Hindawi
Complexity
Volume 2023, Article ID 7111548, 19 pages
https://doi.org/10.1155/2023/7111548

Research Article

PSO with Mixed Strategy for Global Optimization

Jinwei Pang,[1] **Xiaohui Li**,[2] **and Shuang Han**[3]

[1]*School of Computer and Control Engineering, Yantai University, Yantai, China*
[2]*Harbin Vocational and Technical College, Harbin, China*
[3]*Department of Computer Science and Technology, Harbin Engineering University, Harbin, China*

Correspondence should be addressed to Jinwei Pang; pangjinwei789@qq.com

Received 9 December 2021; Revised 31 December 2022; Accepted 19 June 2023; Published 6 September 2023

Academic Editor: Alejandro F. Villaverde

Particle swarm optimization (PSO) is an evolutionary algorithm for solving global optimization problems. PSO has a fast convergence speed and does not require the optimization function to be differentiable and continuous. In recent two decades, a lot of researches have been working on improving the performance of PSO, and numerous PSO variants have been presented. According to a recent theory, no optimization algorithm can perform better than any other algorithm on all types of optimization problems. Thus, PSO with mixed strategies might be more efficient than pure strategy algorithms. A mixed strategy PSO algorithm (MSPSO) which integrates five different PSO variants was proposed. In MSPSO, an adaptive selection strategy is used to adjust the probability of selecting different variants according to the rate of the fitness value change between offspring generated by each variant and the personal best position of particles to guide the selection probabilities of variants. The rate of the fitness value change is a more effective indicator of good strategies than the number of previous successes and failures of each variant. In order to improve the exploitation ability of MSPSO, a Nelder–Mead variant method is proposed. The combination of these two methods further improves the performance of MSPSO. The proposed algorithm is tested on CEC 2014 benchmark suites with 10 and 30 variables and CEC 2010 with 1000 variables and is also conducted to solve the hydrothermal scheduling problem. Experimental results demonstrate that the solution accuracy of the proposed algorithm is overall better than that of comparative algorithms.

1. Introduction

In recent years, optimization algorithms are applied more and more widely in various fields [1]. One of the most famous ones is PSO. In 1995, Kennedy and Eberhart developed particle swarm optimization (PSO) [2]. PSO as a global optimization method is an important tool to solve difficult optimization problems without a good problem-specific approach efficiently. Its original inspiration comes from birds flocking behaviours. In PSO, each individual in the population is a particle. A particle represents a potential solution in solution space. Particles scan the search area and converge to the optimum by flying in the space and adjusting its flying velocity based on its personal best historical experience and the best solution in the population. PSO is a robust stochastic optimization algorithm that is easy to implement. Its parameter settings are negligible. On account of its simple realization and high efficiency, PSO has been successfully applied to various real-world problems such as wireless sensor networks [3], feature selection [4], traffic control [5], road identification [6], task allocation [7], and crowd user selection [8].

Recently, various improvements of PSO are proposed to enhance these comprehensive performances. In this research, according to a recent theory [9], this study presented a mixed strategy PSO (MSPSO). In the theory, the hardest problem to one evolutionary algorithm might be the easiest for another algorithm and vice versa. Thus, the mixed strategy PSO algorithms might be more efficient than pure strategy PSO algorithms. Just as a company wants to run well, it needs talents who are good at management, good at marketing, and good at purchasing to work together. If a company employs talents who are good at management for all work, the company will not operate well because employees who are not good at what they do spend more time and get worse results. Inspired by this theory, MSPSO integrates five different PSO variants and adopts a new probability update strategy according to the proportion of differences in fitness values. According to our experimental verification, the probability of those variants is guided by the rate of change of fitness value between offspring generated by each variant and the personal best positions of the particles. According to the rank of the rate of change, the variants are assigned the different probabilities. Using the rate of fitness change to guide the selection probabilities of variants could increase the probability of selecting excellent variants than using the number of previous successes and failures of each variant. In addition, in order to enhance the exploitation ability of MSPSO, a local search method inspired by the Nelder–Mead method is proposed.

In summary, we have made the following contributions:

(1) We propose a cooperative strategy to integrate multiple PSO operators, and the integrated algorithm can achieve better generalization capability

(2) We add a local search operator to the integrated algorithm, so that the algorithm can further obtain better performance

(3) We also demonstrate the performance of MSPSO on benchmark suites and real-world problem instances

The rest of the study is organized as follows. First, related methods are reviewed in Section 2. Second, Section 3 introduces the MSPSO. Third, Section 4 evaluates the proposed MSPSO and gives the results of the experiments. Finally, the conclusion of the study is shown in Section 5.

2. Related Work

2.1. Canonical PSO. In the optimization process, the velocity vector V_i for the ith particle in the population is updated using (1) given in [2] iteratively through the guidance of $p\text{best}_i$ and $g\text{best}$.

$$v_{id} = v_{id} + c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right) + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right). \tag{1}$$

Acceleration parameters c_1 and c_2 are usually set to 2.0. r_{1id} and r_{2id} are two random numbers within [0,1] for the dth dimension of the ith particle.

To avoid the premature convergence, the authors in [10] introduced an inertia weight ω to update the flying velocities of particles. The particle velocity is adjusted through the following formula:

$$v_{id} = \omega * v_{id} + c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right) + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right). \tag{2}$$

In (2), ω commonly decreases linearly from 0.9 to 0.4 with generations to balance exploration capability and exploitation capability. A large value of ω enhances the exploration capability, whereas a small value of ω encourages the capability of convergence during the search process.

2.2. PSO Variants. To enhance the performance of PSO on global optimization problems, a lot of researches have been working on improving PSO algorithms, and numerous PSO variants have been presented. Designing new strategies, new techniques and topological structures of PSO are an important research trend. Various topologies have been suggested. In PSO, the trajectory of particles is adjusted by their own personal best positions and the best position in the population. However, this may cause premature convergence when solving multimodal functions. Because the best particle in the population is the best solution for the whole population, it could be a local optimum for a multimodal function and is far away from the global optimum. The authors in [11] proposed a social learning PSO (SL-PSO) which introduced social learning into PSO. The advantage of social learning was that individuals could learn from others without paying for their own trials and mistakes. In SL-PSO, each particle was updated based on any better particles in the current population. Furthermore, to reduce parameter settings, SL-PSO proposed a dimension-dependent parameter control method. Compared with other optimization algorithms, SL-PSO could be implemented easily, be computed efficiently, and require no complicated adjustment of the control parameters. In order to accelerate convergence speed and improve exploitation ability, the authors in [12] proposed prey-predator PSO (PP-PSO). PP-PSO achieved this goal by deleting or transforming “slothful particles” which were the particles with low velocities. It was hard for these slothful particles to find the global optimum, and this reduced the convergence speed. Furthermore, in order to enhance population diversity, PP-PSO designed a proportional-integral control parameter to control the population to fluctuate within a relatively stable range during the iterative process. The above-mentioned PSO variant algorithm mainly improves the classical PSO algorithm from the perspective of designing a new information-sharing mode between particles and building a new particle search model.

However, when solving some complex optimization problems, PSO and its variants are still prone to premature convergence in the search process. Furthermore, when trapped in the local optimum, it is difficult for particles to get rid of this region. Therefore, in the past decades, researchers have also tried to solve this problem by proposing various improvement strategies based on existing PSO algorithms.

Another popular modification is to combine PSO with other mathematical methods or evolutionary computation techniques. The authors in [13] integrated a PSO algorithm with the sine cosine algorithm (SCA) and the Lévy flight approach, to overcome the shortcoming that PSO tended to fall into a local optimum. The solution in SCA was updated by sine and cosine functions to ensure the exploitation and exploration capabilities. In addition, SCA used Lévy distribution which was a more effective search to produce a random walk in the search space. The combination of SCA, Lévy flight, and PSO enhanced the exploration capability of the original PSO and prevented being trapped in the local minimum. In addition, the hybridization of PSO with GAs has also been presented in [14, 15]. A hybrid PSO with bat algorithm (BA) has been proposed in [16] for numerical optimization problems. A communicating strategy provided information flow between the population of PSO and the population of BA. In this work, several best individuals in BA replaced the worst individuals in PSO after fixed iterations, and on the contrary, the finest particles of PSO replaced the poorer individuals of BA.

Multipopulation strategy and ensemble optimizer are also effective methods to optimize the performance of PSO. In order to avoid the phenomena of "oscillation" and "two steps forward, one step back" in PSO, the authors in [17] proposed a two-swarm learning PSO algorithm called TSLPSO. The algorithm hybridized two different learning strategies which were dimensional learning strategy (DLS) and comprehensive learning strategy, respectively. One of the swarms used DLS to construct the learning exemplars. DLS used the information of the best particle in the population for the local search of the particles. However, in order to guide the global search, the other swarm used the comprehensive learning strategy to construct the learning exemplars. In [18], Xu et al. constructed a DMS-PSO-CLS algorithm that combined the dynamic multiswarm particle swarm optimizer (DMS-PSO) and a new cooperative learning strategy (CLS). In the CLS subpopulation, in order to learn more excellent examples, the two poor particles updated their dimensions with the better particle which was selected from two random subswarms using a tournament selection strategy. By using this method, particles could search the global optimum more easily. The simulation results showed that the performance of the DMS-PSO-CLS algorithm was superior compared with other comparison PSO variants. The above algorithms also have their limitations. For example, some hybrid algorithm frameworks require more computing resources to execute the iterative process of different algorithms, while most multipopulation strategies cannot perform fine local search at the later stage of the search process, so it is difficult to obtain the final search results with high accuracy.

2.3. Complementary Strategy Theorem.

The authors in [19] proposed a complementary strategy theorem. According to this theorem, mixed strategy evolutionary algorithms might outperform pure strategy evolutionary algorithms. One advantage was that the overall performance of mixed strategy evolutionary algorithms might be the same as the best performance of pure strategy evolutionary algorithms.

Theorem 1. *If a pure strategy evolutionary algorithm PS_2 is better than another pure strategy evolutionary algorithm PS_1, then for any initial population P, the expected hitting time of mixed strategy evolutionary algorithms MS derived from PS_1 and PS_2 satisfies that $m_{MS}(P) \geq m_{PS2}(P)$ and $m_{MS}(P) > m_{PS2}(P)$ for some state P.*

Theorem 2. *If a pure strategy evolutionary algorithm PS_2 is equivalent to another pure strategy evolutionary algorithm PS_1, then for any initial population P, the expected hitting time of mixed strategy evolutionary algorithm MS derived from PS_1 and PS_2 satisfies that $m_{MS}(P) = m_{PS2}(P)$.*

Theorem 3. *If a pure strategy evolutionary algorithm PS_2 complements with another pure strategy evolutionary algorithm PS_1, then there exists a mixed strategy evolutionary algorithm MS derived from PS_1 and PS_2, and its expected hitting time satisfies that $m_{MS}(P) \leq m_{PS2}(P)$ for any initial population P and $m_{MS}(P) < m_{PS2}(P)$ for some initial population P.*

Theorem 4 (complementary strategy theorem). *The condition that a pure strategy evolutionary algorithm PS_2 is complementary to another pure strategy evolutionary algorithm PS_1 is sufficient and necessary if there exists a mixed strategy evolutionary algorithm MS derived from them such that $m_{MS}(P) \leq m_{PS2}(P)$ for any initial population P and $m_{MS}(P) < m_{PS2}(P)$ for some initial population P.*

The complementary strategy theorem can be interpreted intuitively as follows:

(1) If one pure strategy evolutionary algorithm is better than another pure strategy evolutionary algorithm, then the design of a mixed strategy evolutionary algorithm with the same performance as the better pure strategy evolutionary algorithm is impossible. So mixed strategy evolutionary algorithms do not usually outperform pure strategy evolutionary algorithms that they derived from.

(2) If one pure strategy evolutionary algorithm is complementary to another, then the design of a mixed strategy evolutionary algorithm better than both pure strategy evolutionary algorithms is possible. However, this does not mean all mixed strategy evolutionary algorithms will outperform pure strategy evolutionary algorithms that they derived from.

(3) The following principle should be followed when a better-mixed strategy evolutionary algorithm is designed: if a pure strategy evolutionary algorithm has a better performance than another at a state, then

the mixed strategy evolutionary algorithm should apply the pure strategy with a higher probability at that state.

3. Mixed Strategy PSO

3.1. PSO Strategies. MSPSO hybridizes PSO [10], MCLPSO [20], LIPS [21], HPSO-TVAC [22], and FDR-PSO [23] with an adaptive selection strategy. Velocity update formulae of these four PSO variants except PSO are given as follows.

3.1.1. MCLPSO. We proposed a modified CLPSO (MCLPSO) in [20] a few years ago. Compared with CLPSO, MCLPSO could improve the convergence ability while maintaining the population diversity. Furthermore, MCLPSO has a better balance of exploration and exploitation than CLPSO. The updating equation of MCLPSO for a particle velocity is given as follows:

$$\begin{aligned}
&if\ \text{rand} < \alpha * \left(1 - \frac{g-1}{\text{Max \_Gen}}\right),\\
&\quad v_{id} = \omega * \left(v_{id} - \text{mean}v_d\right) + c * r_{id} * \left(p\text{best}_{fi(d)d} - x_{id}\right),\\
&\quad\text{else}\\
&\quad v_{id} = \omega * v_{id} + c * r_{id} * \left(p\text{best}_{fi(d)d} - x_{id}\right),
\end{aligned} \tag{3}$$

where $g = [1, 2, \ldots, \text{Max \_Gen}]$ is the current generation number, α is an adjustment coefficient between 0 and 1, Max _Gen is the maximum number of generations, rand is a random number within range [0, 1], meanv_d is the dth dimension of the average value of velocities in the whole population, and $p\text{best}_{fi(d)d}$ represents the dth dimension of the best position of the particle located in a list of particles selected randomly from the whole population, and the rest of the parameters have the same meanings as those in (2).

Using the above equation, the velocity of MCLPSO is fast with a high probability in the early stage of the search. Conversely, in the later stage, the velocity is slow with a high probability for better exploitation.

3.1.2. LIPS. LIPS used the best experiences of adjacent particles rather than the global best experience of the population to guide the particles to the optimum [21]. This algorithm adopted the personal best position of neighbor particles measured by Euclidean distance to adjust the particle velocity. The formula is given as follows:

$$\begin{aligned}
v_{id} &= 0.7298 * \left(v_{id} + \phi * \left(pn_{id} - x_{id}\right)\right),\\
pn_i &= \frac{\sum_{k=1}^{n\text{size}} \left(\phi_k * n\text{best}_k\right)/n\text{size}}{\sum_{k=1}^{n\text{size}} \phi_k},\\
\phi_k &\sim U\left(0, 4.1/n\text{size}\right),
\end{aligned} \tag{4}$$

where $n\text{best}_k$ is the best position of the kth neighbor particle of the ith particle, ϕ_k is a random number that obeys uniform distribution within [0, 4.1/$n$size], and $n$size is the number of neighbor particles.

3.1.3. HPSO-TVAC. In order to make the particles converge quickly to the global optimum, the authors in [22] designed a new formula for calculating velocity without using the previous velocity. The formula is given as follows:

$$v_{id} = c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right) + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right), \tag{5}$$

where c_1, r_{1id}, c_2, and r_{2id} have the same meanings as those in (1).

3.1.4. FDR-PSO. In order to avoid the premature convergence, the authors in [23] added the position of neighbor particle in the formula of particle velocity. The formula is given as follows:

$$\begin{aligned}
v_{id} = {} & \omega * v_{id} + c_1 * r_{1id} * \left(p\text{best}_{id} - x_{id}\right)\\
& + c_2 * r_{2id} * \left(g\text{best}_d - x_{id}\right) + c_3 * r_{3id} * \left(n\text{best}_{id} - x_{id}\right),
\end{aligned} \tag{6}$$

where $n\text{best}_{\text{id}}$ is the dth dimension of the best experience of the neighbor of the ith particle which minimizes the fineness-distance ratio (FDR), and the rest of the parameters have the same meanings as those in (2). The formula of fineness-distance ratio for a minimization problem is given as follows:

$$\text{FDR} = \frac{\text{Cost}\left(P_i\right) - \text{Cost}\left(X_i\right)}{\left|P_{id} - X_{id}\right|}, \tag{7}$$

where P_i denotes the best experience of other particles in the population except the ith particle.

3.2. Local Search. A local search method inspired from the Nelder–Mead method is used in MSPSO, in order to improve its exploitation ability. The Nelder–Mead method is a numerical algorithm that adapts to local landscapes [24]. It makes down-hill search using a simplex instead of derivatives. Introducing the Nelder–Mead method into MSPSO can further improve the performance of MSPSO.

In our work, we make a modification of the Nelder–Mead method, in order to reduce time consumption. The number of testing points is set to 3, rather than $n + 1$ (the dimension). The following is the detail of the method. Given 3 test points x_1,x_2, x_3, a Nelder–Mead variant is given in Algorithm 1. In Line 1, three individuals are sorted in the order of the function value from low to high. In Line 2, various x_{o} is the centre of triangle $\triangle x_1x_2x_3$. Lines 3–22 are used to implement the Nelder–Mead process.

3.3. Improved Adaptive Probability Adjustment Method. In some ensemble evolutionary algorithms, the selection probability of different variants is adjusted based on the number of previous successes and failures of each variant at a fixed iteration interval. However, the number of successes

```
Input: population $P$ with three individuals.
Sort the three points in the order: $f(x_1) \le f(x_2) \le f(x_3)$.
Calculate $x_o$ as follows:
        $x_o = 1/3\sum_{k=1}^{3} x_k$
[Reflection] Compute the reflected point $x_r = x_o + \alpha(x_o - x_3)$. Where $\alpha$ is a reflection coefficient. Its standard value is $\alpha = 1$.
if $f(x_1) \le f(x_r) < f(x_3)$, then
  $x_3' =$ the reflected point $x_r$.
else if $f(x_r) < f(x_1)$ then
  [Expansion] Compute the expanded point $x_e = x_o + \gamma(x_r - x_o)$. Where $\gamma$ is an expansion coefficient. Its standard value is $\gamma = 2$.
  if $f(x_e) < f(x_r)$ then
    $x_3' =$ the expanded point $x_e$
  else
    $x_3' =$ the reflected point $x_r$
  end if
else
  [Contraction] Compute the contracted point $x_c = x_o + \rho(x_3 - x_o)$. Where $\rho$ is a contraction coefficient. Its standard values is $\rho = 1/2$.
  if $f(x_c) < f(x_3)$ then
    $x_3' =$ the contracted point $x_c$
  else
    for $i = 2, 3$ do
      [Shrink] $x_i' = x_1 + \sigma(x_i - x_1)$. Where $\sigma$ is a shrink coefficient. Its standard value is $\sigma = 1/2$.
    end for
  end if
end if
Output: population $P = \{x_1', x_2', x_3'\}$.
```

Algorithm 1: Nelder–Mead variant.

and failures is not a perfect indicator of a good strategy because it cannot measure the degree of improvement of successful offspring generated by each variant. Thus, we use the rate of change of fitness value between offspring generated by each variant and the personal best position to adjust the selection probabilities of variants. The adaptive probability adjustment method is given as follows.

Step 5. Initialize the probability p_k of the kth PSO variant to $1/K$ and the change rate Cr_k of the kth variant as 0. In our work, K is equal to 5.

Step 6. Generate a random number rand_{pk}. If $0 \le \text{rand}_{pk} < p_1$, choose the first variant to generate an offspring. If $\sum_{i=1}^{k-1} p_i \le \text{rand}_{pk} < \sum_{i=1}^{k} p_i$, $1 < k < K$, choose the kth variant to generate an offspring. If $\sum_{i=1}^{k} p_i \le \text{rand}_{pk} \le 1$, choose the Kth variant to generate an offspring.

Step 7. If the kth variant is selected to generate an offspring for a particle, then the change rate is recorded as follows:

$$Cr_k = \frac{Cr_k - \left(f_o - f_p\right)}{f_p}, \tag{8}$$

where f_o and f_p are the fitness of the offspring and the fitness of the personal best historical position of the particle, respectively. When the fitness of the offspring is larger than the fitness of *pbest*, i.e., the offspring is worse than the parent, the change rate is reduced. Otherwise, the change rate is increased.

Step 8. After lp generations, update the probability p_k of each PSO variant, and set Cr_k to 0. The probability p_k is updated by the following steps:

(1) Sort the K strategies in descending order based on Cr_k. Get a new sequence K'.

(2) Assign probabilities to the strategies according to their ranking, $p_{K'} = [0.4, 0.3, 0.15, 0.12, 0.03]$, i.e., the probability of the strategy with the largest Cr is set to 0.4.

In this study, we use the change rate rather than the difference between the fitness of the offspring and the fitness of *pbest* to guide the adjustment of the probability. Because the difference cannot reflect the merits and demerits of each strategy especially when the fitness values of a strategy and *pbest* are large, however, the fitness values of another strategy and *pbest* are small. In this situation, the strategy with a small difference in fitness value may be better than the strategy with a large difference. Furthermore, we use a fixed probability distribution to assign a significantly larger selection probability to a good strategy and a significantly smaller selection probability to a bad strategy.

3.4. Framework of MSPSO. MSPSO integrates PSO, MCLPSO, LIPS, HPSO-TVAC, and FDR-PSO together. It adopts two subpopulations in the early stage and a whole population in the later stage of the search process. In the early stage, MSPSO adopts a subpopulation that implements MCLPSO and a subpopulation that implements ensemble

PSO. In the later stage, the whole population implements ensemble PSO. Furthermore, the Nelder–Mead method variant is used in this stage to improve the exploitation ability of MSPSO. In this way, the population diversity and convergence ability of the algorithm can be improved in the early stage. Its convergence ability can be improved in the later stage. The pseudo-code of MSPSO is given in Algorithm 2. In lines 1-2, population and parameters are initialized. Lines 3–10 give the steps of the early stage of MSPSO. In this stage, the whole population is divided into two subpopulations, one of which is composed of μ_1 individuals and the other is composed of $\mu - \mu_1$ individuals. Lines 5–7 indicate that MCLPSO is implemented in the first subpopulation, while lines 8–10 indicate that the ensemble PSO is implemented in the second subpopulation. Lines 11–16 give the steps of the late stage of MSPSO. In this stage, the whole population implements the ensemble PSO, and then, the Nelder–Mead method variant is implemented. In lines 17–22, the best fitness value of the current population is obtained, and the best fitness value of the algorithm is updated if necessary. The framework figure is given in Figure 1.

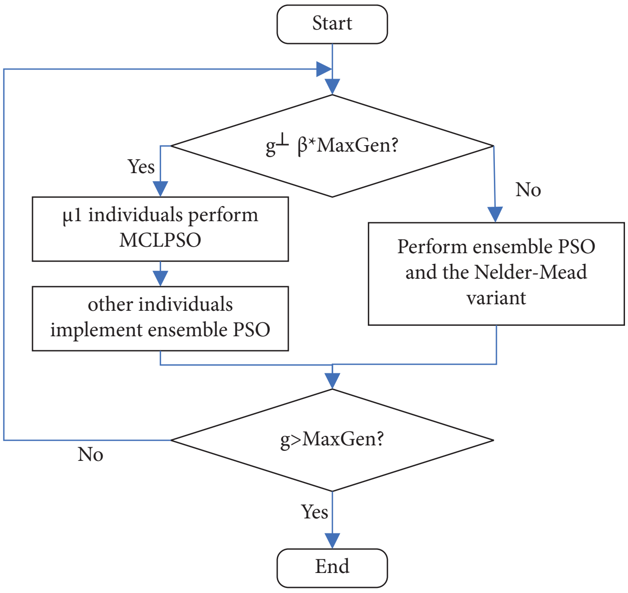
Start
g⊥ β*MaxGen?
Yes
No
μ1 individuals perform MCLPSO
other individuals implement ensemble PSO
Perform ensemble PSO and the Nelder-Mead variant
g>MaxGen?
No
Yes
End

Figure 1: Framework figure of MSPSO.

4. Experiments and Results

4.1. Benchmark Functions and Comparative Algorithms. CEC 2014 [25] benchmark functions are used to evaluate the performance of MSPSO. Benchmark problems in CEC 2014 are developed with several novel features such as novel basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. In CEC 2014 benchmark suite, F1–F3 are unimodal functions, F4–F16 are simple multimodal functions, F17–F22 are hybrid functions, and F23–F30 are composition functions. In order to evaluate the mean and standard deviation of solution errors, we take thirty independent runs for each algorithm on each problem in 10 and 30 dimensions. For 10 dimensions, each run lasts up to 100,000 function evaluations (FES). For 30 dimensions, it is up to 300,000 FES per run.

The performance of MSPSO is compared with eight other PSO variants, which are PSO [10], CLPSO [26], LIPS [21], HPSO-TVAC [22], FDR-PSO [23], EPSO [27], OSC-PSO [28], and A-PSO [29]. All the selected peer algorithms are proposed in the last decade. EPSO is an ensemble PSO. OSC-PSO drives particles into oscillatory trajectories. A-PSO introduces the nonlinear dynamic acceleration coefficients, logistic map, and a modified particle position update approach in PSO. In order to verify the effectiveness of all the improved strategies proposed by us, we compare MSPSO with the original CLPSO algorithm. The parameter settings of these algorithms are listed in Table 1. The parameter settings of MSPSO in different dimensions are given in Table 2. The high dimension of the function is more complex compared to the low dimension of the function, so we use a larger population size for the high dimension of the function to maintain the diversity of the population. Parameters limit_{gp}, α, and β can affect the population diversity and convergence of MSPSO. The larger the limit_{gp}, the greater the probability of updating particles in MCLPSO based on the global optimal position. Also, the larger the α, the greater the probability of updating particle velocities in MCLPSO based on the mean velocity of the population. The smaller the β, the more times ensemble PSO is used in the whole population. The benchmark problems with 30 variables in CEC 2014 are more complex than those with 10 variables. So, we use different parameter settings in MSPSO for different dimensions.

Experimental results in the CEC 2014 suite with 10 and 30 dimensions are reported in Tables 3 and 4, respectively. The error is an absolute value of the difference between the best value for 30 runs and the actual optimal value of a specific objective function.

The nonparametric statistical test has become an important method to compare a group of evolutionary algorithms recently [30]. In this study, the Wilcoxon signed-rank test is employed to estimate MSPSO and other PSO variants with the significance level of 5%. For each algorithm, Tables 3 and 4 show the number of best/2nd best/worst ranking, the number of average ranking, and the number of +/=/− in the last three rows, respectively. The algorithms are ranked according to the mean error of each algorithm. Symbol "+," "=," and "−" indicate that MSPSO is significantly better than, similar to, and worse than the compared PSO variant, respectively.

The simulation results on 30 functions with 10 variables in CEC 2014 are shown in Table 3. The results show that MSPSO outperformed the other eight algorithms on functions F2, F3, F5, F6, F13, F17, F21, F24, F25, F27, and F30. For function F8 and F26, the mean error of MSPSO was equal to other optimal algorithms. Specifically, for unimodal functions, compared with other algorithms, MSPSO generally outperformed other algorithms. It is superior or equal to the other eight algorithms on all functions except F1. But it ranked third on functions F1. For simple multimodal functions, MSPSO shows the best performance on four

```
Input: fitness function $f(x)$, dimension $n$, population size $\mu$, subpopulation size $\mu_1$, maximum number of generation Max_Gen,
MCLPSO limit value $\text{limit}_{gp1} and$ $\text{limit}_{gp2}$, MCLPSO adjustment coefficient $\alpha_1$ and $\alpha_2$, iterative parameter $\beta$, interval iteration $lp$.
Generate an initial population $P$ consisting of $\mu$ individuals at random.
Set $lp = 10$, $Cr_k = 0$, $p_k = 1/5$.
for $g = 1, 2, \cdots$, Max_Gen do
  if $g \leq \beta *$ Max_Gen
  for $i = 1, 2, \cdots, \mu_1$ do
    Perform MCLPSO to generate offsprings.
  end for
  for $i = \mu_1 + 1, \mu_1 + 2, \cdots, \mu$ do
      Using improved adaptive probability adjustment method to generate offsprings and update the selection probability.
    end for
  else
    for $i = 1, 2, \cdots, \mu$ do
      Using improved adaptive probability adjustment method to generate offsprings and update the selection probability.
    end for
    Update the best three individuals using Algorithm 1;
  end if
  Obtain the fitness value $f_{\text{best}}$ of the optimal in the population
  if $f_{\min} > f_{\text{best}}$ then
    $f_{\min} = f_{\text{best}}$
    $I_{\min} = x_{\min}$
  end if
end for
Output: the best fitness value $f_{\min}$.
```

Algorithm 2: MSPSO.

Table 1: Parameter setting for 8 algorithms.

Methods	ω	Contraction coefficient	Acceleration parameters	Neighbourhood size
PSO	0.9–0.2	—	$c1 = 2$, $c2 = 2$	—
FDR-PSO	0.9–0.2	0.729	$c1 = 1$, $c2 = 1$, $c3 = 2$	—
HPSO-TVAC	—	—	$c1 = 2.5–0.5$ $c2 = 0.5–2.5$	—
LIPS	—	0.729	$c = 2$	3
FIPS	—	0.7298	c1 = $c2 = 2.05$	—
CLPSO	0.9–0.2	—	$c = 1.49445$	—

Table 2: Parameter setting for MSPSO.

Parameters	μ_1	μ	limit_{gp1}	limit_{gp2}	α_1	α_2	β	lp
CEC 2014 10D	9	20	0.5	0.5	1	0.9	0.9	10
CEC 2014 30D	15	40	0.5	0.5	1	0.9	0.9	10

functions and a moderate performance compared with other comparative algorithms on other functions. With regard to hybrid functions, the performance of MSPSO ranked within the top 4 on all functions. As for composition functions, MSPSO shows the best performance on five functions. And it ranked within the top 2 on all functions except F28. For other algorithms, EPSO and CLPSO perform well, ranking second and third, respectively. PSO, FDR-PSO, HPSO-TVAC, and OSC-PSO perform moderately. All of these algorithms win on no more than 3 functions. LIPS and A-PSO also win on a few functions, but their average ranking is the lowest. To sum up, first, compared with the other eight competitors, MSPSO indicates the best overall performance on all 30 functions in CEC 2014 with 10 variables in terms of the number of the best and average ranking. Second, MSPSO is significantly different from other algorithms in most of the functions.

The simulation results on 30 functions with 30 variables in CEC 2014 are shown in Table 4. The results show that MSPSO outperformed the other eight algorithms on functions F2, F3, F4, F5, F6, F7, F11, F15, F17, F20, and F27. For function F26, the mean error of MSPSO was equal to other optimal algorithms. Specifically, for unimodal functions, MSPSO generally performs better than most of the other algorithms. Also, it is superior or equal to the other eight algorithms on all functions except F1. Furthermore, for simple multimodal functions, MSPSO exhibits a better performance. Also, it is superior to EPSO in eight functions. Regarding hybrid functions, MSPSO shows a better performance compared with other comparative PSO variants. Also, it ranked within the top 3 on all functions. As for

Table 3: Computational result of benchmark functions in CEC 2014 with 10 variables.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F1	Mean	1.96E+04	**5.37E+02**	3.95E+04	1.25E+05	3.60E+04	2.79E+03	3.84E+06	7.86E+03	6.59E+03
	Std.	1.33E+04	4.83E+02	3.92E+04	2.08E+05	3.29E+04	2.48E+03	2.16E+06	6.56E+03	4.78E+03
	Rank	5 (+)	1 (−)	7 (+)	8 (+)	6 (+)	2 (−)	9 (+)	4 (+)	3
F2	Mean	3.06E+03	9.91E+02	2.51E+02	2.29E+03	4.40E+02	4.57E+02	3.84E+08	2.04E+02	**6.21E+00**
	Std.	1.77E+03	4.93E+02	3.10E+02	2.26E+03	7.18E+02	1.83E+02	4.63E+07	2.21E+02	8.55E+00
	Rank	8 (+)	6 (+)	3 (+)	7 (+)	4 (+)	5 (+)	9 (+)	2 (+)	1
F3	Mean	6.67E+02	2.26E+02	5.22E+02	7.37E+02	1.86E+02	3.04E+02	2.34E+03	2.69E+01	**2.57E+00**
	Std.	4.30E+02	1.74E+02	8.57E+02	7.51E+02	7.21E+01	2.61E+02	3.12E+02	4.42E+01	1.20E+01
	Rank	7 (+)	4 (+)	6 (+)	8 (+)	3 (+)	5 (+)	9 (+)	2 (=)	1
F4	Mean	3.74E+01	1.16E+01	1.38E+00	3.16E+01	**3.00E−02**	2.42E+01	4.22E+01	5.58E−02	1.23E+00
	Std.	2.81E−01	2.01E+01	2.98E−01	2.74E+01	5.61E−03	2.10E+01	1.10E+00	5.58E−02	1.43E+00
	Rank	8 (+)	5 (+)	4 (+)	7 (+)	1 (−)	6 (+)	9 (+)	2 (−)	3
F5	Mean	2.02E+01	2.02E+01	2.00E+01	2.00E+01	2.01E+01	2.02E+01	2.04E+01	2.01E+01	**1.33E+01**
	Std.	8.16E−02	1.42E−01	3.96E−06	1.30E−03	1.36E−02	4.26E−02	2.63E−02	4.18E−02	1.15E+01
	Rank	6 (+)	6 (+)	2 (−)	2 (−)	4 (+)	6 (+)	9 (+)	4 (+)	1
F6	Mean	1.30E+00	2.29E+00	2.24E+00	1.61E+00	1.05E−01	1.95E+00	6.10E+00	**9.10E−02**	2.50E−01
	Std.	1.11E+00	2.63E+00	8.34E−01	7.65E−02	2.75E−02	1.60E+00	1.39E+00	2.45E−02	2.47E−01
	Rank	4 (+)	8 (+)	7 (+)	5 (+)	2(−)	6 (+)	9 (+)	1 (−)	3
F7	Mean	8.61E−02	1.51E−01	3.40E−01	**1.15E−02**	5.08E−02	1.68E−01	8.48E+00	6.27E−02	5.90E−02
	Std.	3.49E−02	1.14E−01	6.29E−02	1.24E−02	1.65E−02	7.28E−02	1.42E+00	3.94E−02	1.61E−02
	Rank	5 (+)	6 (+)	8 (+)	1 (−)	2 (−)	7 (+)	9 (+)	4 (+)	3
F8	Mean	3.32E+00	3.65E+00	3.32E−01	9.62E+00	**0.00E+00**	5.97E+00	3.12E+01	**0.00E+00**	**0.00E+00**
	Std.	2.07E+00	2.50E+00	5.74E−01	2.07E+00	0.00E+00	3.45E+00	4.75E+00	0.00E+00	0.00E+00
	Rank	5 (+)	6 (+)	4 (+)	8 (+)	1 (=)	7 (+)	9 (+)	1 (=)	1
F9	Mean	5.97E+00	1.03E+01	1.56E+01	1.39E+01	4.04E+00	9.95E+00	4.21E+01	4.76E+00	**3.31E+00**
	Std.	9.95E−01	4.60E+00	1.10E+01	5.26E+00	9.93E−01	6.52E+00	9.11E−01	3.72E−01	1.14E+00
	Rank	4 (+)	6 (+)	8 (+)	7 (+)	2 (+)	5 (+)	9 (+)	3 (+)	1
F10	Mean	6.24E+01	1.31E+02	2.08E+02	4.99E+02	**0.00E+00**	2.68E+02	7.66E+02	2.32E−01	1.24E−01
	Std.	7.38E+01	1.23E+02	5.85E+01	1.38E+02	0.00E+00	1.24E+01	2.06E+02	7.09E−02	1.05E−12
	Rank	4 (+)	5 (+)	6 (+)	8 (+)	1 (−)	7 (+)	9 (+)	3 (=)	2
F11	Mean	2.86E+02	5.05E+02	1.49E+02	5.36E+02	**9.18E+01**	7.72E+02	7.40E+02	9.60E+01	1.80E+02
	Std.	1.18E+02	1.37E+02	2.34E+02	1.58E+02	6.63E+01	9.76E+01	9.10E+01	1.34E+02	1.18E+02
	Rank	5 (+)	6 (+)	3 (=)	7 (+)	1 (−)	9 (+)	8 (+)	2 (−)	4
F12	Mean	1.21E−01	1.02E−01	2.08E−01	**7.06E−02**	1.70E−01	4.34E−01	1.03E+00	2.14E−01	7.70E−02
	Std.	1.33E−02	6.71E−02	1.08E−01	6.77E−02	1.54E−02	7.11E−02	2.00E−01	1.17E−01	6.42E−02
	Rank	4 (+)	3 (+)	6 (+)	1 (−)	5 (+)	8 (+)	9 (+)	7 (+)	2
F13	Mean	1.80E−01	9.31E−02	4.14E−01	9.39E−02	1.24E−01	1.71E−01	5.39E−01	9.94E−02	**5.92E−02**
	Std.	1.03E−01	2.51E−02	2.35E−01	3.28E−02	2.78E−02	2.27E−02	3.97E−02	1.48E−02	3.89E−02
	Rank	7 (+)	2 (+)	8 (+)	3 (+)	5 (+)	6 (+)	9 (+)	4 (+)	1

Table 3: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F14	Mean	1.22E − 01	1.06E − 01	2.43E − 01	3.77E − 01	1.51E − 01	2.11E − 01	6.32E − 01	**8.31E − 02**	8.86E − 02
	Std.	2.79E − 02	7.31E − 02	9.40E − 02	9.76E − 02	6.60E − 02	1.37E − 01	1.75E − 01	7.21E − 02	6.66E − 02
	Rank	4 (+)	2 (=)	7 (+)	8 (+)	5 (+)	6 (+)	9 (+)	1 (−)	2
F15	Mean	6.96E − 01	9.91E − 01	2.14E + 00	6.20E − 01	**5.85E − 01**	8.12E − 01	6.07E + 00	7.13E − 01	7.01E − 01
	Std.	1.19E − 01	1.79E − 01	8.23E − 01	3.34E − 01	6.62E − 02	2.77E − 01	1.56E − 01	3.53E − 01	1.37E − 01
	Rank	3 (−)	7 (+)	8 (+)	2 (−)	1 (−)	6 (+)	9 (+)	5 (+)	4
F16	Mean	1.95E + 00	2.72E + 00	2.57E + 00	2.85E + 00	1.84E + 00	1.93E + 00	3.22E + 00	**1.42E + 00**	1.59E + 00
	Std.	2.79E − 01	4.14E − 01	4.70E − 01	4.06E − 01	1.76E − 01	4.50E − 01	1.56E − 01	2.08E − 01	4.22E − 01
	Rank	5 (+)	7 (+)	6 (+)	8 (+)	3 (+)	4 (+)	9 (+)	1 (−)	2
F17	Mean	3.66E + 03	2.97E + 03	2.01E + 03	4.29E + 04	6.04E + 03	4.29E + 03	8.11E + 03	2.86E + 03	**1.18E + 03**
	Std.	2.87E + 03	2.96E + 03	1.91E + 03	1.92E + 04	4.31E + 03	1.75E + 03	1.68E + 03	3.31E + 03	1.03E + 03
	Rank	5 (+)	4 (+)	2 (+)	9 (+)	7 (+)	6 (+)	8 (+)	3 (+)	1
F18	Mean	1.02E + 02	1.96E + 03	5.24E + 03	3.26E + 03	**3.09E + 01**	3.24E + 03	9.19E + 03	6.08E + 02	5.06E + 02
	Std.	6.78E + 01	1.63E + 03	5.39E + 03	1.63E + 03	2.39E + 01	3.87E + 03	1.21E + 03	4.26E + 02	4.35E + 03
	Rank	2 (−)	5 (+)	8 (+)	7 (+)	1 (−)	6 (+)	9 (+)	4 (+)	3
F19	Mean	1.07E + 00	1.51E + 00	1.71E + 00	2.66E + 00	4.34E − 01	2.71E + 00	3.96E + 00	**1.09E − 01**	5.32E − 01
	Std.	8.56E − 01	8.67E − 01	1.20E + 00	1.08E + 00	5.39E − 01	1.59E + 00	4.41E − 01	6.57E − 02	4.44E − 01
	Rank	4 (+)	5 (+)	6 (+)	7 (+)	2 (−)	8 (+)	9 (+)	1 (−)	3
F20	Mean	7.26E + 01	3.10E + 03	1.75E + 03	8.52E + 03	3.16E + 01	4.60E + 01	8.79E + 02	**1.38E + 01**	3.59E + 01
	Std.	2.01E + 01	4.25E + 03	2.49E + 03	1.27E + 04	4.11E + 01	1.81E + 01	7.50E + 01	4.36E + 00	1.62E + 01
	Rank	5 (+)	8 (+)	7 (+)	9 (+)	2 (−)	4 (+)	6 (+)	1 (−)	3
F21	Mean	1.05E + 02	7.95E + 01	1.84E + 03	3.17E + 03	1.30E + 02	9.02E + 01	5.67E + 03	3.08E + 01	**2.87E + 01**
	Std.	7.52E + 01	1.36E + 02	1.22E + 03	4.48E + 03	1.12E + 02	3.28E + 01	4.18E + 03	3.67E + 01	3.56E + 01
	Rank	5 (+)	3 (+)	7 (+)	8 (+)	6 (+)	4 (+)	9 (+)	2 (+)	1
F22	Mean	4.86E + 01	5.48E + 01	1.27E + 02	8.51E + 01	**3.69E − 01**	2.09E + 01	5.27E + 01	1.72E + 00	4.86E − 01
	Std.	6.67E + 01	7.74E + 01	4.40E + 01	1.08E + 02	4.55E − 01	4.77E − 01	8.41E + 00	1.19E + 00	3.30E − 01
	Rank	5 (+)	7 (+)	9 (+)	8 (+)	1 (−)	4 (+)	6 (+)	3 (+)	2
F23	Mean	3.29E + 02	3.29E + 02	3.29E + 02	3.29E + 02	3.29E + 02	3.29E + 02	**2.92E + 02**	3.29E + 02	3.29E + 02
	Std.	0.00E + 00	2.64E − 13	1.89E − 12	2.08E − 12	2.64E − 13	2.64E − 13	7.94E + 01	2.64E − 13	2.64E − 13
	Rank	2 (=)	2 (=)	2 (=)	2 (=)	2 (=)	2 (=)	1 (−)	2 (=)	2
F24	Mean	1.22E + 02	1.18E + 02	1.55E + 02	1.39E + 02	1.13E + 02	1.23E + 02	1.82E + 02	1.11E + 02	**1.10E + 02**
	Std.	1.96E + 00	5.08E + 00	4.32E + 01	2.19E + 01	3.32E + 00	3.42E + 00	3.17E + 01	1.00E + 01	1.43E + 00
	Rank	5 (+)	4 (+)	8 (+)	7 (+)	3 (+)	6 (+)	9 (+)	2 (+)	1
F25	Mean	1.78E + 02	1.48E + 02	1.83E + 02	1.75E + 02	1.82E + 02	1.76E + 02	2.00E + 02	1.54E + 02	**1.27E + 02**
	Std.	3.76E + 01	4.11E + 01	2.92E + 01	3.96E + 01	4.17E + 00	4.59E + 01	0.00E + 00	4.04E + 01	4.63E + 00
	Rank	6 (+)	2 (+)	8 (+)	4 (+)	7 (+)	5 (+)	9 (+)	3 (+)	1
F26	Mean	**1.00E + 02**	1.33E + 02	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**	**1.00E + 02**
	Std.	2.13E − 02	5.77E + 01	1.32E − 01	5.34E − 02	3.45E − 02	1.15E − 02	7.89E − 02	1.99E − 02	2.93E − 02
	Rank	1 (+)	9 (+)	1 (+)	1 (=)	1 (+)	1 (+)	1 (+)	1 (+)	1

Table 3: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F27	Mean	3.41E+02	2.36E+02	2.75E+02	3.59E+02	1.13E+02	3.73E+00	1.47E+02	1.12E+02	**2.87E+00**
	Std.	5.15E+01	2.08E+02	2.35E+02	3.60E+01	1.86E+02	1.51E+00	2.24E+02	1.90E+02	1.85E+00
	Rank	9 (+)	6 (+)	7 (+)	8 (+)	4 (+)	2 (+)	5 (+)	3 (+)	1
F28	Mean	4.59E+02	5.24E+02	**3.85E+02**	5.44E+02	**3.85E+02**	4.09E+02	3.96E+02	3.92E+02	3.88E+02
	Std.	7.56E+01	1.03E+02	1.37E+02	5.80E+01	4.73E+01	7.60E+01	2.97E+00	3.44E+01	2.28E+01
	Rank	7 (+)	8 (+)	1 (−)	9 (+)	1 (−)	6 (+)	5 (+)	4 (+)	3
F29	Mean	7.14E+02	3.41E+02	**2.20E+02**	8.06E+02	2.83E+02	6.48E+02	6.83E+05	2.51E+02	2.68E+02
	Std.	6.66E+02	6.05E+01	1.87E+01	3.82E+02	4.96E+01	8.79E+01	1.18E+06	8.97E+00	1.00E+01
	Rank	7 (+)	5 (+)	1 (−)	8 (+)	4 (+)	6 (+)	9 (+)	3 (+)	2
F30	Mean	6.17E+02	1.39E+03	9.52E+02	1.52E+03	6.10E+02	7.84E+02	9.24E+02	6.87E+02	**6.01E+02**
	Std.	7.36E+01	3.32E+02	9.20E+02	8.71E+01	1.14E+02	1.14E+02	3.29E+02	7.65E+01	3.35E+01
	Rank	3 (+)	8 (+)	7 (+)	9 (+)	2 (+)	5 (+)	6 (+)	4 (+)	1
Best/2nd best/worst ranking		1/2/0	1/4/0	3/3/0	3/3/4	9/7/0	1/3/1	2/0/21	7/7/0	13/7/0
Avg. ranking		5	5.2	5.57	6.2	2.97	5.33	7.83	2.73	1.98
+/=/−		27/1/2	27/2/1	24/3/3	24/2/4	17/2/11	28/1/1	29/0/1	19/4/7	
Algorithms		PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO

Bold values represent the best results of each function.

Table 4: Computational result of benchmark functions in CEC 2014 with 30 variables.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F1	Mean	8.17E+06	9.82E+04	4.47E+05	2.78E+07	5.33E+06	1.53E+07	1.35E+08	**7.82E+04**	1.36E+05
	Std.	7.53E+06	5.80E+04	3.46E+05	3.62E+07	2.23E+06	3.78E+06	3.40E+07	7.51E+03	1.00E+05
	Rank	6 (+)	2 (−)	4 (+)	8 (+)	5 (+)	7 (+)	9 (+)	1 (−)	3
F2	Mean	1.80E+01	4.43E+02	2.37E+03	4.99E+03	1.72E+02	6.98E+01	9.61E+09	**2.86E−01**	4.73E+00
	Std.	3.00E+01	7.63E+02	1.40E+03	8.23E+03	1.82E+02	1.08E+02	2.11E+09	3.09E−01	7.48E+00
	Rank	3 (+)	6 (+)	7 (+)	8 (+)	5 (+)	4 (+)	9 (+)	1 (−)	2
F3	Mean	3.89E+01	4.14E+01	3.83E+02	8.42E+03	1.67E+02	3.70E+01	3.18E+04	8.36E−02	**6.07E−02**
	Std.	3.88E+01	6.44E+01	1.79E+02	1.11E+04	4.40E+01	1.29E+01	3.34E+03	7.31E−02	6.59E−02
	Rank	4 (+)	5 (+)	7 (+)	8 (+)	6 (+)	3 (+)	9 (+)	2 (+)	1
F4	Mean	1.56E+02	2.24E+01	3.37E+01	2.64E+02	6.36E+01	1.53E+02	6.14E+02	4.93E−01	**3.61E−01**
	Std.	5.06E+01	3.76E+01	3.64E+01	1.43E+02	3.18E+01	2.26E+01	1.01E+02	7.10E−02	1.94E−01
	Rank	7 (+)	3 (+)	4 (+)	8 (+)	5 (+)	6 (+)	9 (+)	2 (+)	1
F5	Mean	2.09E+01	2.04E+01	**2.00E+01**	**2.00E+01**	2.04E+01	2.08E+01	2.09E+01	2.03E+01	**2.00E+01**
	Std.	1.24E−01	3.92E−01	8.20E−06	1.91E−04	6.69E−02	1.20E−01	9.54E−02	1.32E−02	1.25E−05
	Rank	8 (+)	5 (+)	1 (−)	1 (−)	5 (+)	7 (+)	8 (+)	4 (+)	1
F6	Mean	1.28E+01	7.56E+00	1.94E+01	1.74E+01	1.23E+01	1.87E+01	2.89E+01	9.00E+00	**6.89E+00**
	Std.	3.46E+00	2.86E+00	2.27E+00	3.22E+00	7.72E−01	3.57E+00	1.07E+00	2.94E−01	1.74E+00
	Rank	5 (+)	2 (+)	8 (+)	6 (+)	4 (+)	7 (+)	9 (+)	3 (+)	1
F7	Mean	2.21E−02	1.97E−02	9.01E−03	3.29E−03	4.98E−07	1.23E−02	8.47E+01	**9.79E−09**	1.54E−07
	Std.	1.95E−02	1.91E−02	1.56E−02	5.69E−03	2.09E−07	2.46E−03	8.05E+00	1.70E−08	2.68E−07
	Rank	8 (+)	7 (+)	5 (+)	4 (+)	3 (+)	6 (+)	9 (+)	1 (−)	2
F8	Mean	2.35E+01	3.71E+01	9.62E+00	5.64E+01	**1.14E−13**	6.90E+01	2.23E+02	2.27E−13	3.33E−12
	Std.	2.07E+00	1.38E+01	3.20E+00	1.65E+01	0.00E+00	3.51E+01	7.48E+00	0.00E+00	2.36E−12
	Rank	5 (+)	6 (+)	4 (+)	7 (+)	1 (−)	8 (+)	9 (+)	2 (−)	3
F9	Mean	7.10E+01	5.84E+01	9.72E+01	6.57E+01	4.78E+01	9.75E+01	2.55E+02	4.90E+01	**3.97E+01**
	Std.	4.49E+00	1.96E+01	6.53E+01	1.43E+01	5.63E+00	1.15E+01	7.61E+00	6.55E+00	3.58E+00
	Rank	6 (+)	4 (+)	7 (+)	5 (+)	2 (+)	8 (+)	9 (+)	3 (+)	1
F10	Mean	5.60E+02	7.87E+02	3.67E+02	1.94E+03	**7.11E−01**	1.67E+03	5.53E+03	5.84E+00	4.00E+00
	Std.	3.28E+02	2.87E+02	2.39E+02	2.99E+02	2.14E−01	5.97E+02	5.05E+02	2.84E+00	6.51E−01
	Rank	5 (+)	6 (+)	4 (+)	8 (+)	1 (−)	7 (+)	9 (+)	3 (+)	2
F11	Mean	2.48E+03	2.82E+03	3.08E+03	2.69E+03	2.22E+03	3.47E+03	6.93E+03	**2.02E+03**	2.35E+03
	Std.	7.18E+02	8.01E+02	6.64E+02	3.29E+02	2.92E+02	6.89E+02	9.08E+01	1.78E+02	1.25E+02
	Rank	4 (+)	6 (+)	7 (+)	5 (+)	2 (−)	8 (+)	9 (+)	1 (−)	3
F12	Mean	1.66E+00	7.75E−01	2.59E−01	1.55E−01	4.27E−01	8.86E−01	2.38E+00	1.94E−01	**1.40E−01**
	Std.	4.46E−01	6.84E−01	8.74E−02	5.77E−02	1.36E−01	6.91E−01	1.62E−01	1.94E−01	2.57E−02
	Rank	8 (+)	6 (+)	4 (+)	2 (=)	5 (+)	7 (+)	9 (+)	3 (+)	1
F13	Mean	5.09E−01	2.67E−01	4.12E−01	2.62E−01	3.34E−01	4.12E−01	2.08E+00	2.75E−01	**1.81E−01**
	Std.	1.05E−01	8.41E−02	7.90E−02	2.27E−02	3.69E−02	2.87E−02	2.91E−01	3.02E−02	6.19E−02
	Rank	8 (+)	3 (−)	6 (+)	2 (+)	5 (+)	6 (+)	9 (+)	4 (+)	1

Table 4: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F14	Mean	2.91E − 01	2.25E − 01	3.07E − 01	2.67E − 01	2.51E − 01	3.21E − 01	2.54E + 01	**2.19E − 01**	2.36E − 01
	Std.	4.78E − 02	2.41E − 02	4.75E − 02	1.47E − 02	9.50E − 03	4.43E − 02	2.63E + 00	3.79E − 02	1.55E − 02
	Rank	6 (+)	2 (−)	7 (+)	5 (+)	4 (+)	8 (+)	9 (+)	1 (−)	3
F15	Mean	7.08E + 00	4.61E + 00	3.64E + 01	7.49E + 00	7.43E + 00	1.12E + 01	1.50E + 03	4.39E + 00	**2.67E + 00**
	Std.	4.15E + 00	9.58E − 01	8.33E + 00	3.07E + 00	1.09E + 00	7.51E − 01	9.12E + 02	7.15E − 01	6.67E − 01
	Rank	4 (+)	3 (+)	8 (+)	6 (+)	5 (+)	7 (+)	9 (+)	2 (+)	1
F16	Mean	1.10E + 01	9.92E + 00	1.03E + 01	1.17E + 01	1.03E + 01	1.07E + 01	1.27E + 01	9.84E + 00	**8.95E + 00**
	Std.	1.23E + 00	1.10E − 01	5.82E − 01	2.26E − 01	4.98E − 02	1.37E + 00	1.04E − 01	3.80E − 01	7.44E − 01
	Rank	7 (+)	3(+)	4 (+)	8 (+)	4 (+)	6 (+)	9 (+)	2 (+)	1
F17	Mean	7.03E + 05	3.82E + 04	9.13E + 04	2.50E + 05	6.83E + 05	8.62E + 04	4.10E + 06	4.08E + 04	**2.05E + 04**
	Std.	3.10E + 05	1.81E + 04	7.73E + 04	9.41E + 04	1.41E + 05	6.02E + 04	3.89E + 06	2.18E + 04	2.41E + 03
	Rank	8 (+)	2 (+)	5 (+)	6 (+)	7 (+)	4 (+)	9 (+)	3 (+)	1
F18	Mean	2.22E + 02	3.07E + 03	2.41E + 02	5.83E + 02	**8.79E + 01**	1.90E + 03	1.06E + 08	2.71E + 02	1.01E + 02
	Std.	1.18E + 02	1.60E + 03	1.16E + 02	4.58E + 02	1.71E + 01	5.34E + 02	3.87E + 07	1.55E + 02	9.15E + 01
	Rank	3 (+)	8 (+)	4 (+)	6 (+)	1 (−)	7 (+)	9 (+)	5 (+)	2
F19	Mean	8.06E + 00	5.22E + 00	1.65E + 01	3.29E + 01	6.97E + 00	1.23E + 01	7.17E + 01	7.30E + 00	**4.68E + 00**
	Std.	3.38E + 00	1.74E + 00	2.48E + 00	4.20E + 01	6.06E − 01	1.92E + 00	1.20E + 01	1.63E + 00	7.77E − 01
	Rank	5 (+)	2 (+)	7 (+)	8 (+)	3 (+)	6 (+)	9 (+)	4 (+)	1
F20	Mean	3.10E + 02	5.79E + 02	1.76E + 03	8.36E + 03	2.05E + 03	1.14E + 03	1.52E + 04	2.22E + 02	**1.63E + 02**
	Std.	3.58E + 01	2.53E + 02	9.85E + 02	4.39E + 03	5.18E + 02	9.09E + 02	7.81E + 03	1.14E + 02	7.36E + 01
	Rank	3 (+)	4 (+)	6 (+)	8 (+)	7 (+)	5 (+)	9 (+)	2 (+)	1
F21	Mean	3.35E + 04	2.77E + 04	1.54E + 04	3.10E + 05	3.30E + 04	**1.25E + 04**	1.25E + 06	1.58E + 04	1.49E + 04
	Std.	1.47E + 04	9.11E + 03	1.21E + 04	3.96E + 05	1.06E + 04	1.26E + 04	2.35E + 05	1.24E + 04	1.13E + 04
	Rank	7 (+)	5 (+)	3 (=)	8 (+)	6 (+)	1 (−)	9 (+)	4 (+)	2
F22	Mean	3.55E + 02	2.82E + 02	4.55E + 02	3.20E + 02	**7.27E + 01**	5.36E + 02	7.18E + 02	2.40E + 02	1.54E + 02
	Std.	2.50E + 02	1.26E + 02	1.47E + 02	1.09E + 02	6.20E + 01	8.25E + 01	1.34E + 02	1.32E + 02	1.80E + 01
	Rank	6 (+)	4 (+)	7 (+)	5 (+)	1 (−)	8 (+)	9 (+)	3 (+)	2
F23	Mean	3.16E + 02	3.15E + 02	3.14E + 02	3.24E + 02	3.15E + 02	3.16E + 02	**3.08E + 02**	3.15E + 02	3.15E + 02
	Std.	1.25E − 01	2.36E − 12	2.30E − 11	8.89E + 00	2.01E − 09	2.20E − 01	9.45E + 01	2.24E − 12	9.46E − 13
	Rank	7 (+)	3 (=)	2 (−)	9 (+)	3 (=)	7 (+)	1 (−)	3 (=)	3
F24	Mean	2.34E + 02	2.24E + 02	2.30E + 02	2.39E + 02	2.24E + 02	2.25E + 02	**2.00E + 02**	2.26E + 02	2.25E + 02
	Std.	1.27E + 01	1.91E + 00	1.49E + 00	7.90E + 00	3.46E − 01	8.14E − 01	3.52E − 07	3.21E − 01	7.88E − 01
	Rank	8 (+)	2 (−)	7 (+)	9 (+)	2 (−)	4 (=)	1 (−)	6 (=)	5
F25	Mean	2.08E + 02	2.09E + 02	2.18E + 02	2.14E + 02	2.08E + 02	2.15E + 02	**2.00E + 02**	2.07E + 02	2.05E + 02
	Std.	1.91E + 00	3.98E − 01	4.81E + 00	1.16E + 00	9.54E − 01	4.31E + 00	0.00E + 00	9.08E − 01	1.29E + 00
	Rank	4 (+)	6 (+)	9 (+)	7 (+)	4 (+)	8 (+)	1 (−)	3 (+)	2
F26	Mean	1.01E + 02	1.67E + 02	1.34E + 02	1.34E + 02	**1.00E + 02**	**1.00E + 02**	1.67E + 02	**1.00E + 02**	**1.00E + 02**
	Std.	8.25E − 02	5.76E + 01	5.74E + 01	5.76E + 01	1.05E − 01	5.60E − 02	5.66E + 01	1.45E − 01	4.39E − 03
	Rank	5 (+)	8 (+)	6 (+)	6 (+)	1 (+)	1 (+)	8 (+)	1 (+)	1

Table 4: Continued.

Functions	Criteria	PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO
F27	Mean	6.72E + 02	4.89E + 02	6.74E + 02	5.30E + 02	4.11E + 02	8.68E + 02	9.09E + 02	**4.03E + 02**	**4.03E + 02**
	Std.	4.43E + 01	9.15E + 01	2.36E + 02	2.13E + 02	1.32E + 00	8.05E + 01	3.24E + 02	1.56E + 00	1.87E + 00
	Rank	6 (+)	4 (+)	7 (+)	5 (+)	3 (+)	8 (+)	9 (+)	1 (=)	1
F28	Mean	1.00E + 03	1.34E + 03	1.34E + 03	1.62E + 03	**8.67E + 02**	1.38E + 03	1.12E + 03	9.79E + 02	8.71E + 02
	Std.	6.73E + 01	7.40E + 02	7.96E + 02	1.60E + 02	3.56E + 01	3.24E + 02	8.97E + 01	2.11E + 01	4.58E + 01
	Rank	4 (+)	6 (+)	6 (+)	9 (+)	1 (−)	8 (+)	5 (+)	3 (+)	2
F29	Mean	8.15E + 06	1.15E + 03	**2.16E + 02**	1.27E + 03	1.01E + 03	5.40E + 06	3.65E + 05	1.07E + 03	9.81E + 02
	Std.	1.41E + 07	1.87E + 02	2.92E + 00	9.06E + 01	1.11E + 02	9.35E + 06	2.14E + 05	1.13E + 02	5.81E + 01
	Rank	9 (+)	5 (+)	1 (−)	6 (+)	3 (+)	8 (+)	7 (+)	4 (+)	2
F30	Mean	2.24E + 03	1.74E + 03	**8.95E + 02**	3.35E + 04	2.61E + 03	6.61E + 03	3.21E + 04	2.20E + 03	1.98E + 03
	Std.	2.34E + 02	6.64E + 02	4.21E + 02	2.02E + 04	2.20E + 02	3.92E + 03	1.49E + 03	2.22E + 02	5.36E + 02
	Rank	5 (+)	2 (+)	1 (−)	9 (+)	6 (+)	7 (+)	8 (+)	4 (+)	3
Best/2nd best/worst ranking		0/0/0	1/8/0	3/1/1	1/2/4	6/3/0	2/0/0	3/0/22	7/6/0	14/9/0
Avg. ranking		5.8	4.33	5.27	6.4	3.67	6.23	7.9	2.7	1.83
+/=/−		30/0/0	25/1/4	25/1/4	28/1/1	22/1/7	28/1/1	27/0/3	21/3/6	
Algorithms		PSO	FDR-PSO	HPSO-TVAC	LIPS	CLPSO	OSC-PSO	A-PSO	EPSO	MSPSO

Bold values represent the best results of each function.

composition functions, MSPSO shows the best performance on two functions and a moderate performance compared with other comparative algorithms on other functions. For other algorithms, EPSO and CLPSO also perform well, ranking second and third, respectively. The rest of these algorithms win on no more than 3 functions. The average ranking of A-PSO is the lowest. In summary, first, MSPSO also has the best overall performance compared with the other eight competitors on all functions in CEC 2014 with 30 variables. Second, MSPSO is also significantly different from other algorithms in most of the functions.

From Tables 3 and 4, we see that the performance of two ensemble PSO algorithms (EPSO and MSPSO) is better than other types of PSO algorithms. This result can be explained by the theory of easy and hard fitness functions [17]. According to that theory, the hardest problem to one evolutionary algorithm could be the easiest to another algorithm. Thus, given an ensemble of different PSO algorithms, a hard problem might be solved easily by one of them. Of course, if a problem is hard (or easy) to all of them, using an ensemble does not bring too much improvement.

The convergent speed is evaluated in Figures 2–8. From Figure 2, we can see that MSPSO obtains a better performance than other PSO variants. The convergent speed of MSPSO is not fast, but its optimization accuracy is higher than other competitors in many functions. This is because, in the early stage of the search, different particle generation strategies may interfere with the direction in which particles quickly find good positions. However, in the later stage of the search, different particle generation strategies increase the chances of particles finding good positions.

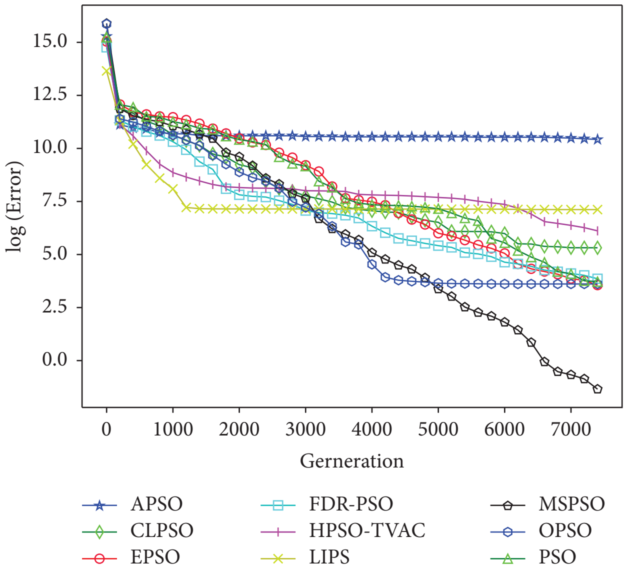

Figure 2: The results on F3 with 30 variables.

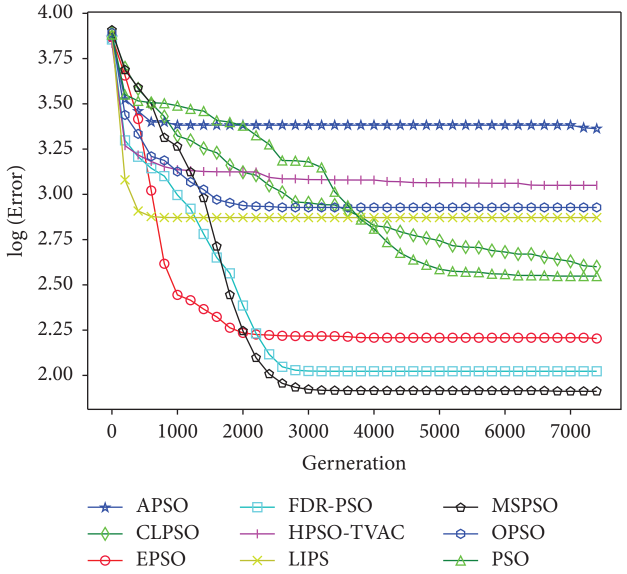

Figure 3: The results on F6 with 30 variables.

4.2. Application to the Hydrothermal Scheduling Problem. The hydrothermal scheduling problem [31] is a complex optimization problem from the real world. Its main objective is to schedule the power generations of the thermal and hydro units in the system to meet the load demands, under the premise of satisfying the constraints of the hydraulic systems and the power system networks. In order to evaluate the performance of hydrothermal scheduling problem in dealing with real-world problems, we apply MSPSO to solving this problem. In the hydrothermal scheduling problem, decision variables are nonlinearly related to the major operation problem of hydrothermal systems. The objective of the problem is to minimize the fuel cost of thermal units for 24 hours with four hydro units in the system, and the dimension of the problem is 96.

In order to meet load requirements during the scheduling period, the total fuel cost of the thermal system operation is expressed by F. The objective function is given as follows:

$$\text{Minimize } F = \sum_{i=1}^{M} f_i(P_{Ti}). \qquad (9)$$

In the previous formula, P_{Ti} is the power generation of an equivalent thermal unit at ith interval, and f_i represents the cost function corresponding to P_{Ti}. M is the total number of intervals considered for the short-term planning. The cost function f_i is expressed as follows:

$$f_i(P_{Ti}) = a_i P_{Ti}^2 + b_i P_{Ti} + c_i + \left| e_i \sin\left(f_i\left(P_{Ti}^{\min} - P_{Ti} \right) \right) \right|. \qquad (10)$$

MSPSO is compared with five other algorithms in three hydrothermal scheduling instances, which are CoBiDE [32], TLBO [33], ALC-PSO [34], DNS-PSO [35], and EPSO [27]. CoBiDE incorporates the covariance matrix learning and the bimodal distribution parameter setting into DE. TLBO designs an optimization mechanism inspired by the effect of the influence of a teacher on learners. TLBO divides the optimization process into “Teacher Phase” and “Learner Phase.” ALC-PSO transplants the aging mechanism to PSO

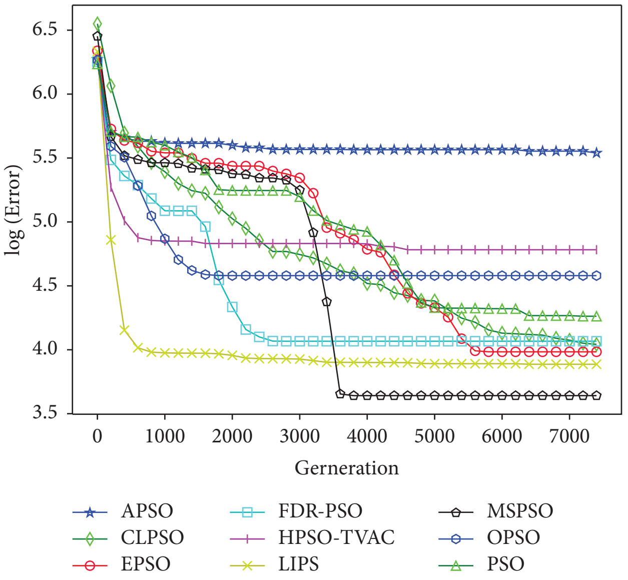

Figure 4: The results on F9 with 30 variables.

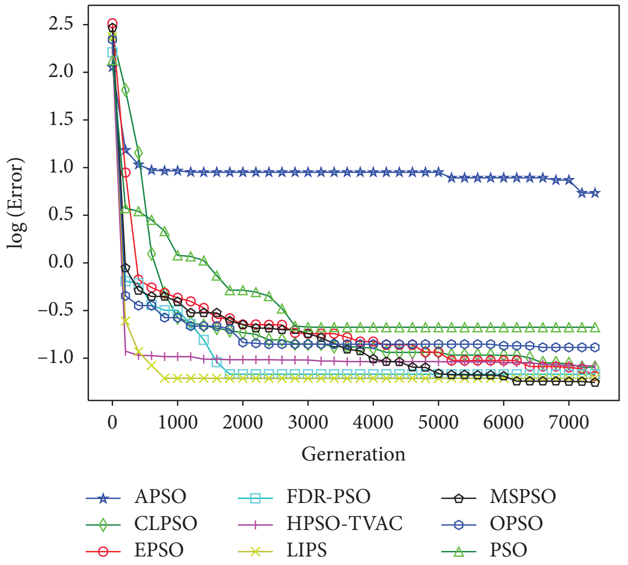

Figure 5: The results on F13 with 30 variables.

to overcome the problem of premature convergence. DNS-PSO employs a diversity-enhancing mechanism and neighborhood search strategies in PSO to achieve a trade-off between exploration and exploitation abilities.

The computational results of hydrothermal scheduling instances are shown in Table 5. From the table, MSPSO outperformed the other five comparative algorithms in two instances. This means that MSPSO is a good alternative algorithm for solving the hydrothermal scheduling problem.

4.3. The IEEE CEC 2010 Standard Test Functions Set. In order to further analyze the performance of MSPSO to solve the large-scale global optimization problem, CEC 2010 [36] is

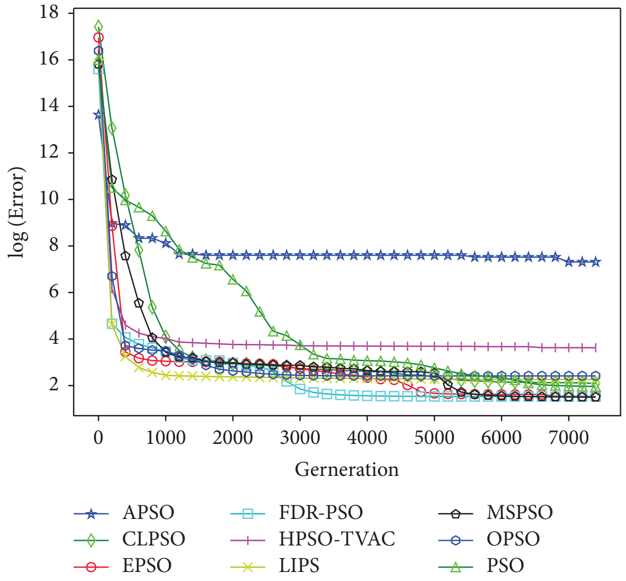

Figure 6: The results on F15 with 30 variables.

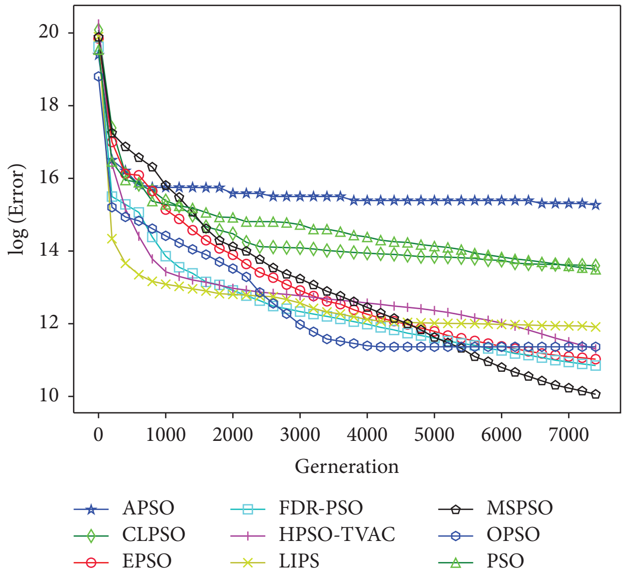

Figure 7: The results on F17 with 30 variables.

employed in experiments. The performance of MSPSO is compared with PSO [2], grey wolf optimizer (GWO) [37], standard sine cosine algorithm (SCA) [38], and slap swarm algorithm (SSA) [39]. All experiments are tested 30 times in 1000 dimensions. The mean and standard deviation of all algorithms are shown in Table 6. The average rank and rank are also recorded in the last two rows of Table 6. From Table 6, it shows that MSPSO has outperformance than other comparative algorithms to solve the large-scale global optimization problems. For most CEC 2010 functions, the MSPSO improves the accuracy by some orders of magnitudes. Therefore, the experimental results demonstrate that MSPSO has a good performance in solving the large-scale optimization problems.

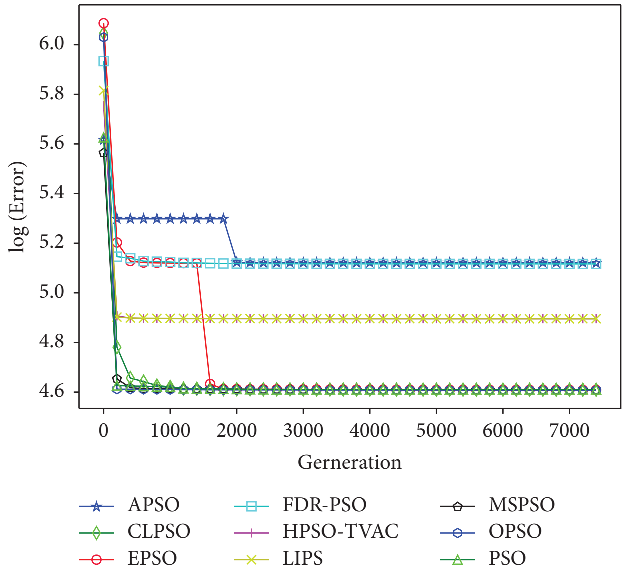

Figure 8: The results on F26 with 30 variables.

Table 5: Results of the hydrothermal scheduling instances.

Problems	Criteria	CoBiDE	TLBO	ALC-PSO	DNS-PSO	EPSO	MSPSO
Instance 1	Mean	1.10E + 06	1.28E + 06	1.06E + 06	9.77E + 05	9.42E + 05	9.38E + 05
	Std	7.90E + 04	8.49E + 05	1.69E + 05	8.38E + 04	2.72E + 03	2.93E + 03
Instance 2	Mean	1.73E + 06	1.54E + 06	1.51E + 06	1.41E + 06	1.15E + 06	1.27E + 06
	Std	1.32E + 05	7.93E + 05	3.27E + 05	1.90E + 05	1.71E + 05	1.11E + 05
Instance 3	Mean	1.09E + 06	1.28E + 06	1.20E + 06	9.78E + 05	9.46E + 05	9.45E + 05
	Std	7.02E + 04	8.49E + 05	4.45E + 05	8.57E + 04	3.66E + 03	6.32E + 03

Table 6: Computational result of CEC 2010 with 1000 variables.

Functions	Criteria	PSO	GWO	SCA	SSA	MSPSO
f1	Mean	4.32E + 10	5.69E + 10	1.57E + 11	6.16E + 10	1.59E + 10
	Std	6.39E + 09	5.44E + 09	1.08E + 10	5.31E + 09	1.26E + 09
f2	Mean	1.71E + 04	1.35E + 04	1.77E + 04	1.46E + 04	8.30E + 03
	Std	3.81E + 02	1.64E + 02	4.48E + 02	2.38E + 02	2.95E + 02
f3	Mean	1.64E + 01	9.37E + 00	1.61E + 01	1.53E + 01	1.96E + 01
	Std	3.72E − 01	5.56E − 02	1.43E + 00	1.041E − 01	3.88E − 02
f4	Mean	6.34E + 13	1.32E + 14	1.70E + 15	7.78E + 13	6.16E + 12
	Std	2.77E + 13	6.96E + 13	2.56E + 14	1.19E + 13	6.96E + 11
f5	Mean	5.87E + 08	2.51E + 08	6.30E + 08	3.94E + 08	4.22E + 08
	Std	8.68E + 07	5.12E + 07	1.84E + 07	6.93E + 07	2.84E + 07
f6	Mean	1.99E + 07	1.28E + 07	1.95E + 07	1.31E + 07	1.97E + 07
	Std	3.60E + 05	1.92E + 06	4.09E + 05	7.28E + 06	5.76E + 04
f7	Mean	2.70E + 11	3.91E + 10	1.35E + 11	3.85E + 10	3.43E + 08
	Std	1.51E + 11	6.54E + 09	1.42E + 10	9.09E + 09	1.82E + 08
f8	Mean	4.92E + 14	1.39E + 15	2.19E + 16	1.12E + 11	2.17E + 08
	Std	9.30E + 14	1.76E + 15	5.00E + 15	2.11E + 11	1.76E + 08
f9	Mean	7.71E + 10	5.31E + 10	1.78E + 11	6.99E + 10	6.33E + 08
	Std	1.25E + 10	6.91E + 09	1.49E + 10	7.14E + 09	1.35E + 08
f10	Mean	1.77E + 04	1.34E + 04	1.78E + 04	1.50E + 04	8.86E + 03
	Std	3.82E + 02	2.56E + 02	4.50E + 02	1.93E + 02	2.64E + 02

Table 6: Continued.

Functions	Criteria	PSO	GWO	SCA	SSA	MSPSO
f11	Mean	2.30E + 02	2.23E + 02	1.38E + 03	2.26E + 02	2.16E + 02
	Std	2.92E − 01	1.75E + 00	2.43E + 00	5.89E − 01	9.10E − 02
f12	Mean	6.10E + 06	3.72E + 06	1.98E + 07	6.45E + 06	7.48E + 05
	Std	9.86E + 05	1.82E + 05	2.17E + 06	4.43E + 05	5.44E + 05
f13	Mean	1.07E + 11	2.94E + 11	9.48E + 11	3.66E + 11	5.77E + 05
	Std	1.20E + 10	2.65E + 10	1.71E + 11	3.17E + 10	2.80E + 04
f14	Mean	8.97E + 10	4.38E + 10	1.75E + 11	7.61E + 10	1.66E + 09
	Std	9.52E + 09	4.19E + 09	8.56E + 09	5.94E + 09	1.36E + 07
f15	Mean	1.75E + 04	1.30E + 04	1.85E + 04	1.50E + 04	1.03E + 04
	Std	3.33E + 02	3.28E + 02	8.81E + 01	2.87E + 02	1.80E + 02
f16	Mean	4.20E + 02	4.13E + 02	2.23E + 03	4.14E + 02	3.93E + 02
	Std	1.05E + 00	2.52E + 00	5.40E + 00	0.54E + 00	5.33E − 01
f17	Mean	1.26E + 07	4.80E + 06	3.63E + 07	9.87E + 06	1.91E + 06
	Std	2.68E + 06	5.46E + 05	3.39E + 06	8.82E + 05	3.15E + 04
f18	Mean	5.90E + 11	8.38E + 11	1.17E + 12	1.27E + 12	2.78E + 06
	Std	5.56E + 10	2.84E + 10	3.58E + 09	5.23E + 10	9.23E + 05
f19	Mean	6.85E + 07	1.12E + 07	9.06E + 07	2.41E + 07	4.97E + 06
	Std	2.69E + 07	1.18E + 06	1.52E + 07	2.53E + 06	1.51E + 06
f20	Mean	6.94E + 11	1.01E + 12	8.86E + 10	1.50E + 12	2.80E + 06
	Std	5.88E + 10	5.11E + 10	5.58E + 09	5.69E + 10	3.15E + 05
Avg. ranking		3.55	2.35	4.55	3.10	1.45
Rank		4	2	5	3	1

5. Conclusions

The paper proposes a mixed-strategy PSO algorithm called MSPSO. MSPSO uses the rate of fitness change which measures the degree of improvement of successful offspring generated by each variant to guide selection probabilities of variants. Compared with previous PSO algorithms which use the number of previous successes and failures of each variant to adjust selection probabilities, MSPSO can increase the probability of selecting excellent variants. Furthermore, the proposed Nelder–Mead variant method is introduced in MSPSO to improve the exploitation ability. The proposed algorithm is tested on CEC 2014 benchmark suites with 10 and 30 variables. Experimental results demonstrate that MSPSO has a better overall performance than the other eight PSO algorithms on all problems in terms of the solution accuracy. MSPSO is also applied to three instances of the hydrothermal scheduling problem. Computational results show that the MSPSO algorithm also has a good performance in dealing with this real-world optimization problem. MSPSO is further tested on CEC 2010 with 1000 variables. The experimental results show that MSPSO has a good performance in solving large-scale optimization problems.

Our work shows a promising direction for designing efficient mixed strategy PSO algorithms; that is, the rate of fitness change guides the selection probabilities of variants. Thus, using the rate of fitness change to design other mixed strategy evolutionary algorithms will be left for testing as a future work.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

The short version of this study has been accepted by the 2021 IEEE International Conference on Space-Air-Ground Computing (SAGC 2021). We expanded that conference paper in this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200, the National Natural Science Foundation of China under Grant Nos. 61872105 and 62072136, and the Educational Science Planning of Heilongjiang Province under Grant Nos. ZJB1421113 and GJB1421251.

References

[1] X. Yang, S. Zheng, T. Zhou, Y. Liu, and X. Che, "Optimized relinearization algorithm of the multikey homomorphic encryption scheme," *Tsinghua Science and Technology*, vol. 27, no. 3, pp. 642–652, 2022.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November 1995.

[3] H. Wu, J. Liu, Z. Dong, and Y. Liu, "A hybrid mobile node localization algorithm based on adaptive MCB-PSO approach in wireless sensor networks," *Wireless Communications and*

Mobile Computing, vol. 2020, Article ID 3845407, 17 pages, 2020.

[4] Y. Wang, W. Peng, C. Qiu, J. Jiang, and S. R. Xia, "Fractional-order Darwinian PSO-based feature selection for media-adventitia border detection in intravascular ultrasound images," *Ultrasonics*, vol. 92, pp. 1–7, 2019.

[5] N. Sridevi, V. Nagarajan, and K. Sakthidasan, "Efficient traffic control and lifetime maximization in mobile ad hoc network by using PSO–BAT optimization," *Wireless Networks*, vol. 27, pp. 1–10, 2019.

[6] Y. Makhlouf and A. Daamouche, "Automatic generation of adaptive structuring elements for road identification in VHR images," *Expert Systems with Applications*, vol. 119, pp. 342–349, 2019.

[7] Y. Wang, Z. Cai, Z.-H. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 1033–1046, 2020.

[8] Z. Lu, Y. Wang, X. Tong, C. Mu, C. Yu, and Y. Li, "Data-driven many-objective Crowd worker selection for mobile crowdsourcing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 531–540, 2023.

[9] J. He, T. Chen, and X. Yao, "On the easiest and hardest fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 295–305, 2015.

[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 69–73, Anchorage, AK, USA, May 1998.

[11] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Information Sciences*, vol. 291, pp. 43–60, 2015.

[12] H. Zhang, M. Yuan, Y. Liang, and Q. Liao, "A novswarm optimization based on prey–predator relationship," *Applied Soft Computing*, vol. 68, pp. 202–218, 2018.

[13] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Lévy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.

[14] H. Garg, "A hybrid pso-ga algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.

[15] Q. Zhang, R. M. Ogren, and S. C. Kong, "A comparative study of biodiesel engine performance optimization using enhanced hybrid pso-ga and basic ga," *Applied Energy*, vol. 165, pp. 676–684, 2016.

[16] T.-S. Pan, T. K. Dao, and S. C. Chu, "Hybrid particle swarm optimization with bat algorithm," *Genetic and evolutionary computing*, pp. 37–47, Springer, Cham, Switzerland, 2015.

[17] G. Xu, Q. Cui, X. Shi et al., "Particle swarm optimization based on dimensional learning strategy," *Swarm and Evolutionary Computation*, vol. 45, pp. 33–51, 2019.

[18] X. Xu, Y. Tang, J. Li, C. Hua, and X. Guan, "Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy," *Applied Soft Computing*, vol. 29, pp. 169–183, 2015.

[19] J. He, W. Hou, and H. Dong, "Mixed strategy may outperform pure strategy: an initial study," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 562–569, Cancun, Mexico, June 2013.

[20] J. Pang, H. Dong, J. He, and R. Ding, "A modified comprehensive learning particle swarm optimizer," *International Journal of Performability Engineering*, vol. 15, no. 9, pp. 2553–2562, 2019.

[21] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 387–402, 2013.

[22] 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.

[23] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," *Proceedings of Swarm Intelligence Symposium SIS*, vol. 03, pp. 174–181, 2003.

[24] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[25] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," *Computational Intelligence Laboratory*, Zhengzhou University, Zhengzhou, China, 2013.

[26] 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.

[27] N. Lynn and P. N. Suganthan, "Ensemble particle swarm optimizer," *Applied Soft Computing*, vol. 55, pp. 533–548, 2017.

[28] H. Shi, S. Liu, H. Wu et al., "Oscillatory particle swarm optimizer," *Applied Soft Computing*, vol. 73, pp. 316–327, 2018.

[29] K. Chen, F. Zhou, Y. Wang, and L. Yin, "An ameliorated particle swarm optimizer for solving numerical optimization problems," *Applied Soft Computing*, vol. 73, pp. 482–496, 2018.

[30] 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.

[31] S. Das and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*, Jadavpur University, Kolkata, Inida, 2010.

[32] Y. Wang, H. X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, pp. 232–247, 2014.

[33] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[34] 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.

[35] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J. S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.

[36] T. Ke, X. Li, and P. N. Suganthan, "Benchmark functions for the cec' 2010 special session and competition on large-scale global optimization," *Nature Inspired Computation & Applications Laboratory*, vol. 17, 2009.

[37] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[38] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[39] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.