

## Research Article

# A Novel Particle Swarm Optimization with Improved Learning Strategies and Its Application to Vehicle Path Planning

En Lu <sup>1,2</sup>, Lizhang Xu <sup>1</sup>, Yaoming Li,<sup>1</sup> Zheng Ma,<sup>1</sup> Zhong Tang,<sup>1</sup> and Chengming Luo<sup>3</sup>

<sup>1</sup>School of Agricultural Equipment Engineering, Jiangsu University, Zhenjiang 212013, China

<sup>2</sup>World Precise Machinery (China) Co., Ltd., World Industrial Park, Picheng Town, Danyang 212311, China

<sup>3</sup>College of Internet of Things Engineering, Hohai University, Changzhou 213022, China

Correspondence should be addressed to Lizhang Xu; [justxlz@ujs.edu.cn](mailto:justxlz@ujs.edu.cn)

Received 18 August 2019; Revised 8 October 2019; Accepted 5 November 2019; Published 22 November 2019

Academic Editor: Erik Cuevas

Copyright © 2019 En Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to balance the exploration and exploitation capabilities of the PSO algorithm to enhance its robustness, this paper presents a novel particle swarm optimization with improved learning strategies (ILSPSO). Firstly, the proposed ILSPSO algorithm uses a self-learning strategy, whereby each particle stochastically learns from any better particles in the current personal history best position (*pbest*), and the self-learning strategy is adjusted by an empirical formula which expresses the relation between the learning probability and evolution iteration number. The cognitive learning part is improved by the self-learning strategy, and the optimal individual is reserved to ensure the convergence speed. Meanwhile, based on the multilearning strategy, the global best position (*gbest*) of particles is replaced with randomly chosen from the top *k* of *gbest* and further improve the population diversity to prevent premature convergence. This strategy improves the social learning part and enhances the global exploration capability of the proposed ILSPSO algorithm. Then, the performance of the ILSPSO algorithm is compared with five representative PSO variants in the experiments. The test results on benchmark functions demonstrate that the proposed ILSPSO algorithm achieves significantly better overall performance and outperforms other tested PSO variants. Finally, the ILSPSO algorithm shows satisfactory performance in vehicle path planning and has a good result on the planned path.

## 1. Introduction

The general optimization problem is to find strategies or a set of parameter values, such that one- or multicriteria attain to the optimum under some restricted conditions. Research studies on optimization are general problems in every field of engineering technology, science research, natural world, and human society activities. With the development of production becoming deeper and deeper, traditional optimization algorithms may no longer satisfy with the request for the actual application owing to the limit of performance, and consequently entail new rapid and efficient algorithms [1]. Over the last decades, researchers have designed a lot of feasible and effective heuristic intelligent optimization algorithms for optimization problems based on simulating all kinds of organism in nature. Intelligent optimization algorithms are a typical heuristic algorithm, and they can be

divided into two types according to the number of generated solutions in each iteration [2]. One is individual-based algorithms, such as Simulated Annealing (SA) [3], Mountain Climb Servo (MCS) [4], and Tabu Search (TS) [5]. Another is population-based algorithms, such as Genetic Algorithms (GAs) [6], Ant Colony Optimization (ACO) [7], Particle Swarm Optimization (PSO) [8], and Artificial Bee Colony (ABC) [9]. The abovementioned algorithms have been successfully implemented in the practical application.

The PSO algorithm is a relatively new embranchment of swarm intelligence. The PSO algorithm was proposed in 1995 by Kennedy and Eberhart [8], and its core idea is to use information sharing mechanism and learning from each other particles' experience, particle experience from each other, the development of population is achieved. It is characterized by simplistic implementation, fast convergence, and few parameters. As a result, the PSO algorithm is

developing rapidly in recent years and has been successfully applied in various scientific and engineering fields, such as in water distribution network design [10], power systems [11], manipulator motion planning [12], optimal control [13], image processing [14], artificial neural networks [15], and other fields [16–19]. The velocity and position of each particle in the basic PSO algorithm are updated through its own personal history best position (*pbest*) and global best position (*gbest*). The PSO algorithm does not use the external information during the search process and only utilizes the fitness value of the function. In the application to solving the complicated large-scale combinatorial optimization problems, the basic PSO algorithm usually tends to converge prematurely and lacks quality solution because of the insufficient diversity of individuals [20]. Besides, the algorithm converges prematurely in high dimension optimization problems because of an intense spread of information flow to attract particles to the *gbest* in the basic PSO algorithm. Therefore, the basic PSO algorithm has some inherent shortcomings which are hard to overcome and in need of a modified PSO algorithm. Numerous PSO variants have been proposed to enhance the searching performance, which can be roughly divided into the following categories, the adaptive control of the parameters, new learning strategies to enhance population diversity, hybridization with other algorithms, and direct control of the probability distribution of the population. Moreover, other researchers proposed some improved PSO algorithms for specific application environments. However, these PSO variants often maintain population diversity with the cost of slow convergence speed or complex algorithmic structures. Thus, it is still a challenging task to synthetically improve the performance of the PSO in the algorithm research. This article proposed a new PSO variant by further modifying the velocity update mechanism in the PSO algorithm and named PSO with improved learning strategies (ILSPSO). This new PSO variant employs the self-learning strategy and multilearning strategy to increase the convergence speed and global searching capability of the PSO algorithm. Then, the performances of ILSPSO are tested by 16 well-known benchmark functions and compared with five representative PSO variants. The proposed ILSPSO algorithm is superior to the other tested PSO variants in the majority of the benchmark functions.

In recent years, the agricultural equipment industry has developed rapidly in China. As one of the key technologies of autonomous navigation, path planning is the research focus of intelligent agricultural equipment. According to the application purpose, the path planning of mobile agricultural vehicles can be divided into optimal time, optimal path, and so on. Under known environmental information, according to the prebuilt environment model, the path planning from the initial location to the target location is called global path planning. The advantage is that the optimality and accessibility of the planning path are guaranteed. The optimal path search algorithm for global path planning can be mainly divided into graph search algorithm, random sampling algorithm, intelligent algorithm, and so on [21]. GA and ACO in the intelligent algorithm have potential

parallelism, which is suitable for solving and optimizing complex problems, but there are problems such as slow operation speed and premature solution. Therefore, the PSO algorithm with simple and fast convergence has broad application prospects in the path planning of mobile vehicles.

This article aims to propose a new PSO variant by further modifying the velocity update mechanism in PSO and employ it to achieve the vehicle path planning. The article is organized as follows. The basic PSO algorithm and several PSO-related studies are briefly discussed in Section 2. Section 3 elucidates the methodologies of the ILSPSO algorithm. Subsequently, to verify the performance of the proposed ILSPSO algorithm, the experimental settings and simulation results are given in Section 4. Section 5 applies the proposed ILSPSO algorithm to the vehicle path planning. Finally, the article is concluded with a brief summary in Section 6.

## 2. Related Works

Firstly, this section discusses the mechanism of the basic PSO algorithm. Then, the PSO variants are reviewed according to the different improvement ideas of scholars.

**2.1. Basic PSO Algorithm.** In the basic PSO algorithm, each particle has its own trajectory, namely, position  $x_i$  and velocity  $v_i$  and moves in the search space. The position of each particle represents the potential solution, and it is evolved with regard to the fitness function. In  $D$ -dimensional search space, the position and velocity of  $i$ th particle are expressed as  $X_i = [x_i^1, x_i^2, \dots, x_i^D]$  and  $V_i = [v_i^1, v_i^2, \dots, v_i^D]$ , respectively. During the search process, the trajectory of  $i$ th particle is adjusted according to the current velocity, *pbest*<sub>*i*</sub> and *gbest*. Among them,  $pbest_i = [pbest_i^1, pbest_i^2, \dots, pbest_i^D]$  and  $gbest = [gbest^1, gbest^2, \dots, gbest^D]$ . Therefore, the velocity and position of  $i$ th particle on the dimension  $d$  are updated by using equations (1) and (2) as follows [22]:

$$v_i^d(t+1) = w \cdot v_i^d(t) + c_1 \cdot \text{rand}_1^d \cdot (pbest_i^d(t) - x_i^d(t)) + c_2 \cdot \text{rand}_2^d \cdot (gbest^d(t) - x_i^d(t)), \quad (1)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1), \quad (2)$$

where  $w$  is the inertia weight which is used to measure the percentage of the previous velocity,  $c_1$  and  $c_2$  are the acceleration parameters,  $\text{rand}_1^d$  and  $\text{rand}_2^d$  are two uniformly distributed random numbers in the range of  $[0, 1]$ ,  $pbest_i^d$  is the personal history best position of  $i$ th particle, and  $gbest^d$  is the global best position of the particle. The 1st term in equation (1) represents the effect of the inertia of the particle, the 2nd term represents the cognitive learning (particle memory influence), and the 3rd term represents the social learning (swarm influence). It would be worthwhile to point out that, in general, the social aspect of the PSO algorithm is guided by the best solution in the particle's neighborhood.

**2.2. PSO Variants and Improvements.** Since the introduction of the PSO algorithm, researchers have studied it from different methods and angles to support the complex multimodal function optimization requirements and proposed many PSO variants. A brief review of these variants is presented as follows.

The adaptive control of the parameters is one of the widely used strategies to enhance the searching performance of the basic PSO algorithm. As a global parameter of the PSO algorithm, inertia weight  $w$  plays an important role in the algorithm, and it can easily control searching capability and convergence speed of the algorithm. An inertia weight  $w$  within  $[0.9, 1.2]$  was found to be a good choice, but the strategy of linearly decreasing inertia weight can also help faster convergence [22]. Shi and Eberhart proposed a fuzzy adaptive method to modify inertia weight  $w$  [23]. A novel parameter automation strategy for the acceleration coefficients  $c_1$  and  $c_2$  (HPSO-TVAC) is introduced by Ratnaweera et al. [24]. Tang et al. proposed a feedback learning PSO with quadratic inertia weight (FLPSO-QIW), and the parameters are controlled by introducing a fitness feedback mechanism [25]. Xu developed a new strategy in which the inertia weight is dynamically adjusted according to the mean absolute value of velocity [26]. In the CDW-PSO algorithm, Chen et al. proposed a chaotic map and dynamic weight is introduced to modify the search process [27]. Overall, adjusting parameters can improve the performance of the PSO algorithm, but this strategy is mainly temporary [28].

Another category of PSO variants introduces new learning strategies to enhance population diversity. Peram et al. proposed the fitness-distance-ratio-based PSO (FDR-PSO) algorithm adds a new dimension to this approach; each particle also learns from the experience of the neighboring particles that have a better fitness than itself [29]. Mendes et al. developed a fully informed PSO (FIPS) in which each particle is updated based on the history best positions of several neighbors instead of  $gbest$  or  $pbest$  [30]. Liang et al. proposed comprehensive learning PSO (CLPSO) by learning from different personal history best positions [31]. A distance-based locally informed particle swarm optimizer is proposed by Qu et al. [32]. Sabat et al. proposed a novel integrated learning PSO (ILPSO) for optimizing complex multimodal functions [33]. Cheng and Jin aimed to improve the learning efficiency of particles by introducing social learning mechanisms into PSO to develop a social learning PSO (SL-PSO) [34]. A self-regulating PSO (SRPSO) algorithm is incorporating the best human learning strategies for finding the optimum solution proposed by Tanweer et al. [35]. To solve unconstrained single-objective optimization problems with continuous search space, Lim and Mat Isa proposed a new PSO variant with adaptive division of labor (ADOLPSO) [20]. Wang et al. developed a hybrid PSO algorithm which employs self-learning based on the candidate generation strategy to ensure the exploration ability, and competitive learning-based prediction strategy to guarantee exploitation of the algorithm [36].

The Hybrid PSO algorithm by combining PSO with other search strategies to enhance performance can be

classified as the third category of PSO variants. Lvbjerg et al. added crossover operation which is one of the three basic genetic operators used in the genetic algorithm to increase the population diversity [37]. Liu et al. have recently proposed a novel algorithm (PSO-DE) which integrates PSO with differential evolution (DE) to solve constrained numerical and engineering optimization problems [38]. A diversity-guided hybrid PSO based on gradient search is proposed by Han and Liu [39]. Other specific search operators have also been introduced into PSO, including repulsion techniques [40], variable neighborhood search [41], local stochastic search [42], Newton's laws of motion [43], multicrossover mechanism of the GA, and bee colony mechanism [44]. In general, the hybrid PSO algorithm improves the performance of the basic PSO algorithm, but they are difficult to implement and generally requires more time to calculate than the basic PSO algorithm.

The PSO algorithm is a stochastic global optimization technique, and the searching performance is closely related to the probability distribution of the population. Therefore, the algorithm can be improved through the direct control of the probability distribution of the population. Kennedy analyzed the distribution of particles with the change of iterations and proposed Bare Bones PSO (BBPSO) based on probability distribution [45]. In the literature [46], Kennedy summarized the previous studies, and the algorithm is decomposed into its essential steps and the alternative interpretations of those steps are studied. Based on Kennedy's research, Lin and Feng replaced the normal distribution with chaotic search which can avoid the influence of pseudo-random number [47]. Mendes and Kennedy improved the selection way of the center of the normal distribution function, explored the mean of the personal best positions of several particles in the same neighborhood and the  $\beta$ -distribution is used to define the mean of the Gaussian sample. [48]. Zhao and Suganthan proposed two local bests based multiobjective PSO which focuses the search around small regions in the parameter space in the vicinity of the best existing fronts and applies different mutation operators to different subswarms to accelerate the convergence [49].

### 3. ILSPSO Algorithm

Although many PSO variants are reported in the related literatures (as shown in Section 2.2), they still have some problems in the algorithm implementation or lack of quality solution. In this section, a novel particle swarm optimization with improved learning strategies (ILSPSO) is proposed to further modify the cognitive learning strategy and social learning strategy in the basic PSO algorithm. In the ILSPSO algorithm, the self-learning strategy and multilearning strategy are adopted to overcome the shortcomings and improve the performance of the basic PSO algorithm for optimizing higher dimension multimodal objective functions.

**3.1. Self-Learning Strategy.** In the basic PSO algorithm, each particle only learns from its  $pbest$  in the cognitive learning part (the 2nd term in equation (1)). The  $pbest$  represents the

personal history best position for each particle. However,  $pbest$  is not always better in the whole particle swarm within the current evolutionary search space. As a result, the potential information of the better particles is not efficiently used.

In view of the above problem, this section presents a self-learning strategy to improve  $pbest$  in cognitive learning part of the basic PSO algorithm. In order to evaluate the  $pbest$ , the  $pbest$  is sorted according to the fitness values. Subsequently, the  $pbest$  of the  $i$ th particle, except for the best one, will stochastic learn from any better particles in the current personal best solutions. After the self-learning strategy, the  $pbest$  of the  $i$ th particle on the dimension  $d$  can be expressed as  $pbest_{si}^d$ . Therefore, the  $pbest_i^d$  of the cognitive learning part of equation (1) can be replaced by the  $pbest_{si}^d$ .

From the various simulation experiments, it can be seen that the fixed self-learning probability is bad for population diversity [31]. For achieving better population diversity and faster convergence speed, the empirically self-learning probability is adopted:

$$P_i = 0.3 + 0.6 \cdot \frac{e^{(5(i-1)/M-1)} - 1}{e^5 - 1}. \quad (3)$$

Figure 1 presents an example of assigned for an evolution iteration number of 100000. The self-learning probability from 1 to 100000 has a value ranging from 0.3 to 0.9. Finally, the self-learning description can be seen in Algorithm 1, where  $M$  is the population size and  $D$  is the search dimensionality.

**3.2. Multilearning Strategy.** In addition, learning from a single  $gbest$  in the social learning part (the 3rd term in equation (1)) improves the convergence speed in the basic PSO algorithm. Yet, the population diversity of particles is reduced and easily attracted to the local optimal region. Hence, the basic PSO algorithm often falls into local optimum in the process of solving higher dimension multimodal objective functions. For the heuristic swarm intelligence optimization algorithms including PSO, the crux of optimization is that the algorithm should not only provide good local searching capability, but also with strong global searching capability. Generally, the global searching capability of the PSO algorithm is more important [50] because only the global searching capability can search through the total solution space and find the globally optimal solution over a domain. The global searching capability of the PSO algorithm is close to the diversity of population. Therefore, after self-learning of  $pbest$ , the global searching capability should be enhanced in the improved PSO algorithm.

In order to balance the exploration and exploitation capabilities of the PSO algorithm and enhance its robustness, multilearning strategy is adopted to improve the social learning part in the process of velocity updating (the 3rd term in equation (1)), which will ultimately achieve rapid convergence speed and strong global searching capability in looking for answers. Assuming  $gbestd = [gbest_1^d, gbest_2^d, \dots, gbest_k^d]$ ,  $gbestd$  indicates the top  $k$  of  $gbest$  on the  $d$  dimension, and  $k$  determines the greediness of the novel

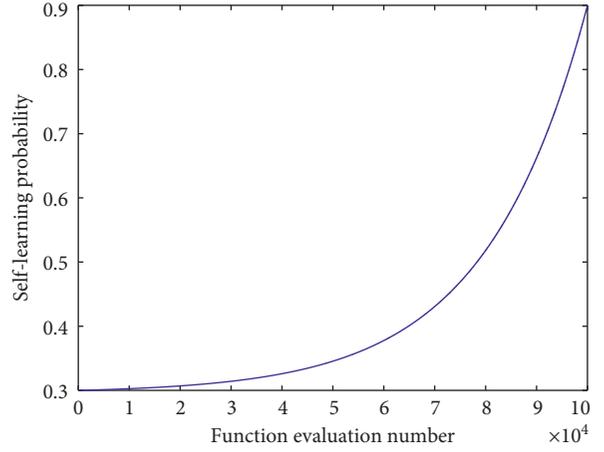


FIGURE 1: Self-learning probability.

multilearning strategy. Therefore, the  $gbest$  on the  $d$  dimension can be expressed as  $gbestk^d = gbestd(unidrnd(k))$ , and function  $unidrnd(k)$  generates random numbers for the discrete uniform distribution with maximum  $k$ . After the multilearning strategy, the  $gbest$  of the  $i$ th particle on the dimension  $d$  can be expressed as  $gbestk^d$ . Therefore, the  $gbest^d$  of the social learning part of equation (1) can be replaced by the  $gbestk^d$ .

The multilearning strategy can be described by the following Figure 2 [33]. In Figure 2, the orange dots represent  $gbestd = [gbest_1^d, gbest_2^d, \dots, gbest_k^d]$  and the  $gbestd$  are surrounded by a circle which shows the probable learning region of particles and the black dots represent the position of particles in the evolutionary search space. The  $gbestd$  represents the multiple  $gbest$  solutions on the corresponding  $d$  dimension which include the current  $gbest$  solution, and the number is decided by the parameter  $k$  ( $k=3$  in Figure 2). For the particles on the dimension  $d$ , the role of the current  $gbest^d$  solution can be replaced through randomly selecting from the  $gbestd$  according to the multilearning strategy. That is, all particles move toward the  $gbestk^d$  ( $gbestk^d = gbestd(unidrnd(k))$ ) in the social learning part (the 3rd term in equation (1)). In this way, the multilearning strategy not only can keep the population diversity but also avoid premature convergence.

**3.3. Proposed Approach.** Based on Sections 3.1 and 3.2, the velocity updating equation (1) can be rewritten as follows:

$$v_i^d(t+1) = w \cdot v_i^d(t) + c_1 \cdot \text{rand}_1^d \cdot (pbest_{si}^d(t) - x_i^d(t)) + c_2 \cdot \text{rand}_2^d \cdot (gbestk^d(t) - x_i^d(t)), \quad (4)$$

where  $pbest_{si}^d(t)$  is obtained through the self-learning strategy and  $gbestk^d(t)$  is obtained through the multilearning strategy. The position updating of  $i$ th particle on the dimension  $d$  still uses equation (2). In each generation, iteration procedure of any particle in the ILPSO algorithm is illustrated in Figure 3 [16].

The proposed ILPSO algorithm employs the improved learning strategies (self-learning strategy and multilearning

```

(1) Sort  $pbest$  according to fitness values of  $pbest$ ;
(2) For  $i = 1$  to  $M$  do
(3)  $P_i = 0.3 + 0.6 \cdot (e^{(5(i-1)/M-1)} - 1/e^5 - 1)$ ; //refer to equation (3)
(4) If  $P_i > rand$  then
(5) For  $d = 1: D$  do
(6)  $pbest_{si}^d(t) = pbest(i + fix(rand \cdot (M - i)), d + fix(rand \cdot (D - d)))$ ; //function  $fix(\dots)$  rounds the element to the nearest integer toward zero
(7) End For
(8) End if
(9) End For
    
```

ALGORITHM 1: Self-learning strategy.

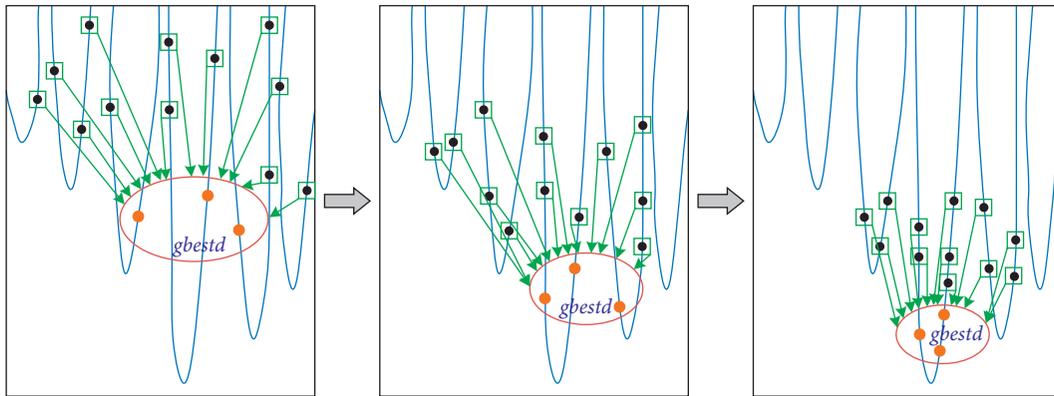


FIGURE 2: Schematic diagram of multilearning strategy.

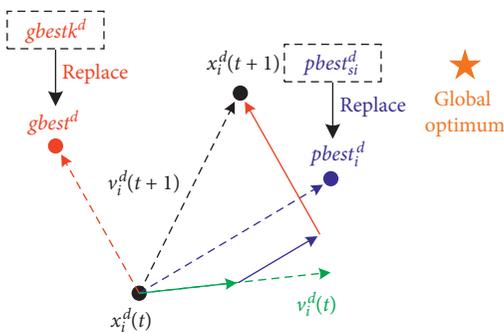


FIGURE 3: Iteration scheme of the particles in the ILSPSO algorithm.

strategy) to improve the performance of the basic PSO algorithm. The self-learning strategy aims to increase the convergence speed, and the multilearning strategy aims to enhance the global exploratory capability. Algorithm 2 shows the detail steps of the ILSPSO algorithm (Max\_gen is the maximum evolution number).

#### 4. Experimental Settings and Simulation Results

In this section, firstly, the experimental settings used in the performance evaluation of the proposed ILSPSO algorithm

are described. Subsequently, the simulation results have been displayed.

4.1. Benchmark Functions. A total of 16 well-known benchmark functions are used to extensively test the performances of the algorithms. The involved benchmark functions, which are listed in Table 1, can be divided into three classes. F1–F4 are unimodal functions and called conventional problems. They only have a single optimum. Hence, the exploitation capability of algorithms can be tested through these unimodal functions. F5–F8 are complex multimodal functions and called complex problems. They have one global optimum and many local optimum, and the local optimum number of the functions increase exponentially with increasing problem dimension. Thus, they are suitable for testing the exploration capability of algorithms. Moreover, it is very important to obtain good results on multimodal functions for the optimization algorithms because these good results show that the algorithm can jump out from the local optimum and gradually move toward the global optimum. Finally, F9–F16 are the rotated version of the F1–F8 and called rotated problems [51]. These rotated functions are complex and can be used to test the exploitation and exploration capabilities of algorithms at the same time. The involved benchmark functions are listed in Table 1, which consists of their formulae, feasible search ranges, global optimal values, and accuracy levels ( $\epsilon$ ) [33, 52, 53].

```

(1) /* Initialization */
(2) Initialize  $M, D, \text{gen}, \text{Max\_gen}, w, c_1, c_2, k$ 
(3) Initialize  $X_{\min}, X_{\max}, X(0), V_{\min}, V_{\max}, V(0)$ 
(4) Evaluate fitness values  $f(0)$ 
(5)  $pbest(0) \leftarrow X(0), fpbest(0) \leftarrow f(0), gbest(0) \leftarrow Xbest(0), fgbest(0) \leftarrow fbest(0)$ 
(6) /* Main loop */
(7) while  $\text{gen} \leq \text{Max\_gen}$  do
(8)   Sort  $X(t), V(t)$  according to fitness values  $f(t)$ ;
(9)   Sort  $pbest(t)$  according to fitness values  $fpbest(t)$ ;
(10)  For  $i = 1$  to  $M$  do
(11)    For  $d = 1:D$  do
(12)      /* Self-learning strategy, refer to Algorithm 1 */
(13)      Calculate the  $pbest_{si}^d$  of  $i$ th particle according to Algorithm 1;
(14)      /* Multilearning strategy */
(15)      Calculate the  $gbestk^d$  of  $i$ th particle according to the top  $k$  of  $gbest$ ;
(16)      /* Update the velocity and position of  $i$ th particle */
(17)      Calculate the velocity  $v_i^d$  of  $i$ th particle using equation (4);
(18)      Limit the velocity using  $v_i^d = \min(v_{\max}^d, \max(v_{\min}^d, v_i^d))$ ;
(19)      Update the position  $x_i^d$  of  $i$ th particle using equation (2);
(20)      Limit the position using  $x_i^d = \min(x_{\max}^d, \max(x_{\min}^d, x_i^d))$ ;
(21)    End For
(22)    Calculate the fitness value  $f(t)$ ;
(23)    Update the personal best solution  $pbest(t), fpbest(t)$ ;
(24)    Update the global best solution  $gbest(t), fgbest(t)$ ;
(25)  End For
(26)   $\text{gen}++$ ;
(27) End While
(28) Output the final optimal solution.

```

ALGORITHM 2: The pseudocode of the proposed ILSPSO algorithm.

**4.2. Parameter Settings for the Involved PSO Variants.** In this study, five representative PSO variants have been chosen to thoroughly compare with the proposed ILSPSO algorithm. The parameter configurations of all involved PSO variants are chosen based on the suggestions of the relevant literatures and summarized in Table 2.

In order to ensure that all PSO variants have a fair comparison, they are tested using the same population size ( $M$ ), search dimensionality ( $D$ ), and maximum evolution number ( $Max\_gen$ ) in each run for each test function. Their values are 40, 30, and  $1 \times 10^4$ , respectively. Moreover, to minimize statistical errors, all the tested PSO variants are run 30 times independently for every function. The experiments have been conducted on a PC with an Intel Core i5-6600 3.3 GHz CPU, Microsoft Windows 10 64-bit operating system, and MATLAB 2015a.

**4.3. Comparison of ILSPSO with Other Representative PSO Variants.** In this study, the performance of the PSO variants are tested in terms of accuracy, reliability, and efficiency through the mean fitness value ( $F_{\text{mean}}$ ) and algorithm's rank (Rank). The  $F_{\text{mean}}$  and Rank results on 16 benchmark functions are listed in Table 3, and the best results obtained for each function are expressed in boldface. Besides, if  $F_{\text{mean}}$  statistically significantly outperforms the accuracy level, a "+" is labeled below the algorithm; if there is no significance between  $F_{\text{mean}}$  and the accuracy level, a "=" is labeled, and if  $F_{\text{mean}}$  is outperformed, a "-" is labeled. A summary of total

number of "+," "=", and "-" average rank, and overall rank are counted in the bottom of Table 3. The convergence chart of the tested PSO variants is shown in Figure 4. Table 3 and Figure 4 show that the proposed ILSPSO algorithm achieved better performance in the majority of test functions.

- (1) Conventional problems: as shown in Table 3, for the four conventional problems F1–F4, the proposed ILSPSO algorithm achieved the best performance on all of the four functions F1–F4. Also, the  $h_c$  also indicates that this dominance is significant statistically. Furthermore, the convergence chart of the tested PSO variants in Figures 4(c) and 4(d) show that the proposed ILSPSO algorithm has a faster convergence speed than other tested algorithms. In Figures 4(a) and 4(b), the convergence chart of the ILSPSO algorithm is second and the result is also satisfactory.
- (2) Complex problems: Table 3(F5–F8) and Figures 4(e)–4(h) show that the results of the ILSPSO algorithm are better than other tested PSO variants except on the function F7. In comparison, only the CLPSO has a better performance than the ILSPSO algorithm on the function F7. By the way, both them have the close results on the F7 as can be seen from Table 3(F7) and Figure 4(g). All the above illustrates that the proposed ILSPSO algorithm exhibits a satisfactory exploration capability on the complex multimodal functions. This

TABLE 1: Sixteen benchmark functions used in this study.

No.	Function name	Formulae	Search range	Optimum	$\epsilon$
<i>Category I: conventional problems</i>					
F1	Sphere	$\sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0	$1.0E-6$
F2	Schwefel 2.22	$\sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	$[-10, 10]^D$	0	$1.0E-6$
F3	Schwefel 2.21	$\max  x_i, 1 \leq i \leq D $	$[-100, 100]^D$	0	$1.0E-2$
F4	Quartic	$\sum_{i=1}^D i x_i^4 + \text{rand}$	$[-1.28, 1.28]^D$	0	$1.0E-2$
<i>Category II: complex problems</i>					
F5	Ackley	$-20 \exp(-0.2 \sqrt{(1/D) \sum_{i=1}^D x_i^2}) - \exp((1/D) \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$	$[-32, 32]^D$	0	$1.0E-2$
F6	Griewank	$(1/4000) \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^D$	0	$1.0E-2$
F7	Zakharov	$\sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5 i x_i)^2 + (\sum_{i=1}^D 0.5 i x_i)^4$	$[-100, 100]^D$	0	$1.0E-2$
F8	Weierstrass	$\sum_{i=1}^D (\sum_{k=0}^{k \max} [a^k \cos(2\pi b^k (x_i + 0.5))]) - D \sum_{k=0}^{k \max} [a^k \cos(2\pi b^k 0.5)]$ $a = 0.5, b = 3, k \max = 20$	$[-0.5, 0.5]^D$	0	$1.0E-2$
<i>Category III: rotated problems</i>					
F9	Rotated F1	$\sum_{i=1}^D y_i^2, y = M \times x$	$[-100, 100]^D$	0	$1.0E-6$
F10	Rotated F2	$\sum_{i=1}^D  y_i  + \prod_{i=1}^D  y_i , y = M \times x$	$[-10, 10]^D$	0	$1.0E-6$
F11	Rotated F3	$\max  y_i, 1 \leq i \leq D , y = M \times x$	$[-100, 100]^D$	0	$1.0E-2$
F12	Rotated F4	$\sum_{i=1}^D i y_i^4 + \text{rand}, y = M \times x$	$[-1.28, 1.28]^D$	0	$1.0E-2$
F13	Rotated F5	$-20 \exp(-0.2 \sqrt{(1/D) \sum_{i=1}^D y_i^2}) - \exp((1/D) \sum_{i=1}^D \cos(2\pi y_i)) + 20 + e,$ $y = M \times x$	$[-32, 32]^D$	0	$1.0E-2$
F14	Rotated F6	$(1/4000) \sum_{i=1}^D y_i^2 - \prod_{i=1}^D \cos(y_i/\sqrt{i}) + 1, y = M \times x$	$[-600, 600]^D$	0	$1.0E-2$
F15	Rotated F7	$\sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5 i x_i)^2 + (\sum_{i=1}^D 0.5 i x_i)^4, y = M \times x$	$[-100, 100]^D$	0	$1.0E-2$
F16	Rotated F8	$\sum_{i=1}^D (\sum_{k=0}^{k \max} [a^k \cos(2\pi b^k (y_i + 0.5))]) - D \sum_{k=0}^{k \max} [a^k \cos(2\pi b^k 0.5)]$ $a = 0.5, b = 3, k \max = 20, y = M \times x$	$[-0.5, 0.5]^D$	0	$1.0E-2$

TABLE 2: Parameter settings of the involved PSO algorithms.

Algorithm	Parameter settings
PSO-W [22]	$w_{\max} = 0.9, w_{\min} = 0.4, c_1 = 1.494, c_2 = 1.494, V_{\min} = -0.5 \times \text{range}, V_{\max} = 0.5 \times \text{range}$
GPSO [47]	$w = 0.729, c_1 = 1.494, c_2 = 1.494, P_c = 0.9, V_{\min} = -0.5 \times \text{range}, V_{\max} = 0.5 \times \text{range}$
FDR-PSO [29]	$w_{\max} = 0.9, w_{\min} = 0.4, c_1 = 1, c_2 = 1, c_3 = 2, V_{\min} = -0.5 \times \text{range}, V_{\max} = 0.5 \times \text{range}$
CLPSO [31]	$w = 0.729, c_1 = 1.494, c_2 = 1.494, V_{\min} = -0.5 \times \text{range}, V_{\max} = 0.5 \times \text{range}$
SL-PSO [34]	$w = 0.729, c_1 = 1.494, c_2 = 1.494, V_{\min} = -0.5 \times \text{range}, V_{\max} = 0.5 \times \text{range}$
ILSPSO	$w = 0.729, c_1 = 1.494, c_2 = 1.494, k = 3, V_{\min} = -0.5 \times \text{range}, V_{\max} = 0.5 \times \text{range}$

is mainly because the novel learning strategy enhances the global searching capability and the improvement strategy is effective in solving the most complex problems.

- (3) Rotated problems: in F9–F16, the rotation operation imposed on the traditional functions increases the complexity of functions. Nevertheless, Table 3(F9–F16) and Figures 4(i)–4(p) reflect the relative

robustness of the ILSPSO algorithm in most of the problems and significantly outperforms the other PSO variants in F9, F10, and F12–F16. Hence, the proposed ILSPSO algorithm has better exploitation and exploration capabilities than other PSO variants.

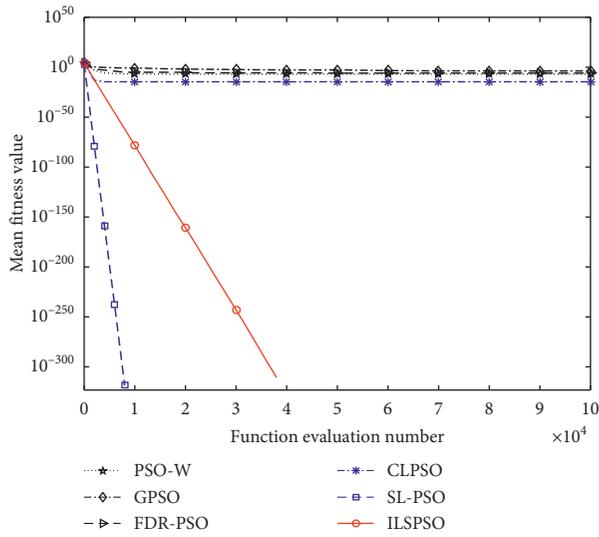
In summary, the analysis results of Table 3 and Figure 4 show that the proposed ILSPSO algorithm significantly improves the PSO’s performance and better than other

TABLE 3:  $F_{\text{mean}}$ ,  $h_c$  and rank results on 16 benchmark functions (bold values represent the best empirical fitness values in all the tested PSO variants).

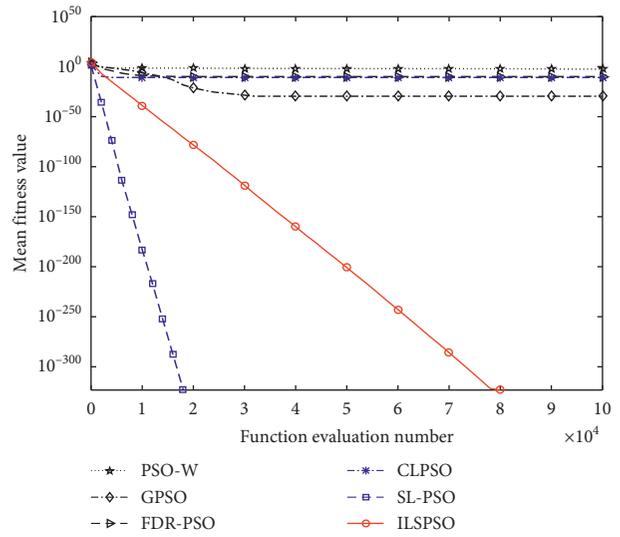
Function		PSO-W	GPSO	FDR-PSO	CLPSO	SL-PSO	ILSPSO	$\epsilon$
F1	$F_{\text{mean}}$	1.409913E-07	1.827236E-04	1.395674E-06	2.877963E-15	<b>0.000000E+00</b>	<b>0.000000E+00</b>	1.0E-6
	$h_c$	=	—	—	=	+	+	
	Rank	4	6	5	3	1	1	
F2	$F_{\text{mean}}$	3.435390E-03	2.737578E-30	1.142845E-10	1.618731E-11	<b>0.000000E+00</b>	<b>0.000000E+00</b>	1.0E-6
	$h_c$	—	+	=	=	+	+	
	Rank	6	3	5	4	1	1	
F3	$F_{\text{mean}}$	3.517346E-01	7.430126E-02	3.260374E-03	8.635655E-01	4.756715E-109	<b>5.526265E-119</b>	1.0E-2
	$h_c$	—	—	=	—	+	+	
	Rank	5	4	3	6	2	1	
F4	$F_{\text{mean}}$	3.684153E-04	2.460653E-04	1.638116E-05	2.964027E-03	5.370374E-03	<b>8.225893E-06</b>	1.0E-2
	$h_c$	=	=	=	=	=	=	
	Rank	4	3	2	5	6	1	
F5	$F_{\text{mean}}$	4.545800E+00	1.934159E-02	5.030825E-05	9.254524E-09	2.664535E-12	<b>2.664535E-15</b>	1.0E-2
	$h_c$	—	—	=	=	=	+	
	Rank	6	5	4	3	2	1	
F6	$F_{\text{mean}}$	7.432450E-03	1.330222E-02	3.450061E-06	4.357625E-13	<b>0.000000E+00</b>	<b>0.000000E+00</b>	1.0E-2
	$h_c$	=	—	=	+	+	+	
	Rank	5	6	4	3	1	1	
F7	$F_{\text{mean}}$	1.099038E+01	4.356033E+00	3.122345E-04	<b>3.738876E-11</b>	7.950674E-01	7.874552E-06	1.0E-2
	$h_c$	—	—	=	=	—	=	
	Rank	5	6	3	1	4	2	
F8	$F_{\text{mean}}$	1.459828E-3	1.000020E-03	2.215035E-02	2.894092E-10	4.406297E+01	<b>0.000000E+00</b>	1.0E-6
	$h_c$	=	—	—	=	—	+	
	Rank	2	4	5	3	6	1	
F9	$F_{\text{mean}}$	8.450319E-09	2.743254E-04	3.954267E-06	7.111280E-16	<b>0.000000E+00</b>	<b>0.000000E+00</b>	1.0E-6
	$h_c$	=	—	—	=	+	+	
	Rank	4	6	5	3	1	1	
F10	$F_{\text{mean}}$	1.031281E-01	1.208434E-02	4.143803E-03	3.922345E-07	<b>0.000000E+00</b>	<b>0.000000E+00</b>	1.0E-6
	$h_c$	—	—	—	=	+	+	
	Rank	6	5	4	3	1	1	
F11	$F_{\text{mean}}$	3.526003E+00	2.286126E-02	3.412021E-05	1.923376E-01	<b>4.867431E-324</b>	3.210145E-155	1.0E-2
	$h_c$	—	—	=	—	+	+	
	Rank	6	4	3	5	1	2	
F12	$F_{\text{mean}}$	3.810015E-04	1.939540E-04	1.547576E-04	2.625040E-03	2.392401E-03	<b>3.762522E-05</b>	1.0E-2
	$h_c$	=	=	=	=	=	=	
	Rank	4	3	2	6	5	1	
F13	$F_{\text{mean}}$	3.045789E+00	4.797606E-03	6.124084E-04	4.704666E-08	6.217249E-15	<b>2.664535E-17</b>	1.0E-2
	$h_c$	—	=	=	=	+	+	
	Rank	6	5	4	3	2	1	
F14	$F_{\text{mean}}$	6.487984E-01	7.895264E-03	2.375360E-06	1.863198E-07	<b>0.000000E+00</b>	<b>0.000000E+00</b>	1.0E-2
	$h_c$	—	=	=	=	+	+	
	Rank	6	5	4	3	1	1	
F15	$F_{\text{mean}}$	8.945142E+00	7.714974E-01	1.708216E-08	1.088321E-03	1.032058E-05	<b>0.000000E+00</b>	1.0E-2
	$h_c$	—	—	=	=	=	+	
	Rank	6	5	2	4	3	1	
F16	$F_{\text{mean}}$	1.307397E-09	3.508363E-04	1.458235E-03	9.597393E-09	4.270480E+03	<b>0.000000E+00</b>	1.0E-6
	$h_c$	=	—	—	=	—	+	
	Rank	2	4	5	3	6	1	
+/-/-		0/7/9	1/4/11	0/11/5	1/13/2	11/4/3	13/3/0	
Average rank		4.8125	4.625	3.75	3.625	2.6875	1.125	
Overall rank		6	5	4	3	2	1	

tested PSO algorithms in most benchmark functions due to the balance of the global and local searching capabilities of the algorithm. The proposed self-learning strategy of the

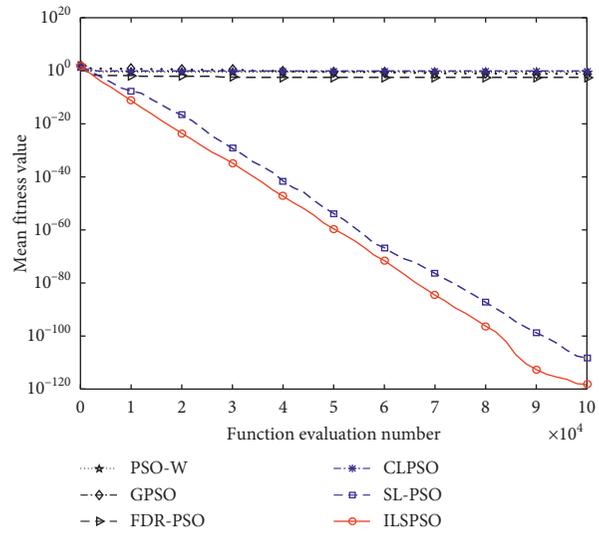
ILSPSO algorithm increases the convergence speed on the premise of population diversity and promotes the exploitation capability. The multilearning strategy of the ILSPSO



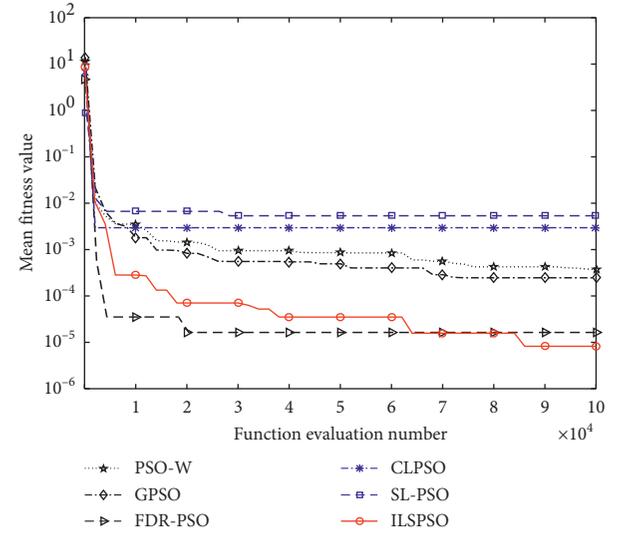
(a)



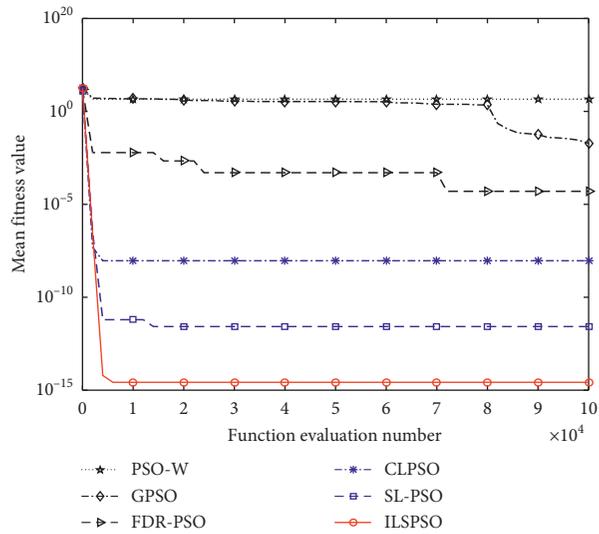
(b)



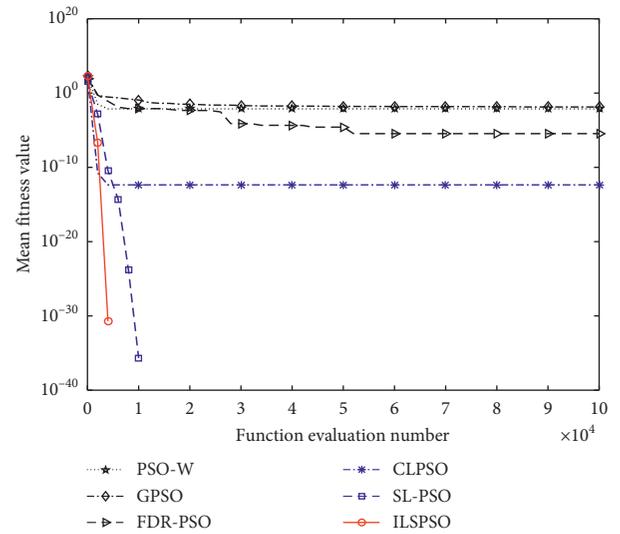
(c)



(d)

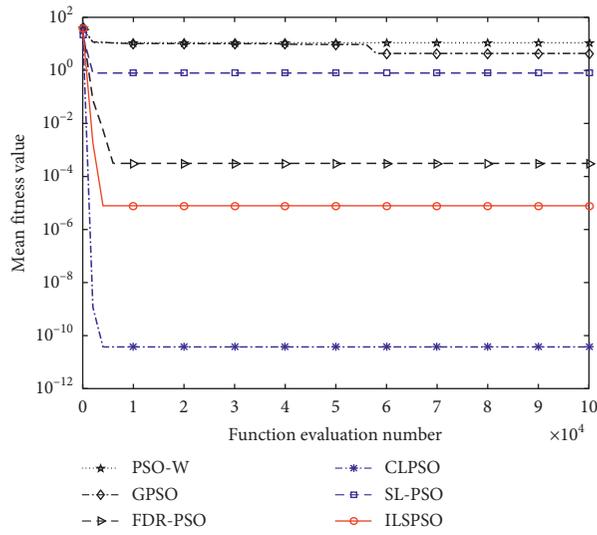


(e)

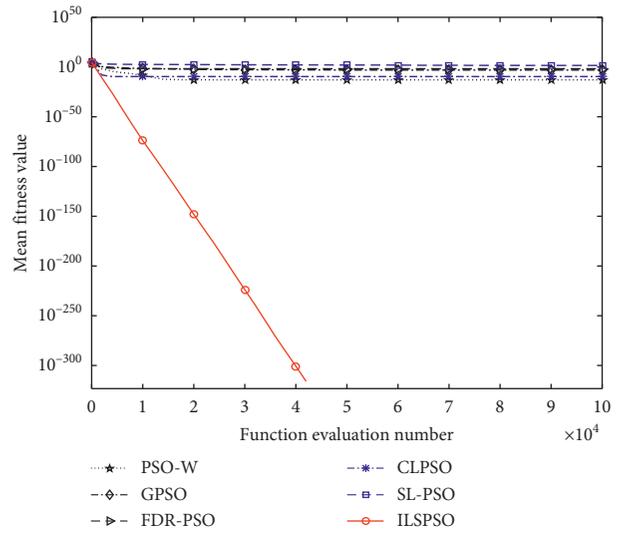


(f)

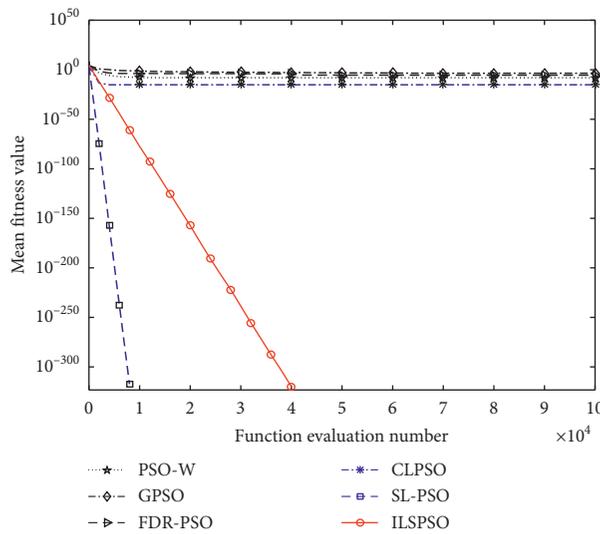
FIGURE 4: Continued.



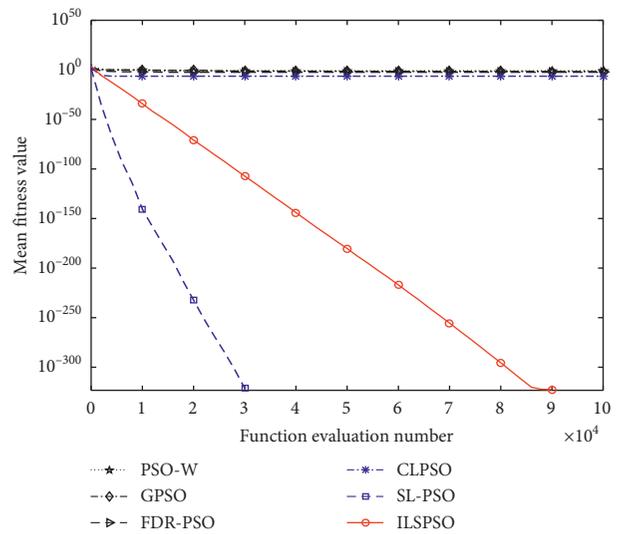
(g)



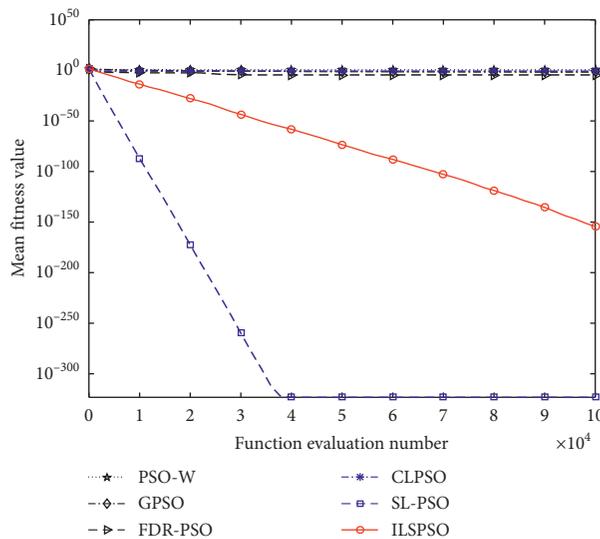
(h)



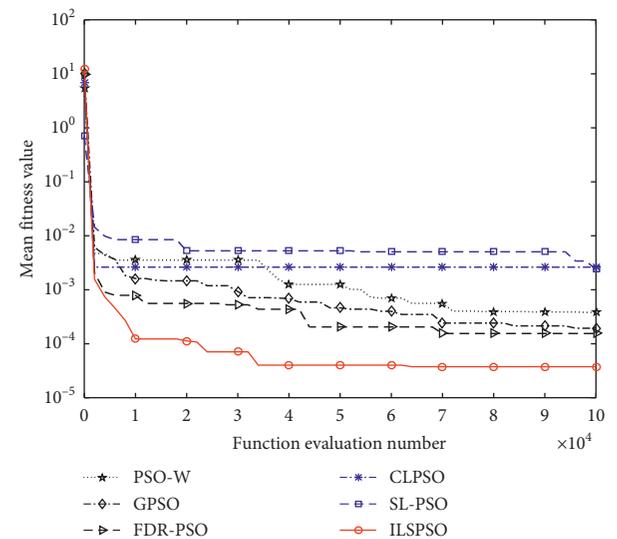
(i)



(j)



(k)



(l)

FIGURE 4: Continued.

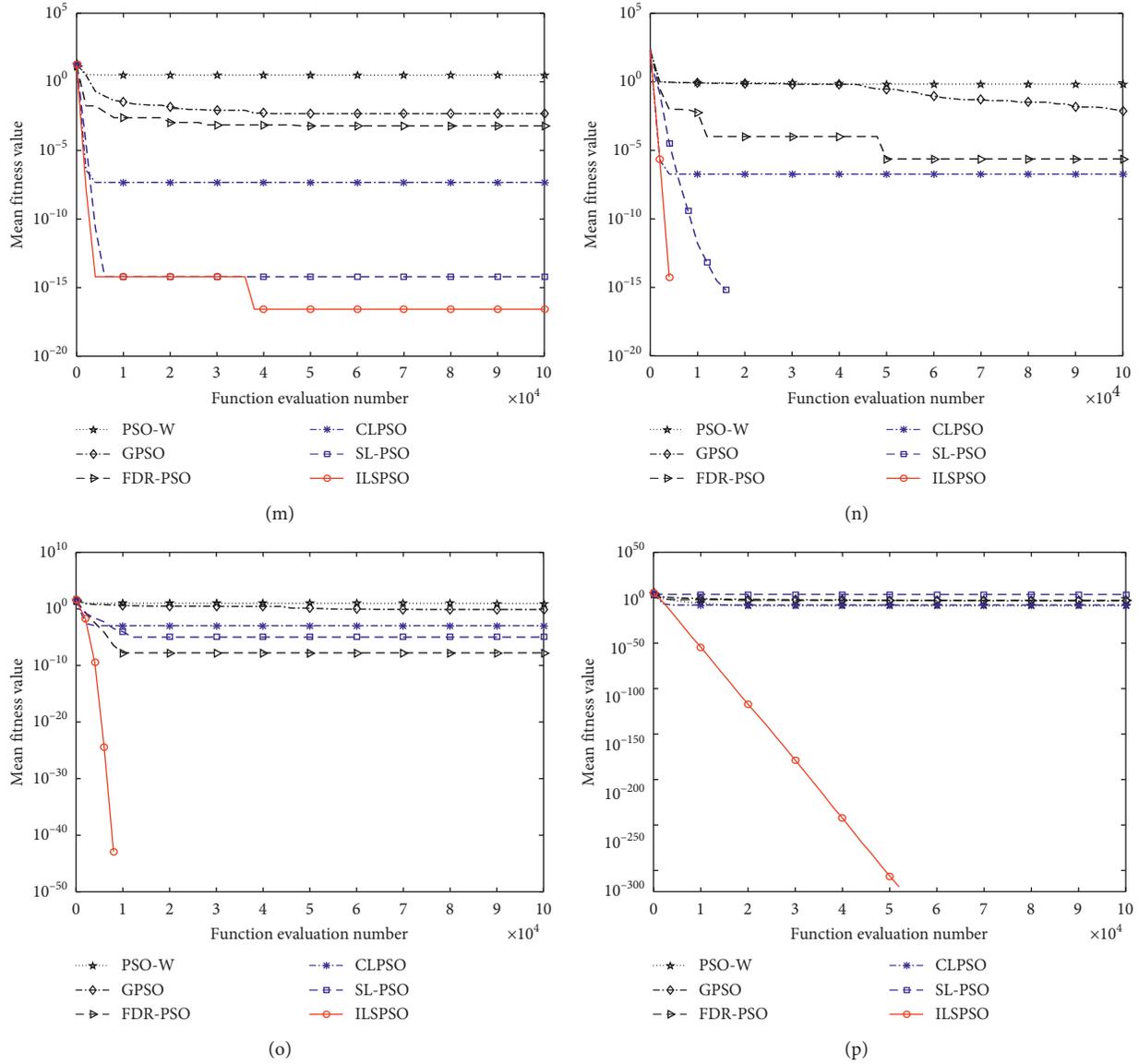


FIGURE 4: Convergence chart of the tested PSO variants: (a) F1; (b) F2; (c) F3; (d) F4; (e) F5; (f) F6; (g) F7; (h) F8; (i) F9; (j) F10; (k) F11; (l) F12; (m) F13; (n) F14; (o) F15; (p) F16.

algorithm enhances the exploration capability, assists the algorithm in implementing global search, and effectively prevents premature convergence.

### 5. Vehicle Path Planning Using the ILSPSO Algorithm

Path planning is one of the basic links of vehicle navigation [54]. It searches for an optimal or near-optimal collision-free path from the initial state to the target state according to a certain performance index. In Section 4, the performance of the ILSPSO algorithm was compared with five representative PSO variants in the experiments. The test results on benchmark functions demonstrate that the proposed ILSPSO algorithm achieves significantly better overall performance and outperforms other tested PSO variants. In this

section, the ILSPSO algorithm will be applied to vehicle path planning. In order to simplify the problem, the path planning is divided into two steps: (1) using the MAKLINK graph theory method and Dijkstra algorithm to find a near-optimal collision-free path and (2) optimizing the near-optimal collision-free path with the ILSPSO algorithm to obtain the global optimal collision-free path.

*5.1. Problem Description and Modeling.* It can be assumed that the vehicle travels in 2-dimensional space ( $200 \times 200$  m), the coordinate of the starting point  $S$  is (20, 180), and the coordinate of the ending point  $E$  is (160, 90). The path planning of the vehicle is to find an optimal collision-free path from the starting point  $S$  to the ending point  $E$ . There are 4 obstacles in the 2-dimensional space: The coordinates of the four vertices of the obstacle 1 are (40, 140), (60, 160),

(100, 140), and (60, 120), respectively. The coordinates of the four vertices of the obstacle 2 are (50, 30), (30, 40), (80, 80), and (100, 40), respectively. The coordinates of the four vertices of the obstacle 3 are (120, 160), (140, 100), (180, 170), and (165, 180), respectively. The coordinates of the three vertices of the obstacle 4 are (120, 40), (170, 40), and (140, 80), respectively.

In order to establish a reasonable environment model, the MAKLINK graph theory method is used to establish the 2-dimensional path planning feasible space of vehicle through the MAKLINK lines. The MAKLINK lines are defined as the lines (do not intersect with obstacles) between the vertices of the two obstacles, and the lines connecting the vertices of the obstacle to the boundary, as shown as the purple dotted line in Figure 5. Among the figure, the green polygons are the obstacles and the dotted lines are the MAKLINK lines. The number of MAKLINK lines is 20, and their midpoint are marked as  $m_1, m_2, \dots, m_{20}$ , respectively. The undirected network map of the initial path planning (they are the lines on which the vehicle can move freely) is established by connecting the midpoints of all MAKLINK lines, starting point  $S$ , and ending point  $E$ , as shown with orange solid line in Figure 6.

**5.2. Solution of Near-Optimal Collision-Free Path Based on the Dijkstra Algorithm.** Through the environmental model established above, the path planning problem of the vehicle is transformed into the shortest path problem of the undirected network map. Because the Dijkstra algorithm is a typical single-source shortest path algorithm, it is used to calculate the shortest path from one node to all other nodes in a nonnegative weight graph. Therefore, the Dijkstra algorithm can be used to generate a near-optimal collision-free path through the path nodes  $S, D_1, D_2, \dots, D_m, E$  on the MAKLINK map. The free connections corresponding to the nodes are named as  $L_i (i = 1, 2, \dots, n)$ .

The flow of the Dijkstra algorithm is as follows [55].

*Step 1.* Initializing a set of nodes  $V$  that has not determined the shortest path and a set of nodes  $W$  that has determined the shortest path. Then, the adjacency matrix arcs with weighted graph are used to initialize the source point to other nodes for the shortest path length  $F$ . If the source point has a join arc to other nodes, the corresponding value is the weight of the join arc; otherwise, the corresponding value is the maximum value.

*Step 2.* Assuming that the minimum value is  $F[i]$  in  $F$ ,  $F[i]$  is the shortest path length from the source point to point  $i$ , and the point  $i$  is taken out of the set  $V$  and placed in the set  $W$ .

*Step 3.* Modifying the path length value of the source point in the array  $F$  to the node  $k$  in the set  $V$  according to the node  $i$ .

*Step 4.* Repeat Step 2 and Step 3 until the shortest path from the source to all nodes is found.

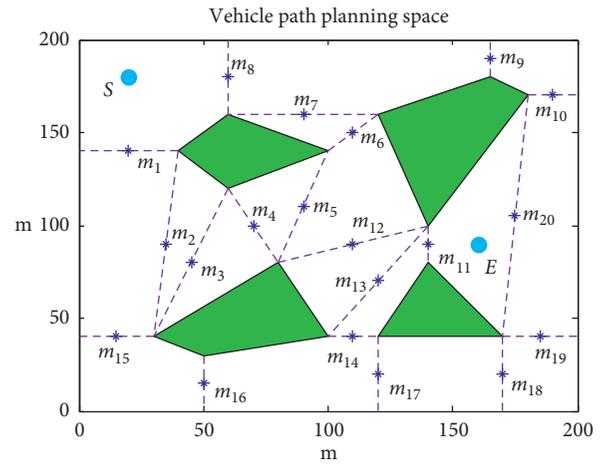


FIGURE 5: MAKLINK diagram based on convex polygon.

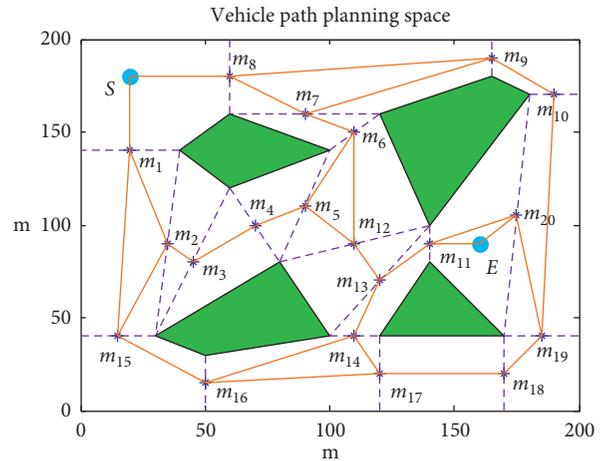


FIGURE 6: MAKLINK undirected network diagram.

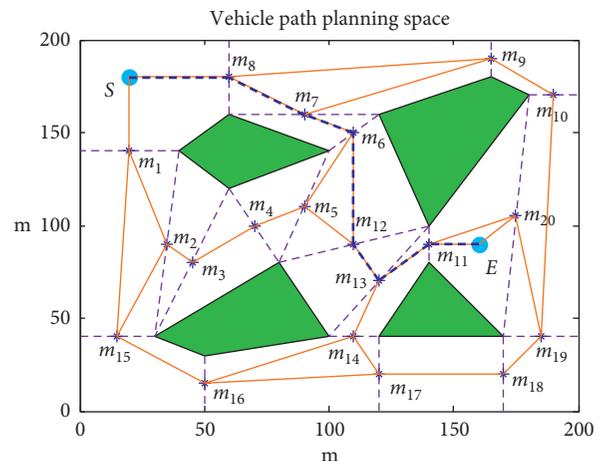


FIGURE 7: Near-optimal collision-free path obtained with the Dijkstra algorithm.

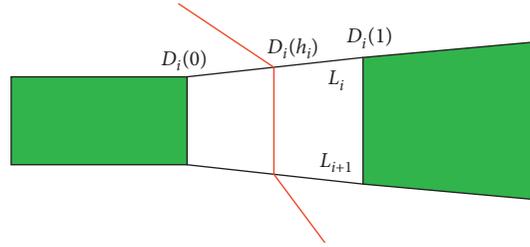
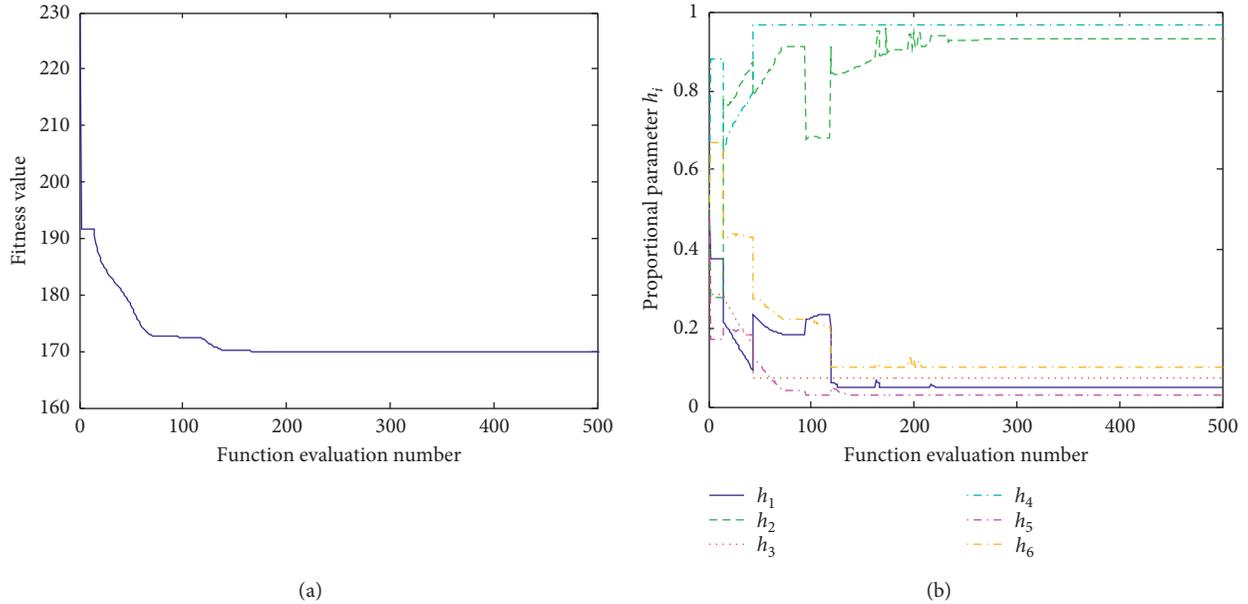


FIGURE 8: Path coding method.


 FIGURE 9: Optimization results of the global optimal collision-free path based on the ILSPSO algorithm: (a) convergence chart of fitness value and (b) optimization curves of proportional parameter  $h_i$ .

Finally, the shortest path length obtained with the Dijkstra algorithm is 229.0611 m, and the result is shown in Figure 7.

**5.3. Solution of Global Optimal Collision-free Path Based on the ILSPSO Algorithm.** The Dijkstra algorithm described above implements the shortest path solution for the undirected network map in Figure 6. However, the vehicle can follow the boundary of the 2-dimensional path planning feasible space which is constructed by the MAKLINK line, rather than having to follow the undirected network path in Figure 6. Therefore, the above path is not necessarily the optimal solution of the entire planning space. Subsequently, the near-optimal collision-free path is optimized using the ILSPSO algorithm to obtain a global optimal collision-free path.

In order to ensure that the vehicle is not close to obstacles when operating, the boundary of the obstacle is extended outward by 1/2 of the maximum dimension of the vehicle body in the length and width direction plus the minimum sensing distance of the sensor [56]. At this time, the vehicle is considered a mass point and the size is negligible. As shown in Figure 8, assume that the two endpoints of the MAKLINK line  $L_i$  are  $D_i(0)$  and  $D_i(1)$ , respectively, and the point  $D_i(h_i)$  can slide freely on the MAKLINK line  $L_i$ . The free space of the vehicle is composed of convex

polygons, which ensures that the new path formed when the  $D_i(h_i)$  slides on the MAKLINK line  $L_i$  does not intersect the obstacle. Any point  $D_i(h_i)$  on the MAKLINK line  $L_i$  can be expressed as follows:

$$D_i(h_i) = D_i(0) + [D_i(1) - D_i(0)] \times h_i, \quad (5)$$

$$h_i \in [0, 1], \quad i = 1, 2, 3, \dots, n,$$

where  $h_i$  is the proportional parameter and  $n$  is the number of MAKLINK lines on the near-optimal collision-free path.

In the near-optimal collision-free path shown in Figure 7, there are 6 free-sliding path points. Therefore, the search space of particles in the ILSPSO algorithm is 6 dimensions. In the process of optimization solution, the population size is 30, the maximum evolution number is 500, the inertia weight  $w$  is 0.729, the acceleration parameters  $c_1$  and  $c_2$  are 1.494 and 1.494, and the parameter  $k$  is 3, respectively. The optimization results of the global collision-free path based on the ILSPSO algorithm are shown in Figure 9. From the Figure 9(a), the ILSPSO algorithm can converge to optimization solution after about 200 evolution iteration (the shortest path length is 169.9), 25.83% lower than the initial path. The proportional parameter  $h_i$  of 6 free-sliding path points are shown in

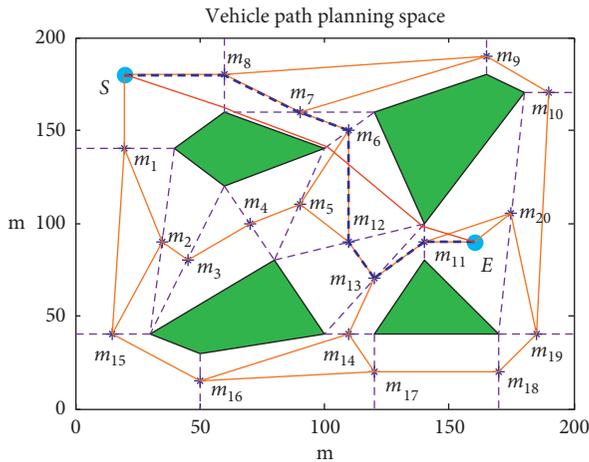


FIGURE 10: Comparison between the global optimal collision-free path and near-optimal collision-free path.

Figure 9(b). Finally, the comparison between the global optimal collision-free path and near-optimal path is shown in Figure 10.

## 6. Conclusions

This paper presents a novel particle swarm optimization (ILSPSO) which employs the improved learning strategies, namely, self-learning strategy and multilearning strategy. In the self-learning strategy, each particle of the  $pbest$ , except for the best one, will stochastically learn from any better particles in the current  $pbest$  solutions according to the sort of particles. At the same time, in order to insure the population diversity in the searching process, an empirical formula is used to adjust the learning probability along with the change of evolution iterations. This strategy aims to increase the convergence speed on the premise of population diversity. Moreover, the  $gbest$  is replaced with randomly chosen from the top  $k$  of  $gbest$  ( $gbestk$ ) in the corresponding dimension according to the multilearning strategy, and it enhances the global searching capability of the algorithm. Subsequently, the proposed ILSPSO algorithm is compared with other five representative PSO variants based on the 16 well-known benchmark test functions belonging to three classes. The experimental and statistical analyses demonstrate that the proposed ILSPSO algorithm shows significantly better overall performance and outperforms other tested PSO variants in most benchmark functions irrespective of whether they are unrotated or rotated.

The vehicle's path is initially planned by MAKLINK graph theory and Dijkstra algorithm. Subsequently, the proposed ILSPSO algorithm is employed to the quadratic programming of the initial path. The vehicle path obtained by quadratic programming of the ILSPSO algorithm is 25.83% shorter than the initial plan, and it shows that the ILSPSO algorithm has a satisfactory performance in vehicle path planning. All the results above prove that the proposed ILSPSO algorithm is a feasible and effective way to solve real-life application problems.

## Data Availability

The simulation code and data used to support the findings of this study have been deposited in the GitHub website, and this web URL is <https://github.com/jsluen/PSO.git>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the 13th Five-Year Plan of the National Key Research Project (no. 2016YFD0702101), the Natural Science Foundation of Jiangsu Province (BK20180189), the National Natural Science Foundation of China (no. 51475217), and the China Postdoctoral Science Foundation (224882) and the project was funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (no. PAPD-2018-87).

## References

- [1] A. Mortazavi and V. Toğan, "Sizing and layout design of truss structures under dynamic and static constraints with an integrated particle swarm optimization algorithm," *Applied Soft Computing*, vol. 51, pp. 239–252, 2017.
- [2] Z. Wan, G. Wang, and B. Sun, "A hybrid intelligent algorithm by combining particle swarm optimization with chaos searching technique for solving nonlinear bilevel programming problems," *Swarm and Evolutionary Computation*, vol. 8, pp. 26–32, 2013.
- [3] P. J. M. Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, Norwell, MA, USA, 1987.
- [4] J. He, R. Z. Zhou, and Z. L. Hong, "Modified fast climbing search auto-focus algorithm with adaptive step size searching technique for digital camera," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 257–262, 2003.
- [5] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Norwell, MA, USA, 1997.
- [6] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press (Reprint), Cambridge, UK, 1992.
- [7] M. Dorigo, V. Maniezzo, and A. Colnori, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, December 1995.
- [9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report TR06, Erciyes University, Kayseri, Turkey, 2005.
- [10] I. Montalvo, J. Izquierdo, R. Pérez, and P. L. Iglesias, "A diversity-enriched variant of discrete PSO applied to the design of water distribution networks," *Engineering Optimization*, vol. 40, no. 7, pp. 655–668, 2008.
- [11] H. S. Ramadan, A. F. Bendary, and S. Nagy, "Particle swarm optimization algorithm for capacitor allocation problem in distribution systems with wind turbine generators," *International Journal of Electrical Power & Energy Systems*, vol. 84, pp. 143–152, 2017.

- [12] J.-J. Kim and J.-J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620–631, 2015.
- [13] R. Ruiz-Cruz, E. N. Sanchez, F. Ornelas-Tellez, A. G. Loukianov, and R. G. Harley, "Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1698–1709, 2013.
- [14] Y. Li, X. Bai, L. Jiao, and Y. Xue, "Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation," *Applied Soft Computing*, vol. 56, pp. 345–356, 2017.
- [15] G. Das, P. K. Pattnaik, and S. K. Padhy, "Artificial neural network trained by particle swarm optimization for non-linear channel equalization," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3491–3496, 2014.
- [16] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2018.
- [17] X. Wu, W. Bai, Y. Xie, X. Sun, C. Deng, and H. Cui, "A hybrid algorithm of particle swarm optimization, metropolis criterion and RTS smoother for path planning of UAVs," *Applied Soft Computing*, vol. 73, pp. 735–747, 2018.
- [18] S. M. Mousavi, A. Bahreininejad, S. N. Musa, and F. Yusof, "A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network," *Journal of Intelligent Manufacturing*, vol. 28, no. 1, pp. 191–206, 2017.
- [19] C. Grosan, A. Abraham, and M. Chis, *Swarm Intelligence in Data Mining*, Springer-Verlag, Berlin, Germany, 2009.
- [20] W. H. Lim and N. A. Mat Isa, "Adaptive division of labor particle swarm optimization," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5887–5903, 2015.
- [21] C. Y. Wang, P. Liu, and H. Z. Qin, "Review on intelligent path planning algorithm of mobile robots," *Transducer and Microsystem Technologies*, vol. 37, no. 8, pp. 5–8, 2018.
- [22] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, AK, USA, 1998.
- [23] Y. H. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, no. 12, pp. 101–106, 2011.
- [24] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [25] Y. Tang, Z. Wang, and J.-A. Fang, "Feedback learning particle swarm optimization," *Applied Soft Computing*, vol. 11, no. 8, pp. 4713–4725, 2011.
- [26] G. Xu, "An adaptive parameter tuning of particle swarm optimization algorithm," *Applied Mathematics and Computation*, vol. 219, no. 9, pp. 4560–4569, 2013.
- [27] K. Chen, F. Zhou, and A. Liu, "Chaotic dynamic weight particle swarm optimization for numerical function optimization," *Knowledge-Based Systems*, vol. 139, pp. 23–40, 2018.
- [28] W.-N. Chen, J. Zhang, Y. Lin et al., "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.
- [29] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," *Swarm Intelligence Symposium*, vol. 1019, no. 1, pp. 174–181, 2003.
- [30] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [31] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [32] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multi modal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 387–402, 2013.
- [33] S. L. Sabat, L. Ali, and S. K. Udgata, "Integrated learning particle swarm optimizer for global optimization," *Applied Soft Computing*, vol. 11, no. 1, pp. 574–584, 2011.
- [34] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, no. 1, pp. 43–60, 2015.
- [35] M. R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, and N. Srikanth, "Directionally driven self-regulating particle swarm optimization algorithm," *Swarm and Evolutionary Computation*, vol. 28, no. 6, pp. 98–116, 2016.
- [36] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, and X.-L. Shen, "A hybrid particle swarm optimization algorithm using adaptive learning strategy," *Information Sciences*, vol. 436–437, pp. 162–177, 2018.
- [37] M. Løvbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimizer with breeding and subpopulations," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, San Francisco, CA, USA, July 2001.
- [38] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing*, vol. 10, no. 2, pp. 629–640, 2010.
- [39] F. Han and Q. Liu, "A diversity-guided hybrid particle swarm optimization based on gradient search," *Neurocomputing*, vol. 137, pp. 234–240, 2014.
- [40] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211–224, 2004.
- [41] Y. Marinakis, A. Migdalas, and A. Sifaleras, "A hybrid particle swarm optimization—variable neighborhood search algorithm for constrained shortest path problems," *European Journal of Operational Research*, vol. 261, no. 3, pp. 819–834, 2017.
- [42] J. Ding, J. Liu, K. R. Chowdhury, W. Zhang, Q. Hu, and J. Lei, "A particle swarm optimization using local stochastic search and enhancing diversity for continuous optimization," *Neurocomputing*, vol. 137, no. 8, pp. 261–267, 2014.
- [43] Z. Beheshti and S. M. Hj. Shamsuddin, "CAPSO: centripetal accelerated particle swarm optimization," *Information Sciences*, vol. 258, no. 2, pp. 54–79, 2014.
- [44] M. J. Mahmoodabadi, Z. Salahshoor Mottaghi, and A. Bagheri, "HEPSO: high exploration particle swarm optimization," *Information Sciences*, vol. 273, no. 7, pp. 101–111, 2014.
- [45] J. Kennedy, "Bare bones particle swarms," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*, pp. 80–87, Indianapolis, IN, USA, April 2003.
- [46] J. Kennedy, "In search of the essential particle swarm," in *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*, pp. 1694–1701, Vancouver, Canada, July 2006.

- [47] C. Lin and Q. Y. Feng, "Chaotic particle swarm optimization algorithm based on the essence of particle swarm," *Journal of Southwest Jiaotong University*, vol. 42, no. 6, pp. 665–669, 2007.
- [48] R. Mendes and J. Kennedy, "Stochastic barycenters and beta distribution for Gaussian particle swarms," in *Conference on Progress in Artificial Intelligence*, vol. 4874, pp. 259–270, Springer, Berlin, Germany, 2007.
- [49] S.-Z. Zhao and P. N. Suganthan, "Two-lbests based multi-objective particle swarm optimizer," *Engineering Optimization*, vol. 43, no. 1, pp. 1–17, 2011.
- [50] N. D. Jana and J. Sil, "Particle swarm optimization with exploratory move," in *Pattern Recognition and Machine Intelligence*, Springer, Berlin, Germany, 2013.
- [51] R. Salomon, "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions," *Biosystems*, vol. 39, no. 3, pp. 263–278, 1996.
- [52] K. Yu, X. Wang, and Z. Wang, "Multiple learning particle swarm optimization with space transformation perturbation and its application in ethylene cracking furnace optimization," *Knowledge-Based Systems*, vol. 96, pp. 156–170, 2016.
- [53] Ş. Gülcü and H. Kodaz, "A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 45, no. 10, pp. 33–45, 2015.
- [54] C. Luo, W. Li, X. Fan et al., "Positioning technology of mobile vehicle using self-repairing heterogeneous sensor networks," *Journal of Network and Computer Applications*, vol. 93, pp. 110–122, 2017.
- [55] F. Gao, *MATLAB Intelligent Algorithm Super Learning Manual*, Posts and Telecom Press, Beijing, China, 2014.
- [56] Y. Q. Qin, D. B. Sun, N. Li, and Q. Ma, "Path planning for mobile robot based on particle swarm optimization," *Robot*, vol. 26, no. 3, pp. 222–225, 2004.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

