Information Sciences 578 (2021) 457–481

Contents lists available at ScienceDirect

Information Sciences

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

Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization

Xinmin Tao [*,1], Xiangke Li, Wei Chen, Tian Liang, Yetong Li, Jie Guo, Lin Qi

College of Engineering and Technology, Northeast Forestry University. Harbin, Heilongjiang 150040, China

A R T I C L E I N F O

Article history:
Received 19 December 2020
Received in revised form 22 May 2021
Accepted 2 July 2021
Available online 12 July 2021

Keywords:
Particle swarm optimization
Comprehensive learning
Elitism learning
Self-adaptive parameters updating

A B S T R A C T

Particle Swarm Optimization (PSO) is a simple yet efficient population based evolutionary algorithm, which has been widely applied to solve global optimization problems. Similar to other evolutionary algorithms (EAs), PSO algorithm still suffers from loss of diversity and slow convergence caused by the contradiction between exploration and exploitation abilities. In order to address the drawbacks, we propose Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization (SAHLPSO) to improve optimization performance by using exploration-role and exploitation-role learning strategies with self-adaptively updating parameters manner. The exploration-role learning strategies with external archive adopt comprehensive learning to generate the learning exemplars, which can effectively maintain population diversity and thus avoid premature. On the other hand, the exploitation-role learning strategies utilize elitism learning to generate the learning exemplars so as to improve convergence performance. Considering that the performance improvement of PSO depends on not only the adaptation of different learning strategies but also the reasonable parameter settings, we also develop a self-adaptive parameter updating manner based on success history information. This manner can automatically update the learning parameters to appropriate values and avoid a user's prior knowledge about the relationship between the parameter settings and the characteristics of optimization problems. In addition, the usage of the linear population size reduction is also helpful to well balance the exploration and exploitation along with the process of evolution. Simulation results on some classical and CEC2013 benchmark functions show that LSAHLPSO is better than other PSO variants or at least comparable to other state-of-the-art evolutionary algorithms.

1. Introduction

Particle swarm optimization (PSO), originally proposed by Eberhart and Kennedy [14] in 1995, has been shown to be a simple yet efficient evolutionary algorithm. Similar to other evolutionary algorithms (EAs), PSO is a population-based optimization algorithm. In PSO, each individual in the population is called a particle. During its evolutionary process, each particle in PSO adjusts its flying direction and updating size according to its own past experience and the information provided by its neighbors. Due to its cooperative behavior among particles during evolutionary process, PSO has achieved great

* Corresponding author.
E-mail address: taoxinmin@nefu.edu.cn (X. Tao).
[1] College of Engineering and Technology, Northeast Forestry University, #26 Hexing Road, Xiangfang District, Harbin, Heilongjiang 150040, China.

https://doi.org/10.1016/j.ins.2021.07.008

success in solving many optimization problems in real-world applications including neural network training [11,15], engineering optimization [30,36] and other industry problems [16,17].

It is noteworthy that the optimization performance of PSO is primarily dependent on two cornerstones, namely exploration and exploitation. Good exploration means good global search, which can diversify the populations to explore the whole solution space to find the promising region and simultaneously avoid premature. On the other hand, good exploitation indicates good local search, which can aggregate the populations to exploit the found promising region to refine the search for more precisely optimal solution. Specifically, paying more attention to exploration will result in a waste of evolution in searching over some inferior regions of the solution space and thus slow down the convergence speed. On the other hand, paying more attention to exploitation will carry the risk of diversity loss, especially during the early evolutionary stage and thus possibly get trapped into the local optimum. Therefore, to improve optimization efficiency of PSO, especially in solving complex optimization problems, it is of great importance to achieve a good balance between exploration and exploitation of the search space in PSO.

During the past few decades, various PSO versions based on different improvement strategies have been developed such as parameter-tuning [18], topology-modified [31,41], hybrid with other techniques [33] and learning modified strategy [23]. Among these PSO improvement strategies mentioned above, the application of different learning strategies has been shown to achieve promising results in enhancing optimization performance. For instance, comprehensive learning Particle Swarm Optimization (CLPSO) [19] has been proven to perform well on some complex multimodal optimization problems. This novel learning strategy can allow CLPSO to well preserve diversity of the population and thus avoid premature convergence. Afterwards, various modifications related to CLPSO were suggested to further improve the optimization performance of original CLPSO, which mainly include HCLSPO [34], TSLPSO [9] and CLPSO-LOT [22]. Several previous researches have demonstrated that the application of reasonable learning strategies in PSO has shown great potentials in balancing the exploration and exploitation.

Although the application of different learning strategies in PSO can obtain desired results, its performance is still quite dependent on the setting of learning control parameters. Different learning control parameters often show different characteristics since PSO is sensitive to parameter settings as other evolutionary algorithms. Indeed, there is no fixed parameter setting that is suitable for various problems or even at different evolution stages of a single problem. Therefore, in addition to the design of suitable learning strategies, it is still open and essential issue to develop self-adaptive mechanisms for adjusting learning control parameters during the search process to improve the performance of PSO.

Based on the above considerations, in this paper, we present a novel self-adaptive hybrid learning PSO referred to as SAHLPSO. In SAHLPSO, the exploration-role and exploitation-role learning strategies are applied to balance the exploration and exploitation during the evolutionary process. The exploration-role learning strategy uses comprehensive learning mechanism to generate learning exemplars, in which all other exploration-role particles' historical personal best information has chance to provide their own information for current exploration-role particle to learn. The main difference between them is that the exploration-role particles are responsible for global search and maintaining population diversity, while the exploitation-role particles are used to improve the convergence performance. It is worth mentioning that the exploration-role particles do not use the information of any exploitation-role particles' historical best information, which can avoid the premature of exploration-role particles caused by the rapidly transferred information by exploitation-role particles. On the contrary, the exploitation-role particles have chance to use the information provided by all particles including exploration-role ones, which can favor speeding up convergence rate. In our proposed learning strategies, there are two important learning control parameters: crossover probability and learning step size, which have a significant impact on the performance of the presented algorithm and are further discussed in the later Section. Unfortunately, the trial-and-error method for tuning the learning control parameters usually requires tedious optimization trials and often fails to achieve satisfactory results. Motivated by this consideration, we develop a self-adaptation mechanism for two crucial parameters tuning, which is based on successful parameter settings that were previously used found during the run. In addition, due to the special characteristics of crossover probability in the proposed algorithm, that is, a smaller value is effective for global search while a larger one can accelerate the convergence, we incorporated the restart mechanism into the crossover probability set. This would be helpful for the population to keep great diversity at early stage and improve convergence performance at later stage. In addition to the learning control parameters, the size of the population used by PSO also plays a significant role in controlling the rate of convergence. The optimal population size depends on complicated interaction of several factors, including learning strategies, and the values of other control parameters, as well as the problem to be solved. Thus, adaptive population resizing methods along with evolutionary process have attracted much attentions. To further improve the performance of the presented algorithm, we incorporated a population size self-tuning strategy, which continuously reduces the population size in accordance with a linear function during the run.

The presented SAHLPSO has been tested on CEC2013 benchmark test functions. The effectiveness and efficiency of the proposed SAHLPSO are compared with those of advanced peer algorithms, which include A Hybrid Particle Swarm Optimization Algorithm Using Adaptive Learning Strategy (ALPSO) [8], comprehensive learning Particle Swarm Optimization (CLPSO) [19], Heterogeneous comprehensive learning strategy (HCLPSO) [34], The fully informed particle swarm (FIPS) [38], Enhancing comprehensive learning particle swarm optimization with local optima topology (CLSPO-LOT) [22], Self Regulating Particle Swarm Optimization (SRPSO) [25], ensemble particle swarm optimizer (EPSO) [32], Particle swarm optimization base on dimensional learning strategy (TSLPSO) [9], and genetic learning particle swarm optimization (GL-PSO)[46]. In addition, to further verify the performance of SAHLPSO, we also compare the proposed algorithm with other four outstanding evolutionary algorithms, which mainly include Differential evolution algorithm with strategy adaptation for global numerical optimization (SaDE) [4], Differential evolution with composite trial vector generation strategies and control parameters (CoDE) [47], Completely derandomized self-adaptation in evolution strategies (CMA-ES) [35], and SHADE using Linear Population Size Reduction (LSHADE) [39]. The CEC2013 experimental results demonstrate that SAHLPSO outperforms other PSO variants and is highly competitive with other state-of-the-art EAs. In order to verify the significance of difference, the statistical comparative analysis in terms of MeanRank is also performed using the Holm test and Wilcoxon singed ranks test. The experimental results obtained from SAHLPSO and its statistical results in terms of MeanRank prove that SAHLPSO is significantly better than the selected other compared PSO variants and SaDE and CMA-ES in the accuracy of final solution at least with 95% confidence level. Compared to CoDE and LSHADE, SAHLPSO also obtains the comparable results and show great potentials. To summarize, the above statistical analysis results proved the effectiveness and efficiency of the two-role learning strategies and self-adaptive parameters updating manner applied in SAHLLPSO for performance improvement.

The main contributions of this work are summarized as follows:

In this paper, a self-adaptive two roles hybrid learning strategies-based particle swarm optimization (SAHLPSO) is proposed. In SAHLPSO, the exploration-role and exploitation-role learning strategies are developed to balance the exploration and exploitation during the evolutionary process. The exploration-role learning strategy with external archive adopts comprehensive learning to generate the learning exemplars to keep strong population diversity. On contrast, the exploitation-role learning strategy uses elitism learning mechanism to generate learning exemplars to accelerate convergence speed. During the evolutionary process, the exploration-role particles do not learn the information of any exploitation-role particles' historical best information, which can avoid the premature of exploration-role particles due to the rapidly transferred information from exploitation-role particles. On the contrary, the exploitation-role particles can use the information provided by exploration-role ones to favor speeding up convergence rate.

In addition to the application of new learning strategies, we also develop a self-adaptive parameters updating manner based on success history information to further balance the exploration and exploitation. This manner can automatically update the learning parameters to appropriate values according to previous evolution information during the whole evolutionary process and avoid a user's prior knowledge about the relationship between the parameter settings and the characteristics of optimization problems. In addition, the usage of the linear population size reduction is also helpful to well balance the exploration and exploitation along with the process of evolution.

The remainder of the paper is organized as follows: The canonical PSO algorithm and its related variants are briefly reviewed in Sections 2 and 3, respectively. Section 4 presents a novel SAHLPSO algorithm, with detailed explanations on its exploration-role, exploitation-role learning strategies and the self-adaptive parameter control tuning mechanism. Section 5 shows extensive experimental results and discussions. Finally, conclusions are drawn with future work in Section 6.

2. The canonical PSO

PSO is a population-based heuristic search algorithm. Firstly, PSO produces an initial population by randomly sampling several individuals (each individual is called a particle) from the search space. In PSO, each particle stands for a potential solution which has usually two crucial components: namely velocity and position. During the run, the velocity and position of a particle are updated according to its own and other particles' best experiences to find potential global best solution to a problem. More specifically, in the D dimensional solution space, the velocity V_i^d and position X_i^d of the dth dimension of the ith particle can be formulated as follows:

$$V_i^d = wV_i^d + c_1 rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 rand2_i^d * \left(gbest^d - X_i^d\right) \tag{1}$$

$$X_i^d = X_i^d + V_i^d \tag{2}$$

where $\boldsymbol{X}_i = (X_i^1, X_i^2, \cdots, X_i^D)$ is the position of the ith particle in the population, and $\boldsymbol{V}_i = (V_i^1, V_i^2, \cdots, V_i^D)$ represents the velocity of the ith particle. $\boldsymbol{pbest}_i = (pbest_i^1, pbest_i^2, \cdots, pbest_i^D)$ denotes the best historical position for the ith particle, and $\boldsymbol{gbest} = (gbest^1, gbest^2, \cdots, gbest^D)$ is the best position previously discovered by the whole population. c_1 and c_2 are two acceleration coefficients controlling the weights of learning terms that pull each particle toward $pbest$ and $gbest$, respectively, which are also called 'self-cognitive learning' and 'global learning'. $rand1_i^d$ and $rand2_i^d$ are two random numbers generated within the range [0,1]. w is the inertia weight, which is employed to control how much information of previous velocity is preserved for its updating. Without loss of generality, this research focuses mainly on the minimization optimization problems.

3. Related work

Since its emergence in 1995, PSO has attracted much attention from researchers. Despite its advantages, similar to other population-based algorithms, PSO still suffers from loss of diversity and slow convergence caused by the contradiction between exploration and exploitation abilities. To address the aforementioned limitations, numerous PSO improved variants have been reported, which can be mainly divided into four aspects: (1) parameter tuning strategy, (2) topology-modified strategy, (3) hybrid Strategy with other techniques, (4) learning modified strategy. Since our work mainly focuses on parameter tuning strategy and learning modified strategy. We only provide literature reviews on PSOs based on the above two strategies, for more details on the other two ones, please refer to [42].

3.1. Parameter tuning strategy

Numerous studies reveal that the appropriate strategy for parameter settings will be helpful to enhancing the performance of PSO. Generally, it seems to be desirable and reasonable for population-based evolutionary algorithms to pay more attention on its global search at the early stage and shift its focus on local search at the later stage during the run [5]. Based on such idea, Shi and Eberhart [45] first advised to set the inertia weight to start with 0.9 and then linearly decrease to 0.4 along with evolutionary process. Afterwards, other PSO variants with improved inertia weight were developed. For instance, Tanweer et al. [25] proposed a Self Regulating Particle Swarm Optimization (SRPSO) using self-adaptively regulating inertia weight setting strategy. Moreover, Javad et al. [1] presented a nonlinear strategy to set inertia weight, called flexible exponential inertia weight. Apart from inertia weight, acceleration coefficient is another widely used key parameter for PSO to balance global and local search abilities. As a pioneer on concerning the acceleration coefficients, Clerc and Kennedy [26] developed a constriction function related to two acceleration coefficients to improve the convergence rate. Afterwards, Chen et al. [21] proposed a hybrid particle swarm optimizer with sine cosine acceleration coefficients. Similarly, Ratnaweera et al. [2] incorporated nonlinear dynamic acceleration coefficients and a modified particle position updating rule in PSO. To utilize evolutionary information of particles, Marco et al. [27] exploited Fuzzy Logic to independently estimate inertia weight, acceleration coefficients, minimum and maximum velocity for each particle and thus developed a fuzzy self-tuning PSO algorithm. Empirical results show that it outperforms other PSO variants. Furthermore, Tian et al. [7] also proposed an improved PSO algorithm (MPSO) and successfully applied it to address standard image segmentation. In this algorithm, a logistic map is used to generate uniformed distributed particle and then a sigmoid-like inertia weight is formulated, which make MPSO adaptively apply the inertia weight between linearly decreasing and nonlinearly decreasing strategies to improve search efficiency and convergence rate. In addition, Xia et al. [43] developed a fitness-based multi-role PSO (FMPSO) algorithm with multi-role-based parameter mechanism. The particles in FMPSO perform three different roles according to their corresponding fitness values, and then different strategies for inertia weights and acceleration coefficients setting are assigned to different roles, which is helpful for the population to perform different search patterns.

3.2. Learning modified strategy

Recently, some researchers investigated the adoption of different learning strategies in PSO to improve the optimization performance. The main motivation is that different learning strategies may be suitable to balance exploration and exploitation during the run. Wu et al. [10] presented a superior solution-guided strategy-based particle swarm optimization (SSG-PSO) with an individual level mutation mechanism. Similarly, Lim et al. [40] applied an elitist learning strategy and orthogonal learning strategy to avoid premature convergence. In Ref. [12], the improved global-based guided particle swarm optimizer (IGPSO) is developed, which can well balance exploration and exploitation. Moreover, Loau et al. [24] proposed an orthogonal learning particle swarm optimization (OPSO) by adopting the orthogonal diagonalization process to update particles such that enhance population diversity. In Ref. [37], a social learning mechanism is incorporated into PSO. Inspired by committee-based active learning strategy, Wang et al. [13] proposed a novel surrogate-assisted particle swarm optimization and experimental results demonstrate its effectiveness in solving expensive optimization problems. In addition, Tanweer et al. [28,29] reported a series of improved PSO algorithms by the way of self-regulating inertia weight and self-perceiving the global search directions. To improve the performance of PSO, a genetic learning particle swarm optimization (GL-PSO) is proposed in Ref. [46]. To further enhance exploration capability of GL-PSO, Lin et al. [3] introduced ring topology to enlarge the potential global search capability. In order to solve multi-modal optimization problems, Liang et al. [19] proposed a comprehensive learning particle swarm optimizer, and extensive experimental results indicate that CLPSO significantly outperforms most of existing improved PSO variants, especially in dealing with multimodal optimization problems. Afterwards, some researchers developed several improved CLPSO variants to further balance exploration and exploitation. For instance, Nandar et al. [34] proposed a heterogeneous comprehensive learning particle swarm optimization (HCLPSO), where the whole swarm population is divided into one explorative sub population as well as one exploitative sub population, and the reported results show this strategy can well balance the exploration and exploitation. Based on the same idea, G. Xu et al. [9] proposed a two–swarm learning PSO algorithm (TSLPSO) by adopting a dimensional learning strategy (DLS) to generate an exemplar for all particles in one exploitative sub population. In addition to the application of self-regulating inertia weight as described above, SRPSO [25] also used self-perception on global search direction to find the promising solution and the final experimental results demonstrate its effectiveness. Zhang et al. [44] proposed a differential mutation based social learning PSO (DSPSO) where a differential mutation strategy was used to update the three best particles and the reported results show the usage of the strategy can significantly improve search performance in social learning PSO algorithm. Inspired by ensemble idea, N. Lynn et al. [32] introduced an ensemble of different particle swarm optimization algorithms called the ensemble particle swarm optimizer (EPSO) to solve real-parameter optimization problems. K. Zhang et al. [22] proposed a local optimal topology structure based on the comprehensive learning particle swarm optimization (CLSPO-LOT).

CLSPO-LOT first identified a local optimum and then constructed a new topology space. A learning exemplar is finally generated by randomly selecting from this space to update the whole particles. In our previous work, we presented a multiple scale self-adaptive cooperation mutation strategy-based particle swarm optimization (MSMPSO) where multi-scale mutation strategy and the updating learning strategy without inertia items are integrated to balance exploration and exploitation in Ref. [42]. By incorporating the bee-foraging search mechanism into PSO, Xu et al. [6] developed a novel bee-foraging learning PSO (BFL-PSO) where three different learning strategies are applied which include employed learning, onlooker learning and scout learning. Employed learning is similar to traditional PSO and onlooker learning is used to enhance exploitation search around the better solutions, while scout learning is adopted to reinitialize these stagnated solution. The proposed BFL-PSO algorithm is evaluated on CEC2014 benchmark functions to verify its outstanding performance.

Based on the above analysis, we can find that keeping good balance between global exploration and local exploitation during the whole optimization process is still vitally essential for PSO algorithms to achieve promising performance. The modification of learning strategy has been proven to be an effective method in tradeoff between the exploration and exploitation. As mentioned above, due to the application of comprehensive learning strategy (CLS), CLPSO has achieved great success on solving complex multimodal optimization problems. However, CLPSO cannot perform well on unimodal optimization issues as the author claimed. To address this limitation, various CLPSO modified versions have been developed as reported above. The existing literatures regarding CLPSO variants show that using multiple learning strategies can achieve the balance between global exploration and local exploitation and thus improve the accuracy of final solution. Therefore, we adopt self-adaptive two roles hybrid learning strategies: the exploration-role and exploitation-role learning strategies to further improve the performance of PSO in this paper. The exploration-role learning strategies adopt comprehensive learning to generate the learning exemplars to keep strong population diversity and thus avoid premature. The exploitation-role learning strategies utilize elitism learning to generate the learning exemplars to improve convergence performance. The interaction between particles with different learning strategies ensures the coevolution. In addition, considering that the balanced between exploration and exploitation also depends on control parameter configurations, we also develop a self-adaptive parameters updating manner to automatically tune the learning parameters to appropriate values based on the previous evolutionary information. The detailed information about the proposed algorithm is provided below.

4. Proposed approach

In this section, we propose a novel PSO, named SAHLPSO. SAHLPSO contains two main components: two-role learning strategy and self-adaptive parameters tuning mechanism. Next, the implementation of the above main components will be discussed in detail.

4.1. Traditional learning strategy or its variant

As mentioned previously, the application of different learning strategies plays a crucial role in improving the performance of PSO. In traditional PSO, a particle learns from its own history personal best and global best. However, previous researches have shown that such learning strategy tends to cause the particle oscillating between the two exemplars when they are located in opposite side of the candidate particle [19]. Therefore, researchers attempted to develop various learning strategies. In order to maintain population diversity, the following learning scheme is one of the most widely used ones in the literature

$$V_i^d = wV_i^d + c * rand_i^d * \left(e_i^d - X_i^d\right) \tag{3}$$

Different from the learning strategy used in traditional PSO, the above learning strategy applies a composite exemplar $\boldsymbol{E}_i = (\ e_i^1,\ \ e_i^2,\ \ \cdots,\ \ e_i^D\ )$ to guide the particle instead of the history personal best and global best. In fact, the velocity update strategy in Eq. (1) can be formulated into the above simplified one by regarding e_i^d as a linear combination of $pbest_i^d$ and $gbest_i^d$

$$e_i^d = \frac{c_1 rand1_i^d * pbest_i^d + c_2 rand2_i^d * gbest_i^d}{c_1 rand1_i^d + c_2 rand2_i^d} \tag{4}$$

Recall that most of the above mentioned PSO variants with different learning strategies attempted to construct a single composite exemplar $\boldsymbol{E}_i$ with different phenotype combination to attract the candidate particle. For instance, a full-informed PSO (FIPS) uses the linear combination of all neighbors' personal bests to generate the exemplar. CLPSO constructs this exemplar by setting e_i^d to be $pbest_i^d$ within pre-specified probability range and the dth dimension of another particle's personal best using a tournament selection. GL-PSO applies genetic operators to generate high-quality exemplar for the candidate particle to learn.

4.2. Two-role learning strategy

Based on the above description, the exemplar $\boldsymbol{E}_i$ plays an important role in determining the search trajectory of the candidate particle. To be specific, the more diverse the distribution of the set of the exemplar $\boldsymbol{E}_i$ in the entire search space for each particle is, the stronger exploration capability the population possesses. On the contrary, the more aggregative the distribution of the set of the exemplar $\boldsymbol{E}_i$ in the superior search space for each particle is, the stronger exploitation capability the population possesses. In order to balance the exploration and exploitation, a two-role learning strategy is proposed in this paper. The two-role learning strategy contains two components: the exploration-role learning strategy with external archive and the exploitation-role learning strategy. The exploration-role learning strategy mainly focuses on the global search while the exploitation-role learning strategy pays more attention on the local search. The detailed implementation of the two-role learning strategy is presented in the following.

Motivated by the success of FIPS and CLPSO, the generation of the exemplar $\boldsymbol{E}_i$ should learn from all particles' history personal best information rather than the linear combination of its own personal best and the global best, which will favor maintaining population diversity and discourage premature convergence of PSO. Therefore, in the proposed algorithm, the exploration-role learning strategy also uses other particles' history personal best information to generate the exemplar. Similar to the CLPSO, we integrate the information of other particles' history personal best information by randomly picking two individuals in the history personal best set to undergo dimensional crossover as in traditional GA. For each particle ***i***, the dimensional crossover operation is performed on randomly selected two individuals in the history personal best set $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ to generate a donor vector $\boldsymbol{O}_i = [o_i^1, o_i^2, \cdots, o_i^D]$:

$$o_{id} = \begin{cases} p_m^d & iff(\boldsymbol{P}_m) < f(\boldsymbol{P}_n) \\ p_n^d & otherwise \end{cases} \tag{5}$$

where f represents the fitness function. From Eq. (5), it can be found that the generated donor vector $\boldsymbol{O}_i$ is likely to have some dimensions from its own history personal best and the other dimensions from other particles' history personal best or even global best. Compared to Eq. (4), this will be more helpful to maintain the generated donor vectors diverse.

Although the generated donor vector has better diversity than a linear combination of the own personal best and the global best, the generated donor vector still has the same biases towards superior solutions due to elitist preservation scheme indicated in Eq. (5). This bias is likely to make the learning exemplars more aggregative and thus cause premature at early stage. In order to more diversify the learning exemplars, the generated donor vector $\boldsymbol{O}_i$then undergoes the crossover with its corresponding element $\boldsymbol{A}_i$ in the external archive $\boldsymbol{A}$ with a probability bounded by crossover probability (cr_i) again. Specifically, for each dimension d, a random number$crd \in [0,1]$is generated, and then, if crd is smaller than cr_i, this dimension of $\boldsymbol{E}_i$ is set to be the corresponding dimension of $\boldsymbol{O}_i$, otherwise $\boldsymbol{A}_i$

$$e_{id} = \begin{cases} o_i^d & ifcrd < cr_i \\ a_i^d & otherwise \end{cases} \tag{6}$$

The purpose of introducing the second crossover operation is that it can bring in diversity of the learning exemplars and make them reach more exploratory search space. In fact, like the mutation of GA, the second crossover operation can discourage the premature convergence of PSO due to the rapidly learning from $\boldsymbol{O}_i$. In this paper, the external archive is composed of the inferior historical personal bests. The archive provides the previous information about the progress direction and has capabilities to improve the diversity of the generated learning exemplars. Different from CLPSO, which uses the current $\boldsymbol{pbest}_i$ to undergo the second crossover operation, previous inferior $\boldsymbol{pbest}_i$ can provide more exploratory search space since the current $\boldsymbol{pbest}_i$ is likely to have already aggregated within the local optimum during the run.

In order to illustrate the mechanism of the exploration-role learning strategy with external archive, we showed a 2-D search space in Fig. 1. For the particle $\boldsymbol{X}_i$, $\boldsymbol{Pbest}_{m1}$ and $\boldsymbol{Pbest}_{n1}$ are selected to generate the two candidate X-dimensional values. Since the fitness value of $\boldsymbol{Pbest}_{m1}$ is better than that of $P\boldsymbol{best}_{n1}$, the X-dimensional value of $\boldsymbol{O}_i$ is set to be that of $\boldsymbol{Pbest}_{m1}$. Based on similar reason, the Y-dimensional value of $\boldsymbol{O}_i$ is set to be that of $\boldsymbol{Pbest}_{m2}$. Due to the preservation of elitist, the generated donor vector $\boldsymbol{O}_i$ is generally located in the vicinity of relatively better solutions. In order to enhance the diversity of the generated exemplar, we applied the second crossover with the inferior personal bests maintained in the external archive according to the given crossover probability. This is because the current personal bests tend to be trapped into the vicinity of relatively better solutions as $\boldsymbol{O}_i$ during the evolutionary process. As shown in Fig. 1, $\boldsymbol{Pbest}_i$ is located around the local optimum as other particles' personal bests, which can result in the failure of the generated exemplar $\boldsymbol{E}_\mathrm{i}$ to explore wider search space although the second crossover is performed between $\boldsymbol{O}_i$ and $\boldsymbol{Pbest}_i$ to improve the diversity of $\boldsymbol{E}_\mathrm{i}$. On the contrary, the second crossover between $\boldsymbol{O}_i$ and $\boldsymbol{A}_i$ would be more likely to diversify $\boldsymbol{E}_\mathrm{i}$ than $\boldsymbol{Pbest}_i$ due to the relatively far distance of $\boldsymbol{A}_i$ from better solutions. Certainly, the crossover probability and other learning control parameters also play the crucial role in balancing exploration and exploitation, which will be discussed in detailed in the next Section.

From the above illustration, we can find that the exploration-role learning strategy can enhance the diversity of the generated exemplar by means of using the inferior personal best information. However, the enhancement of diversity of the generated exemplars $\boldsymbol{E}_i$ tends to take the risk of deteriorating exploitation capability, thus slowing down the convergence rate. To well balance exploration and exploitation, we developed another exploitation-role learning strategy.

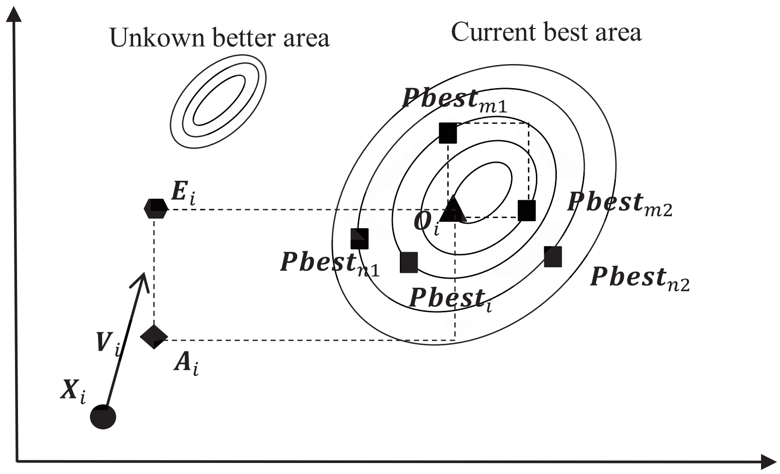

Fig. 1. Illustration of an exploration-role learning scheme with external archive in 2-D space.

Different from the exploration-role learning strategy, in the exploitation-role learning strategy, for each particle $\boldsymbol{X}_i$, a donor vector $\boldsymbol{O}_i = [o_i^1, o_i^2, \cdots, o_i^D]$ is generated from a randomly selected individual from only the top $p * 100$ $p \in (0, 1)$ history personal bests. In addition, the second cross operator is performed between $\boldsymbol{O}_i$ and its own current personal best according to the given crossover probability.

$$q_i^d = \begin{cases} o_i^d & \text{if } crd < cr_i \\ pbest_i^d & \text{otherwise} \end{cases} \tag{7}$$

In order to further increase the effect of the global best on the particles performing exploitation-role learning strategy, the final e_i^d is formed by linear combination of q_i^d and $gbest_i^d$

$$e_i^d = rand1_i^d * q_i^d + (1 - rand1_i^d) * gbest_i^d \tag{8}$$

The illustration of the exploitation-role learning strategy is shown in Fig. 2. Compared with the exploration-role learning strategy with external archive shown in Fig. 1, the generated exemplar $\boldsymbol{E}_i$ by the exploitation-role learning strategy for the particle $\boldsymbol{X}_i$ is more biased toward the direction of potential global optimum (possibly a local minimum).

It is worth mentioning that for each particle performing exploration-role learning strategy, $m \in \{1, 2, \cdots, LG\}$ is randomly selected from the set of the particles performing exploration-role learning strategy. LG denotes the size of the set. That is, the particles performing exploration-role learning strategy only learn from the history personal best information provided by other particles performing exploration-role learning strategy. The main purpose is to avoid the premature caused by the rapid information transform from the particle performing exploitation-role learning strategy. Generally, we set $LG = Lg * M$, $Lg \in (0, 1)$ where M is the size of the whole population. Different from the particles performing exploration-role learning strategy, the particles performing the exploitation-role learning strategy learn from all particles' history personal bests no matter performing exploration-role learning strategy or exploitation-role learning strategy. This can make the particles performing the exploitation-role learning strategy move rapidly towards directions of the global best and thus accelerate convergence rate.

4.3. self-adaptively tuning parameter settings

As mentioned above, in addition to the application of different learning strategies, the learning control parameters related to them also play a significant role on balancing exploration and exploitation. In the aforementioned two-role learning strategy, there are two key learning control parameters: crossover probability and learning step size. The crossover probability determines how much information of the exemplar is inherited from the donor vector $\boldsymbol{O}_i$, which explicitly controls the trade-off between the exploration and exploitation as shown in Eq. (5) or Eq. (6). Learning step size is another key parameter which indicates how many generations the particle will learn from the current exemplar and then generate a novel exemplar. The learning step size can implicitly balance the exploration and exploitation. As we know, the learning control parameter settings highly depend on the problem under consideration. In the paper, motivated by LSHADE [39], which is considered as one of the most powerful evolutionary algorithms, we developed a novel mechanism of self-adaptive parameter settings for crossover probability and learning step size during the evolution. The implementation about the mechanism for self-adaptively tuning the two main learning control parameters is discussed in detail in the following.

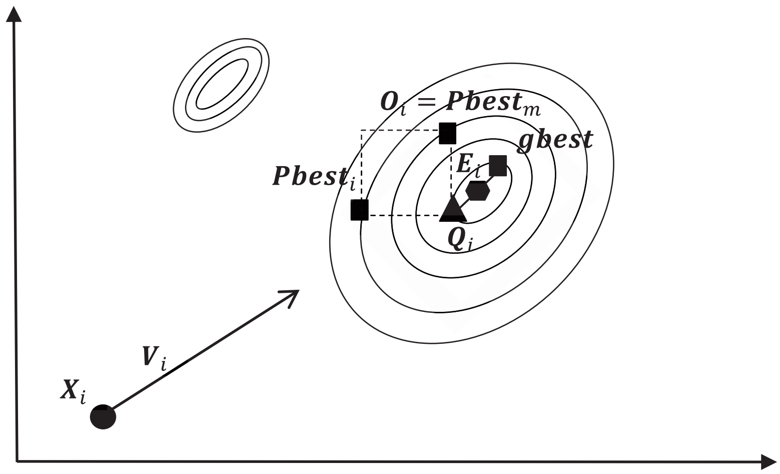

Fig. 2. Illustration of an exploitation-role learning scheme in 2-D space.

4.3.1. Cross-over probability tuning with re-start

Generally speaking, the greater crossover probability will encourage exploitation capability and thus speed up convergence rate. On the contrary, the smaller crossover probability will favor exploration capability and thus avoid premature. Obviously, the choice of crossover probability is sensitive to problems with different characteristics. The proper choice of crossover probability can lead to successful optimization performance while an improper choice may deteriorate the performance. In addition, the crossover probability values should be set to relatively small value at early stage, which is beneficial to keeping population diversity while should be set to relatively great value at later stage, which can be biased towards local search. Motivated by these above considerations, we introduced a novel crossover probability tuning mechanism in the proposed algorithm.

As shown in Table 1, we maintain a candidate set $M_{CR}$with H entries for the crossover probability and its corresponding successful rate performing previously well in the past within a fixed number of previous generations hereby named learning period (LP). The probabilities with respect to each entry in M_{CR} are initialized as $1/H_{CR}$, i.e., all M_{CR} entries have the equal probability to be chosen. At each generation, after evaluating all the generated particles, the number of the particles which selects kth M_{CR} entry as its cr_i is recorded as $nf_{CRk,G}$ and the number of the particles with the selected kth M_{CR} entry as its cr_i that can successfully update its corresponding current personal best is also recorded as $ns_{CRk,G}$. After the initial LP generations, the probabilities of choosing different entries in M_{CR} will be updated at every subsequent LP generations based on the recorded success memory information.

$$P_{CR,k} = \frac{S_{CRk,G}}{\sum_{k=1}^{H_{CR}} S_{CRk,G}} \tag{9}$$

where

$$S_{CRk,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{CRk,G}}{\sum_{g=G-LP}^{G-1} nf_{CRk,G}} + \varepsilon, k \in \{1, 2, \cdots, H_{CR}, G > LP\} \tag{10}$$

where $S_{CRk,G}$ represents the success rate of the particles generated by selecting kth M_{CR} entry as its cr_i and successfully updating its corresponding personal best within the previous LP generations. The small constant value ε is used to avoid the possible null success rates, by default 0.001. To guarantee that the probabilities of choosing M_{CR} are always summed to 1, we further divide $S_{CRk,G}$ by $\sum_{k=1}^{H_{CR}} S_{CRk,G}$. Obviously, the larger the success rate for the kth M_{CR} entry within the previous LP generations is, the larger the probability of applying it as the crossover probability at the current generation is. Here, we employ stochastic universal selection method to select one M_{CR} entry as the crossover probability cr_i for each particle in the current population Table 2.

In order to further ensure the diversity at the early evolutionary stage and convergence at the later evolutionary stage, we apply restart strategy to update M_{CR}. More specifically, if no successful particles by selecting kth M_{CR} entry as its cr_i according to its corresponding probability are recorded, i.e., $\sum_{k=1}^{H_{CR}} ns_{CRk,G} = 0$, we update M_{CR} by increasing an additional entry and set H_{CR} to be $H_{CR} + 1$. Correspondingly, the initial choosing probabilities are set equally to be $1/H_{CR}$. Certainly, the additional entry is larger than the all entries previous saved in M_{CR}, which can make the particles have chance to select the greater M_{CR} entry as its cr_i and thus improve the convergence rate along with the running of evolution.

Table 1
The candidate set M_{CR} and its historical memory of successful.M_{CR}

Index	1	2	···	H_{CR}-1	H_{CR}
M_{CR}	$M_{CR,1}$	$M_{CR,2}$	···	$M_{CR,H_{CR}-1}$	$M_{CR,H_{CR}}$
P_{CR}	$P_{CR,1}$	$P_{CR,2}$	···	$P_{CR,H_{CR}-1}$	$P_{CR,H_{CR}}$

Table 2
The candidate set M_{LS} and its historical memory of successful M_{LS}.

Index	1	2	···	H_{LS}-1	H_{LS}
M_{LS}	$M_{LS,1}$	$M_{LS,2}$	···	$M_{LS,H-1}$	$M_{LS,H}$
P_{LS}	$P_{LS,1}$	$P_{LS,2}$	···	$P_{LS,H-1}$	$P_{LS,H}$

4.3.2. Learning step size tuning

Learning step size indicates how many generations the particle learns from the current exemplar and then updates the new exemplar. The smaller value of learning step size means that the particle only learns from the current exemplar for few generations or even only once, which can result in loss of explorations and even oscillate due to frequent change of the exemplar. On the other hand, the greater value of learning step size suggests the stronger exploration capability but can tend to risk losing exploitation and thus slow down the convergence rate. Consequently, we also developed a self-adaptively tuning mechanism for the learning step size based on success-history information.

Like the crossover probability, we also maintain a candidate set M_{LS} and its corresponding successful rate of performing previously well in the past within LP for the learning step size, as shown in Table 3. Similarly, the probabilities related to each entry in M_{LS} are initialized as $1/H_{LS}$, i.e., all M_{LS} entries have the equal probability to be chosen. At each generation, after evaluating all the generated particles, the number of the particles which select kth M_{LS} entry as their ls_i is recorded as $nf_{LSk,G}$ and the number of the particles with the selected kth M_{LS} entry as their ls_i that can successfully update its corresponding current personal best is also recorded as $ns_{LSk,G}$. After the initial LP generations, the probabilities of choosing different entries in M_{LS} will be updated at every subsequent LP generations based on the recorded success memory information.

$$P_{LS,k} = \frac{S_{LSk,G}}{\sum_{k=1}^{H_{LS}} S_{LSk,G}} \tag{11}$$

where

$$S_{LSk,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{LSk,G}}{\sum_{g=G-LP}^{G-1} nf_{LSk,G}} + \varepsilon, k \in \{1, 2, \cdots, H_{LS}, G > \mathrm{LP}\} \tag{12}$$

where $S_{LSk,G}$ represents the success rate of the particles generated by selecting kth M_{LS} entry as its ls_i and successfully updating its corresponding personal best within the previous LP generations. The small constant value ε is used to avoid the possible null success rates. To guarantee that the probabilities of choosing M_{LS} are always summed to 1, we further divide $S_{LSk,G}$ by $\sum_{k=1}^{LS} S_{LSk,G}$. Obviously, the larger the success rate for the kth M_{LS} entry within the previous LP generations is, the larger the probability of applying it as the learning step size at the current generation is. Similarly, we employ stochastic universal selection method to select one M_{LS} entry as the learning step size ls_i for each particle in the current population.

4.3.3. Population size self-tuning strategy

As mentioned in Section I, adaptive population resizing strategy has been shown to be greatly effective in improving EA performance. In order to further improve the performance of the proposed algorithm, we incorporate a population size reduction method [39] for dynamically resizing the population during the evolutionary process. Let the population size at the beginning of the run be M, and the population at the end of the run be M_{min}. After each generation G, the population size in the next generation, M_{G+1}, is computed according to the following formula.

$$M_{G+1} = \mathrm{round}\left[\left(\frac{M_{min} - M}{MAXNFE}\right) * NFE + M\right] \tag{13}$$

where M_{min} is set to be the smallest possible value such that the evolutionary operators can at least be applied. In the case of the proposed algorithm, $M_{min} = 4$ because the two-role learning strategy shown in Eqs. (6) and (7) requires 4 individuals. NFE denotes the current number of fitness evaluations, whenever $M_{G+1} < M_G$, $M_G - M_{G+1}$ worst-ranking particles with respect to their corresponding personal bests are deleted from the population. It is worth noting that in order to well balance the exploration and exploitation, the deletion operations are performed iteratively.

Table 3
Description of Parameters settings.

Algorithm	Parameters settings	Reference
ALPSO	$\omega = 0.4\ 0.9, c1 = c2 = 2$	[8]
CLPSO	$\omega = 0.9\ 0.4, c1 = c2 = 1.49445, \mathrm{m} = 7$	[19]
HCLPSO	$\omega = 0.2\ 0.99, c1 = 2.5\ 0.5, c2 = 0.5\ 2.5, c = 3\ 1.5$	[34]
SRPSO	$w_I = 1.05, w_F = 0.5, c_1 = c_2 = 1.49445$	[25]
EPSO	$N_1 = 15, N_2 = 25, strategyns = 5, learningperiodLR = 50$	[32]
FIPS	$\chi = 0.729, \varphi = 4.1$	[38]
CLPSO-LOT	$\omega = 0.9\ 0.4, c1 = c2 = 1.49445, \mathrm{N} = 40, m = 7, a = 0.2, g = 30$	[22]
TSLPSO	$\omega = 0.9\ 0.4, c1 = c2 = 1.5, c3 = 0.5\ 2.5$	[9]
GL-PSO	$\chi = 0.7298, c = 1.49618, pm = 0.01, sg = 7$	[46]
SaDE	}rand/1/bin}, ″rand-to-best/2/bin}, }rand/2/bin}, }current-to-rand/1}$[F\varepsilon N(0.5, 0.3), C_r \varepsilon NCR_m, 0.1]$	[4]
CoDE	}rand/1/bin}, ″rand/2/bin}, ″current-to-rand/1}, $[F = 1.0, C_r = 0.1], [F = 1.0, C_r = 0.9], [F = 0.8, C_r = 0.2]$	[45]
*CMA-ES	–	[35]
LSHADE	$\mathrm{r}^{Ninit} = \{15, 16, \cdots, 24, 25\}, \mathrm{r}^{arc} = \{1.0, 1.1, \cdots 2.9, 3.0\}\mathrm{p} = \{0.05, 0.06, \cdots, 0.14, 0.15\}, \mathrm{H} = \{2, 3, \cdots, 9, 10\}$	[39]
SAHLPSO	$Lg = 0.3p = 0.2, \mathrm{c1} = \mathrm{c2} = 1.49445$	
	$Lg = 0.2, LP = 20, H_{LS} = 15, M_{LS} = [1, 2, \cdots, 15], H_{CR} = 5, M_{CR} = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5]$	–

* There are too many parameters in CMA-ES. To describe simply, the parameter settings can refer to the literature [35].

4.3.4. Other parameter settings

In addition to the above key parameters, the inertia weight indicated in Eqs. (1) and (3) also significantly impacts the performance of PSO. Generally, the greater inertia weight encourages population diversity while the smaller one can favor rapid convergence. However, due to the absence of available information about landscape in advance, we cannot provide a deterministic function related to the value of inertia weight. In view of such consideration, we applied bimodal distribution to set the inertia weight w_i for each particle $\boldsymbol{X}_i$. To be specific, if the current particle can successfully update its corresponding personal best,$w_{i,G+1} = w_{i,G}$. Otherwise, $w_{i,G+1}$is regenerated for the next generation according to the bimodal distribution parameter setting. The distribution for $w_{i,G} (i = 1, 2, \cdots, M_G)$ includes two Cauchy distributions as follows:

$$w_{i,G} = \begin{cases} randc_i(0.7, 0.1) & rand < 0.5 \\ randc_i(0.3, 0.1) & otherwise \end{cases} \tag{14}$$

where $rand$ is a uniformly distributed random number between 0 and 1, and $randc_i(a, b)$ is a random number which satisfies Cauchy distribution with location parameter a and scale parameter b. If the value of >0$w_{i,G}$.9, then is truncated to 0.9 and if the value of $w_{i,G}$ is <0.2, then is truncated to 0.2. Based on the above analysis, the application of Eq. (14) can achieve an effective tradeoff between the exploration and exploitation in case of no available information about the characteristics of the problems to be solved.

4.4. Framework of the proposed SAHLPSO

By combining two-role learning strategy and self-adaptive parameter tuning mechanism, a novel SAHLPSO is presented. The pseudo-code and flowchart of SAHLPSO have been shown in Figs. 3 and 4, respectively. It can be found that SAHLPSO is easy to implement.

Recall that the exploration emphasizes on finding the more promising region and thus can avoid being trapped into local optimum especially at the early state, while exploitation focuses on the fine search in some already found optimal regions and thereby can improve convergence rate. This means that a desirable PSO variant should possess strong exploration capability at early stage and simultaneously strong exploitation capability at later stage. To measure the exploration and exploitation during the run, we calculate the population diversity proposed by Ref. [34] of the proposed algorithm along with the increasing of generations. The population diversity for each generation is calculated according to the following equation:

$$Diversity = \frac{1}{M} \sum_{i=1}^{M} \sqrt{\sum_{j=1}^{D} \left(x_{i,j} - \bar{x}_j\right)^2} \tag{15}$$

$$\bar{x}_j = \frac{\sum_{i=1}^{M} x_{i,j}}{M} \tag{16}$$

where M is the size of the population, and D is the dimension of the solution space. $x_{i,j}$ represents the$j$th dimension of the ith particle and $\bar{x}_j$ denotes the jth dimension of the mean position of the population.

To investigate the different utilities of two different role learning strategies, we provide two modified versions: SAHLPSOv1 with only exploration learning strategy and SAHLPSOv2 with only exploitation learning strategy and analyze their characteristics related to exploration and exploitation. For the proposed SAHLPSO, we set $Lg = 0.5$, which indicates that

Algorithm 1 SAHLPSO

Input: M: the initial number of the whole population
$MAXNFE$: the maximum number of function evaluation
Lg: the ratio of the particles to perform the exploration learning strategy.
p : the ratio of the top history personal bests for the exploitation learning strategy to generate the donor vector.
M_{CR}: the candidate set of the crossover probability for two-role learning strategy.
H_{CR}: the length of the initial M_{CR}
M_{LS}: the candidate set of the learning step size for two-role learning strategy
H_{LS}: the length of the initial M_{LS}
LP: learning period for M_{CR} and M_{LS}, By default 20.
1: /* Initialization */
2: G=1, Generate an initial $\boldsymbol{V}_i$,$\boldsymbol{X}_i, i \in (1,2,\cdots,M)$,by randomly sampling from the search space
3: Evaluate $f(\boldsymbol{X}_i), NFS = NFS + M$.
4:$\boldsymbol{Pbest}_i = \boldsymbol{X}_i$, set $\boldsymbol{gbest}$ to the current position of the particles.
5: $\boldsymbol{A}_i = \boldsymbol{Pbest}_i$, $P_{CR,k} = \frac{1}{H_{CR}}, nf_{CRk,G} = 0, ns_{CRk,G} = 0, k \in (1,2,\cdots,H_{CR}), P_{LS} = \frac{1}{H_{LS}}, nf_{LSk,G} = 0, ns_{LSk,G} = 0, k \in (1,2,\cdots,H_{LS})$,
6: randomly select $Lg * M$ particles to perform the exploration learning strategy
7: /* Main Loop */
8: while $NFS < MAXNFE$ do
9: for i=1 to M_G do
10: if mod $(G, LP) = 0$ or G=1 then
11: select an entry from M_{CR} as cr_i by stochastic universal selection method according to the corresponding P_{CR}
12: select an entry from M_{LS} as ls_i by stochastic universal selection method according to the corresponding P_{LS}
13: end
14: if $stagm_i = ls_i$ then
15: /* Generate a new exemplar*/

Fig. 3. The pseudo-code of the proposed algorithm.

the half of the particles perform the exploration learning strategy, $H_{CR} = 5$,$M_{CR} = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5]$, $H_{LS} = 15$,$M_{LS} = [1, 2, \cdots, 15]$ and $p = 0.2$ due to the framework of asynchronous SAHLPSO. Fig. 5 illustrates intuitively the diversity curves of the two widely used Griewank and Weierstrass multimodal functions, where the solution space, the population size and the maximum number of generations are 30, 40, and 10,000, respectively.

As shown in Fig. 5, SAHLPSOv1 has the strongest diversity especially at the early stage, which suggests the exploration learning strategy has good exploration capability. As we expected, SAHLPSOv2 has the worst diversity during the whole evolutionary process, which means the exploitation learning strategy has good convergence rate. Compared to the two versions, the proposed SAHLPSO can achieve a good tradeoff between population diversity and convergence rate. As seen in Fig. 5, at the early stage, the proposed SAHLPSO can keep better population diversity, and has stronger exploration capability to

for d=1 to D do
  if $i \in Lg * M$, % the ith particle performs the exploration learning strategy
    if $f(\boldsymbol{pbest}_m) < f(\boldsymbol{pbest}_n)$ then $o_i^d = \boldsymbol{pbest}_m^d$
    else $o_i^d = \boldsymbol{pbest}_n^d$ end
  else % the ith particle performs the exploitation learning strategy
    $o_i^d = pbest_{j\in[1,p*M_G]}^d$;% one of the top p best $p * M_G$ particles
    $a_i = pbest_i$;
    end
  if rand < cr_i then $e_i^d = o_i^d$
  else $e_i^d = a_i^d$ end
  if $i \sim \in Lg * M$ % the ith particle performs the exploitation learning strategy
    $e_i^d = rand1_i^d * e_i^d + (1 - rand1_i^d) * gbest_i^d$
  end
end %end of generating a new exemplar
if cr_i=$M_{CR,k}$ then $nf_{CRk,G} = nf_{CRk,G} + 1$; if ls_i=$M_{LS,k}$ then $nf_{LSk,G} = nf_{LSk,G} + 1$;
/*Update the particle*/
$\boldsymbol{V}_i = w_i * \boldsymbol{V}_i + c_1 * \boldsymbol{rand}.* (\boldsymbol{E}_i - \boldsymbol{X}_i)$, $\boldsymbol{X}_i = \boldsymbol{X}_i + \boldsymbol{V}_i$
evaluate $f(\boldsymbol{X}_i)$, $NFS = NFS + 1$
if $f(\boldsymbol{X}_i) < f(\boldsymbol{pbest}_i)$ then
  $\boldsymbol{A}_i = \boldsymbol{pbest}_i; \boldsymbol{pbest}_i = \boldsymbol{X}_i$; if $f(\boldsymbol{pbest}_i) < f(\boldsymbol{gbest})$ then $\boldsymbol{gbest} = \boldsymbol{pbest}_i$; end
  $ns_{CRk,G} = ns_{CRk,G} + 1; ns_{LSk,G} = ns_{LSk,G} + 1$
else
  if rand < 0.5 then $w_i = randc_i(0.7,0.1)$ else $w_i = randc_i(0.3,0.1)$ end
end
end /*end of learning of all particles*/
/*Update M_{CR}, P_{CR} and M_{LS}, P_{LS}*/
if mod (G, LP) $= 0$ then
  Update $P_{CR,k}$ according to Eqs. (9) and (10); Update $P_{LS,k}$ according to Eqs. (11) and (12)
  if $\sum_{k=1}^{H_{CR}} S_{CRk,G} = 0$ then
    restart M_{CR}, P_{CR}, $H_{CR} = H_{CR} + 1$ as described in subsection 4.3.1
  end
end
Calculate M_{G+1} according to Eq. (14)
if $M_{G+1} < M_G$ then
  Sort individuals based on their $\boldsymbol{pbest}_i$ fitness values and delete lowest $M_G - M_{G+1}$ members according the described strategy in Section 4.4
  Resize the all variables related to M_G
end
G = G+1
end /*end of main loop*/

Fig. 3 (*continued*)

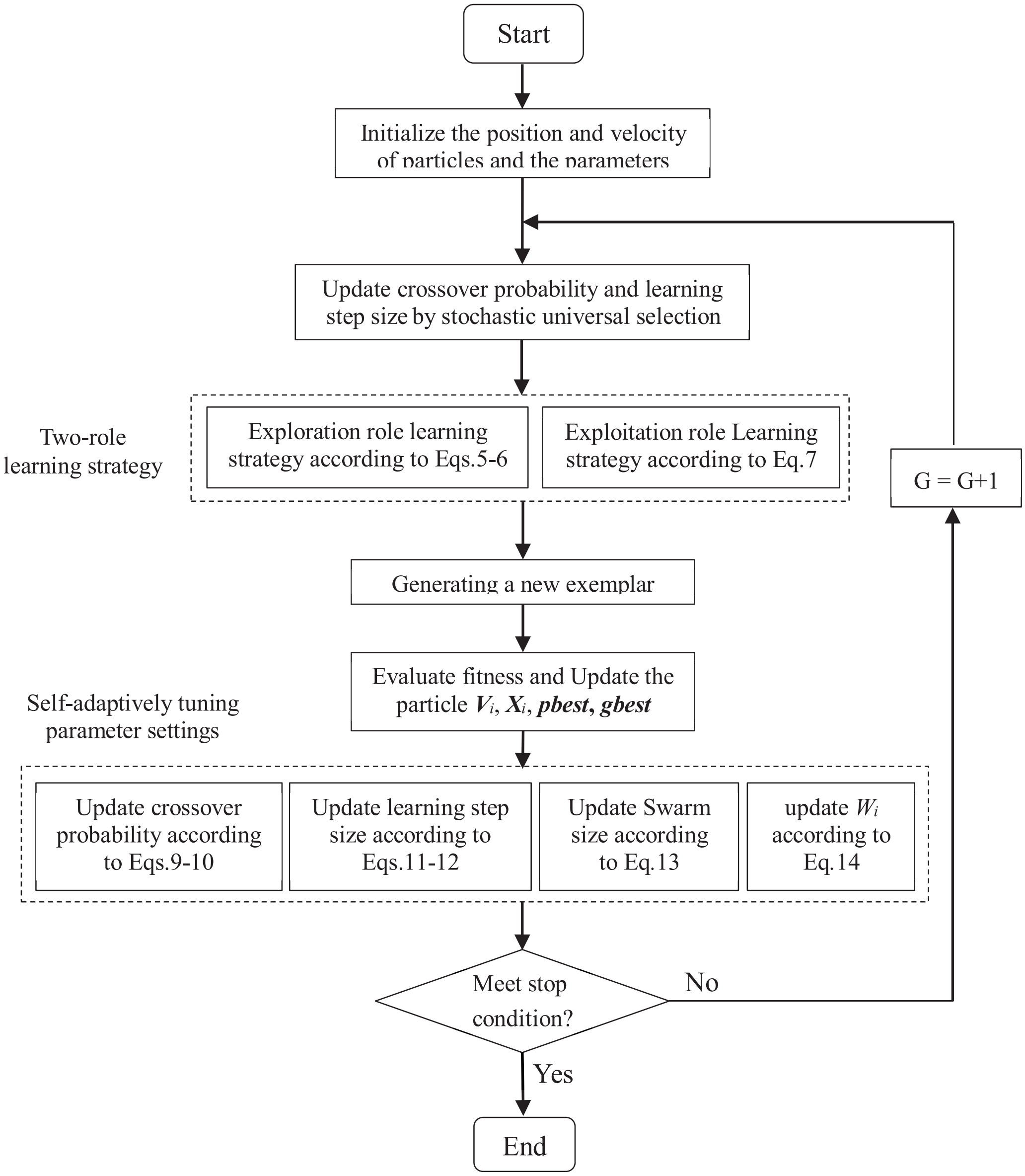

Fig. 4. Flowchart of SAHLPSO.

search more potential promising regions as SAHLPSOv1. In addition, the proposed SAHLPSO algorithm can maintain the strong exploitation capability due to the application of exploitation learning strategy and self-adaptive parameters tuning mechanism in the later evolutionary stage. In fact, in addition to the fact that the application of two-role learning strategy can encourage the balance of exploration and exploitation, the self-adaptively tuning parameter settings mechanism also plays a positive role in keeping population diversity and accelerating convergence rate. Here, to illustrate the mechanism of parameter self-adaptively tuning, we investigate the change of selecting probability values related to learning step sizes on the above two used multimodal functions. To facilitate the comparison, we set $H_{LS} = 5$, $M_{LS} = [1, 2, \cdots, 5]$. Learning period ($LP$) is set to be 20 be default. The changes of corresponding probability values to every entry in M_{LS} for Griewank and Weierstrass are shown in Fig. 6(a) and (b), respectively.

From the results, we can find that corresponding probability values related to each entry in M_{LS} change with successful update rate of it during the previous learning period. The higher the successful updating rate of each entry has, the greater its corresponding selecting probability is. Due to the characteristics of landscape concerning multimodal functions, the greater learning step size is more helpful to diversify population and thus survive successfully from local optimum. The observations from the above results can also coincide with such claim. To be specific, for the two multimodal functions, the probabilities of selecting 5 and 4 step sizes become greater increasingly along with the run while the probabilities of selecting 1 and 2 step

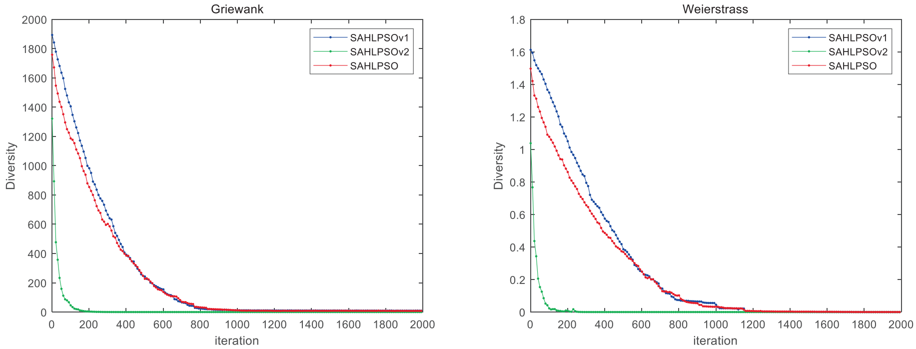

Fig. 5. The diversity plot of different learning strategies applied in the SAHLPSO.

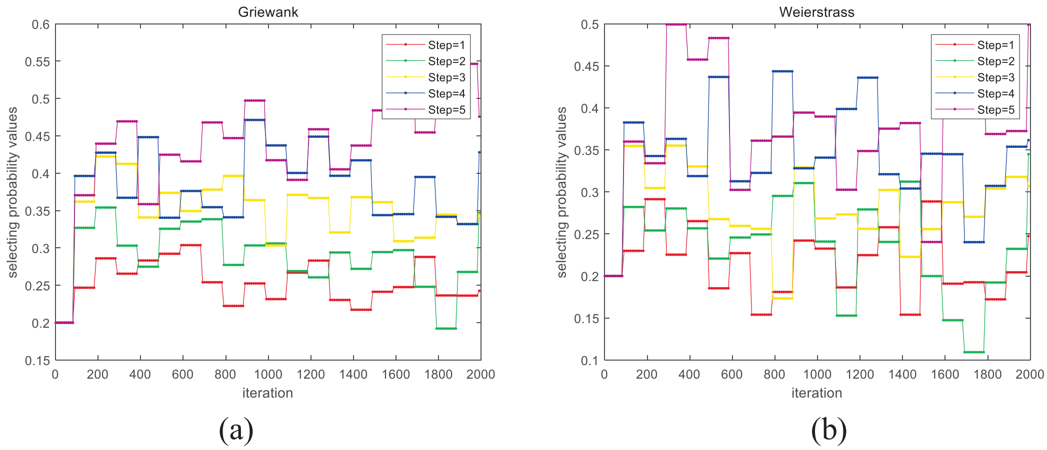

Fig. 6. The change of selecting probabilities corresponding to every entry in M_{LS}.

sizes become smaller increasingly along with the run as shown in Fig. 6. The experimental results also demonstrate the effectiveness of success-history based self-adaptively parameters tuning mechanism to well balance exploration and exploitation.

5. Experimental results and comparisons

5.1. Experimental setup

To thoroughly evaluate the effectiveness of the proposed SAHLPSO algorithm in solving different types of optimization problems, we performed a series of experiments on popular test suites. The test suite is from the CEC2013 test suite, which can be divided into three categories: 5 unimodal functions (F1-F5), 15 multimodal functions (F6-F20), and 8 compositions functions (F21-F28). The detailed descriptions about CEC2013 test suite can be found in the literature [20]. The purpose of using these shifted and rotated functions are to test the performance of different algorithms in more difficult and complex cases. The effectiveness and efficiency of the proposed SAHLPSO are then compared with those of advanced peer algorithms, which include ALPSO [8], CLPSO [19], HCLPSO [34], SRPSO [25], EPSO [32], FIPS [38], CLPSO-LOT [22], TSLPSO [9], and GL-PSO [46]. ALPSO is a novel hybrid improved PSO variant which attempts to adopt a self-adaptive mutation strategy to enhance population diversity, and extensive experimental results have shown its superiority in solving most of optimization problems. FIPS is a classical topology modified PSO version, all neighbors' experiences of a particle are used to update its velocity according to their weights based on their own fitness values and neighbor size. SRPSO is an enhanced PSO variant where self-regulation and self-perception strategies are used to improve the learning scheme for the tradeoff between exploration and exploitation and the experimental results show its outstanding performance. EPSO is an ensemble PSO version and has been proven to perform well on some real-world applications. GL-PSO is a genetic learning particle swarm optimization, which utilizes genetic operators to generate high-quality exemplar for particle updates. CLPSO is a well-known advanced PSO variant which applies a novel comprehensive learning strategy to enhance population diversity and its superiority has been proven in some recently reported researches especially in solving multimodal optimization functions. In addition to CLPSO, some CLPSO modified variants such as HCLPSO, TSLPSO and CLPSO-LOT are also used as compared methods since they all attempted to develop novel learning schemes and construct effective exemplars, and thus are closely relevant to the proposed SAHLPSO algorithm. The parameters of all compared algorithms are set according to those proposed in their corresponding references and are listed in Table 3. For fair comparison, we empirically set the same parameter settings on all cases: $Lg = 0.2$,$p = 0.2$ for two-role learning strategy, and $H_{CR} = 5$, $M_{CR} = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5]$, $H_{LS} = 15$,$M_{LS} = [1, 2, \cdots, 15]$, and $LP = 20$ for the success-history self-adaptively parameter tuning mechanism. The population size is fixed at 40 for all the algorithms. These algorithms are tested on 30-D functions with the same $MaxFEs = 10000 * D$. For each benchmark, each algorithm is repeated 51 times independently and then statistical results are obtained. All compared algorithms are implemented in MATLAB 2016b and run under Windows10 operation system on the same hardware environment with an INTEL 2.4 GHz and 8G memory.

5.2. The benefit of SAHLPSO components

In this section, we are interested in identifying the benefit of the two components of the proposed SAHLPSO algorithm: two-role learning strategy and parameters self-adaptively tuning mechanism. For this purpose, we consider three variants of SAHLPSO, g-SAHLPSO, l-SAHLPSO and nona-SAHLPSO. g-SAHLPSO and l-SAHLPSO differ from SAHLPSO with $Lg = 0.5$ in that g-SAHLPSO only applies exploration-role learning strategy while l-SAHLPSO only applies exploitation-role learning strategy. nona-SAHLPSO does not adopt adaptive learning control parameters, i.e., instead of updating $cr_i$and ls_i according to successful parameter settings previously used found during the run, we set $cr_i = 0.01$ and $ls_i = 5$ throughout the evolutionary process. Some unimodal functions and multimodal functions in the test suite which are listed in the Appendix are tested in the experiments. The simulation and statistical results obtained by SAHLPSO, g-SAHLPSO, l-SAHLPSO and nona-SAHLPSO on them are summarized in Tables 4 and 5, respectively.

From the results, we can find that on all tested unimodal functions without local optimum Sphere, Schwefel 2.22 and High Conditioned Elliptic, l-SAHLPSO perform the best in terms of solution accuracy and convergence rate, which indicates that l-SAHLPSO has strong exploitation capability due to its relatively greedy exploitation-role learning strategy. Compared to l-SAHLPSO, SAHLPSO ranks second due to the application of two-role learning strategy. Owing to the absence of the particles with the exploitation-role learning strategy, g-SAHLPSO cannot obtain the same desired results as l-SAHLPSO and SAHLPSO on Sphere, Schwefel 2.22 and High Conditioned Elliptic. However, on Rosenbrock, which is relatively more complex than the other three functions, g-SAHLPSO performs the best compared to other peer variants due to its strong exploration capability. SAHLPSO still ranks the second. These results also show that SAHLPSO with two-role learning strategy can well balance the exploration and exploitation. To facilitate the comparison of solution accuracy and convergence rate, we present final statistical results in terms of mean ranking about Mean and FEs metrics in Fig. 7. From the results, we can find that SAHLPSO can obtain relatively satisfactory results on unimodal functions due to the application of two-role learning strategy.

As we expected, on all tested multimodal functions, l-SAHLPSO performs poorly among the compared variants except nona-SAHLPSO. This results also indicate that the greedy exploitation-role learning strategy makes l-SAHLPSO incapable of maintaining sufficiently high population diversity and thus suffer from frequent premature convergence on these tested multimodal functions. On contrast, g-SAHLPSO obtains the best results since the application of exploration-role learning strategy can keep the strong population diversity and avoid premature especially at the early stage. Compared to g-SAHLPSO, SAHLPSO can achieve the same performance on some multimodal functions except Rastrigin and Ackley ones. On the two functions, SAHLPSO ranks the second. The results can again show that the tradeoff between the exploration and exploitation plays a crucial role in addressing different types of optimization problems. The final statistical results shown in Fig. 7 can demonstrate this conclusion.

No matter on unimodal and multimodal functions, nona-SAHLPSO cannot yield satisfactory results due to the absence of self-adaptive parameter tuning mechanism, which means that the application of self-adaptive parameter tuning mechanism can play a remarkably positive role in improving optimization performance. This is because that self-adaptive parameter tuning mechanism is able to adapt parameters to appropriate values according to the pervious evolutionary information and thus well balance the exploration and exploitation during the run.

5.3. Experimental results and comparisons on test suite

To further validate the performance of SAHLPSO, comparisons are made on shifted and rotated functions included in CEC2013, which are very complex and difficult to solve. The mean and standard deviations of the fitness values obtained by all compared algorithms are listed in Table 6, where the performance rank of all compared algorithms in terms of fitness mean value are also presented. From the results we can observe that the proposed SAHLPSO algorithm is a greatly competitive PSO variant. Specifically, SAHLPSO ranks the first on 12 functions out of all 28 tested functions, the second on 6 functions and the thirst on 5 functions. To facilitate intuitive comparison, the mean rankings of all compared algorithms are shown in Fig. 8. Compared with the peer algorithms, the proposed algorithm can obtain relatively better or competitive results on most of functions except F14 and F16.

Table 4
Results of different SAHLPSO versions on some unimodal benchmarks.

Functions	Algorithms	Mean	Dev	Rank	SR	FEs
Sphere (1e−60)	g_SAHLPSO	6.60e−103	2.08e−102	3	100	245,893
	l_SAHLPSO	**2.67**e−**234**	**0**	**1**	**100**	**96,846**
	nona_SAHLPSO	5.55e + 02	1.13e + 02	4	0	–
	SAHLPSO	2.73e−118	6.69e−118	2	100	133,752
Schwefel 2.22 (1e−35)	g_SAHLPSO	3.57e−56	1.13e−55	3	100	247,526
	l_SAHLPSO	**1.07**e−**108**	**3.25**e−**108**	**1**	**100**	**99,074**
	nona_SAHLPSO	8.07e + 00	6.07e−01	4	0	–
	SAHLPSO	2.38e−58	3.46e−48	2	100	203,653
High Conditioned Elliptic (1e−50)	g_SAHLPSO	6.59e−100	2.06e−99	3	100	233,582
	l_SAHLPSO	**3.41**e−**229**	**0**	**1**	**100**	**90,761**
	nona_SAHLPSO	1.73e + 05	3.37e + 04	4	0	–
	SAHLPSO	3.42e−116	7.86e−116	2	100	116,358
Rosenbrock (1e + 01)	g_SAHLPSO	**1.82**e−**02**	**2.21**e−**02**	**1**	**100**	**84,993**
	l_SAHLPSO	8.13e−01	1.70e + 00	3	100	188,284
	nona_SAHLPSO	1.41e + 03	3.61e + 02	4	0	–
	SAHLPSO	3.36e−02	4.62e−02	2	100	108,488

Table 5
Results of different SAHLPSO versions on some multimodal benchmarks.

Functions	Algorithms	Mean	Dev	Rank	SR	FEs
Weierstrass (0)	g_SAHLPSO	**0**	**0**	**1**	**100**	**157,562**
	l_SAHLPSO	6.65e−01	1.06e + 00	2	0	–
	nona_SAHLPSO	6.95e + 00	9.22e−01	3	0	–
	SAHLPSO	**0**	**0**	**1**	**100**	**161,841**
Dminima (4.6e−10)	g_SAHLPSO	**4.57**e−**10**	**0**	**1**	**100**	**97,831**
	l_SAHLPSO	1.09e + 01	3.00e + 00	4	0	–
	nona_SAHLPSO	2.99e−01	1.00e−01	3	0	–
	SAHLPSO	**4.57**e−**10**	**0**	**1**	**100**	**105,685**
Griewan (0)	g_SAHLPSO	**0**	**0**	**1**	**100**	**196,181**
	l_SAHLPSO	1.03e−02	1.11e−02	3	33.33	240,239
	nona_SAHLPSO	2.16e + 00	3.32e−01	4	0	–
	SAHLPSO	**0**	**0**	**1**	**100**	**217,079**
Schewfel (0)	g_SAHLPSO	**0**	**0**	**1**	**100**	**167,792**
	l_SAHLPSO	5.18e + 03	9.94e + 02	4	0	–
	nona_SAHLPSO	2.32e + 02	2.00e + 02	3	0	–
	SAHLPSO	7.11e + 01	1.08e + 02	2	60	185,171
Rastrigin (0)	g_SAHLPSO	**0**	**0**	**1**	**100**	**137,825**
	l_SAHLPSO	4.21e + 01	1.35e + 01	3	0	–
	nona_SAHLPSO	1.07e + 02	1.84e + 01	4	0	–
	SAHLPSO	1.99e−01	5.58e−01	2	86.67	155,213
Ackley (7.2e−15)	g_SAHLPSO	**6.16**e−**15**	**1.63**e−**15**	**1**	**100**	**193,009**
	l_SAHLPSO	3.82e−01	6.73e−01	3	0	–
	nona_SAHLPSO	4.33e + 00	5.30e−01	4	0	–
	SAHLPSO	7.34e−15	1.63e−15	2	86.67	231,463

To verify the statistical significance of the proposed algorithm compared to the peer algorithms, we also perform Holm's test on the test suite with regard to the derived mean rankings, where the proposed SAHLPSO algorithm was considered as the control algorithm. The null hypothesis is that the proposed SAHLPSO algorithm does not perform better than all other compared algorithms as the control algorithm. Table 7 presents all adjusted α values and the corresponding p-values for each compared results. From the results, we can find that the null hypothesis is rejected for all pairwise comparison at least at a significance level of $\alpha = 0.05$, indicating that the proposed algorithm outperforms the other compared algorithms with significant difference.

In addition, the convergence curves of all compared algorithms on some functions selected from CEC2013 test suite are plotted in Fig. 9. From results shown in Fig. 9, we can find that in most cases, SAHLLPSO can exhibit the rapidest convergence rate compared to the peer algorithms and simultaneously it is not easy to fall into local optima. This is mainly attributed to the fact that the application of tow-role learning strategy and success-history based parameter self-adaptively tuning mechanism can not only benefit maintaining good population diversity but also accelerating convergence rate, thus well balancing

Table 6
Results of different algorithms on the CEC2013 test suite.

Functions	F1		F2		F3		F4	
Metrics	Mean ± Dev.	Rank	Mean ± Dev.	Rank	Mean ± Dev.	Rank	Mean ± Dev.	Rank
ALPSO	1.31e−27 ± 5.30e−27	7	7.03e + 06 ± 4.82e + 06	7	**1.95e + 07 ± 2.67e + 07**	**1**	1.54e + 03 ± 6.08e + 02	2
CLPSO	2.54e−28 ± 3.74e−28	6	2.22e + 07 ± 5.97e + 06	8	5.37e + 08 ± 4.09e + 08	10	2.20e + 04 ± 4.83e + 03	10
HCLPSO	**0 ± 0**	**1**	6.06e + 06 ± 2.17e + 06	6	1.40e + 08 ± 1.03e + 08	8	5.90e + 03 ± 2.66e + 03	6
SRPSO	1.29e−20 ± 5.37e−20	9	4.22e + 06 ± 2.41e + 06	5	9.46e + 07 ± 1.44e + 08	7	5.39e + 03 ± 3.05e + 03	5
EPSO	1.01e−26 ± 4.99e−27	8	**1.31e + 05 ± 6.44e + 04**	**1**	7.46e + 07 ± 1.18e + 08	6	**6.63e + 01 ± 4.08e + 01**	**1**
FIPS	1.53e−19 ± 1.40e−19	10	2.87e + 07 ± 6.94e + 06	10	3.31e + 07 ± 4.55e + 07	3	1.62e + 04 ± 3.37e + 03	7
CLPSO-LOT	**0 ± 0**	1	2.62e + 07 ± 6.98e + 06	9	7.32e + 07 ± 8.46e + 07	5	2.17e + 04 ± 5.01e + 03	9
TSLPSO	**0 ± 0**	**1**	9.47e + 05 ± 4.61e + 05	3	1.74e + 08 ± 2.17e + 08	9	3.62e + 03 ± 1.58e + 03	4
GL-PSO	**0 ± 0**	**1**	2.61e + 05 ± 1.91e + 05	2	4.69e + 07 ± 5.86e + 07	4	1.97e + 04 ± 8.16e + 03	8
SAHLPSO	**0 ± 0**	**1**	9.48e + 05 ± 3.72e + 05	4	2.06e + 07 ± 2.02e + 07	2	2.04e + 03 ± 7.93e + 02	3
Functions	F5		F6		F7		F8	
ALPSO	1.27e−15 ± 3.81e−15	7	7.53e + 01 ± 3.86e + 01	10	3.70e + 01 ± 9.80e + 00	3	**2.09e + 01 ± 5.54e−02**	**1**
CLPSO	2.96e−21 ± 2.94e−21	6	3.49e + 01 ± 7.81e + 00	7	7.79e + 01 ± 1.01e + 01	10	**2.09e + 01 ± 5.13e−02**	**1**
HCLPSO	**0 ± 0**	**1**	1.77e + 01 ± 4.69e + 00	3	6.60e + 01 ± 1.00e + 01	7	**2.09e + 01 ± 5.12e−02**	**1**
SRPSO	2.00e−14 ± 2.54e−14	9	5.90e + 01 ± 2.74e + 01	9	2.35e + 01 ± 1.62e + 01	2	**2.09e + 01 ± 4.10e−02**	**1**
EPSO	1.99e−14 ± 6.07e−15	8	**1.01e + 01 ± 8.86e + 00**	**1**	3.75e + 01 ± 1.50e + 01	5	**2.09e + 01 ± 4.00e−02**	**1**
FIPS	1.17e−12 ± 4.45e−13	10	2.62e + 01 ± 4.28e−01	6	3.72e + 01 ± 1.57e + 01	4	2.10e + 01 ± 4.67e−02	10
CLPSO-LOT	**0 ± 0**	**1**	4.23e + 01 ± 8.80e + 00	8	6.98e + 01 ± 1.15e + 01	9	**2.09e + 01 ± 5.89e−02**	**1**
TSLPSO	**0 ± 0**	**1**	2.01e + 01 ± 1.79e + 01	5	6.89e + 01 ± 2.28e + 01	8	**2.09e + 01 ± 5.85e−02**	**1**
GL-PSO	**0 ± 0**	**1**	1.91e + 01 ± 1.53e + 01	4	5.57e + 01 ± 1.77e + 01	6	**2.09e + 01 ± 6.37e−02**	**1**
SAHLPSO	**0 ± 0**	**1**	1.55e + 01 ± 2.27e + 00	2	**2.08e + 01 ± 8.31e + 00**	**1**	**2.09e + 01 ± 5.62e−02**	**1**
Functions	F9		F10		F11		F12	
ALPSO	2.23e + 01 ± 3.77e + 00	5	1.60e−01 ± 9.87e−02	4	2.66e + 01 ± 8.80e + 00	8	7.68e + 01 ± 2.32e + 01	5
CLPSO	2.92e + 01 ± 1.52e + 00	9	5.19e + 00 ± 1.54e + 00	10	**0 ± 0**	**1**	1.23e + 02 ± 1.54e + 01	9
HCLPSO	2.70e + 01 ± 1.96e + 00	7	3.93e−01 ± 1.47e−01	8	3.32e−02 ± 1.82e−01	4	1.09e + 02 ± 2.62e + 01	8
SRPSO	**1.43e + 01 ± 3.55e + 00**	**1**	1.58e−01 ± 7.50e−02	3	2.71e + 01 ± 8.66e + 00	9	5.98e + 01 ± 3.16e + 01	3
EPSO	2.10e + 01 ± 2.41e + 00	4	1.30e−01 ± 5.93e−02	2	1.33e−01 ± 3.44e−01	5	**5.32e + 01 ± 1.47e + 01**	**1**
FIPS	3.31e + 01 ± 2.88e + 00	10	4.22e−01 ± 3.16e−01	9	6.07e + 01 ± 1.37e + 01	10	1.79e + 02 ± 7.73e + 00	10
CLPSO-LOT	2.80e + 01 ± 1.34e + 00	8	3.87e−01 ± 4.97e−01	7	3.52e + 00 ± 3.46e + 00	7	7.75e + 01 ± 1.30e + 01	6
TSLPSO	2.46e + 01 ± 2.67e + 00	6	**1.10e−01 ± 6.28e−02**	**1**	5.92e−17 ± 3.24e−16	2	9.23e + 01 ± 2.43e + 01	7
GL-PSO	2.05e + 01 ± 4.14e + 00	3	1.67e−01 ± 8.80e−02	5	1.33e−01 ± 3.44e−01	5	5.99e + 01 ± 1.77e + 01	4
SAHLPSO	1.99e + 01 ± 4.12e + 00	2	2.16e−01 ± 8.62e−02	6	1.18e−16 ± 6.49e−16	3	5.44e + 01 ± 1.59e + 01	2
Functions	F13		F14		F15		F16	
ALPSO	1.42e + 02 ± 3.18e + 01	7	1.01e + 03 ± 2.11e + 02	8	5.42e + 03 ± 1.37e + 03	9	2.25e + 00 ± 2.58e−01	7
CLPSO	1.69e + 02 ± 1.76e + 01	9	**9.02e−01 ± 1.31e + 00**	**1**	4.29e + 03 ± 3.86e + 02	6	1.37e + 00 ± 2.05e−01	4
HCLPSO	1.37e + 02 ± 2.69e + 01	6	4.39e + 00 ± 2.16e + 01	3	3.55e + 03 ± 4.15e + 02	2	9.08e−01 ± 2.28e−01	2
SRPSO	**1.12e + 02 ± 3.14e + 01**	**1**	1.42e + 03 ± 3.57e + 02	9	4.86e + 03 ± 1.56e + 03	8	2.48e + 00 ± 2.29e−01	8
EPSO	1.18e + 02 ± 3.45e + 01	3	4.26e + 01 ± 5.79e + 01	6	3.59e + 03 ± 4.92e + 03	3	1.52e + 01 ± 3.08e−01	10
FIPS	1.81e + 02 ± 8.51e + 00	10	5.18e + 03 ± 3.98e + 02	10	7.16e + 03 ± 3.56e + 02	10	2.49e + 00 ± 2.14e−01	9
CLPSO-LOT	**1.17e + 02 ± 1.65e + 01**	2	4.69e + 02 ± 1.86e + 02	7	4.43e + 03 ± 3.67e + 02	**7**	1.42e + 00 ± 2.13e−01	6
TSLPSO	1.42e + 02 ± 1.92e + 01	7	1.20e + 01 ± 2.08e + 01	4	4.05e + 03 ± 7.78e + 02	5	1.36e + 00 ± 3.13e−01	3
GL-PSO	1.21e + 02 ± 2.84e + 01	5	1.21e + 00 ± 1.07e + 00	2	3.92e + 03 ± 6.51e + 02	4	**5.67e−01 ± 4.00e−01**	**1**
SAHLPSO	1.20e + 02 ± 2.54e + 01	4	1.57e + 01 ± 3.68e + 01	5	**3.46e + 03 ± 4.55e + 02**	**1**	1.38e + 00 ± 3.06e−01	5
Functions	F17		F18		F19		F20	
ALPSO	6.06e + 01 ± 1.84e + 01	9	1.66e + 02 ± 4.36e + 01	7	3.37e + 00 ± 9.65e−01	9	1.17e + 01 ± 5.92e−01	4
CLPSO	**3.05e + 01 ± 4.86e−02**	**1**	1.80e + 02 ± 1.41e + 01	8	3.04e−01 ± 1.01e−01	2	1.34e + 01 ± 5.79e−01	9
HCLPSO	**3.05e + 01 ± 4.16e−02**	**1**	1.11e + 02 ± 1.60e + 01	4	**2.48e−01 ± 7.87e−02**	**1**	1.08e + 01 ± 6.07e−01	3
SRPSO	5.35e + 01 ± 5.95e + 00	8	1.94e + 02 ± 1.47e + 01	9	2.65e + 00 ± 5.16e−01	8	1.32e + 01 ± 1.97e + 00	7
EPSO	3.33e + 01 ± 2.04e + 00	6	**7.51e + 01 ± 1.30e + 01**	**1**	2.04e + 00 ± 5.95e−01	7	**1.06e + 01 ± 8.50e−01**	**1**
FIPS	1.70e + 02 ± 1.13e + 01	10	2.05e + 02 ± 1.31e + 01	10	1.26e + 01 ± 1.08e + 00	10	1.49e + 01 ± 3.95e−01	10
CLPSO-LOT	3.41e + 01 ± 1.76e + 00	7	1.31e + 02 ± 1.18e + 01	5	1.50e + 00 ± 1.43e−01	5	1.25e + 01 ± 8.25e−01	6
TSLPSO	3.07e + 01 ± 1.65e−01	3	1.31e + 02 ± 2.52e + 01	5	1.85e + 00 ± 5.25e−01	6	1.21e + 01 ± 7.15e−01	5
GL-PSO	3.13e + 01 ± 5.56e−01	5	7.58e + 01 ± 1.48e + 01	2	1.48e + 00 ± 3.03e−01	4	1.33e + 01 ± 1.59e + 00	8
SAHLPSO	3.07e + 01 ± 1.76e−01	3	8.50e + 01 ± 1.60e + 01	3	1.35e + 00 ± 2.09e−01	3	**1.06e + 01 ± 9.66e−01**	**1**
Functions	F21		F22		F23		F24	
ALPSO	2.79e + 02 ± 8.08e + 01	5	1.06e + 03 ± 3.35e + 02	8	5.04e + 03 ± 1.34e + 03	7	2.65e + 02 ± 1.07e + 01	5
CLPSO	2.94e + 02 ± 1.88e + 01	6	**8.10e + 01 ± 4.04e + 01**	**1**	5.47e + 03 ± 4.66e + 02	9	2.74e + 02 ± 6.41e + 00	8
HCLPSO	2.60e + 02 ± 5.63e + 01	3	1.06e + 02 ± 2.42e + 01	2	4.39e + 03 ± 4.11e + 02	5	2.71e + 02 ± 9.19e + 00	7
SRPSO	3.09e + 02 ± 8.33e + 01	8	1.37e + 03 ± 3.54e + 02	9	4.34e + 03 ± 1.07e + 03	4	2.43e + 02 ± 1.57e + 01	2

(*continued on next page*)

Table 6 (*continued*)

Functions	F1		F2		F3		F4	
Metrics	Mean ± Dev.	Rank	Mean ± Dev.	Rank	Mean ± Dev.	Rank	Mean ± Dev.	Rank
EPSO	2.51e + 02 ± 4.95e + 01	2	1.38e + 02 ± 4.67e + 01	5	4.07e + 03 ± 4.84e + 02	2	2.52e + 02 ± 9.33e + 00	4
FIPS	3.07e + 02 ± 6.95e + 01	7	5.16e + 03 ± 5.44e + 02	10	7.57e + 03 ± 2.09e + 02	10	2.95e + 02 ± 5.12e + 00	10
CLPSO-LOT	3.12e + 02 ± 6.80e + 01	9	3.43e + 02 ± 2.01e + 02	7	5.27e + 03 ± 4.09e + 02	8	2.79e + 02 ± 4.63e + 00	9
TSLPSO	2.65e + 02 ± 4.73e + 01	4	1.61e + 02 ± 7.26e + 01	6	4.67e + 03 ± 6.64e + 02	6	2.68e + 02 ± 1.14e + 01	6
GL-PSO	3.17e + 02 ± 8.00e + 01	10	1.06e + 02 ± 4.61e + 01	2	4.32e + 03 ± 7.40e + 02	3	2.47e + 02 ± 1.21e + 01	3
SAHLPSO	**2.32e + 02 ± 4.35e + 01**	**1**	1.11e + 02 ± 4.34e + 01	4	**3.89e + 03 ± 6.95e + 02**	**1**	**2.28e + 02 ± 1.26e + 01**	**1**
Functions	F25		F26		F27		F28	
ALPSO	2.81e + 02 ± 1.09e + 01	4	3.10e + 02 ± 7.64e + 01	10	8.86e + 02 ± 7.61e + 01	8	3.87e + 02 ± 3.37e−13	10
CLPSO	2.97e + 02 ± 5.00e + 00	9	2.02e + 02 ± 6.32e−01	5	6.97e + 02 ± 3.30e + 02	2	3.00e + 02 ± 1.90e−04	6
HCLPSO	2.95e + 02 ± 7.87e + 00	8	**2.00e + 02 ± 7.72e−02**	**1**	8.29e + 02 ± 3.14e + 02	7	**2.80e + 02 ± 6.08e + 01**	**1**
SRPSO	2.75e + 02 ± 9.60e + 00	3	3.04e + 02 ± 5.92e + 01	9	7.16e + 02 ± 8.86e + 01	3	3.57e + 02 ± 2.76e + 02	9
EPSO	2.89e + 02 ± 7.85e + 00	6	**2.00e + 02 ± 7.11e−03**	**1**	7.46e + 02 ± 1.86e + 02	4	2.87e + 02 ± 5.05e + 01	2
FIPS	2.99e + 02 ± 2.51e + 00	10	2.65e + 02 ± 8.86e + 01	8	1.25e + 03 ± 4.58e + 01	10	3.01e + 02 ± 2.63e + 00	8
CLPSO-LOT	2.87e + 02 ± 5.31e + 00	5	2.08e + 02 ± 2.98e + 01	6	1.03e + 03 ± 3.22e + 01	9	3.00e + 02 ± 8.18e−14	6
TSLPSO	2.92e + 02 ± 6.48e + 00	7	**2.00e + 02 ± 3.17e−02**	**1**	7.53e + 02 ± 2.84e + 02	5	2.87e + 02 ± 5.07e + 01	2
GL-PSO	2.71e + 02 ± 8.08e + 00	2	2.56e + 02 ± 6.96e + 01	7	7.76e + 02 ± 1.13e + 02	6	2.93e + 02 ± 3.65e + 01	4
SAHLPSO	**2.63e + 02 ± 2.27e + 01**	**1**	**2.00e + 02 ± 1.91e−02**	**1**	**5.57e + 02 ± 1.43e + 02**	**1**	2.93e + 02 ± 3.57e + 01	4

exploration and exploitation. To summarize, the above results have demonstrated the effectiveness and efficiency of the two strategies applied in SAHLLPSO for performance improvement Fig. 10.

5.4. Comparisons with other outstanding evolutionary algorithms

In this case, we further compare the proposed algorithm with other four outstanding evolutionary algorithms: SaDE [4], CoDE [47], CMA-ES [35] and LSHADE [39]. 30-D problems are tested in the experiments, and related parameter configurations are set according to the corresponding references. Table 8 listed the comparison test results and Table 9 summarizes statistical results of this test at 0.05 significance level comparing each algorithm versus SAHLPSO on the 28 functions by Wilcoxon singed ranks test. From the results, we can find that the proposed algorithm outperforms SaDE and CMA-ES with significant difference and obtains comparable results with CoDE. However, we have to emphasize that the proposed algorithm fails on most of benchmark functions compared to LSHADE. It is well-known that the variants of SHADE have won most of the CEC single objective optimization competition in recent years, but we believe some different strategies on other optimization algorithms still contribute a lot to the improvement of optimization.

5.5. Investigation of parameters

In this section, the sensitivity of SAHLPSO to the parameters Lg and p were investigated. First, we perform SAHLPSO with Lg =$[0, 0.1, 0.2, \cdots, 1]$, respectively, and fix the other parameters as those presented previously. Unimodal functions Sphere and Schwefel 2.22 and multimodal functions Dminima, Schwefel, Rastrigin, and Ackley are tested in the experiments. The effect of different settings of Lg on the performance of SAHLPSO is plotted in Fig. 11, where the horizontal axis is the Lg value and the vertical axis is the log (mean fitness) value or log (average Fes) for each function. From the results, we can observe that with the increase of Lg, the solution accuracies on the unimodal functions become worse while the solution accuracies on the multimodal functions becomes better. This is because that the smaller Lg suggests that more particles will perform the exploitation learning and accelerate convergence rate, thus favor enhancing the solution accuracy of the unimodal functions. On contrast, the greater Lg indicates that there are more particles performing the exploration learning and keep strong population diversity, thereby encouraging solving multimodal problems. In terms of log (average Fes), the convergence rate of SAHLPSO decreases on unimodal functions with the increasing Lg, which indicates more particles performing exploitation learning strategy is helpful to improve the convergence speed on unimodal functions. However, for multimodal functions, the convergence rate of the proposed method does not exhibit obvious decreasing tendency with the increasing of Lg. This is because that when fewer particles perform exploration learning strategy in case of smaller Lg, SAHLPSO cannot reach pre-specified threshold fitness value due to the occurrence of premature at early evolution stage and thus result in the obtained invalid FEs. In fact, with the increase of Lg, more particles with exploration learning strategy would contribute to make SAHLPSO more rapidly reach pre-specified threshold fitness value, thus enhancing convergence rate. Certainly, the parameters self-adaptively tuning mechanism including linear population size reduction also plays a positive role in balancing the exploration and exploitation. Therefore, to keep population diversity and thus avoid premature convergence at early stage,

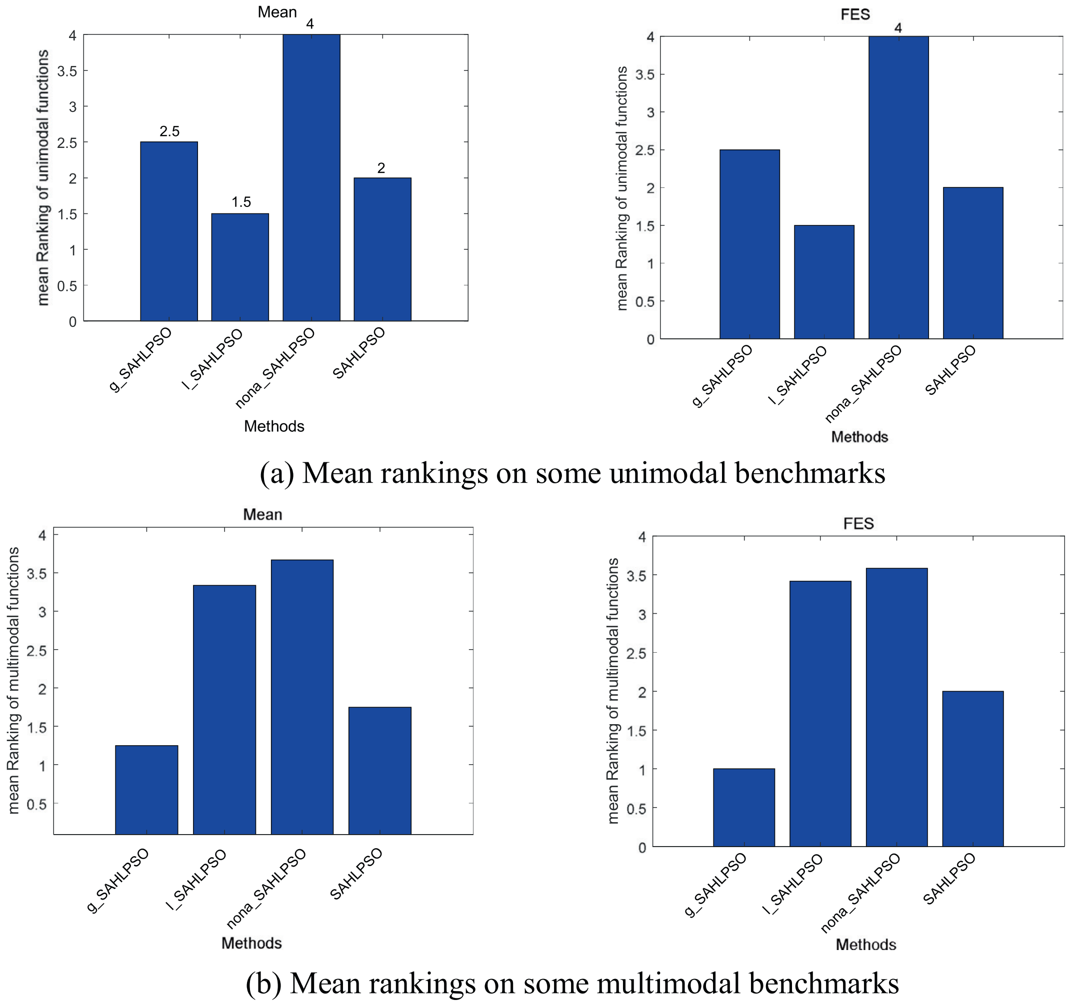

Fig. 7. Mean rankings of different SAHLPSO versions on some benchmarks.

Table 7
Results of Holm's test with SAHLPSO as the control algorithm on the second test suite.

Algorithms	$\alpha_{0.05}$	p-value
ALPSO	7.3008e−03	7.4548e−09
CLPSO	8.5124e−03	2.1940e−08
HCLPSO	1.6952e−02	5.5224e−03
SRPSO	1.0206e−02	1.3100e−07
EPSO	5.0000e−02	4.1570e−02
FIPS	5.6830e−03	3.6475e−20
CLPSO-LOT	6.3912e−03	3.3925e−09
TSLPSO	1.2741e−02	1.3913e−03
GL-PSO	2.5321e−02	1.2694e−02

Lg should be set to between 0.2 and 0.3 so that the proposed algorithm can exhibit favorable performance on both unimodal and multimodal problems.

In addition, the proposed SAHLPSO algorithm is also tested with $p = [0.1, \cdots, 1]$ and the other parameters are fixed as described previously. Fig. 12(a) shows that the solution accuracy obtained by SAHLPSO is slightly sensitive to the parameter p. The smaller p values mean that only a few the top rank best solutions participate in the generation of the learning exem-

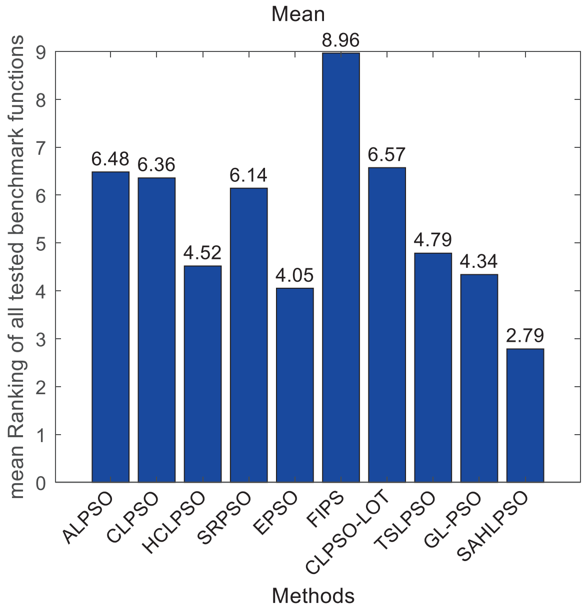

Fig. 8. Mean rankings on the CEC2013 of different algorithms in terms of Mean metrics.

plar, which can improve the solution accuracy especially in solving the unimodal problems. On the other hand, the greater p values indicate that most of best solutions contribute to the generation of the learning exemplar, which is helpful to maintain population diversity and thus avoid premature. Moreover, in Fig. 12(b), the FEs concerning the convergence rate of SAHLPSO using different p values are presented and compared. From the results, we can find the smaller p values can favor improving the search efficiency of SAHLPSO, especially on the unimodal problems. In addition, in the process of the exemplar construction for the exploitation learning strategy in SAHLPSO, the global best of the entire swarm is utilized to generate new exemplars, which can effectively enhance convergence rate. Similar toLg, the convergence rate of SAHLPSO on multimodal functions does not exhibit obvious decreasing tendency with the increasing of p due to the same reason as discussed above. Therefore, to keep population diversity and speed up convergence rate, p should be set to between 0.1 and 0.2 so that the proposed algorithm can exhibit favorable performance on both unimodal and multimodal problems.

6. Conclusion

In this paper, a Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization (SAHLPSO) was proposed. Specifically, in the proposed algorithm, two-role learning strategy is adopted to construct the learning exemplars: the exploration learning strategy and the exploitation learning strategy. The exploration-role learning strategy uses all other exploration-role particles' historical personal best information to generate learning exemplars. In addition, to further diversify the learning exemplars and avoid premature, recently previously explored inferior personal best solutions are used to inject diverse information into the generated exemplar. On the other hand, the exploitation learning strategy adopts elitism mechanism to construct learning exemplars, which can encourage generating highly qualified exemplars and thus provide improved guidance for the evolving particles to accelerate convergence rate. In addition to the two-role learning strategy, a self-adaptively parameter tuning mechanism is also developed, where the learning control parameters are self-adaptively set to appropriate values according to successful parameter settings that were previously used found during the run. The experimental results on some unimodal and multimodal benchmark functions show that the application of the two-role learning strategy and self-adaptively parameter tuning mechanism can be helpful to balance the exploration and exploitation so as to improve the optimization performance. In order to further evaluate the optimization performance, the proposed algorithm is compared with several state-of-the-art PSO variants and other outstanding evolutionary algorithms in CEC 2013 test suites. Experimental results demonstrate that the proposed SAHLPSO statistically outperforms the peer PSO algorithms or at least comparable to other advanced evolutionary algorithms on a majority of benchmarks in terms of the solution accuracy, search speed with significant difference. It is worth mentioning that the success of SAHLPSO mainly depends on the self-adaptive tradeoff between the exploration and exploitation due to the application of the two-role learning strategy and self-adaptive parameter tuning mechanism. This will encourage future research into the cascade hybridization of learning strategies with some emerging techniques and more promising self-adaptive parameter setting such as self-adaptively tuning for different parameter combinations. In addition, this would also lead to interesting future application of PSO algorithms for dynamic and multi-objective optimization, as well as real-world applications.

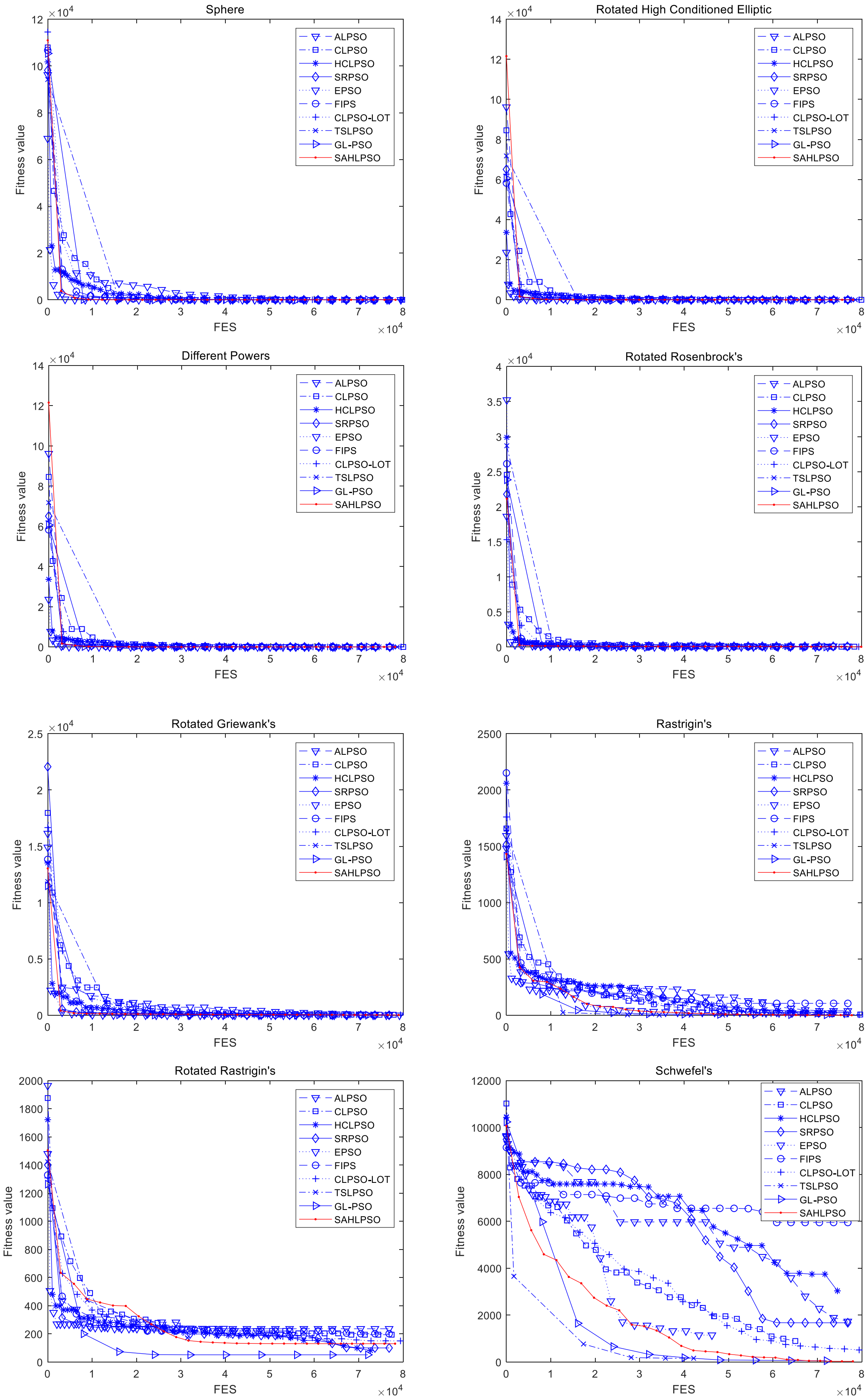

Fig. 9. Convergence curves of some CEC2013 functions obtained by different compared variants.

Table 8
The comparison test results with outstanding EAs on 30-D CEC2013 functions.

Func.	SaDE	CoDE	CMA-ES	LSHADE	SAHLPSO
F1	=0	>1.11e−13	>4.27e−13	=0	0
F2	>3.91e + 05	<1.07e + 05	<4.43e−03	<2.12e−11	9.48e + 05
F3	>2.18e + 07	<2.39e + 06	<3.56e + 02	<1.66e + 00	2.06e + 07
F4	>2.70e + 03	<1.13e + 00	<4.07e−13	<5.96e−20	2.04e + 03
F5	=0	>7.50e−14	>8.19e−13	=0	0
F6	>2.95e + 01	<4.73e + 00	<1.62e + 00	<5.66e−04	1.55e + 01
F7	>2.89e + 01	<9.78e + 00	<1.41e + 01	<7.94e−01	2.08e + 01
F8	=2.09e + 01	>2.15e + 01	>2.15e + 01	<2.08e + 01	2.09e + 01
F9	<1.73e + 01	<1.36e + 01	>4.26e + 01	>2.62e + 01	1.99e + 01
F10	>2.98e−01	<3.47e−02	<3.32e−02	=0	2.16e−01
F11	>2.49e−01	>1.99e−02	>9.39e + 01	>2.19e−05	1.18e−16
F12	<4.86e + 01	<3.71e + 01	>6.57e + 02	<4.99e + 00	5.44e + 01
F13	<9.84e + 01	<7.79e + 01	>1.75e + 03	<6.16e + 00	1.20e + 02
F14	<9.40e−01	>8.69e + 02	>5.39e + 03	<2.91e−02	1.57e + 01
F15	>4.66e + 03	>3.47e + 03	>5.14e + 03	<2.64e + 03	3.46e + 03
F16	>2.22e + 00	<3.56e−01	<8.72e−02	<6.05e−01	1.38e + 00
F17	<3.05e + 01	>3.47e + 01	>4.07e + 03	<3.04e + 01	3.07e + 01
F18	>1.29e + 02	<6.40e + 01	>4.17e + 03	<5.14e + 01	8.50e + 01
F19	>4.07e + 00	>2.32e + 00	>3.53e + 00	<1.16e + 00	1.35 + 00
F20	>1.07e + 01	=1.06e + 01	>1.27e + 01	<9.95e + 00	1.06e + 01
F21	>3.15e + 02	>2.81e + 02	>3.11e + 02	>2.93e + 02	2.32e + 02
F22	>1.24e + 02	>7.56e + 02	>7.18e + 03	<1.09e + 02	1.11e + 02
F23	>4.84e + 03	<3.60e + 03	>6.74e + 03	<2.54e + 03	3.89e + 03
F24	<2.27e + 02	<2.23e + 02	>6.74e + 02	<2.10e + 02	2.28e + 02
F25	>2.65e + 02	<2.56e + 02	>3.27e + 02	<2.40e + 02	2.63e + 02
F26	>2.03e + 02	>2.04e + 02	>5.10e + 02	=2.00e + 02	2.00e + 02
F27	>6.10e + 02	>6.13e + 02	>5.61e + 02	<3.01e + 02	5.57e + 02
F28	>3.00e + 02	>3.00e + 02	>1.14e + 03	>3.00e + 02	2.93e + 02
w/t/l	18 /3/7	13/1/14	21/0/7	4/3/21	

Table 9
Wilcoxon singed ranks test of 28 benchmark function.

Pairs of algorithm	R+	R-	n+	n-	Ties	p-value
SAHLPSO VS SaDE	294	106	18	7	3	1.8341e−02
SAHLPSO VS CoDE	161.5	243.5	13	14	1	8.3345e−01
SAHLPSO VS CMA-ES	303	103	21	7	0	1.0889e−02
SAHLPSO VS LSHADE	50	350	4	21	3	9.9984e−01

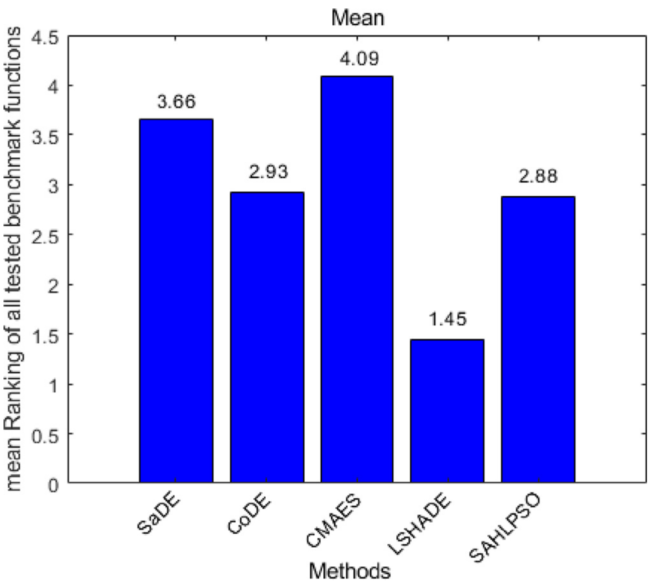

Fig. 10. Mean rankings of comparison with outstanding EAs in terms of Mean metrics.

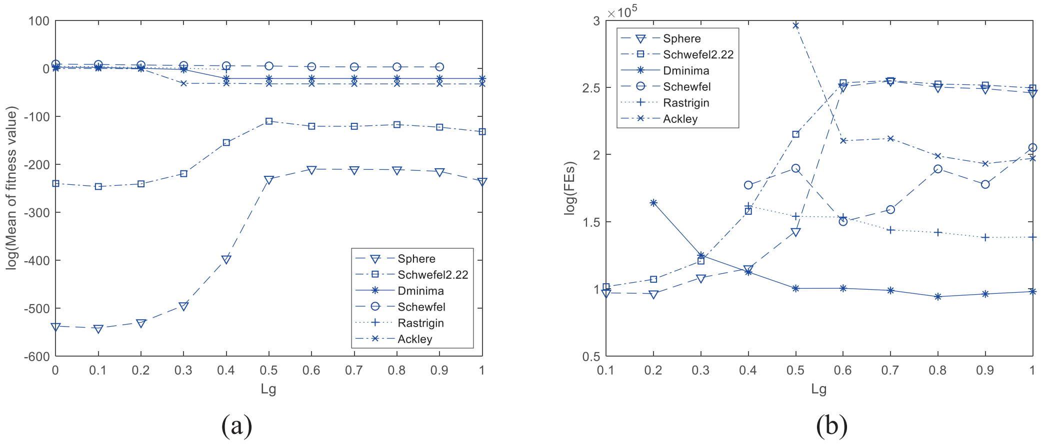

Fig. 11. The effect of *Lg* values of SAHLPSO on the solution accuracy and convergence rate.

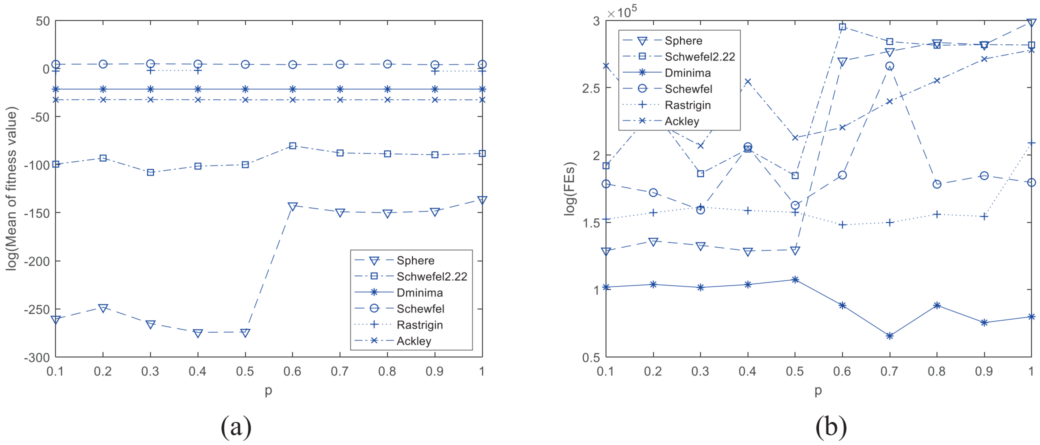

Fig. 12. The effect of *p* values of SAHLPSO on the solution accuracy and convergence rate.

CRediT authorship contribution statement

Xinmin Tao: Conceptualization, Methodology, Software. **Xiangke Li:** Data curation, Writing - original draft, Software. **Wei Chen:** Visualization, Investigation. **Tian Liang:** Writing – review & editing, Supervision. **Yetong Li:** Writing – review & editing, Validation. **Jie Guo:** Writing – review & editing. **Lin Qi:** Writing – review & editing.

Declaration of Competing Interest

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

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities no. 2572017EB02, Innovative talent fund of Harbin science and technology Bureau (No. 2017RAXXJ018), Double first-class scientific research foundation of Northeast Forestry University (411112438). We gratefully thank W.J., Guo and R, Liu who gave us helpful suggestions from many different points of view. In addition, we thank J., Cui., J.T., Wang and J.R., Zhou for many helpful discussions and persistent support of our work. The authors are grateful to the anonymous reviewers for their valuable comments and suggestions which were very helpful in improving the quality and presentation of this paper.

Appendix 1. The definition of the benchmark functions

Name	Definition	Domain	fmin
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[−100, 100]	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \Pi_{i-1}^{D} \lvert x_i \rvert$	[−10, 10]	0
High Conditioned Elliptic	$f_4(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} * x_i^2$	[−100, 100]	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1}))^2 + (x_i - 1)^2)$	[−10, 10]	0
Weierstrass	$f_3(x) = \sum_{i=1}^{D} \left(\sum_{k=0}^{kmax} \left[a^k \cos\left(2\pi b^k \cdot (x_i + 05)\right)\right]\right) - n$ $\cdot \sum_{k=0}^{kmax} \left[a^k \cos\left(2\pi b^k \cdot 0.5\right)\right], a = 0.5, b = 3, kmax = 20$	[−0.5, 0.5]	0
Dminima	$f_4(x) = 78.332331408 + \sum_{i=1}^{D} \frac{x_i^4 - 16x_i^2 + 5x_i}{D}$	[−5, 5]	0
Griewank	$f_5(x) = \sum_{i=1}^{D} {x_i}^2/4000 - \prod_{i=1}^{D} \cos\left(x_i/\sqrt{i}\right) + 1$	[−600, 600]	0
Schewfel	$f_6(x) = 418.982887273 \times D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	[−500, 500]	0
Rastrigin	$f_7(x) = \sum_{i=1}^{D} ({x_i}^2 - 10\cos(2\pi x_i) + 10)$	[−5, 5]	0
Ackley	$f_9(x) = -20 \exp\left(-0.2\sqrt{\sum_{i=1}^{D} \frac{x_i^2}{D}}\right) - \exp\left(\sum_{i=1}^{D} \cos(2\pi x_i)/D\right) + 20 + e$	[−32, 32]	0

References

[1] M.J. Amoshahy, M. Shamsi, M.H. Sedaaghi, Y. Deng, A novel flexible inertia weight particle swarm optimization algorithm, PLOS One. 11 (8) (2016) e0161558, https://doi.org/10.1371/journal.pone.0161558.

[2] 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) (2004) 240–255.

[3] A.P. Lin, Global genetic learning particle swarm optimization swarm optimization with diversity enhancement by ring topology, Swarm Evol. Comput. 44 (2019) 571–583.

[4] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.

[5] B. Wei, X. Xia, F. Yu, Y. Zhang, X. Xu, H. Wu, L. Gui, G. He, Multiple adaptive strategies based particle swarm optimization algorithm, Swarm Evol. Comput. 57 (2020) 100731.

[6] C. Xu, H.G. Tian, W.L. Du, Bee-foraging learning particle swarm optimization, Swarm, Evol. Comput. 102 (2021) 107–134.

[7] D.O. Tian, Z.Z. Shi, MPSO: Modified particle swarm optimization and its applications, Swarm Evol. Comput. 41 (2018) 49–68.

[8] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, X.-L. Shen, A hybrid particle swarm optimization algorithm using adaptive learning strategy, Inf. Sci. 436-437 (2018) 162–177.

[9] G.P. Xu, Particle swarm optimization base on dimensional learning strategy, Swarm Evol. Comput. 45 (2019) 33–51.

[10] G. Wu, D. Qiu, Y. Yu, W. Pedrycz, M. Ma, H. Li, Superior solution guided particle swarm optimization combined with local search techniques, Expert Syst. Appl. 41 (16) (2014) 7536–7548.

[11] H. Pang, F. Liu, Z. Xu, Variable universe fuzzy control for vehicle semi-active suspension system with MR damper combining fuzzy neural network and particle swarm optimization, Neurocomputing 306 (2018) 130–140.

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

[13] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Trans. Cybern. 47 (9) (2017) 2664–2677.

[14] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. IEEE Int. Conf. Neural Netw, (1995) 1942–1948..

[15] J.O. Pedro, M. Dangor, O.A. Dahunsi, M.M. Ali, Dynamic neural network-based feedback linearization control of full-car suspensionsusing pso, Appl. Soft. Comput. 70 (2018) 723–736.

[16] J. Han, G. Zhang, Y. Hu, J. Lu, A solution to bi/tri-level programming problems using particle swarm optimization, Inf. Sci. 370-371 (2016) 519–537.

[17] J. Li, Y. Zhang, X. Chen, Y. Xiang, Secure attribute-based data sharing for resource-limited users in cloud computing, Comput Secur. 72 (2018) 1–12.

[18] J. Jeslin Drusila Nesamalar, P. Venkatesh, S. Charles Raja, Managing multi-line power congestion by using Hybrid Nelder–Mead–Fuzzy Adaptive Particle Swarm Optimization (HNM-FAPSO), Appl. Soft. Comput. 43 (2016) 222–234.

[19] 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) (2006) 281–295.

[20] J.J. Liang, P. Definitions, E. Criteria, for the CEC, Special Session on Realparameter Optimization, Technical report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, 2013, Nanyang Technological University, Singapore, 2013.

[21] K. Chen, F. Zhou, L. Yin, S. Wang, Y. Wang, F. Wan, A hybrid particle swarm optimizer with sine cosine acceleration coefficients, Inf. Sci. 422 (2018) 218–241.

[22] K. Zhang, Q. Huang, Y. Zhang, Enhancing comprehensive learning particle swarm optimization with local optima topology, Inf. Sci. 471 (2019) 1–18.

[23] L. Cao, L. Xu, E.D. Goodman, A neighbor-based learning particle swarm optimizer with short-term and long-term memory for dynamic optimization problems, Inf. Sci. 453 (2018) 463–485.

[24] L.T. Al-Bahrani, J.C. Patra, a novel orthogonal PSO algorithm based on orthogonal diagonalization, Swarm Evol. Comput. 40 (2018) 1–23.

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

[26] 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.

[27] M.S. Nobile, P. Cazzanign, D. Besozzi, Fuzzy Self-tuning PSO: A setting-free Aglorithm for Global Optimization, Swarm Evol. Comput. 39 (2017).

[28] M.R. Tanweer, S. Suresh, N. Sundararajan, Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems, Inf. Sci. 326 (2016) 1–24.
[29] M.R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self-regulating particle swarm optimization algorithm, Swarm Evol. Comput. 28 (2016) 98–116.
[30] N. Kalaiarasi, S. Paramasivam, S.S. Dash, P. Sanjeevikumar, L. Mihet-Popa, PSO based MPPT implementation in dspace controller integrated through z-source inverter for photovoltaic applications, Energies 9 (2017) 1–10.
[31] N. Lynn, M.Z. Ali, P.N. Suganthan, Population topologies for particle swarm optimization and differential evolution, Swarm Evol. Comput. 39 (2018) 24–35.
[32] N. Lynn, P.N. Suganthan, Ensemble particle swarm optimizer, Appl. Soft Comput. 55 (2017) 533–548.
[33] N.-J. Li, W.-J. Wang, C.-C. James Hsu, Hybrid particle swarm optimization incorporating fuzzy reasoning and weighted particle, Neurocomputing. 167 (2015) 488–501.
[34] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhancing exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[35] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[36] Q. Zhang, W. Liu, X. Meng, B. Yang, A.V. Vasilakos, Vector coevolving particle swarm optimization algorithm, Inf. Sci. 394-395 (2017) 273–298.
[37] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. 291 (2015) 43–60.
[38] R. Mendes, J. Kennedy, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (3) (2004) 204–210.
[39] R. Tanabe, A.S. Fukunaga, Improving the Search Performance of SHADE Using Linear Population Size Reduction [C], IEEE Congress on Evolutionary Computation (CEC), July 6–11, 2014, Beijing, China, 2014.
[40] W.H. Lim, N.A. Mat Isa, An adaptive two-layer particle swarm optimization with elitist learning strategy, Inf. Sci. 273 (2014) 49–72.
[41] 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 (2018) 126–140.
[42] X. Tao, W. Guo, Q. Li, C. Ren, R. Liu, Multiple scale self-adaptive cooperation mutation strategy-based particle swarm optimization, Appl. Soft. Comput. 89 (2020) 106124, https://doi.org/10.1016/j.asoc.2020.106124.
[43] X. Xia, Y. Xing, B. Wei, Y. Zhang, X. Li, X. Deng, L. Gui, A fitness-based multi-role particle swarm optimization, Swarm Evol. Comput. 44 (2019) 349–364.
[44] X. Zhang, X. Wang, Q. Kang, J. Cheng, Differential mutation and novel social learning particle swarm optimization algorithm, Inform. Sci. 480 (2019) 109–129.
[45] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, Evolutionary computation, CEC99, Proceeding of the 1999 Congress 1 (1999) 320–324.
[46] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H.S.-H. Chung, Y.-H. Shi, J. Zhang, Genetic learning particle swarm optimization, IEEE Trans. Cybern. 46 (10) (2016) 2277–2290.
[47] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–66.