Research Article

Particle Swarm Optimization with Double Learning Patterns

Yuanxia Shen, Linna Wei, Chuanhua Zeng, and Jian Chen

School of Computer Science and Technology, Anhui University of Technology, Maanshan 243002, China

Correspondence should be addressed to Yuanxia Shen; yuanxiashen@163.com

Received 16 July 2015; Revised 11 October 2015; Accepted 15 October 2015

Academic Editor: Manuel Grana

Copyright © 2016 Yuanxia Shen 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.

Particle Swarm Optimization (PSO) is an effective tool in solving optimization problems. However, PSO usually suffers from the premature convergence due to the quick losing of the swarm diversity. In this paper, we first analyze the motion behavior of the swarm based on the probability characteristic of learning parameters. Then a PSO with double learning patterns (PSO-DLP) is developed, which employs the master swarm and the slave swarm with different learning patterns to achieve a trade-off between the convergence speed and the swarm diversity. The particles in the master swarm and the slave swarm are encouraged to explore search for keeping the swarm diversity and to learn from the global best particle for refining a promising solution, respectively. When the evolutionary states of two swarms interact, an interaction mechanism is enabled. This mechanism can help the slave swarm in jumping out of the local optima and improve the convergence precision of the master swarm. The proposed PSO-DLP is evaluated on 20 benchmark functions, including rotated multimodal and complex shifted problems. The simulation results and statistical analysis show that PSO-DLP obtains a promising performance and outperforms eight PSO variants.

1. Introduction

Particle Swarm Optimization (PSO) [1, 2], firstly proposed by Kennedy and Eberhart in 1995, was inspired by the simulation of simplified social behaviors including fish schooling and bird flocking. Similar to genetic algorithm, it is also a population-based algorithm, but it has no evolutionary operations such as crossover, mutation, or selection. PSO finds the global best solution by adjusting the trajectory of each particle not only towards its personal best particle *pbest* but also towards the historically global best particle *gbest* [3]. Recently, PSO has been successfully applied to optimization problems in many fields [4–7].

In the basic PSO [1], each particle in the swarm learns from *pbest* and *gbest*. During the evolutionary process, *gbest* is the only shared information acquired by the whole swarm, which finally leads to all particles converging to the same destination and the diversity losing quickly. If *gbest* is a local optimum far from the global one, the swarm is easy to be trapped in local optimum. The learning mechanism of the basic PSO can cause a fast convergence rate, but it easily leads to the premature convergence when solving multimodal optimization problems. In order to overcome this problem, researchers proposed many strategies to improve it.

An adaptive strategy of the learning parameter [3, 8–18] is an effective way to improve the PSO performance. Shi and Eberhart [8] proposed a linearly decreasing inertia weight (LDIW) to balance the local search and global search. Ratnaweera et al. [3] proposed a time-varying acceleration coefficient (TVAC), which is beneficial to enhancing the exploration ability of particles in the early evolutionary phase and improving the local searching ability of particles in the late phase. In [3], the two variants of the PSO-TVAC were developed, namely, the PSO-TVAC with mutation (MPSO-TVAC) and the self-organizing hierarchical PSO-TVAC (HPSO). Zhan et al. [9] proposed an adaptive PSO in which the learning parameters were adaptively adjusted with the change of the evolutionary states of the swarm. Kundu et al. [10] proposed a nonlinearly time-varying acceleration coefficient and an aging guideline to avoid the premature convergence. They also suggested a mean learning strategy to enhance the exploitation search.

To increase the swarm diversity, auxiliary techniques are introduced into PSO's framework, such as genetic operators [3, 12, 13, 19], differential evolution [20], and artificial bee colony (ABC) [21, 22]. Mahmoodabadi et al. [22] combined the multicrossover and the bee colony mechanism to improve

the exploration capability of PSO. In [9], an elitist learning strategy, similar to the mutation operation, is developed to help the *gbest* particle jump out of the local optima.

The topological structure of the swarm has a significant effect on the performance of PSO [23–26]. Kennedy [23] pointed out that small neighborhood is fit for complex problems, while large neighborhood is used for simple problems. Parsopoulos and Vrahatis [24] integrated the benefits of the global PSO and the local PSO and then proposed a unified PSO (UPSO). Mendes et al. [25] proposed a fully informed PSO (FIPSO) in which the updating velocity depends on neighborhoods of each particle instead of *gbest* and *pbest*. Bratton and Kennedy [26] proposed a standard version of PSO (SPSO) in which a local ring topology is employed. Experimental results indicated that the local model is more effective than the global model on many test problems.

The design of learning strategies improves the performance of PSO in complex multimodal problems [27–31]. In the basic PSO, each particle learns from *gbest*. Hence, the swarm diversity loses easily in the initial evolutionary process. Zhou et al. [27] developed a random position PSO (RPPSO). In [27], if the randomly generated number is smaller than the acceptance probability, a random position is used to guide the particle. Liang et al. [28] proposed a comprehensive learning PSO (CLPSO) in which each particle can select its *pbest* or other's *pbest* as the learning exemplar according to the given probability. Li et al. [29] developed a self-learning PSO which contains four learning strategies: exploitation, jumping out, exploration, and convergence. Huang et al. [30] proposed an example-based learning PSO (ELPSO) algorithm that uses multiple global best positions as elite examples to retain the swarm diversity. Chen et al. [31] proposed a PSO with an aging leader and challengers (ALC-PSO). In [31], an aging mechanism is designed to promote a suitable leader to lead the evolution of the swarm.

Multiswarm PSO (MS-PSO) [32–36] is developed to maintain the balance of the exploration/exploitation search. In a homogenous MS-PSO, each swarm adopts a similar learning strategy. On the contrary, each swarm uses different learning strategy to implement the different search task in a heterogeneous MS-PSO. Niu et al. [32] presented a multiswarm cooperative optimizer (MCPSO) where the population consists of a master swarm and several slave swarms. Each slave swarm searches the better solution independently, while the master swarm collects the best particles from the slaves to refine the global optimization. Sun and Li [33] presented a two-swarm cooperative PSO (TCPSO) where one swarm is used to concentrate around the local optimum and accelerate the convergence, and the particles of the other one are dispersed around the search interval to keep the diversity.

Local topological structures and dynamical exemplar strategies can keep the swarm diversity to efficiently prevent the premature convergence, but the convergence rate is slow. The heterogeneous multiswarm method is powerful in balancing the local search and the global search by taking advantage of different learning strategies of the subswarms. In this method, it is important to design learning strategies which directly influence the performance of the algorithm. In order to develop efficient learning strategies, this paper

analyzes the motion behavior of the swarm based on the probability characteristic of the learning parameters. Meanwhile, we point out that the probability characteristic of the learning parameters has an influence on the search space of the particles. Then, we propose a PSO with double learning patterns (PSO-DLP) to improve both the convergence rate and the accuracy of PSO. PSO-DLP adopts the master swarm and the slave swarm to obtain a good balance between the exploitation and exploration search. The master swarm is used in the exploration search, and the slave swarm is employed to carry out the exploitation search and to accelerate the convergence. The two swarms fulfill their tasks by adjusting the probability characteristics of the learning parameters. An interaction mechanism between two swarms is developed, which can help the slave swarm in jumping out of the premature convergence and improve the convergence precision of the master swarm. The simulation results show that PSO-DLP has powerful ability of global search and fast convergence speed. Experimental studies on 20 well-known benchmark functions show that the proposed PSO-DLP obtains a promising performance in terms of the accuracy and the convergence speed.

The rest of this paper is organized as follows. Section 2 describes the basic PSO. Section 3 presents the behavior analysis of the basic PSO. Section 4 presents the methodologies of PSO-DLP in detail. Section 5 provides the experimental settings and the results. Finally, Section 6 concludes this work.

2. Basic PSO

PSO is a population-based algorithm, which consists of a group of particles. Each particle i is represented by two vectors, namely, a position vector $\mathbf{X}_i = (X_{i1}, X_{i2}, \dots, X_{iD})$ and a velocity vector $\mathbf{V}_i = (V_{i1}, V_{i2}, \dots, V_{iD})$. The position of each particle in the search space is treated as a potential solution. Each particle i updates its velocity and position with the following equations:

$$V_{id}(t+1) = wV_{id}(t) + c_1r_1(t)(Pb_{id}(t) - X_{id}(t)) + c_2r_2(t)(Gb_d(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1), \quad (2)$$

where $X_{id}(t+1)$ and $V_{id}(t+1)$ represent the d th dimension of the position and the velocity of particle i at the $(t+1)$ th iteration. $\mathbf{Pb}_i = (Pb_{i1}, Pb_{i2}, \dots, Pb_{iD})$ (*pbest*) is the personal best experience of the i th particle and $\mathbf{Gb} = (Gb_1, Gb_2, \dots, Gb_D)$ (*gbest*) is the group best experience found by the whole swarm. w is an inertia weight; c_1 and c_2 are acceleration coefficients reflecting the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest*, respectively. Random factors r_1 and r_2 are two independent random numbers in the range of $[0, 1]$.

The first item of (1) (i.e., wV_{id}) is the previous velocity, which provides the necessary momentum for particles to roam around the search space. The second item (i.e., $c_1r_1(Pb_{id} - X_{id})$), known as the “cognitive” component, represents the personal thinking of each particle. The cognitive component encourages the particles to move towards *pbest*.

The third item (i.e., $c_2 r_2 (Gb_d - X_{id})$), regarded as the “social” component, expresses the collaborative effect of the particles in finding the global optimal solution. The social component always pulls the particles to $gbest$.

3. Behavior Analysis of PSO

In the basic PSO, each particle toward the optimum solution is guided by the cognitive component and the social component. Therefore, proper control of these two components is very important to find the optimum solution accurately and efficiently [3]. Hence, researchers have proposed various strategies, such as the linearly varying acceleration coefficients [3], the nonlinearly varying acceleration coefficients [9–11], and the time-varying acceleration coefficients with the evolutionary state [16, 17]. In addition, Krohling [14] presented a Gaussian PSO in which random factors and acceleration coefficients are instead of the two positive random numbers generated according to the Gaussian distribution. Similar to the Gaussian PSO, Richer and Blackwell [15] introduced the Lévy distribution to replace the uniform distribution of random factors. The previous works show that the setting of acceleration coefficients and the probability distribution of random factors can affect the performance of PSO.

In order to facilitate the analysis, the velocity updating equation needs to be transformed. Substituting (1) into (2), we obtain

$$X_{id}(t+1) = X_{id}(t) + c_1 r_1(t) (Pb_{id}(t) - X_{id}(t)) + c_2 r_2(t) (Gb_d(t) - X_{id}(t)) + wV_{id}(t). \quad (3)$$

According to (3), we can also write

$$X_{id}(t+1) = X_{id}(t) + wV_{id}(t) + (c_1 + c_2) \cdot \left[\frac{c_1 r_1}{(c_1 + c_2)} + \frac{c_2 r_2}{(c_1 + c_2)} \right] \cdot \left[\frac{c_1 r_1 Pb_{id}(t)}{(c_1 r_1 + c_2 r_2)} + \frac{c_2 r_2 Gb_d(t)}{(c_1 r_1 + c_2 r_2)} - X_{id}(t) \right]. \quad (4)$$

Let $k = c_1/(c_1 + c_2)$. Then (4) can be simplified as follows:

$$X_{id}(t+1) = X_{id}(t) + wV_{id}(t) + (c_1 + c_2) \left[kr_1 + (1-k)r_2 \right] \left[\frac{kr_1 Pb_{id}(t)}{(kr_1 + (1-k)r_2)} + \frac{(1-k)r_2 Gb_d(t)}{(kr_1 + (1-k)r_2)} - X_{id}(t) \right]. \quad (5)$$

Let $D_f = kr_1 + (1-k)r_2$ and $L_f = kr_1/(kr_1 + (1-k)r_2)$; then from (5), we get

$$X_{id}(t+1) = X_{id}(t) + (c_1 + c_2) \cdot D_f \left[L_f Pb_{id}(t) + (1-L_f) Gb_d(t) - X_{id}(t) \right] + wV_{id}(t), \quad (6)$$

where D_f and L_f are functions with respect to the random factors r_1, r_2 and the undetermined parameter k . Hence D_f and L_f are correlative random variables in $[0, 1]$. From (6), the movement of the particles from the t th iteration to the $(t+1)$ th iteration can be divided into two parts. The particles firstly enter into a search space, defined by the second term of (6), and then make an inertia motion decided by the third term. Let $Pu = L_f Pb_{id}(t) + (1-L_f) Gb_d(t)$, where Pu is a point on the line connected by $Pb_{id}(t)$ and $Gb_d(t)$. Given a fixed L_f , the term $(Pu - X_{id}(t))$ represents a line segment from $X_{id}(t)$ to Pu , namely, one-step search space. The acceleration coefficient and the variable D_f can influence the size of the one-step search space, and D_f can also decide the distribution of the location of particles in this space. The variable L_f is the weighting coefficient, which reflects the differences of exploitation of $pbest$ and $gbest$. When $L_f = 1$, the particle learns from $pbest$; when $L_f = 0$, the particle learns from $gbest$. Variables L_f and D_f are the learning factor and the distribution factor.

In order to analyze the effect of the two factors on the one-step search space, we need to calculate the probabilistic characteristics of the factors. Considering a general situation, that is, $c_1 = c_2$ (i.e., $k = 0.5$), we obtain

$$L_f = \frac{r_1}{(r_1 + r_2)} \quad (7)$$

$$D_f = 0.5r_1 + 0.5r_2.$$

r_1 and r_2 are two independent uniform random numbers in the range of $[0, 1]$. Hence we calculate the density function ($f_{L_f}(l_f)$) of the learning factor and the joint density ($f(d_f, l_f)$) of the learning factor and the distribution factor (see the Appendix). The density function $f_{L_f}(l_f)$ and the joint density $f(d_f, l_f)$ are given by

$$f_{L_f}(l_f) = \begin{cases} \frac{1}{(2(1-l_f)^2)}, & 0 \leq l_f < 0.5 \\ \frac{1}{(2l_f^2)}, & 0.5 \leq l_f \leq 1 \\ 0, & \text{other} \end{cases} \quad (8)$$

$$f(d_f, l_f) = \begin{cases} 4d_f, & 0 \leq d_f \leq 1, \max\left(0, 1 - \frac{1}{2d_f}\right) \leq l_f \leq \min\left(1, \frac{1}{2d_f}\right) \\ 0, & \text{other.} \end{cases} \quad (9)$$

Using (8) and (9), we can calculate $f_{D_f|L_f}(d_f | l_f)$ the conditional probability density of D_f given L_f :

$$f_{D_f|L_f}(d_f | l_f) = \begin{cases} 8d_f(1-l_f)^2, & 0 \leq l_f < \frac{1}{2}, 0 \leq d_f < \frac{1}{2(1-l_f)} \\ 8d_f l_f^2, & \frac{1}{2} \leq l_f < 1, 0 \leq d_f < \frac{1}{2l_f}. \end{cases} \quad (10)$$

We can see from (8) that $f(l_f)$ is a unimodal function and it is symmetrical at L_f equal to 0.5 (Figure 1(a)). If the interval of the learning factor is divided into three smaller ones, that

FIGURE 1: The probability characteristics of the learning parameters.

is, $(0, 0.25)$, $(0.25, 0.75)$, and $(0.75, 1)$, the probability of the learning factor lying in the interval $[0, 0.25]$ is the same as that in interval $[0.75, 1]$. When the learning factor is located in the interval $[0, 0.25]$, we consider the particle emphasis on learning from $gbest$. If, on the contrary, the learning factor is located in the interval $[0.75, 1]$, the particle pays attention to learning from $pbest$.

From (10), we can see that the range of the distribution factor depends on the value of the learning factor. The relationship between the learning factor and the distribution factor is shown in Figure 1(b). When the learning factor equals 0.5, the value of the distribution factor ranges from 0 to 1. When the learning factor equals 0 or 1, the value of the distribution factor ranges from 0 to 0.5. If given a fixed Z , $f(d_f | l_f)$ increases with the rising of the distribution factor. This means that the particles tend to the large value of the distribution factor.

In the basic PSO, the probability characteristics of the learning factor and the distribution factor may bring forth the clustering of the swarm. (1) During an iteration, each particle emphasizes learning from $gbest$ with the probability that $P(0 < l_f < 0.25) = 1/6$; at the same time, the value range of the distribution factor is restricted because the learning factor is in the interval of $[0, 0.25]$, which means that the one-step space of the particle is lessened. This situation will eventually lead to the swarm clustering move toward $gbest$. (2) The conditional probability density $f(d_f | l_f)$ of the distribution factor is an increasing function with the increased distribution factor, which means that the particle trends to a “long-distance flying” in the one-step space. This flying may accelerate the clustering of the particles. (3) When the learning factor equals 0.5, the value of the distribution factor falls into its maximum range from 0 to 1. But the probability of the learning factor around 0.5 is 0.182, $P(0.5 - \delta < d_f < 0.5 + \delta) = 0.182$ ($\delta = 0.05$). In other words, the one-step space of the particle is shrunk with the probability

of 0.182. All of the above reasons may bring about the quick clustering of the swarm.

4. PSO with Double Learning Patterns (PSO-DLP)

In this section, we describe PSO-DLP in detail. According to the analysis of the basic PSO, we know that the probability characteristics of the learning parameters can influence the search behavior of particles. PSO-DLP takes advantage of the probability characteristics of the leaning parameters to achieve the global search.

4.1. Learning Patterns. In PSO-DLP, we develop two learning patterns, called a uniform exploration and an enhanced exploitation. And two swarms, namely, a master swarm and a slave swarm, are also employed. The master swarm adopts the uniform exploration to avoid the premature convergence and the slave swarm uses the enhanced exploitation to accelerate the convergence speed.

In the uniform exploration learning, we present three novel strategies. Firstly, the learning factor and the distribution factor are independent. The value of the distribution factor varies from 0 to 1, which enlarges the search space of particles efficiently. In the basic PSO, the value of the distribution factor reaches its maximum value of 1 only when the value of the learning factor equals 0.5. Secondly, the distribution factor is subject to the uniform distribution in $[0, 1]$, which is beneficial for preserving the swarm diversity. Thirdly, the learning factor decreases with iteration, which helps the particles in emphasizing the exploration in the earlier stage of the evolution and enhancing the convergence in the later stage. For particle i in the master swarm, its velocity V_i^m is updated as follows:

$$V_{id}^m(t+1) = wV_{id}^m(t) + (c_1^m + c_2^m) \cdot D_f^m \left[L_f^m \cdot Pb_{id}^m(t) + (1 - L_f^m) Gb_d^m(t) - X_{id}^m(t) \right] \quad (11)$$

$$f_{L_f^m}(l_f^m) = \begin{cases} 1, & 0 \leq l_f^m < 1 \\ 0, & \text{other} \end{cases} \quad (12)$$

$$L_f^m = L_{f,\max}^m - \frac{(L_{f,\max}^m - L_{f,\min}^m) \text{iter}}{\text{iter}_{\max}}, \quad (13)$$

where $L_{f,\max}^m = 1$, $L_{f,\min}^m = 0$; iter is the number of fitness evaluations (FEs); and $\text{iter}_{\max}$ is the maximum FEs defined by the user. Gb^m is the *gbest* of the master swarm.

The purpose of the enhanced exploitation learning is to make the search focus on a region for refining a promising solution. In this pattern, the learning factor and the distribution factor are independent. The learning factor is a uniform random number in $[0, 0.5]$, which can make the search concentrate around *gbest*. The distribution factor is also a random number generated in the interval of $[0, 0.5]$ uniformly, which shrinks the search space to accelerate the convergence.

For particle i in the master swarm, its velocity V_i updates as follows:

$$V_{id}^s(t+1) = wV_{id}^s(t) + (c_1^s + c_2^s) \cdot D_f^s [L_f^s \cdot Pb_{id}^s(t) + (1 - L_f^s) Gb_d^s(t) - X_{id}^s(t)] \quad (14)$$

$$f_{L_f^s}(l_f^s) = \begin{cases} 2, & 0 \leq l_f^s < 0.5 \\ 0, & \text{other} \end{cases} \quad (15)$$

$$f_{D_f^s}(d_f^s) = \begin{cases} 2, & 0 \leq d_f^s < 0.5 \\ 0, & \text{other}, \end{cases} \quad (16)$$

where Gb^s is the *gbest* of the slave swarm.

4.2. Interaction between Swarms. In PSO-DLP, two learning patterns play different roles in the evolutionary process. The master swarm is used for the global search, while the slave swarm is employed for the local search. But particles in the slave swarm get easily trapped in the local optima. To improve the convergence precision of the slave swarm, the interaction between two swarms is necessary. This interaction is unidirectional and the information flow is from the master swarm to the slave swarm. Particles in the master swarm do not receive information from the slave swarms in order to keep the ability to perform the global search. When the best particle (Gb^s) in the slave swarm is not improved from L successive FEs or the fitness value of Gb^s is less than the one of the best particle (Gb^m) in the slave swarm (the higher the fitness value, the better the performance of the particle), particles in the slave swarm can learn from the best particle in the master swarm (Gb^m) ($Gb^s = Gb^m$). Too small values and too large ones of L are not desirable, as the former tends to weaken the exploitation capability of particles in the slave swarm and the latter leads to wasting computation resources (as the slave swarm may suffer from the premature convergence). In this study, we set $L = 50$.

4.3. PSO-DLP Procedure. PSO-DLP is developed based on the two learning patterns and the interaction between the swarms discussed above. The pseudocode of the PSO-DLP is given in Algorithm 1. As no additional operation is introduced into PSO-DLP, the computational complexity and the storage memory are the same as the basic PSO. Therefore, PSO-DLP keeps the simplicity of the basic PSO.

5. Experimental Setup and Simulation Results

5.1. Benchmark Functions. The 20 scalable benchmark functions are used to investigate the performance of the proposed algorithm, including unimodal functions, multimodal functions, rotated functions, and shifted functions. These functions are widely adopted in [3, 8–13, 19–40]. These problems are minimization problems and the expressions of the functions are given in Table 1. The shifting and rotating methods used in the test functions are from [28, 39]. In Table 1, M denotes the orthogonal matrix, and o denotes the shifted global optimum and f_{bias} denotes the shifted fitness value. All functions are evaluated with 30 variables.

5.2. Parameter Settings for the Involved PSO Variants. For the comprehensive comparison with PSO-DLP, eight PSO variants are employed in this paper. They are PSO-W [8], HPSO [3], FIPS [25], CLPSO [28], SPSO [26], HEPHO [22], COMPSO [32], and TS-CPSO [33]. The parameter settings of each peer algorithm are extracted from the corresponding literatures and they are given in Table 2.

The swarm size is set to 40 [28] for eight algorithms except for COMPSO. For COMPSO, the population size is set at 80, as the suggestion proposed by its author. COMPSO has four subswarms, and TS-CPSO and PSO-DLP have two subswarms. In the three algorithms, each subswarm has the same size which is set to 20. To ensure a fair comparison, the maximum number of fitness evaluations (FEs) is set at $1.00E + 5$. Each algorithm was run 30 times independently to reduce random discrepancy.

5.3. Performance Metrics. In this study, we adopted the fitness mean (Fm), success rate (SR), and success performance (SP) to assess the accuracy, the reliability, and the efficiency of PSO, respectively [39]. Fm is the mean difference $f(x) - f(x^*)$ between the best fitness value of the algorithm and the global optimum (x^* and x represent the global solution and the best solution found by an algorithm, resp.). SR denotes the consistency of an algorithm to achieve the solution with a predefined ϵ . SP denotes the number of FEs required by an algorithm to solve the problem with a predefined ϵ . The Wilcoxon test [41, 42] is applied to perform pairwise comparison between PSO-DLP and its peers. The confidence level is fixed at 0.95. If the performance of PSO-DLP is better than its peer, the h value is denoted by “+”; if the h value is “=” or “−”, this indicates that the performance of PSO-DLP is almost the same as or is significantly worse than its peers, respectively. The average ranking can be calculated to undertake multiple comparisons [41, 42].

TABLE 1: Benchmark functions used in this paper.

Number	Description and expression	Search space	ϵ	$F(X^*)$
Group 1: conventional problems				
F_1	Sphere $F_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	10^{-6}	0
F_2	Schwefel's function 1.2 $F_2(x) = \sum_{i=1}^D (\sum_{k=1}^i x_k)^2$	$[-100, 100]^D$	10^{-6}	0
F_3	Noise quadric $F_3(x) = \sum_{i=1}^D i x_i^4 + \text{random}(0, 1)$	$[-1.28, 1.28]^D$	10^{-2}	0
F_4	Rosenbrock $F_4 = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	10^{-2}	0
F_5	Ackley $F_5(x) = -20 \exp(-0.2 \sqrt{(\sum_{i=1}^D x_i^2)/D}) - \exp((\sum_{i=1}^D \cos(2\pi x_i))/D) + 20 + e$	$[-32.768, 32.768]^D$	10^{-6}	0
F_6	Griewank $F_6(x) = \sum_{i=1}^D x_i^2 / 4000 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) + 1$	$[-600, 600]^D$	10^{-6}	0
F_7	Rastrigin $F_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	10^{-6}	0
F_8	Noncontinuous Rastrigin $F_8(x) = F_7(y)$ ($y_i = \{x_i, x_i < 0.5; \text{round}(2x_i)/2, x_i \geq 0.5\}$)	$[-5.12, 5.12]^D$	10^{-6}	0
F_9	Expanded Schaffer $F_9(x) = \sum_{i=1}^{D-1} y(x_i, x_{i+1}) + y(x_D, x_1)$ $y(u, v) = 0.5 + (\sin^2(\sqrt{u^2 + v^2}) - 0.5) / (1 + 0.001(u^2 + v^2))^2$	$[-100, 100]^D$	10^{-6}	0
Group 2: rotated problems				
F_{10}	Rotated Rosenbrock $F_{10}(x_i) = F_4(q_i)$, $q_i = M * x_i$, M is an orthogonal matrix	$[-10, 10]^D$	10	0
F_{11}	Rotated Ackley $F_{11}(x_i) = F_5(q_i)$, $q_i = M * x_i$	$[-32.768, 32.768]^D$	10	0
F_{12}	Rotated Griewank $F_{12}(x_i) = F_6(q_i)$, $q_i = M * x_i$	$[-600, 600]^D$	10	0
F_{13}	Rotated Rastrigin $F_{13}(x_i) = F_7(q_i)$, $q_i = M * x_i$	$[-5.12, 5.12]^D$	10	0
F_{14}	Rotated noncontinuous Rastrigin $F_{14}(x_i) = F_8(q_i)$, $q_i = M * x_i$	$[-5.12, 5.12]^D$	10	0
Group 3: shifted problems				
F_{15}	Shifted Sphere $F_{15}(x_i) = F_1(q_i) + f_{\text{bias}1}$, $q_i = x_i - o$, $f_{\text{bias}1} = -450$	$[-100, 100]^D$	10^{-6}	-450
F_{16}	Shifted Rosenbrock $F_{16}(x_i) = F_4(q_i) + f_{\text{bias}2}$, $q_i = x_i - o$, $f_{\text{bias}2} = 390$	$[-10, 10]^D$	10^{-6}	390
F_{17}	Shifted Rastrigin $F_{17}(x_i) = F_7(q_i) + f_{\text{bias}4}$, $q_i = x_i - o$, $f_{\text{bias}4} = -330$	$[-5.12, 5.12]^D$	10^{-6}	-330
F_{18}	Shifted non-Rastrigin $F_{18}(x_i) = F_8(q_i) + f_{\text{bias}5}$, $q_i = x_i - o$, $f_{\text{bias}5} = -330$	$[-5.12, 5.12]^D$	10^{-6}	-330
F_{19}	Shifted rotated Ackley's function with global optimum on bounds	$[-32.76, 32.76]^D$	10^{-6}	-140
F_{20}	Shifted rotated Rastrigin's function	$[-5.12, 5.12]^D$	10^{-6}	-330

Input:

Master swarm size (Nm), slave swarm size (Ns), the dimensionality of the problem space (D), maximum number of fitness evolution (iterMax), objective function (f)

- (1) Randomly initialize the position X_i^m and velocity V_i^m of all particles in the master swarm.
- (2) Randomly initialize the position X_i^s and velocity V_i^s of all particles in the slave swarm.
- (3) Calculate the fitness values of X^m and X^s ,
- (4) Set X^m and X^s to be $Pb^m = (Pb_1^m, Pb_2^m, \dots, Pb_{Nm}^m)$ and $Pb^s = (Pb_1^s, Pb_2^s, \dots, Pb_{Ns}^s)$ for each particle of the master swarm and slave swarm, respectively.
- (5) Set the particle with the best fitness of the master swarm to be Gb^m , and set the particle with the best fitness of the slave swarm to be Gb^s .
- (6) Set generation $k = 0$, the counter $fj = 0$.
- (7) **while** $k \leq \text{iterMax}$ **do**
- (8) **for** $i = 1 : Nm$ **do**
- (9) **for** $d = 1 : D$ **do**
- (10) Update the dimensional velocity $V_{i,d}^m$ according to (11), and Update the dimensional position $X_{i,d}^m$ according to (2).
- (11) **end for**
- (12) Calculate the fitness value $f(X_i^m)$;
- (13) **if** $f(X_i^m) < f(Pb_i^m)$ **then**
- (14) $Pb_i^m = X_i^m$; $f(Pb_i^m) = f(X_i^m)$;
- (15) **if** $f(X_i^m) < f(Gb^m)$ **then**
- (16) $Gb^m = X_i^m$; $f(Gb^m) = f(X_i^m)$;
- (17) **end if**
- (18) **end if**
- (19) **end for**
- (20) **for** $i = 1 : Ns$ **do**
- (21) **for** $d = 1 : D$ **do**
- (22) Update the dimensional velocity $V_{i,d}^s$ according to (14), and Update the dimensional position $X_{i,d}^s$ according to (2);
- (23) **end for**
- (24) Calculate the fitness value $f(X_i^s)$;
- (25) **if** $f(X_i^s) < f(Pb_i^s)$ **then**
- (26) $Pb_i^s = X_i^s$; $f(Pb_i^s) = f(X_i^s)$;
- (27) **if** $f(X_i^s) < f(Gb^s)$ **then**
- (28) $Gb_i^s = X_i^s$; $f(Gb_i^s) = f(X_i^s)$; $fj = 0$;
- (29) **else**
- (30) $fj = fj + 1$;
- (31) **end if**
- (32) **end if**
- (33) **end for**
- (34) **if** $fj == L$ **then**
- (35) $Gb^s = Gb^m$;
- (36) **end if**
- (37) $k = k + 1$;
- (38) **end while**
- (39) Set the global best position $Gb = \min(Gb^s, Gb^m)$
- (40) **return** Gb

ALGORITHM 1: Particle swarm optimization with double learning patterns.

5.4. Parameter Sensitivity Analysis. The effect of parameter L on the performance of PSO-DLP was investigated. The seven benchmarks are selected, namely, F_1 , F_4 , F_7 , F_8 , F_{10} , F_{12} , F_{14} , and F_{17} . The L value of PSO-DLP was set to an integer value varying from 10 to 100. Table 3 presents the mean values and the standard deviation obtained by PSO-DLP with different L .

From Table 3, we can see that the experimental results of functions F_1 and F_4 decrease with the increasing of L because F_1 and F_2 are unimodal and the higher value of L is beneficial

to the exploitation search. For multimodal functions F_7 , F_8 , F_{10} , F_{12} , F_{14} , and F_{17} , the performance of PSO-DLP tends to deteriorate when L is set too low (i.e., $L = 10, 20$, and 30) or too high (i.e., $L = 90, 100$). When L is set too low, the interaction between subswarms is overemphasized such that the slave swarm receives the best solution from the master swarm so frequently that the search of the slave swarm is not sufficient to refine the search. When L is set too high, the interaction between swarms is not enough such that the slave swarm may get into the premature convergence for solving

TABLE 2: Parameter settings of algorithms used in the comparisons.

Algorithm	Year	Population topology	Parameter settings
PSO-LDIW	1998	Fully connected	$\omega: 0.9-0.4, c_1 = c_2 = 2.0$
HPSO	2004	Fully connected	$\omega: 0.9-0.4, c_1: 2.5-0.5, c_2: 0.5-2.5$
FIPSO	2004	Local Ring	$\lambda = 0.729, \Sigma c_i = 4.1$
CLPSO	2006	Comprehensive learning	$\omega: 0.9-0.4, c = 2.0, m = 7$
SPSO	2007	Local Ring	$\omega = 0.729, c_1 = c_2 = 1.193$
COMPSO	2007	Multiswarm (fully connected)	$\omega: 0.9-0.4, c_1^s = c_2^s = 2.05, c_1^M = c_2^M = c_3^M = 2$
HEPSO	2014	Fully connected	$\omega: 0.9-0.4, c_1: 2.5-0.5, c_2: 0.5-2.5, Pc = 0.95, Pb = 0.02$
TS-COMPSO	2014	Multiswarm (fully connected and local ring)	$\omega^M = 0.9, c_1^s = c_2^s = 3.4, c_1^M = c_2^M = c_3^M = 4.5$
CMA-ES	2007	—	$\sigma = 0.25, \mu = (4 + \lfloor 2 \log(N) \rfloor)/2$
SADE	2009	—	$F \sim N(CR, 0.3), CR \sim N(CR_m, 0.1), CR_0 = 0.5, LP = 50$
JADE	2009	—	$p = 0.05, c = 0.1$
PSO-DLP	—	Multiswarm (fully connected)	$\omega: 0.9-0.3, c_1^s = c_2^s = c_1^m = c_2^m = 2.0, L = 50$

TABLE 3: Effects of the parameter L on PSO-DLP in 30-D.

	Fm	$L = 10$	$L = 20$	$L = 30$	$L = 40$	$L = 50$	$L = 60$	$L = 70$	$L = 80$	$L = 90$	$L = 100$
F_1	SD	$2.80E-191$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Fm	$0.00E+00$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_4	SD	$7.97E-17$	$7.57E-17$	$8.07E-18$	$7.97E-17$	2.01E-19	$3.04E-19$	$7.97E-17$	$7.97E-17$	$8.27E-17$	$8.05E-17$
	Fm	$1.17E-17$	$2.10E-17$	$3.05E-18$	$2.00E-17$	6.19E-19	$2.33E-19$	$2.01E-17$	$1.02E-17$	$2.31E-17$	$5.01E-17$
F_7	SD	$5.90E-13$	$4.76E-14$	$3.55E-16$	$1.06E-15$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	$1.59E-14$	$1.17E-14$
	Fm	$1.28E-12$	$1.04E-13$	$7.94E-16$	$2.38E-15$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	$2.43E-14$	$7.53E-15$
F_8	SD	$1.93E-13$	$6.82E-14$	$2.84E-15$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	$3.19E-15$	$4.28E-15$
	Fm	$2.67E-13$	$1.48E-13$	$4.08E-15$	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	$7.14E-15$	$1.56E-15$
F_{10}	SD	$1.71E+00$	$1.71E+00$	$1.19E+00$	$7.46E-01$	2.00E-01	$2.01E-01$	$2.87E-01$	$1.72E+00$	$1.09E+00$	$1.94E+00$
	Fm	$1.10E+00$	$2.10E+00$	$1.54E+00$	$1.53E-01$	2.83E-01	$2.93E-01$	$1.86E-01$	$1.55E+00$	$1.72E+00$	$1.91E+00$
F_{12}	SD	$7.90E-03$	$7.90E-03$	$5.40E-03$	$7.40E-03$	3.70E-03	3.70E-03	$1.08E-02$	$4.90E-03$	$6.94E-03$	$5.91E-03$
	Fm	$5.11E-03$	$1.39E-03$	$8.60E-03$	$7.00E-03$	5.23E-03	5.23E-03	$1.01E-02$	$7.00E-03$	$9.60E-03$	$5.70E-03$
F_{14}	SD	$3.48E+01$	$3.56E+01$	$3.02E+01$	$2.65E+01$	2.55E+01	2.55E+01	$2.92E+01$	$2.98E+01$	$2.56E+01$	$2.60E+01$
	Fm	$4.02E+00$	$8.08E+00$	$7.22E+00$	$4.60E+00$	1.34E+01	$1.54E+01$	$5.97E+00$	$3.56E+00$	$6.58E+00$	$9.87E+00$
F_{17}	SD	$5.68E-14$	$5.11E-14$	$5.11E-14$	$5.11E-14$	1.27E-14	$1.54E-14$	$1.97E-14$	$3.97E-14$	$4.54E-14$	$3.41E-14$
	Fm	$0.00E+00$	$1.27E-14$	$1.27E-14$	$1.27E-14$	1.48E-13	$1.55E-14$	$1.57E-14$	$1.55E-14$	$1.55E-14$	$1.27E-14$

multimodal functions. If the interaction interval L is long, the master swarm could not help the slave swarm in getting out of the local optimum. From the above experimental findings, the parameter value of L at 50 had the promising performance of PSO-DLP. Hence, this parameter setting was adopted in the following experiments.

5.5. Experimental Results and Discussions

5.5.1. Comparison among the Fm Results. Table 4 presents the results of the fitness means Fm and standard deviation SD of the nine algorithms on the conventional problems, rotated problems, and shifted problems. The best results among the nine algorithms are shown in bold.

From Table 4, we observe that PSO-DLP obtains the best searching accuracy in the eight conventional problems, except for the function F_5 . To be specific, PSO-DLP finds

the global optima of F_1, F_6, F_7 , and F_8 . In addition, PSO-DLP is the only algorithm to achieve the accuracy level of 10^{-19} for the function F_4 . For the function F_5 , COMPSO obtains the best result and PSO-DLP ranks the third.

The functions in group 2 ($F_{10}-F_{14}$) are multimodal functions with coordinate rotations. As can be seen from Table 3, all the algorithms perform worse for the rotated functions than the unrotated counterparts. PSO-DLP performs the best on functions F_{10}, F_{11} , and F_{14} , that is, 3 out of 5 functions, in terms of the Fm. For function F_{13} , TS-CPSO obtains the best fitness value, followed by PSO-DLP.

From Table 4, we observe that all involved algorithms cannot find the global optima for shifted problems ($F_{15}-F_{18}$) and complex problems ($F_{19}-F_{20}$), except for function 15. Function 15 is an easy shifted problem; only PSO-DLP achieves the global optima (Fm = 0). PSO-DLP surpasses all other algorithms on the four functions. This implies that

TABLE 4: Experimental result comparisons among nine PSOs.

		PSO-W	SPSO	CLPSO	HPSO	FIPS	HEPSO	TS-CPSO	COMPSO	PSO-DLP
F_1	Fm	$1.34E-117$	$8.86E-97$	$3.35E-87$	$5.87E-322$	$5.66E-05$	0.00E+00	$5.92E-322$	$1.48E-323$	0.00E+00
	SD	$2.16E-116$	$1.86E-96$	$5.37E-87$	$2.18E-322$	$9.24E-06$	0.00E+00	$4.09E-322$	$1.87E-323$	0.00E+00
	h	+	+	+	=	+	+	=	=	
F_2	Fm	$1.41E-45$	$5.29E-02$	$4.48E-07$	$7.61E-53$	$1.56E-02$	$4.81E-43$	$3.96E-09$	$2.92E-47$	3.00E-86
	SD	$1.47E-45$	$3.47E-02$	$2.26E-07$	$1.68E-52$	$8.60E-02$	$2.48E-43$	$5.33E-09$	$5.89E-47$	4.25E-86
	h	+	+	+	+	+	+	+	+	
F_3	Fm	$2.34E-03$	$5.62E-03$	$3.50E-03$	$3.10E-03$	$4.50E-03$	$1.61E-03$	$9.43E-03$	$1.21E-03$	2.05E-04
	SD	$9.83E-04$	$1.61E-03$	$8.73E-04$	$2.40E-03$	$1.30E-03$	$6.15E-04$	$4.80E-03$	$4.60E-04$	1.24E-05
	h	+	+	+	+	+	+	+	+	
F_4	Fm	$2.53E+01$	$2.25E+01$	$1.40E+01$	$2.89E+00$	$2.02E+01$	$1.36E+01$	$7.71E-02$	$9.806E-01$	2.01E-19
	SD	$1.81E+00$	$3.76E+00$	$3.50E+00$	$4.03E+00$	$2.30E-01$	$2.194E+00$	$3.87E-02$	$2.19E+00$	6.19E-19
	h	+	+	+	+	+	+	+	+	
F_5	Fm	$7.11E-15$	$5.68E-15$	$7.10E-15$	$7.10E-15$	$7.54E-05$	0.00E+00	$3.34E-13$	$4.97E-15$	$7.11E-15$
	SD	$0.00E+00$	$1.94E-15$	$0.00E+00$	$0.00E+00$	$2.47E-06$	0.00E+00	$3.45E-13$	$1.94E-15$	$0.00E+00$
	h	=	=	=	=	=	=	+	=	
F_6	Fm	$7.39E-03$	$1.04E-14$	0.00E+00	$1.47E-02$	$1.02E-04$	0.00E+00	$2.65E-02$	$5.30E-03$	0.00E+00
	SD	$1.05E-02$	$1.47E-14$	0.00E+00	$2.08E-02$	$5.19E-05$	0.00E+00	$1.46E-02$	$7.30E-03$	0.00E+00
	h	+	+	+	+	+	=	+	+	
F_7	Fm	$2.12E+01$	$1.31E+01$	$1.15E-14$	$1.50E-12$	$8.98E+02$	0.00E+00	$1.42E-15$	$1.65E+01$	0.00E+00
	SD	$1.02E+01$	$1.25E+01$	$3.63E-15$	$3.20E-12$	$8.17E+01$	0.00E+00	$1.48E-15$	$3.19E+00$	0.00E+00
	h	+	+	+	+	+	=	=	+	
F_8	Fm	$2.37E+01$	$3.97E+01$	$4.85E-10$	$2.00E-01$	$7.36E+01$	$9.93E-13$	$1.20E+00$	$1.34E+01$	0.00E+00
	SD	$1.45E+01$	$4.48E+00$	$2.44E-10$	$4.47E-01$	$1.42E+01$	$5.67E-13$	$1.09E+00$	$1.34E+00$	0.00E+00
	h	+	+	+	+	+	=	+	+	
F_9	Fm	$1.27E+00$	$2.36E+00$	$2.59E+00$	$1.25E+00$	$9.07E+00$	$1.31E+00$	$7.03E+00$	$2.18E+00$	6.87E-01
	SD	$4.43E-01$	$5.21E-01$	$5.53E-01$	$1.75E-01$	9.10E-02	$3.33E-01$	$1.43E-01$	$4.36E-01$	$3.60E-01$
	h	+	+	+	+	+	+	+	+	
F_{10}	Fm	$2.51E+01$	$2.55E+01$	$2.09E+01$	$2.57E+01$	$2.17E+01$	$1.35E+01$	$1.70E+01$	$9.65E+00$	2.00E-01
	SD	$2.14E+01$	$1.48E+01$	$4.05E+01$	$2.79E+01$	4.94E-02	$7.16E+00$	$1.80E+00$	$5.14E+00$	$2.83E-01$
	h	+	+	+	+	+	+	+	+	
F_{11}	Fm	$1.34E+00$	7.11E-15	$1.05E-13$	$1.20E+00$	$7.56E-05$	$3.61E-12$	$1.92E+00$	$1.48E+00$	7.11E-15
	SD	$3.14E-01$	0.00E+00	$2.13E-14$	$7.42E-01$	$3.85E-06$	$8.02E-12$	$2.97E-01$	$5.52E-01$	0.00E+00
	h	+	=	+	+	+	+	+	+	
F_{12}	Fm	$3.93E-02$	$1.30E-06$	$8.13E-10$	$4.12E-02$	$6.03E-05$	0.00E+00	$1.28E-02$	$2.16E-02$	$3.70E-03$
	SD	$3.81E-02$	$2.88E-06$	$1.01E-11$	$2.30E-02$	$2.81E-06$	0.00E+00	$1.36E-02$	$3.03E-02$	$5.23E-03$
	h	+	-	-	+	-	-	+	+	
F_{13}	Fm	$5.17E+01$	$5.59E+01$	$3.46E+01$	$5.51E+01$	$1.59E+02$	$2.89E+01$	2.63E+01	$4.29E+01$	$2.87E+01$
	SD	$2.25E+01$	$1.80E+00$	$3.69E+00$	$10.0E+00$	$9.76E+00$	$5.33E+00$	3.02E+00	$7.42E+00$	$3.68E+00$
	h	+	+	+	+	+	+	-	+	
F_{14}	Fm	$4.05E+01$	$5.46E+01$	$3.77E+01$	$3.78E+01$	$1.15E+02$	$2.78E+01$	$3.22E+01$	$4.48E+01$	2.55E+01
	SD	$1.48E+01$	$8.26E+00$	$5.44E+00$	$5.40E+00$	$13.0E+01$	$3.74E+00$	$8.07E+00$	$8.70E+00$	1.34E+01
	h	+	+	+	+	+	+	+	+	
F_{15}	Fm	$5.68E-14$	$4.48E-14$	$5.68E-10$	$4.54E-14$	$2.40E+01$	$5.01E-07$	$5.68E-14$	$3.41E-14$	0.00E+00
	SD	$0.00E+00$	$3.21E-14$	$1.23E-10$	$2.54E-14$	$2.72E+01$	$3.23E-07$	$1.01E-14$	$3.11E-14$	0.00E+00
	h	+	+	+	+	+	+	+	+	
F_{16}	Fm	$1.04E+01$	$3.03E+01$	$1.25E+01$	$2.10E+00$	$2.02E+01$	$4.87E+01$	$1.40E-01$	$4.26E+00$	1.44E-09
	SD	$2.11E+00$	$7.93E+00$	$8.10E+00$	$4.69E+00$	$2.64E-01$	$1.63E+00$	$2.37E-01$	$5.20E+00$	3.21E-09
	h	+	+	+	+	+	+	+	+	

TABLE 4: Continued.

		PSO-W	SPSO	CLPSO	HPSO	FIPS	HEPSO	TS-CPSO	COMPSO	PSO-DLP
F_{17}	Fm	$2.41E+01$	$1.86E+01$	$5.68E-14$	$3.58E+00$	$1.25E+02$	$8.36E+00$	$4.58E+01$	$1.51E+01$	$1.27E-14$
	SD	$1.19E+01$	$1.02E+01$	$1.63E-14$	$4.19E+00$	$3.26E+01$	$6.20E+00$	$2.25E+01$	$3.92E+00$	$1.48E-15$
	h	+	+	+	+	+	+	+	+	
F_{18}	Fm	$2.51E+01$	$1.84E+01$	$8.68E-8$	$4.60E+00$	$1.09E+02$	$5.75E+00$	$3.56E+02$	$1.54E+01$	$4.55E-14$
	SD	$1.32E+01$	$8.73E+00$	$1.83E-8$	$3.04E+00$	$2.63E+01$	$2.29E+00$	$4.65E+02$	$5.77E+00$	$2.54E-14$
	h	+	+	+	+	+	+	+	+	
F_{19}	Fm	$2.06E+01$	$2.08E+01$	$2.07E+01$	$2.06E+01$	$2.09E+01$	$2.05E+01$	$2.04E+01$	$2.08E+01$	$2.04E+01$
	SD	$6.74E-01$	$7.57E-01$	$4.96E-01$	$6.74E-01$	$2.20E-01$	$4.03E-01$	$2.61E-01$	$6.51E-02$	$6.25E-02$
	h	+	+	+	+	+	+	=	+	
F_{20}	Fm	$7.78E+01$	$9.67E+01$	$6.03E+01$	$1.89E+02$	$4.20E+01$	$8.20E+01$	$4.01E+01$	$6.72E+01$	$3.68E+01$
	SD	$1.38E+01$	$2.09E+01$	$9.64E+00$	$9.88E+00$	$8.08E+00$	$2.25E+01$	$2.13E+01$	$1.34E+01$	$2.81E+00$
	h	+	+	+	+	+	+	+	+	

PSO-DLP is the least sensitive one to the shifted problem. Complex problems are rotating and shifting operations (F_{19} - F_{20}). PSO-DLP obtains the best solutions on complex functions.

5.5.2. Comparison of Nonparametric Statistical Test Results. The Wilcoxon test results between PSO-DLP and peers are reported in Table 4. From the test results, PSO-DLP significantly outperforms all other algorithms on functions $F_2, F_3, F_4, F_8, F_9, F_{10}, F_{14}, F_{15}, F_{16}, F_{17}, F_{18}$, and F_{20} . COMPSO performs the best on function F_5 . For function F_5 , PSO-DLP and COMPSO do not show significant difference according to their statistical test values. PSO-DLP obtains the global solution on functions F_6 and F_7 . For functions F_6 and F_7 , the performance of HEPSO, TS-CPSO, and CLPSO is almost the same as that of PSO-DLP as the Wilcoxon test results show.

To compare the performance of multiple algorithms, we used the average ranking of the Friedman test to evaluate the effectiveness of PSO-DLP comprehensively. The algorithm with the better convergence accuracy is assigned to a smaller rank value. Table 5 shows the average ranking of nine PSO variants on functions F_1 - F_{20} . According to the result of the average ranking, nine algorithms can be sorted into the following order: PSO-DLP, HEPSO, COMPSO, CLPSO, HPSO, TS-CPSO, PSO-W, SPSO, and FIPS. PSO-DLP obtains the best performance.

5.5.3. Comparisons of the SR Value and the SP Value. The SR values and the SP values of all algorithms are shown in Table 6. If an algorithm is unable to completely solve the problems, then the value of SR is set at zero (i.e., $SR = 0$) and the value of SP is assigned an infinity value (i.e., $SP = \text{"inf"}$). As shown in Table 6, we observe that PSO-DLP has more superior search reliability among the compared peers. It can completely solve all conventional problems and rotated problems except for functions F_9, F_{13} , and F_{14} . For functions F_9, F_{13} , and F_{14} , none of the compared algorithms can solve the problem completely at the predefined ϵ and PSO-DLP can achieve the best result on functions F_9 and F_{14} from Table 3. For the shifted problems, PSO-DLP completely solves four

TABLE 5: Average rankings achieved by Friedman test for the nine PSO algorithms.

Algorithm	Ranking
PSO-W	5.85 (8)
SPSO	6.4 (7)
CLPSO	4.65 (4)
HPSO	5.5 (5)
FIPS	7.3 (9)
HEPSO	3.4 (2)
TS-CPSO	5.15 (6)
COMPSO	4.45 (3)
PSO-DLP	1.5 (1)

out of six functions at the predefined ϵ . Meanwhile, none of the algorithms are able to solve functions F_{19} and F_{20} . Nevertheless, PSO-DLP can obtain the smallest Fm value for functions F_{19} and F_{20} .

As reported in Table 6, PSO-DLP can get seven best SP values in conventional problems and rotated problems. The results show that PSO-DLP costs the least computation to solve these problems at the predefined ϵ . Figure 2 shows the convergence curves of the nine PSO algorithms on all functions. It can be seen that the convergence curves for functions F_2, F_4, F_{10}, F_{11} , and F_{20} (Figures 2(b), 2(d), 2(j), 2(k), and 2(t)) show that the PSO-DLP has the competitive convergence speed among its peers. For functions F_9, F_{17} , and F_{18} (Figures 2(i), 2(q), and 2(r)), PSO-DLP converges slowly at the beginning and achieves with fast convergence in later stage of the evolution. Particularly, the convergence speed of PSO-DLP in functions F_6, F_7 , and F_8 (Figures 2(f), 2(g), and 2(h)) is significantly faster than its peers in the middle stage of the evolution. Overall, PSO-DLP has better control of the convergence speed and accuracy than the involved PSO variants.

5.5.4. Effect of Different Strategies. The proposed PSO-DLP employs two learning patterns: the uniform exploration

TABLE 6: Comparisons of the SR and the SP among nine PSOs.

		PSO-W	SPSO	CLPSO	HPSO	FIPS	HEPSO	TS-CPSO	COMPSON	PSO-DLP
F_1	SR	100	100	100	100	56.00	100	100	100	0.00E + 00
	SP	$2.17E + 03$	$5.23E + 04$	$6.44E + 04$	$3.45E + 03$	INF	$3.12E + 04$	$2.23E + 03$	$2.47E + 03$	2.04E + 03
F_2	SR	100	0.00	93.33	100	0.00	100	100	100	100
	SP	$8.23E + 04$	INF	$8.89E + 04$	$3.68E + 04$	INF	$3.12E + 04$	$7.15E + 04$	1.12E + 04	$2.54E + 04$
F_3	SR	100	86.66	90.00	100	83.3	100	90.00	100	100
	SP	$6.23E + 04$	$7.48E + 04$	$4.16E + 04$	$3.17E + 04$	$8.33E + 04$	$3.92E + 04$	$3.54E + 04$	$2.57E + 04$	1.83E + 04
F_4	SR	0	0	0	0	0	0	0	0	100
	SP	INF	INF	INF	INF	INF	INF	INF	INF	4.62E + 04
F_5	SR	100	100	100	100	100	100	100	100	100
	SP	$4.21E + 04$	$6.34E + 04$	$5.84E + 04$	$2.12E + 04$	$6.16E + 04$	$5.52E + 04$	$9.23E + 03$	$9.85E + 03$	1.51E + 04
F_6	SR	0	100	100	0	0	100	0	0	100
	SP	INF	$7.56E + 04$	$3.81E + 04$	INF	INF	2.35E + 04	INF	INF	$4.15E + 04$
F_7	SR	0	0	100	100	0	100	100	0	100
	SP	INF	INF	$6.14E + 04$	$5.52E + 04$	INF	$4.36E + 04$	$5.74E + 04$	INF	4.26E + 04
F_8	SR	0	0	100	0	0	100	0	0	100
	SP	INF	INF	$9.78E + 04$	INF	INF	$8.94E + 04$	INF	INF	4.85E + 04
F_9	SR	0	0	0	0	0	0	0	0	0
	SP	INF	INF	INF	INF	INF	INF	INF	INF	INF
F_{10}	SR	0	0	0	0	0	0	0	80.00	100
	SP	INF	INF	INF	INF	INF	INF	INF	$8.15E + 04$	3.27E + 04
F_{11}	SR	100	100	100	100	100	100	100	100	100
	SP	$3.85E + 04$	$4.27E + 04$	$3.27E + 04$	1.32E + 04	$6.28E + 04$	$4.87E + 04$	$3.75E + 04$	$3.27E + 04$	$5.63E + 04$
F_{12}	SR	100	100	100	100	100	100	100	100	100
	SP	$3.61E + 04$	$5.78E + 04$	$2.57E + 04$	$3.15E + 04$	$5.27E + 04$	$3.16E + 04$	$2.10E + 04$	$2.18E + 04$	$1.36E + 04$
F_{13}	SR	0	0	0	0	0	0	0	0	0
	SP	INF	INF	INF	INF	INF	INF	INF	INF	INF
F_{14}	SR	0	0	0	0	0	0	0	0	0
	SP	INF	INF	INF	INF	INF	INF	INF	INF	INF
F_{15}	SR	100	100	100	100	0	90.00	100	100	100
	SP	$4.93E + 04$	$5.63E + 04$	$5.63E + 04$	$5.63E + 04$	INF	$9.29E + 04$	9.08E + 03	$1.06E + 04$	$1.41E + 04$
F_{16}	SR	0	0	0	0	0	0	0	0	100
	SP	INF	INF	INF	INF	INF	INF	INF	INF	$1.35E + 04$
F_{17}	SR	0	0	100	0	0	0	0	0	100
	SP	INF	INF	$5.19E + 04$	INF	INF	INF	INF	INF	5.04E + 04
F_{18}	SR	0	0	100	0	0	0	0	0	100
	SP	INF	INF	4.55E + 04	INF	INF	INF	INF	INF	$8.21E + 04$
F_{19}	SR	0	0	0	0	0	0	0	0	0
	SP	INF	INF	INF	INF	INF	INF	INF	INF	INF
F_{20}	SR	0	0	0	0	0	0	0	0	0
	SP	INF	INF	INF	INF	INF	INF	INF	INF	INF

learning (UEL) and the enhanced exploitation learning (EEL). To investigate the effects of them, we studied the performance of PSO-W, PSO with UEL (PSO-UEL), PSO with EEL (PSO-EEL), and the complete PSO-DLP. PSO-UEL and PSO-EEL adopt one swarm so that there is no interaction between the subswarms. The Fm values of all involved algorithms are presented in Table 7.

From Table 7, PSO-UEL achieves better results than PSO-W on 15 functions, and it successfully solves multimodal

functions F_7 and F_8 , which means that PSO-UEL has strong exploration ability. Meanwhile, we observe that PSO-EEL performed well in unimodal functions (F_1 , F_2 , F_3 , and F_{15}) and Rosenbrock functions (F_4 , F_{10} , and F_{16}), but it fails to solve the other multimodal functions. This observation means that PSO-EEL is easier to be trapped in the local optima for multimodal functions. By integrating the two learning patterns, PSO-DLP obtains excellent performance.

(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 2: Continued.

(g)

(h)

(i)

(j)

(k)

(l)

FIGURE 2: Continued.

FIGURE 2: Continued.

FIGURE 2: Convergence curves of test functions. (a) F_1 , (b) F_2 , (c) F_3 , (d) F_4 , (e) F_5 , (f) F_6 , (g) F_7 , (h) F_8 , (i) F_9 , and (j) F_{10} . Convergence curves of test functions. (k) F_{11} , (l) F_{12} , (m) F_{13} , (n) F_{14} , (o) F_{15} , (p) F_{16} , (q) F_{17} , (r) F_{18} , (s) F_{19} , and (t) F_{20} .

TABLE 7: Fm values achieved by PSO-W and PSO-DLP variants in 30-D problems.

	PSO-W	PSO-UEL	PSO-EEL	PSO-DLP
F_1	$1.34E - 117 \pm 2.16E - 116$	$5.60E - 58 \pm 6.46E - 59$	$5.93E - 323 \pm \mathbf{0.00E + 00}$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$
F_2	$1.41E - 45 \pm 1.47E - 45$	$4.58E - 10 \pm 6.29E - 10$	$6.94E - 184 \pm 0.00E + 00$	$\mathbf{3.00E - 86 \pm 4.25E - 86}$
F_3	$2.34E - 03 \pm 9.83E - 04$	$1.20E - 03 \pm 9.43E - 04$	$4.06E - 03 \pm 1.90E - 03$	$\mathbf{2.05E - 04 \pm 1.24E - 05}$
F_4	$2.53E + 01 \pm 1.81E + 00$	$2.28E + 01 \pm 6.61E - 01$	$4.78E - 17 \pm 6.08E - 17$	$\mathbf{2.01E - 19 \pm 6.19E - 19}$
F_5	$\mathbf{7.11E - 15 \pm 0.00E + 00}$	$2.84E - 14 \pm 6.15E - 15$	$1.18E - 14 \pm 4.10E - 15$	$\mathbf{7.11E - 15 \pm 0.00E + 00}$
F_6	$7.39E - 03 \pm 1.05E - 02$	$2.13E - 03 \pm 2.09E - 03$	$6.68E - 02 \pm 5.87E - 02$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$
F_7	$2.12E + 01 \pm 1.02E + 01$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$	$1.16E + 01 \pm 1.52E + 00$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$
F_8	$2.37E + 01 \pm 1.45E + 01$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$	$3.33E - 01 \pm 5.77E - 01$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$
F_9	$1.27E + 00 \pm 4.43E - 01$	$7.53E - 01 \pm 3.43E - 01$	$1.46E + 00 \pm 5.58E - 01$	$\mathbf{6.87E - 01 \pm 3.60E - 01}$
F_{10}	$2.51E + 01 \pm 2.14E + 01$	$2.37E + 01 \pm 2.50E - 01$	$5.70E - 01 \pm 5.22E - 01$	$\mathbf{2.00E - 01 \pm 2.83E - 01}$
F_{11}	$1.34E + 00 \pm 3.14E - 01$	$3.20E - 14 \pm 6.15E - 15$	$4.47E - 01 \pm 7.74E - 01$	$\mathbf{7.11E - 15 \pm 0.00E + 00}$
F_{12}	$3.93E - 02 \pm 3.81E - 02$	$2.46E - 02 \pm 3.64E - 02$	$1.40E - 02 \pm 7.09E - 03$	$\mathbf{3.70E - 03 \pm 5.23E - 03}$
F_{13}	$5.17E + 01 \pm 2.25E + 01$	$4.21E + 01 \pm 6.77E + 00$	$6.04E + 01 \pm 9.56E + 00$	$\mathbf{2.87E + 01 \pm 3.68E + 00}$
F_{14}	$4.05E + 01 \pm 1.48E + 01$	$2.67E + 01 \pm 3.06E + 00$	$5.40E + 01 \pm \mathbf{2.00E + 00}$	$\mathbf{2.55E + 01 \pm 1.34E + 01}$
F_{15}	$5.68E - 14 \pm 0.00E + 00$	$1.14E - 13 \pm 0.00E + 00$	$7.58E - 14 \pm 3.28E - 14$	$\mathbf{0.00E + 00 \pm 0.00E + 00}$
F_{16}	$1.04E + 01 \pm 2.11E + 00$	$2.27E + 01 \pm 5.57E - 01$	$\mathbf{6.56E - 11 \pm 6.74E - 11}$	$1.44E - 09 \pm 3.21E - 09$
F_{17}	$2.41E + 01 \pm 1.19E + 01$	$1.89E - 13 \pm 3.28E - 14$	$9.62E + 00 \pm 5.11E + 00$	$\mathbf{1.27E - 14 \pm 1.48E - 15}$
F_{18}	$2.51E + 01 \pm 1.32E + 01$	$1.71E - 13 \pm 5.68E - 14$	$5.11E + 00 \pm 2.33E + 00$	$\mathbf{4.55E - 14 \pm 2.54E - 14}$
F_{19}	$2.06E + 01 \pm 6.74E - 01$	$\mathbf{2.04E + 01 \pm 1.85E - 01}$	$2.05E + 01 \pm \mathbf{3.63E - 02}$	$\mathbf{2.04E + 01 \pm 6.25E - 02}$
F_{20}	$7.78E + 01 \pm 1.38E + 01$	$5.93E + 01 \pm 1.14E + 01$	$9.68E + 01 \pm 4.34E + 01$	$\mathbf{3.68E + 01 \pm 2.81E + 00}$

5.5.5. Performance Comparisons between PSO-DLP and Other Evolutionary Algorithms. In this section, we compare PSO-DLP with the covariance matrix adaptation evolution strategy (CMA-ES) [43], self-adaptive DE (SaDE) [44], and adaptive DE with optional external archive (JADE) [45]. The CMA-ES algorithm is a powerful stochastic optimization technique, where the multivariate normal distribution has a mean and a covariance matrix constantly updated during the evolutionary process. Differential evolution

(DE) has been shown to be an efficient stochastic search algorithm for solving optimization problems. SaDE provides a mutation strategy pool, where the strategy is adaptively selected according to its previous performance. The JADE is another adaptive DE-variant, where a novel mutation strategy is developed and an external archive is employed to store information of progress direction. Simulation results showed that JADE has competitive performances.

TABLE 8: Comparisons between PSO-DLP and other evolutionary algorithms in 30-D problems.

	CMA-ES	JADE	SADE	PSO-DLP
F_1	$4.83E - 131 \pm 4.35E - 131$	$5.70E - 116 \pm 3.19E - 116$	$4.57E - 81 \pm 3.10E + 81$	$0.00E + 00 \pm 0.00E + 00$
F_2	$1.51E - 26 \pm 3.61E - 27$	$2.84E - 75 \pm 3.25E - 76$	$9.37E - 62 \pm 5.44E - 63$	$3.00E - 86 \pm 4.25E - 86$
F_3	$5.31E - 01 \pm 2.50E - 03$	$5.14E - 04 \pm 2.14E - 05$	$1.12E - 03 \pm 6.27E - 04$	$2.05E - 04 \pm 1.24E - 05$
F_4	$7.43E - 02 \pm 3.64E - 03$	$7.09E - 02 \pm 2.54E - 03$	$1.25E + 01 \pm 3.18E + 00$	$2.01E - 19 \pm 6.19E - 19$
F_5	$7.11E - 15 \pm 0.00E + 00$	$4.45E - 15 \pm 3.24E - 15$	$1.18E - 14 \pm 4.10E - 15$	$7.11E - 15 \pm 0.00E + 00$
F_6	$0.00E + 00 \pm 0.00E + 00$	$0.00E + 00 \pm 0.00E + 00$	$0.00E + 00 \pm 0.00E + 00$	$0.00E + 00 \pm 0.00E + 00$
F_7	$1.54E + 01 \pm 7.04E - 01$	$0.00E + 00 \pm 0.00E + 00$	$0.00E + 00 \pm 0.00E + 00$	$0.00E + 00 \pm 0.00E + 00$
F_8	$1.65E + 01 \pm 4.95E + 00$	$0.00E + 00 \pm 0.00E + 00$	$6.24E - 14 \pm 3.54E - 15$	$0.00E + 00 \pm 0.00E + 00$
F_9	$0.00E + 00 \pm 0.00E + 00$	$7.53E - 01 \pm 3.43E - 01$	$1.46E + 00 \pm 5.58E - 01$	$6.87E - 01 \pm 3.60E - 01$
F_{10}	$2.88E + 01 \pm 2.20E - 02$	$1.41E + 01 \pm 4.31E + 00$	$2.98E + 01 \pm 3.08E + 00$	$2.00E - 01 \pm 2.83E - 01$
F_{11}	$1.03E + 00 \pm 6.07E - 02$	$7.34E - 15 \pm 1.56E - 15$	$7.11E - 15 \pm 0.00E + 00$	$7.11E - 15 \pm 0.00E + 00$
F_{12}	$0.00E + 00 \pm 0.00E + 00$	$1.23E - 02 \pm 3.98E - 03$	$8.21E - 03 \pm 9.21E - 04$	$3.70E - 03 \pm 5.23E - 03$
F_{13}	$3.12E + 01 \pm 1.05E - 01$	$2.81E + 01 \pm 3.15E - 01$	$3.14E + 01 \pm 9.56E + 00$	$2.87E + 01 \pm 3.68E + 00$
F_{14}	$7.89E + 01 \pm 3.58E - 01$	$2.71E + 01 \pm 1.32E + 00$	$2.86E + 01 \pm 3.45E - 01$	$2.55E + 01 \pm 1.34E + 01$
F_{15}	$1.77E - 26 \pm 5.19E - 27$	$5.68E - 24 \pm 3.42E - 24$	$4.78E - 23 \pm 9.85E - 24$	$0.00E + 00 \pm 0.00E + 00$
F_{16}	$1.23E + 00 \pm 2.50E - 03$	$1.02E + 01 \pm 1.15E + 00$	$4.86E + 01 \pm 5.78E + 00$	$1.44E - 09 \pm 3.21E - 09$
F_{17}	$1.94E + 01 \pm 7.74E + 00$	$1.89E - 14 \pm 3.28E - 15$	$1.62E - 13 \pm 1.71E - 14$	$1.27E - 14 \pm 1.48E - 15$
F_{18}	$2.14E + 01 \pm 6.34E + 00$	$1.71E - 13 \pm 5.68E - 14$	$2.35E - 13 \pm 5.24E - 14$	$4.55E - 14 \pm 2.54E - 14$
F_{19}	$2.12E + 01 \pm 2.14E - 03$	$2.09E + 01 \pm 1.85E - 01$	$2.09 + 01 \pm 4.61E - 02$	$2.04E + 01 \pm 6.25E - 02$
F_{20}	$4.87E + 01 \pm 1.12E + 01$	$3.95E + 01 \pm 8.35E + 00E$	$5.67E + 01 \pm 3.39E + 00$	$3.68E + 01 \pm 2.81E + 00$
Average rank	2.9	2.15	3.25	1.2
Final rank	3	2	4	1

Twenty benchmark functions as presented in Table 1 are used to test the performance of four algorithms. The parameter values of CMA-ES and DEs are selected according to the recommendations of their respective references, as listed in Table 3. The maximum number of fitness evaluations is set at $1.00E + 5$ and population size N is set at 40 ($N = (4 + \lfloor 3 \log(D) \rfloor)$) for CMA-ES for all algorithms. All functions are tested on 30 dimensions ($D = 30$). All algorithms are run independently 30 times to ensure the fair assessment.

The mean (Fm) and the standard deviation (SD) of the results obtained by each algorithm for all algorithms are summarized in Table 8. From Table 8, we observed that PSO-DLP has better performance on eleven functions and has the same performance with DEs or CMA-ES on six functions. From the final rank, we can obtain the order of the involved algorithms: PSO-DLP, JADE, CMA-ES, and SADE. PSO-DLP has competitive performance among these algorithms on the tested functions.

6. Conclusion

In this paper we study the motion behavior of the swarm and point out that the probabilistic characteristics of the learning factor and the distribution factor in the updating equation can affect the searching behavior of particles. We develop a Particle Swarm Optimization with double learning patterns (PSO-DLP), which uses two swarms with different searching ability and the interaction mechanism between swarms to control exploration and exploitation searches. In the PSO-DLP, the master swarm encourages the exploration process

while the slave swarm concentrates on a small region in order to find its best local optimum easily and accelerate its convergence. The two swarms fulfill their searching tasks by adopting two different learning patterns in which the searching behavior is controlled by adjusting the probabilistic characteristics of the learning factor and the distribution factor.

The significant feature of PSO-DLP is that it has a better balance between high precision and fast convergence. Another attractive property of PSO-DLP is that it employs the probabilistic characteristics of the learning parameters to control the swarm diversity so that it does not need any additional operation.

In our future work, we will study the effect of the topological structure of the uniform exploration learning pattern. And PSO-DLP will be extended to the image segmentation and multiobjective problems.

Appendix

Considering a case where $k = 0.5$, we calculate the joint probability distribution of d_f and l_f :

$$F(d_f, l_f) = P(D_f \leq d_f, L_f \leq l_f) = P(kr_1 + (1-k)r_2 \leq d_f, (1-l_f)kr_1 - l_f(1-k)r_2 \leq 0). \quad (A.1)$$

Since r_1 and r_2 are uniform random variables on $[0, 1]$, we can find

$$F_{D_f L_f}(d_f, l_f) = \begin{cases} 0, & d_f < 0, l_f < 0 \\ 2d_f^2 l_f, & 0 \leq d_f < \frac{1}{2}, 0 \leq l_f < 1 \\ \frac{l_f}{2(1-l_f)}, & \frac{1}{2} \leq d_f < 1, 0 \leq l_f < 1 - \frac{1}{2d_f} \\ 2d_f^2 l_f - \frac{1}{2}(2d_f - 1)^2, & \frac{1}{2} \leq d_f < 1, 1 - \frac{1}{2d_f} \leq l_f < \frac{1}{2d_f} \\ \frac{1}{2}(4d_f - 1) - \frac{1}{2}(2d_f - 1)^2 - \frac{1-l_f}{2l_f}, & \frac{1}{2} \leq d_f < 1, \frac{1}{2d_f} \leq l_f < 1 \\ 1, & d_f \geq 1, l_f \geq 1. \end{cases} \quad (\text{A.2})$$

By the joint probability distribution of d_f and l_f , the joint density of d_f and l_f is

$$f(d_f, l_f) = \frac{F(d_f, l_f)}{\partial d_f \partial l_f} = \begin{cases} 4d_f, & 0 \leq d_f \leq 1, \max\left(0, 1 - \frac{1}{2d_f}\right) \leq l_f \leq \min\left(1, \frac{1}{2d_f}\right) \\ 0, & \text{other.} \end{cases} \quad (\text{A.3})$$

Using the joint density of d_f and l_f , we can calculate the marginal density of l_f and the marginal density of d_f .

As

$$f_{L_f}(l_f) = \int_{-\infty}^{+\infty} f(d_f, l_f) d(d_f), \quad (\text{A.4})$$

the marginal density of l_f is

$$f_{L_f}(l_f) = \begin{cases} \frac{1}{(2(1-l_f)^2)}, & 0 \leq l_f < 0.5 \\ \frac{1}{(2l_f^2)}, & 0.5 \leq l_f \leq 1 \\ 0, & \text{other.} \end{cases} \quad (\text{A.5})$$

The marginal density of d_f is

$$f_{D_f}(d_f) = \begin{cases} 4d_f, & 0 \leq d_f < \frac{1}{2} \\ 4(1-d_f), & \frac{1}{2} \leq d_f \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.6})$$

We can get the conditional distribution of d_f given l_f using the following equation:

$$F_{D_f|L_f}(d_f | l_f) = \frac{(\partial/\partial l_f) F_{D_f L_f}(d_f, l_f)}{(\partial/\partial l_f) F_{L_f}(l_f)}. \quad (\text{A.7})$$

So the conditional distribution of d_f given l_f is

$$F_{D_f|L_f}(d_f | l_f) = \begin{cases} 4d_f^2(1-l_f)^2, & 0 \leq l_f < \frac{1}{2}, 0 \leq d_f < \frac{1}{2(1-l_f)} \\ 1, & 0 \leq l_f < \frac{1}{2}, \frac{1}{2(1-l_f)} \leq d_f \leq 1 \\ 4d_f^2 l_f^2, & \frac{1}{2} \leq l_f < 1, 0 \leq d_f < \frac{1}{2l_f} \\ 1, & \frac{1}{2} \leq l_f \leq 1, \frac{1}{2l_f} \leq d_f \leq 1. \end{cases} \quad (\text{A.8})$$

The conditional density of d_f given l_f is

$$f_{D_f|L_f}(d_f | l_f) = \begin{cases} 8d_f(1-l_f)^2, & 0 \leq l_f < \frac{1}{2}, 0 \leq d_f < \frac{1}{2(1-l_f)} \\ 8d_f l_f^2, & \frac{1}{2} \leq l_f < 1, 0 \leq d_f < \frac{1}{2l_f}. \end{cases} \quad (\text{A.9})$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 61300059 and Provincial Project of Natural Science Research for Anhui Colleges of China under Grants nos. KJ2012Z031 and KJ2012Z024.

References

- [1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, December 1995.
- [2] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, IEEE, Nagoya, Japan, October 1995.
- [3] 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.
- [4] Y. Del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.
- [5] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Applied Soft Computing Journal*, vol. 13, no. 6, pp. 3066–3091, 2013.
- [6] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.
- [7] J.-F. Connolly, E. Granger, and R. Sabourin, "An adaptive classification system for video-based face recognition," *Information Sciences*, vol. 192, pp. 50–70, 2012.
- [8] Y. Shi and R. C. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.
- [9] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [10] R. Kundu, S. Das, R. Mukherjee, and S. Debchoudhury, "An improved particle swarm optimizer with difference mean based perturbation," *Neurocomputing*, vol. 129, pp. 315–333, 2014.
- [11] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, 2007.
- [12] P. S. Andrews, "An investigation into mutation operators for particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1044–1051, Vancouver, Canada, July 2006.
- [13] H. Wang, W. Wang, and Z. Wu, "Particle swarm optimization with adaptive mutation for multimodal optimization," *Applied Mathematics and Computation*, vol. 221, pp. 296–305, 2013.
- [14] R. A. Krohling, "Gaussian swarm: a novel particle swarm optimization algorithm," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, pp. 372–376, IEEE, Singapore, December 2004.
- [15] T. J. Richer and T. M. Blackwell, "The Lévy particle swarm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 808–815, IEEE, Vancouver, Canada, July 2006.
- [16] F. Valdez, P. Melin, and O. Castillo, "A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6459–6466, 2014.
- [17] M.-S. Leu, M.-F. Yeh, and S.-C. Wang, "Particle swarm optimization with grey evolutionary analysis," *Applied Soft Computing Journal*, vol. 13, no. 10, pp. 4047–4062, 2013.
- [18] X. J. Cai, Z. H. Cui, J. C. Zeng, and Y. Tan, "Dispersed particle swarm optimization," *Information Processing Letters*, vol. 105, no. 6, pp. 231–235, 2008.
- [19] Z. Xinchao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing*, vol. 10, no. 1, pp. 119–124, 2010.
- [20] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 42, no. 5, pp. 744–767, 2012.
- [21] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Applied Soft Computing*, vol. 13, no. 6, pp. 3066–3091, 2013.
- [22] M. J. Mahmoodabadi, Z. Salahshoor Mottaghi, and A. Bagheri, "HEPSO: high exploration particle swarm optimization," *Information Sciences*, vol. 273, pp. 101–111, 2014.
- [23] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation (CEC '99)*, pp. 1931–1938, IEEE, Washington, DC, USA, July 1999.
- [24] K. E. Parsopoulos and M. N. Vrahatis, "UPSO: a unified particle swarm scheme," in *Proceedings of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE '04)*, vol. 1 of *Lecture series on Computer and Computational Sciences*, pp. 868–873, Athens, Greece, November 2004.
- [25] 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.
- [26] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 120–127, Honolulu, Hawaii, USA, April 2007.
- [27] D. Zhou, X. Gao, G. Liu, C. Mei, D. Jiang, and Y. Liu, "Randomization in particle swarm optimization for global search ability," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15356–15364, 2011.
- [28] 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.
- [29] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 627–646, 2012.
- [30] H. Huang, H. Qin, Z. Hao, and A. Lim, "Example-based learning particle swarm optimization for continuous optimization," *Information Sciences*, vol. 182, no. 1, pp. 125–138, 2012.
- [31] 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.

- [32] B. Niu, Y. Zhu, X. He, and H. Wu, "MCP SO: a multi-swarm cooperative particle swarm optimizer," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1050–1062, 2007.
- [33] S. Y. Sun and J. W. Li, "A two-swarm cooperative particle swarms optimization," *Swarm and Evolutionary Computation*, vol. 15, pp. 1–18, 2014.
- [34] M. El-Abd and M. S. Kamel, "A cooperative particle swarm optimizer with migration of heterogeneous probabilistic models," *Swarm Intelligence*, vol. 4, no. 1, pp. 57–89, 2010.
- [35] M. A. M. De Oca, J. Peña, T. Stützle, C. Pinciroli, and M. Dorigo, "Heterogeneous particle swarm optimizers," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 698–705, IEEE, Trondheim, Norway, May 2009.
- [36] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with local search," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, vol. 1, pp. 522–528, Edinburgh, Scotland, September 2005.
- [37] J. Riget and J. Vesterstroem, *A Diversity-Guided Particle Swarm Optimizer—The ARPSO*, Department of Computer Science, University of Aarhus, 2002.
- [38] J. Jie, J. Zeng, C. Han, and Q. Wang, "Knowledge-based cooperative particle swarm optimization," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 861–873, 2008.
- [39] P. N. Suganthan, N. Hansen, J. J. Liang et al., *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*, Nanyang Technological University, 2005.
- [40] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [41] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.
- [42] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [43] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.
- [44] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [45] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.