Journal of Intelligent & Robotic Systems (2022) 105: 30
https://doi.org/10.1007/s10846-022-01627-y

REGULAR PAPER

Self-Regulating and Self-Perception Particle Swarm Optimization with Mutation Mechanism

Yanjie Chen[1,4] · Jinglin Liang[1] · Yangning Wu[1] · Bingwei He[1] · Lixiong Lin[2] · Yaonan Wang[3,4]

Received: 1 October 2021 / Accepted: 31 March 2022 / Published online: 19 May 2022

Abstract

Particle swarm optimization (PSO) is widely used to solve various optimization problems, such as robotics visual perception and intelligent control under uncertainties, due to its simple rules and easy implementation. However, the PSO has premature convergence in the optimization process, which will lead to inaccurate problems such as uncertainties of the control system. To improve PSOs performance, a self-regulating particle swarm optimization with mutation mechanism (SRM-PSO) is proposed in this paper. SRM-PSO combines the mutation mechanism, self-regulation and self-perception strategy. The mutation mechanism is introduced to generate trial particle moving in different directions to maintain population diversity. Self-regulation and self-perception enable particles to be updated in different ways for fast exploration and intelligent exploitation. To validate the effectiveness of the SRM-PSO, experiments are conducted in the CEC2017 test suite. The test results indicate that SRM-PSO outperforms two related variants, and five representative PSO variants. Further, SRM-PSO is applied to several real-world optimization problems, which demonstrates its potential and competitiveness.

Keywords Particle swarm optimization · Self-regulating inertia weight · Self-perception on search direction · Mutation mechanism · Premature convergence

✉ Yanjie Chen
chenyanjiehnu@gmail.com

Jinglin Liang
jingl.liang98@gmail.com

Yangning Wu
WuYangNing.wyn@hotmail.com

Bingwei He
mebwhe@fzu.edu.cn

Lixiong Lin
linlixiong@126.com

Yaonan Wang
yaonan@hnu.edu.cn

1 School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, 350108, Fujian Province, China

2 School of Ocean Information Engineering, Jimei University, Xiamen, 361021, Fujian Province, China

3 College of Electrical and Information Engineering, Hunan University, Changsha, 410082, Hunan Province, China

4 National Engineering Research Center of Robot Visual Perception and Control Technology, Changsha, 410082, Hunan Province, China

1 Introduction

Optimization problems have been commonly existed in real life, such as power systems [1], image processing [2], control systems [3], parameters tuning [4], text mining [5], grouping problems [6] and so on. Since many optimization problems become increasingly complex, it is difficult to solve them by conventional methods. Therefore, a lot of intelligent algorithms have emerged to solve complex optimization problems in recent decades. For example, genetic algorithm (GA) [7], differential evolution (DE) algorithm [8], artificial bee colony (ABC) algorithm [9], grey wolf optimizer (GWO) [10], across neighborhood search (ANS) algorithm [11], particle swarm optimization (PSO) [12] et al.

Among the above intelligent algorithms, particle swarm optimization (PSO) [12] is a population-based intelligent optimization algorithm inspired by the collective behavior of bird flocking to finding food. Owing to its easy implementation and fast convergence rate, PSO has been widely applied. However, standard PSO suffers from the loss of population diversity and premature convergence. Therefore, methods to further improve the performance of PSO have been proposed continuously. The important variants of PSO can be broadly classified into the following four types:

(1) Parameter setting: The suitable selection and adjustment of parameters, such as inertia weight, acceleration coefficients, significantly influence the optimization results. There are many modes to adjust the inertia weight, such as fixed [13], linearly decreasing [14], linearly increasing [15], random [16], exponential [17]. Meanwhile, Han et al. [18] proposed that linear decreasing inertia weight was the simplest and most effective method. In addition, the adjustment of acceleration coefficients could also improve the performance of PSO [19]. Clerc and Kennedy [20] introduced constriction coefficient to control the dynamical characteristics of the population and achieved effective improvement in some problems.
(2) Neighborhood topology: To enhance the exploration capability of PSO, fully connected and wheel topology [21] have been proposed successively. Mendes and Kennedy [22] proposed fully informed particle swarm (FIPS) and studied the topology of squares, rings, clusters and pyramids. The results showed that FIPS with different topologies could perform well on different problems. Liu et al. [23] systematically investigated the effects of various topologies, and then provided guidance for the topology selection of particle swarm optimization algorithm.
(3)Learning strategy: It is an important variant to introduce different learning strategies to make particle swarm optimization converge to the global optimum effectively. Liang et al. [24] introduced a novel comprehensive learning strategy that updated the velocity of a given particle with the personal best information of all other particles. HCLPSO algorithm [25] proposed heterogeneous subpopulations based on CLPSO. Lin et al. [26] combined global learning component and a ring topology into the gene learning PSO to enhance performance.
(4) Hybridized version: The hybrid version is intended to mitigate the weaknesses of the intelligent algorithm by combining the useful features of different intelligent algorithms. Zhang et al. [27] integrated prey-predator relationship (catch, escape, breeding) into PSO to obtain better performance. To calculate expensive numerical problems, a hybrid scheme of firefly and particle swarm was proposed in [28]. Hybridizing PSO with evolutionary algorithm provided an effective approach to mitigate premature convergence [29].

Among the four variants above, PSO based on learning strategies and hybridized versions are more attractive to researchers because of their better convergence properties. Due to the effectiveness of learning strategies, self-regulation and self-perception strategy was proposed according to the best human learning principle in [30]. Self-regulation refers to adjusting strategies through ones current status. Moreover, learning strategy with self-regulation provides better generalization performance over other algorithms [31]. According to self-regulation, the population is divided into two parts: the current best particle and the rest. Different parts use different strategies to update inertia weight. The self-perception of the global best position is used by the rest of the particles for a better search. In addition, these particles extract experience from other particles to a certain extent to guide a better search direction. However, the self-perception will gradually weaken in the later stage due to the convergence of particles, which will lead to the loss of diversity and weaken the searching ability of the algorithm. For this problem, mutation mechanism [32] is considered to overcome the diversity loss. The mutation mechanism had been demonstrated to maintain population diversity [33]. Therefore, combining mutation mechanism with self-regulation and self-perception can further improve the performance of PSO for better optimization.

Motivated by the above analysis, in this paper, a new particle swarm optimization algorithm named SRM-PSO is proposed to preserve the diversity of population and enhance the search ability. SRM-PSO combines the mutation mechanism, self-regulation and self-perception strategies together. The mutation mechanism works by producing trial particles in the iteration to replace certain particles in the population. The trial particle can motivate particles out of local optima to search the global optimum effectively. The purpose of self-regulation and self-perception strategy is to improve the optimization ability and convergence accuracy. The CEC2017 test suite [34] and real-world optimization problem are employed to test the performance of SRM-PSO. Meanwhile, the representative algorithms are selected for comparison. The main contributions of this paper are as follows.

(1) *Effective combination of two improved strategies*. The proposed new improved algorithm, namely, SRM-PSO, contains two parts: mutation mechanism, self-regulation and self-perception strategy. The mutation mechanism can preserve the diversity of the population to overcome the diversity loss of self-perception strategy. The self-regulation and self-perception strategy improve the exploration and exploitation capabilities, and compensate for the diminishing rate of convergence due to the mutation. The two improved strategies can be effectively combined with each other and further promote the performance of PSO.

(2) *Strong competitiveness in various comparative algorithms.* The experimental comparisons with two related variants, five representative PSO variants, including SRPSO, MPSO, CPSO, SLPSO, HFPSO, PSOG and CHCLPSO, indicate that the SRM-PSO performs better than the representative popular PSO variants in various functions.

(3) *Applicability to real-world optimization problems.* The proposed SRM-PSO are applied to real-world optimization problem, which demonstrates the feasibility and effectiveness of SRM-PSO in practical problems.

The rest of this paper is organized as follows. Section 2 introduces briefly the relevant work, including the basic principle of particle swarm optimization and some variants, as well as the relevant knowledge of self-regulating PSO. Section 3 describes detailedly the various components of the proposed algorithm. Section 4 presents the experimental setup and performance results, compares the results of each algorithm by the CEC2017 test suite, and provides a detailed analysis of SRM-PSO. The convergence rate analysis of SRM-PSO is also included in this section. This section also applies SRM-PSO to real-world optimization problem [35] and analyzes the results. Section 5 summarizes the conclusions of this study.

2 Related Works

This section first outlines the fundamentals of the PSO algorithm and later introduces related PSO variants and several mutation mechanisms applied to PSO algorithm. Further, the essential theory behind the Self-Regulating PSO is described.

2.1 Standard PSO and Some Variants

PSO is a population-based stochastic optimization algorithm that motivated by the behavior of bird flocking [12]. The principle of PSO algorithm is to generate random particles at the beginning. These particles influence the search direction and speed by sharing information about their own position and cooperating with other particles. Each population has N particles, and every particles are assumed to be massless and volumeless in the D-dimensional search space. Each particle i has the random position $X_i^d = (X_i^1, X_i^2, \ldots, X_i^D)$ and random velocity $V_i^d = (V_i^1, V_i^2, \ldots, V_i^D)$ at the initial stage. The position of each particle is a potential solution to the unanswered question. Each particle updates its velocity and position respectively according to the following two equations:

$$V_i^d(t+1) = \omega V_i^d(t) + c_1 r_1 \left(P_i^d(t) - X_i^d(t)\right) + c_2 r_2 \left(G^d(t) - X_i^d(t)\right) \tag{1}$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1) \tag{2}$$

where t and $t+1$ represents the current and next iteration, respectively. P_i^d denotes the P_{best} of particle i, and G^d denotes the G_{best} of the entire swarm. ω is the inertia weight. c_1 and c_2 are acceleration coefficients. r_1 and r_2 are the uniformly distributed random numbers within the range [0, 1]. The neighborhood structure of Eq. (1) is the global topology, in which each particle is connected to every other particle in the swarm, hence the best solution currently found by any particle can be accessed at any time.

In PSO's velocity update Eq. (1), the inertia weight ω balances the global exploration and local exploitation capabilities. The larger ω tends to exploratory search, so that particles can search faster and more widely in the whole solution space. Conversely, the smaller ω tends to be an exploitive search, enabling the swarm to search precisely in the space of potential solutions. Hence, appropriate inertia weight is needed to guide the search process. In [14], the inertia weight ω was defined as a linear decreasing function of the iteration. In addition, acceleration coefficients c_1 and c_2 determine the influence of P_{best} and G_{best}, and reflecting the information exchange and experience sharing among all particles. When the cognitive constant c_1 is higher than the social constant c_2, it will cause the particles to wander excessively. Therefore, Clerc and Kennedy [20] improved PSO by introducing the contraction coefficient to control dynamic characteristics of the particle swarm, including its exploration and exploitation tendencies as follows:

$$V_i^d(t+1) = \chi \left(V_i^d(t) + c_1 r_1 \left(P_i^d(t) - X_i^d(t)\right) + c_2 r_2 \left(G^d(t) - X_i^d(t)\right)\right) \tag{3}$$

$$\chi = \frac{2}{|2 - C - \sqrt{C^2 - 4C}|} \tag{4}$$

where χ denotes contraction coefficient, and $C = c_1 + c_2$.

Particle swarm updates particle velocity and position by learning from all particles' P_{best} and G_{best}. If P_{best} and G_{best} guide the particles to different directions, the particles will produce fluctuations in the search process. On the other hand, if P_{best} and G_{best} guide the particles in the same directions, the particles will follow the same evolutionary direction. In this case, particles will follow the G_{best} into the local optima if G_{best} is away from the global optimum, causing "premature". "Premature" will make the diversity and velocity of swarm to decrease quickly and stop searching early. To avoid falling into local optima due to premature, the introduction of mutation mechanism is the most classical method.

Mutation mechanism is derived from evolutionary computation technique, and one of its principles is to bring the mutation operator into the population by randomly changing the position of particles in the population [33].

The mutation operator can prevent the loss of population diversity and enable the particle to escape from the local optima, so as to make a more sufficient exploration in the search space. Higashi and Iba [36] proposed a mutation operator to extract a random number from Gaussian distribution. By using this extracted random number, particles are selected to mutate during evolution. The positions of particles are mutated according to the following equation:

$$\text{mutate}(x_i^d) = x_i^d\,(1 + \text{gaussian}(\sigma)) \tag{5}$$

where gaussian(σ) is the random number extracted from a Gaussian distribution with a standard deviation of σ. The range of σ value was restricted by the author to ensure that all dimensions have the same shape of the distribution. Sarangi et al. [37] introduced a new Cauchy mutation into the modified particle swarm optimization, enabling particles to increase the probability of escape from the local optima. Tang et al. [38] used an adaptive mutation operator to adjust the adaptive mutation size according to the current search space size. This mutation mechanism is to mutate the current G_{best}, and its mutation equation is as follows:

$$G_j(t+1) = G_j(t) + [b_j(t) - a_j(t)]^{*}\text{rand}\,() \tag{6}$$

$$a_j(t) = \text{Min}(x_{ij}(t)),\ \ b_j(t) = \text{Max}(x_{ij}(t)) \tag{7}$$

where G_j is the j^{th} vector of the global best particle, $a_j(t)$ and $b_j(t)$ are the minimum and maximum values of the j^{th} dimensional search space, respectively. rand() is a random number within the range [0, 1]. t and $t+1$ represents the current and next iteration, respectively.

Nevertheless, the variants mentioned above only mitigated but not solved the disadvantage of the particle falling into local optima. Meanwhile, these variants do not exhibit well application to the complex and high dimensional problems. In this point of view, this paper adopts the mutation mechanism proposed by Khan et al. [32]. This mutation mechanism randomly selects a particle from good subpopulation to produce a trial particle through mutation mechanism, so as to jump out of local optima. It makes the particle swarm preferably maintain the diversity of the population in the iteration.

2.2 Self Regulating PSO

As previously mentioned, the addition of mutation mechanism can avoid the loss of population diversity and premature convergence. However, the principle of mutation is to randomise the particles, which means that convergence may slow down. Therefore, the appropriate improvement of the particle swarm optimization is required to combine the mutation operator. Among the four PSO variants mentioned in the introduction, learning strategies have strong effectiveness in improving PSO [24]. The literature of machine learning shows that the learning algorithm employing self-regulation possesses better generalization performance than other algorithms. Therefore, according to the best human learning principle, Tanweer et al. [30] proposed a Self Regulating Particle Swarm Optimization (SRPSO). Self-regulation can help decision maker to make an efficient adjustment to the expected target according to their current status, and provide favorable conditions for better exploration and exploitation. The PSO schematic based on human learning principle is shown in Fig. 1. By monitoring the results obtained by PSO, the decision maker provides self-regulation and self-perception strategy for particles updating. Consequently, self-regulation and self-perception strategy can provide better solutions in most of the optimization problems, thereby can be applied to practical problems. For example, based on SRPSO, Tanweer et al. [39] added the directional update strategy to make the elite particles get directional update, and added the rotational invariant strategy to explore the rotation variance property of the search space. In [40], a dynamic mentoring scheme was combined with self-regulation scheme, yielding an effective optimization algorithm for real-world applications. Dash et al. [41] proposed a mutation-based self-regulating and self-perception PSO for realistic object tracking problem.

Self regulating particle swarm optimization consists of two strategies. The first strategy is self-regulating inertia weight. In each iteration, the current best particle accelerates its own velocity by an increased inertia weight, and this acceleration process does not interact with other particles. The current best particle will resume normal search strategy when it loses the global best position. The second strategy is the self-perception of search direction. The current best particle only follows its own direction as the best direction, and is unaffected by its own and others experience. The rest of the particles use their perception of global best to find the appropriate direction to follow. While these particles learn from their own experience, they also learn from the experience of others to a certain extent, so as to obtain a better direction.

Both strategies were adopted, hence the general velocity update equation in SRPSO is:

$$\begin{aligned} V_i^d\,(t+1) &= \omega V_i^d\,(t) + c_1 r_1 p_{se}\left(P_i^d\,(t) - X_i^d\,(t)\right) \\ &\quad + c_2 r_2 p_{so}\left(G^d\,(t) - X_i^d\,(t)\right) \end{aligned} \tag{8}$$

where ω is the inertia weight, and the current best particle and others employ different inertia weigh strategies. p_{se} and p_{so} respectively denote the perception for self-cognition and social-cognition, and the they are defined as:

$$p_{se} = \begin{cases} 0, & \text{best particle} \\ 1, & \text{other particles} \end{cases} \tag{9}$$

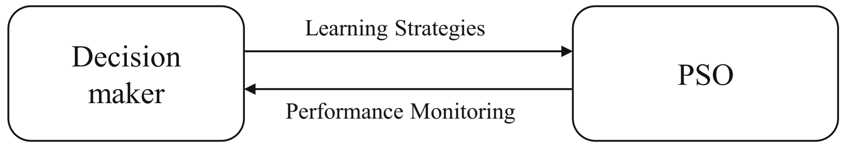


Fig. 1 The Human Learning based PSO Schematic

and

$$p_{so} = \begin{cases} 0, & \text{best particle} \\ \gamma, & \text{other particles} \end{cases} \tag{10}$$

where γ is binary depending on the threshold that defines the confidence.

3 Methodologies

As a swarm intelligence algorithm, PSO suffers from some weaknesses, such as poor convergence, easy to be trapped in the local optima, and the reduction of population diversity in the search process. To improve the performance of PSO, we propose a self-regulating particle swarm optimization with mutation mechanism in this paper, called SRM-PSO.

3.1 SRM-PSO

SRM-PSO is improved by combining two parts: (1) Mutation mechanism; (2) Self-regulating inertia weight and self-perception strategy.

The mutation mechanism is to mutate a randomly selected particle to produce a trial particle for substitution in each iteration. This mechanism can make the population avoid falling into the local optima and jump out to explore the global optimum in the search process. Meanwhile, the mutation mechanism can maintain the diversity of the population during iteration.

Self-regulating inertial weight and self-perception strategy are inspired by the best human learning principle. The current best particle and other particles have different inertia weight to select. The current best particle adopts the linear increasing inertia weight, which makes it reach the region with possible solution faster. The linear decreasing inertia weight is employed in other particles to ensure fast exploration in the early stage and accurate exploitation in the later stage. The self-perception strategy allows particles except current best particle to partially draw on social-cognition, rather than fully believing in it. Hence, this allows the particles to achieve better exploration and exploitation capabilities, and to some extent avoid falling into local optima.

The details of these two parts will be explained in the following Sections 3.2 and 3.3, respectively. The flowchart of SRM-PSO proposed in this paper is shown in Fig. 2. The description in the algorithm iteration is shown in Fig. 3. The pseudo code of SRM-PSO is presented in Algorithm 1.

Algorithm 1 Pseudo code for SRM-PSO.

Initialization
Calculate the fitness value for each particle
Find the P_{best} of each particle
while $t < t_{max}$ **do**
  Calculate the threshold value U of the fitness function of the population using Eq. (11)
  Divide the population into two subpopulations according to the threshold U
  Randomly select P^* from the good subpopulation
  Randomly generate Y_j from the current search space using Eq. (12)
  Generate the trial particle X^* using Eq. (13)
  if $\text{fit}(X^*) < \text{fit}(worst\ particle)$ **then**
    The trial particle replaces the worst particle
  end if
  Find the G_{best} of whole swarm
  for *the best particle* **do**
    Calculate the inertia weight using Eq. (14)
    Update the velocity using Eq. (15)
  end for
  for *the other particles* **do**
    for $j = 1 : dimension$ **do**
      Generate the uniform random number a
      if $(a > \lambda)$ **then**
        Select the directions from global best
      else
        Reject the directions
      end if
    end for
    Update the velocity using Eq. (15)
  end for
  Update the position of each particle using Eq. (2)
end while

3.2 Mutation Mechanism

The first part of the algorithm improvement is the mutation mechanism. It can preserve the diversity of the population and avoid premature convergence. The mutation mechanism is operated by randomly selecting a particle P^* from a good subpopulation to generate a trial particle, then the survival of the fittest will be carried out between the trial particle and the worst particle in the current population. This mechanism can stimulate the particles to move in different directions during the search process to ensure that the particles jump out of the local optimal positions, which is more conducive to the exploration of the global optimal position.

The mutation mechanism needs to be guided by information applied to individual particles in the current population,

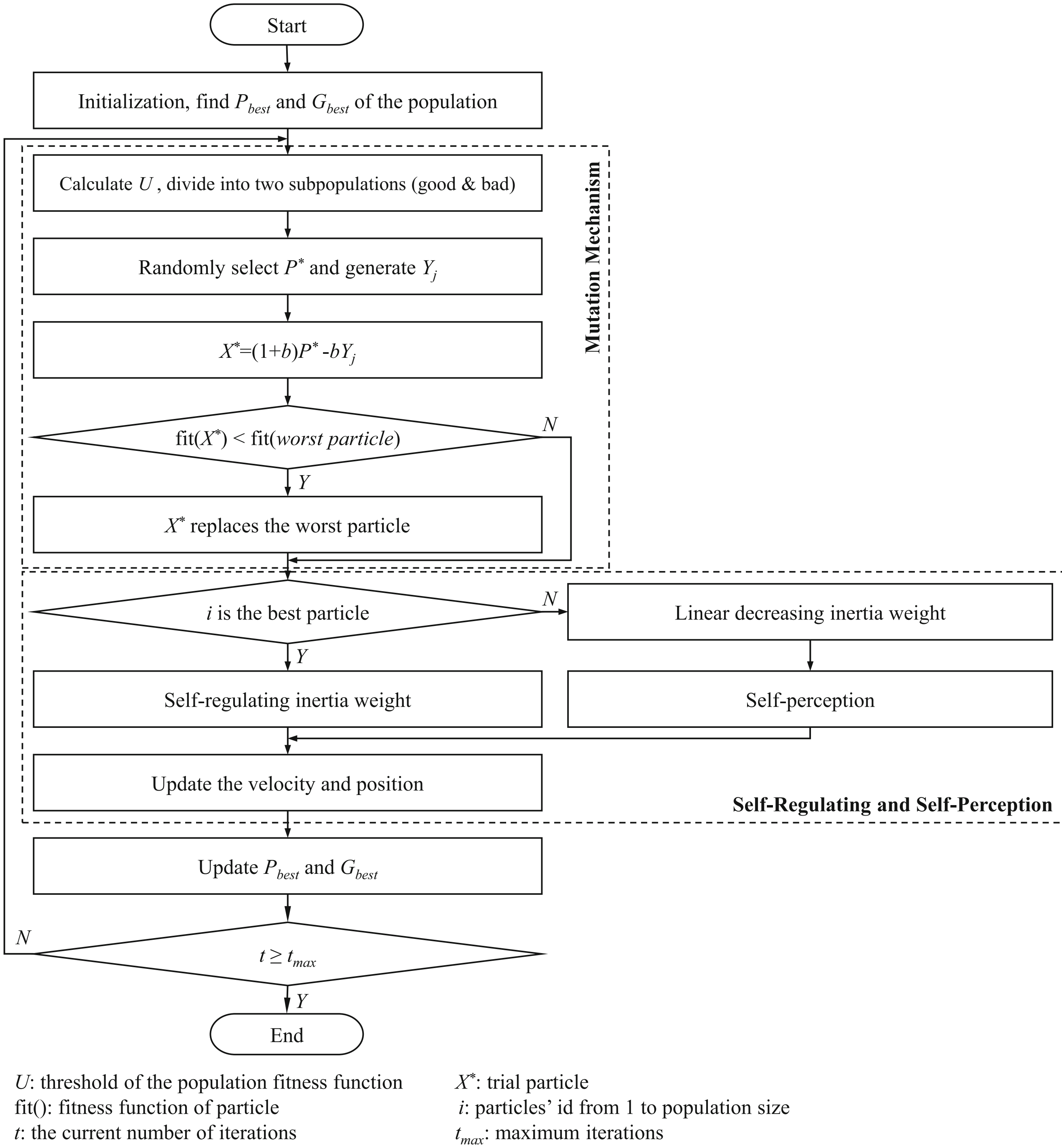


Fig. 2 Flowchart of the SRM-PSO

hence the threshold U of a population fitness function value is defined as follows:

$$U = K \times \frac{\sum_{i=1}^{N} \text{fit}(i)}{N} \quad (11)$$

where fit(i) represents the fitness of particle i. N is the number of particles in the population, and K is a constant between 0 and 1. According to this threshold U, the current population are divided into two sub-populations. In particular, the particles whose fitness value are smaller than this threshold constitute the good subpopulation, and the remaining particles constitute the bad subpopulation. A particle P^* is randomly selected from the good subpopulation for the design of the trial particle.

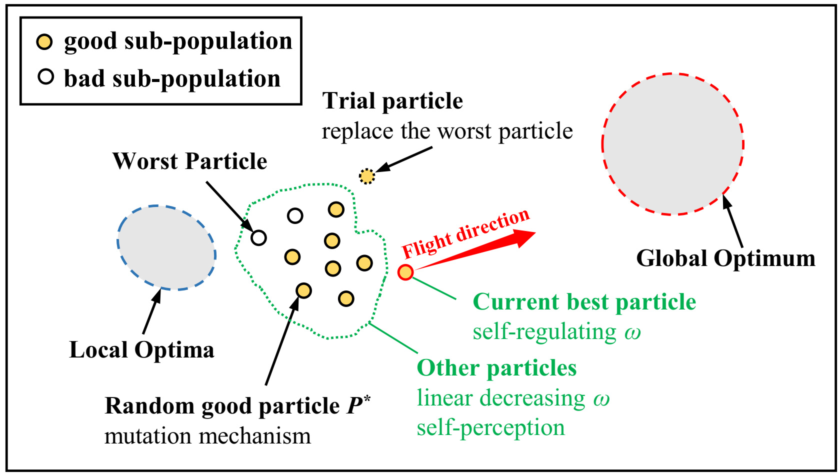


Fig. 3 Motion of particles in iteration of SRM-PSO

First, randomly generate a particle Y_j from the search space:

$$Y_j = \text{rand}\left(l_j, u_j\right), \quad j = 1,\ 2,\ \ldots,\ D \tag{12}$$

where l_j and u_j are the lower and upper boundary of the j^{th} dimensional search space, respectively.

Then, the trial particle $X^* = [X_1^*, X_2^*, \ldots, X_D^*]$ is generated by Eq. (13).

$$X^* = (1 + b)\, P^* - bY_j \tag{13}$$

where b is a constant in the range (0, 1).

When the trial particle is generated, its fitness value is compared with the particle with the worst fitness value in the current population. If the trial particle is better than the worst particle, then the trial particle will replace the worst particle. Otherwise, the worst particle will remain unchanged and continue to the next iteration. The mutation mechanism produces alternative trial particles to preserve the diversity of the population. Furthermore, trial particles can guide particles to search for potential solutions more effectively.

3.3 Self-Regulation and Self-Perception

The second part of the proposed SRM-PSO is the self-regulating inertia weight and self-perception strategy. As mentioned in Introduction, learning algorithm with self-regulation and self-perception has better generalization performance than other algorithms. Therefore, different inertial weight strategy is applied to the current best particle and other particles respectively. The self-regulating inertia weight strategy is defined as follow:

$$\omega = \begin{cases} \omega_i + \frac{t \times (\omega_i - \omega_f)}{t_{max}}, & \text{best particle} \\ \omega_i - \frac{t \times (\omega_i - \omega_f)}{t_{max}}, & \text{other particles} \end{cases} \tag{14}$$

where ω_i and ω_f denote the initial and final inertia weight, respectively. t and t_{max} represent the current iteration number and the maximum iteration number. For the current best particle, the corresponding linear increasing inertia weight is employed for updating. In this case, the current best particle follows the principle that it believes its own direction and speeds up the search for the global optimum in that direction. For other particles, they are updated according to the linear decreasing inertia weight. From the experiments of Harrison et al. [42], the value of inertia weight would be more appropriate within [0.4, 0.9]. Hence, ω_i and ω_f are set to 0.9 and 0.4, respectively, to strengthen early exploration and late exploitation.

Moreover, the velocity update strategy of the current best particle is different from that of other particles. Inspired by human collaborative learning strategy, we adopt a new learning strategy for other particles, in which particles would use their perception of global search directions to obtain information for social development. This new strategy is called self-perception strategy. The velocity update formula of general self-regulating particle swarm optimization algorithm is given by Eq. (15):

$$\begin{aligned} V_i^d(t+1) = {} & \omega V_i^d(t) + c_1 r_1 p_{se}\left(P_i^d(t) - X_i^d(t)\right) \\ & + c_2 r_2 p_{so}\left(G^d(t) - X_i^d(t)\right) \end{aligned} \tag{15}$$

where p_{se} and p_{so} are self-cognition and social-cognition.

Since self-perception can have a significant impact on the direction of particle search, for the current best particle, it completely believes in itself, speeds up the search, and cancels self-cognition and social-cognition. Therefore, the self-cognition and social-cognition of current best particle are both set as 0 in Eq. (15), and its velocity update equation is obtained as follow:

$$V_i^d(t+1) = \omega V_i^d(t), \quad \text{(best particle)} \tag{16}$$

For other particles, their self-cognition and social-cognition are both set as 1. Social-cognition can lead to better convergence, but if a population uses full social-cognition, it may be trapped in local optima in some particular iteration and lead to premature convergence. To improve the insufficiency that may be caused by complete social-cognition, the social-cognition p_{so} is defined as follow:

$$p_{so} = \begin{cases} 1, & \text{if } a > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where a is the uniformly distributed random numbers within the range [0, 1], and λ is a defined threshold value. The value of the threshold λ is an important problem. When the λ is too small, it approaches full social-cognition; And when the λ is too large, social-cognition becomes very low. At this point, particles will almost only fly in their own direction, but will not be attracted by the global optimum, which may lead to failure of convergence or poor convergence effect. In Tanweer's study [30], experiments showed that the λ set as 0.5 is the most suitable value, allowing other particles to speed up updates through self-cognition and partial social-cognition. Meanwhile, the addition of self-perception strategy reduces the probability of particles being attracted to the local optima.

4 Experiments Results and Analysis

In this section, the setup of experimental conditions is introduced first. Secondly, the parameter configurations of related algorithms are explained. The characteristics of SRM-PSO and experimental results are detailedly compared and analyzed later, and the application of real-world optimization problems is also tested and reported.

4.1 Experimental Setup

To test the performance of the proposed SRM-PSO, the CEC2017 test suite with 30 benchmark functions [34] is employed. Compared to previous releases, the CEC2017 test suite has developed questions with new features, such as new basic questions, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on. Moreover, the exact equations of the test functions are not allowed to be used. The summary of the CEC2017 test suite is presented in Table 1.

As shown in Table 1, the CEC2017 test suite contains four categories, including three unimodal functions ($f_1 - f_3$), seven simple multimodal functions ($f_4 - f_{10}$), ten hybrid functions ($f_{11} - f_{20}$), and ten composition functions ($f_{21} - f_{30}$). All test functions are minimization problems, and the final result is the error between the algorithm result and the global optimal solution. F_i^* is the global optimum of each function within a given bound, and there is no need to perform search outside of the given bounds for these problems. In the CEC2017 test suite, the search range of each function is set as [-100,100], and the maximum number of function evaluations (Max_FES) is $10,000 \times D$. The dimension (D) is set as 30 in this study. Each algorithm is running 51 times, and the mean errors are calculated after the running. The experiments have been conducted on a PC with Intel Core i7-7700 2.80 GHz CPU, 8 GB RAM and Matlab R2018b Compiler.

4.2 Parameter Configuration

To test the performance of SRM-PSO, two categories of algorithms are selected to compare. The first category is to test the superiority of PSO that combine the mutation mechanism with self-regulation and self-perception, including PSO, MPSO and SRPSO. The second category is five relatively popular PSO variants. All the comparison algorithms are described as follows. CPSO [43] introduces the notion of charge, thereby modifying the rule for particle accelerations. SLPSO [44] assigns four different search strategies to different particles based on their fitness. HFPSO [28] hybridizes the particle swarm and firefly algorithm, and is able to exploit the strongpoints of both. PSOG [45] is combined with first and second order gradient directions to optimize the algorithm. CHCLPSO [46] adjusts the parameters of heterogeneous comprehensive learning particle swarm optimizer through chaotic mapping.

All the parameters of the comparison algorithm are optimized and obtained from previous original papers. The parameter settings of related PSO variants are presented in Table 2.

In the mutation mechanism part of SRM-PSO, there is an Eq. (11) for defining the threshold U of the population fitness function value.

$$U = K \times \frac{\sum_{i=1}^{N} \text{fit}(i)}{N} \tag{18}$$

where K is a user defined constant within the range [0,1]. The division of the good sub-population and the bad sub-population depends on the value of threshold U. A large U will make the good sub-population contains too many particles, affecting the merits and demerits of particle P^*. A small U will cause no particle to reach that P^* cannot be generated. To select an appropriate K value, the performance of SRM-PSO under different K values are tested and compared. The f_3, f_{10}, f_{17}, f_{23} functions in CEC2017 test suite are employed. The test results are presented in Table 3, including the mean, median and standard deviation error.

Table 1 Summary of the CEC2017 test functions

	No.	Functions	$F_i^* = F_i(x^*)$
Unimodal Functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Sum of Different Power Function	200
	3	Shifted and Rotated Zakharov Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrocks Function	400
	5	Shifted and Rotated Rastrigins Function	500
	6	Shifted and Rotated Expanded Scaffers F6 Function	600
	7	Shifted and Rotated Lunacek Bi_Rastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigins Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefels Function	1000
Hybrid Functions	11	Hybrid Function 1 (N=3)	1100
	12	Hybrid Function 2 (N=3)	1200
	13	Hybrid Function 3 (N=3)	1300
	14	Hybrid Function 4 (N=4)	1400
	15	Hybrid Function 5 (N=4)	1500
	16	Hybrid Function 6 (N=4)	1600
	17	Hybrid Function 6 (N=5)	1700
	18	Hybrid Function 6 (N=5)	1800
	19	Hybrid Function 6 (N=5)	1900
	20	Hybrid Function 6 (N=6)	2000
Composition Functions	21	Composition Function 1 (N=3)	2100
	22	Composition Function 2 (N=3)	2200
	23	Composition Function 3 (N=4)	2300
	24	Composition Function 4 (N=4)	2400
	25	Composition Function 5 (N=5)	2500
	26	Composition Function 6 (N=5)	2600
	27	Composition Function 7 (N=6)	2700
	28	Composition Function 8 (N=6)	2800
	29	Composition Function 9 (N=3)	2900
	30	Composition Function 10 (N=3)	3000

Search Range: $[-100, 100]^D$

The test algorithms in Table 3 contain standard PSO and SRM-PSO with different K values. The best results obtained for each function are highlighted in bold. Compared with standard PSO, SRM-PSO effectively enhances the convergence accuracy of PSO, whereas K=0.8 has provided best mean errors in four of the selected functions as compared to its other values. Based on these test results, the parameter K value is set at 0.8. To more intuitively reflect the convergence characteristics of SRM-PSO, the iteration diagram of standard PSO and SRM-PSO on function f_{10} is made. The iteration diagram is presented in Fig. 4. The horizontal axis is the number of iterations, and the vertical axis is the logarithm of the average error. It can be observed from the figure that SRM-PSO is closer to the global optimum and converges faster. Meanwhile, similar phenomena can be observed in most other functions.

Table 2 Parameter settings of related PSO variants

Algorithm	Parameter settings
MPSO	$Ps = 50, \omega = \text{rand}/Ps, c_1 = 1 + \text{rand}, c_2 = e - \text{rand}, b = 0.5$
SRPSO	$Ps = 50, \omega_{best} = 1.05 - 1.6, \omega_{others} = 1.05 - 0.5, c_1 = c_2 = 1.49445, \lambda = 0.5$
SRM-PSO	$Ps = 50, \omega_{best} = 0.9 - 1.4, \omega_{others} = 0.9 - 0.4, c_1 = c_2 = 1.49445, \lambda = 0.5, b = 0.5$

Table 3 Parameter analysis on selected 30-D CEC2017 Benchmark Functions

Function	Algorithm	Mean	STD.
f_3	Standard PSO	2.164E+04	7.059E+03
	SRM-PSO (K=0.6)	1.136E+00	3.404E+00
	SRM-PSO (K=0.8)	**5.089E-01**	**9.858E-01**
	SRM-PSO (K=1.0)	5.144E-01	1.108E+00
f_{10}	Standard PSO	6.922E+03	**3.719E+02**
	SRM-PSO (K=0.6)	2.552E+03	6.361E+02
	SRM-PSO (K=0.8)	**2.337E+03**	4.457E+02
	SRM-PSO (K=1.0)	2.408E+03	4.892E+02
f_{17}	Standard PSO	6.122E+02	1.247E+02
	SRM-PSO (K=0.6)	1.150E+02	7.285E+01
	SRM-PSO (K=0.8)	**1.096E+02**	**5.873E+01**
	SRM-PSO (K=1.0)	1.397E+02	8.463E+01
f_{23}	Standard PSO	6.143E+02	3.990E+01
	SRM-PSO (K=0.6)	3.920E+02	1.029E+01
	SRM-PSO (K=0.8)	**3.890E+02**	**9.927E+00**
	SRM-PSO (K=1.0)	3.983E+02	1.806E+01

4.3 Comparison Test

4.3.1 Diversity Analysis

In proposed SRM-PSO, a mutation mechanism is employed to maintain population diversity. Therefore, the "diversity-iterations" performance are shown in Fig. 5 for illustration. The diversity of the population is evaluated by the mean Euclidian distance between each particle and the mean position of the particles [26] according to Eq. (19) in

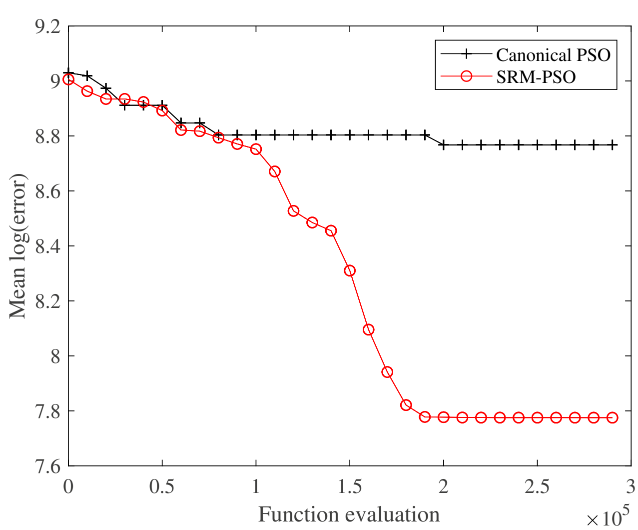


Fig. 4 Convergence of SRM-PSO vs standard PSO

this study. Figure 5(a) indicates that, in unimodal function f_3, the diversity of SRPSO decreases the fastest. With the mutation operator and dynamic parameters, MPSO can preserves more diversity than SRPSO. By combining the mutation mechanism with self-regulating and self-perception strategy, the resultant SRM-PSO can maintain much more diversity than the MPSO and SRPSO in the iteration process. In hybrid multimodal function f_{19}, the diversity of SRPSO is lower than that of MPSO and SRM-PSO in the middle and late stages of the iteration, and continues to decrease. Although MPSO keeps a lower diversity than SRPSO at the beginning, it has maintained a higher diversity than SRPSO afterwards. Because SRM-PSO inherits the mutation mechanism employed in MPSO to generate particles, the occasional diversity increase of SRM-PSO during the optimization process can be seen in Fig. 5(b).

$$diversity = \frac{1}{M}\Sigma_{i=1}^{M}\sqrt{\frac{1}{D}\Sigma_{j=1}^{D}\left(X_{i,j} - \bar{X}_{i,j}\right)^2} \quad (19)$$

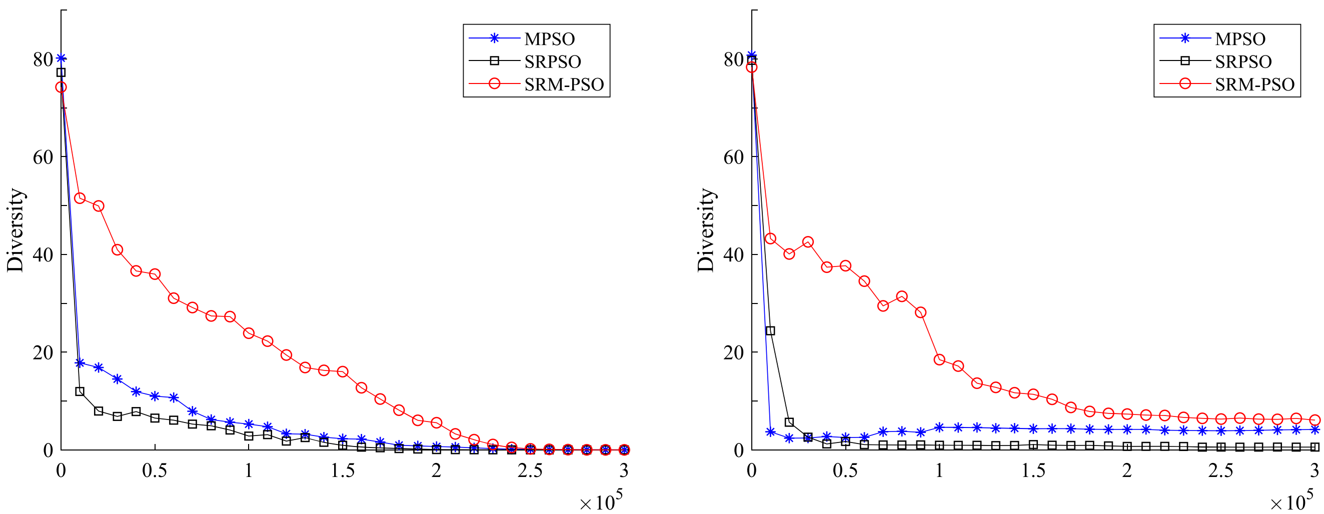
4.3.2 Comparison Test with Related PSO Variants

The SRM-PSO combines the mutation mechanism and the self-regulating and self-perception strategy. Therefore, comparison among SRM-PSO, MPSO and SRPSO is necessary. In this section, SRM-PSO, MPSO and SRPSO are tested and compared on the CEC2017 test suite. The mean errors and standard deviations are presented in Table 4. The symbols ">" and "<" indicate that SRM-PSO "performs better than" and "performs worse than" the comparative algorithm, respectively. The row of "w/l" represents for the total number of SRM-PSO "wins over/loses to" the current comparison algorithm in the mean error result. The row of "*Best*" denotes the total number of the corresponding algorithm with the best mean error in the 30 function tests.

The results in Table 4 indicate that in the unimodal functions ($f_1 - f_3$) test, SRM-PSO achieves the first place in f_1 and f_2, while SRPSO gets first in f_3. Compared with MPSO, SRM-PSO performs better in f_1 and f_2, but worse in f_3. Compared with SRPSO, SRM-PSO also performs better in f_1 and f_2 and worse in f_3. Consequently, SRM-PSO is superior to MPSO and SRPSO in unimodal functions.

In the simple multimodal functions ($f_4 - f_{10}$) test, SRPSO performs better than MPSO and SRM-PSO in f_4. SRM-PSO performs better than MPSO and SRPSO to a large extent on the six functions of $f_5 - f_{10}$.

In the hybrid functions ($f_{11} - f_{20}$) test, SRM-PSO outperforms MPSO and SRPSO in seven function tests, and MPSO performs better than SRPSO and SRM-PSO in the other three functions.



(a) f_3 Function evaluation (b) f_{19} Function evaluation

Fig. 5 Diversity curve of the population

In the composition functions ($f_{21} - f_{30}$) test, MPSO, SRPSO and SRM-PSO obtain first place in one function, one function and eight functions, respectively.

In all thirty functions, MPSO, SRPSO and SRM-PSO show the preferable performance in four functions, three functions and twenty-three functions, respectively. Compared with MPSO, SRM-PSO performs greatly better in twenty-five functions and worse in five functions. Compared with SRPSO, SRM-PSO performs greatly better in twenty-seven functions and worse in three functions. Meanwhile, according to the results of Table 5, SRM-PSO ranks first and obtains the best rating, while SRPSO and MPSO are ranked in descending order. It can be seen from the above test results that PSO combined with mutation mechanism, self-regulating and self-perception strategy (SRM-PSO) is greatly superior to MPSO and SRPSO.

To intuitively reflect the effectiveness of the SRM-PSO algorithm, the convergence of related PSO variants in the three test functions are shown in Fig. 6. As shown in Fig. 6, MPSO maintains a steady but slow convergence due to mutation mechanism. SRPSO has a faster convergence than MPSO on account of its strategy. From the convergence curve, the mutation mechanism, self-regulation and self-perception strategy are effectively combined. The self-regulation and self-perception strategy improves the exploration and exploitation capabilities, and compensates the diminishing rate of convergence due to the mutation, then improve the performance of particle swarm optimization.

Further, related PSO variants are tested and compared in 50-dimensional (50-D) and 100-dimensional (100-D) problems. The test comparison results are shown in Table 6. In 50-D problem test, MPSO and SRPSO perform well respectively in four functions, whereas SRM-PSO displays preferable in twenty-two functions. In 100-D problem test, MPSO and SRPSO perform well in three functions and six functions respectively, whereas SRM-PSO displays favorable in twenty-one functions. It can be seen from the results that SRM-PSO is greatly better than MPSO and SRPSO. It also further verifies the effectiveness and superiority of the proposed SRM-PSO that combining the mutation mechanism with self-regulating and self-perception strategy. Meanwhile, the comparative results in different dimensions also shows the scalability of the SRM-PSO.

4.3.3 Comparison Test with PSO Variants

This section compares five different PSO variants to verify properties of the proposed SRM-PSO. All algorithms are tested by the CEC2017 test suite, and the results are presented in Table 7.

The Table 7 indicates that for the three unimodal functions ($f_1 - f_3$), CHCLPSO and PSOG yield the preferable performance in f_1 and f_2 respectively, whereas SRM-PSO obtains the favorable performance in f_3. Compared with CHCLPSO, SRM-PSO is superior in two functions and inferior in one. Therefore, SRM-PSO is superior to other comparative variants except PSOG.

For seven simple multimodal functions ($f_4 - f_{10}$), SRM-PSO, CHCLPSO and HFPSO achieve first place in four functions, two functions and one functions, respectively. Compared with CPSO, HFPSO and PSOG, SRM-PSO exhibits the favorable performance. For SLPSO, the performance of SRM-PSO has a greater advantage. The performance of CHCLPSO is slightly worse than the proposed SRM-PSO.

In the ten hybrid functions ($f_{11} - f_{20}$) test, SRM-PSO, CHCLPSO and HFPSO obtain the favorable performance

Table 4 Comparison test results of related PSO variants in 30-D CEC2017 functions

Function	MPSO		SRPSO		SRM-PSO	
	Mean	STD.	Mean	STD.	Mean	STD.
f_1	4.708E+03(>)	5.616E+03	1.407E+04(>)	4.461E+03	**3.242E+03**	3.750E+03
f_2	1.109E+24(>)	3.776E+21	2.046E+14(>)	3.939E+22	**1.780E+09**	7.551E+08
f_3	4.747E-02(<)	6.114E-02	**4.958E-05(<)**	2.544E-03	5.089E-01	9.858E-01
f_4	9.482E+01(>)	3.269E+01	**5.400E+01(<)**	3.159E+01	8.422E+01	1.082E+01
f_5	1.612E+02(>)	3.893E+01	1.291E+02(>)	2.836E+01	**4.019E+01**	1.093E+01
f_6	2.161E+01(>)	9.361E+00	2.391E+01(>)	8.024E+00	**2.222E-04**	2.532E-03
f_7	1.894E+02(>)	4.065E+01	1.319E+02(>)	2.380E+01	**7.980E+01**	1.229E+01
f_8	1.331E+02(>)	3.035E+01	9.826E+01(>)	1.984E+01	**4.061E+01**	1.032E+01
f_9	2.274E+03(>)	1.140E+03	1.408E+03(>)	7.092E+02	**2.887E+00**	2.681E+00
f_{10}	4.209E+03(>)	6.575E+02	2.986E+03(>)	6.277E+02	**2.337E+03**	4.457E+02
f_{11}	1.370E+02(>)	3.936E+01	1.021E+02(>)	2.972E+01	**4.116E+01**	2.985E+01
f_{12}	3.411E+05(>)	1.517E+05	1.380E+05(>)	6.865E+04	**4.726E+04**	3.358E+04
f_{13}	1.447E+04(>)	1.533E+04	1.937E+05(>)	1.573E+04	**1.215E+04**	2.064E+04
f_{14}	**9.092E+03(<)**	9.049E+03	2.021E+04(>)	8.395E+03	9.698E+03	8.746E+03
f_{15}	1.689E+04(>)	8.165E+03	1.603E+04(>)	8.075E+03	**7.621E+03**	9.151E+03
f_{16}	1.127E+03(>)	3.384E+02	8.917E+02(>)	2.312E+02	**4.027E+02**	2.091E+02
f_{17}	5.595E+02(>)	2.346E+02	3.405E+02(>)	1.682E+02	**1.096E+02**	5.873E+01
f_{18}	**8.690E+04(<)**	8.898E+04	3.382E+05(>)	8.843E+04	1.903E+05	1.435E+05
f_{19}	**6.844E+03(<)**	6.029E+03	1.152E+04(>)	5.732E+03	1.068E+04	1.329E+04
f_{20}	5.251E+02(>)	1.871E+02	3.379E+02(>)	1.019E+02	**1.293E+02**	7.950E+01
f_{21}	3.300E+02(>)	3.659E+01	3.189E+02(>)	2.821E+01	**2.424E+02**	1.445E+01
f_{22}	9.082E+02(>)	1.389E+03	2.457E+03(>)	1.627E+03	**1.850E+02**	4.226E+02
f_{23}	5.939E+02(>)	1.004E+02	5.880E+02(>)	7.172E+01	**3.890E+02**	9.927E+00
f_{24}	6.218E+02(>)	8.807E+01	6.688E+02(>)	6.403E+01	**4.616E+02**	1.081E+01
f_{25}	3.925E+02(>)	1.876E+01	3.933E+02(>)	1.532E+01	**3.873E+02**	3.233E-01
f_{26}	1.773E+03(>)	1.758E+03	2.123E+03(>)	1.567E+03	**1.194E+03**	5.123E+02
f_{27}	5.950E+02(>)	3.872E+01	5.813E+02(>)	5.432E+01	**5.011E+02**	9.469E+00
f_{28}	**3.566E+02(<)**	5.192E+01	4.969E+02(>)	5.737E+01	3.824E+02	5.587E+01
f_{29}	8.214E+02(>)	2.246E+02	8.164E+02(>)	2.029E+02	**4.974E+02**	7.252E+01
f_{30}	6.631E+03(>)	2.873E+03	**5.586E+03(<)**	3.163E+03	6.408E+03	3.587E+03
w/l	25/5		27/3			
Best	4		3		23	

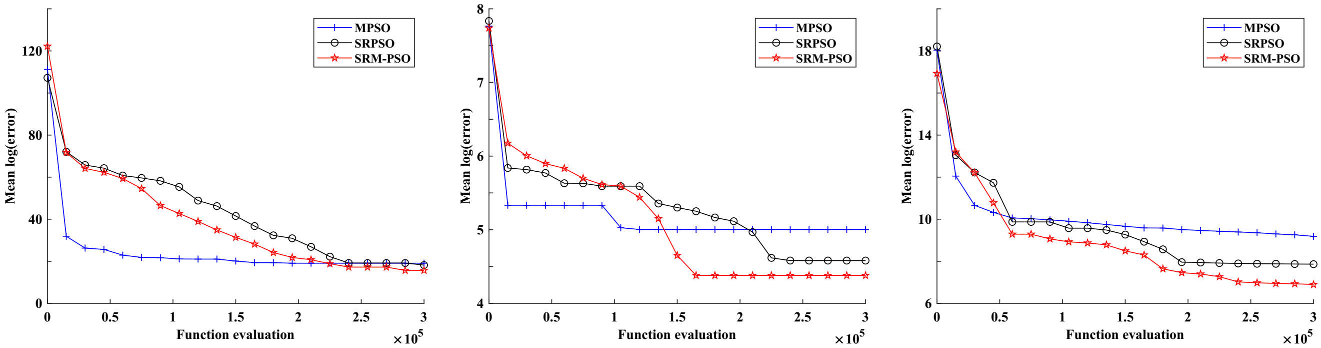
in four functions, three functions and two functions, respectively. PSOG only gets the first place in f_{12}. Generally speaking, the proposed SRM-PSO performs better than CPSO, SLPSO, PSOG and CHCLPSO, equally well with HFPSO.

For the composition functions ($f_{21} - f_{30}$), SRM-PSO exhibits the preferable performance in four functions. The CHCLPSO achieves the preferable performance in five functions. SLPSO only ranks first in one function. Consequently, the proposed SRM-PSO performs slightly worse than CHCLPSO, but greatly better than the remaining four variants in the composition functions test.

Table 5 Ranks of related PSO variants in 30-D CEC2017 functions

Algorithm	MPSO	SRPSO	SRM-PSO
Avg. rank	2.473	2.200	1.327
Final rank	3	2	1

In all thirty functions, CHCLPSO, SLPSO, HFPSO and PSOG exhibit the favorable performance in eleven functions, one function, three functions and two functions, respectively. Moreover, SRM-PSO obtain first place in remaining thirteen functions. The results of Table 8 shows that CHCLPSO has certain competitiveness. However, the SRM-PSO finally ranks the best, in first place. In conclusion, the performance of SRM-PSO is better than the involved five PSO variants. Therefore, the introducing mutation



(a) Convergence curves of f_2 (b) Convergence curves of f_7 (c) Convergence curves of f_{14}

Fig. 6 The convergence curves of related PSO variants

mechanism can effectively maintain the diversity of the population, and it can search more effectively with the combination of self-regulation and self-perception strategy.

4.3.4 Rate of Convergence

To show the convergence process more visually, the convergence curves of the top three algorithms (SRM-PSO, CHCLPSO, and SLPSO) in 30-D CEC2017 functions are shown in Fig. 7. The six representative functions are selected, namely, unimodal function f_3, simple multimodal function f_7, two hybrid function f_{16}, f_{20} and two composition functions f_{23}, f_{27}, respectively.

In unimodal function f_3, it can be seen from Fig. 7(a) that CHCLPSO and SLPSO converge faster in the early stage, but is surpassed by SRM-PSO in the later stage. Due to the existence of mutation mechanism and self-perception strategy, SRM-PSO converges slowly in the early stage. From the middle stage, the self-regulating inertia weight increases the convergence speed of SRM-PSO, surpassing the other comparison algorithms, and finally achieves the lowest mean error.

For simple multimodal function f_7, Fig. 7(b) shows that, SRM-PSO and CHCLPSO maintain a steady convergence rate in the early stage. SLPSO converges quickly in the beginning due to its special search strategies, and then falls into the local optima. SRM-PSO accelerates the convergence rate in the middle stage and finally obtains better results than other algorithms.

For hybrid function, it can be seen from Fig. 7(c) that three algorithms maintain a similar trend in the early convergence process of f_{16}. From the middle stage, the proposed SRM-PSO performs a faster convergence rate and obtained better results than CHCLPSO and SLPSO. Figure 7(d) indicates that, for f_{20}, SRM-PSO shows a steady and more faster convergence rate than the other two algorithms.

For composition functions f_{23}, Fig. 7(e) shows that the convergence rates of CHCLPSO and SRM-PSO are similar before the middle iteration. After the midterm, CHCLPSO converges slowly and the final result is slightly worse than SRM-PSO. SLPSO converges quickly and then stopped early. In f_{27}, the convergence trends of SRM-PSO, CHCLPSO and SLPSO are almost the same, and finally SRM-PSO yields the lowest mean error.

In general, the convergence rate of SRM-PSO in various functions is slow but steady in the early stage, which indicates the proposed SRM-PSO can search for more potential solutions. The convergence speed of SRM-PSO will be gradually accelerated with the search progress, and the SRM-PSO could result in a high accuracy result. Meanwhile, SRM-PSO can be applied to various functions and shows high accuracy performances.

4.3.5 Sensitivity Analysis

As mentioned in Section 2.1, several classical mutation mechanisms are introduced, including Gaussian mutation (GMPSO, [36]), Cauchy mutation (CMPSO, [37]), and adaptive mutation (AMPSO, [38]). Therefore, several PSOs with the aforementioned mutation mechanisms are tested and compared by CEC2017 test suite in this section. The results are given in Table 9. In all thirty functions, CMPSO, AMPSO and SRM-PSO exhibit the favorable performance in one, four and twenty-five functions, respectively. This result also indicates that the SRM-PSO is greatly better than GMPSO, CMPSO and AMPSO.

We further conducted a sensitivity analysis experiment on the number of mutations in the particles of proposed SRM-PSO. The six representative functions are selected for test, namely, unimodal function f_3, simple multimodal function f_7, two hybrid function f_{16}, f_{20} and two composition functions f_{21}, f_{26} in the 30-D CEC2017 functions. The experimental comparison results of different numbers of mutant particles are shown in Table 10.

The test results in Table 10 show that the mean error increases with the increase of the number of mutated

Table 6 Comparison test results of related PSO variants in high dimensional CEC2017 functions

Function	50D						100D					
	MPSO		SRPSO		SRM-PSO		MPSO		SRPSO		SRM-PSO	
	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.
f_1	3.406E+07(>)	3.573E+03	5.330E+07(>)	2.359E+03	**6.058E+03**	2.388E+03	1.042E+08(>)	7.559E+03	1.362E+08(>)	6.004E+03	**9.965E+03**	6.386E+03
f_2	4.652E+44(>)	2.324E+19	2.196E+48(>)	9.579E+09	**1.361E+26**	4.370E+19	2.356E+82(>)	5.424E+66	**3.608E+79**(<)	8.480E+45	9.601E+79	1.048E+66
f_3	**4.668E+03**(<)	3.558E+03	5.288E+03(<)	4.880E+03	1.067E+04	1.013E+04	**1.444E+05**(<)	1.393E+05	1.896E+05(<)	1.822E+05	2.165E+05	2.134E+05
f_4	1.031E+02(<)	1.013E+02	**9.242E+01**(<)	1.009E+02	1.187E+02	1.299E+02	2.943E+02(>)	2.828E+02	2.709E+02(>)	2.625E+02	**2.423E+02**	2.340E+02
f_5	2.916E+02(>)	2.846E+02	2.368E+02(>)	2.383E+02	**9.116E+01**	9.154E+01	7.404E+02(>)	7.412E+02	6.012E+02(>)	6.169E+02	**2.310E+02**	2.278E+02
f_6	2.311E+01(>)	2.218E+01	3.111E+01(>)	3.173E+01	**4.919E-02**	3.418E-02	3.196E+01(>)	3.149E+01	4.421E+01(>)	4.394E+01	**1.702E+00**	1.474E+00
f_7	4.196E+02(>)	3.997E+02	2.454E+02(>)	2.372E+02	**1.678E+02**	1.630E+02	1.292E+03(>)	1.278E+03	7.411E+02(>)	7.349E+02	**4.535E+02**	4.531E+02
f_8	2.989E+02(>)	2.945E+02	2.263E+02(>)	2.234E+02	**8.791E+01**	8.756E+01	8.196E+02(>)	8.039E+02	6.572E+02(>)	6.472E+02	**2.270E+02**	2.236E+02
f_9	9.642E+03(>)	9.745E+03	6.249E+03(>)	6.195E+03	**5.989E+01**	4.921E+01	2.389E+04(>)	2.191E+04	1.847E+04(>)	1.684E+04	**7.130E+02**	6.192E+02
f_{10}	6.396E+03(>)	6.432E+03	5.388E+03(>)	5.440E+03	**5.325E+03**	5.029E+03	1.446E+04(<)	1.448E+04	**1.332E+04**(<)	1.332E+04	1.506E+04	1.316E+04
f_{11}	2.335E+02(>)	2.140E+02	1.545E+02(>)	1.522E+02	**9.191E+01**	9.090E+01	1.346E+03(>)	1.272E+03	8.955E+02(>)	8.961E+02	**7.999E+02**	7.572E+02
f_{12}	3.523E+06(>)	7.705E+05	**5.744E+05**(<)	4.749E+05	1.179E+06	1.095E+06	1.995E+07(>)	3.744E+06	1.759E+07(>)	2.312E+06	**3.913E+06**	3.497E+06
f_{13}	1.864E+05(>)	2.954E+03	**4.694E+03**(<)	3.202E+03	1.056E+04	6.599E+03	8.355E+03(>)	7.426E+03	1.295E+05(>)	4.074E+03	**7.387E+03**	3.705E+03
f_{14}	**3.925E+04**(<)	3.359E+04	4.346E+04(<)	3.354E+04	5.916E+04	5.331E+04	**4.199E+05**(<)	4.007E+05	4.303E+05(<)	4.317E+05	7.669E+05	6.908E+05
f_{15}	9.964E+03(>)	9.296E+03	6.559E+03(>)	4.153E+03	**6.475E+03**	4.210E+03	2.510E+03(<)	1.588E+03	**2.365E+03**(<)	1.111E+03	4.150E+03	2.249E+03
f_{16}	1.726E+03(>)	1.694E+03	1.443E+03(>)	1.390E+03	**1.031E+03**	1.026E+03	4.030E+03(>)	4.093E+03	3.899E+03(>)	3.834E+03	**2.898E+03**	2.819E+03
f_{17}	1.527E+03(>)	1.510E+03	1.190E+03(>)	1.153E+03	**8.247E+02**	8.488E+02	3.551E+03(>)	3.552E+03	3.198E+03(>)	3.172E+03	**2.474E+03**	2.438E+03
f_{18}	**1.694E+05**(<)	1.408E+05	4.153E+05(<)	3.018E+05	8.475E+05	4.461E+05	**8.569E+05**(<)	7.800E+05	1.346E+06(<)	1.189E+06	2.203E+06	1.799E+06
f_{19}	1.842E+04(>)	1.579E+04	1.627E+04(>)	1.401E+04	**1.205E+04**	1.141E+04	4.199E+03(<)	1.372E+03	**3.686E+03**(<)	1.563E+03	6.639E+03	2.929E+03
f_{20}	1.156E+03(>)	1.089E+03	7.715E+02(>)	7.328E+02	**5.802E+02**	6.227E+02	3.259E+03(>)	3.243E+03	**2.563E+03**(<)	2.562E+03	3.112E+03	2.927E+03
f_{21}	5.095E+02(>)	5.039E+02	4.502E+02(>)	4.372E+02	**2.932E+02**	2.915E+02	1.207E+03(>)	1.208E+03	1.071E+03(>)	1.079E+03	**4.841E+02**	4.836E+02
f_{22}	**4.028E+03**(<)	6.020E+03	6.105E+03(>)	6.700E+03	4.710E+03	5.229E+03	1.640E+04(>)	1.614E+04	1.566E+04(>)	1.560E+04	**1.551E+04**	1.453E+04
f_{23}	1.001E+03(>)	9.859E+02	9.199E+02(>)	9.110E+02	**5.162E+02**	5.128E+02	1.692E+03(>)	1.703E+03	2.086E+03(>)	2.064E+03	**7.569E+02**	7.525E+02
f_{24}	9.952E+02(>)	9.826E+02	1.107E+03(>)	1.104E+03	**5.884E+02**	5.833E+02	2.107E+03(>)	2.089E+03	2.780E+03(>)	2.693E+03	**1.129E+03**	1.124E+03
f_{25}	5.550E+02(>)	5.632E+02	5.585E+02(>)	5.678E+02	**5.064E+02**	4.804E+02	7.921E+02(>)	7.900E+02	**7.576E+02**(<)	7.689E+02	7.660E+02	7.715E+02
f_{26}	4.248E+03(>)	5.488E+03	4.064E+03(>)	5.367E+03	**2.054E+03**	2.128E+03	1.495E+04(>)	1.713E+04	1.352E+04(>)	1.556E+04	**5.748E+03**	5.735E+03
f_{27}	9.897E+02(>)	9.955E+02	9.509E+02(>)	8.978E+02	**5.486E+02**	5.477E+02	1.166E+03(>)	1.149E+03	1.162E+03(>)	1.102E+03	**6.828E+02**	6.828E+02
f_{28}	5.081E+02(>)	5.025E+02	5.135E+02(>)	5.025E+02	**4.746E+02**	4.588E+02	6.264E+02(>)	6.165E+02	6.460E+02(>)	6.183E+02	**5.677E+02**	5.648E+02
f_{29}	1.786E+03(>)	1.829E+03	1.695E+03(>)	1.676E+03	**6.410E+02**	6.001E+02	4.328E+03(>)	4.358E+03	4.089E+03(>)	4.070E+03	**2.674E+03**	2.722E+03
f_{30}	1.019E+06(>)	9.383E+05	**9.239E+05**(<)	8.802E+05	9.606E+05	9.176E+05	8.484E+04(>)	4.025E+04	6.162E+04(>)	1.664E+04	**7.317E+03**	5.566E+03
w/l	25/5		23/7				24/6		21/9			
$Best$	4		4		22		3		6		21	

Table 7 Comparison test results of six PSO variants in 30-*D* CEC2017 functions

Function	CPSO		SLPSO		HFPSO		PSOG		CHCLPSO		SRM-PSO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
f_1	7.494E+09(>)	6.112E+09	6.760E+03(>)	5.820E+03	6.270E+08(>)	2.330E+08	2.340E+03(<)	2.243E+03	**1.586E+02**(<)	3.051E+02	3.242E+03	3.750E+03
f_2	1.415E+37(>)	8.174E+37	2.780E+11(>)	1.240E+12	8.740E+29(>)	6.105E+30	**7.930E+08**(<)	6.080E+08	1.106E+12(>)	2.935E+12	1.780E+09	7.551E+08
f_3	5.126E+03(>)	1.307E+04	4.220E+04(>)	1.080E+04	4.120E+03(>)	1.440E+03	4.620E+02(>)	5.413E+01	9.213E+03(>)	3.354E+03	**5.089E-01**	9.858E-01
f_4	9.730E+02(>)	1.192E+03	6.679E+01(<)	3.220E+01	4.570E+02(>)	2.970E+01	4.190E+02(>)	4.660E+00	**6.448E+01**(<)	2.232E+01	8.422E+01	1.082E+01
f_5	1.376E+02(>)	3.600E+01	6.921E+01(>)	2.064E+01	6.010E+02(>)	1.360E+01	5.630E+02(>)	1.251E+01	4.902E+01(>)	8.404E+00	**4.019E+01**	1.093E+01
f_6	1.259E+01(>)	6.770E+00	7.900E-05(<)	4.253E-04	6.290E+02(>)	7.630E+00	6.100E+02(>)	1.610E+00	**1.956E-07**(<)	7.000E-07	2.222E-04	2.532E-03
f_7	1.846E+02(>)	8.138E+01	1.071E+02(>)	1.588E+01	8.150E+02(>)	6.890E+00	8.100E+02(>)	1.571E+01	8.822E+01(>)	9.430E+00	**7.980E+01**	1.229E+01
f_8	1.300E+02(>)	3.173E+01	6.783E+01(>)	2.622E+01	8.660E+02(>)	3.580E+00	8.700E+02(>)	8.690E+00	4.988E+01(>)	8.406E+00	**4.061E+01**	1.032E+01
f_9	2.416E+03(>)	1.671E+03	1.440E+02(>)	1.200E+02	1.060E+03(>)	1.020E+02	9.000E+02(>)	2.000E-02	1.958E+02(>)	1.041E+02	**2.887E+00**	2.681E+00
f_{10}	3.621E+03(>)	6.640E+02	2.934E+03(>)	5.110E+02	**2.050E+03**(<)	2.120E+02	3.680E+03(>)	2.682E+02	2.055E+03(<)	3.400E+02	2.337E+03	4.457E+02
f_{11}	3.178E+02(>)	1.820E+02	7.831E+01(>)	3.450E+01	1.120E+03(>)	8.450E+00	1.230E+03(>)	1.209E+01	5.756E+01(>)	2.513E+01	**4.116E+01**	2.985E+01
f_{12}	5.929E+08(>)	7.246E+08	8.080E+05(>)	9.620E+05	7.650E+05(>)	4.300E+05	**3.260E+04**(<)	2.623E+03	3.998E+05(>)	3.115E+05	4.726E+04	3.358E+04
f_{13}	1.592E+08(>)	5.000E+08	1.360E+04(>)	1.330E+04	6.170E+03(<)	2.040E+03	8.430E+04(>)	6.386E+03	**6.017E+02**(<)	9.905E+02	1.215E+04	2.064E+04
f_{14}	8.975E+04(>)	2.659E+05	2.880E+04(>)	4.090E+04	**4.370E+03**(<)	1.420E+03	7.390E+03(<)	4.281E+02	2.234E+04(>)	1.802E+04	9.698E+03	8.746E+03
f_{15}	6.551E+04(>)	5.717E+04	1.000E+04(>)	1.360E+04	1.600E+03(<)	1.710E+02	4.360E+04(>)	2.806E+03	**4.546E+02**(<)	5.353E+02	7.621E+03	9.151E+03
f_{16}	1.347E+03(>)	3.436E+02	6.090E+02(>)	2.230E+02	1.970E+03(>)	2.370E+01	1.860E+03(>)	1.605E+01	6.317E+02(>)	1.694E+02	**4.027E+02**	2.091E+02
f_{17}	6.455E+02(>)	2.786E+02	1.560E+02(>)	8.360E+01	1.700E+03(>)	9.580E+00	1.870E+03(>)	1.014E+01	2.095E+02(>)	1.028E+02	**1.096E+02**	5.873E+01
f_{18}	1.146E+06(>)	2.189E+06	3.120E+05(>)	2.260E+05	**1.850E+03**(<)	5.660E+01	8.680E+04(<)	4.720E+03	1.970E+05(>)	1.463E+05	1.903E+05	1.435E+05
f_{19}	9.783E+06(>)	3.619E+07	1.660E+04(>)	1.710E+04	1.900E+03(<)	5.350E+01	3.170E+04(>)	1.567E+03	**2.868E+02**(<)	3.138E+02	1.068E+04	1.329E+04
f_{20}	4.675E+02(>)	1.946E+02	1.860E+02(>)	1.040E+04	2.090E+03(>)	4.530E+01	2.330E+03(>)	1.646E+01	2.431E+02(>)	1.015E+02	**1.293E+02**	7.950E+01
f_{21}	3.569E+02(>)	3.627E+01	2.610E+02(>)	2.030E+01	4.162E+02(>)	2.536E+01	2.360E+03(>)	1.258E+01	**2.402E+02**(<)	3.911E+01	2.424E+02	1.445E+01
f_{22}	3.307E+03(>)	1.493E+03	9.800E+02(>)	1.360E+03	3.609E+03(>)	3.348E+03	2.300E+03(>)	4.550E+00	**1.169E+02**(<)	3.402E+01	1.850E+02	4.226E+02
f_{23}	6.407E+02(>)	7.024E+01	4.270E+02(>)	2.500E+01	6.007E+02(>)	3.216E+01	2.720E+03(>)	1.842E+01	4.044E+02(>)	9.398E+00	**3.890E+02**	9.927E+00
f_{24}	7.081E+02(>)	8.129E+01	5.030E+02(>)	2.440E+01	6.513E+02(>)	2.729E+01	2.860E+03(>)	1.901E+01	5.082E+02(>)	1.599E+01	**4.616E+02**	1.081E+01
f_{25}	6.270E+02(>)	2.056E+02	3.940E+02(>)	1.420E+01	5.848E+02(>)	4.607E+01	2.890E+03(>)	1.548E+01	**3.864E+02**(<)	1.606E+00	3.873E+02	3.233E-01
f_{26}	3.648E+03(>)	1.221E+03	1.980E+03(>)	2.570E+02	2.268E+03(>)	1.084E+03	3.590E+03(>)	3.821E+01	**4.260E+02**(<)	4.608E+02	1.194E+03	5.123E+02
f_{27}	6.330E+02(>)	6.999E+01	5.320E+02(>)	1.360E+01	5.811E+02(>)	3.662E+01	3.221E+03(>)	1.930E+01	5.159E+02(>)	7.278E+00	**5.011E+02**	9.469E+00
f_{28}	1.110E+03(>)	7.831E+02	**3.450E+02**(<)	6.240E+01	6.422E+02(>)	4.719E+01	3.105E+03(>)	1.071E+01	4.034E+02(>)	4.553E+00	3.824E+02	5.587E+01
f_{29}	1.261E+03(>)	3.532E+02	6.200E+02(>)	1.220E+02	1.529E+03(>)	2.520E+02	3.823E+03(>)	3.176E+01	5.680E+02(>)	8.851E+01	**4.974E+02**	7.252E+01
f_{30}	1.988E+06(>)	3.342E+06	9.540E+03(>)	4.580E+03	1.172E+08(>)	5.130E+07	6.471E+04(>)	2.060E+03	**4.264E+03**(<)	9.753E+02	6.408E+03	3.587E+03
w/l	30/0		27/3		24/6		25/5		18/12			
Best	0		1		3		2		11		13	

Table 8 Ranks of six PSO variants in 30-D CEC2017 functions

Algorithm	CPSO	SLPSO	HFPSO	PSOG	CHCLPSO	SRM-PSO
Avg. rank	5.037	3.337	4.273	4.117	2.348	1.889
Final rank	6	3	5	4	2	1

particles. The lowest mean error occurs when there is only one particle mutation. Meanwhile, similar phenomena can be observed in other functions. Hence, SRM-PSO can achieve satisfactory results by mutating only one particle.

4.4 Real-World Optimization Problems

In this section, the feasibility of the proposed SRM-PSO algorithm in real-world optimization problems is investigated. Four common engineering problem are employed for testing, namely (F_1) parameter estimation for Frequency-Modulated (FM) sound waves, (F_2) Lennard-Jones Potential Problem and (F_3) spread spectrum radar polly phase code design. Meanwhile, simple and effective PSO variants are employed for comparison. Finally, the performance and competitiveness of SRM-PSO in the comparison algorithm are analyzed.

4.4.1 Parameter Estimation for Frequency-Modulated (Fm) Sound Waves

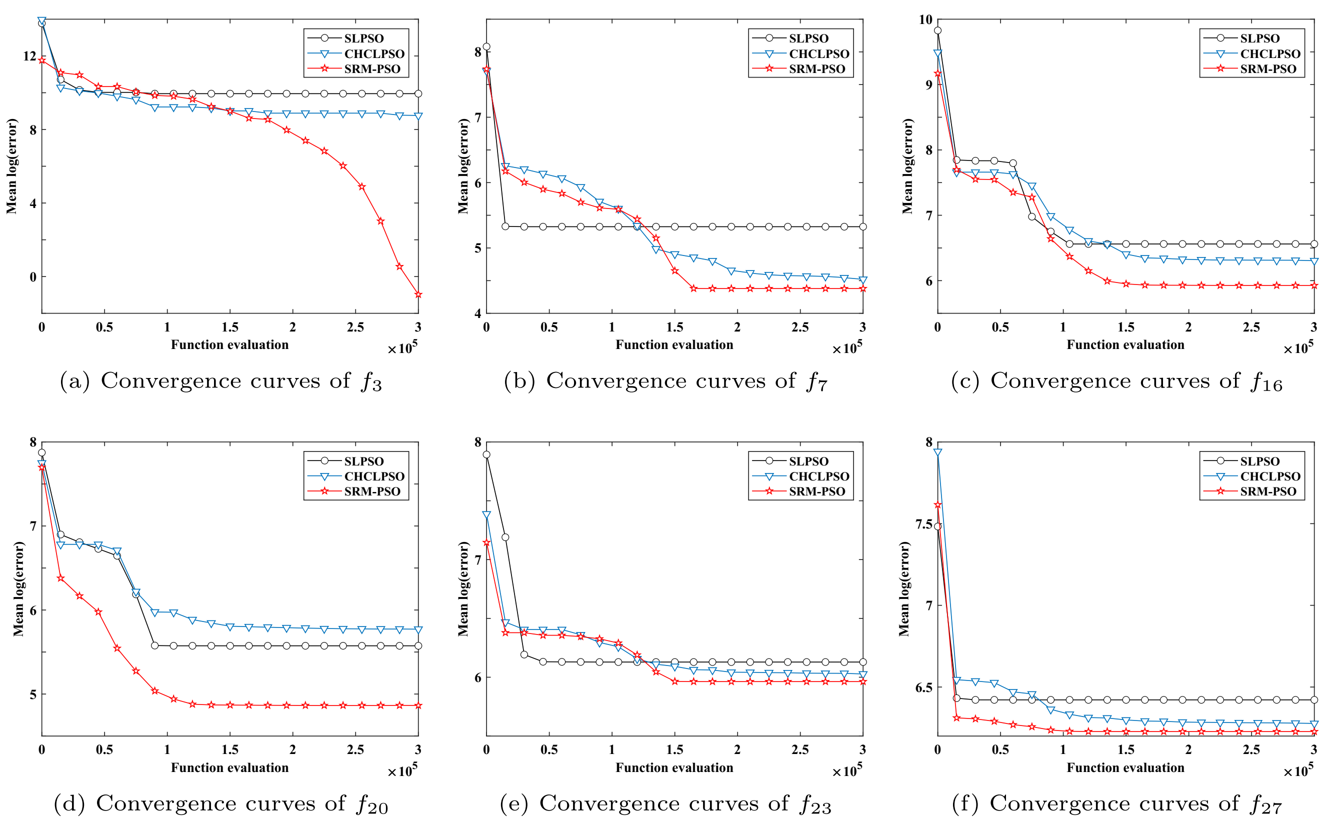
In this problem [47], the aim is to determine the optimal parameters of six decision variables namely a_1, b_1, a_2, b_2, a_3, b_3 for FM synthesizer. This problem is a highly complex multimodal in nature. Mathematical model of this problem is provided as:

$$Minimize\ f(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{20}$$

where $\vec{X} = \{a_1, b_1, a_2, b_2, a_3, b_3\}$, the parameters are defined in the range [-6.4, 6.35]. The $y(t)$ and $y_0(t)$ is:

$$y(t) = a_1 \sin(b_1 t\theta + a_2 \sin(b_2 t\theta + a_3 \sin(b_3 t\theta))) \tag{21}$$

$$y_0(t) = \sin(5t\theta - 1.5\sin(4.8t\theta + 2\sin(4.9t\theta))) \tag{22}$$



(a) Convergence curves of f_3 (b) Convergence curves of f_7 (c) Convergence curves of f_{16}

(d) Convergence curves of f_{20} (e) Convergence curves of f_{23} (f) Convergence curves of f_{27}

Fig. 7 The convergence curves of six representative CEC2017 functions

Table 9 Comparison test results of different mutation mechanisms in 30-D CEC2017 functions

Function	GMPSO	CMPSO	AMPSO	SRM-PSO
	Number of Best Mean Error			
$f_1 - f_3$	0	0	1	2
$f_4 - f_{10}$	0	0	0	7
$f_{11} - f_{20}$	0	1	3	6
$f_{21} - f_{30}$	0	0	0	10
w/l	30/0	29/1	25/5	
$Best$	0	1	4	25

where the value of θ is constant and equal to $2\pi/100$. The obtained numerical results are listed in Table 10. The results are achieved by conducting 30 runs and utilizing 60,000 function evaluations.

4.4.2 Lennard-Jones Potential Problem

This is a potential energy minimization problem that involves the minimization of molecular potential energy associated with pure Lennard-Jones (LJ) cluster [47]. Moreover, it is a multi-modal optimization problem comprised of an exponential number of local minima. The mathematical model of the Leonard-Jones pair potential for N atoms is:

$$\vec{p}_i = \{\vec{x}_i, \vec{y}_i, \vec{z}_i\},\ i = 1,\ \ldots,\ N \tag{23}$$

$$V_n(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(r_{ij}^{-12} - 2r_{ij}^{-6}\right) \tag{24}$$

where r_{ij}=$\|\vec{p}_j - \vec{p}_i\|$ with gradient:

$$\nabla_j V_n(p) = -12 \sum_{i=1, i\neq j}^{N} \left(r_{ij}^{-14} - r_{ij}^{-8}\right) \left(\vec{p}_j - \vec{p}_i\right) \tag{25}$$

Lennard-Jones potential has minimum value at a particular distance between the two points. In this test, L-J potential is a ten atom problem, i.e., 30 dimension. The maximum function evaluation is set as 150,000.

4.4.3 Spread Spectrum Radar Polly Phase Code Design

Pulse compression is the most widely used technology in radar system. Polyphase codes are competitive for the lower side-lobes in signal compression and easier implementation of digital processing techniques. The polyphase code design problem can be modeled as a continuous minimum and maximum nonlinear non-convex optimization problem [35]. It is a continuous minimum maximum global optimization problem with a large number of local optimal solutions. Its mathematical model is defined as:

$$Global\ \min\ f(x) = \max\{\Phi_1(X), \ldots,\ \Phi_{2m}(X)\} \tag{26}$$

where $X = \{(x_1, x_2, \ldots, x_D) \in R_D | 0 \leq x_j \leq 2\pi\}, m = 2D - 1$,and $\Phi(X)$ is:

$$\Phi_{2i-1}(X) = \sum_{j=1}^{D} \cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k\right) \tag{27}$$

$$\Phi_{2i}(X) = 0.5 + \sum_{j=i+1}^{D} \cos\left(\sum_{k=|2i-j|+1}^{j} x_k\right) \tag{28}$$

$$\Phi_{m+i}(X) = -\Phi_i(X),\ i = 1,\ 2,\ \ldots,\ m \tag{29}$$

where x_k represents symmetric phase differences. The ultimate purpose of this problem is to minimize the biggest sample autocorrelation function Φ among the modules. To determine the solution, 30 trials are performed with 200,000 function evaluations.

Table 10 The mutation particle number analysis on selected 30-D CEC2017 function

Function	The mutation particle number (n)							
	n=1		n=2		n=5		n=10	
	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.
f_3	**5.089E-01**	**9.858E-01**	7.038E-01	1.178E+00	9.641E-01	1.956E+00	1.240E+00	3.094E+00
f_7	**7.980E+01**	**1.229E+01**	8.233E+01	1.342E+01	8.153E+01	1.441E+01	8.163E+01	1.346E+01
f_{16}	**4.027E+02**	**2.091E+02**	4.700E+02	2.159E+02	4.875E+02	2.544E+02	4.809E+02	2.172E+02
f_{20}	**1.293E+02**	7.950E+01	1.331E+02	8.032E+01	1.394E+02	**7.152E+01**	1.435E+02	7.496E+01
f_{21}	**2.424E+02**	**1.445E+01**	2.456E+02	1.449E+01	2.458E+02	1.517E+01	2.455E+02	1.506E+01
f_{26}	**1.194E+03**	**5.123E+02**	1.248E+03	5.581E+02	1.260E+03	5.508E+02	1.274E+03	5.632E+02

Table 11 Comparison results on four real-world optimization problems

Algorithm		SRM-PSO	MPSO[32]	SRPSO[30]	FIPS[22]	CLPSO[24]	SLPSO[44]	CPSO[43]
F_1	Mean	**8.789E+00**	2.041E+01	1.093E+01	3.118E+01	2.269E+01	1.640E+01	1.655E+01
	STD.	5.784E+00	4.697E+00	6.520E+00	**2.330E+00**	2.341E+00	5.200E+00	4.829E+00
	Median	**1.111E+01**	2.185E+01	1.144E+01	3.074E+01	2.354E+01	1.874E+01	1.606E+01
F_2	Mean	**-2.266E+01**	-1.982E+01	-1.389E+01	-2.282E+00	-4.661E+00	-1.960E+01	-4.937E+00
	STD.	1.134E+00	4.297E+00	2.979E+00	9.743E-01	**7.173E-01**	8.450E-01	9.976E-01
	Median	**-2.313E+01**	-2.206E+01	-1.306E+01	-2.033E+00	-4.676E+00	-1.830E+01	-4.765E+00
F_3	Mean	8.494E-01	1.580E+00	1.128E+00	2.901E+00	2.095E+00	**8.359E-01**	1.938E+00
	STD.	1.735E-01	6.529E-01	5.195E-01	3.566E-01	1.722E-01	**1.407E-01**	8.048E-01
	Median	8.445E-01	1.392E+00	8.607E-01	2.949E+00	2.102E+00	**8.410E-01**	1.515E+00
F_4	Mean	5.772E-08	1.747E-06	4.239E-07	8.145E-07	6.290E-06	**5.119E-08**	6.699E-05
	STD.	**2.614E-07**	8.171E-06	2.214E-06	3.206E-06	1.691E-05	2.642E-07	3.653E-04
	Median	**4.863E-14**	2.767E-08	1.243E-12	5.876E-09	2.514E-07	2.186E-11	2.183E-10

4.4.4 Inverse Kinematics Calculation of 7-DOF Manipulator

The inverse kinematics of the manipulator is to obtain the required joint angles through a mathematical model when the Cartesian space position of the target is known. The 7-DOF manipulator improves the flexibility of the robot and increases the difficulty of inverse kinematics calculations. The kinematic model of the manipulator is expressed by the D-H matrix [48]:

$$T_{End} = {}_0^1T \cdot {}_1^2T \cdot {}_2^3T \cdot {}_3^4T \cdot {}_4^5T \cdot {}_5^6T \cdot {}_6^7T = {}_0^7T \tag{30}$$

$${}_0^7T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

The transformation matrix ${}_0^7T$ contains variables such as joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7)$, hence the fitness value calculated by inverse kinematics is to minimize the following function:

$$fitness = \sqrt{(\overline{p_x} - p_x)^2 + (\overline{p_y} - p_y)^2 + (\overline{p_z} - p_z)^2} \tag{32}$$

where $(\overline{p_x}, \overline{p_y}, \overline{p_z})$ is the desired target position. The maximum function evaluation is set as 40,000.

4.4.5 Test Results of the Four Real-World Problems

The test results over 30 independent runs are presented in Table 11, which contains the values of mean fitness (Mean), standard deviation (STD.) and median fitness (Median).

For the FM parameter estimation, the proposed SRM-PSO achieves the first place in the mean value and median value. FIPS achieves lowest standard deviation and exhibits robust performance, but its search accuracy is worse than SRM-PSO. In the Lennard-Jones potential problem,

Table 12 Literature summary of PSO variants for robotics problems

Reference	Algorithm	Test problem	Result
[49]	Inertia Weight PSO	7-DOF serial robot manipulator inverse kinematics problem	Global-local best inertia weight PSO can be efficiently used for inverse kinematics solution
[50]	A novel multi-swarm hybrid FOA-PSO	Target searching in unknown environments	The proposed approach performs well in multi-target searching multi-swarm robots and large environments
[51]	A quantum behaved particle swarm algorithm	7-DOF serial robot manipulator inverse kinematics problem	QPSO can obtains satisfactory inverse solution quality with short computation time, fewer iterations, and the number of particles
[52]	An improved PSO algorithm	Path planning of mobile robot	An improved PSO algorithm effectively combining the continuous high-degree Bezier curve for smooth path planning
[48]	A random descending velocity inertia weight PSO	7-DOF serial robot manipulator inverse kinematics problem	The result obtained is improved thousands of times with very small movements

SRM-PSO yields the most favorable performance among the comparison algorithms in terms of mean and median value. CLPSO has the minimum standard deviation and shows robustness. For the spread spectrum radar polly phase problem, SLPSO achieves the favorable performance in three numerical values. SRM-PSO exhibits its competitive searching accuracy in solving this problem and successfully achieves the second place. SRM-PSO is only slightly inferior to SLPSO, and outperforms other comparison variants. In the inverse kinematics calculation, SLPSO is slightly better than SRM-PSO in mean fitness, while SRM-PSO shows preferable results in the other two numerical values. Although the results of SRM-PSO in four practical problems are not all best, it can achieve high accuracy and satisfactory robustness. Therefore, it can be concluded that the application of SRM-PSO in practical optimization problems is feasible and efficient.

Due to the complexity of robotics problems, classical methods are usually complicated or unsolvable, while the improved particle swarm optimization algorithms can produce effective results in a short time. The development of these intelligent optimization algorithms has provided great help to the solution of problems in the field of robotics (Table 12).

5 Discussion and Conclusions

In this paper, a new PSO variant, SRM-PSO, is proposed by combining the mutation mechanism, self-regulating inertia weight and self-perception strategy. The strategies are to improve the inaccuracy performance caused by the premature convergence of PSO in real-world problems, such as, estimating for the uncertainties of the robotic control system. The mutation mechanism can maintain the diversity of the population during the search, and overcome the local optima. The self-regulating inertia weight and self-perception strategies can enhance the exploration and exploitation capabilities of the population. To test the effectiveness and superiority of the SRM-PSO, the CEC2017 test suite with 30 benchmark functions are adopted. In particular, two levels of comparisons are conducted orderly, including related PSO variants comparison, and the latest PSO variants comparison. Compared with related PSO variants, the tested results show the effectiveness of the combination of mutation mechanism, self-regulating inertia weight and self-perception strategies. Compared with five popular PSO variants, the test results also show that the SRM-PSO is better. From the convergence curves of the six representative functions, although the convergence rate of SRM-PSO is relatively slow in the early stage, it can maintain a steady convergence rate and achieve a favorable result in the end. At last, this paper adopts SRM-PSO to test actual problems in the real world and compares with six variants. The comparison results verify the feasibility and superiority of SRM-PSO in practical optimization problems.

For future studies, more intelligent mutation operators, self-regulation and self-perception deserve further research. Various parameter collocations can adaptively adjust each part of the proposed algorithm to improve the performances (such as efficiency, accuracy, and etc.) in challenging image recognition and robotic control system.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Project No. 61803089), the Natural Science Foundation of Fujian Province (No. 2019J01213). The authors thank Anping Lin for sharing code.

References

1. Maharana, H.S., Dash, S.K.: Comparative Optimization Analysis of Ramp Rate Constriction Factor Based PSO and Electro Magnetism Based 1 for Economic Load Dispatch in Electric Power System. In: 2019 International Conference on Applied Machine Learning (ICAML), pp. 63–68 (2019)
2. Pham, T.X., Siarry, P., Oulhadj, H.: Segmentation of mr brain images through hidden markov random field and hybrid metaheuristic algorithm. IEEE Trans. Image Process. **29**, 6507–6522 (2020)
3. Wang, Y., Chen, Y., Miao, Z.: Obstacle avoidance planning for de-icing robot based on PSO-fuzzy. Control Engineering of China **21**(2), 298–302 (2014)
4. Li, X., Chen, Y., Yu, X.: Parameters Optimization of Mechanical Arm PID Controllers Based on CLPSO Algorithm. In: 2011 6Th International Conference on Computer Science & Education (ICCSE), pp. 56–60 (2011)
5. Bai, X., Gao, X., Xue, B.: Particle Swarm Optimization Based Two-Stage Feature Selection in Text Mining. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8 (2018)
6. Ramos-Figueroa, O., Quiroz-Castellanos, M., Mezura-Montes, E., Schtze, O.: Metaheuristics to solve grouping problems: A review and a case study. Swarm Evol. Comput. **100643**, 53 (2020)
7. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, Q Rev Biol. 69 (1) 126C137 (Mar 1994)
8. Wu, G., Shen, X., Li, H., Chen, H., Lin, A., Suganthan, P.N.: Ensemble of differential evolution variants. Inf. Sci. **423**, 172–186 (2018)
9. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**, 459–471 (2007)
10. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer, adv. Eng. Softw. **69**, 46–61 (2007)
11. Wu, G.: Across neighbourhood search for numerical optimization. Inf. Sci. **329**, 597–618 (2016)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, 4, pp. 1942–1948 (1995)
13. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. World Congress on Computational Intelligence (Cat. No. 98TH8360), pp. 69–73 (1998)
14. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary

Computation-CEC99 (Cat. No. 99TH8406), 3, pp. 1945–1950 (1999)
15. Zheng, Y., Ma, L., Zhang, L., Qian, J.: A unified particle swarm optimization scheme. In: Proceedings of the IEEE International Conference of Computational Methods in Sciences and Engineering, pp. 221–226 (2003)
16. Gao, Y., Duan, Y.: A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy. In: 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007), pp. 199–203 (2007)
17. Wu, J., He, X., Zhao, W., Wang, R.: Exponential Inertia Weight Particle Swarm Algorithm for Dynamics Optimization of Electromechanical Coupling System. In: 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, pp. 479–483 (2009)
18. Han, W., Yang, P., Ren, H., Sun, J.: Comparison Study of Several Kinds of Inertia Weights for PSO. In: 2010 IEEE International Conference on Progress in Informatics and Computing, pp. 280–284 (2010)
19. Bao, G., Mao, K.: Particle Swarm Optimization Algorithm with Asymmetric Time Varying Acceleration Coefficients. In: 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2134–2139 (2009)
20. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (Feb 2002)
21. Kennedy, J.: Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), 3, pp. 1931–1938 (1999)
22. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: simpler, maybe better. IEEE Trans. Evol. Comput. **8**(3), 204–210 (Jun 2004)
23. Liu, Q., Wei, W., Yuan, H., Zhan, Z., Li, Y.: Topology selection for particle swarm optimization, Inf. Sci. **363**, 154–173 (2016)
24. Liang, J., Qin, A., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. **10**(3), 281–295 (2006)
25. Lynn, N., Suganthan, P.N.: (2015) Heterogeneous Comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evol. Comput. **24**, 11–24 (2015)
26. Lin, A., Sun, W., Yu, H., Wu, G., Tang, H.: Global genetic learning particle swarm optimization with diversity enhancement by ring topology, Swarm Evol. Comput. **44**, 571–583 (2019)
27. Zhang, H., Yuan, M., Liang, Y., Liao, Q.: A novel particle swarm optimization based on preyCpredator relationship, Appl. Soft. Comput. **68**, 202–218 (2018)
28. Aydilek, I.B., hybrid firefly, A.: particle swarm optimization algorithm for computationally expensive numerical problems. Appl. Soft Comput. **66**, 232–249 (2018)
29. Tan, C., Chang, S., Liu, L.: Hierarchical genetic-particle swarm optimization for bistable permanent magnet actuators, Appl. Soft. Comput. **61**, 1–7 (2017)
30. Tanweer, M.R., Suresh, S., Sundararajan, N.: Self regulating particle swarm optimization algorithm, Inf. Sci. **294**, 182–202 (2015)
31. Suresh, S., Savitha, R., Sundararajan, N.: A sequential learning algorithm for complex-valued self-regulating resource allocation network-CSRAN. IEEE Trans. Neural Netw. **22**(7), 1061–1072 (2011)
32. Khan, S.U., Yang, S., Wang, L., Liu, L.: A modified particle swarm optimization algorithm for global optimizations of inverse problems. IEEE Trans. Magn. **52**(3), 1–4 (2016)
33. Andrews, P.S.: An Investigation into Mutation Operators for Particle Swarm Optimization. 2006 IEEE International Conference on Evolutionary Computation, pp. 1044–1051 (2006)
34. Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P.: Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Tech Rep. (2016)
35. Perez-Bellido, A.M., Salcedo-Sanz, S., Ortiz-Garcia, E.G., Portilla-Figueras, A.: A Hybrid Evolutionary Programming Algorithm for Spread Spectrum Radar Polyphase Codes Design. In: Gecco 2007: Genetic and Evolutionary Computation Conference, pp. 682–688 (2007)
36. Higashi, N., Iba, H.: Particle swarm optimization with Gaussian mutation. In: Proceedings of the 2003 IEEE Swarm Intelligence Symphosium, pp. 72–79 (2003)
37. Sarangi, A., Samal, S., Sarangi, S.K.: Analysis of Gaussian & Cauchy Mutations in Modified Particle Swarm Optimization Algorithm. In: 2019 5Th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 463–467 (2019)
38. Tang, J., Zhao, X.: Particle Swarm Optimization with Adaptive Mutation. In: 2009 WASE International Conference on Information Engineering, pp. 234–237 (2009)
39. Tanweer, M.R., Auditya, R., Suresh, S., Sundararajan, N., Srikanth, N.: Directionally driven self-regulating particle swarm optimization algorithm, Swarm Evol. Comput. **28**, 98–116 (2016)
40. Tanweer, M.R., Suresh, S., Sundararajan, N.: Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems. Inf. Sci. **326**, 1–24 (2016)
41. Dash, P.P., Patra, D.: Mutation based self regulating and self perception particle swarm optimization for efficient object tracking in a video. Measurement **144**, 311–327 (2019)
42. Harrison, K.R., Engelbrecht, A.P., Ombuki-Berman, B.M.: Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm. Swarm Evol. Comput. **41**, 20–35 (2018)
43. Blackwell, T.M., Bentley, P.J.: Dynamic search with charged swarms. In: Proceedings of the 4th annual conference on genetic and evolutionary computation, pp. 19-26 (2002)
44. Li, C., Yang, S., Nguyen, T.T.: A self-learning particle swarm optimizer for global optimization problems, IEEE Trans. Syst. Man Cybern. b: Cybern. 42 (3) 627C646 (2012)
45. Salajegheh, F., Salajegheh, E.: PSOG: Enhanced Particle swarm optimization by a unit vector of first and second order gradient directions. Swarm Evol Comput. **46**, 28–51 (2019)
46. Yousri, D., Allam, D., Eteiba, M.B., Suganthan, P.N.: Chaotic heterogeneous comprehensive learning particle swarm optimizer variants for permanent magnet synchronous motor models parameters estimation. Iran. J. Sci. Technol. Trans. Electr. Eng. **44**, 1299–1318 (2020)
47. Das, S., Suganthan, P.N.: Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Technical Report, Jadavpur university, Nanyang Technological University, 341–359 (2010)
48. Dereli, S., Koker, R.: Strengthening the PSO algorithm with a new technique inspired by the golf game and solving the complex engineering problem. Complex Intell Syst. **7**(3), 1515–1526 (2021)
49. Dereli, S., Koker, R.: IW-PSO Approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator. Sigma. J. Eng. Nat. Sci. **36**(1), 77–85 (2018)

50. Tang, H., Sun, W., Yu, H.: A.Lin, M. xue, Y. Song, A.novel hybrid algorithm based on PSO and FOA for target searching in unknown environments. Appl. Intell. **49**(7), 2603–2622 (2019)
51. Dereli, S., Koker, R.: A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm. Artif. Intell. Rev. **53**(2), 949–964 (2020)
52. Song, B., Wang, Z., Zou, L., An improved, P.S.O.: Algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve. Appl. Soft Comput. **106960**, 100 (2021)

Yanjie Chen received the B.S. degree in electrical engineering and its automation from Southwest Jiaotong University, Chengdu, China in 2011, the M.S. and Ph.D. degrees in control science and engineering from Hunan University, Changsha, China, in 2013 and 2017, respectively. From 2014 to 2015, he was a Research Assistant with the Department of Engineering, National University of Singapore, Singapore. From 2017 to 2021, he was an Assistant Professor with the School of Mechanical Engineering and Automation, Fuzhou University. He is currently an Associate Professor with School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, China. He is also Associate Scientist with the National Engineering Research Center of Robot Visual Perception and Control Technology, Changsha, China. His research interests include robotics, unmanned aerial manipulator, motion planning and artificial intelligence.

Jinglin Liang received the B.S. from Fuzhou University, Fuzhou, China in 2020. He is currently pursuing the M.S. degrees with School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou, China. His research interests include motion planning and mobile robot.

Yangning Wu received the B.S. degree in mechanical design manufacture and automation from Fuzhou University, Fuzhou, China, in 2019, where he is currently working toward the M.S. degree in mechatronic engineering from Fuzhou University, Fuzhou, China. His research interests include unmanned aerial robot and visual servoing.

Bingwei He received the B.S. degree in mechanical engineering from Yanshan University, China, in 1996, and the M.S. and Ph.D. degrees in mechanical engineering from Xian Jiaotong University, in 1999 and 2003, respectively. He is currently a Professor with the School of Mechanical Engineering and Automation, Fuzhou University. His research interests include intelligent mechanical and medical engineering.

Lixiong Lin received the B.S. degree in electrical engineering and automation from Huaqiao University, China, in 2008, and the M.S. degree in detection technology and the Ph.D. degree in control theory and control engineering from Xiamen University, China, in 2011 and 2016, respectively. From 2017 to 2021, he was an Assistant Professor with the School of Mechanical Engineering and Automation, Fuzhou University, China. He is currently an Associate Professor with School of Ocean Information Engineering, Jimei University, China. His research interests include fault detection, multi-sensor fusion, robotic control systems, and autonomous navigation.

Yaonan Wang received the B.S. degree in computer engineering from East China University of Science and Technology, Fuzhou, China, in 1981 and the M.S. and Ph.D. degrees in control engineering from Hunan University, Changsha, China, in 1990 and 1994, respectively. He was a Post-Doctoral Research Fellow with the National University of Defense Technology, Changsha, from 1994 to 1995, a Senior Humboldt Fellow in Germany from 1998 to 2000, and a Visiting Professor with the University of Bremen, Bremen, Germany, from 2001 to 2004. He has been a Professor with Hunan University since 1995. He has been an academician of China Engineering Academy since 2019. His research interests include robot control, intelligent control and information processing, industrial process control, and image processing.