Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions

J. J. Liang, A. K. Qin, *Student Member, IEEE*, Ponnuthurai Nagaratnam Suganthan, *Senior Member, IEEE*, and S. Baskar

***Abstract*—This paper presents a variant of particle swarm optimizers (PSOs) that we call the comprehensive learning particle swarm optimizer (CLPSO), which uses a novel learning strategy whereby all other particles' historical best information is used to update a particle's velocity. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence. Experiments were conducted (using codes available from http://www.ntu.edu.sg/home/epnsugan) on multimodal test functions such as Rosenbrock, Griewank, Rastrigin, Ackley, and Schwefel and composition functions both with and without coordinate rotation. The results demonstrate good performance of the CLPSO in solving multimodal problems when compared with eight other recent variants of the PSO.**

***Index Terms*—Composition benchmark functions, comprehensive learning particle swarm optimizer (CLPSO), global numerical optimization, particle swarm optimizer (PSO).**

I. Introduction

OPTIMIZATION has been an active area of research for several decades. As many real-world optimization problems become increasingly complex, better optimization algorithms are always needed. Unconstrained optimization problems can be formulated as a D-dimensional minimization problem as follows:

$$\text{Min } f(\boldsymbol{x}), \boldsymbol{x} = [x_1, x_2, \ldots, x_D]$$

where D is the number of the parameters to be optimized.

The particle swarm optimizer (PSO) [1], [2] is a relatively new technique. Although PSO shares many similarities with evolutionary computation techniques, the standard PSO does not use evolution operators such as crossover and mutation. PSO emulates the swarm behavior of insects, animals herding, birds flocking, and fish schooling where these swarms search for food in a collaborative manner. Each member in the swarm adapts its search patterns by learning from its own experience and other members' experiences. These phenomena are studied and mathematical models are constructed. In PSO, a member in the swarm, called a *particle,* represents a potential solution which is a point in the search space. The global optimum is regarded as the location of food. Each particle has a fitness value and a velocity to adjust its flying direction according to the best experiences of the swarm to search for the global optimum in the D-dimensional solution space.

The PSO algorithm is easy to implement and has been empirically shown to perform well on many optimization problems. However, it may easily get trapped in a local optimum when solving complex multimodal problems. In order to improve PSO's performance on complex multimodal problems, we present the comprehensive learning particle swarm optimizer (CLPSO) utilizing a new learning strategy.

This paper is organized as follows. Section II introduces the original PSO and some current variants of the original PSO. Section III describes the comprehensive learning particle swarm optimizer. Section IV presents the test functions, the experimental setting for each algorithm, the results, and discussions. Conclusions are given in Section V.

II. Particle Swarm Optimizers

A. Particle Swarm Optimizer

PSO emulates the swarm behavior and the individuals represent points in the D-dimensional search space. A particle represents a potential solution. The velocity V_i^d and position X_i^d of the dth dimension of the ith particle are updated as follows [1], [2]:

$$V_i^d \leftarrow V_i^d + c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * \left(gbest^d - X_i^d\right) \quad (1)$$

$$X_i^d \leftarrow X_i^d + V_i^d \quad (2)$$

where $\boldsymbol{X}_i = (X_i^1, X_i^2, \ldots, X_i^D)$ is the position of the ith particle; $\boldsymbol{V_i} = (V_i^1, V_i^2, \ldots, V_i^D)$ represents velocity of particle i. $\boldsymbol{pbest}_i = (pbest_i^1, pbest_i^2, \ldots, pbest_i^D)$ is the best previous position yielding the best fitness value for the ith particle; and $\boldsymbol{gbest} = (gbest^1, gbest^2, \ldots, gbest^D)$ is the best position discovered by the whole population. c_1 and c_2 are the acceleration constants reflecting the weighting of stochastic acceleration terms that pull each particle toward $pbest$ and $gbest$ positions, respectively. $rand1_i^d$ and $rand2_i^d$ are two random numbers in the range [0, 1]. A particle's velocity on each dimension is clamped to a maximum magnitude $Vmax$. If $|V_i^d|$ exceeds a positive constant value $\mathrm{V}_{\max}^{\mathrm{d}}$ specified by the user, then the velocity of that dimension is assigned to $sign(|V_i^d|)\mathrm{V}_{\max}^{\mathrm{d}}$.

When updating the velocity of a particle using (1), different dimensions have different $rand1_i^d$ and $rand2_i^d$. Some researchers use the following updating equation:

$$V_i^d \leftarrow V_i^d + c_1 * rand1_i * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i * \left(gbest^d - X_i^d\right). \quad (3)$$

Manuscript received May 3, 2005; revised July 15, 2005.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore (e-mail: liangjing@pmail.ntu.edu.sg; qinkai@pmail.ntu.edu.sg; epnsugan@ntu.edu.sg; baskar_mani@yahoo.com).

Digital Object Identifier 10.1109/TEVC.2005.857610

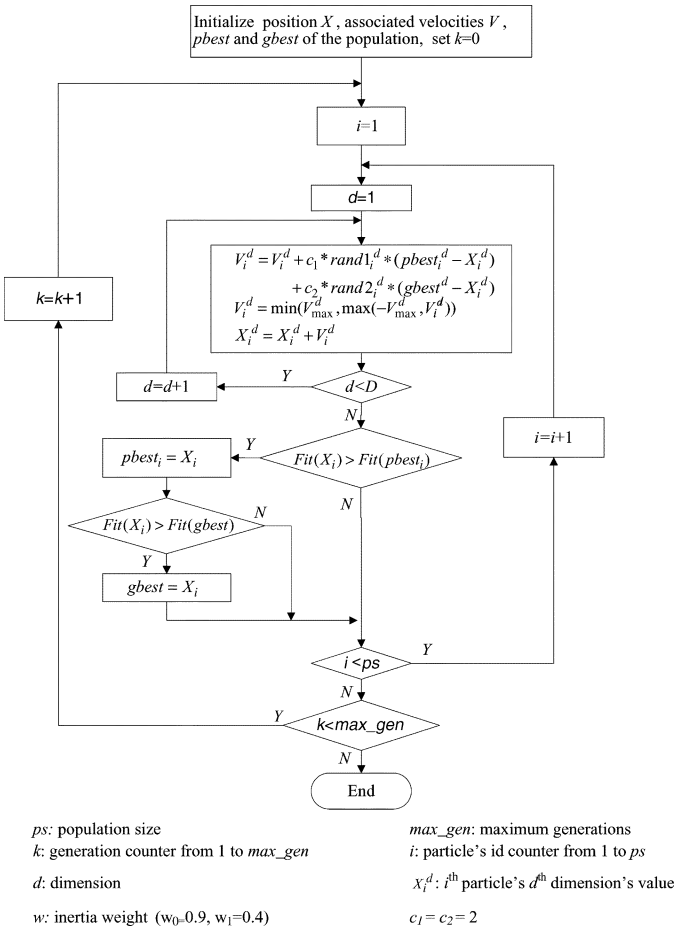


Fig. 1. Flowchart of the conventional PSO.

Comparing the two variants in (1) and (3), the former can have a larger search space due to independent updating of each dimension, while the second is dimension-dependent and has a smaller search space due to the same random numbers being used for all dimensions. Equation (1) always yields better performance on unrotated problems than the rotated version of the problems. Equation (3) performs consistently on unrotated and rotated problems [3]. As the first updating strategy achieves a larger search space and always performs better, we use (1) in this paper. The flowchart of the standard PSO employing (1) is given in Fig. 1.

B. Some Variants PSO

Since its introduction in 1995 by Kennedy and Eberhart [1], [2], PSO has attracted a high level of interest [4]–[7]. Many researchers have worked on improving its performance in various ways, thereby deriving many interesting variants. One of the variants [8] introduces a parameter called inertia weight w into the original PSO algorithms as follows:

$$V_i^d \leftarrow w * V_i^d + c_1 * rand1_i^d * \left(pbest_i^d - X_i^d\right) + c_2 * rand2_i^d * \left(gbest^d - X_i^d\right). \quad (4)$$

The inertia weight is used to balance the global and local search abilities. A large inertia weight is more appropriate for global search, and a small inertia weight facilitates local search. A linearly decreasing inertia weight over the course of search was proposed by Shi and Eberhart [8]. Parameters in PSO are discussed in [9]. Shi and Eberhart designed fuzzy methods to nonlinearly change the inertia weight [10]. In [11], inertia weight is set at zero, except at the time of reinitialization. In addition to the time-varying inertia weight, a linearly decreasing $\boldsymbol{V_{max}}$ is introduced in [12]. By analyzing the convergence behavior of the PSO, a PSO variant with a constriction factor was introduced by Clerc and Kennedy [13]. Constriction factor guarantees the convergence and improves the convergence velocity.

Improving PSO's performance by designing different types of topologies has been an active research direction. Kennedy [14], [15] claimed that PSO with a small neighborhood might perform better on complex problems, while PSO with a large neighborhood would perform better on simple problems. Suganthan [16] applied a dynamically adjusted neighborhood where the neighborhood of a particle gradually increases until it includes all particles. In [17], Hu and Eberhart also used a dynamic neighborhood where m closest particles in the performance space are selected to be its new neighborhood in each generation. Parsopoulos and Vrahatis combined the global version and local version together to construct a unified particle swarm optimizer (UPSO) [18]. Mendes and Kennedy introduced a fully informed PSO in [19]. Instead of using the $pbest$ and $gbest$ positions in the standard algorithm, all the neighbors of the particle are used to update the velocity. The influence of each particle to its neighbors is weighted based on its fitness value and the neighborhood size. Veeramachaneni *et al.* developed the fitness-distance-ratio-based PSO (FDR-PSO) with near neighbor interactions [20]. When updating each velocity dimension, the FDR-PSO algorithm selects one other particle $nbest$, which has a higher fitness value and is nearer to the particle being updated.

Some researchers investigated hybridization by combining PSO with other search techniques to improve the performance of the PSO. Evolutionary operators such as selection, crossover, and mutation have been introduced to the PSO to keep the best particles [21], to increase the diversity of the population, and to improve the ability to escape local minima [22]. Mutation operators are also used to mutate parameters such as the inertia weight [23]. Relocating the particles when they are too close to each other [24] or using some collision-avoiding mechanisms [25] to prevent particles from moving too close to each other in order to maintain the diversity and to escape from local optima has also been used. In [22], the swarm is divided into subpopulations, and a breeding operator is used within a subpopulation or between the subpopulations to increase the diversity of the population. Negative entropy is used to discourage premature convergence in [27]. In [28], deflection, stretching, and repulsion techniques are used to find as many minima as possible by preventing particles from moving to a previously discovered minimal region. Recently, a cooperative particle swarm optimizer (CPSO-H) [29] was proposed. Although CPSO-H uses one-dimensional (1-D) swarms to search each dimension separately, the results of these searches are integrated by a global swarm to significantly improve the performance of the original PSO on multimodal problems.

III. Comprehensive Learning Particle Swarm Optimizer

Although there are numerous variants for the PSO, premature convergence when solving multimodal problems is still the main deficiency of the PSO. In the original PSO, each particle learns from its $pbest$ and $gbest$ simultaneously. Restricting the social learning aspect to only the $gbest$ makes the original PSO converge fast. However, because all particles in the swarm learn from the $gbest$ even if the current $gbest$ is far from the global optimum, particles may easily be attracted to the $gbest$ region and get trapped in a local optimum if the search environment is complex with numerous local solutions. As $f(\boldsymbol{x}) = f([x^1, x^2, \ldots, x^D])$, the fitness value of a particle is possibly determined by values of all D parameters, and a particle that has discovered the region corresponding to the global optimum in some dimensions may have a low fitness value because of the poor solutions in the other dimensions. In order to make better use of the beneficial information, we proposed new learning strategies to improve the original PSO [30]. In [30], all particles' $pbests$ are used to update the velocity of any one particle. This novel strategy ensures that the diversity of the swarm is preserved to discourage premature convergence. Three versions of PSO using the comprehensive learning strategy were discussed and demonstrated with significantly improved performances on solving multimodal problems in comparison to several other variants of the PSO. Among the three variants, the CLPSO is the best, based on the results. Hence, we further investigate the CLPSO in this paper.

A. Compehensive Learning Strategy

In this new learning strategy, we use the following velocity updating equation:

$$V_i^d \leftarrow w * V_i^d + c^* rand_i^d * \left(pbest_{fi(d)}^d - X_i^d\right) \qquad (5)$$

where $\boldsymbol{f}_i = [f_i(1), f_i(2), \ldots, f_i(D)]$ defines which particles' $pbest$s the particle i should follow. $pbest_{fi(d)}^d$ can be the corresponding dimension of any particle's $pbest$ including its own $pbest$, and the decision depends on probability Pc, referred to as the learning probability, which can take different values for different particles. For each dimension of particle i, we generate a random number. If this random number is larger than Pc_i, the corresponding dimension will learn from its own $pbest$; otherwise it will learn from another particle's $pbest$. We employ the tournament selection procedure when the particle's dimension learns from another particle's $pbest$ as follows.

1) We first randomly choose two particles out of the population which excludes the particle whose velocity is updated.
2) We compare the fitness values of these two particles' $pbest$s and select the better one. In CLPSO, we define the fitness value the larger the better, which means that when solving minimization problems, we will use the negative function value as the fitness values.
3) We use the winner's $pbest$ as the exemplar to learn from for that dimension. If all exemplars of a particle are its own $pbest$, we will randomly choose one dimension to learn from another particle's $pbest$'s corresponding dimension. The details of choosing $\boldsymbol{f}_i$ are given in Fig. 2.

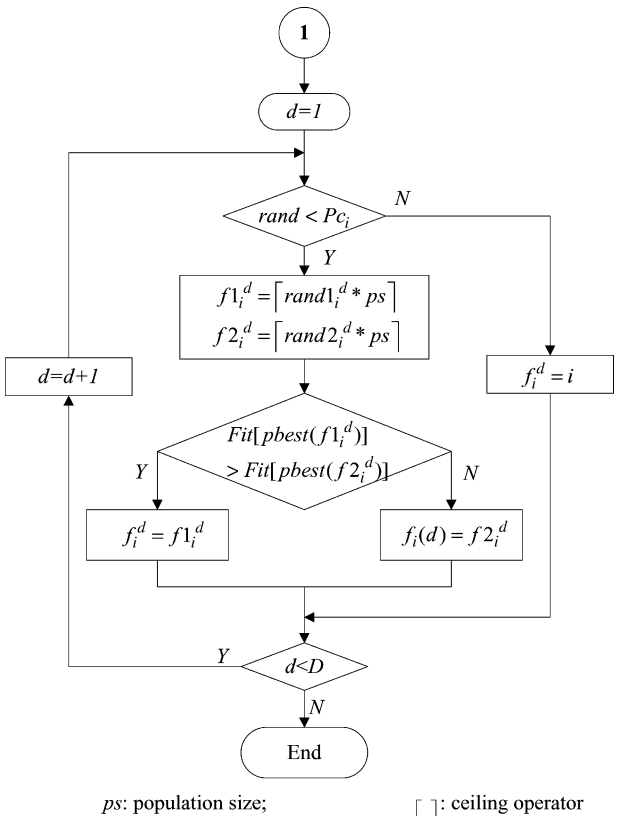


Fig. 2. Selection of exemplar dimensions for particle i.

All these $pbest_{fi}$ can generate new positions in the search space using the information derived from different particles' historical best positions. To ensure that a particle learns from good exemplars and to minimize the time wasted on poor directions, we allow the particle to learn from the exemplars until the particle ceases improving for a certain number of generations called the refreshing gap m, then we reassign f_i for the particle. We observe three main differences between the CLPSO and the original PSO [2].

1) Instead of using particle's own $pbest$ and $gbest$ as the exemplars, all particles' $pbest$s can potentially be used as the exemplars to guide a particle's flying direction.
2) Instead of learning from the same exemplar particle for all dimensions, each dimension of a particle in general can learn from different $pbest$s for different dimensions for a few generations. In other words, each dimension of a particle may learn from the corresponding dimension of different particle's $pbest$.
3) Instead of learning from two exemplars ($gbest$ and $pbest$) at the same time in every generation as in the original PSO (1) and (3), each dimension of a particle learns from just one exemplar for a few generations.

B. CLPSO's Search Behavior

The above operations increase the swarm's diversity to yield improved performance when solving complex multimodal problems. In the original PSO, for a certain dimension, if the $pbest$ and $gbest$ are on opposite sides of the particle's current position

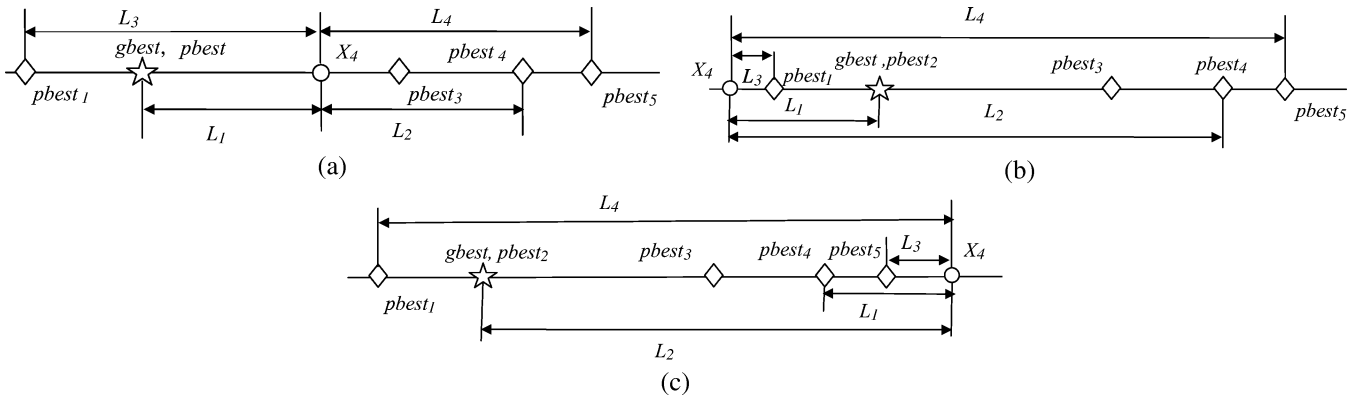


Fig. 3. The CLPSO's and the original PSO's possible search regions per variable in a swarm with five members. (a) $X_4 \geq \min(pbest_i)$. $X_4 \leq \max(pbest_i)$. (b) $X_4 < \min(pbest_i)$. (c) $X_4 > \max(pbest_i)$.

$\boldsymbol{X}$, the $pbest$ and $gbest$ may make the particle oscillate. However, the $gbest$ is more likely to provide a larger momentum, as $|\boldsymbol{gbest}-\boldsymbol{X}|$ is likely to be larger than the $|\boldsymbol{pbest}-\boldsymbol{X}|$. Hence, the $gbest$ may influence the particle to move in its direction even if it is in a local optimum region far from the global optimum. If $pbest$ and $gbest$ are on the same side of the particle's current position and if it points to a local optimum, the particle will move in that direction and it may be impossible to jump out of the local optimum area once its $pbest$ falls into the same local optimum region where the $gbest$ is. However, in our new learning strategy, the particle can fly in other directions by learning from other particles' $pbest$ when the particle's $pbest$ and $gbest$ fall into the same local optimum region. Hence, our new learning strategy has the ability to jump out of local optimum via the cooperative behavior of the whole swarm.

In order to compare the original PSO's and CLPSO's potential search spaces, first we omit the old velocity $w * V_i^d$ component. If we let c_1, c_2 in the original PSO and c in CLPSO all be equal to one, the update equations of the original PSO and CLPSO reduce to the following equations:

$$\begin{aligned}\text{PSO}: V_i^d \leftarrow & rand1_i^d * \left(pbest_i^d - X_i^d\right) \\ & + rand2_i^d * \left(gbest^d - X_i^d\right) \qquad (6)\\ \text{CLPSO}: V_i^d \leftarrow & rand_i^d * \left(pbest_{fi(d)}^d - X_i^d\right). \qquad (7)\end{aligned}$$

Let us consider the fourth particle in a swarm with five members as an example. The potential search spaces of the original PSO and the CLPSO on one dimension are plotted as a line in Fig. 3. For the fourth particle whose position is X_4, three different cases are illustrated in Fig. 3: (a) $X_4 \geq \min(pbest_j)$ and $X_4 \leq \max(pbest_j)$; (b) $X_4 < \min(pbest_j)$; and (c) $X_4 > \max(pbest_j)$, $j = [1, 2, 3, 4, 5]$. In this example, $pbest_2$ is the $gbest$, $\min(pbest_j)$ is the $pbest_1$, and $\max(pbest_j)$ is the $pbest_5$,

We extend the three cases to the dth dimension of the ith particle in a swarm of size ps. Let the length of the potential space of the PSO and CLPSO for the dth dimension of the ith particle be r_{1i}^d and r_{2i}^d, respectively. We obtain the potential search ranges for the ith particle of PSO as

$$r_{1i}^d = L_{1i}^d + L_{2i}^d = \left|gbest^d - X_i^d\right| + \left|pbest_i^d - X_i^d\right| \qquad (8)$$

and CLPSO, as shown in (9) at the bottom of the page.

Hence, the volumes of PSO's and CLPSO's potential search spaces for the ith particle are $R_{1i} = \prod_i^D r_{1i}$ and $R_{2i} = \prod_i^D r_{2i}$, respectively. $\overline{R_1}$ and $\overline{R_2}$ are the mean values of the volumes of PSO's and CLPSO's potential search spaces for the whole swarm. In order to compare the potential search spaces of PSO and CLPSO, both algorithms are run 20 times on a (unimodal) sphere function and a (multimodal) Rastrigin function defined in Section IV-A. $\overline{R_1}$, $\overline{R_2}$, and $\overline{\mathrm{R_2}/\mathrm{R_1}}$ in each iteration are recorded. Table I presents $\overline{\overline{\mathrm{R_2}/\mathrm{R_1}}}$'s mean value of the 20 runs. $\overline{R_1}$ and $\overline{R_2}$ and $\overline{\mathrm{R_2}/\mathrm{R_1}}$ versus the iterations are plotted in Fig. 4.

From Table I and Fig. 4, we observe that CLPSO's updating strategy yields a larger potential search space than that of the original PSO. The multimodal Rastrigin's function's mean $(\overline{\overline{\mathrm{R_2}/\mathrm{R_1}}})$ is ten times larger than that of the unimodal sphere function. By increasing each particle's potential search space, the diversity is also increased. As each particle's $pbest$ is possibly a good area, the search of CLPSO is neither blind nor random. Compared to the original PSO, CLPSO searches more promising regions to find the global optimum. Experimental results in Section IV support this description.

C. Learning Probability Pc

From our previous experiments [31], we found that different Pc values yielded different results on the same problem if the same Pc value was used for all the particles in the population. On unrotated problems, smaller Pc values perform better,

$$\begin{cases} \text{if} \quad X_i^d \geq \min\left(pbest_j^d\right) \& X_i^d \leq \max\left(pbest_j^d\right), & then \quad r_{2i}^d = L_{3i}^d + L_{4i}^d = \max\left(pbest_j^d\right) - \min\left(pbest_j^d\right) \\ \text{if} \quad X_i^d < \min\left(pbest_j^d\right), & then \quad r_{2i}^d = \max\left(L_{3i}^d, L_{4i}^d\right) = \max\left(pbest_j^d\right) - X_i^d \\ \text{if} \quad X_i^d > \max\left(pbest_j^d\right), & then \quad r_{2i}^d = \max\left(L_{3i}^d, L_{4i}^d\right) = X_i^d - \min\left(pbest_j^d\right) \end{cases}$$
$$\Rightarrow r_{2i}^d = \max\left(pbest_j^d, X_i^d\right) - \min\left(pbest_j^d, X_i^d\right), \quad i = 1, 2, \ldots, ps,\ j = 1, 2, \ldots, ps,\ d = 1, 2, \ldots, D \qquad (9)$$

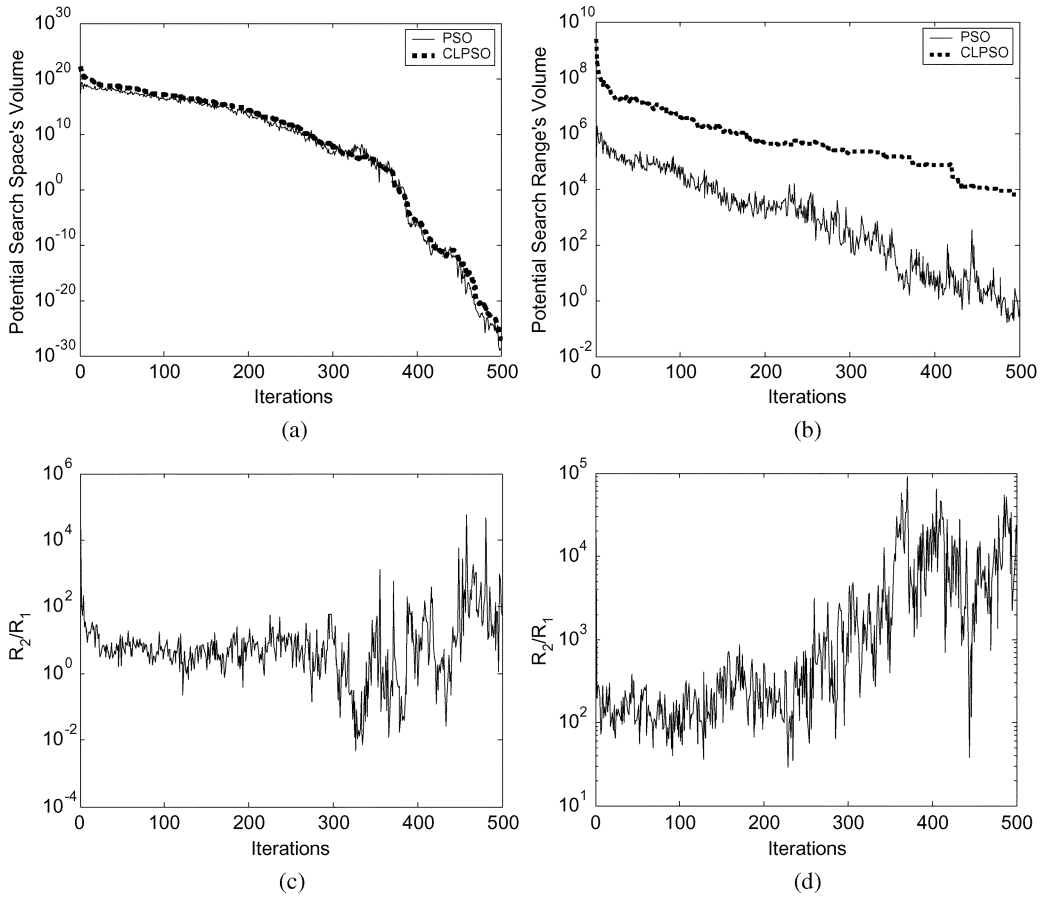


Fig. 4. Comparison of PSO's and CLPSO's potential search space. (a) $\overline{R_1}$ and $\overline{R_2}$ for sphere function. (b) $\overline{R_1}$ and $\overline{R_2}$ for Rastrigin's function. (c) $\overline{\mathrm{R}_2}/\overline{\mathrm{R}_1}$ for sphere function. (d) $\overline{\mathrm{R}_2}/\overline{\mathrm{R}_1}$ for Rastrigin's function.

TABLE I
MEAN VALUE OF $\overline{\mathrm{R}_2}/\overline{\mathrm{R}_1}$ FOR SPHERE AND RASTRIGIN FUNCTIONS IN 20 RUNS

	Sphere	Rastrigin
mean($\overline{\overline{\mathrm{R}_2}/\overline{\mathrm{R}_1}}$)	4.2059e+003	1.5971e+004

while on the rotated problems, different Pc values yield the best performance for different problems. Different Pc values yield similar results on simple unimodal problems while seriously affecting CLPSO's performance on multimodal problems. In order to address this problem in a general manner, we propose to set Pc such that each particle has a different Pc value. Therefore, particles have different levels of exploration and exploitation ability in the population and are able to solve diverse problems. We empirically developed the following expression to set a Pc_i value for each particle:

$$Pc_i = 0.05 + 0.45 * \frac{\left(\exp\left(\frac{10(i-1)}{ps-1}\right) - 1\right)}{(\exp(10) - 1)}. \quad (10)$$

Fig. 5 presents an example of Pc assigned for a population of 30. Each particle from 1 to 30 has a Pc value ranging from 0.05 to 0.5.

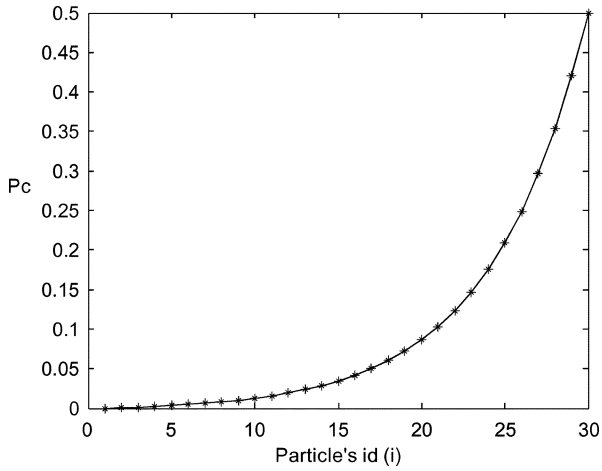
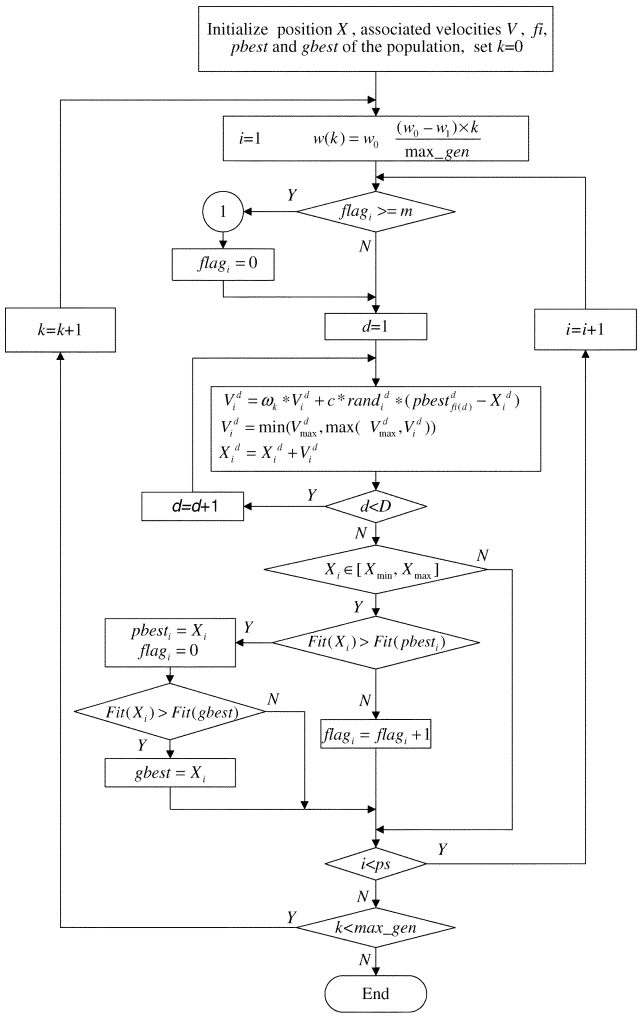


Fig. 5. Each particle's Pc with a population size of 30.

D. Implementation of Search Bounds

Though we have shown in [30] the CLPSO to be robust to initialization and independent of upper and lower search bounds, in many practical problems, there are bounds on the variables' ranges. The search range for a problem is $[X_{\min}, X_{\max}]$. In order to prevent particles moving out of the search bounds, some researchers use the equation $X_i^d = \min(X_{\max}^d, \max(X_{\min}^d, X_i^d))$ to restrain a particle on the border. In our algorithm, we use a different method to constrain the particles within the range as follows: Calculate the fitness value of a particle and update its $pbest$ and $gbest$ only if the particle is in the range. Since all exemplars are within the



(1) :Insert the flowchart in Fig. 1 here.

ps: population size
max_gen: maximum generations
k: generation counter from 1 to *max_gen*
i: particle's id counter from 1 to *ps*
d: dimension
X_i^d: i^{th} particle's d^{th} dimension's value
w: inertia weight (w_0=0.9, w_1=0.4)
c=1.49445 *m*: refreshing gap

$flag_i$: the number of generations the i^{th} particle has not improved its own *pbest*.

Fig. 6. Flowchart of the CLPSO algorithm.

range, the particle will eventually return to the search range. The complete flowchart of the CLPSO is given in Fig. 6.

E. Adjusting the Refreshing Gap m

The refreshing gap parameter m needs be tuned. In this section, six different kinds of ten-dimensional (10-D) test functions are used to investigate the impact of this parameter. They are the sphere, Rosenbrock, Ackley, Griewank, Rastrigin, and rotated Rastrigin functions as defined in Section IV. The CLPSO is run 20 times on each of these functions, and the mean values of the final results are plotted in Fig. 7. As all test functions are minimization problems, the smaller the final result, the better it is. From Fig. 7, we can observe that m can influence the results. When m is zero, we obtained a faster convergence velocity and better results on the sphere function. For the other five test functions, better results were obtained when m is around seven. Hence, in our experiments, the refreshing gap m is set at seven for all test functions.

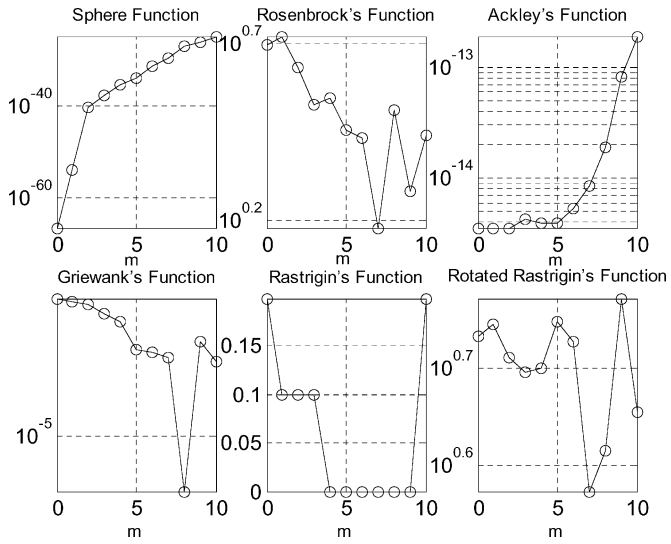


Fig. 7. CLPSO's results on six test functions with different refreshing gap m.

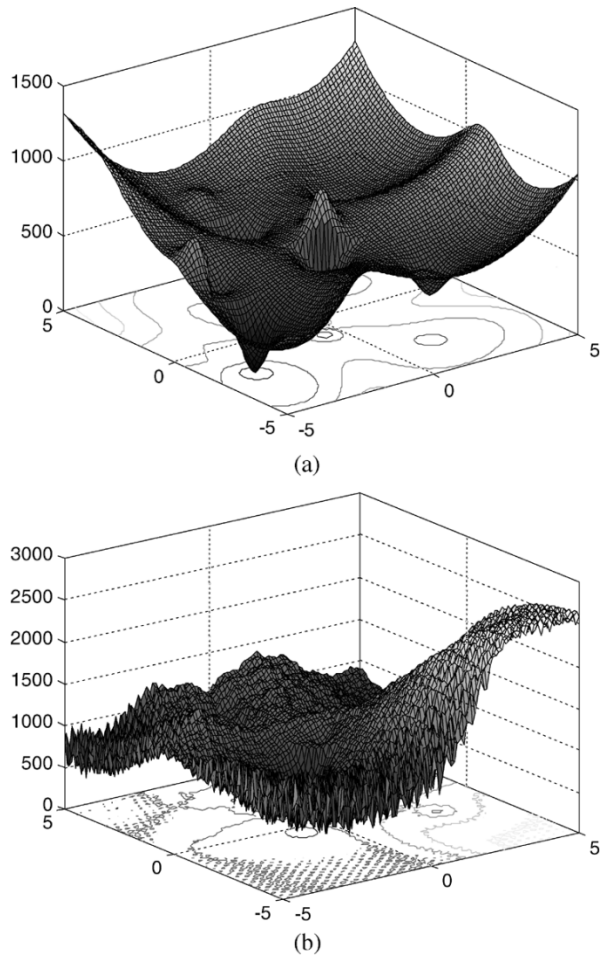


Fig. 8. The landscape maps of Group D problems. (a) Composition function 1 (CF1). (b) Composition function 5 (CF5).

IV. Experimental Results and Discussion

A. Test Functions

As we wish to test the CLPSO on diverse test functions and our main objective is to improve PSO's performance on multimodal problems, we choose two unimodal functions and 14 multimodal benchmark functions [32]–[35]. All functions are tested on ten and 30 dimensions. According to their properties, these functions are divided into four groups: *unimodal problems, unrotated multimodal problems, rotated multimodal problems,* and *composition problems.* The properties and the formulas of these functions are presented below.

Group A: Unimodal and Simple Multimodal Problems:

1) Sphere function

$$f_1(x) = \sum_{i=1}^{D} x_i^2. \tag{11}$$

2) Rosenbrock's function

$$f_2(x) = \sum_{i=1}^{D-1} \left(100\left(x_i^2 - x_{i+1}\right)^2 + (x_i - 1)^2\right). \tag{12}$$

The first problem is the sphere function and is easy to solve. The second problem is the Rosenbrock function. It can be treated as a multimodal problem. It has a narrow valley from the perceived local optima to the global optimum. In the experiments below, we find that the algorithms that perform well on sphere function also perform well on Rosenbrock function.

Group B: Unrotated Multimodal Problems:

3) Ackley's function

$$f_3(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e. \tag{13}$$

4) Griewanks's function

$$f_4(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \tag{14}$$

5) Weierstrass function

$$f_5(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k \cos(2\pi b^k \cdot 0.5)\right], \quad a = 0.5,\ b = 3,\ kmax = 20. \tag{15}$$

6) Rastrigin's function

$$f_6(x) = \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right). \tag{16}$$

7) Noncontinuous Rastrigin's function

$$f_7(x) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right)$$
$$y_i = \begin{cases} x_i & |x_i| < \frac{1}{2} \\ \frac{round(2x_i)}{2} & |x_i| >= \frac{1}{2} \end{cases} \text{ for } i = 1, 2, \ldots, D. \tag{17}$$

8) Schwefel's function

$$f_8(x) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin\left(|x_i|^{\frac{1}{2}}\right). \tag{18}$$

In this group, there are six multimodal test functions. Ackley's function has one narrow global optimum basin and many minor local optima. It is probably the easiest problem among the six as its local optima are shallow. Griewank's function has a $\prod_{i=1}^{D}\cos(x_i/\sqrt{i})$ component causing linkages among variables, thereby making it difficult to reach the global optimum. An interesting phenomenon of Griewank's function is that it is more difficult for lower dimensions than higher dimensions [36]. The Weierstrass function is continuous but differentiable only on a set of points. Rastrigin's function is a complex multimodal problem with a large number of local optima. When attempting to solve Rastrigin's function, algorithms may easily fall into a local optimum. Hence, an algorithm capable of maintaining a larger diversity is likely to yield better results. Noncontinuous Rastrigin's function is constructed based on the Rastrigin's function and has the same number of local optima as the continuous Rastrigin's function. The complexity of Schwefel's function is due to its deep local optima being far from the global optimum. It will be hard to find the global optimum if many particles fall into one of the deep local optima.

Group C: Rotated Multimodal Problems: In Group B, some functions are separable and can be solved by using D 1-D searches, where D is the dimensionality of the problem. Hence, in Group C, we have four rotated multimodal problems. To rotate a function, first an orthogonal matrix $\mathbf{M}$ should be generated. The original variable $\boldsymbol{x}$ is left multiplied by the orthogonal matrix $\mathbf{M}$ to get the new rotated variable $\boldsymbol{y} = \mathbf{M} * \boldsymbol{x}$. This variable $\boldsymbol{y}$ is used calculate the fitness value f

$$\text{If } \mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & \ldots & m_{1D} \\ m_{21} & m_{22} & \ldots & m_{2D} \\ \ldots & \ldots & \ldots & \ldots \\ m_{D1} & m_{D2} & \ldots & m_{DD} \end{bmatrix}$$
$$\boldsymbol{x} = [x_1, x_2, \ldots, x_D]^T \text{ and } \boldsymbol{y} = [y_1, y_2, \ldots, y_D]^T \tag{19}$$
$$\text{then } y_i = m_{i1}x_1 + m_{i2}x_2 + \ldots + m_{iD}x_D, \quad i = 1, 2, \ldots, D. \tag{20}$$

When one dimension in $\boldsymbol{x}$ vector is changed, all dimensions in vector $\boldsymbol{y}$ will be affected. Hence, the rotated function cannot be solved by just D one-dimensional searches. The orthogonal rotation matrix does not affect the shape of the functions. In this paper, we used Salomon's method [37] to generate the orthogonal matrix.

9) Rotated Ackley's function

$$f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi y_i)\right) + 20 + e, \quad \boldsymbol{y} = \mathbf{M} * \boldsymbol{x}. \tag{21}$$

10) Rotated Griewanks's function

$$f_{10}(x) = \sum_{i=1}^{D}\frac{y_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{y_i}{\sqrt{i}}\right) + 1, \ \boldsymbol{y} = \mathbf{M} * \boldsymbol{x}. \tag{22}$$

11) Rotated Weierstrass function

$$f_{11}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k(y_i+0.5)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k\cos(2\pi b^k\cdot 0.5)\right], \quad a=0.5,\ b=3,\ kmax=20,\ \boldsymbol{y}=\mathbf{M}*\boldsymbol{x}. \tag{23}$$

12) Rotated Rastrigin's function

$$f_{12}(x) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right), \ \boldsymbol{y} = \mathbf{M} * \boldsymbol{x}. \tag{24}$$

13) Rotated noncontinuous Rastrigin's function

$$f_{13}(x) = \sum_{i=1}^{D}\left(z_i^2 - 10\cos(2\pi z_i) + 10\right)$$
$$z_i = \begin{cases} y_i & |y_i| < \frac{1}{2} \\ \frac{round(2y_i)}{2} & |y_i| >= \frac{1}{2} \end{cases}$$
$$\text{for } i = 1, 2, \ldots, D, \ \boldsymbol{y} = \mathbf{M}^*\boldsymbol{x}. \tag{25}$$

14) Rotated Schwefel's function

$$f_{14}(x) = 418.9829 \times D - \sum_{i=1}^{D} z_i$$
$$z_i = \begin{cases} y_i\sin\left(|y_i|^{\frac{1}{2}}\right) & if\ |y_i| <= 500 \\ 0.001\left(|y_i| - 500\right)^2 & if\ |y_i| > 500 \end{cases},$$
$$\text{for } i = 1, 2, \ldots, D$$
$$\boldsymbol{y} = \boldsymbol{y}' + 420.96,\ \boldsymbol{y}' = \mathbf{M} * (\boldsymbol{x} - 420.96). \tag{26}$$

In rotated Schwefel's function, in order to keep the global optimum in the search range after rotation, noting that the original global optimum of Schwefel's function is at [420.96, 420.96, ..., 420.96], $\boldsymbol{y}' = \mathbf{M} * (\boldsymbol{x} - 420.96)$ and $\boldsymbol{y} = \boldsymbol{y}' + 420.96$ are used instead of $\boldsymbol{y} = \mathbf{M}*\boldsymbol{x}$. Since Schwefel's function has better solutions out of the search range $[-500\ 500]^D$, when $|y_i| > 500$, $z_i = 0.001(|y_i| - 500)^2$, i.e. z_i is set in portion to the square distance between y_i and the bound.

Group D: Composition Problems: Composition functions are constructed using some basic benchmark functions to obtain more challenging problems with a randomly located global optimum and several randomly located deep local optima. The Gaussian function is used to combine the simple benchmark functions and blur the function's structures. The composition functions are asymmetrical multimodal problems, with different properties in different areas. The details of how to construct this class of functions and six composition functions are presented in [38].[1] Two of the six composition functions defined in [38] are used here to test the CLPSO.

15) Composition function 1 (CF1) in [38]: f_{15} (CF1) is composed using ten sphere functions. The global optimum is easy to find once the global basin is found.

16) Composition function 5 (CF5) in [38]: f_{16} (CF2) is composed using ten different benchmark functions: two rotated Rastrigin's functions, two rotated Weierstrass functions, two rotated Griewank's functions, two rotated Ackley's functions, and two sphere functions. CF5 is more complex than CF1 since even after the global basin is found, the global optimum is not easy to locate. The landscape maps of these two composition functions are illustrated in Fig. 8.

The global optima $\boldsymbol{x}^*$, the corresponding fitness value $f(\boldsymbol{x}^*)$, the search ranges $[\boldsymbol{X_{min}}, \boldsymbol{X_{max}}]$, and the initialization range of each function are given in Table II. Biased initializations are used for the functions whose global optimum is at the centre of the search range.

B. Parameter Settings for the Involved PSO Algorithms

Experiments were conducted to compare nine PSO algorithms including the proposed CLPSO algorithm on the 16 test problems with ten dimensions and 30 dimensions. The algorithms and parameters settings are listed below:

- PSO with inertia weight (PSO-w) [8];
- PSO with constriction factor (PSO-cf) [13];
- Local version of PSO with inertia weight (PSO-w-local);
- Local version of PSO with constriction factor (PSO-cf-local) [15];
- UPSO [18];
- Fully informed particle swarm (FIPS) [19];
- FDR-PSO [20];
- CPSO-H [29];
- CLPSO.

Among these PSO local versions, PSO_w_local and PSO_cf_local were chosen as these versions yielded the best results [15] with von Neumann neighborhoods where neighbors above, below, and one each side on a two-dimensional lattice were connected. FIPS with U-ring topology that achieved the highest success rate [19] is used. When solving the 10-D problems, the population size is set at ten and the maximum fitness evaluations (FEs) is set at 30 000. When solving the 30-dimensional (30-D) problems, the population size is set at 40 and the maximum FE is set at 200 000. All experiments were run 30 times. The mean values and standard deviation of the results are presented.

[1]More composition functions can be found at http://www.ntu.edu.sg/home/EPNSugan/.

TABLE II
GLOBAL OPTIMUM, SEARCH RANGES AND INITIALIZATION RANGES OF THE TEST FUNCTIONS

f	x^*	$f(x^*)$	Search Range	Initialization Range
f_1	[0,0,…,0]	0	$[-100, 100]^D$	$[-100, 50\ ]^D$
f_2	[1,1,…,1]	0	$[-2.048, 2.048]^D$	$[-2.048, 2.048]^D$
f_3	[0,0,…,0]	0	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
f_4	[0,0,…,0]	0	$[-600, 600]^D$	$[600, 200]^D$
f_5	[0,0,…,0]	0	$[-0.5,\ 0.5]^D$	$[-0.5,\ 0.2]^D$
f_6	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_7	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_8	[420.96, 420.96,…420.96]	0	$[-500, 500]^D$	$[-500, 500]^D$
f_9	[0,0,…,0]	0	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
f_{10}	[0,0,…,0]	0	$[-600, 600]^D$	$[600, 200]^D$
f_{11}	[0,0,…,0]	0	$[-0.5,\ 0.5]^D$	$[-0.5,\ 0.2]^D$
f_{12}	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_{13}	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_{14}	[420.96, 420.96,…420.96]	0	$[-500, 500]^D$	$[-500, 500]^D$
f_{15}	Predefined rand number distributed in the search range	0	$[-500, 500]^D$	$[-5, 5]^D$
f_{16}	Predefined rand number distributed in the search range	0	$[-500, 500]^D$	$[-5, 5]^D$

When solving real-world problems, usually the fitness calculation accounts for the most time as the PSO is highly computation efficient. Hence, the algorithm-related computation times of these algorithms are not compared in this paper. Further, the main difference between the CLPSO and the original PSO is the modified velocity updating equation, which has been made simpler in the CPSO. The complexity of the new algorithm is similar to the original PSO. In the experiments, a serial implementation is used, while it is easy to be modified to a parallel implementation. With a parallel form, the performance is likely to be not affected much due to batch updating of $pbests$ while computational efficiency improves.

C. Experimental Results and Discussions

1) Results for the 10-D Problems: Table III presents the means and variances of the 30 runs of the nine algorithms on the sixteen test functions with $D = 10$. The best results among the nine algorithms are shown in bold. In order to determine whether the results obtained by CLPSO are statistically different from the results generated by other algorithms, the nonparametric Wilcoxon rank sum tests are conducted between the CLPSO's result and the best result achieved by the other eight PSO versions for each problem. The h values presented in the last row of Tables III and IV are the results of t-tests. An h value of one indicates that the performances of the two algorithms are statistically different with 95% certainty, whereas h value of zero implies that the performances are not statistically different. Fig. 9 presents the convergence characteristics in terms of the best fitness value of the median run of each algorithm for each test function.

From the results, we observe that for the Group A unimodal problems, since CLPSO has a large potential search space, it could not converge as fast as the original PSO. CLPSO achieved better results on all three multimodal groups than the original PSO. CLPSO surpasses all other algorithms on functions 4, 5, 7, 8, 10, 12, 13, 14, 15, and 16, and especially significantly improves the results on functions 7 and 8. According to the results of t-tests, these results are different from the second best results. The CLPSO achieved the same best result as the CPSO-H on function 6, and they both are much better than the other variants on this problem. The FIPS also performs well on multimodal problems. The FIPS performed better than the CLPSO on functions 3, 9, and 11. However, the CLPSO performs better on more complex problems when the other algorithms miss the global optimum basin. The Schwefel's function is a good example, as it traps all other algorithms in local optima. The CLPSO successfully avoids falling into the deep local optimum which is far from the global optimum. On the two composition functions with randomly distributed local and global optima, CLPSO performs the best.

Comparing the results and the convergence graphs, among these nine PSO algorithms, FDR-PSO has good local search ability and converges fast. PSO with inertia weight (PSO-w) and PSO with constriction factor (PSO-cf) are two global versions where the whole population is the neighborhood. PSO with constriction factor converges faster than the PSO with inertia weight. But PSO with inertia weight performs better on multimodal problems. UPSO combines global PSO and local PSO together to yield a balanced performance between the global and the local versions. PSO with inertia weight (PSO-w-local), PSO with constriction factor (PSO-cf-local)

TABLE III
RESULTS FOR 10-D PROBLEMS

Func / PSOs	Group A	Group A	Group B	Group B
	1	2	3	4
PSO-w	7.96e-051 ± 3.56e-050	3.08e+000 ± 7.69e-001	1.58e-014 ± 1.60e-014	9.69e-002 ± 5.01e-002
PSO-cf	9.84e-105 ± 4.21e-104	**6.98e-001 ± 1.46e+000**	9.18e-001 ± 1.01e+000	1.19e-001 ± 7.11e-002
PSO-w-local	2.13e-035 ± 6.17e-035	3.92e+000 ± 1.19e+000	6.04e-015 ± 1.67e-015	7.80e-002 ± 3.79e-002
PSO-cf-local	1.37e-079 ± 5.60e-079	8.60e-001 ± 1.56e+000	5.78e-002 ± 2.58e-001	2.80e-002 ± 6.34e-002
UPSO	**9.84e-118 ± 3.56e-117**	1.40e+000 ± 1.88e+000	1.33e+000 ± 1.48e+000	1.04e-001 ± 7.10e-002
FDR	2.21e-090 ± 9.88e-090	8.67e-001 ± 1.63e+000	3.18e-014 ± 6.40e-014	9.24e-002 ± 5.61e-002
FIPS	3.15e-030 ± 4.56e-030	2.78e+000 ± 2.26e-001	**3.75e-015 ± 2.13e-014**	1.31e-001 ± 9.32e-002
CPSO-H	4.98e-045 ± 1.00e-044	1.53e+000 ± 1.70e+000	1.49e-014 ± 6.97e-015	4.07e-002 ± 2.80e-002
CLPSO	5.15e-029 ± 2.16e-028	2.46e+000 ± 1.70e+000	4.32e-014 ± 2.55e-014	**4.56e-003 ± 4.81e-003**
h	1	1	1	**1**
Func / PSOs	Group B	Group B	Group B	Group B
	5	6	7	8
PSO-w	2.28e-003 ± 7.04e-003	5.82e+000 ± 2.96e+000	4.05e+000 ± 2.58e+000	3.20e+002 ± 1.85e+002
PSO-cf	6.69e-001 ± 7.17e-001	1.25e+001 ± 5.17e+000	1.20+001 ± 4.99e+000	9.87e+002 ± 2.76e+002
PSO-w-local	1.41e-006 ± 6.31e-006	3.88e+000 ± 2.30e+000	4.77e+000 ± 2.84e+000	3.26e+002 ± 1.32e+002
PSO-cf-local	7.85e-002 ± 5.16e-002	9.05e+000 ± 3.48e+000	5.95e+000 ± 2.60e+000	8.78e+002 ± 2.93e+002
UPSO	1.14e+000 ± 1.17e+000	1.17e+001 ± 6.11e+000	5.85e+000 ± 3.15e+000	1.08e+003 ± 2.68e+002
FDR	3.01e-003 ± 7.20e-003	7.51e+000 ± 3.05e+000	3.35e+000 ± 2.01e+000	8.51e+002 ± 2.76e+002
FIPS	2.02e-003 ± 6.40e-003	2.12e+000 ± 1.33e+000	4.35e+000 ± 2.80e+000	7.10e+001 ± 1.50e+002
CPSO-H	1.07e-015 ± 1.67e-015	**0 ± 0**	2.00e-001 ± 4.10e-001	2.13e+002 ± 1.41e+002
CLPSO	**0 ± 0**	**0 ± 0**	**0 ± 0**	**0 ± 0**
h	**1**	0	**1**	**1**
Func / PSOs	Group C	Group C	Group C	Group C
	9	10	11	12
PSO-w	2.80e-001 ± 5.86e-001	1.64e-001 ± 9.40e-002	6.66e-001 ± 7.12e-001	9.90e+000 ± 3.76e+000
PSO-cf	1.19e+000 ± 1.13e+000	1.38e-001 ± 1.07e-001	2.17e+000 ± 1.30e+000	1.44e+001 ± 6.04e+000
PSO-w-local	6.39e-015 ± 3.18e-015	8.04e-002 ± 4.46e-002	2.14e-001 ± 3.65e-001	9.25e+000 ± 2.74e+000
PSO-cf-local	2.56e-001 ± 5.33e-001	7.90e-002 ± 5.55e-002	1.20e+000 ± 1.22e+000	1.35e+001 ± 6.81e+000
UPSO	1.00e+000 ± 9.27e-001	7.76e-002 ± 6.40e-002	2.61e+000 ± 9.48e-001	1.52e+001 ± 5.25e+000
FDR	1.40e-001 ± 4.38e-001	1.44e-001 ± 7.84e-002	3.34e-001 ± 3.90e-001	9.25e+000 ± 2.50e+000
FIPS	**2.25e-015 ± 1.54e-015**	1.70e-001 ± 1.26e-001	**5.93e-014 ± 1.86e-013**	1.20e+001 ± 6.22e+000
CPSO-H	1.36e+000 ± 8.85e-001	1.20e-001 ± 8.07e-002	4.35e+000 ± 1.35e+000	2.67e+001 ± 1.06e+001
CLPSO	3.56e-005 ± 1.57e-004	**4.50e-002 ± 3.08e-002**	3.72e-010 ± 4.40e-010	**5.97e+000 ± 2.88e+000**
h	1	**1**	1	**1**
Func / PSOs	Group C	Group C	Group D	Group D
	13	14	15	16
PSO-w	1.02e+001 ± 3.58e+000	5.69e+002 ± 2.16e+002	1.20e+002 ± 8.94e+001	1.38e+002 ± 1.80e+002
PSO-cf	1.53e+001 ± 6.38e+000	1.19e+003 ± 4.23e+002	1.60e+002 ± 1.64e+002	2.31e+002 ± 1.93e+002
PSO-w-local	1.09e+001 ± 4.08e+000	4.72e+002 ± 3.07e+002	4.00e+001 ± 5.98e+001	1.53e+002 ± 1.53e+002
PSO-cf-local	1.07e+001 ± 2.81e+000	9.09e+002 ± 3.25e+002	9.00e+001 ± 8.52e+001	1.34e+002 ± 1.71e+002
UPSO	1.47e+001 ± 6.53e+000	1.27e+003 ± 2.29e+002	8.00e+001 ± 8.34e+001	1.79e+002 ± 1.56e+002
FDR	1.07e+001 ± 3.86e+000	1.07e+003 ± 2.23e+002	1.00e+002 ± 9.73e+001	1.53e+002 ± 2.01e+002
FIPS	8.84e+000 ± 3.27e+000	2.89e+002 ± 2.00e+002	6.00e+001 ± 5.16e+001	4.21e+001 ± 6.37e+001
CPSO-H	1.90e+001 ± 9.05e+000	9.67e+002 ± 3.67e+002	1.65e+002 ± 1.42e+002	2.46e+002 ± 2.18e+002
CLPSO	**5.44e+000 ± 1.39e+000**	**1.14e+002 ± 1.28e+002**	**1.64e+001 ± 3.63e+001**	**1.98e+001 ± 2.93e+001**
h	**1**	**1**	**1**	**1**

and FIPS with a U-ring topology are all local versions. They all perform better on multimodal problems than the global versions. Among the three, FIPS yields a comparatively better performance. CPSO-H presents good performance on some unrotated multimodal problems and converges faster when compared to CLPSO. However, its performance is seriously affected after rotation. Although CLPSO's performance is also affected by the rotation, it still performs the best on four rotated problems. It can be observed that all PSO variants failed on the rotated Schwefel's function, as it becomes much harder to solve after applying rotation.

2) Results for the 30-D Problems: The experiments conducted on 10-D problems are repeated on the 30-D problems and the results presented in Table IV. As the convergence graphs are similar to the 10-D problems, they are not presented. From the results in Table IV, we can observe that the algorithms

TABLE IV
RESULTS FOR 30-D PROBLEMS

Func / PSOs	*Group A*	*Group A*	*Group B*	*Group B*
	1	*2*	*3*	*4*
PSO-w	9.78e-030 ± 2.50e-029	2.93e+001 ± 2.51e+001	3.94e-014 ± 1.12e+000	8.13e-003 ± 7.16e-003
PSO-cf	5.88e-100 ± 5.40e-100	1.11e+001 ± 1.81e+000	1.12e+000 ± 8.65e-001	2.06e-002 ± 1.90e-002
PSO-w-local	5.35e-100 ± 4.41e-013	2.39e+001 ± 3.07e+000	9.10e-008 ± 8.11e-008	5.91e-003 ± 6.69e-003
PSO-cf-local	7.70e-054 ± 1.59e-053	1.71e+001 ± 9.16e-001	5.33e-015 ± 1.87e-015	5.91e-003 ± 8.70e-003
UPSO	4.17e-087 ± 3.15e-087	1.51e+001 ± 8.14e-001	1.22e-015 ± 3.16e-015	1.66e-003 ± 3.07e-003
FDR	4.88e-102 ± 1.53e-101	**5.39e+000 ± 1.76e+000**	2.84e-014 ± 4.10e-015	1.01e-002 ± 1.23e-002
FIPS	2.69e-012 ± 6.84e-013	2.45e+001 ± 2.19e-001	4.81e-007 ± 9.17e-008	1.16e-006 ± 1.87e-006
CPSO-H	**1.16e-113 ± 2.92e-113**	7.08e+000 ± 8.01e+000	4.93e-014 ± 1.10e-014	3.63e-002 ± 3.60e-002
CLPSO	4.46e-014 ± 1.73e-014	2.10e+001 ± 2.98e+000	**0 ± 0**	**3.14e-010 ± 4.64e-010**
h	1	1	**1**	**1**
Func / PSOs	*Group B*	*Group B*	*Group B*	*Group B*
	5	*6*	*7*	*8*
PSO-w	1.30e-004 ± 3.30e-004	2.90e+001 ± 7.70e+000	2.97e+001 ± 1.39e+001	1.10e+003 ± 2.56e+002
PSO-cf	4.10e+000 ± 2.20e+000	5.62e+001 ± 9.76e+000	2.85e+001 ± 1.14e+001	3.78e+003 ± 6.02e+002
PSO-w-local	4.94e-003 ± 1.40e-002	2.72e+001 ± 7.58e+000	2.08e+001 ± 4.94e+000	1.53e+003 ± 3.00e+002
PSO-cf-local	1.16e-001 ± 2.79e-001	4.53e+001 ± 1.17e+001	1.54e+001 ± 1.67e+001	3.78e+003 ± 5.37e+002
UPSO	9.60e+000 ± 3.78e+000	6.59e+001 ± 1.22e+001	6.34e+001 ± 1.24e+001	4.84e+003 ± 4.76e+002
FDR	7.49e-003 ± 1.14e-002	2.84e+001 ± 8.71e+000	1.44e+001 ± 6.28e+000	3.61e+003 ± 3.06e+002
FIPS	1.54e-001 ± 1.48e-001	7.30e+001 ± 1.24e+001	6.08e+001 ± 8.35e+000	2.05e+003 ± 9.58e+002
CPSO-H	**7.82e-015 ± 8.50e-015**	**0 ± 0**	1.00e-001 ± 3.16e-001	1.08e+003 ± 2.59e+002
CLPSO	3.45e-007 ± 1.94e-007	4.85e-010 ± 3.63e-010	**4.36e-010 ± 2.44e-010**	**1.27e-012 ± 8.79e-013**
h	1	1	**1**	**1**
Func / PSOs	*Group C*	*Group C*	*Group C*	*Group C*
	9	*10*	*11*	*12*
PSO-w	1.71e+000 ± 4.38e-001	1.77e-002 ± 1.53e-002	7.00e+000 ± 1.98e+000	6.87e+001 ± 2.05e+001
PSO-cf	1.66e+000 ± 1.10e+000	8.62e-003 ± 8.86e-003	8.48e+000 ± 2.54e+000	7.13e+001 ± 1.66e+001
PSO-w-local	5.70e-001 ± 7.60e-001	1.35e-002 ± 1.12e-002	5.96e+000 ± 2.09e+000	4.10e+001 ± 7.93e+000
PSO-cf-local	1.78e-001 ± 5.62e-001	1.30e-002 ± 1.06e-002	5.95e+000 ± 2.95e+000	4.66e+001 ± 1.05e+001
UPSO	2.94e-001 ± 6.71e-001	1.48e-003 ± 3.12e-003	1.85e+001 ± 3.37e+000	7.07e+001 ± 1.70e+001
FDR	3.59e-001 ± 5.93e-001	9.60e-003 ± 1.24e-002	**2.50e+000 ± 1.46e+000**	4.44e+001 ± 1.37e+001
FIPS	**5.23e-007 ± 1.42e-007**	6.92e-004 ± 2.18e-003	9.52e-002 ± 9.53e-002	7.41e+001 ± 2.79e+001
CPSO-H	2.10e+000 ± 3.84e-001	5.54e-002 ± 3.97e-002	1.43e+001 ± 3.53e+000	1.01e+002 ± 2.21e+001
CLPSO	3.43e-004 ± 1.91e-004	**7.04e-010 ± 1.25e-011**	3.07e+000 ± 1.61e+000	**3.46e+001 ± 4.59e+000**
h	1	**1**	0	0
Func / PSOs	*Group C*	*Group C*	*Group D*	*Group D*
	13	*14*	*15*	*16*
PSO-w	6.32e+ 001± 1.79e+001	2.67e+003 ± 7.03e+002	1.00e+002 ± 1.33e+002	2.20e+001 ± 3.34e+001
PSO-cf	7.88e+ 001± 1.88e+001	3.57e+003 ± 9.08e+002	7.00e+001 ± 1.33e+002	1.03e+002 ± 1.86e+002
PSO-w-local	5.67e+ 001± 1.36e+001	2.60e+003 ± 5.11e+002	2.00e+001 ± 3.16e+001	1.08e+001 ± 5.79e+000
PSO-cf-local	4.93e+ 001± 1.11e+001	3.89e+003 ± 9.42e+002	2.00e+001 ± 6.32e+001	6.06e+001 ± 1.24e+002
UPSO	7.74e+ 001± 1.40e+001	5.60e+003 ± 6.50e+002	1.00e+001 ± 2.51e+001	3.23e+001 ± 3.65e+001
FDR	4.36e+ 001± 8.96e+000	3.78e+003 ± 7.59e+002	1.00e+001 ± 3.16e+001	1.00e+001 ± 7.52e+000
FIPS	7.58e+ 001± 1.92e+001	2.60e+003 ± 8.49e+002	1.31e-002 ± 2.92e-002	1.04e+001 ± 4.63e+000
CPSO-H	8.80e+ 001± 2.59e+001	3.64e+003 ± 7.41e+002	1.30e+002 ± 1.64e+002	7.83e+001 ± 1.60e+002
CLPSO	**3.77e+001 ± 5.56e+000**	**1.70e+003 ± 1.86e+002**	**7.50e-005 ± 1.85e-004**	**7.86e+000 ± 3.64e+000**
h	**1**	**1**	**1**	**1**

achieved similar ranking as in the 10-D problems. CLPSO surpasses all other algorithms on functions 3, 4, 7, 8, 10, 12, 13, 14, 15, and 16, and especially significantly improves the results on functions 7 and 8. All 30-D functions become more difficult than their 10-D counterparts, and the results are not as good as in 10-D cases, although we increased the maximum number of FEs from 30 000 to 200 000. Better results were achieved on Griewank's function, since this problem is known to become easier as the number of dimensions increases [36]. The results of composition functions are not affected much since we use the same number of subfunctions with the same fixed local optima values [38].

3) Discussion: By analyzing the results of the CLPSO on 10-D and 30-D problems, one may conclude that the CLPSO does not perform the best for unimodal and simple multimodal problems in Group A. According to the "no free lunch" theorem [39], "any elevated performance over one class of problems is offset by performance over another

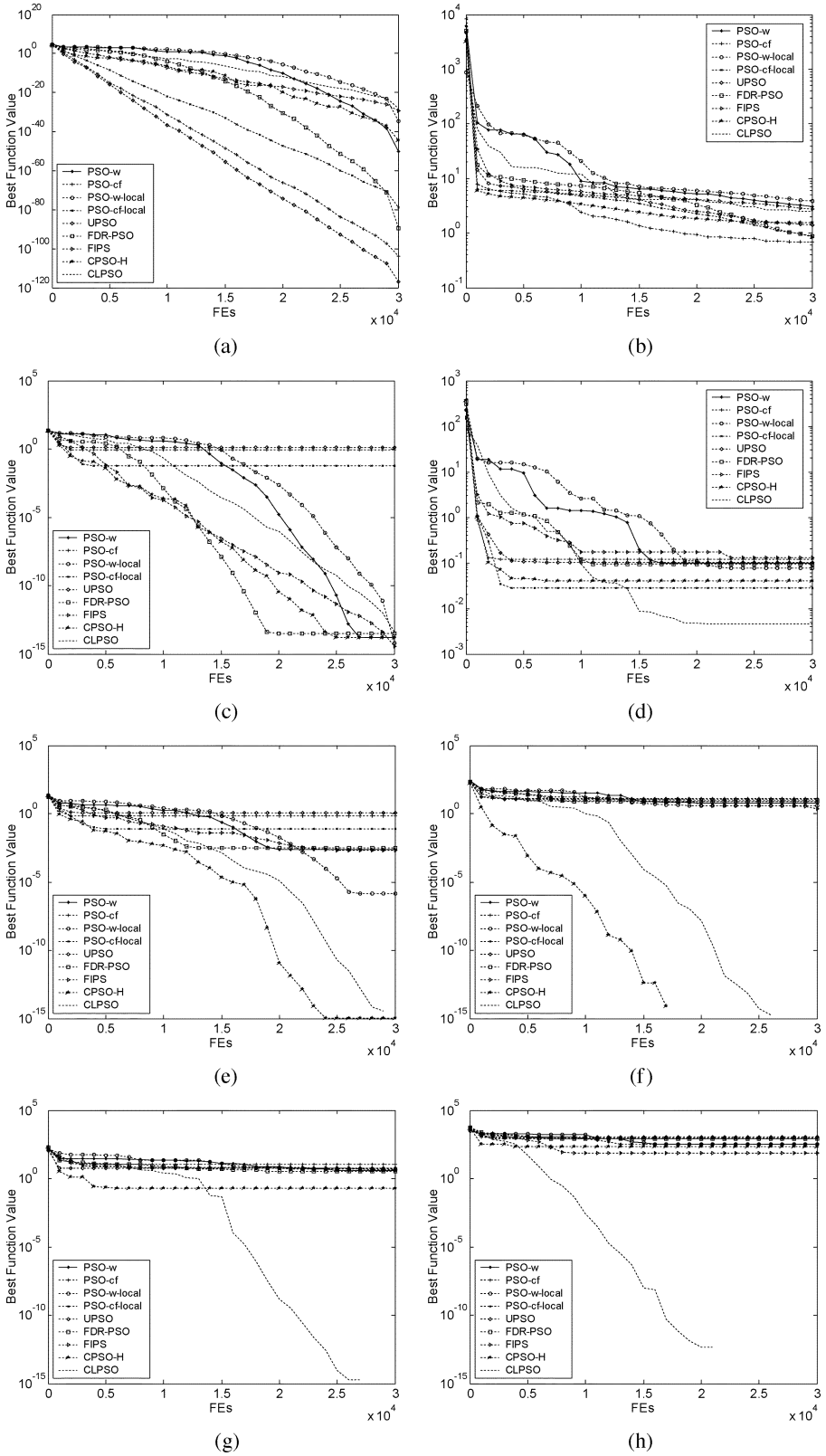


Fig. 9. The median convergence characteristics of 10-D test functions. (a) Sphere function. (b) Rosenbrock's function. (c) Ackley's function. (d) Griewank's function. (e) Weierstrass function. (f) Rastrigin's function. (g) Noncontinuous Rastrigin's function. (h) Schwefel's function.

class." There is a cost for tuning the CLPSO to obtain better performance on multimodal problems, and the cost is the slow convergence on unimodal problems. Therefore, we may not expect the best performance on all classes of problems, as the proposed CLPSO focuses on improving the PSO's performance on multimodal problems.

The CLPSO achieves the best results on most complex multimodal problems in Groups B to D, especially on Group B

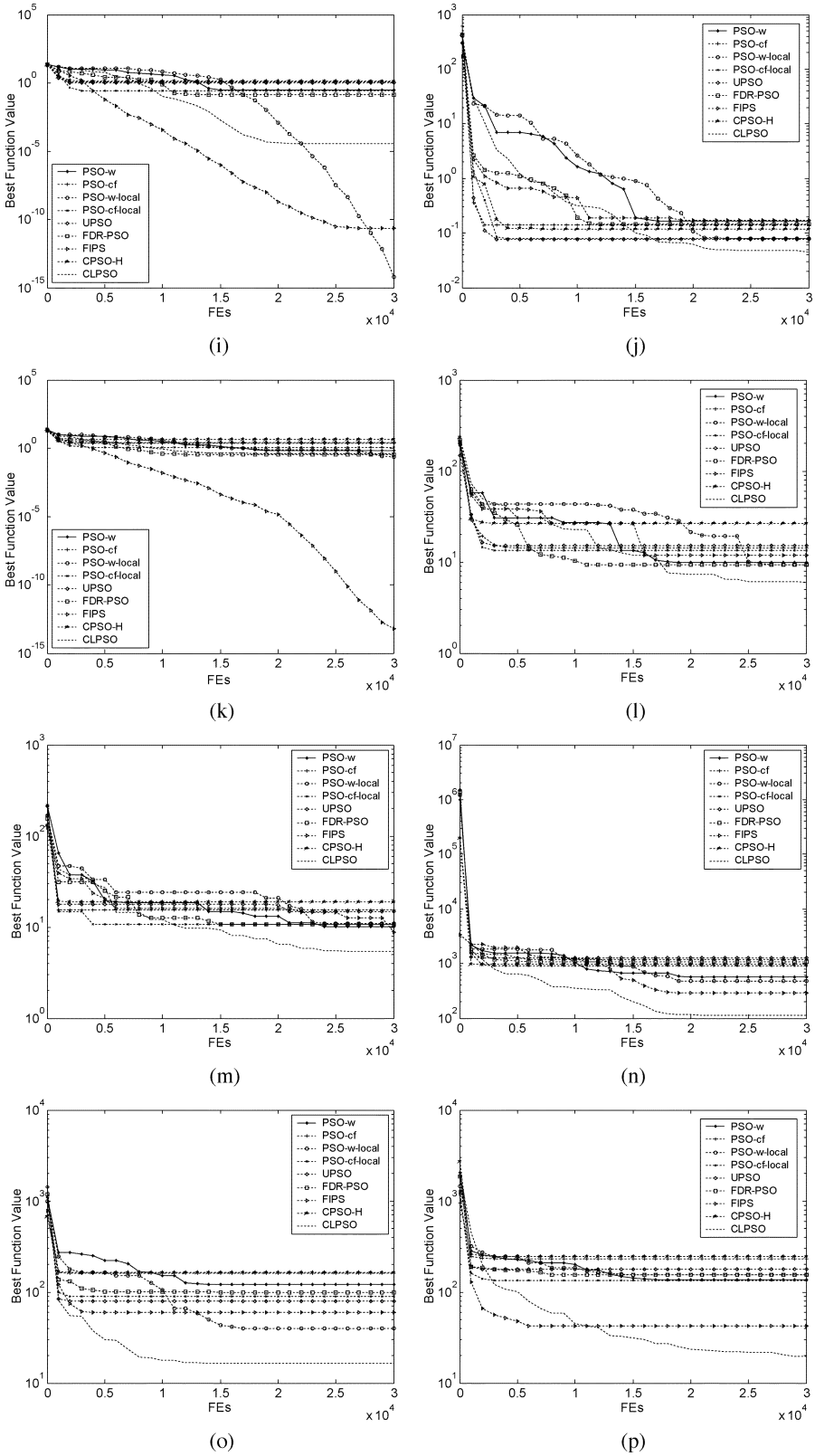


Fig. 9. (*Continued.*) The median convergence characteristics of 10-D test functions. (j) Rotated Griewanks's function. (k) Rotated Weierstrass function. (l) Rotated Rastrigin's function. (m) Rotated noncontinuous Rastrigin's function. (n) Rotated Schwefel's function. (o) Composition function 1 (CF1). (p) Composition function 5 (CF5).

unrotated multimodal problems. This implies that the CLPSO is more effective in solving problems with less linkage. This property is due to the PSO's dimension-wise updating rule, as well as CLPSO's learning of different dimensions from different exemplars. On more complex asymmetrical landscapes in Group D, CLPSO performs better when comparing with the other algorithms.

With the new updating rule, different dimensions may learn from different exemplars. Due to this, the CLPSO explores a larger search space than the original PSO. The larger search

space is not achieved randomly. Instead, it is based on the historical search experience. Because of this, the CLPSO performs comparably to or better than many PSO variants on most of the multimodal problems experimented in this paper.

V. Conclusion

This paper presents a comprehensive learning PSO employing a novel learning strategy where other particles' previous best positions are exemplars to be learned from by any particle and each dimension of a particle can potentially learn from a different exemplar. The new strategy makes the particles have more exemplars to learn from and a larger potential space to fly. From the analysis and experiments, we observe that this learning strategy enables the CLPSO to make use of the information in swarm more effectively to generate better quality solutions frequently when compared to eight PSO variants. Based on the results of the nine algorithms on the 16 chosen test problems belonging to four classes, we can conclude that CLPSO significantly improves the PSO's performance and gives the best performance on most multimodal problems irrespective of whether they are unrotated or rotated when compared with eight other PSO versions.

Although the CLPSO is not the best choice for solving unimodal problems, when solving real-world problems, we do not frequently know the shape of the fitness landscape. Hence, it is advisable to use an algorithm that performs well on multimodal problems since such an algorithm can also solve unimodal problems. By combining the CLPSO with a local search method such as the quasi-Newton method, unimodal problems may be solved more efficiently.

Another attractive property of the CLPSO is that it does not introduce any complex operations to the original simple PSO framework. The only difference from the original PSO is the velocity update equation. The CLPSO is also simple and easy to implement like the original PSO.

Acknowledgment

The authors thank the anonymous reviewers for providing valuable comments to improve this paper.

References

[1] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39–43.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942–1948.

[3] D. N. Wilke, "Analysis of the particle swarm optimization algorithm," master's thesis, Dept. Mechanical and Aeronautical Eng., Univ. of Pretoria, Pretoria, South Africa, 2005.

[4] J. F. Schutte and A. A. Groenwold, "Sizing design of truss structures using particle swarms," *Struct. Multidisc. Optim.*, vol. 25, no. 4, pp. 261–269, 2003.

[5] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga MS, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[6] L. Messerschmidt and A. P. Engelbrecht, "Learning to play games using a PSO-based competitive learning approach," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 280–288, Jun. 2004.

[7] M. P. Wachowiak, R. Smolikova, Y. F. Zheng, J. M. Zurada, and A. S. Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 289–301, Jun. 2004.

[8] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Congr. Evol. Comput.*, 1998, pp. 69–73.

[9] ——, "Parameter selection in particle swarm optimization," in *Proc. 7th Conf. Evol. Programming*, New York, 1998, pp. 591–600.

[10] ——, "Particle swarm optimization with fuzzy adaptive inertia weight," in *Proc. Workshop Particle Swarm Optimization*, Indianapolis, IN, 2001, pp. 101–106.

[11] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time varying accelerating coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 240–255, Jun. 2004.

[12] H. Y. Fan and Y. Shi, "Study on Vmax of particle swarm optimization," in *Proc. Workshop Particle Swarm Optimization*, Indianapolis, IN, 2001.

[13] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[14] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proc. Congr. Evol. Comput.*, 1999, pp. 1931–1938.

[15] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1671–1676.

[16] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in *Proc. Congr. Evol. Comput.*, Washington, DC, 1999, pp. 1958–1962.

[17] X. Hu and R. C. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proc. Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1677–1681.

[18] K. E. Parsopoulos and M. N. Vrahatis, "UPSO—A unified particle swarm optimization scheme," in *Lecture Series on Computational Sciences*, 2004, pp. 868–873.

[19] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 204–210, Jun. 2004.

[20] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proc. Swarm Intelligence Symp.*, 2003, pp. 174–181.

[21] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Anchorage, AK, 1998, pp. 84–89.

[22] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimizer with breeding and subpopulations," in *Proc. Genetic Evol. Comput. Conf.*, 2001, pp. 469–476.

[23] V. Miranda and N. Fonseca, "New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control," in *Proc. 14th Power Syst. Comput. Conf.*, Seville, Spain, 2002. [Online]. Available: http://www.pscc02.org/papers/s21pos.pdf.

[24] M. Lovbjerg and T. Krink, "Extending particle swarm optimizers with self-organized criticality," in *Proc. Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1588–1593.

[25] T. M. Blackwell and P. J. Bentley, "Don't push me! Collision-avoiding swarms," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1691–1696.

[26] T. Krink, J. S. Vesterstroem, and J. Riget, "Particle swarm optimization with spatial particle extension," in *Proc. Congr. Evolut. Comput.*, Honolulu, HI, 2002, pp. 1474–1479.

[27] X. Xie, W. Zhang, and Z. Yang, "A dissipative particle swarm optimization," in *Proc. Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1456–1461.

[28] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 211–224, Jun. 2004.

[29] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 225–239, Jun. 2004.

[30] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Particle swarm optimization algorithms with novel learning strategies," in *Proc. Int. Conf. Systems, Man, Cybernetics*, The Netherlands, Oct. 2004. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan.

[31] ——, "Evaluation of comprehensive learning particle swarm optimizer," in *Proc. 11th Int. Conf. Neural Information Processing*, Science City, Calcutta, India, Nov. 2004. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan.

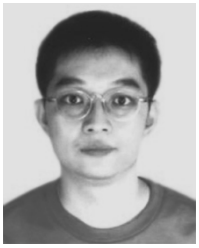
[32] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[33] C.-Y. Lee and X. Yao, "Evolutionary programming using mutations based on the levy probability distribution," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 1–13, Feb. 2004.

[34] Z. G. Tu and L. Yong, "A robust stochastic genetic algorithm (StGA) for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 456–470, 2004.
[35] S. C. Esquivel and C. A. Coello Coello, "On the use of particle swarm optimization with multimodal functions," in *Proc. 2003 Congr. Evol. Comput.*, vol. 2, Canberra, Australia, Dec. 2003, pp. 1130–1136.
[36] D. Whitley, D. Rana, J. Dzubera, and E. Mathias, "Evaluating evolutionary algorithms," *Artif. Intell.*, vol. 85, pp. 245–276.
[37] R. Salomon, "Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions," *BioSystems*, vol. 39, pp. 263–278, 1996.
[38] J. J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Proc. Swarm Intell. Symp.*, Jun. 2005. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan.
[39] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 67–82, Apr. 1997.

J. J. Liang received the B. Eng. degree from Harbin Institute of Technology, Harbin, China. She is currently working towards the Ph.D. degree at Nanyang Technological University, Singapore.

Her main research interests are evolutionary computation, swarm intelligence, and multiobjective optimization.



A. K. Qin (S'04) received the B.E. degree from the Department of Automatic Control Engineering, Southeast University, Nanjing, China, in 2001. Since 2003, he has been working towards the Ph.D. degree at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

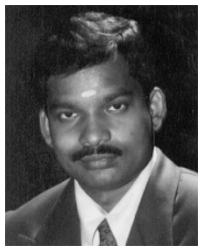
His research interests include pattern recognition, machine learning, neural network, genetic and evolutionary algorithms, computer vision, and bioinformatics.

Mr. Qin is a member of the Pattern Recognition and Machine Intelligence Association, Singapore, and a student member of AAAI.

Ponnuthurai Nagaratnam Suganthan (S'91–M'92–SM'00) received the B.A. degree, postgraduate certificate, and the M.A. degree in electrical and information engineering from the University of Cambridge, U.K., in 1990, 1992, and 1994, respectively, and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

He was a predoctoral Research Assistant in the Department of Electrical Engineering, University of Sydney, Australia, in 1995–1996 and a Lecturer in the Department of Computer Science and Electrical Engineering, University of Queensland, Australia, in 1996–1999. Between 1999 and 2003, he was an Assistant Professor in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is now an Associate Professor. He is an Associate Editor of *Pattern Recognition Journal*. His research interests include evolutionary computation, applications of evolutionary computation, neural networks, pattern recognition, and bioinformatics.

Prof. Suganthan is an Associate Member of the Institution of Electrical Engineers. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.



S. Baskar received the B.E. and the Ph.D. degrees in evolutionary optimization from Madurai Kamaraj University, Madurai, India, in 1991 and 2001, respectively, and the M.E. degree from Anna University, India, in 1993.

He has been an Assistant Professor in the Department of Electrical and Electronics Engineering, Thiagarajar College of Engineering, Madurai, since 1998. His research interests include the development of novel evolutionary algorithms, applications to various engineering optimization problems, and evolutionary multiobjective optimization.

Prof. Baskar is a member of Institution of Engineers (India) and a Life Member of the Indian Society for Technical Education. He received the Young Scientists BOYSACAST Fellowship during 2003–2004 supported by the Department of Science and Technology, Government of India, to pursue postdoctoral research at Nanyang Technological University, Singapore.