thus avoid premature convergence.

Nandar et al. [26] proposed the heterogeneous comprehensive learning particle swarm optimization. HCLPSO contains two heterogeneous subpopulations, and both subpopulations use the CLS proposed in the literature [20] to update the positions of particles. The particles of subpopulation 1 only learn from the personal best experiences of other particles in the same subpopulation, and the particle velocity in subpopulation 1 is updated using Eq. (7); whereas the particles of subpopulation 2 learn from the personal best experience of all particles in the whole population and the population best experience, and the particle velocity in subpopulation 2 is updated using Eq. (8):

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{cl} - x_{i,j}\right) + c_2 r_{2,j}\left(x_j^{gbest} - x_{i,j}\right). \tag{8}$$

Since the particles of subpopulation 1 only learn from the learning exemplar generated by the CLS, it has stronger exploration ability and is mainly responsible for global exploration. The particles of subpopulation 2 not only learn from the learning exemplar generated by the CLS but also the population best experience. Thus, subpopulation 2 has stronger exploitation ability and is primarily responsible for local exploitation.

3. The proposed method

3.1. DLS

According to Eq. (1), the particles in the traditional PSO learn from their personal best experience and population best experience. This learning strategy can cause the phenomena of "oscillation" and "two steps forward, one step back" [39,44]. When the personal best position $\boldsymbol{x}_i^{pbest}$ and population best position $\boldsymbol{x}^{gbest}$ are located in two opposite directions of the current position $\boldsymbol{x}_i$, after particle i approaches $\boldsymbol{x}^{gbest}$, it will move closer to $\boldsymbol{x}_i^{pbest}$ in the next generation because the difference $\boldsymbol{x}_i^{pbest} - \boldsymbol{x}_i$ is large. A particle can always wander between the personal best position and population best position, which will cause "oscillation" and reduce the search efficiency of the algorithm. In addition, for most existing PSOs, the fitness function evaluation is performed only after all dimensions of the solution vector have been updated. Even if the new solution vector improves the fitness function value, the degeneration of some dimensions is possible, resulting in the phenomenon of "two steps forward, one step back". The existing improved PSO variants use the learning exemplars constructed by different learning strategies to guide the particle search to avoid the "oscillation" phenomenon. However, these methods of constructing learning exemplars have a large amount of randomness. Although the population diversity and global search ability are improved, the convergence speed of the algorithm is slowed, and the phenomenon of "two steps forward, one step back" cannot be avoided. In addition, there is no guarantee that the learning exemplars will not degenerate. Particles learning from the degenerated exemplars is not conducive to maintaining the efficiency of the algorithm.

To protect the potential useful information of the particles, this paper proposes a dimensional learning strategy (DLS) inspired by OLPSO. In DLS, the personal best position $\boldsymbol{x}_i^{pbest}$ learns from the population best position $\boldsymbol{x}^{gbest}$ in a way of dimension by dimension to construct a learning exemplar $\boldsymbol{x}_i^{dl}$, which allows the excellent information of $\boldsymbol{x}^{gbest}$ to be passed to the exemplar $\boldsymbol{x}_i^{dl}$, promoting the spreading and protecting of the coding pattern of $\boldsymbol{x}^{gbest}$. To illustrate the proposed DLS, an example is presented in Fig. 1. Suppose that the goal is to minimize a 5-dimensional sphere function $f(\mathrm{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2$, which has the global minimum point $(0, 0, 0, 0, 0)^T$. The personal best position of particle i is $\boldsymbol{x}_i^{pbest} = (1, 0, 3, 2, 4)^T$, shown as individual at the left-up corner in Fig. 1. The current population best position is $\boldsymbol{x}^{gbest} = (2, 2, 2, 4, 0)^T$, shown as the horizontal individual in Fig. 1. It is easy to know that $f(\boldsymbol{x}_i^{pbest}) = 30$ and $f(\boldsymbol{x}^{gbest}) = 28$. Then, $\boldsymbol{x}_i^{pbest}$ learns from $\boldsymbol{x}^{gbest}$ on each dimension to form a learning exemplar $\boldsymbol{x}_i^{dl}$. We set a temporary vector $\boldsymbol{x}^{temp}$ for $\boldsymbol{x}_i^{pbest}$, i.e., $\boldsymbol{x}^{temp} = \boldsymbol{x}_i^{pbest} = (1, 0, 3, 2, 4)^T$. Initially, let $\boldsymbol{x}_i^{dl} = \boldsymbol{x}_i^{pbest} = (1, 0, 3, 2, 4)^T$. As that is shown in Fig. 1, the dimension learning process is as follows:

1) For dimension 1: Let $x_1^{temp} = x_1^{gbest} = 2$, $\boldsymbol{x}^{temp} = (2, 0, 3, 2, 4)^T$, $f(\boldsymbol{x}^{temp}) = 33 > f(\boldsymbol{x}_i^{dl}) = 30$, and thus, $x_{i,1}^{dl}$ remains unchanged; $\boldsymbol{x}_i^{dl} = (1, 0, 3, 2, 4)^T$, $f(\boldsymbol{x}_i^{dl}) = 30$.
2) For dimension 2: $\boldsymbol{x}^{temp} = \boldsymbol{x}_i^{dl}$, $x_2^{temp} = x_2^{gbest} = 2$, $\boldsymbol{x}^{temp} = (1, 2, 3, 2, 4)^T$, $f(\boldsymbol{x}^{temp}) = 34 > f(\boldsymbol{x}_i^{dl}) = 30$, and thus, $x_{i,2}^{dl}$ remains unchanged; $\boldsymbol{x}_i^{dl} = (1, 0, 3, 2, 4)^T$, $f(\boldsymbol{x}_i^{dl}) = 30$.
3) For dimension 3: $\boldsymbol{x}^{temp} = \boldsymbol{x}_i^{dl}$, $x_3^{temp} = x_3^{gbest} = 2$, $\boldsymbol{x}^{temp} = (1, 0, 2, 2, 4)^T$ and $f(\boldsymbol{x}^{temp}) = 25 < f(\boldsymbol{x}_i^{dl}) = 30$, and thus, $x_{i,3}^{dl} = x_3^{gbest} = 2$; $\boldsymbol{x}_i^{dl} = (1, 0, 2, 2, 4)^T$, $f(\boldsymbol{x}_i^{dl}) = 25$.
4) For dimension 4: $\boldsymbol{x}^{temp} = \boldsymbol{x}_i^{dl}$, $x_4^{temp} = x_4^{gbest} = 4$, $\boldsymbol{x}^{temp} = (1, 0, 2, 4, 4)^T$, $f(\boldsymbol{x}^{temp}) = 35 > f(\boldsymbol{x}_i^{dl}) = 25$, and thus, $x_{i,4}^{dl}$ remains unchanged; $\boldsymbol{x}_i^{dl} = (1, 0, 2, 2, 4)^T$, $f(\boldsymbol{x}_i^{dl}) = 25$.
5) For dimension 5: $\boldsymbol{x}^{temp} = \boldsymbol{x}_i^{dl}$, $x_5^{temp} = x_5^{gbest} = 0$, $\boldsymbol{x}^{temp} = (1, 0, 2, 2, 0)^T$, $f(\boldsymbol{x}^{temp}) = 9 < f(\boldsymbol{x}_i^{dl}) = 25$, and thus, $x_{i,5}^{dl} = x_5^{gbest} = 0$; $\boldsymbol{x}_i^{dl} = (1, 0, 2, 2, 0)^T$, $f(\boldsymbol{x}_i^{dl}) = 9$

Finally, the exemplar is $\boldsymbol{x}_i^{dl} = (1, 0, 2, 2, 0)^T$, in which the third and the fifth dimensions with boldface are learnt from the $\boldsymbol{x}^{gbest} = (2, 2, 2, 4, 0)^T$. Hence, in the proposed DLS process, $\boldsymbol{x}_i^{pbest}$ learns from $\boldsymbol{x}^{gbest}$ to construct $\boldsymbol{x}_i^{dl}$, and the final exemplar $\boldsymbol{x}_i^{dl}$ is constructed by combining $\boldsymbol{x}_i^{pbest}$ with the dimensional components learnt from $\boldsymbol{x}^{gbest}$. Moreover, DLS guarantees that $\boldsymbol{x}_i^{dl}$ is not worse than $\boldsymbol{x}_i^{pbest}$. We propose a PSO variant, named as dimension learning particle swarm optimization (DLPSO), by substituting $\boldsymbol{x}_i^{pbest}$ with $\boldsymbol{x}_i^{dl}$ in the classical PSO velocity formula. The improved velocity update formula is as following:

$$v_{i,j} = wv_{i,j} + c_1 r_{1,j}\left(x_{i,j}^{dl} - x_{i,j}\right) + c_2 r_{2,j}\left(x_j^{gbest} - x_{i,j}\right). \tag{9}$$

As illustrated by the example shown in Fig. 1, the temporary exemplar is compared with the current exemplar at each time a dimension of the former is updated. If the temporary exemplar is better than the current learning exemplar, the learning exemplar is updated with the temporary exemplar; otherwise, the current learning exemplar does not change and continues learning for the next dimension. Therefore, the learning exemplar learns only from the dimension of the population best position that could improve its fitness, which ensures that the learning exemplar will not be degraded and hence, weaken the phenomenon of "two steps forward, one step back".

In OLPSO, OED is performed with $\boldsymbol{x}_i^{pbest}$ and $\boldsymbol{x}^{gbest}$, and then an orthogonal array is used to generate M test solutions, where $M = 2^{\log_2(n+1)}$ $(M \geq n)$. According to the OED method, M solutions are used to derive the predictive solution $\boldsymbol{x}^p$. Finally, $M + 1$ additional solutions are compared to select the best one as the learning exemplar $\boldsymbol{x}^{ol}$. Denote $M_{ol} = M + 1$. Therefore, M_{ol} fitness evaluations are additionally required.

In the proposed DLPSO, $\boldsymbol{x}_i^{pbest}$ learns from each dimension of $\boldsymbol{x}^{gbest}$ and continues learning for the next dimension if $\boldsymbol{x}_i^{pbest}$ is the same as $\boldsymbol{x}^{gbest}$ in the jth dimension (as shown in Algorithm 2). Denote the fitness evaluations needed to construct the learning exemplar $\boldsymbol{x}^{dl}$ in the DLPSO algorithm as M_{dl}. It's easy to know that n fitness evaluations is required to construct the learning exemplar $\boldsymbol{x}^{dl}$, at most. Hence, comparing with OLPSO, it holds that $M_{dl} \leq n < M_{ol} = 2^{\log_2(n+1)} + 1$. DLPSO requires much fewer fitness evaluations to construct the learning exemplar, especially,

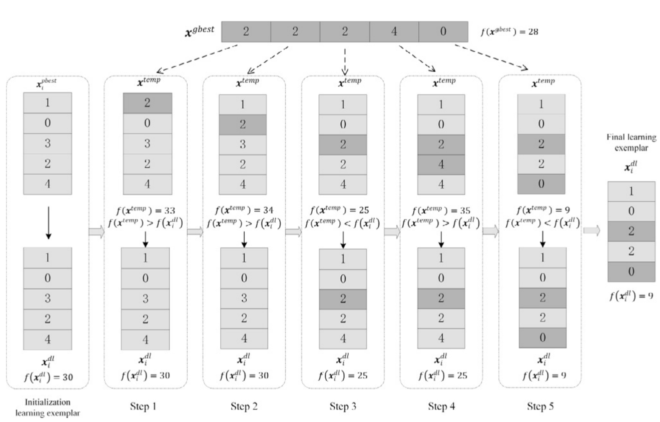

Fig. 1. Dimension learning process.

for the large dimension n. DLPSO is more suitable for the optimization problems with time-consuming fitness evaluations.

Algorithm 1
DLPSO_Phase.

Update the velocity $\boldsymbol{v}_i$ of particle i in swarm1 using Eq. (9)
Update the position $\boldsymbol{x}_i$ of particle i in swarm1 using Eq. (2)
Evaluate the fitness value of $\boldsymbol{x}_i$
if $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_i^{pbest})$ **then**
  $\boldsymbol{x}_i^{pbest} = \boldsymbol{x}_i$
  $flag_i = 1$
else
  $flag_i = 0$
end if
if $flag_i == 1$ **then**
  $Update\_exemplar_{dl}(\boldsymbol{x}_i^{pbest})$
end if

Algorithm 2
Update_exemplar$_{\rm dl}$($\boldsymbol{x}_i^{pbest}$).

$\boldsymbol{x}_i^{dl} = \boldsymbol{x}_i^{pbest}$
for each dimension j **do**
  $temp = \boldsymbol{x}_i^{dl}$
  if $temp(j) == \boldsymbol{x}^{gbest}(j)$
    Continue
  end if
  $temp(j) = \boldsymbol{x}^{gbest}(j)$
  if $f(temp) < f(\boldsymbol{x}_i^{dl})$
    $\boldsymbol{x}_i^{dl}(j) = \boldsymbol{x}^{gbest}(j)$
  end if
end for

Algorithm 3
CLPSO_Phase.

Update the velocity $\boldsymbol{v}_i$ of particle i in swarm1 using Eq. (7)
Update the velocity $\boldsymbol{x}_i$ of particle i in swarm1 using Eq. (2)
Evaluate the fitness value of $\boldsymbol{x}_i$
if $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_i^{pbest})$ **then**
  $\boldsymbol{x}_i^{pbest} = \boldsymbol{x}_i$
  $k_i = 0$
else
  $k_i = k_i + 1$
end if
if $k_i > K$ then
  $Update\_exemplar_{cl}(\boldsymbol{x}^{pbest})$
end if

Algorithm 4
Update_exemplar$_{\rm cl}$($\boldsymbol{x}^{pbest}$).

for each dimension j **do**
  if rand $< \eta_i$
    $a = \mathrm{random}(1, m)$
    $\mathrm{b} = \mathrm{random}(1, m)$
    if $f(\boldsymbol{x}_a^{pbest}) \le f(\boldsymbol{x}_b^{pbest})$ **then**
      $\boldsymbol{x}_i^{dl}(j) = \boldsymbol{x}_a^{pbest}(j)$
    else
      $\boldsymbol{x}_i^{dl}(j) = \boldsymbol{x}_b^{pbest}(j)$
    end if
  else
    $\boldsymbol{x}_i^{dl}(j) = \boldsymbol{x}_i^{pbest}(j)$
  end if
end for

Algorithm 5
TSLPSO_Phase.

Randomly initialize the position of all particle $\boldsymbol{x}$
Randomly initialize the velocity of all particle $\boldsymbol{v}$
swarm 1:$m1$; swarm 2:$m2$
Evaluate the fitness value of $\boldsymbol{x}$
Reproduction and updating loop
while FEs < Max_FEs
  update the swarm 1:
    for each particle i **do**
      DLPSO_Phase
    end for
  update the swarm 1:
    for each particle i **do**
      CLPSO_Phase
    end for
  update $\boldsymbol{x}^{gbest}$
  perform *Gaussian mutation*($\boldsymbol{x}^{gbest}$)
end while

3.2. The proposed TSL PSO algorithm

The exploration focuses on the discovery of more promising solution areas, and the exploitation focuses on the fine search in the current optimal areas. Therefore, balancing exploration and exploitation is a key problem that must be considered in designing effective algorithms. In DLS, each particle learns from $\boldsymbol{x}^{gbest}$, which leads all particles to being close to $\boldsymbol{x}^{gbest}$ with strong exploitation ability, potentially resulting in premature convergence. To improve the global search ability, inspired by HCLPSO, we also design two subpopulations called the DL-subpopulation and CL-subpopulation, respectively. Both designed subpopulations share the same position update formula given in Eq. (2). However, they differ from the strategies used to construct the learning exemplars. The DL-subpopulation adopts DLS proposed in last section to construct the learning exemplars, updating its velocity by using Eq. (9), which endows the subpopulation strong local exploitation ability; the CL-subpopulation adopts the CLS proposed by Liang et al. [20] to construct the learning exemplars, updating its velocity by using Eq. (7), which endows the subpopulation strong global exploration ability. In the DL-subpopulation, if $\boldsymbol{x}_i^{pbest}$ is updated, then a new learning exemplar $\boldsymbol{x}_i^{dl}$ is regenerated using the DLS. The search process of the DL-subpopulation is shown in Algorithm 1, and the process of constructing an exemplar $\boldsymbol{x}_i^{dl}$ is shown in Algorithm 2. In the CL-subpopulation, if $\boldsymbol{x}_i^{pbest}$ is not updated within the successive K generations, then a new learning exemplar $\boldsymbol{x}_i^{cl}$ is regenerated using the CLS. The search process of the CL-subpopulation is shown in Algorithm 3, and the process of constructing exemplar $\boldsymbol{x}_i^{cl}$ is shown in Algorithm 4. In addition, to prevent the population from falling into local optima, a Gaussian mutation is performed on $\boldsymbol{x}^{gbest}$ to help it jump out of local optima. We call this two-swarm learning particle swarm optimization algorithm based on different learning strategies as TSLPSO. Contrary to the work process of the two subpopulations in HCLPSO, the two subpopulations of TSLPSO use different learning strategies: one subpopulation adopts DLS, and the other subpopulation adopts CLS.

The process of TSLPSO is shown in Algorithm 5. In TSLPSO, as shown in Eq. (9), particles of the DL-subpopulation learn not only from the exemplar constructed by the DLS, but also from the best experience of the entire population. Therefore, the DL-subpopulation has stronger exploitation ability. The particles of the CL-subpopulation learn only from the personal best experience of the other particles selected randomly from this single subpopulation, which indicates that there is no information flow from the DL-subpopulation to the CL-subpopulation. The particles in the CL-subpopulation resists the influence of the population best experience and hence, has high population diversity and strong exploration ability. The particles in the DL-subpopulation learn from $\boldsymbol{x}^{gbest}$, and the excellent information of $\boldsymbol{x}^{gbest}$ is inherited by the exemplar $\boldsymbol{x}_i^{dl}$; thus, the particles have strong exploitation ability.

The population diversity can be used to measure the exploration and exploitation [45]. To more intuitively measure the exploration and exploitation abilities of the two subpopulations of the TSLPSO, we compare the diversities of the DL-subpopulation, the CL-subpopulation and the whole population of TSLPSO. The population diversity is calculated according to Eqs. (10) and (11) [45]:

$$Diversity = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}\left(x_{i,j}-\overline{x_j}\right)^2} \tag{10}$$

$$\overline{x_j} = \frac{\sum_{i=1}^{m} x_{i,j}}{m}, \tag{11}$$

where m is the population size, n is the dimension of the search space, $x_{i,j}$ denotes the jth dimension of the ith particle, and $\overline{x_j}$ denotes the jth dimension of the center position $\overline{\boldsymbol{x}}$ of the population.

Fig. 2 shows the diversity curves of some unimodal functions (F1 and F5) and the multimodal functions (F7 and F10), where the dimension of the search space, the population size and the maximum number of function evaluations are, 30, 20 and 300000, respectively. As shown in Fig. 2, the CL-subpopulation has higher diversity than the DL-subpopulation, and the whole population ranks between CL-subpopulation and DL-subpopulation. From the results of diversity, it could be seen that the DL-subpopulation keeps the smallest diversity, consequently, converges rapidly. As expected, the CL-subpopulation maintains the highest diversity, since there is no information from $\boldsymbol{x}^{gbest}$ as a central guiding direction. Finally, the diversity of the whole population is ranked middle, because of the balance of exploitation and exploration provided by the cooperation of DL-subpopulation and CL-subpopulation. Therefore, the experimental results of the diversity comparison verify our design expectation that the DL-subpopulation is mainly responsible for local exploitation, while the CL-subpopulation for global exploration. The interaction and cooperation of the two subpopulations ensure the global and rapid convergence of the population.

3.2. Algorithm complexity analysis

The computational costs of the traditional PSO include the initialization, evaluation, and velocity and position update, which have the complexities of $O\,(mn)$, $O\,(mn)$, and $O\,(2mn)$, respectively (n and m are the dimension and swarm size, respectively). Thus, the time complexity of the PSO is $O\,(mn)$. Compared with the traditional PSO, TSLPSO needs to update the learning exemplars. However, the learning exemplars need to be reconstructed only when the personal best position of the particle is updated or the personal best position is not updated within several generations. The worst-case time complexity of constructing the learning exemplars is $O\,(mn)$. Therefore, the worst-case complexity of TSLPSO is also $O\,(mn)$, including the initialization $O\,(mn+mn)$, evaluation $O\,(mn)$, and update $O\,(2mn+mn)$. According to the above component complexity analysis, TSLPSO has the time complexity at the same level as the basic PSO algorithm. PSO requires m fitness evaluations each generation. TSLPSO requires additional evaluations for constructing the learning exemplars by DLS and requires $m_1 n$ fitness evaluations for each generation (m_1 is the swarm size of the DL-subpopulation). Thus, the total number of fitness evaluations is $m+\,m_1 n$. When the max number of fitness evaluations (*Max_FEs*) are the same, TSLPSO requires fewer iterations than PSO.

4. Experimental verification and analysis

There are 3 experiments presented in this section. Experiment 1 includes 16 classical benchmark problems to verify the basic strength of the proposed algorithms: DLPSO and TSLPSO. Experiment 2 is performed with the CEC2014 test suite to verify the strength on functions with shift and rotation. Experiment 3 is designed for 1 real-world problem. The first

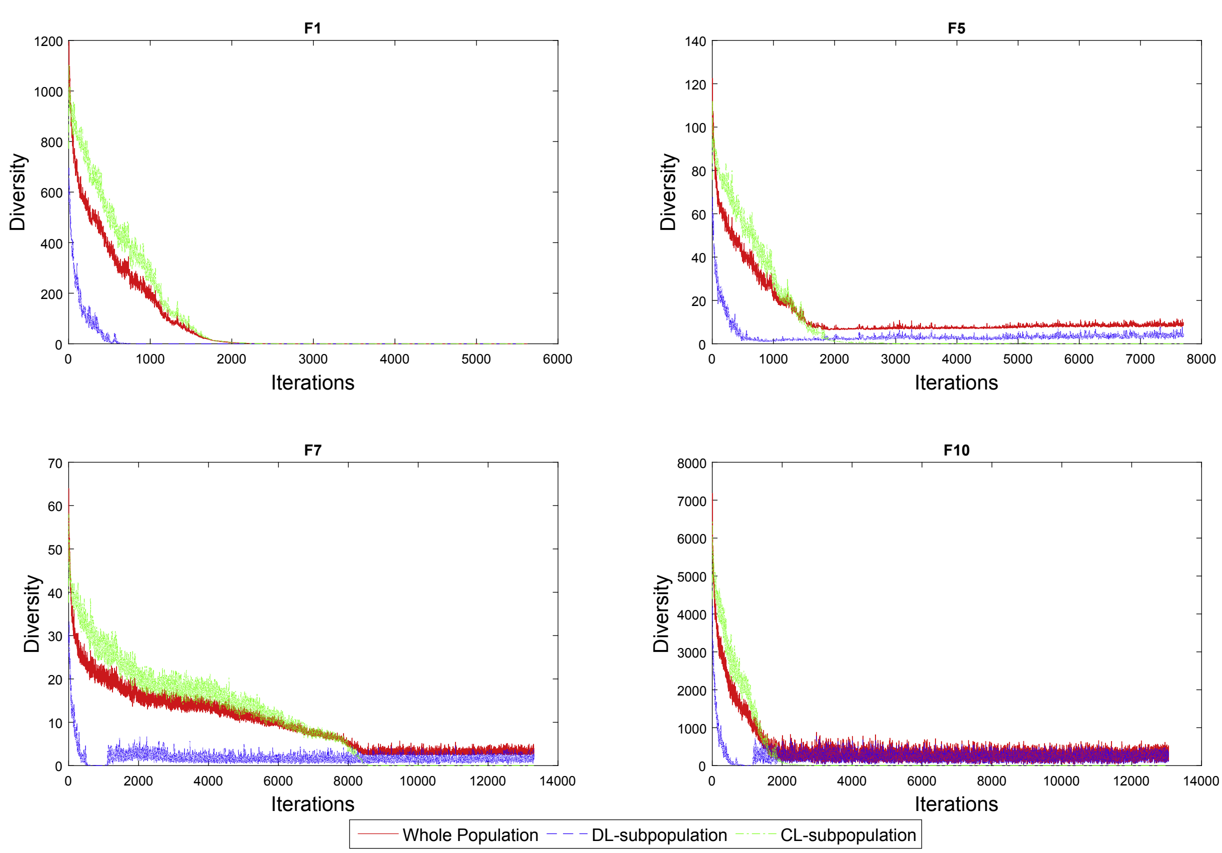

Fig. 2. Diversity comparisons between DL-subpopulation, CL-subpopulation, and the whole population.

test suite (F1-F16) includes 5 unimodal functions and 11 multimodal functions (Appendix 1), which has been widely used in Refs. [20,32,33,46]. The second test suite (F17-F46) consists of 30 shifted and rotated functions from the CEC2014 test suite [47] (Appendix 2). The effectiveness and efficiency of the proposed DLPSO and TSLPSO are compared with 5 representative PSO variants and 1 representative differential evolution (DE) algorithm. The DE is compared because it has become one of the most popular global optimization tools [48–50]. Therefore, the algorithms in comparison include PSO [10], CLPSO [20], OLPSO [32], HCLPSO [26], GL-PSO [33], and L-SHADE [51]. Among these comparing algorithms, HCLPSO, GL-PSO, and L-SHADE have been tested on shifted and rotated functions of CEC test suite in the corresponding existing literature. In order to be consistent, we fine-tuned the parameters of several other algorithms on the CEC2014 test suite. For some dynamic parameters ranging in certain interval, we set the common ranging interval. For example, the ranging interval of inertial coefficient, w, for PSO, CLPSO, OLPSO, DLPSO, and TSLPSO, is uniformly set as [0.9–0.4]. Moreover, we tune the fixed parameters with grid search in a common interval. For example, for the parameter c (including c_1 and c_2), we perform the grid search within [0.5, 1.0, 1.5, 2.0, 2.5, 3.0]. The final parameter settings of these algorithms are listed in Table 1. Each problem is tested 31 times independently. All algorithms are implemented in MATLAB 2015b and run on the same machine with an Intel (R) Xeon (R) CPU W3503 @ 2.40 GHz and 6G memory.

Among these comparing algorithms, DLPSO uses the dimensional learning strategy proposed in this paper to construct learning exemplar. The proposed TSLPSO consists of two subpopulations, named DL-subpopulation and CL-subpopulation, which take into account the advantage of DLPSO and CLPSO, respectively. HCLPSO adopts comprehensive learning strategy to construct learning exemplar, and designs two subpopulations responsible for exploitation and exploration, respectively. OLPSO uses orthogonal learning strategy to construct learning exemplar. CLPSO adopts comprehensive learning strategy, which stochastically combines the positions of personal best individuals to construct learning exemplar. GL-PSO integrates the advantages of genetic algorithm into PSO by introducing crossover, mutation and selection operations to construct learning exemplar. In addition to these PSO variants, L-SHADE (CEC2014 champion algorithm), one of the state-of-the-art DE algorithms, is also involved as a reference algorithm due to its excellent performance on CEC2014 test suite. Originally, SHADE is an adaptive DE, which incorporates success-history based parameter adaptation. L-SHADE extends SHADE with linear population size reduction (LPSR) to continually decrease the population size according to a linear function [51]. For L-SHADE, the same running parameters as those in Ref. [51] is used.

The comparing indexes include the mean (Mean), standard deviation (Std) of the best fitness function value, rank (Rank) of the average best solution for each algorithm, average fitness evaluations (FEs) when acceptable accuracy is achieved, success rate (SR), and success performance (SP). The success rate SR and the success performance SP are calculated as follows [52]:

$$SR = 100 \times \text{number of successful runs/total runs},$$

Table 1
Parameters settings.

Algorithm	Parameters settings	Reference
PSO	$w: 0.9 \sim 0.4,\ c_1 = c_2 = 2.0$	[10]
CLPSO	$w: 0.9 \sim 0.4$, c $= 1.5$	[20]
OLPSO	$w: 0.9 \sim 0.4$, c $= 1.5$	[32]
HCLPSO	$w: 0.99 \sim 0.2,\ c_1: 2.5 \sim 0.5,\ c_2: 0.5 \sim 2.5,\ c: 3 \sim 1.5$	[26]
GL-PSO	$w = 0.7298,\ c_1 = c_2 = 1.49618,\ C_R = 0.5,$ $I = 4,\ \delta = 0.2,\ F_i \in [-1, -0.4] \cup [0.4, 1]$	[33]
L-SHADE	$r^{N^{init}} = 18,\ r^{arc} = 2.6,\ p = 0.11,\ H = 6$	[48]
DLPSO	$w: 0.9 \sim 0.4,\ c_1 = 1.5,\ c_2: 0.5 \sim 2.5$	–
TSLPSO	$w: 0.9 \sim 0.4,\ c_1 = c_1 = 1.5,\ c_3: 0.5 \sim 2.5$	–

$$SP = \text{mean (FEs for successful runs)} \times \text{(total runs) / (number of successful runs)}$$

Convergence speed is measured with the mean number of FEs required to reach an acceptable solution among successful runs. Success rate (SR) is defined as the percentage of trial runs reaching acceptable accuracy. Success performance (SP) is defined as the quotient of the mean FEs and SR.

To make the experimental results more convincing, we use the well-known nonparametric statistics analysis, i.e., the Friedman test and Wilcoxon signed ranks test [53], to verify whether the differences between the algorithm results are significant.

4.1. Experimental results and analysis for benchmark functions

This section tests 8 algorithms, including the proposed DLPSO and TSLPSO on the first test suite (listed in Appendix 1). F1-F5 are unimodal functions. F2 is a noisy quartic function. F5 is unimodal in 2-D and 3-D spaces, but has multiple optima when dimension is larger than 3. These functions are used to compare the convergence feature of the algorithm, since many PSO variants improve the global search ability at the cost of slowing down the convergence speed. Then F6-F16 are multimodal functions with many local optima. F8 is noncontinuous version of the Rastrigin function F7. F15 and F16 are mis-scale versions of F7. These functions are used to test whether the proposed algorithms improve the exploration ability. The problem dimension, population size and max number of fitness evaluations are set uniformly as 30, 20 and 3×10^5 ($n \times 10^4$), according to experience reported in the literature [52]. Each function is run 31 times independently. After primary experimental testing, we set the subpopulation sizes for Subpopulation 1 and Subpopulation 2 as 8 and 12, respectively. The experimental results are shown in Tables 2 and 3.

4.1.1. Comparison and analysis of the algorithm accuracy

Table 2 lists the experimental results of the 8 algorithms on 16 benchmark functions. The boldface in Table 2 indicates the best experimental result among the 8 algorithms. The final rank for each algorithm is the average rank of all 16 benchmark functions. As shown in Table 2, TSLPSO achieves the best rank for the 16 classical benchmark functions, especially, F1, F3, F7, F8, F10, F13, F15, and F16. The global optimum of function F5 is in a narrow area, which is difficult to find for most algorithms. In addition, F5 could be considered as a multimodal function, since the dimension number is higher than 3 in our experiments. Therefore, the results of most reference algorithms for F5 are agreeable. Although L-SHADE finds the optima for 3 out of 5 unimodal problems, one more than the proposed TSLPSO, it performs worst on F2 and F3. DLPSO has the best result on the noise function F2, even with the single learning strategy DLS. Generally, the algorithm TSLPSO has shown better robust performance on unimodal functions, i.e., the first 5 benchmark problems, since TSLPSO obtains smaller rank sum than CEC2014 champion algorithm L-SHADE.

Most of the multimodal functions contain a number of local optima, which may lead to premature convergence of PSO algorithms. It is difficult for the traditional PSO to locate the global optimum of the function F6, because this problem has many deep local optima far away from the global optima. Once a particle of the classical PSO is trapped into a deep local optimum, it could hardly escape. As shown in Table 2, TSLPSO, CLPSO, and GL-PSO have better ability to resist the trapping of local optima and hence, reach the high accuracy of 3.82E-04. However, the other five reference PSO variants, including L-SHDE, could not converge effectively and the worst result is 2.38E+03. The function F7 is a very complex multimodal function with a large number of local optima. For this problem, algorithms that maintain better diversity tend to produce better results. The algorithm TSLPSO obtains the global optimum for the functions F7, F8, F10, F15 and F16, due to the population

Table 2
Search result comparisons of PSOs on benchmark functions.

Functions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
F1	Mean	**0.00E+00**	1.58E-60	4.17E-186	4.03E-41	2.55E-191	3.00E-27	2.20E-209	**0.00E+00**
	Std	0.00E+00	4.64E-60	0.00E+00	8.17E-41	0.00E+00	1.58E-26	0.00E+00	0.00E+00
	Rank	1	5	4	6	3	7	2	1
F2	Mean	2.49E-03	2.35E-03	1.27E-02	3.24E-03	1.93E-03	4.22E-03	2.54E-02	2.00E-01
	Std	1.03E-03	8.17E-04	6.03E-03	7.31E-04	9.56E-04	1.79E-03	7.60E-03	1.41E-01
	Rank	3	2	6	4	1	5	7	8
F3	Mean	**1.19E-169**	1.46E-38	9.83E-103	6.29E-26	1.98E-90	6.02E-15	9.33E-115	5.68E-14
	Std	0.00E+00	1.23E-38	3.79E-102	3.46E-26	1.09E-89	2.27E-14	4.08E-114	0.00E+00
	Rank	1	5	3	6	4	7	2	8
F4	Mean	2.50E-16	5.46E+00	1.11E+00	6.92E+01	1.07E-14	1.95E-12	4.24E-21	**0.00E+00**
	Std	8.14E-16	3.67E+00	5.23E+00	2.67E+01	3.59E-14	3.60E-12	1.59E-20	0.00E+00
	Rank	3	7	6	8	4	5	2	1
F5	Mean	1.73E+00	2.10E+00	5.31E+00	2.02E+01	1.56E+01	3.58E+01	5.20E-01	**0.00E+00**
	Std	3.05E+00	3.90E+00	1.70E+01	1.35E+01	1.64E+01	3.03E+01	5.63E-01	0.00E+00
	Rank	3	4	5	7	6	8	2	1
F6	Mean	3.82E-04	5.35E+01	3.78E+02	**3.82E-04**	7.64E+00	2.38E+03	3.82E-04	2.56E-01
	Std	2.62E-07	8.56E+01	2.22E+02	4.54E-13	2.96E+01	4.25E+02	1.36E-12	5.94E-02
	Rank	3	6	7	1	5	8	2	4
F7	Mean	**0.00E+00**	**0.00E+00**	5.81E+00	**0.00E+00**	**0.00E+00**	3.45E+01	6.28E-14	**0.00E+00**
	Std	0.00E+00	0.00E+00	2.92E+00	0.00E+00	0.00E+00	7.94E+00	2.84E-13	0.00E+00
	Rank	1	1	3	1	1	4	2	1
F8	Mean	**0.00E+00**	1.29E-01	9.61E+00	3.23E-02	**0.00E+00**	2.25E+01	1.03E-14	**0.00E+00**
	Std	0.00E+00	3.41E-01	2.68E+00	1.80E-01	0.00E+00	9.26E+00	2.80E-14	0.00E+00
	Rank	1	4	5	3	1	6	2	1
F9	Mean	2.10E-14	1.56E-14	**4.93E-15**	1.24E-14	1.43E-14	1.25E-14	8.37E-15	1.14E-13
	Std	3.22E-15	3.86E-15	1.76E-15	2.57E-15	3.49E-15	9.57E-15	3.73E-15	0.00E+00
	Rank	7	6	1	3	5	4	2	8
F10	Mean	**0.00E+00**	**0.00E+00**	6.35E-03	**0.00E+00**	1.64E-02	1.87E-02	8.33E-03	**0.00E+00**
	Std	0.00E+00	0.00E+00	1.03E-02	0.00E+00	3.93E-02	1.69E-02	1.16E-02	0.00E+00
	Rank	1	1	2	1	7	5	3	1
F11	Mean	1.57E-32	1.57E-32	6.69E-03	1.57E-32	1.57E-32	2.68E-02	1.67E-32	**0.00E+00**
	Std	5.56E-48	5.56E-48	2.59E-02	5.56E-48	5.56E-48	4.61E-02	2.07E-33	0.00E+00
	Rank	2	2	4	2	2	5	3	1
F12	Mean	1.35E-32	1.35E-32	3.54E-04	1.35E-32	1.35E-32	2.84E-03	3.54E-04	**0.00E+00**
	Std	2.78E-48	2.78E-48	1.97E-03	2.78E-48	2.78E-48	4.89E-03	1.97E-03	0.00E+00
	Rank	2	2	3	2	2	5	4	1
F13	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.60E-15	6.92E-02	3.12E-02	**0.00E+00**
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.02E-15	1.55E-01	3.14E-02	0.00E+00
	Rank	1	1	1	1	2	4	3	1
F14	Mean	4.57E-10	4.57E-10	3.04E-02	**4.57E-10**	4.57E-10	6.32E+00	4.57E-10	4.57E-10
	Std	8.96E-15	6.91E-15	1.69E-01	0.00E+00	7.07E-15	2.19E+00	1.20E-14	3.36E-14
	Rank	2	3	7	1	6	8	4	5
F15	Mean	**0.00E+00**	**0.00E+00**	5.65E+00	**0.00E+00**	**0.00E+00**	4.45E+01	6.42E-01	1.94E-13
	Std	0.00E+00	0.00E+00	2.11E+00	0.00E+00	0.00E+00	1.05E+01	7.51E-01	5.91E-14
	Rank	1	1	4	1	1	5	3	2
F16	Mean	**0.00E+00**	6.42E-02	7.06E+00	**0.00E+00**	**0.00E+00**	4.54E+01	5.91E+00	2.02E-13
	Std	0.00E+00	2.48E-01	3.31E+00	0.00E+00	0.00E+00	1.02E+01	3.64E+00	8.02E-14
	Rank	1	3	5	1	1	6	4	2
Ave Rank		1.94	3.12	3.88	2.82	3.00	5.41	2.76	2.71
Final Rank		1	6	7	4	5	8	3	2

diversity resulted from DLS and CLS.

As observed in the results of the unimodal and multimodal functions, the algorithm TSLPSO achieves the global optima for 8 benchmark functions, i.e., F1, F3, F7, F8, F10, F13, F15, and F16. L-SHADE achieves the global optima for 9 benchmark functions, i.e., F1, F4, F5, F7, F8 and F10~F13. In the final average rank, the proposed TSLPSO ranks first statistically among 8 algorithms, with L-SHADE ranking second, followed by GL-PSO, CLPSO, DLPSO, HCLPSO, OLPSO and PSO. As a reference algorithm, the proposed DLPSO obtains 5 optima for 5 benchmark problems and ranks the fifth. Hence, the experimental results demonstrate the strength of DLS and the importance of subpopulation incorporation.

4.1.2. *Comparison and analysis of the convergence speed and reliability*

To compare the convergence speed, algorithm reliability and success performance, some key comparison indexes are reported in Table 3, including FEs, SR, and SP. According the typical style, FEs and SR are used to quantify the convergence speed and reliability. As shown in Table 3, TSLPSO is the fastest algorithm for 7 benchmark problems, followed by GL-PSO for 3 (F1, F3 and F10), DLPSO for 4 (F2, F6, F7, and F14) and L-SHADE for 2 (F4 and F5). According to the SR, TSLPSO is most reliable with 100% mean success rate, followed by DLPSO with success rate of 98.62%. The rest reliability orders are as follows: L-SHADE, CLPSO, HCLPSO, GL-PSO, OLPSO, and PSO.

Due to the space limit, we present the convergence curves for 5 unimodal functions (F1, F2, F3, F4, and F5) and 6 multimodal functions (F6, F7, F9, F10, F11, and F13) in Fig. 3 and Fig. 4, respectively. As shown in Fig. 3, except for function F5, TSLPSO has a higher convergence accuracy and faster convergence speed on most unimodal functions. As shown in Fig. 4, for most of multimodal function, TSLPSO not only has the highest convergence accuracy, but also converges fastest. Moreover, DLPSO with only DLS shows very competitive performance on these multimodal functions. Therefore, TSLPSO is generally effective for both unimodal functions and multimodal functions with regards to the accuracy and convergence speed.

Table 4 depicts the ranks calculated through the Friedman test for benchmark problems. As shown in Table 4, TSLPSO ranks first among the 8 algorithms. The p-values computed from the statistics of the Friedman

Table 3
Comparisons of convergence speed, algorithm reliability and success performance.

Functions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
F1	FEs	14479.32	62839.94	57385.35	93958.23	20787.97	70541.94	**12415.48**	74063.90
	SR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	SP	14479.32	62839.94	57385.35	93958.23	20787.97	70541.94	12415.48	74063.90
F2	FEs	94123.97	80315.55	175166.29	136060.26	**68450.84**	156250.32	–	123722.00
	SR	100.00	100.00	45.16	100.00	100.00	100.00	0.00	6.45
	SP	94123.97	80315.55	387868.20	136060.26	68450.84	156250.32	–	1917691.00
F3	FEs	21223.71	66216.00	62822.61	109843.61	34687.87	71344.52	**16281.29**	110762.03
	SR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	SP	21223.71	66216.00	62822.61	109843.61	34687.87	71344.52	16281.29	110762.03
F4	FEs	117498.29	–	–	–	131916.16	191465.81	103496.77	**98605.35**
	SR	100.00	0.00	0.00	0.00	100.00	100.00	100.00	100.00
	SP	117498.29	–	–	–	131916.16	191465.81	103496.77	98605.35
F5	FEs	12420.39	44042.61	48204.81	82452.00	12304.52	61100.00	5867.10	**554.23**
	SR	100.00	100.00	100.00	100.00	100.00	96.77	100.00	100.00
	SP	12420.39	44042.61	48204.81	82452.00	12304.52	63136.67	5867.10	554.23
F6	FEs	13815.32	24198.97	6587.87	31656.94	**1256.55**	71431.43	9083.87	195399.84
	SR	100.00	100.00	100.00	100.00	100.00	22.58	100.00	100.00
	SP	13815.32	24198.97	6587.87	31656.94	1256.55	316339.18	9083.87	195399.84
F7	FEs	33727.61	83851.52	–	185481.52	**21293.74**	–	79970.32	55929.71
	SR	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00
	SP	33727.61	83851.52	–	185481.52	21293.74	–	79970.32	55929.71
F8	FEs	**17104.35**	87181.26	–	191519.33	19594.06	–	60219.35	56117.94
	SR	100.00	87.10	0.00	96.77	100.00	0.00	100.00	100.00
	SP	17104.35	100097.00	–	197903.31	19594.06	–	60219.35	56117.94
F9	FEs	**20749.77**	79335.81	70104.74	120877.87	30904.97	79256.77	24698.06	95433.42
	SR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	SP	20749.77	79335.81	70104.74	120877.87	30904.97	79256.77	24698.06	95433.42
F10	FEs	64651.97	74350.87	61292.61	112928.90	27969.56	70380.00	**21383.53**	85860.45
	SR	100.00	100.00	58.06	100.00	80.65	22.58	54.84	100.00
	SP	64651.97	74350.87	105559.50	112928.90	34682.25	311682.86	38993.49	85860.45
F11	FEs	**10338.13**	57477.52	52466.34	88969.10	12598.10	73860.87	23384.52	67228.32
	SR	100.00	100.00	93.55	100.00	100.00	74.19	100.00	100.00
	SP	10338.13	57477.52	56084.71	88969.10	12598.10	99551.61	23384.52	67228.32
F12	FEs	**14732.55**	63539.71	56483.27	95489.32	16048.58	75408.70	19317.33	75370.55
	SR	100.00	100.00	96.77	100.00	100.00	74.19	96.77	100.00
	SP	14732.55	63539.71	58366.04	95489.32	16048.58	101637.81	19961.24	75370.55
F13	FEs	**32381.55**	89545.42	71019.39	140425.13	55120.74	81040.00		118683.55
	SR	100.00	100.00	100.00	100.00	100.00	3.23	0.00	100.00
	SP	32381.55	89545.42	71019.39	140425.13	55120.74	2512240.00		118683.55
F14	FEs	20273.00	46818.74	45181.33	77218.35	**11035.81**	–	12870.97	144050.84
	SR	100.00	100.00	96.77	100.00	100.00	0.00	100.00	100.00
	SP	20273.00	46818.74	46687.38	77218.35	11035.81	–	12870.97	144050.84
F15	FEs	**6109.45**	60667.29	54918.83	136618.61	6219.94	–	41220.65	237566.29
	SR	100.00	100.00	96.77	100.00	100.00	0.00	100.00	100.00
	SP	6109.45	60667.29	56749.46	136618.61	6219.94	–	41220.65	237566.29
F16	FEs	**8613.03**	58050.29	60473.12	135318.55	9711.32	–	146216.00	239769.39
	SR	100.00	100.00	80.65	100.00	100.00	0.00	96.77	100.00
	SP	8613.03	58050.29	74986.67	135318.55	9711.32	–	151089.87	239769.39
Ave.SR		100.00	92.94	72.98	93.55	98.79	49.60	84.27	94.15
SR.Rank		1	5	7	4	2	8	6	3

test strongly indicate that there are significant differences among the 8 algorithms. In addition, we use the Wilcoxon signed rank test to display the magnitude of the differences between TSLPSO and the reference algorithms in Table 5. As shown in Table 5, TSLPSO shows a significant improvement over OLPSO and PSO with a level of significance of $\alpha = 0.01$; TSLPSO shows an improvement over HCLPSO, DLPSO, GLPSO and LSHADE with a level of significance of $\alpha = 0.1$. Although the difference between TSLPSO and CLPSO is not statistically significant, TSLPSO shows better results than CLPSO on most of the test functions. Therefore, the statistics of the Friedman test and Wilcoxon signed rank test verify the general validity of TSLPSO on benchmark problems.

4.1.3. Utility comparison and analysis of DLS and CLS

In order to investigate the utility of DLS and CLS, this section compares the results of TSLPSO with both DLS and CLS, DLPSO with DLS alone, and CLPSO with CLS alone. Appendix 3 shows the comparison results of unimodal functions and multimodal functions, respectively. As shown in Appendix 3, TSLPSO ranks first, followed by DLPSO and CLPSO for unimodal functions; CLPSO ranks first, followed by TSLPSO and DLPSO for multimodal functions. We may infer by this experiment that DLS enables the excellent information of each generation of the population best position to be inherited specifically by each particle through its exemplar, which improves the convergence accuracy and convergence speed. However, the diversity of DLPSO is not enough and easy to fall into local optima. CLS enhances the diversity of the population and is better for solving multimodal functions with many local optima. The experimental results verify our design expectation that DLS focuses on local exploitation and CLS focuses on global exploration. It's the incorporation between DLS and CLS in subpopulations that enables TSLPSO to be the most competitive algorithm for both unimodal and multimodal functions.

4.2. Experimental results and analysis of CEC2014 test suite

To further verify the validity of the proposed TSLPSO algorithm, we evaluate the performance of TSLPSO on the *CEC2014 Special Session on Real-Parameter Single Objective Optimization* benchmark suite [47]. The CEC2014 test suite consists of 30 test functions. For all problems, the

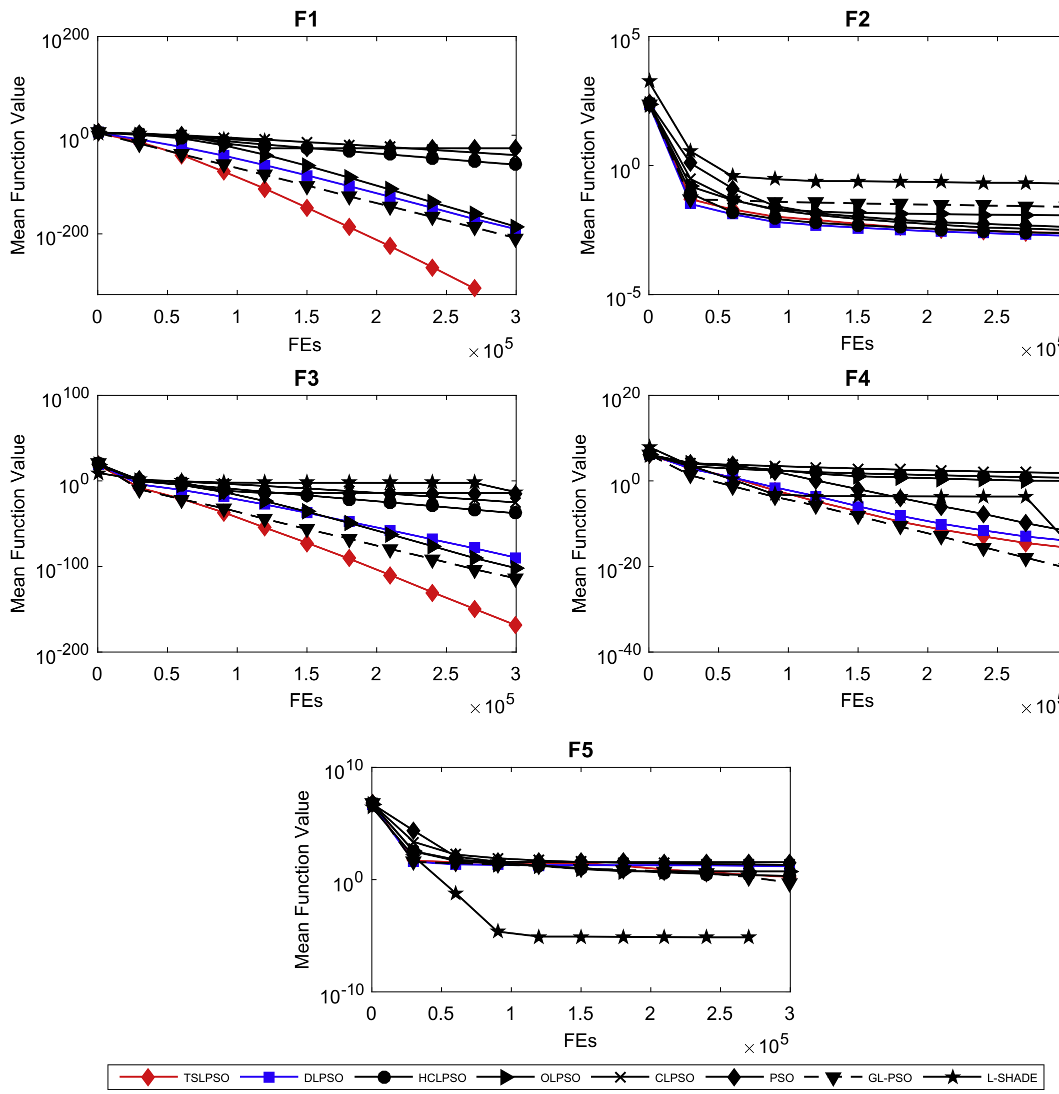

Fig. 3. Convergence performance on the 5 unimodal functions.

search range is [-100,100]^n. Functions F17∼F19 are unimodal. F20∼F32 are simple multimodal functions. F33∼F38 are hybrid functions. Finally, F39-F46 are composite functions, which combine multiple test problems into a complex landscape. The experimental results of CEC2014 test problems are presented in Table 6. The algorithm and running parameters are identical with the former experiments.

For CEC2014 test suite, L-SHADE (CEC2014 champion algorithm) achieved the best performance, ranking first. Moreover, the results of L-SHADE on most of the functions are significantly different from those of other PSO variants. For example, for function F17, the error of L-SHADE is 1.42E-14, while the error of other PSO variants is not smaller than 2.74E+05. It is notable that the proposed DLPSO ranks forth and is superior to some well-known PSO variants, including GLPSO, OLPSO and CLPSO. The reason behind the phenomenon is complicated and worth of detailed investigation in the future.

Table 7 gives the results of the Friedman test. The p-value, 0, indicates that these algorithms have significant differences in the CEC2014 problems. Among all 8 reference algorithms, L-SHADE ranks the first, HCLPSO the second and DLPSO the third and TSLPSO the forth. HCLPSO, DLPSO and TSLPSO get very close Friedman average rank. The Wilcoxon signed rank test in Table 8 shows that L-SHADE is significantly better than TSLPSO with a level of significance $\alpha = 0.01$. TSLPSO is significantly better than PSO and GL-PSO with a level of significance $\alpha = 0.05$, and TSLPSO shows an improvement over OLPSO with a level of significance of $\alpha = 0.1$. The difference between TSLPSO and other three PSOs – HCLPSO, DLPSO and CLPSO is small, although TSLPSO is slightly better than HCLPSO and CLPSO on some test functions.

To investigate the underlying factors, we test the functions with separate shift and rotation, respectively. The experimental results show that TSLPSO is effective for shift functions (Appendix 5). And the rotation functions is difficult all PSO variants, including the proposed TSLPSO (Appendix 5). Generally speaking, L-SHADE is effective for rotation functions. However, there are some rotated functions degenerating L-SHADE drastically. Since the natural difference of PSO variants and L-SHDE is the population size reduction, it is an interesting future work to investigate the PSO variants with the population size reduction.

4.3. TSLPSO performance for a real-world problem

In this section, the performance of TSLPSO is verified for a widely used real-world optimization problem known as the spread spectrum radar polyphase code design [54]. In radar systems with pulse compression, the choice of the appropriate waveform is a major problem [55]. Although the resolution of radar systems could be significantly improved by using short pulses [56], the drawback is that the use of short pulses decreases the average transmitted power in the system. In order to overcome this issue, the majority of modern radar systems generally incorporate pulse compression waveforms, since they allow achieving

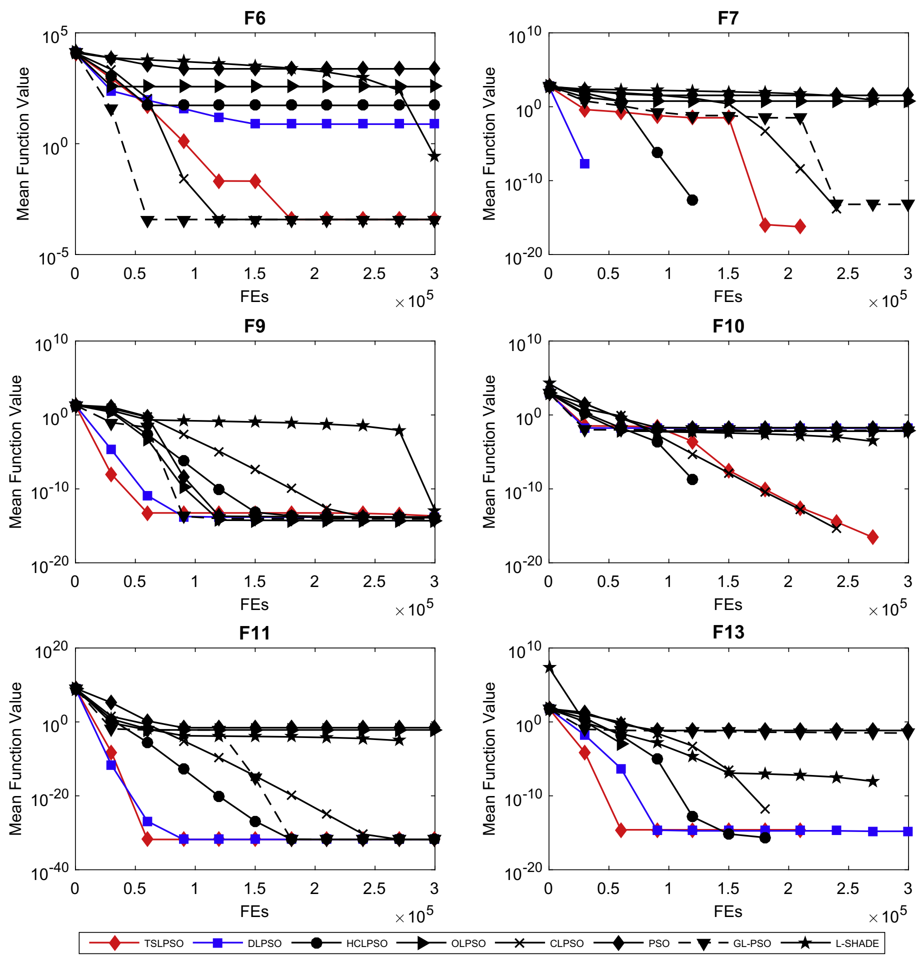

Fig. 4. Convergence performance on the 6 multimodal functions.

Table 4
Rank achieved by the Friedman test for benchmark functions.

	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE	p-Value
Friedman ranks	2.78	4.25	5.59	3.91	4.03	7.31	4.38	3.75	0.000003
Final rank	1	5	7	3	4	8	6	2	

Table 5
Wilcoxon Signed Ranks Test of benchmark functions.

TSLPSO vs.	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
R+	46	117	33	44	133	101	52
R-	9	3	12	11	3	35	14
N+	8	14	6	8	15	12	8
N-	2	1	3	2	1	4	3
T	6	1	7	6	0	0	5
P-value	0.059	0.001	0.214	0.093	0.001	0.088	0.091

the average transmitted power of a relatively long-pulse radar scheme, while simultaneously obtaining the range resolution of a short-pulse radar. Dukic et al., [54] introduced a method, named spread spectrum radar polyphase code design (SSRP), for polyphase pulse-compression code design, based on the properties of their aperiodic autocorrelation function, and also on considering coherent radar pulse processing at the receiver. Traditionally, the min-max nonlinear optimization problem model [57] is applied to solve the problem of spread spectrum radar polyphase code design and the optimization problem is formulated as follows [54]:

$$min_{x\in X}\ max\{\varphi_1(x),\varphi_2(x),\dots,\varphi_{2m}(x)\}, X=\left\{(x_1,x_2,\dots,x_n)\in R^n \middle| 0\le x_j \le 2\pi\right\}, \tag{12}$$

where $m = 2n - 1$ and

$$\varphi_{2i-1}(x)=\sum_{j=i}^{n}\cos\left(\sum_{k=|2i-j-1|+1}^{j}x_k\right),\ i=1,\dots,n$$

Table 6
Comparisons on CEC2014 test suite.

Fcnctions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
F17	Mean	5.94E+05	2.74E+05	7.42E+05	4.82E+06	4.63E+05	6.69E+06	1.39E+07	1.42E-14
	Std	5.11E+05	2.35E+05	6.73E+05	1.68E+06	3.37E+05	8.99E+06	1.31E+07	3.61E-15
	Rank	4	2	5	6	3	7	8	1
F18	Mean	8.94E+02	4.07E+02	1.52E+02	2.93E+02	2.78E+03	7.27E+01	5.90E+02	1.83E-15
	Std	1.39E+03	6.48E+02	1.30E+02	4.61E+02	3.34E+03	2.90E+02	5.94E+02	6.98E-15
	Rank	7	5	3	4	8	2	6	1
F19	Mean	5.20E+02	6.02E+02	1.56E+03	8.25E+02	6.18E+00	4.87E+01	4.25E+02	0.00E+00
	Std	8.53E+02	7.14E+02	1.80E+03	8.39E+02	2.15E+01	6.61E+01	4.84E+02	0.00E+00
	Rank	5	6	8	7	2	3	4	1
F20	Mean	3.78E+01	3.65E+01	2.95E+01	6.54E+01	1.50E+01	1.79E+02	1.15E+02	8.43E-14
	Std	3.38E+01	3.31E+01	3.57E+01	2.64E+01	2.23E+01	5.12E+01	6.30E+01	5.38E-14
	Rank	5	4	3	6	2	8	7	1
F21	Mean	2.00E+01	2.00E+01	2.02E+01	2.04E+01	2.00E+01	2.09E+01	2.00E+01	2.01E+01
	Std	1.93E-02	5.87E-03	2.76E-01	4.58E-02	2.48E-03	8.52E-02	2.79E-02	1.70E-02
	Rank	1	1	3	4	1	5	1	2
F22	Mean	9.31E+00	6.52E+00	4.85E+00	1.10E+01	7.45E+00	1.08E+01	1.00E+01	1.67E-02
	Std	3.15E+00	2.88E+00	2.81E+00	1.43E+00	3.00E+00	2.53E+00	2.25E+00	9.17E-02
	Rank	5	3	2	8	4	7	6	1
F23	Mean	1.67E-05	2.56E-04	3.49E-03	1.93E-06	1.10E-02	2.62E-02	1.13E-02	0.00E+00
	Std	4.35E-05	1.33E-03	8.61E-03	7.79E-06	1.59E-02	2.21E-02	1.54E-02	0.00E+00
	Rank	3	4	5	2	6	8	7	1
F24	Mean	1.50E-12	1.93E-01	5.23E+00	3.21E-02	2.05E-13	2.27E+01	2.29E-12	2.09E-13
	Std	7.38E-12	4.75E-01	2.37E+00	1.79E-01	5.43E-14	6.92E+00	8.97E-12	5.84E-14
	Rank	3	6	7	5	1	8	4	2
F25	Mean	4.59E+01	4.87E+01	4.75E+01	4.35E+01	5.88E+01	6.32E+01	5.79E+01	6.37E+00
	Std	8.71E+00	1.24E+01	1.52E+01	6.97E+00	1.51E+01	1.59E+01	1.57E+01	1.23E+00
	Rank	3	5	4	2	7	8	6	1
F26	Mean	3.75E-01	2.92E+01	7.32E+01	1.46E+00	3.13E-01	4.45E+02	2.22E-01	1.68E-02
	Std	3.23E-01	5.86E+01	9.04E+01	9.10E-01	3.10E-01	1.77E+02	1.95E-01	1.62E-02
	Rank	4	6	7	5	3	8	2	1
F27	Mean	1.77E+03	1.93E+03	2.43E+03	1.99E+03	2.00E+03	2.90E+03	2.47E+03	1.27E+03
	Std	3.83E+02	3.59E+02	5.84E+02	2.90E+02	4.77E+02	8.88E+02	4.64E+02	1.79E+02
	Rank	2	3	6	4	5	8	7	1
F28	Mean	1.41E-01	1.21E-01	1.44E-01	3.65E-01	1.16E-01	1.74E+00	1.38E-01	1.53E-01
	Std	4.42E-02	3.24E-02	7.35E-02	6.36E-02	3.67E-02	3.84E-01	4.74E-02	1.92E-02
	Rank	4	2	5	7	1	8	3	6
F29	Mean	2.52E-01	2.72E-01	2.41E-01	3.16E-01	2.42E-01	4.82E-01	3.54E-01	1.19E-01
	Std	6.14E-02	7.07E-02	6.09E-02	3.90E-02	5.63E-02	1.19E-01	1.11E-01	1.72E-02
	Rank	4	5	2	6	3	8	7	1
F30	Mean	2.52E-01	2.53E-01	3.20E-01	2.41E-01	2.33E-01	2.90E-01	3.01E-01	2.40E-01
	Std	2.59E-02	4.16E-02	1.06E-01	2.77E-02	7.74E-02	9.55E-02	4.30E-02	2.98E-02
	Rank	4	5	8	3	1	6	7	2
F31	Mean	4.74E+00	5.02E+00	4.56E+00	6.67E+00	4.51E+00	6.72E+00	6.43E+00	2.11E+00
	Std	1.27E+00	1.60E+00	2.60E+00	7.40E-01	1.20E+00	2.38E+00	2.16E+00	2.69E-01
	Rank	4	5	3	7	2	8	6	1
F32	Mean	9.62E+00	9.77E+00	1.15E+01	1.00E+01	9.90E+00	1.13E+01	1.03E+01	8.48E+00
	Std	5.17E-01	4.78E-01	8.24E-01	4.46E-01	7.46E-01	7.05E-01	7.20E-01	2.97E-01
	Rank	2	3	8	5	4	7	6	1
F33	Mean	1.71E+05	1.03E+05	4.28E+05	5.31E+05	1.01E+05	2.86E+05	2.53E+05	1.61E+02
	Std	1.36E+05	6.68E+04	5.30E+05	2.92E+05	7.97E+04	3.25E+05	6.14E+05	9.66E+01
	Rank	4	3	7	8	2	6	5	1
F34	Mean	2.57E+02	4.45E+02	3.73E+03	1.26E+02	1.95E+03	3.79E+03	2.88E+03	6.58E+00
	Std	3.69E+02	7.28E+02	3.34E+03	6.85E+01	2.64E+03	4.62E+03	3.25E+03	2.74E+00
	Rank	3	4	7	2	5	8	6	1
F35	Mean	6.12E+00	6.15E+00	1.03E+01	6.98E+00	1.08E+01	7.96E+00	1.18E+01	3.59E+00
	Std	1.24E+00	9.95E-01	2.14E+01	6.24E-01	1.77E+01	1.87E+00	1.11E+01	7.22E-01
	Rank	2	3	6	4	7	5	8	1
F36	Mean	2.19E+03	3.73E+03	1.09E+04	3.63E+03	2.93E+02	3.74E+02	1.15E+04	2.85E+00
	Std	2.72E+03	2.04E+03	8.07E+03	2.35E+03	3.80E+02	2.38E+02	1.00E+04	1.22E+00
	Rank	4	6	7	5	2	3	8	1
F37	Mean	4.30E+04	3.30E+04	5.82E+04	9.27E+04	3.89E+04	6.12E+04	1.75E+05	5.18E+01
	Std	4.77E+04	2.33E+04	5.31E+04	7.81E+04	2.94E+04	6.19E+04	4.94E+05	6.40E+01
	Rank	4	2	5	7	3	6	8	1
F38	Mean	2.35E+02	2.88E+02	4.87E+02	2.18E+02	3.13E+02	2.31E+02	4.75E+02	3.69E+01
	Std	1.23E+02	1.01E+02	2.03E+02	8.58E+01	1.20E+02	1.04E+02	1.78E+02	3.36E+01
	Rank	4	5	8	2	6	3	7	1
F39	Mean	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.16E+02	3.16E+02	3.15E+02
	Std	4.62E-11	7.51E-12	2.64E-11	9.57E-08	9.77E-13	3.52E-01	8.16E-01	1.14E-13
	Rank	1	1	1	1	1	3	2	1
F40	Mean	2.26E+02	2.26E+02	2.30E+02	2.24E+02	2.29E+02	2.30E+02	2.29E+02	2.24E+02
	Std	1.31E+00	1.28E+00	4.74E+00	6.36E-01	4.40E+00	6.64E+00	4.33E+00	1.15E+00
	Rank	3	4	7	2	5	8	6	1
F41	Mean	2.06E+02	2.06E+02	2.06E+02	2.06E+02	2.06E+02	2.09E+02	2.14E+02	2.03E+02
	Std	1.71E+00	2.16E+00	2.29E+00	8.04E-01	3.05E+00	1.65E+00	2.47E+00	4.97E-02

(continued on next page)

Table 6 (*continued*)

Fcnctions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
	Rank	3	4	2	5	6	7	8	1
F42	Mean	1.00E+02	1.04E+02	1.20E+02	1.00E+02	1.07E+02	1.07E+02	1.61E+02	1.00E+02
	Std	9.92E-02	1.79E+01	4.01E+01	1.08E-01	2.49E+01	2.49E+01	4.93E+01	1.29E-02
	Rank	1	2	5	1	3	4	6	1
F43	Mean	4.08E+02	4.02E+02	4.58E+02	4.12E+02	4.46E+02	5.36E+02	5.68E+02	3.00E+02
	Std	5.55E+00	2.70E+00	9.73E+01	8.76E+00	7.19E+01	8.15E+01	1.34E+02	2.31E-13
	Rank	3	2	6	4	5	7	8	1
F44	Mean	9.53E+02	9.42E+02	9.06E+02	8.79E+02	1.04E+03	1.06E+03	1.23E+03	8.45E+02
	Std	9.29E+01	6.29E+01	9.29E+01	4.06E+01	3.22E+02	1.36E+02	3.27E+02	1.29E+01
	Rank	5	4	3	2	6	7	8	1
F45	Mean	7.80E+03	1.08E+03	1.59E+03	9.55E+02	1.32E+03	1.07E+03	6.71E+05	7.16E+02
	Std	1.64E+04	1.64E+02	5.41E+02	1.08E+02	4.70E+02	2.76E+02	3.73E+06	1.76E+00
	Rank	7	4	6	2	5	3	8	1
F46	Mean	1.50E+04	2.04E+03	2.90E+03	2.45E+03	2.78E+03	3.56E+03	1.21E+04	1.09E+03
	Std	1.29E+04	4.63E+02	6.03E+02	5.88E+02	8.42E+02	1.93E+03	8.33E+03	4.56E+02
	Rank	8	2	5	3	4	6	7	1
Ave Rank		3.73	3.70	5.13	4.30	3.77	6.17	6.07	1.27
Final Rank		3	2	6	5	4	8	7	1

Table 7
Ranks achieved by the Friedman test for CEC2014 test suite.

	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE	p-Value
Ave rank	3.97	3.9	5.4	4.57	3.93	6.5	6.37	1.37	0
Final Rank	4	2	6	5	3	8	7	1	

Table 8
Wilcoxon Signed Ranks Test of CEC2014 test suite.

TSLPSO vs.	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
R+	210	306	275	195	333	387	10
R-	225	129	160	240	132	78	455
n+	17	20	19	13	24	24	2
n-	12	9	10	16	6	6	28
ties	1	1	1	1	0	0	0
p-Value	0.871	0.056	0.214	0.627	0.039	0.001	0.000005

$$\varphi_{2i}(x) = 0.5 + \sum_{j=i+1}^{n} \cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k \right), \; i = 1, \ldots, n-1$$

$$\varphi_{m+i}(x) = -\varphi_i(x), \quad i = 1, \ldots, m$$

There exists only a few approaches to solve it by using GAs and some variants of DE, due to its NP-hard property and the fact that its fitness function is piecewise smooth. Hence, there are no efficient mathematical approaches and efficient software to solve the problem of higher dimensions. In this paper, the 20-dimensional spread spectrum radar polyphase code design problem is tested for 8 PSO algorithms. The population size and max number of fitness evaluations are 20 and 2×10^5, respectively, the same as those in Ref. [58]. As shown in Table 9, the proposed TSLPSO is superior to the other 6 PSO variants and close to the CEC2014 champion algorithm L-SHADE. As shown in Fig. 5, all algorithms converge quickly in the early period and slowly in the late period. However, the proposed TSLPSO and DLPSO keeps the highest convergence speed to the end, together with L-SHADE. The algorithm TSLPSO is not only valid on the benchmark functions, but also effective for solving real-world engineering problems.

4.4. Discussions

The experimental results, including the classical benchmark functions, the real-world engineering problem and the CEC2014 test suite, verify the effectiveness of TSLPSO in terms of convergence accuracy, convergence speed and algorithm reliability. The statistical results show that for most functions, TSLPSO is significantly statistically better than the improved PSO variants and the traditional PSO. Different from HCLPSO with two swarms which adopt the same learning strategy, CLPSO, OLPSO, and GL-PSO which adopt single learning strategy, TSLPSO use different learning strategies, DLS and CLS, to maintain two subpopulations. To more intuitively measure the exploration and exploitation abilities of the two subpopulations of the TSLPSO, we compare the diversities of the DL-subpopulation, the CL-subpopulation and the whole population of TSLPSO. At the same time, we also compare and analyze the results of TSLPSO with both DLS and CLS, DLPSO with DLS alone, and CLPSO with CLS alone. We may infer by the second experiment that DLS enables the excellent information of each generation of the population best position to be inherited specifically by each particle through its exemplar, which improves the convergence accuracy and convergence speed. However, the diversity of DLPSO is not

Table 9
Comparison on the 20-dimensional spread spectrum radar polyphase code design problem.

	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
Mean	7.424E-01	8.846E-01	1.100E+00	8.641E-01	8.856E-01	1.022E+00	1.062E+00	6.569E-01
Std	2.425E-01	1.178E-01	2.801E-01	1.508E-01	1.279E-01	1.544E-01	2.473E-01	1.749E-01
Rank	2	4	8	3	5	6	7	1

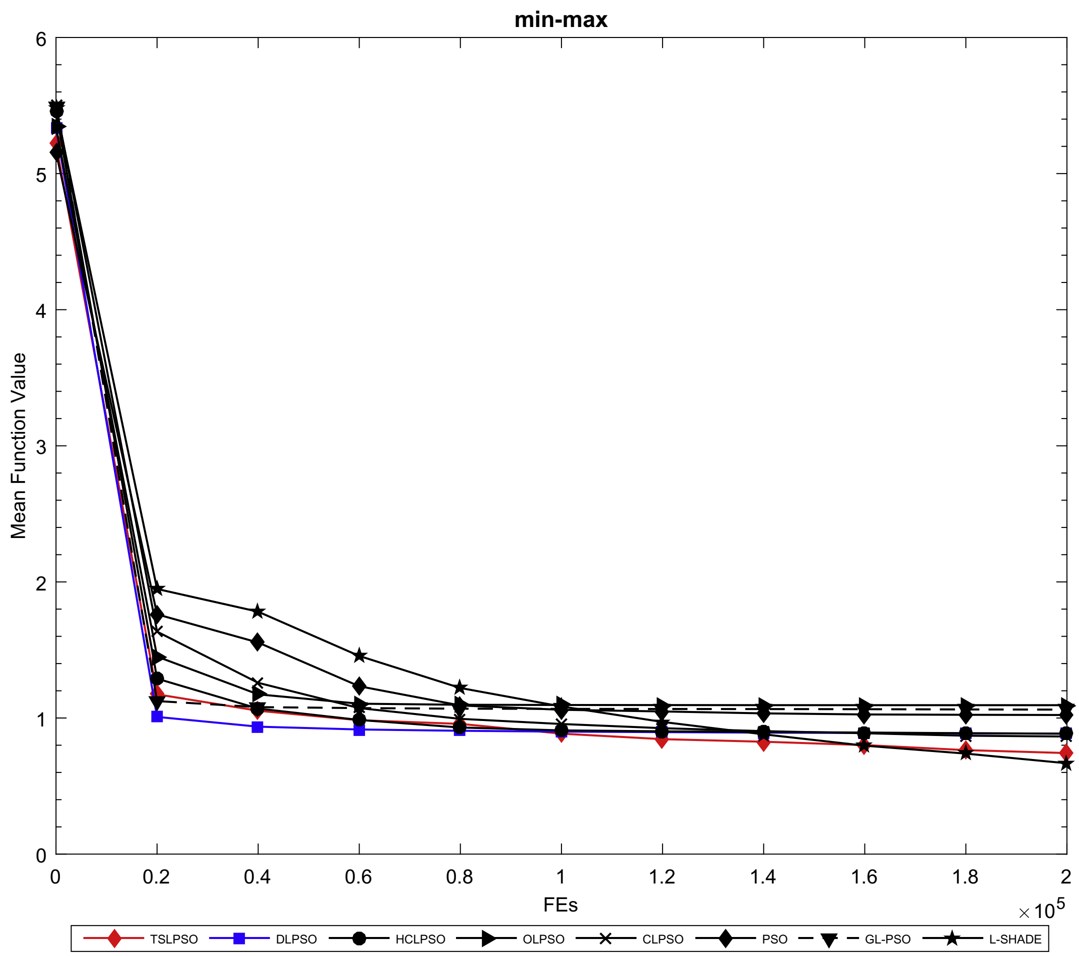

Fig. 5. Convergence curve of on the real-world problem.

enough and easy to fall into local optima. CLS enhances the diversity of the population and is better for solving multimodal functions with many local optima. Both these two experimental results verify our design expectation that the DL-subpopulation is mainly responsible for local exploitation, while the CL-subpopulation for global exploration. It's the incorporation between DLS and CLS in subpopulations that enables TSLPSO to be the most competitive algorithm for both unimodal and multimodal functions.

Most of the existing learning strategies adopt the linear weighting of the personal best position, random combination of the personal best position, and orthogonal combination of the personal best position and population best position to construct the learning exemplars. Thereafter, the learning exemplars are used to guide the particle search instead of the personal best solution and population best solution. This type of single guidance learning mechanism could effectively avoid the phenomenon of "oscillation", which is caused by dual guidance in the traditional PSO. However, the learning exemplars constructed by the above methods have strong randomness. In fact, the personal best position and the population best position are in an evolving process with iterations. Hence, there is no guarantee that the learning exemplars will not degenerate. Particles learning from the degenerated exemplars are not conducive to maintaining the efficiency of the algorithm. The DLS proposed in this paper allows each dimension of the personal best position to learn from the corresponding dimension of the population best position when constructing a learning exemplar. Moreover, dimensional learning is integrated into the learning exemplar only when it could improve the fitness of the exemplar. Thus, DLS avoids the problem of "oscillation" and "two steps forward, one step back", and improve the convergence speed and reliability of the algorithm.

Although the learning exemplar constructed by DLS could guide the particle search to move into a better region, the premature convergence may occur when most of the particles are close to the population best position. For this reason, we have introduced CLS to enhance the population diversity and help the particles escape from local optima. TSLPSO relieves the pressure of fitness evaluations of DLPSO by introducing the CL-subpopulations.

5. Conclusion

This paper first proposes the DLS, in which the personal best position of a particle learns from the corresponding dimensions of the population best position to construct a learning exemplar. The learning exemplar is combined with the excellent information of the personal best position and population best position. DLS enables the excellent information of the population best position in each generation to be inherited by each ordinary particle's personal best position, which improves the convergence accuracy and convergence speed and weakens the phenomenon of "two steps forward, one step back". Motivated by the power of existing CLS to help the population jump out of the local optima, TSLPSO is proposed with two heterogeneous subpopulations. One subpopulation adopts DLS and focuses on local exploitation; the other subpopulation adopts CLS and focuses on global exploration. In the search process, these two subpopulations cooperate with each other through different information interaction mechanisms. The experimental results of the benchmark functions and a real-world engineering problem verify the validity of the proposed TSLPSO. The statistical results show that due to the dual population mechanism, TSLPSO performs better significantly on most test functions than the reference PSO variants, with obvious improvements of the convergence speed, accuracy and reliability. There is an interesting phenomenon that CEC2014 test suite is very difficult for swarm-based intelligent optimization algorithm. Even for some well-known PSO variants, the final results are much worse than L-SHADE, the winner algorithm of CEC2014. There must be some underlying mechanisms worth of a detail investigation in the future, except for the theorem of no free lunch. Integrating population size reduction and dynamical subpopulations size adjustment may play important role conquer complicated problems, such as the benchmark problems with rotation and shift in CEC 2014. Moreover, the importance of large-scale black box optimization is becoming more and more obvious. Hence, the investigation of the proposed algorithm on large-scale black box optimization problems is an interesting and challenging direction in the future.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61572228, 61503150 and 61572226), the Key Development Project of Jilin Province (20180201045GX and 20180201067GX), the Premier Discipline Enhancement Scheme from Zhuhai Government (2015YXXK02), and the Premier Key-Discipline Enhancement Scheme from Guangdong Government (2016GDYSZDXK036).

Appendix 1. Benchmark functions list

	Function Name	Test Function	Search Range	$f_{\min}$	Accept
Unimodal	Sphere (F1)	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]^n	0	10^{-5}
	NoisyQuartic (F2)	$f_2(x) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0,1)$	[-1.28,1.28]^n	0	10^{-2}
	Schwefel 2.22 (F3)	$f_3(x) = \sum_{i=1}^{n} \vert x_i\vert + \prod_{i=1}^{n} \vert x_i\vert$	[-10,10]^n	0	10^{-5}
	Schwefel 1.2 (F4)	$f_4(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[-100, 100]^n	0	10^{-5}
	Rosenbrock (F5)	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-10, 10]^n	0	100
Multimodal	Schwefel (F6)	$f_6(x) = 418.982887273 \times n - \sum_{i=1}^{n} x_i sin(\sqrt{\vert x_i\vert})$	[-500, 500]^n	0	2000
	Rastrigin (F7)	$f_7(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	[-5, 5]^n	0	10^{-5}
	Noncontinuous Rastrigin (F8)	$f_8(x) = \sum_{i=1}^{n} (y_i^2 - 10cos(2\pi y_i) + 10) y_i = \begin{cases} x_i & \vert x_i\vert < 0.5 \\ roud(2x_i)/2 & \vert x_i\vert \geq 0.5 \end{cases}$	[-5, 5]^n	0	10^{-5}
	Ackley (F9)	$f_9(x) = -20\, exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	[-32, 32]^n	0	10^{-5}
	Griewank (F10)	$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]^n	0	10^{-5}
	Penalized1 (F11)	$f_{11}(x) = \frac{\pi}{n}\{10(sin(\pi y_1))^2 + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10(sin(\pi y_{i+1}))^2] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $where\ y_i = 1 + (x_i + 1)/4, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a, i = 1, \ldots, n \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[-50, 50]^n	0	10^{-5}
	Penalized2 (F12)	$f_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2[1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[-50, 50]^n	0	10^{-5}
	Weierstrass (F13)	$f_{13}(x) = \sum_{i=1}^{n}(\sum_{k=0}^{k\,max} [a^k cos(2\pi b^k \cdot (x_i + 0.5))]) - n \cdot \sum_{k=0}^{k\,max} [a^k\, cos(2\pi b^k \cdot 0.5)], a = 0.5, b = 3, kmax = 20$	[-0.5, 0.5]^n	0	10^{-5}
	Dminima (F14)	$f_{14}(x) = 78.332331408 + \sum_{i=1}^{n} (x_i^4 - 16x_i^2 + 5x_i)/n$	[-5, 5]^n	0	10^{-5}
	Rastrigin10 (F15)	$f_{15}(x) = \sum_{i=1}^{n} ((a_i x_i)^2 - 10cos(2\pi a_i x_i) + 10) where\ a_i = 10^{\frac{i-1}{n-1}}, i = 1, \ldots, n$	[-5, 5]^n	0	10
	Rastrigin100 (F16)	$f_{16}(x) = \sum_{i=1}^{n} ((a_i x_i)^2 - 10cos(2\pi a_i x_i) + 10) where\ a_i = 100^{\frac{i-1}{n-1}}, i = 1, \ldots, n$	[-5, 5]^n	0	10

Appendix 2. CEC2014 test suite

	Function Name	Search Range	$F(x^*)$
Unimodal Functions	F17: Rotated High Conditioned Elliptic Function	[-100,100]^n	100
	F18: Rotated Bent Cigar Function	[-100, 100]^n	100
	F19 Rotated Discus Function:	[-100, 100]^n	300
Simple Multimodal Functions	F20: Shifted and Rotated Rosenbrock's Function	[-100, 100]^n	400
	F21: Shifted and Rotated Ackley's Function	[-100, 100]^n	500
	F22: Shifted and Rotated Weierstrass Function	[-100, 100]^n	600
	F23: Shifted and Rotated Griewank's Function	[-100, 100]^n	700
	F24: Shifted Rastrigin's Function	[-100, 100]^n	800
	F25: Shifted and Rotated Rastrigin's Function	[-100, 100]^n	900
	F26: Shifted Schwefel's Function	[-100, 100]^n	1000
	F27 Shifted and Rotated Schwefel's Function 1:	[-100, 100]^n	1100

(*continued on next column*)

(*continued*)

	Function Name	Search Range	$F(x^*)$
	F28: Shifted and Rotated Katsuura Function	$[-100, 100]^n$	1200
	F29: Shifted and Rotated HappyCat Function	$[-100, 100]^n$	1300
	F30: Shifted and Rotated HGBat Function	$[-100, 100]^n$	1400
	F31: Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	$[-100, 100]^n$	1500
	F32: Shifted and Rotated Expanded Scaffer's F6 Function	$[-100, 100]^n$	1600
Hybrid Functions	F33: Hybrid Function 1 (N = 3)	$[-100, 100]^n$	1700
	F34: Hybrid Function 2 (N = 3)	$[-100, 100]^n$	1800
	F35: Hybrid Function 3 (N = 4)	$[-100, 100]^n$	1900
	F36: Hybrid Function 4 (N = 4)	$[-100, 100]^n$	2000
	F37: Hybrid Function 5 (N = 5)	$[-100, 100]^n$	2100
	F38: Hybrid Function 6 (N = 5)	$[-100, 100]^n$	2200
Composition Functions	F39: Composition Function 1 (N = 5)	$[-100, 100]^n$	2300
	F40: Composition Function 2 (N = 3)	$[-100, 100]^n$	2400
	F41: Composition Function 3 (N = 3)	$[-100, 100]^n$	2500
	F42: Composition Function 4 (N = 5)	$[-100, 100]^n$	2600
	F43: Composition Function 5 (N = 5)	$[-100, 100]^n$	2700
	F44: Composition Function 6 (N = 5)	$[-100, 100]^n$	2800
	F45: Composition Function 7 (N = 3)	$[-100, 100]^n$	2900
	F46: Composition Function 8 (N = 3)	$[-100, 100]^n$	3000

Appendix 3. Search results comparison for DLPSO, CLPSO and TSLPSO

Function	Criteria	TSLPSO	CLPSO	DLPSO
F1	Mean	0.00E+00	4.03E-41	2.55E-191
	Std	0.00E+00	8.17E-41	0.00E+00
	Rank	1	3	2
F2	Mean	2.49E-03	3.24E-03	1.93E-03
	Std	1.03E-03	7.31E-04	9.56E-04
	Rank	2	3	1
F3	Mean	1.19E-169	6.29E-26	1.98E-90
	Std	0.00E+00	3.46E-26	1.09E-89
	Rank	1	3	2
F4	Mean	2.50E-16	6.92E+01	1.07E-14
	Std	8.14E-16	2.67E+01	3.59E-14
	Rank	1	3	2
F5	Mean	1.73E+00	2.02E+01	1.56E+01
	Std	3.05E+00	1.35E+01	1.64E+01
	Rank	1	3	2
Ave Rank for F1-F5		**1.20**	**3.00**	**1.80**
F6	Mean	3.82E-04	3.82E-04	7.64E+00
	Std	2.62E-07	4.54E-13	2.96E+01
	Rank	2	1	3
F7	Mean	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1
F8	Mean	0.00E+00	3.23E-02	0.00E+00
	Std	0.00E+00	1.80E-01	0.00E+00
	Rank	1	2	1
F9	Mean	2.10E-14	1.24E-14	1.43E-14
	Std	3.22E-15	2.57E-15	3.49E-15
	Rank	3	1	2
F10	Mean	0.00E+00	0.00E+00	1.64E-02
	Std	0.00E+00	0.00E+00	3.93E-02
	Rank	1	1	2
F11	Mean	1.57E-32	1.57E-32	1.57E-32
	Std	5.56E-48	5.56E-48	5.56E-48
	Rank	1	1	1
F12	Mean	1.35E-32	1.35E-32	1.35E-32
	Std	2.78E-48	2.78E-48	2.78E-48
	Rank	1	1	1
F13	Mean	0.00E+00	0.00E+00	1.60E-15
	Std	0.00E+00	0.00E+00	3.02E-15
	Rank	1	1	2
F14	Mean	4.57E-10	4.57E-10	4.57E-10
	Std	8.96E-15	0.00E+00	7.07E-15
	Rank	2	1	3
F15	Mean	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1
F16	Mean	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1

(*continued on next column*)

(*continued*)

Function	Criteria	TSLPSO	CLPSO	DLPSO
Ave Rank for F6-F16		**1.36**	**1.09**	**1.64**
Ave Rank for F1-F16		**1.31**	**1.69**	**1.69**

Appendix 4. Comparisons on shifted functions

Fcnctions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
FS1	Mean	1.28E-14	2.66E-14	1.56E-14	1.33E-14	2.48E-14	1.51E-14	1.47E-14	0.00E+00
	SD	4.27E-15	7.99E-15	5.63E-15	3.55E-15	8.96E-15	3.55E-15	2.55E-15	0.00E+00
	Rank	2	8	6	3	7	5	4	1
FS2	Mean	2.84E-14	5.50E-14	3.12E-14	2.75E-14	5.59E-14	2.84E-14	3.12E-14	0.00E+00
	SD	1.04E-14	1.63E-14	8.54E-15	5.10E-15	1.55E-14	0.00E+00	1.34E-14	0.00E+00
	Rank	3	7	5	2	8	4	6	1
FS3	Mean	4.95E-14	9.54E-14	6.05E-14	5.68E-14	1.01E-13	5.68E-14	5.68E-14	0.00E+00
	SD	1.94E-14	3.07E-14	1.42E-14	0.00E+00	3.19E-14	0.00E+00	0.00E+00	0.00E+00
	Rank	2	7	6	3	8	4	5	1
FS4	Mean	2.81E+00	2.95E+00	2.30E+00	3.64E+01	5.42E+00	4.59E+01	5.00E+01	1.55E-12
	SD	8.77E+00	7.62E+00	7.17E+00	8.04E+00	1.43E+01	4.02E+01	3.59E+01	3.18E-12
	Rank	3	4	2	6	5	7	8	1
FS5	Mean	2.00E-01	1.97E+01	2.00E+01	2.07E+01	6.76E-02	2.05E+01	2.79E-01	4.21E-05
	SD	1.22E-01	7.00E-01	1.99E-04	7.95E-02	1.15E-01	1.90E-01	4.60E-01	4.43E-05
	Rank	3	5	6	8	2	7	4	1
FS6	Mean	1.43E-13	2.02E-13	3.17E-02	9.17E-14	2.27E-13	4.04E-01	4.39E-02	0.00E+00
	SD	1.14E-13	4.83E-14	1.77E-01	4.57E-14	5.08E-14	8.93E-01	4.97E-02	0.00E+00
	Rank	3	4	6	2	5	8	7	1
FS7	Mean	1.10E-13	9.54E-04	6.34E-03	4.37E-03	1.51E-02	2.16E-02	1.28E-02	0.00E+00
	SD	6.22E-14	3.03E-03	1.17E-02	7.44E-03	2.76E-02	2.54E-02	1.37E-02	0.00E+00
	Rank	2	3	5	4	7	8	6	1
FS8	Mean	6.74E-08	2.89E-01	5.81E+00	2.60E+01	2.31E-13	2.29E+01	1.07E-12	1.76E-13
	SD	3.75E-07	5.26E-01	2.37E+00	1.18E+01	6.87E-14	5.52E+00	2.20E-12	6.35E-14
	Rank	4	5	6	8	2	7	3	1
FS9	Mean	3.18E-01	3.71E+01	1.49E+02	1.07E+03	8.71E-01	4.77E+02	1.98E-01	1.61E-02
	SD	2.17E-01	6.94E+01	1.09E+02	3.51E+02	7.42E-01	2.16E+02	6.24E-02	1.37E-02
	Rank	3	5	6	8	4	7	2	1
FS10	Mean	5.68E-13	1.09E-05	2.90E-13	5.67E-01	4.91E-13	2.39E-04	4.75E-12	9.97E-06
	SD	1.02E-13	1.83E-05	5.75E-14	1.38E-01	8.50E-14	6.13E-04	2.45E-11	2.48E-06
	Rank	3	6	1	8	2	7	4	5
FS11	Mean	2.04E-01	2.69E-01	2.34E-01	3.24E-01	2.28E-01	4.76E-01	3.60E-01	1.19E-01
	SD	5.17E-02	7.16E-02	5.43E-02	5.09E-02	6.37E-02	9.77E-02	1.04E-01	1.91E-02
	Rank	2	5	4	6	3	8	7	1
FS12	Mean	2.33E-01	2.54E-01	3.02E-01	2.65E-01	2.53E-01	2.66E-01	3.37E-01	2.17E-01
	SD	4.28E-02	3.19E-02	1.25E-01	4.99E-02	7.49E-02	3.96E-02	8.84E-02	2.99E-02
	Rank	2	4	7	5	3	6	8	1
FS13	Mean	1.43E+00	7.95E-01	1.53E+00	6.49E+00	1.41E+00	3.77E+00	1.65E+00	1.18E+00
	SD	3.82E-01	2.94E-01	5.45E-01	1.65E+00	4.39E-01	1.08E+00	5.17E-01	1.02E-01
	Rank	4	1	5	8	3	7	6	2
FS14	Mean	8.74E-01	5.82E-01	2.50E+00	8.15E+00	6.38E-01	3.85E+00	6.52E-01	5.61E-01
	SD	4.85E-01	2.73E-01	6.86E-01	8.85E-01	3.70E-01	1.54E+00	3.64E-01	7.59E-02
	Rank	5	2	6	8	3	7	4	1
Ave Rank		2.93	4.71	5.07	5.64	4.43	6.57	5.29	1.36
Final Rank		2	4	5	6	3	8	7	1

Appendix 5. Comparisons on rotated functions

Fcnctions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
FR1	Mean	6.31E+05	2.92E+05	1.13E+06	4.20E+06	4.72E+05	1.68E+06	7.41E+04	1.56E-14
	SD	5.43E+05	2.24E+05	1.76E+06	2.34E+06	4.63E+05	1.87E+06	3.97E+04	6.61E-15
	Rank	5	3	6	8	4	7	2	1
FR2	Mean	5.31E+02	3.85E+01	3.58E+01	5.25E-01	4.91E+02	2.62E+00	1.09E+02	0.00E+00
	SD	7.06E+02	7.86E+01	3.95E+01	1.32E+00	9.23E+02	4.44E+00	1.24E+02	0.00E+00
	Rank	8	5	4	2	7	3	6	1
FR3	Mean	2.60E+01	4.12E+02	1.05E+03	2.09E+01	1.19E-02	3.22E+01	5.14E+02	0.00E+00
	SD	4.64E+01	4.68E+02	1.32E+03	3.52E+01	5.39E-02	4.09E+01	7.27E+02	0.00E+00
	Rank	4	6	8	3	2	5	7	1
FR4	Mean	2.25E+01	2.24E+01	2.49E+01	2.28E+01	2.57E+01	3.44E+01	1.34E+01	5.22E-06
	SD	3.52E+00	1.50E+00	1.33E+01	9.58E-01	1.62E+01	2.02E+01	1.52E+01	1.77E-05
	Rank	4	3	6	5	7	8	2	1
FR5	Mean	2.00E+01	2.00E+01	2.02E+01	2.09E+01	2.00E+01	2.09E+01	2.00E+01	2.01E+01
	SD	2.04E-02	4.25E-03	2.66E-01	6.31E-02	3.94E-03	8.43E-02	1.05E-02	3.25E-02
	Rank	4	3	6	8	2	7	1	5
FR6	Mean	2.42E+00	6.53E+00	4.74E+00	1.64E+00	7.97E+00	1.09E+01	7.43E+00	0.00E+00
	SD	1.79E+00	2.00E+00	2.91E+00	9.67E-01	3.64E+00	3.11E+00	1.98E+00	0.00E+00

(continued on next column)

(*continued*)

Fcnctions	Criteria	TSLPSO	HCLPSO	OLPSO	CLPSO	DLPSO	PSO	GL-PSO	L-SHADE
	Rank	3	5	4	2	7	8	6	1
FR7	Mean	8.31E-12	5.50E-05	3.33E-03	6.70E-03	1.23E-02	1.69E-02	9.04E-03	0.00E+00
	SD	4.21E-11	1.70E-04	7.70E-03	9.05E-03	2.04E-02	1.23E-02	1.28E-02	0.00E+00
	Rank	2	3	4	5	7	8	6	1
FR8	Mean	3.34E+01	4.75E+01	4.95E+01	6.57E+01	6.07E+01	8.32E+01	4.31E+01	6.42E+00
	SD	8.64E+00	9.71E+00	1.57E+01	2.77E+01	1.57E+01	1.62E+01	1.46E+01	9.82E-01
	Rank	2	4	5	7	6	8	3	1
FR9	Mean	1.80E+03	1.73E+03	2.41E+03	4.02E+03	2.06E+03	2.66E+03	2.21E+03	1.04E+03
	SD	4.48E+02	3.51E+02	7.04E+02	5.25E+02	3.58E+02	7.72E+02	4.88E+02	2.10E+02
	Rank	3	2	6	8	4	7	5	1
FR10	Mean	9.20E-02	7.03E-02	1.75E-01	1.82E+00	6.67E-02	1.41E+00	9.81E-02	1.14E-01
	SD	3.11E-02	1.87E-02	2.20E-01	3.48E-01	2.60E-02	5.61E-01	4.40E-02	1.88E-02
	Rank	3	2	6	8	1	7	4	5
FR11	Mean	2.24E-01	2.85E-01	2.72E-01	3.43E-01	2.29E-01	5.35E-01	4.28E-01	1.19E-01
	SD	4.69E-02	6.97E-02	5.97E-02	5.00E-02	6.35E-02	1.13E-01	9.40E-02	1.84E-02
	Rank	2	5	4	6	3	8	7	1
FR12	Mean	2.67E-01	2.76E-01	4.41E-01	3.82E-01	3.03E-01	5.20E-01	3.73E-01	2.43E-01
	SD	8.27E-02	3.72E-02	1.92E-01	1.89E-01	1.52E-01	2.64E-01	1.43E-01	3.61E-02
	Rank	2	3	7	6	4	8	5	1
FR13	Mean	4.74E+00	4.96E+00	4.63E+00	9.96E+00	5.53E+00	7.57E+00	6.39E+00	2.27E+00
	SD	1.33E+00	1.40E+00	2.51E+00	2.25E+00	2.07E+00	3.18E+00	1.91E+00	1.99E-01
	Rank	3	4	2	8	5	7	6	1
FR14	Mean	9.22E+00	8.80E+00	1.13E+01	1.14E+01	9.35E+00	1.09E+01	9.31E+00	7.86E+00
	SD	9.55E-01	6.22E-01	8.11E-01	3.94E-01	6.77E-01	8.74E-01	6.74E-01	6.03E-01
	Rank	3	2	7	8	5	6	4	1
Ave Rank		3.43	3.57	5.36	6.00	4.57	6.93	4.57	1.57
Final Rank		2	3	6	6	4	7	5	1

References

[1] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. 294 (Feb. 2015) 182–202.

[2] X.F. Liu, Z.H. Zhan, J.D. Deng, Y. Li, T. Gu, J. Zhang, An energy efficient ant colony system for virtual machine placement in cloud computing, IEEE Trans. Evol. Comput. 22 (1) (2018) 113–128.

[3] Z.G. Chen, et al., Multiobjective cloud workflow scheduling: a multiple populations ant colony system approach, IEEE Trans. Cybern. PP (99) (2018) 1–15.

[4] X.F. Liu, Z.H. Zhan, Y. Gao, J. Zhang, S. Kwong, J. Zhang, Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization, IEEE Trans. Evol. Comput. (2018), https://doi.org/10.1109/TEVC.2018.2875430.

[5] Y. Li, Z.H. Zhan, S. Lin, J. Zhang, X. Luo, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, Inf. Sci. 293 (3) (2015) 370–382.

[6] 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. MHS '95, 1995, pp. 39–43.

[7] A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part I: background and development, Nat. Comput. 6 (4) (Dec. 2007) 467–484.

[8] A. Banks, J. Vincent, C. Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications, Nat. Comput. 7 (1) (Mar. 2008) 109–124.

[9] S. Cheng, H. Lu, X. Lei, Y. Shi, A quarter century of particle swarm optimization, Complex Intell. Syst. 4 (3) (Apr. 2018) 227–239.

[10] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., the 1998 IEEE International Conference on, 1998, pp. 69–73.

[11] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.

[12] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Trans. Evol. Comput. 8 (3) (Jun. 2004) 240–255.

[13] Z.-H. Zhan, J. Zhang, Y. Li, H.S.-H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybern. Part B Cybern. 39 (6) (Dec. 2009) 1362–1381.

[14] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), vol. 3, 1999, pp. 1931–1938.

[15] P.N. Suganthan, Particle swarm optimiser with neighbourhood operator, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, vol. 3, 1999, pp. 1958–1962.

[16] J. Kennedy, R. Mendes, "Population structure and particle swarm performance, in: Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02, vol. 2, 2002, pp. 1671–1676.

[17] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (3) (Jun. 2004) 204–210.

[18] K.E. Parsopoulos, M.N. Vrahatis, UPSO: a unified particle swarm optimization scheme, in: Int Conf Comput Methods Sci Eng, Ser Lecture Series on Computer & Computational Sciences Attica, Greece: Vsp International Science, vol. 1, 2004, pp. 868–873.

[19] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inf. Sci. 209 (Nov. 2012) 16–36.

[20] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (Jun. 2006) 281–295.

[21] A. Salman, M.G.H. Omran, M. Clerc, S. Alsharhan, Improving the performance of comprehensive learning particle swarm optimizer, J. Intell. Fuzzy Syst. 30 (2) (Feb. 2016) 735–746.

[22] H. Wu, et al., An improved comprehensive learning particle swarm optimization and its application to the semiautomatic design of antennas, IEEE Trans. Antenn. Propag. 57 (10) (Oct. 2009) 3018–3028.

[23] H. Huang, H. Qin, Z. Hao, A. Lim, Example-based learning particle swarm optimization for continuous optimization, Inf. Sci. 182 (1) (Jan. 2012) 125–138.

[24] X. Yu, X. Zhang, Enhanced comprehensive learning particle swarm optimization, Appl. Math. Comput. 242 (Sep. 2014) 265–276.

[25] Ş. Gülcü, H. Kodaz, A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization, Eng. Appl. Artif. Intell. 45 (Oct. 2015) 33–45.

[26] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (Oct. 2015) 11–24.

[27] Z.J. Wang, Z.H. Zhan, J. Zhang, An improved method for comprehensive learning particle swarm optimization, in: Computational Intelligence, 2015 IEEE Symposium, 2015, pp. 218–225.

[28] C. Li, S. Yang, T.T. Nguyen, A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (3) (Jun. 2012) 627–646.

[29] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Inf. Sci. 181 (20) (Oct. 2011) 4699–4714.

[30] W.H. Lim, N.A. Mat Isa, An adaptive two-layer particle swarm optimization with elitist learning strategy, Inf. Sci. 273 (Jul. 2014) 49–72.

[31] H. Ouyang, L. Gao, S. Li, X. Kong, Improved global-best-guided particle swarm optimization with learning operation for global optimization problems, Appl. Soft Comput. 52 (Mar. 2017) 987–1008.

[32] Z.-H. Zhan, J. Zhang, Y. Li, Y.-H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans. Evol. Comput. 15 (6) (Dec. 2011) 832–847.

[33] Y.J. Gong, et al., Genetic learning particle swarm optimization, IEEE Trans. Cybern. 46 (10) (Oct. 2016) 2277–2290.

[34] X. Shi, Y. Li, H. Li, R. Guan, L. Wang, Y. Liang, An integrated algorithm based on artificial bee colony and particle swarm optimization, in: Natural Computation (ICNC), 2010 Sixth International Conference on, vol. 5, 2010, pp. 2586–2590.

[35] M.S. Kıran, M. Gündüz, A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems, Appl. Soft Comput. 13 (4) (Apr. 2013) 2188–2203.

[36] W. Chun-Feng, L. Kui, S. Pei-Ping, Hybrid artificial bee colony algorithm and particle swarm search for global optimization, Math. Probl Eng. 2014 (2014) 1–8.

[37] Z. Li, W. Wang, Y. Yan, Z. Li, PS–ABC: a hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems, Expert Syst. Appl. 42 (22) (Dec. 2015) 8881–8895.

[38] M.A. Potter, K.A. De Jong, A cooperative coevolutionary approach to function optimization, in: Parallel Problem Solving from Nature — PPSN III, 1994, pp. 249–257.

[39] F. vandenBergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (Jun. 2004) 225–239.

[40] X. Li, X. Yao, Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms, in: 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 1546–1553.

[41] X. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Trans. Evol. Comput. 16 (2) (Apr. 2012) 210–224.

[42] X. Xia, L. Gui, Z.-H. Zhan, A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting, Appl. Soft Comput. 67 (Jun. 2018) 126–140.

[43] Shinn-Ying Ho, Hung-Sui Lin, Weei-Hurng Liauh, Shinn-Jang Ho, OPSO: orthogonal particle swarm optimization and its application to task assignment problems, IEEE Trans. Syst. Man Cybern. - Part Syst. Hum. 38 (2) (Mar. 2008) 288–298.

[44] K.E. Parsopoulos, M.N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (Jun. 2004) 211–224.

[45] O. Olorunda, A.P. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: 2008 IEEE Congress on Evolutionary Computation, 2008, pp. 1128–1134.

[46] Q. Cui, et al., Globally-optimal prediction-based adaptive mutation particle swarm optimization, Inf. Sci. 418–419 (Dec. 2017) 186–217.

[47] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, in: Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Rep. Nanyang Technol. Univ. Singap., 2013.

[48] Z.J. Wang, et al., Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, IEEE Trans. Evol. Comput. 22 (6) (Dec. 2018) 894–908.

[49] Z.H. Zhan, et al., Cloudde: a heterogeneous differential evolution algorithm and its distributed cloud version, IEEE Trans. Parallel Distr. Syst. 28 (3) (2017) 704–716.

[50] X.F. Liu, et al., Historical and heuristic-based adaptive differential evolution, in: IEEE Trans. Syst. Man Cybern. Syst., 2018, pp. 1–13.

[51] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 2014, pp. 1658–1665.

[52] P. N. Suganthan, N. Hansen, J. J. Liang, and K. Deb, "Problem Definitions and Evaluation Criteria for the Cec 2005 Special Session on Real-parameter Optimization," pp. 1–50.

[53] 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 Evol. Comput. 1 (1) (Mar. 2011) 3–18.

[54] M.L. Dukic, Z.S. Dobrosavljevic, A method of a spread-spectrum radar polyphase code design, IEEE J. Sel. Area. Commun. 8 (5) (Jun. 1990) 743–749.

[55] S. Gil-López, J.D. Ser, S. Salcedo-Sanz, Á.M. Pérez-Bellido, J.M. Cabero, J.A. Portilla-Figueras, A hybrid harmony search algorithm for the spread spectrum radar polyphase codes design problem, Expert Syst. Appl. 39 (12) (Sep. 2012) 11089–11093.

[56] W.H. Mow, Sequence design for Spread Spectrum, Chinese Univ. Press, Sha Tin, Hong Kong, 1995.

[57] N. Mladenović, J. Petrović, V. Kovačević-Vujčić, M. Čangalović, Solving spread spectrum radar polyphase code design problem by tabu search and variable neighbourhood search, Eur. J. Oper. Res. 151 (2) (Dec. 2003) 389–399.

[58] W.H. Lim, N.A. Mat Isa, Teaching and peer-learning particle swarm optimization, Appl. Soft Comput. 18 (May 2014) 39–58.