Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9599417, 19 pages
https://doi.org/10.1155/2022/9599417

Research Article

Subpopulation Particle Swarm Optimization with a Hybrid Mutation Strategy

Zixuan Xie, Xueyu Huang, and Wenwen Liu

JiangXi University of Science and Technology, Ganzhou 341000, China

Correspondence should be addressed to Zixuan Xie; 6720200821@mail.jxust.edu.cn and Xueyu Huang; 6120190561@mail.jxust.edu.cn

Received 13 December 2021; Revised 18 January 2022; Accepted 26 January 2022; Published 23 February 2022

Academic Editor: Radu-Emil Precup

With the large-scale optimization problems in the real world becoming more and more complex, they also require different optimization algorithms to keep pace with the times. Particle swarm optimization algorithm is a good tool that has been proved to deal with various optimization problems. Conventional particle swarm optimization algorithms learn from two particles, namely, the best position of the current particle and the best position of all particles. This particle swarm optimization algorithm is simple to implement, simple, and easy to understand, but it has a fatal defect. It is hard to find the global optimal solution quickly and accurately. In order to deal with these defects of standard particle swarm optimization, this paper proposes a particle swarm optimization algorithm (SHMPSO) based on the hybrid strategy of seed swarm optimization (using codes available from https://gitee.com/mr-xie123234/code/tree/master/). In SHMPSO, a subpopulation coevolution particle swarm optimization algorithm is adopted. In SHMPSO, an elastic candidate-based strategy is used to find a candidate and realize information sharing and coevolution among populations. The mean dimension learning strategy can be used to make the population converge faster and improve the solution accuracy of SHMPSO. Twenty-one benchmark functions and six industries-recognized particle swarm optimization variants are used to verify the advantages of SHMPSO. The experimental results show that SHMPSO has good convergence speed and good robustness and can obtain high-precision solutions.

1. Introduction

So far, optimization algorithm has been a popular research problem. Particle swarm optimization algorithm is a population-based optimization algorithm, which was first invented by Dr. Eberhart and Dr. Kennedy in 1995 [1, 2]. Particle swarm optimization algorithm is inspired by the behavior of birds looking for food. Particle swarm optimization is a search algorithm in which particles cooperate with each other. Of course, particle swarm optimization algorithm is a population intelligent optimization algorithm. In addition to particle swarm optimization algorithm, common intelligent algorithms include differential evolution algorithm (DE) [3], ant colony optimization (ACO) algorithm [4], artificial bee colony (ABC) [5], programming algorithm (FEP) [6], simulated annealing algorithm [7], neural network [8], text clustering [9], resource allocation [10], and task allocation [11]. Particle swarm optimization algorithm has been widely accepted with the advantages of rapid convergence, excellent robustness, and concise understanding.

At present, many optimization algorithms have achieved significant results in many fields. It includes many applications, such as optimal tuning of type-1 and type-2 fuzzy controllers [12], optimal tuning of interval type-2 fuzzy controllers [13], scheduling planning [14], undergraduate systems engineering curriculum optimization technology [15], improving the performance of FinFET devices [16], brain models of fear processing and conflict modulation [17], multiobjective dynamic optimization problem [18, 19], mixed-variable Newsvendor problems [20], and feature selection [21].

With the advantages of particle swarm optimization algorithm [22], many applications use particle swarm optimization algorithm, such as image processing [23], neural network [24], feature selection [25], data clustering [26], and mixed-variable optimization problems (MVOPs) [27].

Although particle swarm optimization algorithm has been preeminently optimized and improved, it still has some problems to be solved. In particular, when the dimension of the population becomes higher, the particle swarm optimization algorithm is extremely prone to early convergence, which makes it hard to jump out of the local optimal solution in the final stage of the algorithm. In order to continue to improve the performance of particle swarm optimization algorithm, countless particle swarm optimization researchers are committed to four improvement directions, namely, particle swarm optimization parameter adjustment, learning strategy improvement, topology selection, and integration with other algorithms. These four aspects are described as follows:

Parameter control: it includes setting inertia weight, acceleration coefficient, and population size. The particle swarm optimization algorithm with constant inertia weight balances the ability of global development and local exploration. This method limits the particle swarm optimization algorithm in a sense. Another adaptive control parameter is proposed to control local balance and global exploration through adaptive parameters. Shi and Eberhart et al. proposed a PSO algorithm with an inertia weight set to a constant value to balance the global exploration and local exploitation abilities of the algorithm [28]. The inertia weight is constant, which greatly limits the potential search ability of particle swarm optimization. Zhan et al. proposed an APSO algorithm with adaptive control parameters. APSO achieves a locally and globally stable search state through adaptive control parameters [29].

Learning strategy improvement: it includes some comprehensive learning strategies, biogeography-based learning strategies, segmentation based advantage learning strategies, domain based learning strategies, and dynamic domain learning strategies. Yang et al. proposed the SPLSO algorithm that used segment-based predominant learning. This model first segments the dimensions of each poor-performing particle; then, each segment is learned from a better-performing particle, allowing the algorithm to capitalize on the information from the better particles and avoid premature convergence [30]. Liang and Suganthan proposed a DMS-PSO algorithm with a dynamic neighborhood structure in which the learning of each particle is no longer limited to one population; instead, it also includes other populations [31].

Topology: topology can effectively use particle information in the field, but it ignores the global optimal information to a certain extent. Common topologies include ring topology, star topology, network topology, dynamic tree topology, and dynamic competition topology. Li et al. proposed an adaptive particle swarm optimization algorithm using scale-free network topology. Based on the characteristics of scale-free network topology with a power-law distribution, the algorithm can construct a corresponding neighborhood for each particle [32]. Janson et al. constructed a dynamically changing tree topology in which each particle learns from its parent to utilize the information of each particle effectively [33].

Hybrid algorithm integration: mixing with other algorithms is one of the main research fields of particle swarm optimization algorithm, which can effectively improve the performance of the algorithm. Zhang et al. proposed the DSPSO algorithm by combining the differential mutation operation with the SLPSO algorithm [34]. Valdez et al. proposed a new hybrid approach for optimization by combining particle swarm optimization (PSO) and genetic algorithms (GAs) using fuzzy logic to integrate the results [35].

According to the above analysis, an excellent particle swarm optimization algorithm is to have better local exploration and global exploration capabilities. To achieve both, the diversity of population is essential, which can prevent the premature convergence of particle swarm optimization algorithm. Because each optimization problem is different, it is difficult for a single evolutionary strategy to meet each optimization problem. Relevant literature has shown that the combination of several strategies helps to improve the possibility of particle swarm convergence to global optimization. In order to adapt to more optimization problems, inspired by the MPCPSO algorithm, this paper proposed a joint strategy of subpopulation cooperative particle swarm optimization algorithm. The particle swarm is initialized, and the fitness values of all particles for the first time are recorded [36]. According to the fitness values, the population is divided into two populations: dominant population (DP) and poor population (PP). For the two populations, the first population and the second population adopt different evolutionary strategies, respectively, the second population adopts candidate learning strategy, and the first population adopts mean dimension learning. By comparing with other algorithms, the effect of the two strategies in this paper is feasible, and the solution of the function can be obtained at the same time.

The first section of this paper is organized as follows: Section 2 mainly introduces the related work of writing this article. Section 3 introduces the two learning strategies and SHMPSO in detail. In Section 4, the SHMPSO algorithm is tested by using twenty-one benchmark functions and six famous PSO variants. At the end of the article, Section 5 gives the relevant conclusions.

2. Related Work

This part mainly introduces some harvest before completing the experiment. Inspired by two particle swarm optimization algorithms, the first is the classical particle swarm optimization algorithm, and the second is the particle swarm optimization algorithm based on biogeography. Basically, all PSO variants are improved on the classic PSO.

2.1. Classic PSO. The classical particle swarm optimization algorithm is an optimization algorithm based on the whole population. Each particle represents the possibility of a solution. All particles update their positions according to their own historical best positions and global historical best positions. Generally, there are D dimensions in the search space of the whole particle swarm. The position vector of particle i at a certain time t is $X_i = (x_{i1}, x_{i2}, \ldots x_{i\,D})$, and the velocity vector is $V_i = (v_{i1}, v_{i2}, \ldots v_{i\,D})$. All particles of the next-generation population can be generated according to the position vector and velocity vector. The update equation for generating optimized next-generation particles is

$$\begin{aligned} V_{i\,d} &= \omega V_{i\,d}(t) + c_1 r_1 \left(P_{i\,d}(t) - X_{i\,d}(t)\right) \\ &\quad + c_2 r_2 \Big(P_{g\,d}(t) - X_{i\,d}(t)\Big), \end{aligned} \tag{1}$$

$$X_{i\,d}(t+1) = X_{i\,d}(t) + V_{i\,d}(t+1), \tag{2}$$

where ω is an inertia weight, c_1 and c_2 are an acceleration factor, and r_1 and r_2 are a random number between [0, 1]. These two equations are the core equations of standard particle swarm optimization. The equation for speed and position is updated.

Although particle swarm optimization algorithm has made many improvements in recent decades, there is still an algorithmic barrier that is hard to breakthrough. For example, the algorithm is difficult to find the extreme value in the case of high dimension, and it is simple to fall into local optimization. At present, many strategies have been proposed to solve these difficulties. The following introduces the comprehensive learning strategies and biogeography learning strategies.

2.2. Particle Swarm Optimization Algorithm with the Comprehensive Learning Strategy. Liang et al. proposed a CLPSO algorithm, which is different from the conventional particle swarm optimization algorithm introduced earlier [37]. CLPSO never learns from the previously mentioned global optimal particle, but each dimension of the particle is like the historical optimal learning of the sample constructed in its field, and the learning probability is P_c. CLPSO canonical learning scheme avoids falling into local optimization in some multimodal problems. The revised scheme proposed by CLPSO is described as follows:

$$V_{i\,d}(t+1) = \omega V_{i\,d}(t) + cr\Big(P_{f_i(d)}(t) - X_{i\,d}(t)\Big), \tag{3}$$

where ω is the inertia weight, c is an acceleration coefficient, and r is a random number between (0, 1). $f_i(d)$ represents that in the D dimension, the particle i changes from $f_i(d)$ in $p$best that is an optimum position of particle history and $f_i(d) = [f_i(1), f_i(2), \ldots, f_i(D)]$. All sample vectors of particle i are defined in $f_i(d)$. In CLPSO, equation (3) shows that each different particle can learn from different particles from different dimensions. The main core methods of CLPSO are shown in the following flow chart:

(1) Generate random number P between (0, 1), and judge the size of p and P_c.

(2) If $P > P_c$, the current particle i updates according to its optimal position.

(3) If $P < P_c$, the optimal particle is selected by comparing the fitness values of all particles. Let the particle replace the personal optimal position of the particle i and guide particle i to update.

The algorithm employs a comprehensive learning strategy whereby the best position before other particles is a paradigm that can be learned by any particle, and each dimension of a particle has the potential to learn from a different paradigm. The new strategy allows particles to have more learning paradigms and a larger potential flight space.

2.3. Learning Particle Swarm Optimization Algorithm Based on Biogeography (BLPSO). BLPSO is improved according to CLPSO, which is based on biogeographic migration. BLPSO proposes a new learning strategy particle swarm optimization algorithm based on biogeography [38]. All particles in the population are updated by biogeographic migration using their own optimal location and the optimal location combination of other particles. All particles have a migration in and migration out rate, respectively, α_i and β_i to represent:

(1) Sort each particle, and calculate the migration rate of each particle after sorting α_i and β_i.

(2) Generate the index of the sample vector, $f_i(d) = [f_i(1), f_i(2), \ldots, f_i(D)]$.

(3) Generate a random number r. When $r < \alpha_i$, the index j sum of a particle and its migration rate is selected by the roulette selection probabilities β_j. Assign j to f_i. Otherwise, assign i to f_i.

(4) If f_i is equal to particle i, randomly select a particle j that is not equal to particle i, randomly select a dimension l, and assign the dimension of j to $f_i(l)$.

Contrastingly, the biogeography-based learning strategy employs a ranking technique whereby particles can learn more from particles with high-quality personal best positions, and this effectively enhances the exploitation of the original CLPSO.

3. Particle Swarm Optimization Algorithm with the Hybrid Strategy of Seed Swarm Optimization

The third section mainly introduces the particle swarm optimization algorithm based on the hybrid strategy of seed swarm optimization. Section 3.1 mainly introduces the learning strategies of the mean dimension. Section 3.2 mainly introduces the candidate generation strategy based on elasticity. The overall operation framework of this strategy will be given in Section 3.2. At the end of this section, the whole process of SHMPSO is shown in the form of pseudo code.

3.1. Learning Strategy of Mean Dimension. The particle swarm optimization algorithm for subpopulation has been around for a long time, and it has received excellent feedback on some issues. The CMPSODMO proposed by Liu et al. in

2017 handles multiobjective dynamic optimization problems in a complex and changing environment and uses a multiswarm-based particle swarm optimization framework to optimize problems in a dynamic environment [39]. The algorithm has achieved excellent results. In the FTPSO, proposed by Yazdani et al. in 2013, it was clearly proposed that in the algorithm, the advantages of multiple groups should be used to cover multiple peaks of the multiobjective problem in a dynamic space [40]. Inspired by TSLPSO, it was proposed by Xu et al. in 2019 [41]. TSLPSO proposes to adopt two populations and uses different strategies to iterate the two populations in the search space, respectively, and obtain significant results. Yen and Daneshyari proposed a method to exchange information among multiple swarms [42].

In the SHMPSO algorithm proposed in this paper, firstly, a population is initialized, and the population is divided into two populations by using the fitness ranking mechanism. The ranking order is ascending. In each iteration process of the algorithm, the part of particles with smaller fitness values is divided into the dominant population, and the other part is classified as the poor population. The proportional coefficient of the dominant population particle is *s*. The dominant populations use the mean dimension learning strategy to optimize the population, and the poor population uses the candidate generation strategy based on elasticity to learn. In this way, different evolutionary strategies for different populations can effectively ensure the diversity of populations and avoid falling into local optimization. The dominant population can guide the search direction of the whole population and make the population further converge to the solution quickly. There is also a strange phenomenon in PSO algorithm, which is called "spiral rise," that is, the phenomenon of "two steps forward and one step backward" [43]. This means that although the fitness of the particles has been improved, the effects of the minority components of the particles have worsened. In order to overcome this problem, it is inevitable that the evaluation function needs to be changed frequently. Basically, all variants of PSO algorithm have some phenomena that the convergence speed is slow when the whole algorithm runs to the later stage. In SHMPSO, all particles are learning from the dominant population to achieve faster convergence speed.

Classical PSO requires two guiding particles, which are the current particle historical best value and the global historical best value. However, there are many algorithms that only have one guiding particle, such as the CLPSO and BLPSO mentioned above (Section 2). The advantage of this is that the speed update formula has fewer parameters and is easy to understand. The difficulty lies in how to construct this guide particle. For the dominant population, this article uses a guide particle. The following is an introduction to the evolution strategy of the dominant population.

For the whole dominant group, when the particle velocity is updated, the current particle will only receive the influence of $m$best generated by the comprehensive dimension learning.

The velocity update equation of particles is

$$\mathrm{V}_{i\,d}(\mathrm{t}+1) = \omega \mathrm{V}_{i\,d}(\mathrm{t}) + cr\left(m\,\mathrm{best}_{i\,d}(t) - X_{i\,d}(t)\right), \tag{4}$$

where ω, c, and r have the same meanings as in equation (3). Inspired by MPCPSO, $m\,\mathrm{best}_{i\,d}$ is obtained from equation (5). Its purpose is to help particles get rid of the local optimal state. As can be seen from equation (9), $m\,\mathrm{best}_{i\,d}$ is regarded as the only learning paradigm used to guide particle motion. Suppose the algorithm falls into the local extremum, $m\,\mathrm{best}_{i\,d}$ seldom actively help particles find better solutions. In mean dimension learning, each particle learns not only from other particles but also from other related dimensions, which greatly increases the universality of particle domain learning.

$$m\,\mathrm{best}_{i\,d} = \frac{\rho r}{D}\sum_{d=1}^{D} X_{i\,d} + \frac{(1-\rho)(1-r)}{N}\sum_{i=1}^{N} X_{i\,d}, \tag{5}$$

where D represents the dimension of particles, r is a random number between [0, 1], and N represents the total number of particles in the dominant population. Among them, ρ is a dynamically defined value. The specific solution is as follows:

$$\rho = \frac{1}{1 + \exp\left(-\left(X_{i\,d} - 1/D\sum_{d=1}^{D} X_{i\,d}\right)\right)}. \tag{6}$$

The location update equation is as follows:

$$X_{i\,d}(t+1) = \phi X_{i\,d}(t) + V_{i\,d}(t+1), \tag{7}$$

where φ is a dynamic parameter. Its equation (8) is given below:

$$\varphi = \frac{1}{\left(1 + \exp\left(-ave_1/N_1\sum_{i=1}^{N} f(x_i)\right)\right)^{\mathrm{iter}}}, \tag{8}$$

where N_1 is the current population size (the dominant population), ave_1 refers to the average fitness of the dominant population after the first iteration of the population, and iter represents the current number of iterations. Based on the above iterative updating, the dominant population can be updated in a good direction.

It can be seen from equation (7) that the convergence speed of particles of the dominant population will be accelerated to a great extent. With the acceleration of convergence speed, it is inevitable that mean dimension learning is simple to fall into local optimization. In order to avoid this situation, a differential mutation operator is introduced to increase the diversity of the population [44, 45]. Here, we need an operation to randomly select two particles from the dominant population. The values of these two particles cannot be the same as those of the current iteration. At this time, we need to calculate the differential of these two random particles and make them a differential vector. This differential vector also needs to be mutated with the scaling factor of F. After mutation, it is summed with the global optimum ($P_{g\,d}$). The whole operation is described by the following equation:

$$X_{i\,d}(\mathrm{t}+1) = P_{g\,d}(t) + F\left(X_a(t) - X_b(t)\right), \tag{9}$$

where $P_{g\,d}$ is the global optimal position of the current population, F is a mutation coefficient. a and b represent the index of randomly selected particles and meet the condition requirements $i \neq a \neq b$. The equation of mutation operation not only improves the search ability of the algorithm in the dominant particle swarm optimization but also expands the diversity in the process of population search. Algorithm 1 lists the pseudo code of mean dimensions learning.

3.2. Candidate Generation Strategy Based on Elasticity. Next, the strategy of poor population is the candidate generation strategy based on elasticity. SHMPSO uses different populations and adopts different evolutionary strategies to evolve, respectively. This strategy is conducive to the rapid convergence of the population without losing the diversity of the population. The most important thing about this is how to generate candidates, which is the core part of the whole strategy. Here, the speed update equation has changed, instead of learning from the global optimization ($P_{g\,d}$) like the traditional particle swarm optimization algorithm, but introducing candidates and learning from candidates. New speed update equation is as follows:

$$\begin{aligned} V_{i\,d}(t+1) &= \omega V_{i\,d}(t) + c_1 r_1 \left(p\text{best}\,2_{i\,d}(t) - X_{i\,d}(t)\right) \\ &\quad + c_2 r_2 \left(\text{Candidate}_{i\,d}(t) - X_{i\,d}(t)\right), \end{aligned} \tag{10}$$

where ω, c_1, c_2, r_1, and r_2 have the same meanings as in equation (1). The vector $p\text{best}\,2_{i\,d}$ represents the historical optimum of the current particle in the poor population. The generation method of $\text{Candidate}_{i\,d}$ is given below. Inspired by the elastic force generated by spring compression, an elastic coefficient prob is introduced here. The elastic coefficient of the poor population particle is set as prob = 0.5. This parameter size is set by the user. Like a spring, it can be stretched or compressed. Particles are sorted in ascending order according to the fitness value, the larger the value, the worse the performance of the particles. A random number r between [0, 1] is randomly generated. If the elastic coefficient *prob* is greater than r, then the $\text{Candidate}_{i\,d}$ is generated by the equation:

$$\text{Candidate}_{id} = g\,\text{best}\,1_{id}.*\text{step zize}_{id}*N, \tag{11}$$

where the vector of $g\,\text{best}\,1_{id}$ represents the global optimal position of the dominant population. N represents the number of particles. The vector of $\text{step zize}_{i\,d}$ is a D-dimensional vector obtained by equation (17) and generated by Levy flight. An introduction to Levy flight is given below. The generation of Levy flight random number includes two parts. The first part is the selection of random direction, and the second part is the generation of Levy distribution. Random walks are derived from Levy stability. This distribution is a simple power-law equation:

$$L(s) \sim |s|^{-1-\beta}, \tag{12}$$

where $0 < \beta < 2$. It is an index.

Definition 1. To determine Levy distribution mathematically, it defined by the following equation:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\dfrac{\gamma}{2\pi}}\exp\left[-\dfrac{\gamma}{2(s-\mu)}\right]\dfrac{1}{(s-\mu)^{3/2}}, & \text{if } 0 < \mu < s < \infty, \\ 0, & \text{if } s \leq 0, \end{cases} \tag{13}$$

where the parameter μ is a control displacement parameter or position, $\gamma > 0$ represents the scale parameter (the scale used to control the distribution).

Definition 2. Usually, Levy distribution is defined by Fourier transform. The specific equation is given in the following:

$$\mathrm{F(k)} = \exp\left[-\alpha|\mathrm{k}|^{\beta}\right], 0 < \beta \leq 2, \tag{14}$$

where α is a parameter in the interval [−1, 1], which is usually called skewness or scale factor. Stability index β is controlled between (0, 2), which is also commonly referred to as Levy index. β in most cases, his analytical form of integration is unknown. For random walking, the step S can be calculated by Mantegna's equation:

$$S = \frac{u}{|v|^{1/\beta}}, \tag{15}$$

where u and v in equation (13) obey a positive distribution:

$$\begin{gathered} u \sim N\left(0,\sigma_u^2\right), v \sim N\left(0,\sigma_u^2\right), \\ \sigma_u = \left\{\frac{\tau(1+\beta)\sin(\pi\beta/2)}{\tau[1+\beta/2]\beta 2^{(\beta-1/2)}}\right\}^{1/\beta}. \end{gathered} \tag{16}$$

The step size can then be calculated by the following equation:

$$\text{step size} = 0.01 \times S, \tag{17}$$

where the factor 0.01 comes from the typical step factor $L/100$, and L is a typical length ratio. Otherwise, Levy flight will become too radical and jump out of the design plan (waste evaluation).

When the elasticity factor prob < r, in order to ensure the diversity of the population, two particles m, n are selected from the dominant population. These two particles and the global optimal particle $P_{g\,d}$ are required which is different from each other. Compare particle $X_{m\,d}$ and $X_{n\,d}$ fitness value, and select the particle with a smaller fitness value $X_{n\,d}$. $\text{Candidate}_{i\,d}$ is obtained by the equation:

$$\text{Candidate}_{i\,d} = X_{n\,d}.*\text{step zize}_{i\,d}*N, \tag{18}$$

where $\text{stepzize}_{i\,d}$ and N have the same meanings as in (12). The candidate improves the diversity of the population in this way. Algorithm 2 lists the pseudo code of the candidate generation strategy based on elasticity.

3.3. Overall Framework of the SHMPSO Algorithm. Based on the improved strategies of 3.1 and 3.2, SHMPSO is constructed. The specific steps of the SHMPSO algorithm are as follows:

```
Input: elitist population PP
Output: position vector $X_i$, velocity vector $V_i$
for $i = 1 \longrightarrow$ N1 do
    Calculate the mean of $X_i \longrightarrow$ X_mean$_i$
    Calculate the mean of $X_d \longrightarrow$D_mean$^d$
    Calculate $m$best by equation (5)
    if $M > 6$ then %$M$ is the number of the algorithm falls into the local optimum
        Update $X_i$, $V_i$ by equations (4) and (9)
    else
        Update $X_i$, $V_i$ by equations (4) and (7)
    end if
end for
```

Algorithm 1: Mean learning strategy.

```
Input: dominant population DP, poor population PP
Output: position vector $X_i$, velocity vector $V_i$
for $i = 1 \longrightarrow$ N2 do
    if prob$_i$ > rand() % prob is an elastic factor
            Update Candidate$_{i\,d}$ by equation (11)
    else
        if prob$_i$ < rand()
            Select two particles $X_{m\,d}$, $X_{n\,d}$ randomly and calculate their fitness values
            Choose particles with less fitness $X_{i\,d}$; % it is different from $P_{g\,d}$
    end if
            Update Candidate$_{i\,d}$ by equation (18)
    end if
    Update $X_i$, $V_i$ by equations (2) and (10)
end for
```

Algorithm 2: Strategy of candidate generation based on elasticity.

Step 1: initialize the population and set the parameters, the mutation factor $F = 0.5$, the population proportion scale factor $s = 0.5$, and the times of falling into local optimum $M = 6$.

Step 2: calculate the fitness value $f(X_i)$ and the global optimal value ($P_{g\,d}$) of all particles.

Step 3: sorting in ascending order according to the fitness value, taking $s * N$ (all particles) particles as the dominant population (DP) and the rest particles as the poor population (PP)

Step 4: optimizing disadvantaged populations based on flexible candidate strategy, which uses equations (2) and (10). The mean dimension learning is used to optimize the DP subpopulation through equations (4) and (7).

Step 5: when $M > 6$, equation (9) is used to update the position of particles in the dominant population (DP).

Step 6: recalculate the fitness values of all particles, and update the current global optimal value ($P_{g\,d}$).

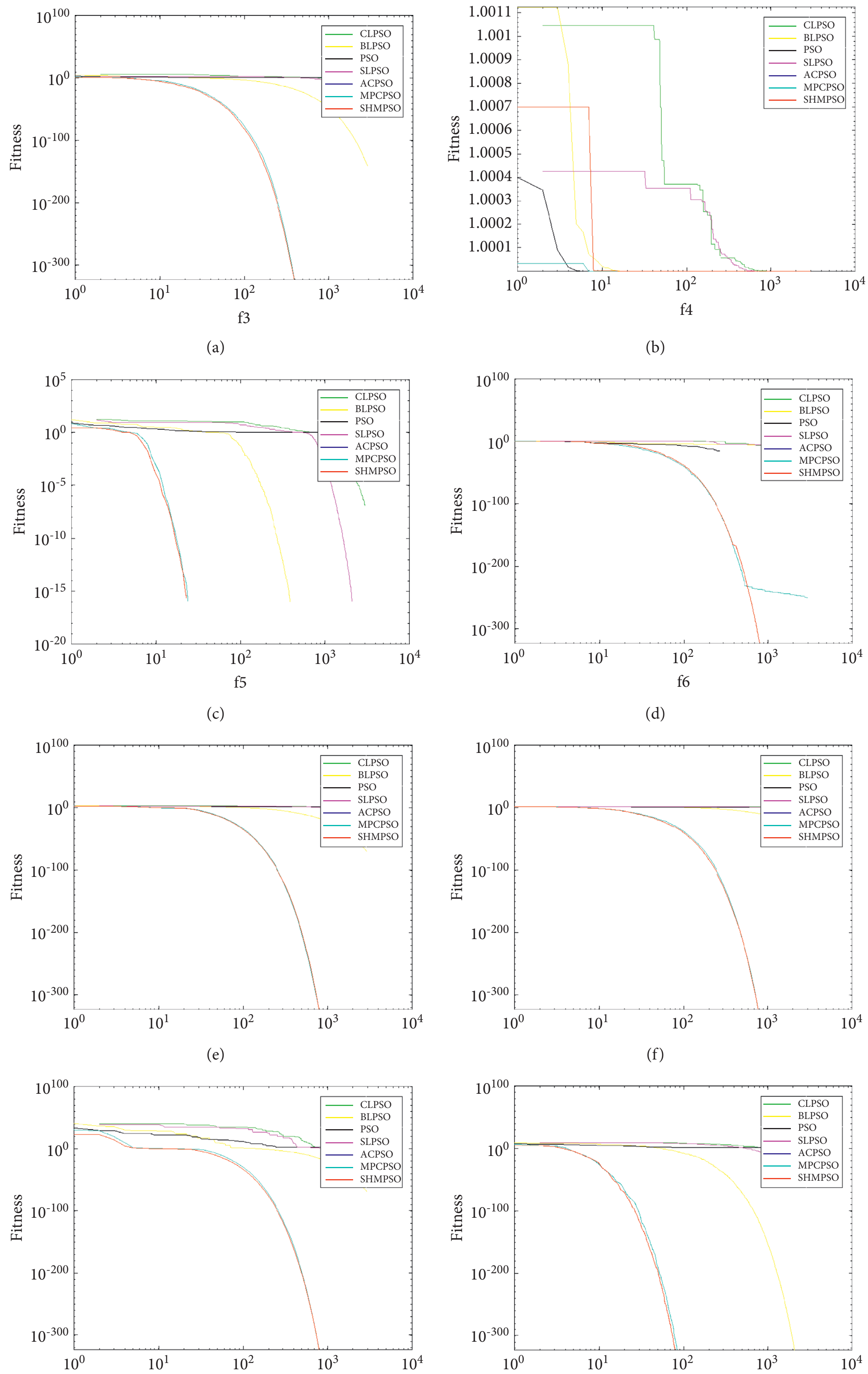
Step 7: repeat steps 3–6 when the maximum allowed times of iteration is bigger than the times of the maximum number of iterations.

The algorithm flow chart of SHMPSO is shown in Figure 1. This algorithm mainly uses the global search ability to find a better search space. The algorithm starts from the global situation, finds the current global optimal position, and can converge to the global optimal solution faster. Thirdly, the general framework of the algorithm is given in Algorithm 3.

4. Experiment

In this section, in order to verify the reliability and efficiency of the proposed SHMPSO, twenty-one widely used benchmark functions are adopted. Comparing SHMPSO with other varieties of PSO, the results are verified. The experiment process is as follows.

4.1. Benchmark Function and Parameter Setting. The twenty-one benchmark functions listed in Table 1 are used to demonstrate the superiority of SHMPSO. In Table 1, the first column represents the function number, the second column represents the function name, the third column represents the function mathematical expression, the fourth column represents the function search range, and the fifth column



(a)

(b)

(c)

(d)

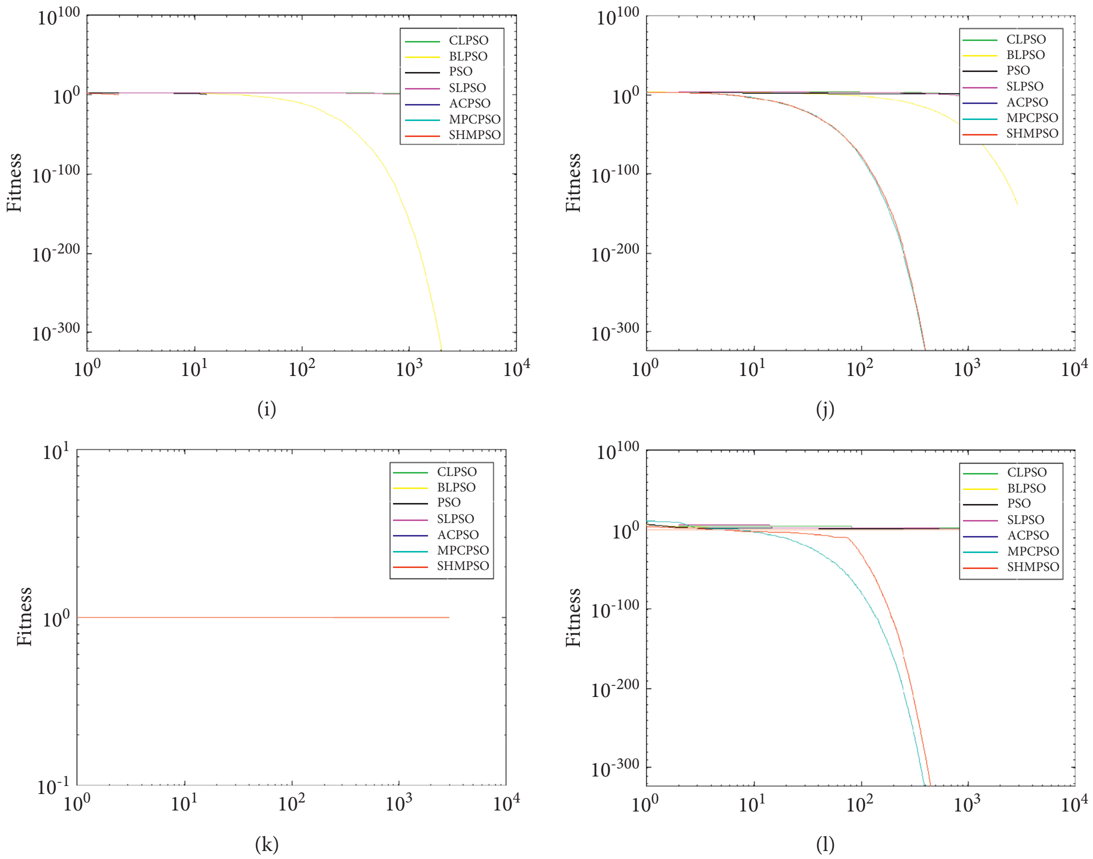
(e)

(f)

(g)

(h)

Figure 1: Continued.



(i)

(j)

(k)

(l)

Figure 1: Convergence curves of 1–12 functions. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_5. (f) f_6. (g) f_7. (h) f_8. (i) f_9. (j) f_{10}. (k) f_{11}. (l) f_{12}.

```
Initialization: particle's position $X_i$, velocity $V_i$, $F = 0.5$, $M = 6$
Calculate particle's fitness value $f(X_i)$, $P_{g\,d}$, fit = $P_{g\,d}$
  while iter < max gen do
      Plan the number of DP, PP; N1 = $s \times N$, N2 = $N -$ N1;
    Sorting population based on fitness values into DP, PP;
    for $i$ = 1 ⟶ N2 do
      Update Candidate$_{i\,d}$ by equations (12) and (18)
      Update $X_i$, $V_i$ by equations (2) and (10)
    end for
    for $i$ = 1 ⟶ N1 do
    Calculate $m$best by equation (5)
    if $M > 6$ then
      Update $X_i$, $V_i$ by equations (4) and (9)
    else
      Update $X_i$, $V_i$ by equations (4) and (7)
    end if
  end for
  Calculate new particle's fitness value $f(X_i)$
  Update $P_{g\,d}$
  if $P_{g\,d}$ < fit then
    fit = $P_{g\,d}$, $M = 0$
  else
    $M = M + 1$
  end if
end while
```

Algorithm 3: Grouping-mixed-based particle swarm optimization algorithm.

Table 1: Twenty-one benchmark functions.

Function	Function name	Test function	Search range	$f_{\min}$
f_1	Brown function	$\sum_{i=1}^{n-1}(x_i^2)^{(x_{i+1}^2+1)}+(x_{i+1}^2)^{(x_i^2+1)}$	[−1, 4]	0
f_2	Exponential function	$-\exp(-0.5\sum_{i=1}^{n}x_i^2)$	[−1, 2]	0
f_3	Griewank function	$1+\sum_{i=1}^{n}x_i^2/4000-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})$	[−600, 600]	0
f_4	Ridge function	$x_i+d*(\sum_{i=1}^{n}x_i^2)^{\alpha}, d=1, \alpha=0.5$	[−5, 5]	0
f_5	Schwefel2.20 function	$\sum_{i=1}^{n}\lvert x_i\rvert$	[−100, 100]	0
f_6	Schwefel2.21 function	$\max_{i=1,\dots,n}\lvert x_i\rvert$	[−100, 100]	0
f_7	Schwefel2.22 function	$\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[−100, 100]	0
f_8	Schwefel2.23 function	$\sum_{i=1}^{n}x_i^{10}$	[−10, 10]	0
f_9	Sphere	$\sum_{i=1}^{n}x_i^2$	[−5.12, 5.12]	0
f_{10}	Sum squares	$\sum_{i=1}^{n}ix_i^2$	[−10, 10]	0
f_{11}	Xin-she Yang N.3	$\exp(-\sum_{i=1}^{n}(x_i/\beta)^{2m})-2\exp(-\sum_{i=1}^{n}x_i^2)\prod_{i=1}^{n}\cos^2(x_i)$	[−2 π, 2 π]	0
f_{12}	Zakharou	$\sum_{i=1}^{n}x_i^2+(0.5*\sum_{i=1}^{n}ix_i)^2$	[−5, 10]	0
f_{13}	Ackley 1	$20+\exp(1)-20*\exp(0.2\sqrt{\sum_{i=1}^{n}x_i^2/n})-\exp[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)]$	[−35, 35]	0
f_{14}	Alphine 1	$\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	[0, 10]	0
f_{15}	Happy cat	$(\sum_{i=1}^{n}x_i^2-\mathrm{n})^{\alpha}+\frac{1}{n}(0.2\sum_{i=1}^{n}x_i^2+\sum_{i=1}^{n}x_i)+0.5,\ \alpha=0.5$	[−2, 2]	0
f_{16}	Periodic	$1+\sum_{i=1}^{n}\sin^2(x_i)-0.1\exp(-\sum_{i=1}^{n}x_i^2)$	[−10, 10]	0
f_{17}	Rastrign	$10\mathrm{n}+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$	[−5.12, 5.12]	0
f_{18}	Xin-she Yang 2	$(\sum_{i=1}^{n}\lvert x_i\rvert^{i})*\exp[-\sum_{i=1}^{n}\sin(x_i^2)]$	[−2 π, 2 π]	0
f_{19}	Schwefel 1.2	$\sum_{i=1}^{n}(x_1+x_2+\cdots+x_n)^2$	[−100, 100]	0
f_{20}	Step2	$\sum_{i=1}^{n}x_i+0.5^2$	[−100, 100]	0
f_{21}	Penalized2	$((\sin^2(3\pi x_i)+\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+(x_n-1)^2)*[1+\sin^2(2\pi x_n)])*0.1+\sum_{i=1}^{n}100(a_i-5)^4,\ a_i=\begin{cases}x_i, x_i>5\\-x_i, x_i<-5\end{cases}$	[−50, 50]	0

Table 2: Parameter settings of different PSO variants.

Algorithm	Parameter setting
PSO	$\omega = 0.729, \mathrm{c1} = \mathrm{c2} = 1.49445, V_{\max} = 0.2 \times \mathrm{range}$
ACPSO	$\omega = 0.9 \sim 0.4, \mathrm{c1} = \mathrm{c2} = 1.49445, V_{\max} = 0.2 \times \mathrm{range}, \mathrm{alpha} = 0.1; \mathrm{beta} = 0.1$
SLPSO	$m = \mathrm{M} + \mathrm{floor}\,(D/100), \mathrm{c} = D/M * 0.01, \mathrm{M} = 100$
CLPSO	$\omega = 0.9 \sim 0.2, \mathrm{gamp} = 5, \mathrm{c} = 1.49445, V_{\max} = 0.2 \times \mathrm{range}$
BLPSO	$\omega = 0.9 \sim 0.2, \mathrm{gamp} = 5, \mathrm{I} = E = 1, c = 1.49445, = 0.2 \times \mathrm{range}$
MPCPSO	$\omega = 0.729, \mathrm{c1} = \mathrm{c2} = 1.49445, F = 0.5, t = 0.5, V_{\max} = 0.2 \times \mathrm{range}$
SHMPSO	$\omega = 0.729, \mathrm{c1} = \mathrm{c2} = 1.49445, F = 0.5, s = 0.5, \mathrm{prob} = 0.5, V_{\max} = 0.2 \times \mathrm{range}$

shows the minimum value of the function. The tested functions include 11 unimodal functions ($f_1 - f_{2,} f_4 - f_{12}$) and 10 multimodal functions ($f_3, f_{13} - f_{21}$). The optimal value of all benchmark functions tested is 0. In order to compare the superior performance of the algorithm, six PSO variants are selected, including inertia weight PSO [40], ACPSO [46], SLPSO [47], CLPSO [37], BLPSO [38], and MPCPSO [36].

4.2. Selection of the Population Proportion Coefficient. In order to get a fairer comparison result, all the parameters used in the data experiment are the same, including the maximum number of independent runs (runNumber), the number of evaluations (maxFEs), and the maximum number of iterations (maxgen) where N represents the number of all particles in the population. All the comparison algorithms tested the twenty-one benchmark functions and got the mean and standard deviation after 30 runs. The parameter settings of all PSO variants are given in Table 2. According to the experimental results, all the functions are ranked and compared. In order to test the proportion of the dominant population in the SHMPSO algorithm, parameter c is tested, and the experimental results are as follows.

Through the overall analysis of Table 3 and ranking of each function, it is concluded that when $s = 0.5$, when the proportion of the dominant population in this paper is kept at 50%, the population optimization effect is the best. All the following comparisons are the experimental results based on the population scale coefficient of 0.5.

4.3. Comparison of Experimental Data between SHMPSO and Other PSO Variants. At the same time, 30, 50, and 100 dimensions are used to evaluate the given test function. The total population of SLPSO has its own definition, and all other functional population sizes are tested at ($N = 100$). When the particle dimension of all populations is 30 dimensions, maxgen is set to 3 × 10ˆ3, and the function evaluation times maxFEs is set to 3 × 10ˆ5. When the dimension of the particle is 50 dimensions, maxgen is set to 5 × 10ˆ3, and the function evaluation times maxFEs is set to 5 × 10ˆ5. When the particle dimension is 100 dimensions, the population's maxgen is set to 1 × 10ˆ4, and the function evaluation times maxFEs is set to 1 × 10ˆ6.

It can be seen from Tables 4 and 5 that the performance tested by the proposed SHMPSO is relatively stable in 30 and 50 dimensions, and the results obtained are similar. It can be seen that the performance of SHMPSO is excellent in the 30-dimensional and 50-dimensional convergence processes. SHMPSO is in f_1, f_3, $f_5 - f_{10}$, f_{12}, f_{14}, f_{16}, f_{17}, f_{19}, and f_{20}. The optimal solution of these 14 functions can be found by the strategy proposed in this paper. These results explain that the strategy proposed in this paper can be well applied to these functions. Of course, the effects of f_{15} and f_{21} become worse with the increase of dimensions, and the optimization effect of SHMPSO on some multimodal functions which are difficult to optimize needs to be improved. In f_{18}, the effect of SHMPSO is not as good as that of CLPSO, BLPSO, and SLPSO. However, in the overall 21 test functions, the average rank of SHMPSO ranks first, 1.61 and 1.67, respectively. Through the above analysis, the performance of SHMPSO is better than that of other six comparison algorithms.

In order to further verify the scalability and high efficiency of SHMPSO analysis, the proposed strategy is used to solve the 100-dimension problem, and the set parameters are the same as those in Table 4. The experimental results are shown in the following table.

In Table 6, it can be seen that SHMPSO still has high convergence accuracy and good robustness when solving high-dimensional problems. SHMPSO and MPCPSO rank first on the 100-dimensional problem. Further analysis, the performance of f_{15}, f_{18}, and f_{21} deteriorates drastically with increasing dimension. But the performance of other functions is still very good.

It can be seen from Figures 1 and 2 that the convergence performance of SHMPSO has obtained the global optimal value on most problems. Especially in $f_3, f_{8,} f_{13}, f_{19}, f_{20}$, these five problems not only rank first in convergence accuracy but also the fastest convergence speed. Analyzing the reasons for convergence, there are three main reasons as follows. First of all, the information is highly shared among particles in the dominant population, and the guiding particles formed by the information sharing promote the dominant population to quickly converge to the global optimal value. Second, in order to prevent the population from falling into the local optimal value prematurely, a mutation operation is performed on the global optimal particle. Third, the use of random dominant population particles to guide the poor population particles not only improves the convergence speed of the poor population but also increases the diversity of the population. The particles of the dominant population are selected to guide the particles of the poor population, rather than the global optimal value of the poor population to guide the poor population. There are two reasons as follows: the first reason is that the fitness value of the poor population particles is larger than the

Table 3: Selection of parameter s.

		$s = 0.1$	$s = 0.2$	$s = 0.3$	$s = 0.4$	$s = 0.5$	$s = 0.6$	$s = 0.7$
f_1	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_2	Mean	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00 + 00	1.00E + 00	1.00E + 00
	Std.	6.78E − 16	6.78E − 16	6.78E − 16	6.78E − 16	6.78E − 16	6.78E − 16	6.78E − 16
	Rank	1	1	1	1	1	1	1
f_3	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_4	Mean	4.70E − 03	1.34E − 05	3.38E − 04	3.10E − 13	2.28E − 216	2.80E − 202	3.74E − 193
	Std.	2.57E − 02	5.10E − 05	1.80E − 03	1.70E − 12	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	7	5	6	4	1	2	3
f_5	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_6	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_7	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_8	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_9	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{10}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	0	1	1	1	1	1	1
f_{11}	Mean	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{12}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{13}	Mean	8.88E − 16	8.88E − 16	8.88E − 16	8.88E − 16	8.88E − 16	8.88E − 16	8.88E − 16
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{14}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{15}	Mean	1.07E + 00	4.47E − 01	4.51E − 01	5.90E − 01	2.19E − 01	1.01E + 00	1.75E − 01
	Std.	7.80E − 01	2.73E − 01	4.98E − 01	1.43E + 00	1.59E − 01	3.47E + 00	1.20E − 01
	Rank	7	3	4	5	2	6	1
f_{16}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{17}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{18}	Mean	6.18E − 06	3.71E − 06	4.44E − 07	7.78E − 07	2.57E − 07	3.67E − 08	2.49E − 07
	Std.	7.85E − 06	7.21E − 06	1.54E − 06	3.19E − 06	6.04E − 07	1.10E − 07	1.34E − 06
	Rank	7	6	4	5	3	1	2
f_{19}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{20}	Mean	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1

Table 3: Continued.

		$s = 0.1$	$s = 0.2$	$s = 0.3$	$s = 0.4$	$s = 0.5$	$s = 0.6$	$s = 0.7$
f_{21}	Mean	2.62E+00	2.89E+00	3.10E+00	2.81E+00	2.85E+00	2.83E+00	2.96E+00
	Std.	6.76E−01	6.23E−01	1.60E+00	4.34E−01	2.98E−01	5.88E−01	2.03E−01
	Rank	1	5	7	2	4	3	6
Average rank		1.86	1.71	1.81	1.57	1.29	1.38	1.38

Table 4: Comparison of results of benchmark functions on various PSO variants (30-D).

		PSO	CLPSO	BLPSO	ACPSO	SLPSO	MPCPSO	SHMPSO
f_1	Mean	1.34E+01	9.66E−11	5.10E−30	1.00E−11	2.08E−140	0.00E+00	0.00E+00
	Std.	1.47E+01	3.51E−11	4.15E−30	1.27E−11	3.39E−140	0.00E+00	0.00E+00
	Rank	7	6	4	5	3	1	1
f_2	Mean	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	Std.	6.78E−16	9.38E−13	4.65E−16	0.00E+00	6.78E−16	6.78E−16	6.78E−16
	Rank	3	7	2	1	3	3	3
f_3	Mean	1.05E+00	1.89E−07	0.00E+00	5.48E−02	5.75E−04	0.00E+00	0.00E+00
	Std.	6.97E−02	1.23E−07	0.00E+00	3.27E−02	2.21E−03	0.00E+00	0.00E+00
	Rank	7	4	1	6	5	1	1
f_4	Mean	0.00E+00	2.66E−08	5.58E−07	8.65E−14	1.10E−06	3.49E−18	5.98E−184
	Std.	0.00E+00	4.12E−08	8.24E−07	1.22E−13	1.72E−06	1.91E−17	0.00E+00
	Rank	1	5	6	4	7	3	2
f_5	Mean	3.13E+01	8.20E−05	1.24E−16	1.50E−04	2.70E−70	0.00E+00	0.00E+00
	Std.	1.44E+01	1.70E−05	6.60E−17	9.56E−05	3.09E−70	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
f_6	Mean	1.50E+01	1.17E+01	6.12E−01	5.21E−03	1.14E−37	0.00E+00	0.00E+00
	Std.	3.55E+00	9.31E−01	2.62E−01	4.08E−03	1.09E−37	0.00E+00	0.00E+00
	Rank	7	6	5	4	3	1	1
f_7	Mean	2.10E+02	1.17E−04	1.49E−16	1.70E−04	3.17E−69	0.00E+00	0.00E+00
	Std.	1.38E+02	2.44E−05	8.02E−17	6.14E−05	2.02E−69	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
f_8	Mean	5.55E+00	3.54E−28	1.54E−47	7.56E−49	0.00E+00	0.00E+00	0.00E+00
	Std.	1.17E+01	6.44E−28	5.49E−47	2.67E−48	0.00E+00	0.00E+00	0.00E+00
	Rank	7	6	5	4	1	1	1
f_9	Mean	0.00E+00	2.62E−11	2.75E−23	1.50E−11	0.00E+00	0.00E+00	0.00E+00
	Std.	0.00E+00	9.72E−12	3.65E−23	1.89E−11	0.00E+00	0.00E+00	0.00E+00
	Rank	1	7	5	6	1	1	1
f_{10}	Mean	6.40E+01	2.97E−09	5.47E−27	9.22E−10	1.07E−138	0.00E+00	0.00E+00
	Std.	3.67E+01	1.08E−09	5.16E−27	9.09E−10	1.81E−138	0.00E+00	0.00E+00
	Rank	7	6	4	5	3	1	1
f_{11}	Mean	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1	1	1	1	1
f_{12}	Mean	6.44E+01	1.18E+01	4.00E−05	1.77E−05	2.11E+00	0.00E+00	0.00E+00
	Std.	3.60E+01	2.35E+00	4.30E−05	2.46E−05	2.27E+00	0.00E+00	0.00E+00
	Rank	7	6	4	3	5	1	1
f_{13}	Mean	8.53E+00	1.48E−04	2.95E−14	1.88E−05	5.74E−15	8.88E−16	8.88E−16
	Std.	9.51E−01	2.87E−05	1.79E−14	9.74E−06	1.23E−15	0.00E+00	0.00E+00
	Rank	7	6	4	5	3	1	1
f_{14}	Mean	7.33E+00	5.68E−04	2.16E−10	1.54E−05	5.00E−17	0.00E+00	0.00E+00
	Std.	2.56E+00	1.13E−04	5.15E−10	9.12E−06	1.31E−16	0.00E+00	0.00E+00
	Rank	7	6	4	5	3	1	1
f_{15}	Mean	2.89E−01	5.33E−02	6.84E−03	8.89E−03	4.42E−02	7.36E+00	2.19E−01
	Std.	1.01E−01	6.61E−03	1.55E−03	3.70E−03	9.95E−03	1.03E+01	1.59E−01
	Rank	6	4	1	2	3	7	5
f_{16}	Mean	5.43E−01	1.01E−01	1.00E−01	1.00E−01	1.00E−01	0.00E+00	0.00E+00
	Std.	2.53E−01	3.44E−04	2.94E−06	3.21E−06	1.13E−16	0.00E+00	0.00E+00
	Rank	7	6	4	5	3	1	1
f_{17}	Mean	6.91E+01	1.14E−04	0.00E+00	2.48E−07	1.34E+01	0.00E+00	0.00E+00
	Std.	1.20E+01	4.23E−05	0.00E+00	6.30E−07	3.87E+00	0.00E+00	0.00E+00

Table 4: Continued.

		PSO	CLPSO	BLPSO	ACPSO	SLPSO	MPCPSO	SHMPSO
	Rank	7	5	1	4	6	1	1
f_{18}	Mean	6.27E − 11	3.52E − 12	3.53E − 12	3.51E − 12	8.54E − 12	9.64E − 05	2.57E − 07
	Std.	4.87E − 11	1.33E − 15	8.58E − 14	5.05E − 19	1.45E − 12	2.81E − 04	6.04E − 07
	Rank	5	2	3	1	4	7	6
f_{19}	Mean	1.68E + 03	2.38E + 03	1.33E + 01	7.40E − 02	1.72E − 11	0.00E + 00	0.00E + 00
	Std.	1.09E + 03	4.68E + 02	6.45E + 00	1.07E − 01	2.36E − 11	0.00E + 00	0.00E + 00
	Rank	6	7	5	4	3	1	1
f_{20}	Mean	5.78E + 02	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	2.18E + 02	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	7	1	1	1	1	1	1
f_{21}	Mean	8.23E + 01	3.95E − 08	2.71E − 26	3.95E + 07	1.35E − 32	2.91E + 00	2.82E + 00
	Std.	9.32E + 01	1.50E − 08	1.97E − 26	2.16E + 08	5.57E − 48	1.30E − 01	3.37E − 01
	Rank	6	3	2	7	1	5	4
Average rank		6	4.95	2.73	4.05	3.1	1.95	1.71

Table 5: Comparison of results of benchmark functions on various PSO variants (50-D).

		PSO	CLPSO	BLPSO	ACPSO	SLPSO	MPCPSO	SHMPSO
f_1	Mean	4.38E + 01	8.38E − 11	8.18E − 35	3.50E − 12	8.09E − 160	0.00E + 00	0.00E + 00
	Std.	2.67E + 01	2.48E − 11	8.13E − 35	2.59E − 12	1.44E − 159	0.00E + 00	0.00E + 00
	Rank	7	6	4	5	3	1	1
f_2	Mean	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00
	Std.	2.26E − 16	6.21E − 16	3.66E − 16	0.00E + 00	4.46E − 16	2.26E − 16	2.26E − 16
	Rank	2	7	5	1	6	2	2
f_3	Mean	1.58E + 00	2.19E − 08	0.00E + 00	2.43E − 02	1.64E − 03	0.00E + 00	0.00E + 00
	Std.	1.31E − 01	9.27E − 09	0.00E + 00	2.76E − 02	3.85E − 03	0.00E + 00	0.00E + 00
	Rank	7	4	1	6	5	1	1
f_4	Mean	0.00E + 00	1.16E − 08	2.99E − 07	8.88E − 17	1.52E − 06	8.36E − 11	2.24E − 179
	Std.	0.00E + 00	1.78E − 08	2.24E − 07	2.71E − 16	1.11E − 06	4.58E − 10	0.00E + 00
	Rank	1	5	6	3	7	4	2
f_5	Mean	7.91E + 01	9.12E − 05	1.04E − 19	1.34E − 04	9.39E − 81	0.00E + 00	0.00E + 00
	Std.	3.68E + 01	1.24E − 05	6.29E − 20	2.81E − 05	1.56E − 80	0.00E + 00	0.00E + 00
	Rank	7	5	4	6	3	1	1
f_6	Mean	2.33E + 01	1.50E + 01	1.41E + 00	6.37E − 03	1.69E − 28	0.00E + 00	0.00E + 00
	Std.	2.60E + 00	1.10E + 00	2.77E − 01	2.11E − 03	3.90E − 28	0.00E + 00	0.00E + 00
	Rank	7	6	5	4	3	1	1
f_7	Mean	9.10E + 02	1.38E − 04	5.29E − 20	1.50E − 04	2.00E − 79	0.00E + 00	0.00E + 00
	Std.	1.89E + 02	2.09E − 05	2.52E − 20	3.13E − 05	2.02E − 79	0.00E + 00	0.00E + 00
	Rank	7	5	4	6	3	1	1
f_8	Mean	1.20E + 03	7.04E − 29	1.15E − 45	6.36E − 54	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	1.44E + 03	7.10E − 29	4.92E − 45	2.80E − 53	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	7	6	5	4	1	1	1
f_9	Mean	4.00E + 01	8.82E − 11	3.24E − 23	6.60E − 12	0.00E + 00	0.00E + 00	0.00E + 00
	Std.	5.63E + 01	2.56E − 11	2.83E − 23	3.31E − 12	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	7	6	4	5	1	1	1
f_{10}	Mean	4.92E + 02	5.20E − 09	2.40E − 31	4.80E − 10	8.33E − 158	0.00E + 00	0.00E + 00
	Std.	1.75E + 02	1.15E − 09	3.14E − 31	2.07E − 10	2.17E − 157	0.00E + 00	0.00E + 00
	Rank	7	6	4	5	3	1	1
f_{11}	Mean	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00
	Std.	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	Rank	1	1	1	1	1	1	1
f_{12}	Mean	2.70E + 02	5.06E + 01	7.70E − 03	6.54E − 09	1.83E + 02	0.00E + 00	0.00E + 00
	Std.	9.69E + 01	7.82E + 00	4.95E − 03	3.77E − 09	3.04E + 01	0.00E + 00	0.00E + 00
	Rank	7	5	4	3	6	1	1
f_{13}	Mean	1.18E + 01	1.06E − 04	6.45E − 15	1.01E − 05	6.34E − 15	8.88E − 16	8.88E − 16
	Std.	8.40E − 01	1.31E − 05	9.01E − 16	2.35E − 06	6.49E − 16	0.00E + 00	0.00E + 00
	Rank	7	6	4	5	3	1	1
	Mean	2.17E + 01	1.13E − 03	9.04E − 11	3.83E − 05	1.85E − 16	0.00E + 00	0.00E + 00

Table 5: Continued.

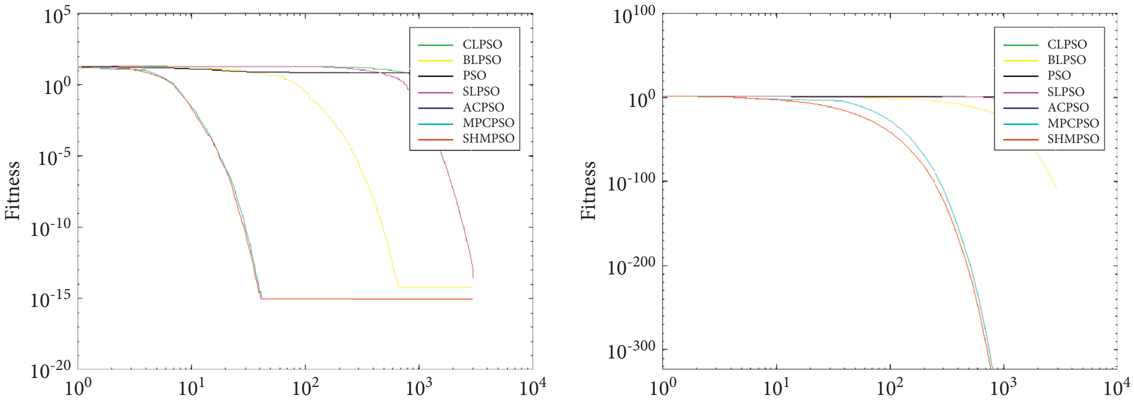
		PSO	CLPSO	BLPSO	ACPSO	SLPSO	MPCPSO	SHMPSO
f_{14}	Std.	5.10E+00	1.74E−04	4.94E−10	1.70E−05	2.62E−16	0.00E+00	0.00E+00
	Rank	6	5	4	7	3	1	1
f_{15}	Mean	5.13E−01	7.88E−02	1.24E−02	1.01E−02	1.29E−01	6.22E+00	2.96E+00
	Std.	1.22E−01	7.61E−03	2.09E−03	3.16E−03	2.24E−02	1.41E+01	6.63E+00
	Rank	5	3	2	1	4	7	6
f_{16}	Mean	2.73E+00	1.02E−01	1.00E−01	1.00E−01	1.00E−01	0.00E+00	0.00E+00
	Std.	8.90E−01	4.24E−04	2.40E−06	6.10E−05	7.06E−17	0.00E+00	0.00E+00
	Rank	7	6	4	5	3	1	1
f_{17}	Mean	1.87E+02	2.30E−04	3.79E−15	1.87E−07	3.01E+01	0.00E+00	0.00E+00
	Std.	3.36E+01	7.25E−05	2.08E−14	1.08E−07	8.15E+00	0.00E+00	0.00E+00
	Rank	7	5	3	4	6	1	1
f_{18}	Mean	1.17E−16	1.21E−20	1.27E−20	1.21E−20	3.40E−20	1.98E−08	7.73E−12
	Std.	2.25E−16	7.47E−24	6.13E−22	2.18E−26	4.04E−21	7.77E−08	3.83E−11
	Rank	5	2	3	1	4	7	6
f_{19}	Mean	8.29E+03	1.21E+04	3.47E+02	1.26E−02	2.69E−03	0.00E+00	0.00E+00
	Std.	3.39E+03	1.99E+03	8.55E+01	7.02E−03	5.59E−03	0.00E+00	0.00E+00
	Rank	6	7	5	4	3	1	1
f_{20}	Mean	3.38E+03	0.00E+00	0.00E+00	0.00E+00	3.33E−02	0.00E+00	0.00E+00
	Std.	9.39E+02	0.00E+00	0.00E+00	0.00E+00	1.83E−01	0.00E+00	0.00E+00
	Rank	7	1	1	1	6	1	1
f_{21}	Mean	5.43E+04	2.43E−08	1.20E−30	6.50E−06	1.46E−03	4.88E+00	4.89E+00
	Std.	1.22E+05	5.25E−09	7.11E−31	3.55E−05	3.80E−03	1.61E−01	5.37E−01
	Rank	7	2	1	3	4	5	6
Average rank		5.9	4.71	3.52	4.29	3.71	1.95	1.81

Table 6: Comparison of results of benchmark functions on various PSO variants (100-D).

		PSO	CLPSO	BLPSO	ACPSO	SLPSO	MPCPSO	SHMPSO
f_1	Mean	4.10E+02	2.01E−11	9.65E−40	2.73E−07	8.53E−179	0.00E+00	0.00E+00
	Std.	2.77E+02	3.26E−12	1.85E−39	7.60E−08	0.00E+00	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
f_2	Mean	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	Std.	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Rank	1	1	1	1	1	1	1
f_3	Mean	3.82E+00	1.15E−09	1.11E−17	9.76E−03	9.86E−04	0.00E+00	0.00E+00
	Std.	4.56E−01	4.82E−10	3.39E−17	1.05E−02	3.08E−03	0.00E+00	0.00E+00
	Rank	7	4	3	6	5	1	1
f_4	Mean	7.49E−01	1.49E−09	1.60E−07	0.00E+00	4.33E−07	1.51E−182	2.26E−217
	Std.	5.53E−01	1.88E−09	2.31E−07	0.00E+00	3.96E−07	0.00E+00	0.00E+00
	Rank	7	4	5	1	6	3	2
f_5	Mean	3.31E+02	5.58E−05	1.94E−24	2.24E−02	2.67E−91	0.00E+00	0.00E+00
	Std.	2.00E+02	5.49E−06	3.15E−24	2.89E−03	4.86E−91	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
f_6	Mean	2.96E+01	1.92E+01	4.20E+00	2.50E−01	1.43E−12	0.00E+00	0.00E+00
	Std.	2.80E+00	6.32E−01	5.58E−01	3.29E−02	1.02E−12	0.00E+00	0.00E+00
	Rank	7	6	5	4	3	1	1
f_7	Mean	1.95E+03	9.77E−05	5.23E−25	2.21E−02	1.23E−89	0.00E+00	0.00E+00
	Std.	2.03E+02	1.21E−05	4.15E−25	3.14E−03	1.18E−89	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
f_8	Mean	1.08E+05	1.73E−30	8.17E−41	7.07E−28	1.51E−317	0.00E+00	0.00E+00
	Std.	9.88E+04	1.04E−30	1.90E−40	6.30E−28	0.00E+00	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
f_9	Mean	2.30E+02	1.64E−10	2.77E−17	5.44E−07	0.00E+00	0.00E+00	0.00E+00
	Std.	1.60E+02	2.82E−11	1.19E−16	1.60E−07	0.00E+00	0.00E+00	0.00E+00
	Rank	7	5	4	6	1	1	1
f_{10}	Mean	4.74E+03	2.65E−09	1.29E−35	7.50E−05	5.73E−176	0.00E+00	0.00E+00
	Std.	1.36E+03	5.23E−10	4.69E−35	2.55E−05	0.00E+00	0.00E+00	0.00E+00
	Rank	7	5	4	6	3	1	1
	Mean	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00

Table 6: Continued.

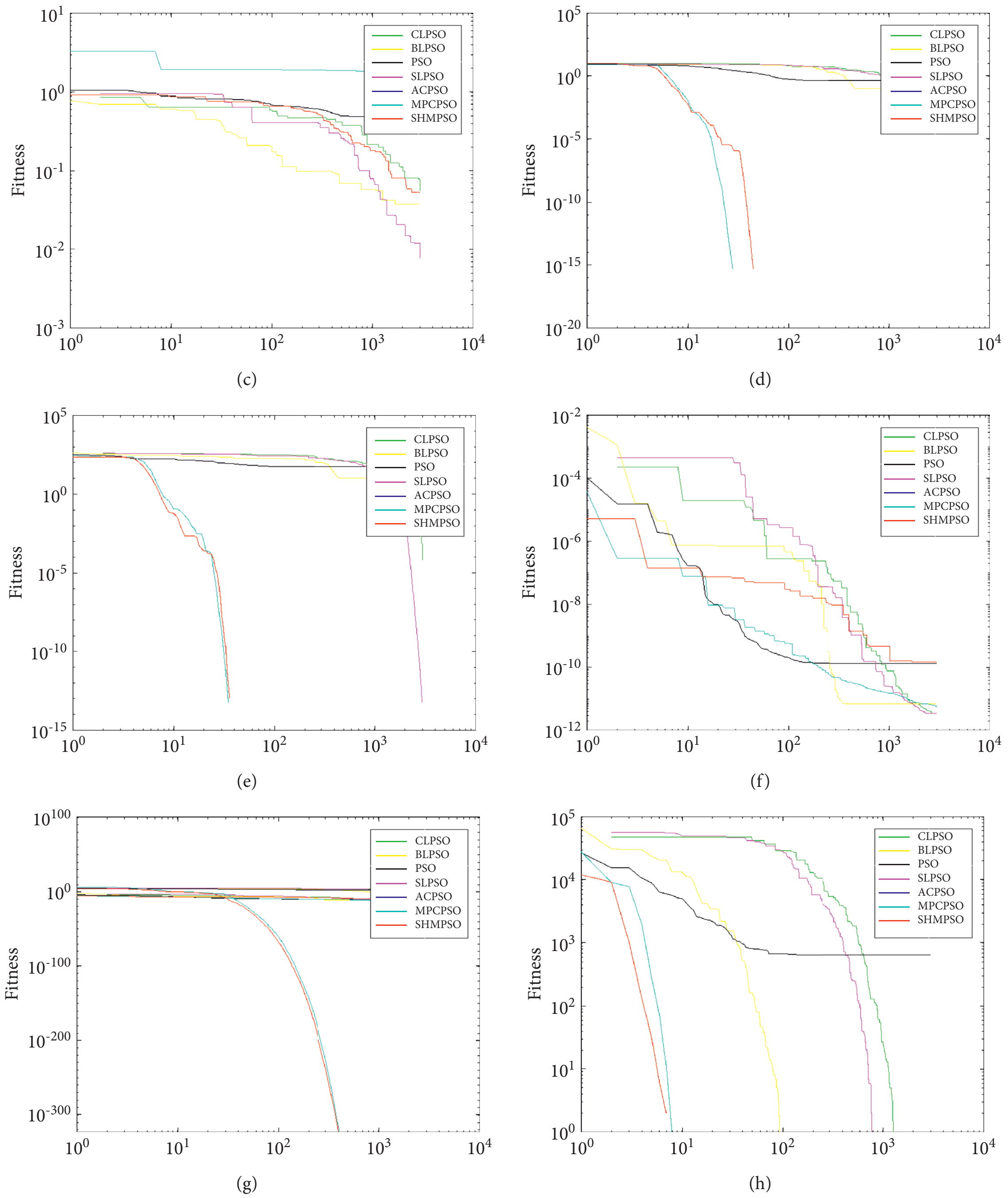
		PSO	CLPSO	BLPSO	ACPSO	SLPSO	MPCPSO	SHMPSO
f_{11}	Std.	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	Rank	1	1	1	1	1	1	1
f_{12}	Mean	$1.54E+03$	$2.38E+02$	$3.27E+00$	$9.07E-10$	$9.24E+02$	$0.00E+00$	$0.00E+00$
	Std.	$4.11E+02$	$1.98E+01$	$8.08E-01$	$4.46E-10$	$9.08E+01$	$0.00E+00$	$0.00E+00$
	Rank	7	5	4	3	6	1	1
f_{13}	Mean	$1.44E+01$	$3.30E-05$	$1.46E-14$	$1.70E-03$	$1.38E-14$	$8.88E-16$	$8.88E-16$
	Std.	$5.44E-01$	$2.68E-06$	$2.87E-15$	$2.56E-04$	$4.04E-15$	$0.00E+00$	$0.00E+00$
	Rank	7	5	4	6	3	1	1
f_{14}	Mean	$6.11E+01$	$2.45E-03$	$6.56E-16$	$7.36E-03$	$1.16E-15$	$0.00E+00$	$0.00E+00$
	Std.	$7.72E+00$	$2.57E-04$	$1.47E-15$	$1.32E-03$	$1.02E-15$	$0.00E+00$	$0.00E+00$
	Rank	7	5	3	6	4	1	1
f_{15}	Mean	$6.90E-01$	$1.49E-01$	$2.59E-02$	$2.29E-02$	$2.79E-01$	$1.88E+00$	$3.27E+00$
	Std.	$9.15E-02$	$1.46E-02$	$2.70E-03$	$3.60E-03$	$3.54E-02$	$4.41E-02$	$9.98E+00$
	Rank	5	3	2	1	4	6	7
f_{16}	Mean	$1.23E+01$	$1.04E-01$	$1.00E-01$	$1.51E-01$	$1.00E-01$	$0.00E+00$	$0.00E+00$
	Std.	$2.63E+00$	$6.29E-04$	$4.18E-06$	$1.13E-02$	$1.07E-16$	$0.00E+00$	$0.00E+00$
	Rank	7	5	4	6	3	1	1
f_{17}	Mean	$5.54E+02$	$3.63E-04$	$2.98E-01$	$1.14E-01$	$1.08E+02$	$0.00E+00$	$0.00E+00$
	Std.	$3.99E+01$	$7.70E-05$	$5.32E-01$	$4.63E-02$	$2.79E+01$	$0.00E+00$	$0.00E+00$
	Rank	7	3	5	4	6	1	1
f_{18}	Mean	$7.45E-28$	$4.68E-42$	$6.04E-42$	$5.51E-42$	$1.62E-41$	$4.49E-11$	$1.68E-18$
	Std.	$3.03E-27$	$4.16E-45$	$3.89E-43$	$5.39E-43$	$1.49E-42$	$2.39E-10$	$9.22E-18$
	Rank	5	1	3	2	4	7	6
f_{19}	Mean	$3.71E+04$	$6.84E+04$	$1.30E+04$	$9.40E+04$	$1.61E+04$	$0.00E+00$	$0.00E+00$
	Std.	$1.14E+04$	$5.96E+03$	$1.84E+03$	$5.15E+05$	$7.73E+03$	$0.00E+00$	$0.00E+00$
	Rank	5	6	3	7	4	1	1
f_{20}	Mean	$1.67E+04$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$4.67E-01$	$0.00E+00$	$0.00E+00$
	Std.	$2.68E+03$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$1.66E+00$	$0.00E+00$	$0.00E+00$
	Rank	7	1	1	1	6	1	1
f_{21}	Mean	$2.49E+06$	$6.38E-09$	$2.15E-30$	$1.67E+08$	$2.56E-03$	$9.82E+00$	$9.95E+00$
	Std.	$1.56E+06$	$1.11E-09$	$7.87E-30$	$9.17E+08$	$4.73E-03$	$2.31E-01$	$2.38E-01$
	Rank	6	2	1	7	3	4	5
Average rank		5.1	3.9	3.29	4.38	3.57	1.76	1.76



(a)

(b)

Figure 2: Continued.



(c)

(d)

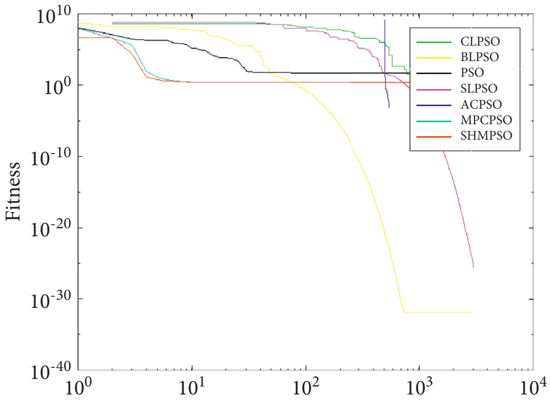
(e)

(f)

(g)

(h)

Figure 2: Continued.



(i)

Figure 2: Convergence curves of 13–21 functions. (a) f_{13}. (b) f_{14}. (c) f_{15}. (d) f_{16}. (e) f_{17}. (f) f_{18}. (g) f_{19}. (h) f_{20}. (i) f_{21}.

fitness value of the population; the second reason is that the use of a single poor population global optimal particle is not conducive to increasing the diversity of the population.

According to the experiments in this paper, SHMPSO has good performance, and it has good performance in most functions. The success of SHMPSO mainly depends on the strategies proposed in this paper: First, based on the flexible candidate learning strategy, elite particles are selected from the dominant population to let the poor population particles learn, and the two populations share information, effectively jumping out of the local optimal solution. In addition, the mean dimension learning strategy can make the population particles have a better search range in complex and changeable multimodal functions, greatly improve the learning samples of particles, and provide more effective information for all particles. Therefore, SHMPSO has excellent properties and convergence accuracy.

5. Conclusion

Inspired by MPCPSO, this paper proposes a particle swarm optimization algorithm based on the strategy of subpopulation mixing. In this paper, the mean dimension learning strategy ensures the searching ability of the algorithm and the breadth of learnable samples, which provides the searching potential for the whole population. At the same time, the candidate learning strategy is used to improve the diversity of the population and prevent the population from falling into local optimum. At the same time, this paper compares SHMPSO with six well-known PSO variants to verify the effectiveness of SHMPSO proposed in this paper. Of course, SHMPSO still has some shortcomings, and f_2 and f_{11} fall into local optimum. As the population searching ability needs to be improved, our future work will focus on the global searching ability of SHMPSO, and at the same time, we will deeply study the practical application of SHMPSO.

Data Availability

All data of the paper can be obtained through the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by projects from the National Key Research and Development Program (grant no. 2020YFB1713700).

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. Neural Netw.*vol. 4, pp. 1942–1948, 1995.
[2] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
[3] W. Li, S. Li, Z. Chen, L. Zhong, and C. Ouyang, "Self-feedback differential evolution adapting to fitness landscape characteristics," *Soft Computing*, vol. 23, no. 4, pp. 1151–1163, 2019.
[4] O. Engin and A. Güçlü, "A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems," *Applied Soft Computing*, vol. 72, pp. 166–176, 2018.
[5] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Computing*, vol. 22, no. 9, pp. 2935–2952, 2018.
[6] X. Xin Yao, Y. Yong Liu, and G. Guangming Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
[7] L. Wei, Z. Zhang, D. Zhang, and S. C. H. Leung, "A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints," *European Journal of Operational Research*, vol. 265, no. 3, pp. 843–859, 2018.
[8] J. Zhou, Z. Ji, and L. Shen, "Simplified intelligence single particle optimization based neural network for digit recognition," in *Proceedings of the Chinese Conference on Pattern Recognition*, pp. 1–5, IEEE, Beijing, China, October 2008.
[9] L. Cagnina, M. Errecalde, D. Ingaramo, and P. Rosso, "An efficient particle swarm optimization approach to cluster short texts," *Information Sciences*, vol. 265, pp. 36–49, 2014.
[10] Y.-J. Gong, J. Zhang, H. S.-H. Chung et al., "An efficient resource allocation scheme using particle swarm

optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 801–816, 2012.

[11] S.-Y. Shinn-Ying Ho, H.-S. Hung-Sui Lin, W.-H. Weei-Hurng Liauh, and S.-J. Shinn-Jang Ho, "OPSO: orthogonal particle swarm optimization and its application to task assignment problems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 38, no. 2, pp. 288–298, 2008.

[12] C.-A. Bojan-Dragos, R.-E. Precup, S. Preitl, R.-C. Roman, E.-L. Hedrea, and A.-I. Szedlak-Stinean, "GWO-based optimal tuning of type-1 and type-2 fuzzy controllers for electromagnetic actuated clutch systems," *IFAC-PapersOnLine*, vol. 54, no. 4, pp. 189–194, 2021.

[13] R.-E. Precup, R.-C. David, R.-C. Roman, A.-I. Szedlak-Stinean, and E. M. Petriu, "Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using Slime Mould Algorithm," *International Journal of Systems Science*, pp. 1–16, 2021.

[14] F. Wang, Y. X. Li, A. Zhou, and K. Tang, "An estimation of distribution algorithm for mixed-variable Newsvendor problems," *IEEE Transactions on Evolutionary Computation*, 2019.

[15] R.-E. Precup, E.-L. Hedrea, R.-C. Roman, E. M. Petriu, A.-I. Szedlak-Stinean, and C.-A. Bojan-Dragos, "Experiment-based approach to teach optimization techniques," *IEEE Transactions on Education*, vol. 64, no. 2, pp. 88–94, 2021.

[16] G. Kaur, S. S. Gill, and M. Rattan, "Whale optimization algorithm for performance improvement of silicon-on-insulator FinFETs[J]," *International Journal of Artificial Intelligence*, vol. 18, no. 1, pp. 63–81, 2020.

[17] M. Moattari and M. H. Moradi, "Conflict monitoring optimization heuristic inspired by brain fear and conflict systems," *International Journal of Artificial Intelligence*, vol. 18, no. 1, pp. 45–62, 2020.

[18] F. Wang, F. Liao, Y. Li, and H. Wang, "A new prediction strategy for dynamic multi-objective optimization using Gaussian mixture model," *Information Sciences*, vol. 580, pp. 331–351, 2021.

[19] F. Wang, Y. Li, F. Liao, and H. Yan, "An ensemble learning based prediction strategy for dynamic multi-objective optimization," *Applied Soft Computing*, vol. 96, Article ID 106592, 2020.

[20] F. Wang, Y. Li, A. Zhou, and K. Tang, "An estimation of distribution algorithm for mixed-variable Newsvendor problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 479–493, 2020.

[21] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning*, pp. 412–420, Morgan Kaufmann Publishers, July 1997.

[22] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 2, p. 1980, June 2004.

[23] S. Yadav, A. Ekbal, and S. Saha, "Feature selection for entity extraction from multiple biomedical corpora: a PSO-based approach," *Soft Computing*, vol. 22, no. 20, pp. 6881–6904, 2018.

[24] F. Wang, H. Zhu, W. Li, and K. Li, "A hybrid convolution network for serial number recognition on banknotes," *Information Sciences*, vol. 512, pp. 952–963, 2020.

[25] A. R. Jordehi, "Particle swarm optimisation (PSO) for allocation of FACTS devices in electric transmission systems: a review," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 1260–1267, 2015.

[26] M. Alswaitti, M. Albughdadi, and N. A. M. Isa, "Density-based particle swarm optimization algorithm for data clustering," *Expert Systems with Applications*, vol. 91, pp. 170–186, 2018.

[27] F. Wang, H. Zhang, and A. Zhou, "A particle swarm optimization algorithm for mixed-variable optimization problems," *Swarm & Evolutionary Computation*, vol. 60, Article ID 100808, 2021.

[28] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," *Lecture Notes in Computer Science. In Evolutionary Programming VII*, Springer Berlin Heidelberg, NY, USA, pp. 591–600, 1998.

[29] Z. H. Zhi-Hui Zhan, J. Jun Zhang, Y. Yun Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, 2009.

[30] Q. Yang, W. N. Chen, T. Gu et al., "Segment-based predominant learning swarm optimizer for large-scale optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2896–2910, 2016.

[31] J. Liang and P. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 124–129, Hong Kong, China, June 2005.

[32] W. Li, B. Sun, Y. Huang, and S. Mahmoodi, "Adaptive particle swarm optimization using scale-free network topology," *Journal of Network Intelligence*, vol. 6, no. 3, pp. 500–517, 2021.

[33] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1272–1282, 2005.

[34] X. Zhang, X. Wang, Q. Kang, and J. Cheng, "Differential mutation and novel social learning particle swarm optimization algorithm," *Information Sciences*, vol. 480, pp. 109–129, 2019.

[35] F. Valdez, P. Melin, and O. Castillo, "Modular neural networks architecture optimization with a new nature inspired method using a fuzzy combination of particle swarm optimization and genetic algorithms," *Information Sciences*, vol. 270, pp. 143–153, 2014.

[36] W. Li, X. Meng, Y. Huang, and Z.-H. Fu, "Multipopulation cooperative particle swarm optimization with a mixed mutation strategy," *Information Sciences*, vol. 529, pp. 179–196, 2020.

[37] 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.

[38] X. Chen, H. Tianfield, C. Mei, W. Du, and G. Liu, "Biogeography-based learning particle swarm optimization," *Soft Computing*, vol. 21, no. 24, pp. 7519–7541, 2017.

[39] R. Liu, J. Li, J. fan, C. Mu, and L. Jiao, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *European Journal of Operational Research*, vol. 261, no. 3, pp. 1028–1051, 2017.

[40] D. Yazdani, B. Nasiri, A. Sepas-Moghaddam, and M. R. Meybodi, "A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization," *Applied Soft Computing*, vol. 13, no. 4, pp. 2144–2158, 2013.

[41] 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.

[42] G. G. Yen and M. Daneshyari, "Diversity-based information exchange among multiple swarms in particle swarm optimization," *International Journal of Computational Intelligence and Applications*, vol. 07, no. 01, pp. 57–75, 2008.

[43] F. vandenBergh and A. P. Engelbrecht, “A cooperative approach to particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[44] F. Wang, H. Zhang, Y. Li, Y. Zhao, and Q. Rao, “External archive matching strategy for MOEA/D,” *Soft Computing*, vol. 22, no. 23, pp. 7833–7846, 2018.

[45] Y. Huang, W. Li, C. Ouyang, and Y. Chen, “A self-feedback strategy differential evolution with fitness landscape analysis,” *Soft Computing*, vol. 22, no. 23, pp. 7773–7785, 2018.

[46] H. Mohammad, R. M. Mohammad, and M. E. Mohammad, “Adaptive cooperative particle swarm optimizer,” *Applied Intelligence*, vol. 39, pp. 397–420, 2013.

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