IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 7, 2020, accepted July 15, 2020, date of publication July 20, 2020, date of current version August 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010543

A Molecular Interactions-Based Social Learning Particle Swarm Optimization Algorithm

YUN HOU[1], GUO-SHENG HAO[2], YONG ZHANG[1], (Member, IEEE), FENG GU[3], XIA WANG[2], AND TING-TING ZHANG[2]

[1]School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China
[2]School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221116, China
[3]Department of Computer Science, College of Staten Island, The City University of New York, Staten Island, NY 10314, USA

Corresponding author: Guo-Sheng Hao (hgskd@jsnu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61876185 and Grant 61877030, in part by the Society Development Foundation of Xuzhou under Grant KC19213, and in part by the Cernet Foundation for Technology Innovation for Next Generation of Internet under Grant NGII20190513.

ABSTRACT Social learning particle swarm optimization (SL-PSO) allows individuals to learn from others to improve the scalability with easy parameter settings. However, it still suffers from the poor convergence for those multi-modal problems due to the loss of swarm diversity. To improve both the diversity and the convergence, this paper proposes a novel algorithm to apply the mechanism of molecular interactions to SL-PSO, in which the molecular attraction aims to improve the convergence, and the molecular repulsion intends to enhance the diversity. In the experiments, we compare our algorithm with the SL-PSO algorithm and other representative PSO and evolutionary algorithms on 49 benchmark functions. The results show the performance of the proposed algorithm is better than that of the SL-PSO algorithm and other representative PSO and evolutionary algorithms on average. This work builds the solid foundation for the integration of the molecular interaction mechanism with PSO and other optimization algorithms.

INDEX TERMS Convergence, diversity, learning, molecular interactions, particle swarm optimization.

I. INTRODUCTION

Particle swarm optimization (PSO) is a computational method that seeks the optima of a problem by iteratively updating of particles' positions. PSO was proposed by Kennedy and Eberhart in 1995 [1], [2] to simulate social behaviors, representing the movement in a bird flock or fish school. Due to few or no assumptions about the optimization problems, PSO has been widely applied in many fields, such as power systems [3], [4], image processing [5]–[7], and motor parameter settings [8], [9].

Social Learning PSO (SL-PSO) [10] was proposed by Cheng and Jin in 2015. SL-PSO introduced the learning strategy that imitated behaviors in social learning to traditional PSO to gain the advantages of high convergence speed and strong optimization ability. SL-PSO performs well in low dimensional problems and has promising results for high dimensional problems. However, due to the inability to effectively balance the convergence and diversity of population, SL-PSO has some limitations in the optimization of complex functions. For example, the optimization results are often not ideal for some multi-modal problems of ill-condition.

Molecular interactions include attractive and repulsive forces between molecules, and help scientists understand a molecular function and its behavior in different areas including protein folding [11], [12], drug design [13], [14], voltage sensors [15], [16], and nanotechnology [17]. However, molecular interactions have only few applications in swarm intelligence. In the limited work, we are aware that the repulsion is generally used to increase the population diversity while the attraction is used to increase the population convergence. Studies have different focuses, such as the diversity among population [18] and the diversity and convergence equilibrium among the particle swarms [19].

In PSO, the tradeoff between the convergence and the diversity needs to be balanced. We introduce the molecular interactions into SL-PSO in this paper to control the distributions of particles in the search space. The contributions of this paper lie in four folds below. Firstly, we apply the concept of molecular interactions including attractive forces and repulsive forces to SL-PSO to improve both the convergence

The associate editor coordinating the review of this manuscript and approving it for publication was Seyedali Mirjalili.

and the diversity of particles. Secondly, we propose the new position update strategy by predefining a threshold to control the convergence speed using attractive forces and calibrate the threshold for given optimization problems. Thirdly, we calculate the repulsive force according to the collision of elastic objects to improve the diversity of particles to update their positions, and add the initialization policy to avoid premature convergence. Finally, we test our proposed algorithm on different optimized benchmark functions and compare with SL-PSO and other representative PSO and evolutionary algorithms to show its effectiveness.

The rest of the paper is organized as follows. Section 2 introduces the related work. Then the basics of the conventional PSO algorithm and the SL-PSO algorithm are described in Section 3. Section 4 presents the SL-PSO algorithm based on molecular interactions, analyzes its time complexity, and justifies its convergence. The experiments and results are presented in Section 5. Section 6 gives the discussions about some related issues. Finally, this paper is concluded in Section 7.

II. RELATED WORK

A. SOME PSO VARIANTS

One of the most important problems in the traditional PSO algorithm is the premature convergence when the particles are trapped in a local optimal and the particles' velocities tend to be zero. Many modifications have been recently proposed to improve the performance and overcome the disadvantages of the PSO algorithm. For example, Mendes *et al.* [20] proposed an efficient and simple PSO algorithm called fully informed particle swarm (FIPS), which used all the information of the particle neighborhood to adjust its movement velocity. Liang *et al.* [21] proposed a comprehensive learning PSO (CLPSO) with a new effective learning strategy, with which particles updated each decision variable by learning from different historical local best positions. Orthogonal learning was introduced in PSO by Zhan *et al.* [22] to learn from both global and local orthogonal directions. Chen *et al.* [23] presented an organizational dynamic adjustment PSO based particle filter algorithm, which allowed the particles to be adaptive to the dynamic environment and reach the new global optimum. Jin *et al.* [24] combined PSO with the gradient method to avoid premature convergence. Du *et al.* [25] presented a particle swarm optimization with limited information called LIPSO, in which particles are influenced by the top individuals of the population sorted by performance. Niu *et al.* [26] proposed a new variant of PSO, in which particles not only exchange social experience with others that are from their own sub-swarms, but also are influenced by the experience of particles from other fellow sub-swarms, named Symbiosis-based Alternative Learning Multi-swarm Particle Swarm Optimization. Yang *et al.* [27] proposed a level-based learning swarm optimizer to settle large-scale optimization, which separates particles into a number of levels according to their fitness values and randomly selects two predominant particles from two different higher levels in the current swarm to guide the learning of particles.

In addition to abovementioned algorithms, some other improved PSO algorithms include adaptive PSO based on clustering [28], a memetic PSO for dynamic multi-modal optimization [29], developmental swarm intelligence in PSO [30], parasitic behavior integrated PSO [31], genetic learning embedded PSO [32], diversity purposed neighborhood search enforced PSO [33], biogeography learning based PSO [34], a generalized theoretical deterministic model based PSO [35], parallel implementation based multi-swarm PSO [36], adaptive time-varying topology connectivity based PSO [37], jumping time-varying acceleration coefficients incorporated PSO [38], the global best-guided PSO [39], the inter swarm interactive learning strategy PSO [40], and the PSO with dual-level task allocation [41]. Although these variants of PSO algorithms improve the performance, they also suffer from the burden of the multi-parameter settings. For the purpose of easy parameters setting and the performance enhancement, social learning PSO was proposed, in which each particle learns from one of the better particles in the current swarm and only one dimension-dependent parameter control method is applied [10]. However, it still may suffer from the tradeoff between the diversity and the convergence.

B. SOME REPULSIVE-ATTRACTIVE ALGORITHMS

Molecular interactions, a ubiquitous physical phenomenon, have inspired many researchers to apply the concepts in different applications, such as medical treatment, materials, sensors and so on. Zandevakili *et al.* [19] proposed the gravitational search algorithm (GSA) by using centrifugal force, which was called as the repulsion. A distance threshold value was used, beyond which the repulsion worked according to the gravity formula. Although it used both the gravity and the centrifugal force, it has not been applied to PSO algorithms. In addition, it suffers from high time complexity and complicated parameter settings.

Krink *et al.* [42] first proposed an attractive-repulsive framework, in which, the particle velocity update based on the repulsive mechanism was contrary to traditional PSO. The diversity was calculated at each iteration and used to switch between the attractive and the repulsive phases. When the diversity fell below a certain threshold value, the swarm switched to the repulsion phase to increase its diversity. When the diversity exceeded the threshold value, the swarm switched to the attractive phase, the particles had a normal behavior, and the swarm tended to converge again. However, it simply used the learning as the attractive force and the collision as the repulsive force. Therefore, the update strategy of positions was simple and the improvement of the performance was limited. Its related applications can be referred to [43], [44].

Mo *et al.* [45] presented another attractive and repulsive fully informed PSO. In their study, the attraction meant

learning from particles with better fitness around them, and the repulsion referred to moving away from particles with the worse fitness. However, the attraction and the repulsion were only conceptually proposed and used. Blackwell and Branke [18] introduced the repulsive force to the dynamic multi-swarms, which focused on the dynamic multi-peak optimization problem. The repulsion mechanism was added to ensure the optimal solution tracked only by one swarm that repulsed other swarm for more peaks, but the attractive force was not used.

III. BASICS OF PSO AND SL-PSO

In the canonical PSO, each solution is regarded as a point in the search space, called a ''particle''. Each particle has the fitness determined according to the optimization problems. There is a velocity property for each particle, and it is dynamically updated according to the flying experience of the particles and the population. In each generation, each particle follows two other particles. The first one is its own historical optimal particle, called the personal optimal and labeled as *Pbest*. The second one is the optimal particle that the entire population has found so far, called the global best position and denoted as *Gbest*. Particles update their statuses according to the following formulas as shown in Equation (1) and Equation (2).

$$V_{i,j}(t+1) = w \cdot V_{i,j}(t) + c_1 \cdot r_1 \cdot (Pbest_{i,j}(t) - X_{i,j}(t)) + c_2 \cdot r_2 \cdot (Gbest_j(t) - X_{i,j}(t)) \quad (1)$$

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

where $j = 1, 2, \ldots, D$, represents the dimension; $i = 1, 2, \ldots, m$, represents the particle; $X_{i,j}$ represents the positions for the jth dimension of the ith particle; $V_{i,j}$ represents the velocity of the jth dimension of the ith particle; r_1 and r_2 are random values uniformly distributed in [0,1]; c_1 and c_2 are the acceleration coefficients; and w is the inertia weight used to balance between flight inertia and changing momentum.

SL-PSO is inspired by social learning behaviors, which apply the learning behavior of ''imitation'' to PSO algorithms. The SL-PSO algorithm changes the selection mechanism of the canonical PSO, in which particles select *Gbest* and *Pbest* to learn, while in SL-PSO, the poor particles (called as imitators) learn from good particles (called as demonstrators).

The SL-PSO algorithm includes three components, namely fitness evaluation, swarm sorting, and behavior learning. Firstly, we calculate the fitness of each particle to sort the population. In the sorted population, the ith particle learns from the kth particle who dominates it according to the fitness. So the kth demonstrator is randomly selected from the (i+1)th to mth particles in the sorted population as the learning object. In each generation, a particle maybe acts as a demonstrator for different imitators more than once. The specific formula is shown in Equation (3).

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}(t) + \Delta X_{i,j}(t+1), & if\ p_i(t) \le p_i^L \\ X_{i,j}(t), & otherwise \end{cases} \quad (3)$$

where $X_{i,j}(t+1)$ is the particle behavior modification, which is composed of three components, the inertial component, the imitation component, and the social influence component as shown in Equation (4), where $X_{i,j}(t)$ is the inertial component same as that in the canonical PSO; $I_{i,j}(t)$ is the imitation component aiming to learn from demonstrators in the current swarm as shown in Equation (5); $C_{i,j}(t)$ is the social influence component whose purpose is to learn from the collective behavior of the swarm as shown in Equation (6); p_i^L is the learning probability as shown in Equation (7); and r_1, r_2, r_3 are random values within (0,1). In Equation (5), $k \in (i+1, m)$ is the particle sequence of the demonstrator. In Equation (6), $\overline{X_j(t)}$ is the mean position and calculated according to Equation (8). In Equation (7), M is the base swarm size, n is the dimension, α is a constant, empirically, $\alpha = 0.5$.

$$\Delta X_{i,j}(t+1) = r_1(t) \cdot \Delta X_{i,j}(t) + r_2(t) \cdot I_{i,j}(t) + r_3(t) \cdot \varepsilon \cdot C_{i,j}(t) \quad (4)$$

$$I_{i,j}(t) = X_{k,j}(t) - X_{i,j}(t) \quad (5)$$

$$C_{i,j}(t) = \overline{X_j(t)} - X_{i,j}(t) \quad (6)$$

$$p_i^L = (1 - \frac{i-1}{m})^{\alpha \cdot \log(\lceil \frac{n}{M} \rceil)} \quad (7)$$

$$\overline{X_j(t)} = \frac{\sum_{i=1}^{m} X_{i,j}}{m} \quad (8)$$

In each iteration of PSO, the swarm go through three components including fitness evaluation, swarm sorting, and behavior learning to evolve to the next iteration until the satisfactory results are achieved. The SL-PSO algorithm abandons the mechanism of saving *Pbest* and instead uses the demonstrator as the learning object. Therefore, the complexity of the algorithm is reduced, and the value of the acceleration factor does not need to be considered. Although the algorithm is simple and applicable, it can still be improved for the following reasons: (1) The exchange of spatial information between particles, especially the communication of the distance information between particles, is not considered; (2) During the process of the particle evolution, particles learn from a random better particle in each iteration, resulting in slow convergence. Therefore, the SL-PSO algorithm is suitable for uni-modal functions, and for complex multi-modal functions, its performance is not competitive.

IV. MOLECULAR INTERACTION BASED SL-PSO (MISL-PSO)

This section introduces the main idea of this paper and the improvement strategies in detail of molecular interactions based SL-PSO (MISL-PSO). Its time complexity and convergence are also analyzed.

A. FRAMEWORK OF MISL-PSO

Learning plays an important role in PSO. Humans, at different ages, learn in various ways. It is reported that children learn by imitation. Then imitative learning gives way to the conditional learning when children grow up. With the conditional learning, one can quickly master a series of social knowledge

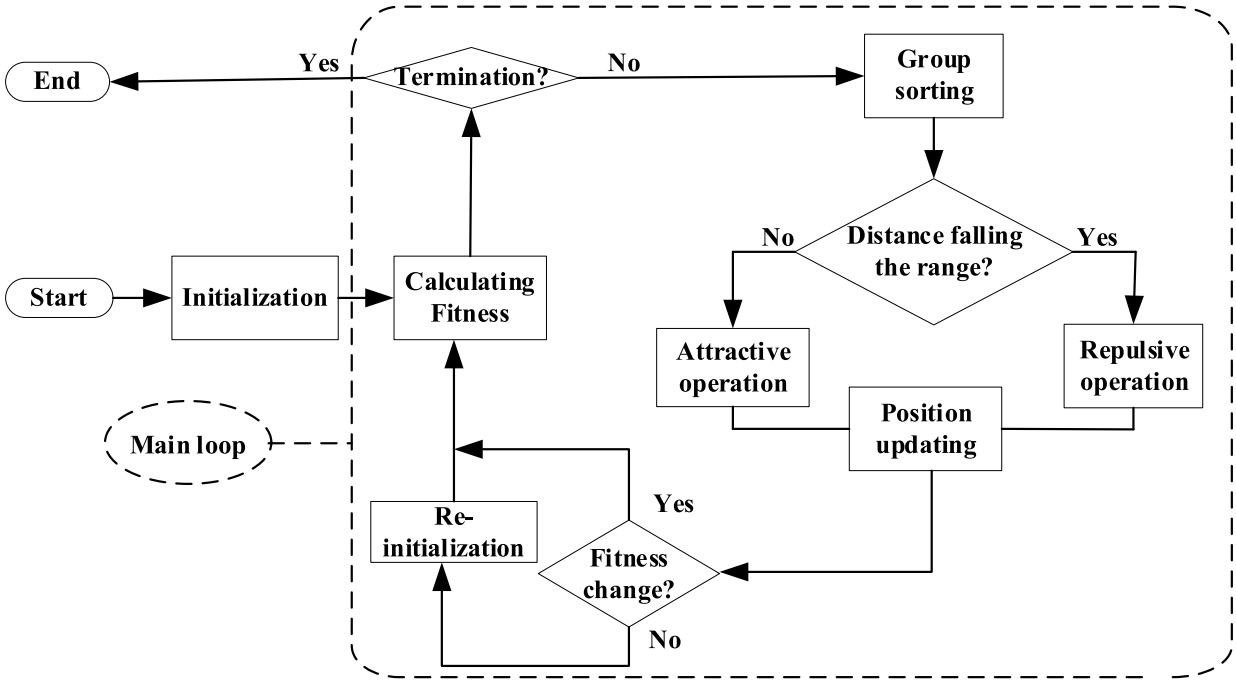

FIGURE 1. Flowchart of the Proposed MISL-PSO.

and skills [46]–[48]. The SL-PSO algorithm mainly uses the imitation learning to improve the selection mechanism of the algorithm. However, the blind imitation will reduce the diversity of the population, resulting in the decline of the ability to jump out from the local optimum. Therefore, we consider the conditional learning and set up the information interaction mechanism to make feedbacks among particles to balance the diversity and convergence. The conditional learning [10] is characterized by adding conditional restrictions within a certain range. It can be used for avoiding premature by preventing particles from agglomerating, which helps control the diversity of population.

In order to balance the tradeoff between the convergence and the diversity of PSO, we introduce a molecular force field, by which there are always repulsive forces and attractive forces. With repulsive forces, the molecules move away from each other after collisions when they fall into the repelling radius. And when out of this radius, the attraction force takes effect. Through this mechanism, when there is no other external forces, large-scale aggregation or diffusion of particles will be prevented.

We put forward molecular interactions based PSO by introducing the attraction and repulsion to balance the convergence and the diversity. The implementation of our algorithm is based on the SL-PSO framework. Fig. 1 shows the flowchart of the MISL-PSO algorithm. From the figure, we can see there are five steps in the MISL-PSO algorithm: (1) calculating the fitness of each particle; (2) sorting the particles according to their fitness values; (3) comparing the distances among particles; (4) applying attractive operations or repulsive operations according to the distance; (5) finally, executing the initialization strategy based on the total fitness value.

The MISL-PSO algorithm takes advantage of molecular interaction forces to adjust the position of particles to achieve a balance between the convergence and the diversity. For this goal, three key issues need to be studied: (1) the determination of the range of the particle interaction fields; (2) the determination of attraction forces between particles; (3) the determination of repulsion forces between particles.

B. THE RANGE OF PARTICLE INTERACTION FIELDS

In the PSO algorithm, most particles distribute near the local optimal and the global optimal. Once particles aggregate, it is difficult to disperse without the external forces, which results in the decline of the exploration ability. To this end, it is necessary to consider the adjustment of their positions by particles interactions carried out by the attraction and the repulsion. The first thing is to determine the distance range for the trigger of interaction. Euclidean distance is considered for particles X_i and X_j as shown in Equation (9).

$$dis(X_i, X_j) = \|X_i - X_j\| = \sqrt{\sum_k (X_{i,k} - X_{j,k})^2} \qquad (9)$$

Based on this distance of particles, we can determine the force field range with a radius R. When the distance between two particles falls in the range, a repulsive action will be trigged and the force brings the change of both the direction and the position, which thereby increases the diversity.

Empirically, the force field decreases too quickly, so that diversity cannot be maintained; if the range of force field changes too slowly, the algorithm will fail to converge and skip the optimal solution. Considering the linear relationship between the convergence speed and the time, the size of the force field will also decrease linearly with the time, as shown in Equation (10).

$$\begin{cases} R = U - U \cdot \dfrac{t}{0.5 \cdot T_{\max}}, & t \le 0.5 \cdot T_{\max} \\ R = 0, & t > 0.5 \cdot T_{\max} \end{cases} \tag{10}$$

where t is the current number of iterations; T_{max} is the specified maximum number of iterations. When t is 0, R reaches its maximum value U; when t is $0.5T_{max}$, R reaches its minimum value 0. When $t > 0.5T_{max}$, the force field is 0, there is no repulsion. However, at this stage, when the total fitness of the swarm is not changing, we use the re-initialization strategy. The process of the intermolecular interaction can be viewed as an optimization of the rearrangement of molecules in the expectation of less energy. Therefore, this paper adds an initialization strategy based on fitness changes. According to the idea of the nature of molecular forces, we initialize all but the optimal particles of the particle swarm when the total potential energy or the total fitness, is not changing. This is a repulsive strategy based on the fitness change.

U is defined as the maximum collision distance in the solution space, which is related to the boundary range of the function. Suppose that the kth dimension of the search space has its upper value ub_k and a lower value lb_k, then we formulate U as shown in Equation (11).

$$U = \max_k \frac{ub_k - lb_k}{m}, \quad k = 1, 2, \cdots, n \tag{11}$$

where n is the number of dimension of the search space.

C. ATTRACTION BETWEEN PARTICLES

There is the attractive force among particles. The value of the attractive force is not only related with the distance but also with the fitness: its value increases with the decrease of the distance and increases with the fitness of the particles, which ensures the learning from better particles.

In the SL-PSO algorithm, the imitated objects are randomly selected from better demonstrators. However, the random selection does not identify which one is better, and the blindly optimal learning will quickly fall into the local optimum which reduces the diversity of the population. Therefore, we consider the attractive forces among particles. A fixed attraction probability p_k is used. When the random value is less than p_k, the random imitation learning behavior is performed, otherwise, the best particle m is treated as the learnt object. This can be formulated as Equation (12).

$$\begin{aligned} X_{i,j}(t+1) &= X_{i,j}(t) + \Delta X_{i,j}(t+1) \\ \Delta X_{i,j}(t+1) &= \begin{cases} X_{m,j}(t) - X_{i,j}(t), & r > p_k \\ X_{k,j}(t) - X_{i,j}(t), & r < p_k \end{cases} \end{aligned} \tag{12}$$

where particle k is the selected imitator. The algorithm for particles attraction is given in Table 1.

TABLE 1. Particle attraction algorithm.

Input:	A sorted swarm and the specified p_k
Output:	Demonstrator particle
Functions:	Find the particles imitated;

```
r4=rand(0,1);//generate a random number
  if r4 <pk
    k= rand(i+1,m);//generate learnt objects randomly
  else
    k = m;//particle m is the best one in the current swarm
  end
```

D. REPULSION BETWEEN PARTICLES

The repulsive force between molecules can be ignored if their distances exceed a certain threshold. Otherwise, molecules will make a collision, repulse and rebound. The way to bounce determines whether the diversity of population has been increased and whether the search space can be more effectively explored.

We consider the repulsive force between particles from two aspects: (1) Consider a wide range of bounces for some particles to move far, therefore to improve the diversity of the population. (2) Consider a small range of bounces for other particles to avoid destroying the convergence of the algorithm itself. The value of the repulsive force is calculated as follows. When particles i and j collide with each other, particle i with dominated fitness performs a small range shock on particle j with the dominating fitness. Thus, the position update range of particle i is smaller than that of particle j, taking both exploitation and exploration into consideration. This is formulated as Equations (13) and (14).

$$X_{i,k}(t+1) = -r_5 \cdot X_{i,k}(t) - r_6 \cdot \Delta X_{i,k}(t+1) \tag{13}$$

$$X_{j,k}(t+1) = r_6 \cdot X_{j,k}(t) - r_5 \cdot \Delta X_{j,k}(t+1) \tag{14}$$

where r_5 and r_6 are random values in (0,1) and k stands for the dimension. It is noted that Equation (12) and Equation (13) are the improvements on Equation (3) according to molecular repulsion. Where $X_{i,k}(t+1)$ and $X_{j,k}(t+1)$ are the particle behavior modification, as is shown in Equation (4).

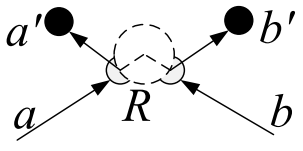

FIGURE 2. Particle Repulsion Process.

To better understand the collision motion of particles, Fig. 2 shows the molecular collision rebound, where a and b

are two moving molecules. The collision occurs when their distance reaches the threshold of the repulsive rebound. In Fig. 2, *a*' and *b*' are the positions after the collision.

Fig. 2 shows the actual physical repulsion process. On this basis, this paper proposes an exclusion strategy. Considering that the PSO algorithm is a process of random search, random factors are added to Equations (12) and (13) to better realize the process of exploration and exploitation.

E. MISL-PSO ALGORITHM

The algorithm consists of two steps. The first step is to initialize the particle population and define the operator parameters. The second step is to perform iterative learning. First, the swarm is sorted according to the fitness, then the Euclidean distance is calculated to determine whether the attraction strategy shown in Equation (12) or the exclusion strategy shown in Equation (13) and Equation (14) is implemented. Table 2 shows the details of the MISL-PSO algorithm.

TABLE 2. MISL-PSO algorithm.

Input:	Problem to be optimized
Output:	Global optimal particle and its fitness
Functions:	Find the best solution for the problem
Step 1:	Initialize maximum number of iterations T_{max}, particle size m, the position X_i and velocity V_i for each particle; and set the parameters such as ε, p_i^L, U, M, α and p_k.

Step 2:	**While** $t < T_{max}$
	Evaluate the fitness and sort particles;
	Set R of particles according to Equation (9);
	Calculate the Euclidean distance among the particles according to Equation (8);
	if $dis(X_i,X_j) < R$ && $t < 0.5T_{max}$
	update the position of particle i according to (13);
	update the position of particle j according to (14);
	else
	update the position of particle i according to (3);
	Call the particle attraction algorithm in Table 1;
	evaluate the position;
	if the total fitness not change, initialize all but the optimal particles of the particle swarm;
	end if
	$t = t + 1$
	End while
	Output the best individual and its fitness

F. CONVERGENCE AND TIME COMPLEXITY ANALYSIS

Cheng and Jin [10] had given the proof of convergence for the SL-PSO algorithm. Based on the SL-PSO algorithm, the attractive mechanism in our algorithm accelerates the convergence, and the repulsive mechanism is included to keep the diversity. Therefore, we take the attractive and repulsive mechanism in the analysis based on Cheng and Jin's proof for the convergence.

The update method of performing the random imitation learning behavior is the same as that in the SL-PSO algorithm. When learning from the optimal particle under a certain probability, we replace the random numbers r_1, r_2, and r_3, with their mathematical-expected value 0.5, and then Equation (4) can be rewritten as Equation (15).

$$\Delta X_{i,j}(t+1) = 0.5 \cdot \Delta X_{i,j}(t) + 0.5 \cdot (X_{m,j}(t) - X_{i,j}(t)) + 0.5 \cdot \varepsilon \cdot (\overline{X_j(t)} - X_{i,j}(t)) \quad (15)$$

Compared with [10], it is equivalent to turning particle X_k in Equation (12) into particle X_m in Equation (11), and the proof is similar. Therefore, it can be concluded that the process must converge. So, MISL-PSO can converge to the global optima.

MISL-PSO mainly contains 4 components: the fitness calculation, the group sorting, the attraction and repulsion mechanism, and learning behaviors. According to [10], the complexity of the fitness assessment stage is different due to various optimization problems, so we do not consider the optimized problem factor here. In the fitness ranking stage, the time complexity is $T_s = O(m^2)$, which is for the fitness ranking stage, and m is the population size. In the learning behavior stage, the time complexity is $T_b = O(mn)$, which is for the learning behavior stage, m is the population size, and n is the decision variables' dimension.

Now let's consider the attraction and repulsion stages. The model of the attraction mechanism is relatively simple, in which only one judgment is added. Compared to SL-PSO, our method doesn't adopt the random selection when learning from the optimal particles, and the time complexity of the two algorithms can be considered as the same.

There are two main steps in the repulsion stage: (1) calculation of Euclidean distance, which is related to the number of particles m; (2) update the position of the repulsive behavior, which is related to m and the dimension n. Therefore, the time complexity of our method is shown in Equation (16).

$$T = O(\tfrac{1}{2} \cdot m \cdot (m-1) + m \cdot n) + O(m^2) + O(mn) \quad (16)$$

V. EXPERIMENTS AND RESULTS

In order to examine the performance of MISL-PSO, the experiments mainly consider two kinds of optimization functions, low-dimensional functions and high-dimensional functions. For the comparison of the performance, an overall ranking evaluation method is given. The best performance is highlighted in bold. According to the t-test (significance level $\alpha = 0.05$) among MISL-PSO and state-of-the-art evolutionary algorithms, the signals of ' 1 ' ' −1 ' and ' 0 ' in the last column for each function indicate whether or not MISL-PSO performs significantly better, significantly worse, or comparably in comparing with its competitors. All the results are in 30 independent runs.

A. ALGORITHMS AND BENCHMARK FUNCTIONS

Except for the MISL-PSO algorithm, the SL-PSO algorithm and the AR-GSA algorithm, a state-of-the-art PSO variant is selected for the comparison, which uses ring topology and elitist learning as non-greedy strategies in place of the traditional update method based on *Gbest* and *Pbest*, called the elitist learning PSO algorithm with scaling mutation and ring topology(LSERPSO) [49]. These algorithms are classic and excellent algorithms that have been verified by experiments, and have excellent performance on low-dimensional and high-dimensional multi-modal problems. For the convenience of comparisons, we use the same parameters settings in [10] for the SL-PSO. Besides, the population size and the maximum number of fitness evaluations are set to the same value for all algorithms. Other parameters are set as the recommendation references.

Totally 49 benchmark functions are applied in this work. Among them, 42 low-dimensional functions and 7 high-dimensional complex functions are included. These 42 low-dimensional functions are composed of 12 popular benchmarks and 30 benchmarks from CEC'17 [50] function set. Besides, 7 high-dimensional functions include CEC'08 [51] on large-scale complex optimization problems. The optima fitness for all of these benchmark functions is 0. In the following tests, 12 low-dimensional benchmark problems with dimension 30 are verified firstly, and then followed by 30 low-dimensional functions with dimension 50. Finally, 7 high-dimensional functions with dimensions of 100, 500, and 1,000 respectively are included.

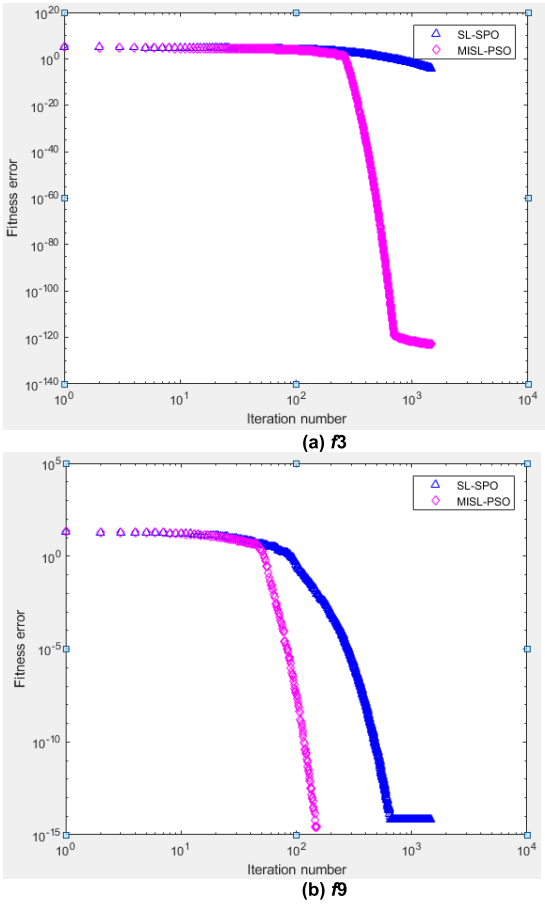

FIGURE 3. The Convergence Profiles.

B. LOW-DIMENSIONAL BENCHMARK FUNCTIONS

For the 42 low-dimensional functions, we label the 12 popular benchmarks from $f1$ to $f12$, among which, $f1$ to $f5$ are single-modal functions, $f6$ is a discontinuous step function, and $f7$ to $f12$ are multi-modal functions. The expressions and ranges of the 12 benchmark functions are described in details in [10]. The MISL-PSO algorithm, the SL-PSO algorithm and the above 2 algorithms were run for $f\ 1 \sim f\ 12$.

These 12 functions have no operations such as rotations and translations. Experiments show that the performance of MISL-PSO is significantly improved compared with other algorithms. As showed in Table 3, in addition to $f\ 5, f\ 7, f\ 8$, $f\ 10$, and $f\ 11$, the results of other test functions are significantly improved. Because of the attractive force, the results of this algorithm will not be worse than the original algorithm. For each algorithm, the average value is represented by "mean" and the variance is represented by "std". In order to further justify the above conclusion, the convergence profiles of one typical uni-modal function $f3$ and one typical multi-modal function $f\ 9$ are plotted in Fig. 3.

It can be seen from Table 3 and Fig. 3 that the performance of the proposed algorithm is significantly improved on the 12 low-dimensional benchmark functions. For the uni-modal problems $f1 \sim f8$ and $f\ 11$, the MISL-PSO algorithm gains improvements compared with the SL-PSO algorithm. For the multi-modal functions $f\ 9, f\ 10$, and $f\ 12$, great improvements have also been made.

In order to further compare the performance of the MISL-PSO algorithm with SL-PSO, AR-GSA and LSERPSO algorithms, 30 test functions from the CEC'17 functions set are selected. Among them, $f13 \sim f15$ are unimodal functions; $f16 \sim f22$ are simple multi-modal functions; $f23 \sim f32$ are hybrid functions; $f33 \sim f40$ are composition functions. The result is shown in Table 4. It can be seen that the proposed algorithm outperforms other algorithms on $f14 \sim f16$, $f21$, $f23 \sim f27$, $f29 \sim f31$ and $f41$, $f42$, while the improvement on the other optimization problems is not obvious. We give the reasons as follows: (1) Most of CEC'17 benchmark functions are composition functions with rotation and translation operations. For composition functions, the algorithm doesn't have obvious advantages due to its learning strategy. (2) The proposed algorithm tries to make it out of local optimization as much as possible and strengthen local search. However, because the repulsion is randomly carried out, and the algorithm improvement is decoupled from the dimension, so the effect is not obvious.

For test functions of f23 $\sim$ f32, as shown in Table 4 and Fig. 4, MISL-PSO has achieved new improvements. It can be seen that the performance of the MISL-PSO algorithm on the hybrid test functions is beyond expectation. Therefore,

TABLE 3. Results on Test Functions f1 ∼ f12 (D = 30).

	MISL-PSO		SL-PSO			AR-GSA			LSERPSO		
	mean	std	mean	std		mean	std		mean	std	
$f1$	**1.07E-222**	**0.00E+00**	6.24E-67	1.19E-66	1	1.81E-19	3.27E-20	1	5.86E-55	1.86E-55	1
$f2$	**8.11E-114**	**9.70E-114**	6.99E-35	5.46E-35	1	2.47E-09	2.62E-20	1	6.44E-28	6.84E-28	1
$f3$	**9.19E-82**	**5.29E-20**	3.00E-04	3.21E-04	1	2.23E+02	1.30E-01	1	9.21E-04	2.21E-04	1
$f4$	**1.41E-93**	**2.58E-93**	2.90E-18	1.66E-18	1	2.47E-10	2.64E-10	1	4.26E-13	1.26E-13	1
$f5$	**2.42E+01**	**7.36E-01**	2.54E+01	1.38E+01	0	2.51E+01	1.74E+00	0	2.40E+01	1.40E+01	0
$f6$	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	0	0.00E+00	0.00E+00	0	0.00E+00	0.00E+00	0
$f7$	**1.43E+03**	**3.72E+02**	1.51E+03	4.18E+02	0	8.99E+03	4.97E+02	0	1.30E+03	1.00E+03	0
$f8$	**1.22E+01**	**3.35E+00**	1.40E+01	4.57E+00	0	1.29E+01	8.08E+00	0	6.96E+01	1.96E+00	1
$f9$	**-8.88E-16**	**1.00E-31**	5.39E-15	1.53E-15	1	3.66E-10	1.00E-31	1	2.66E-15	6.65E-16	1
$f10$	**0.00E+00**	**0.00E+00**	4.11E-04	2.25E-03	1	0.00E+00	0.00E+00	0	0.00E+00	0.00E+00	0
$f11$	**1.57E-32**	**1.11E-47**	1.57E-32	1.11E-47	0	2.36E-21	1.11E-23	1	1.57E-32	3.57E-32	0
$f12$	**1.35E-32**	**5.57E-48**	3.66E-04	2.01E-03	1	3.03E-20	5.57E-22	1	1.35E-32	1.15E-32	0

we can say that the molecular interaction mechanism can not only improve the performance of the SL-PSO algorithm, but also show well performance on those optimization functions that SL-PSO cannot well deal with.

In order to compare the performance of the algorithms more comprehensively, the average time for each algorithm to run all test functions in different dimensions is plotted in Fig. 5, where the vertical coordinate is the average running time in seconds. It can be seen that the running time of the SL-PSO is the shortest, and that of the MISL-PSO is in the middle position. For problems with the dimension 30, although the time complexity of the MISL-PSO algorithm increases, its time is not very long. What's more, the optima it gains is better than other algorithms, as shown in Table 3. While for the problems with the dimension 50, the difference between the running time of the MISL-PSO algorithm and the SL-PSO algorithm is small and the improvement is also obvious, as shown in Table 4.

Besides, in order to obtain better algorithm performance, we have carried out an experimental test on the probability parameter p_k. Four representative functional problems are selected, including a low-dimensional uni-modal problem $f8$, a low-dimensional simple multi-modal problem f 20, a low-dimensional complex rotation problem f 35, and a high-dimensional complex combination function problem $f45$.The results are shown in Fig. 6. In the figure, the algorithm works best when $p_k = 0.9$. Besides, it can be seen from the experiment that this parameter is not sensitive to problems, which increases the availability of the framework.

C. HIGH-DIMENSIONAL TEST FUNCTIONS

In order to further test the performance of the improved algorithm on high-dimensional functions, we use the 7 high-dimensional test functions proposed on CEC'08. Table 5 shows the test results when the dimension is 100, 500, and 1,000 respectively. From the table, we can see that for the dimension of 100, MISL-PSO performs better than SL-PSO for all the functions. When the dimension increases to 500, the performance of MLSI-PSO is better than SL-PSO for four out of seven functions. After the dimension reaches 1,000, the average fitness values of MISL-PSO is better than those of SL-PSO for five of seven functions. Therefore, MISL-PSO perform better than SL-PSO for most of seven high-dimensional functions.

D. AVERAGE RANK

In this section, we compare the performance of each algorithm. For $f1 \sim f12$ functions, Fig. 7 shows the Friedman test chart at a significance level of 0.05, in which the horizontal axis is the average sequence value, and the vertical axis is displayed for each algorithm used in this article. The abscissa corresponding to "$*$" is the average sequence value of the corresponding algorithm; the line segment centered on "$*$" represents the length of the critical value domain CD. If the line segments corresponding to the two algorithms do not overlap, it indicates that the two algorithms are significantly different, and the algorithm with a small average order value is significantly better than the algorithm with a large average order value. If the line segments corresponding to the two algorithms overlap, then there is no significant difference. We do the same ranking and average sequence value calculations for 30 low-dimensional functions with the dimension 50 too as shown in Fig. 8. From Fig. 9, we know that MISL-PSO has the first average rank for all the scenarios in low-dimensional functions. What's more, we can see from Fig. 8 and Fig. 9 that MISL-PSO is significantly better than other algorithms.

For the 7 high-dimensional functions, we calculate their average value for each function. Then we use Friedman test

TABLE 4. Results on Test Functions of CEC'17 (D = 50).

	MISL-PSO		SL-PSO			AR-GSA			LSERPSO		
	mean	std	mean	std		mean	std		mean	std	
*f*13	**1.51E+04**	**8.04E+03**	2.32E+05	6.88E+05	1	1.70E+11	1.24E+10	1	2.04E+11	3.54E+10	1
*f*14	**8.56E+02**	**2.08E+03**	1.01E+09	5.30E+09	1	6.61E+44	9.46E+44	1	6.59E+90	3.48E+91	1
*f*15	**8.90E+04**	**1.40E+04**	1.80E+06	4.84E+06	1	3.63E+07	1.09E+08	1	1.32E+07	2.78E+07	1
*f*16	**5.28E+02**	**5.75E+01**	9.73E+03	2.44E+04	1	3.07E+04	3.12E+03	1	8.91E+04	2.08E+04	1
*f*17	**5.83E+02**	**1.86E+01**	8.53E+02	2.81E+02	0	1.26E+03	3.96E+01	1	1.53E+03	9.24E+01	1
*f*18	**6.00E+02**	**1.76E-01**	6.20E+02	5.22E+01	0	7.23E+02	6.74E+00	0	7.45E+02	9.49E+00	0
*f*19	**8.18E+02**	**2.17E+01**	1.51E+03	1.09E+03	0	3.61E+03	1.76E+02	0	4.50E+03	1.08E+03	1
*f*20	**8.83E+02**	**2.72E+01**	1.14E+03	2.90E+02	0	1.45E+03	3.46E+01	0	1.85E+03	8.56E+01	0
*f*21	**9.12E+02**	**2.79E+01**	1.25E+04	3.03E+04	1	3.66E+04	3.98E+03	1	8.97E+04	9.46E+03	1
*f*22	1.40E+04	3.54E+02	**5.08E+03**	**4.87E+03**	-1	8.81E+03	2.44E+02	0	1.75E+04	5.21E+02	0
*f*23	**1.30E+03**	**1.02E+02**	1.27E+04	2.96E+04	1	1.54E+05	3.34E+04	1	1.07E+05	5.91E+04	1
*f*24	**2.16E+06**	**1.29E+06**	1.62E+10	4.22E+10	1	2.20E+10	1.96E+09	1	1.29E+11	2.29E+10	1
*f*25	**2.68E+04**	**1.24E+04**	9.68E+09	2.60E+10	1	6.22E+09	8.17E+08	1	6.89E+10	1.39E+10	1
*f*26	**3.32E+04**	**2.74E+04**	2.78E+07	7.87E+07	1	1.04E+07	2.21E+06	1	1.98E+08	1.06E+08	1
*f*27	**9.78E+03**	**5.54E+03**	2.47E+09	6.68E+09	1	2.32E+09	4.00E+08	1	2.46E+10	7.52E+09	1
*f*28	**2.80E+03**	**4.21E+02**	3.66E+03	3.84E+03	0	3.42E+04	2.36E+03	1	1.32E+04	2.10E+03	1
*f*29	**2.54E+03**	**3.62E+02**	4.26E+04	1.24E+05	1	1.02E+05	2.08E+04	1	3.94E+05	3.13E+05	1
*f*30	**2.17E+05**	**9.47E+04**	1.07E+08	3.22E+08	1	5.46E+07	1.98E+07	1	7.27E+08	4.08E+08	1
*f*31	**1.81E+04**	**1.12E+04**	1.25E+09	3.44E+09	1	2.08E+09	6.89E+08	1	9.05E+09	2.85E+09	1
*f*32	3.09E+03	3.76E+02	2.67E+03	1.14E+03	0	3.11E+03	8.16E+01	0	**5.13E+02**	**3.49E+01**	-1
*f*33	2.39E+03	2.33E+01	2.68E+03	3.25E+02	0	**4.57E+02**	**5.19E+02**	-1	3.47E+03	8.57E+01	0
*f*34	1.47E+04	2.41E+03	6.21E+03	5.25E+03	0	**4.66E+02**	**1.55E+02**	-1	1.91E+04	5.11E+02	0
*f*35	2.82E+03	2.83E+01	3.05E+03	7.39E+02	0	**5.42E+02**	**6.88E+02**	-1	4.99E+03	2.31E+02	0
*f*36	2.99E+03	2.25E+01	3.29E+03	7.97E+02	0	4.12E+03	5.56E+01	0	**5.13E+02**	**3.49E+01**	-1
*f*37	**3.04E+03**	**3.78E+01**	7.75E+03	1.24E+04	0	1.24E+04	1.35E+03	0	4.30E+04	1.07E+04	1
*f*38	**4.59E+03**	**2.31E+02**	7.32E+03	8.48E+03	0	2.36E+04	1.83E+03	1	2.73E+04	3.21E+03	1
*f*39	3.36E+03	6.31E+01	3.31E+03	6.53E+01	0	3.85E+03	8.44E+01	0	**3.20E+03**	**5.19E-05**	0
*f*40	3.30E+03	1.68E+01	**3.29E+03**	**1.94E+01**	0	7.56E+03	4.96E+02	0	3.30E+03	5.08E-05	0
*f*41	**3.68E+03**	**1.93E+02**	1.11E+05	3.52E+05	1	1.05E+04	3.15E+03	0	8.85E+05	8.76E+05	1
*f*42	**1.28E+06**	**3.06E+05**	1.52E+09	4.27E+09	1	6.36E+09	3.12E+08	1	1.64E+10	4.41E+09	1

(significance level $\alpha = 0.05$) to compare the correlation between MISL-PSO and SL-PSO as shown in Fig. 9. It can be seen that the ranking of MISL-PSO is also the better. In conclusion, compared with other algorithms, the performance of MISL-PSO has been greatly improved.

VI. DISCUSSIONS

In this paper, we apply the molecular interaction mechanism into SL-PSO to improve the performance. The experimental results show that the proposed MISL-PSO outperforms SL-PSO, the classical PSO algorithms, and other representative evolutionary algorithms. Below we discuss the related issues in this work.

Although the similar mechanism has been applied to the traditional PSO algorithm, it only uses simple concepts of the attractive force and the repulsive force, in which the attractive force is the learning from other particles in PSO, and the repulsive force represents moving away from worse particles. They lack of the real integration of molecular concepts and calculation into PSO. In our work, we integrate the molecular mechanism including the attractive force and the repulsive force into SL-PSO. The attractive force is more flexible since particles learn from better particles or the global optimal according to the predefined threshold. This prevents the premature problem. The repulsive force is decided by the physical distance between particles, and the dominating particle has less shock than the dominated particle, thus to keep the diversity of particles. The introduction and integration of the attractive force and the repulsive force guarantee both the convergence and the diversity.

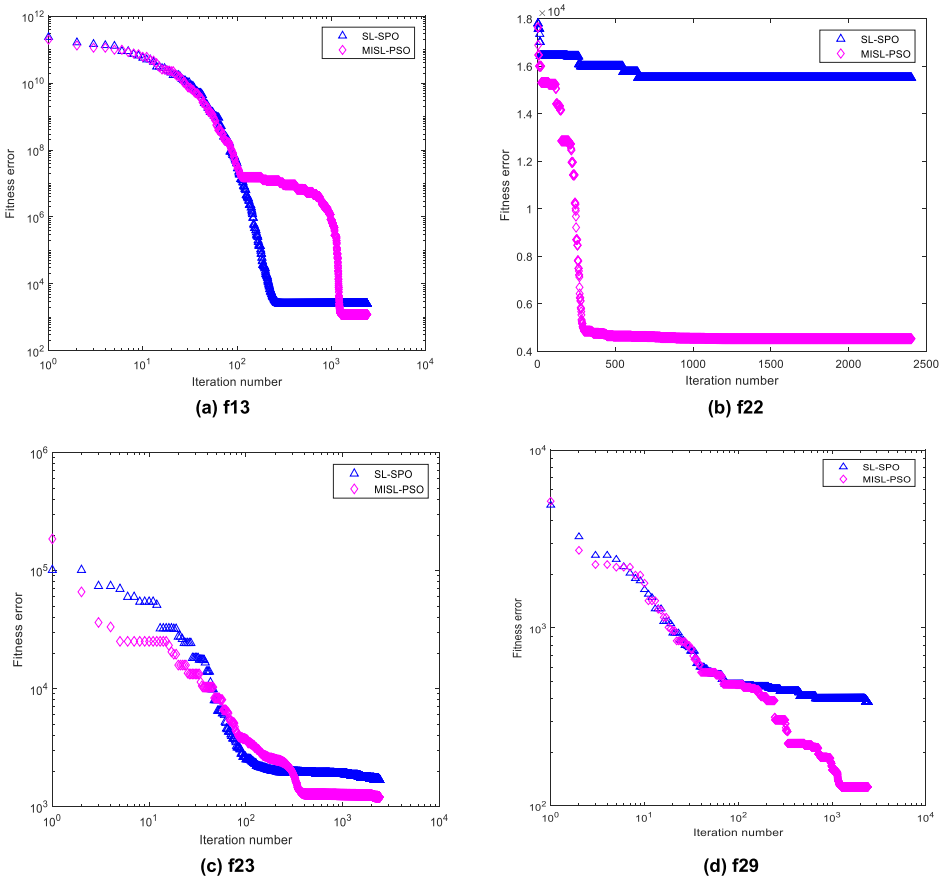

FIGURE 4. Results of MISL-PSO and SL-PSO on Functions f13, f22, f23, and f29.

TABLE 5. Test results at the CEC '08 function with a dimension of 100, 500, 1,000.

Function		MISL-PSO	SL-PSO		MISL-PSO	SL-PSO		MISL-PSO	SL-PSO	
		D=100			D=500			D=1,000		
$f41$	mean	**1.07E-27**	1.09E-27	0	1.17E-23	**7.58E-24**	-1	**7.10E-23**	7.49E-23	0
	std	**4.48E-28**	3.97E-28		5.03E-25	**2.44E-25**		**3.29E-24**	2.17E-24	
$f42$	mean	**1.90E-06**	5.11E-06	1	**3.47E+01**	3.56E+01	0	**8.13E+01**	8.95E+01	0
	std	**7.81E-06**	2.95E-06		**9.06E-01**	7.74E-01		**1.14E+01**	4.37E+00	
$f43$	mean	2.80E+02	**1.97E+02**	0	6.37E+02	**5.14E+02**	0	**1.01E+03**	1.02E+03	0
	std	3.06E+02	**2.83E+02**		2.33E+02	**4.38E+01**		**1.13E+01**	3.71E+01	
$f44$	mean	**7.52E+01**	7.88E+01	0	**7.28E+02**	2.88E+03	1	5.80E+02	**5.70E+02**	0
	std	**1.39E+01**	1.43E+01		**2.88E+01**	2.04E+02		3.36E+01	**2.43E+01**	
$f45$	mean	**5.75E-04**	2.86E-02	1	**3.14E-16**	3.37E-16	0	7.22E-16	**5.40E-16**	0
	std	**2.21E-03**	7.66E-02		**1.67E-17**	2.03E-17		7.85E-17	**3.84E-17**	
$f46$	mean	**1.87E-14**	2.18E-14	0	1.84E-13	**1.49E-13**	0	**3.38E-13**	3.51E-13	0
	std	**4.56E-15**	4.08E-15		5.27E-15	**3.01E-15**		**5.02E-15**	5.02E-15	
$f47$	mean	**1.49E+03**	1.48E+03	0	**6.93E+03**	5.03E+03	0	**1.38E+04**	1.31E+04	0
	std	**2.07E+01**	2.24E+01		**3.33E+01**	4.26E+01		**9.45E+01**	5.17E+01	

SL-PSO improves the traditional PSO by learning better particles instead of the local and global optima for easy parameter settings and improved performance. In this paper, based on SL-PSO, we introduce the molecular

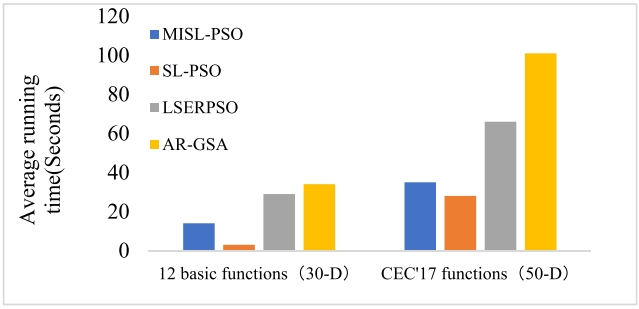

FIGURE 5. Average running time.

mechanism into SL-PSO to further improve the performance by enhancing both of the convergence and the diversity. The results show the proposed MLSL-PSO performs better than SL-PSO and most of other algorithms for all the compared low-dimensional functions. However, for high-dimensional functions, MLSL-PSO also has better performance than SL-PSO in general and has comparable performance to most of other algorithms. If we consider the overall performance by the overall score, MLSL-PSO achieves the best performance.

In the paper, we analyze the time complexity of the proposed MLSL-PSO algorithm and it has the same complexity as the SL-PSO algorithm. The experimental results show that the time consumption is comparable to the SL-PSO algorithm and especially when the dimension increases. This is consistent to our analysis. Therefore, the proposed algorithm improves the SL-PSO algorithm without significant computational costs. We will convey this issue in our further work.

In the attractive force, we decide the particles learn from better ones or the global optima by the threshold. Therefore, the threshold is an important factor for the performance. An inappropriate choice of the threshold will lead to the loss of the diversity or divergence. To achieve the balance of the diversity and the convergence, we calibrate the threshold using experiments. It has been shown that 0.9 is the best choice for the chosen bench functions with four different types of problems. In other functions, we use the calibrated threshold of 0.9. However, the choice of threshold may be different for various functions and applications. We will systemically study this for different types of problems in our future work.

In the repulsive force, the force field range is used to decide when particles should collide. The force field range decreases with the increase the number of iterations. We have done this to improve the exploitation. When they collide, they don't follow the rule of the physical collision of two objects. We set up two random values to control the speed and the direction noises, thus to increase both the exploration and the exploitation. Therefore, in both the attractive force and the repulsive force, we intend to achieve both of the convergence and the diversity.

This work integrates the molecular mechanism into SL-PSO and it is a framework to apply promising technologies into PSO. Although we applied the attractive force and

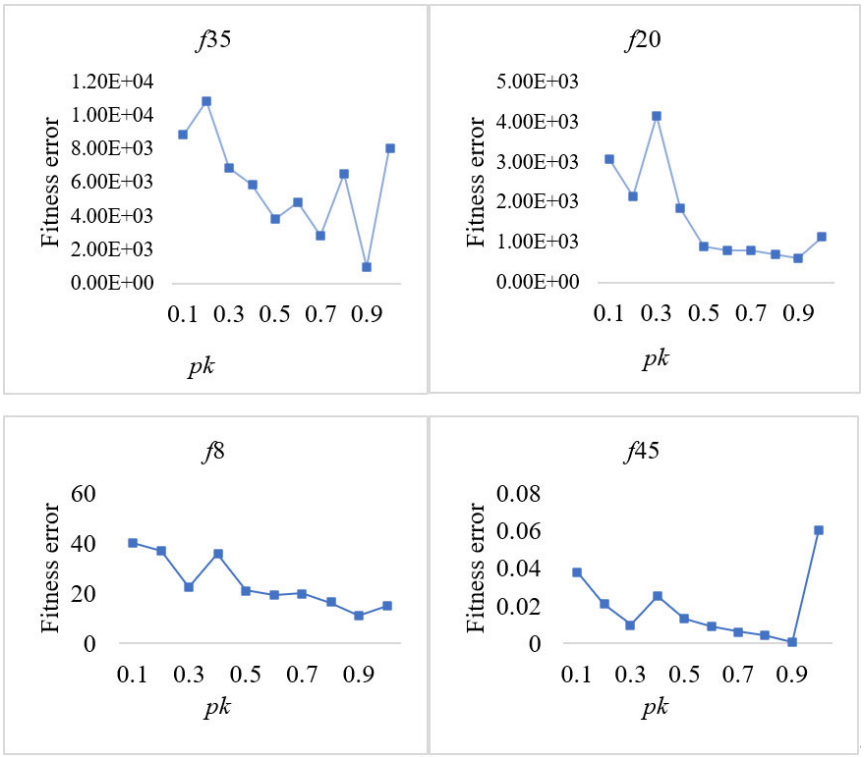

FIGURE 6. Tests on the Probability Parameter p_k.

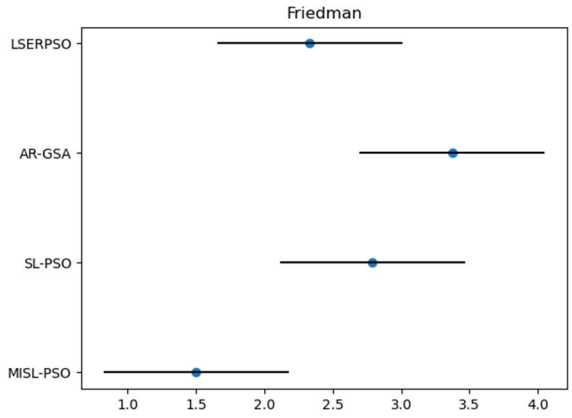

FIGURE 7. Friedman test of $f1 \sim f12$ functions at significance level of 0.05.

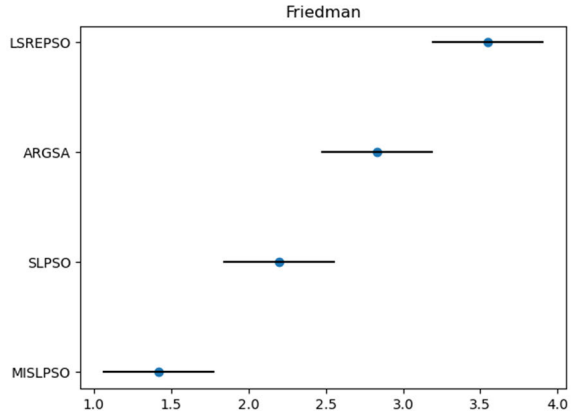

FIGURE 8. Friedman test of $f13 \sim f42$ functions at significance level of 0.05.

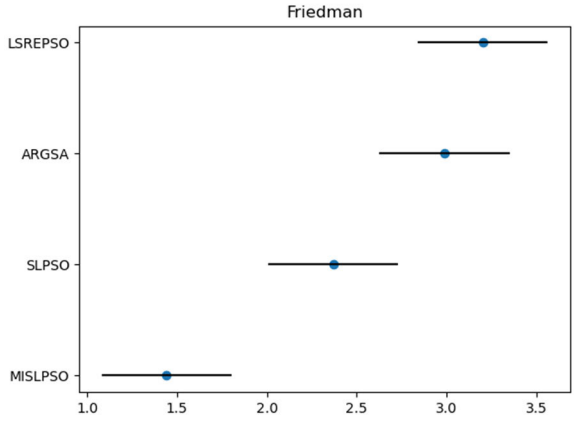

FIGURE 9. Friedman test of low dimension functions at significance level of 0.05.

the repulsive force into SL-PSO in this work, we can apply them into any other PSO algorithms and other evolutionary algorithms. This will enhance both of the diversity and the convergence. We choose SL-PSO since its overall performance is promising. We will investigate the combination and integration of the molecular mechanism into other improved swarm optimization algorithms in the future.

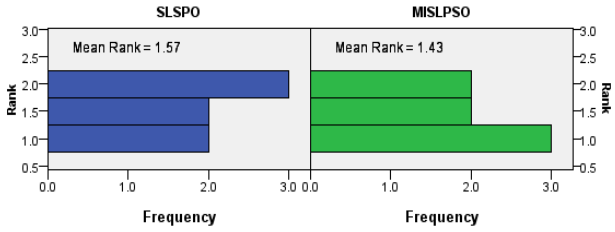

FIGURE 10. Friedman test of high dimension functions at significance level of 0.05.

VII. CONCLUSIONS AND FUTURE WORK

This paper introduces the molecular interaction into PSO to balance the convergence and diversity of the algorithm. Two operations simulating molecular forces are added to SL-PSO, the attractive operation and the repulsive operation. From various aspects, it is verified that the proposed algorithm outperforms most of the popular algorithms. The improved algorithm has better performance in dealing with low-dimensional problems and comparable performance for higher-dimensional complex problems. It also proves that the molecular interaction mechanism plays a key role in improving the optimization ability.

The molecular interaction mechanism is only a conditional learning method, and its time complexity is not obviously high. However, when this mechanism is combined with the algorithm, it can bring improvement without destroying the performance of the original algorithm. The further work to be carried out mainly includes: (1) Use new indicators, especially the domination landscape theory [52] to reduce the time complexity of the algorithm without losing the accuracy of control, (2) Construct a more efficient and concise attractive and repulsive operation framework applicable for most swarm algorithms to improve their performance.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN*, Perth, WA, Australia, Aug. 2002, pp. 1942–1948.
[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. MHS95. Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, Oct. 1995, pp. 39–43.
[3] G. Yuan and W. Yang, "Study on optimization of economic dispatching of electric power system based on hybrid intelligent algorithms (PSO and AFSA)," *Energy*, vol. 183, pp. 926–935, Sep. 2019.
[4] S. Guo, S. Zhang, J. Song, Y. Zhao, and W. Zhu, "Tuning approach for power system stabilizer PSS4B using hybrid PSO," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 192, Nov. 2018, Art. no. 012039.
[5] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "A particle swarm optimization-based flexible convolutional autoencoder for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2295–2309, Aug. 2019.
[6] E. Vellasques, R. Sabourin, and E. Granger, "Fast intelligent watermarking of heterogeneous image streams through mixture modeling of PSO populations," *Appl. Soft Comput.*, vol. 13, no. 6, pp. 3130–3148, Jun. 2013.
[7] H.-H. Tsai, Y.-J. Jhuang, and Y.-S. Lai, "An SVD-based image watermarking in wavelet domain using SVR and PSO," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2442–2453, Aug. 2012.
[8] S. Liu, Y. Liu, C. Wang, and J. Ren, "Marine asynchronous propulsion motor parameter identification using dynamic particle swarm optimization," in *Energy Development, Pts* (Advanced Materials Research), vols. 860–863, Q. Xu, Y. Li, and X. Yang, Eds. Hoboken, NJ, USA: Wiley-Blackwell, 2014, pp. 2211–2217.

[9] Z.-J. Wang, Z.-H. Zhan, W.-J. Yu, Y. Lin, J. Zhang, T.-L. Gu, and J. Zhang, "Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2715–2729, Jun. 2020.

[10] R. Cheng and Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization," *Inf. Sci.*, vol. 291, pp. 43–60, Jan. 2015.

[11] W. K. Tang, M. J. Borgnia, A. L. Hsu, L. Esser, T. Fox, D. V. Natalia, and D. Xia, "Structures of AAA protein translocase Bcs1 suggest translocation mechanism of a folded protein," *Nature Struct. Mol. Biol.*, vol. 27, no. 2, p. 202–209, Feb. 2020.

[12] S. Hirota, N. Funasaki, Y. Fujimoto, H. Okumura, N. Katagiri, Y. Watanabe, J. Choi, M. Terazima, S. Arie, K. Tanaka, M. Shionoya, T. Okajima, and T. Takabe, "Structural change of plastocyanin. Molecular interaction and protein folding," *J. Inorganic Biochem.*, vol. 96, no. 1, p. 32, Jul. 2003.

[13] N. B. Chandar, I. Efremenko, I. Silman, J. M. L. Martin, and J. L. Sussman, "Molecular dynamics simulations of the interaction of mouse and torpedo acetylcholinesterase with covalent inhibitors explain their differential reactivity: Implications for drug design," *Chemico-Biol. Interact.*, vol. 310, Sep. 2019, Art. no. 108715.

[14] A. Hajiebrahimi, Y. Ghasemi, and A. Sakhteman, "FLIP: An assisting software in structure based drug design using fingerprint of protein-ligand interaction profiles," *J. Mol. Graph. Model.*, vol. 78, pp. 234–244, Nov. 2017.

[15] C. H. Y. Lau, G. F. King, and M. Mobli, "Molecular basis of the interaction between gating modifier spider toxins and the voltage sensor of voltage-gated ion channels," *Sci. Rep.*, vol. 6, no. 1, Sep. 2016, Art. no. 34333.

[16] H. Tao, J. Chen, Y. Xiao, Z. Liu, and S. Liang, "259. Molecular basis for the interaction of tarantula toxin jingzhaotoxin-III (β-TRTX-Cj1α) with the voltage sensor of Kv2.1 channel," *Toxicon*, vol. 60, no. 2, p. 229, Aug. 2012.

[17] K. Chen, T. Fu, W. Sun, Q. Huang, P. Zhang, Z. Zhao, X. Zhang, and W. Tan, "DNA-supramolecule conjugates in theranostics," *Theranostics*, vol. 9, no. 11, pp. 3262–3279, 2019.

[18] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 459–472, Aug. 2006.

[19] H. Zandevakili, E. Rashedi, and A. Mahani, "Gravitational search algorithm with both attractive and repulsive forces," *Soft Comput.*, vol. 23, no. 3, pp. 783–825, Feb. 2019.

[20] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

[21] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[22] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.

[23] Z. Chen, Y. Bo, P. Wu, and W. Zhou, "A new particle filter based on organizational adjustment particle swarm optimization," *Appl. Math. Inf. Sci.*, vol. 7, no. 1, pp. 179–186, Jan. 2013.

[24] Y. Jin, J. Wang, and L. Wu, "An improved particle swarm optimization algorithm," in *Proc. ICE-CC.*, Sep. 2011, pp. 1864–1867.

[25] W.-B. Du, Y. Gao, C. Liu, Z. Zheng, and Z. Wang, "Adequate is better: Particle swarm optimization with limited-information," *Appl. Math. Comput.*, vol. 268, pp. 832–838, Oct. 2015.

[26] B. Niu, H. Huang, L. Tan, and Q. Duan, "Symbiosis-based alternative learning multi-swarm particle swarm optimization," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 4–14, Jan. 2017.

[27] Q. Yang, W.-N. Chen, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 578–594, Aug. 2018.

[28] X. Liang, W. Li, Y. Zhang, and M. Zhou, "An adaptive particle swarm optimization method based on clustering," *Soft Comput.*, vol. 19, no. 2, pp. 431–448, Feb. 2015.

[29] H. Wang, S. Yang, W. H. Ip, and D. Wang, "A memetic particle swarm optimisation algorithm for dynamic multi-modal optimisation problems," *Int. J. Syst. Sci.*, vol. 43, no. 7, pp. 1268–1283, Jul. 2012.

[30] Y. Shi, "Developmental swarm intelligence: Developmental learning perspective of swarm intelligence algorithms," *Int. J. Swarm Intell. Res.*, vol. 5, no. 1, pp. 36–54, Jan. 2014.

[31] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Biomimicry of parasitic behavior in a coevolutionary particle swarm optimization algorithm for global optimization," *Appl. Soft Comput.*, vol. 32, pp. 224–240, Jul. 2015.

[32] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[33] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Inf. Sci.*, vol. 223, pp. 119–135, Feb. 2013.

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

[35] C. W. Cleghorn and A. P. Engelbrecht, "A generalized theoretical deterministic particle swarm model," *Swarm Intell.*, vol. 8, no. 1, pp. 35–59, Mar. 2014.

[36] Ş. Gülcü and H. Kodaz, "A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 33–45, Oct. 2015.

[37] W. H. Lim and N. A. Mat Isa, "Particle swarm optimization with adaptive time-varying topology connectivity," *Appl. Soft Comput.*, vol. 24, pp. 623–642, Nov. 2014.

[38] M. Ghasemi, J. Aghaei, and M. Hadipour, "New self-organising hierarchical PSO with jumping time-varying acceleration coefficients," *Electron. Lett.*, vol. 53, no. 20, pp. 1360–1361, Sep. 2017.

[39] H.-B. Ouyang, L.-Q. Gao, S. Li, and X.-Y. Kong, "Improved global-best-guided particle swarm optimization with learning operation for global optimization problems," *Appl. Soft Comput.*, vol. 52, pp. 987–1008, Mar. 2017.

[40] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2238–2251, Oct. 2016.

[41] W. H. Lim and N. A. M. Isa, "Particle swarm optimization with dual-level task allocation," *Eng. Appl. Artif. Intell.*, vol. 38, pp. 88–110, Feb. 2015.

[42] T. Krink, J. S. Vesterstrom, and J. Riget, "Particle swarm optimisation with spatial particle extension," in *Proc. Congr. Evol. Comput. CEC*, May 2002, pp. 1474–1479.

[43] H. Dallard, S. S. Lam, and S. Kulturel-Konak, "Solving the orienteering problem using attractive and repulsive particle swarm optimization," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2007, p. 12.

[44] D. M. Munoz, C. H. Llanos, L. D. S. Coelho, and M. Ayala-Rincon, "Hardware particle swarm optimization based on the attractive-repulsive scheme for embedded applications," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, Dec. 2010, pp. 55–60.

[45] S. Mo, J. Zeng, and W. Xu, "Attractive and repulsive fully informed particle swarm optimization based on the modified fitness model," *Soft Comput.*, vol. 20, no. 3, pp. 863–884, Mar. 2016.

[46] H. Yonenoh, R. Suzuki, and T. Arita, "Effects of individual and social learning on the evolution of co-creative linguistic communication," *Artif. Life Robot.*, vol. 24, no. 4, pp. 534–541, Dec. 2019.

[47] C. M. Heyes, E. D. Ray, C. J. Mitchell, and T. Nokes, "Stimulus enhancement: Controls for social facilitation and local enhancement," *Learn. Motivat.*, vol. 31, no. 2, pp. 83–98, May 2000.

[48] G.-Y. Lin, "Scripts and mastery goal orientation in face-to-face versus computer-mediated collaborative learning: Influence on performance, affective and motivational outcomes, and social ability," *Comput. Edu.*, vol. 143, Jan. 2020, Art. no. 103691.

[49] G. Xu, X. Zhao, T. Wu, R. Li, and X. Li, "An elitist learning particle swarm optimization with scaling mutation and ring topology," *IEEE Access*, vol. 6, pp. 78453–78470, 2018.

[50] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., Nov. 2016, pp. 1–18.

[51] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang, "Benchmark functions for the CEC 2008 special session and competition on large scale global optimization," in *Nature Inspired Computation and Applications Laboratory*. Hefei, China: USTC, 2008, pp. 1–18.

[52] G.-S. Hao, M.-H. Lim, Y.-S. Ong, H. Huang, and G.-G. Wang, "Domination landscape in evolutionary algorithms and its applications," *Soft Comput.*, vol. 23, no. 11, pp. 3563–3570, Jun. 2019.

YUN HOU received the B.Sc. degree in control theory and control engineering from Soochow University. He is currently pursuing the master's degree with the School of Information and Control Engineering, China University of Mining and Technology.

GUO-SHENG HAO was born in Wanquan, Hebei, China, in 1972. He received the B.S. degree in exploration engineering, and the M.S. and Ph.D. degrees in control theory and application from the China University of Mining and Technology, Xuzhou, China, in 1997, 2005, and 2009, respectively. He is currently a Professor at the School of Computer Science and Technology, Jiangsu Normal University, China. His current research interests include evolutionary computation, big data, and software design.

YONG ZHANG (Member, IEEE) received the B.Sc. and Ph.D. degrees in control theory and control engineering from the China University of Mining and Technology, in 2006 and 2009, respectively. He is currently a Professor at the School of Information and Control Engineering, China University of Mining and Technology. His research interests include intelligence optimization and data mining.

FENG GU received the B.S. degree in mechanical engineering from the China University of Mining and Technology, the M.S. degree in information systems from the Beijing Institute of Machinery, and the M.S. and Ph.D. degrees in computer science from Georgia State University. He is currently an Associate Professor of computer science at the College of Staten Island, The City University of New York, and a Doctoral Faculty Member at the Graduate Center of the City University of New York. His research interests include modeling and simulation, complex systems, and high performance computing. His research was supported by the National Science Foundation, the National Institute of Justice, CUNY IRG, and PSC-CUNY.

XIA WANG was born in Shanxi, China, in 1978. She is currently a Lecturer at the School of Computer Science and Technology, Jiangsu Normal University, China. Her main research interests include machine learning, support machine vector, and complex calculations.

TING-TING ZHANG was born in Xuzhou, China, in 1997. She received the B.S. degree in medical information engineering from Xuzhou Medical University, in 2019. She is currently pursuing the master's degree in software engineering with the School of Computer Science and Technology, Jiangsu Normal University.