Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04510-4

ORIGINAL ARTICLE

Spherical search optimizer: a simple yet efficient meta-heuristic approach

Jie Zhao[3] · Deyu Tang[1,2,4] · Zhen Liu[1,4] · Yongming Cai[1,4] · Shoubin Dong[2]

Received: 16 March 2019 / Accepted: 21 September 2019

Abstract
In these years, more meta-heuristic approaches have been proposed inspired by nature. However, the search mode has not been researched deeply. In this paper, we find that search style and individual selection mechanism for interaction are the core problems for a meta-heuristic algorithm. In particular, we focus on search style and have studied the principle of basic hypercube search style and basic reduced hypercube search style. Inspired by them, we propose a spherical search style. Furthermore, we design a spherical search optimizer by the spherical search style and tournament selection method. And then, theoretical analysis of it is provided. To validate the performance of the proposed method, we compare our approach against nine state-of-the-art algorithms. The CEC2013, CEC2014, CEC2015 and CEC2017 suites and the data clustering optimization problem in the real world are used. Experimental results and analysis verify that it is a simple yet efficient method to solve continuous optimization problems.

Keywords Meta-heuristic approach · Hypercube search style · Spherical search style · Data clustering

1 Introduction

Most of scientific, engineering and industrial problems can be formulated as a corresponding continuous or discrete optimization problem. It includes the knapsack problem, resource allocation problem, dimensionality reduction, network optimization, cell formation and feature selection problem [1]. Actually, such kinds of problems are NP-hard from the time complexity perspective, so that there are no exact algorithms to solve them in polynomial time budget. On this basis, using exact methods is impractical especially where the dimensionality of the problem at hand is high. Hence, meta-heuristic algorithms, which have a significant potential as general problem solvers, to solve such kinds of problems have been receiving increasing attention in recent years [2]. Meta-heuristic approaches (MHAs) are often used to find a global optimum solution of some complicated real-world optimization problem. It can be continuous or discontinuous, concave or convex, neither concave nor convex, differentiable or non-differentiable, or even their mathematical expressions not known. More MHAs focus on the characteristics of a population (such as the interaction of the individuals and operators) to solve the optimization problem without considering the

✉ Deyu Tang
tangdeyu@gdpu.edu.cn

Jie Zhao
zhaojie@gdut.edu.cn

Zhen Liu
liuzhen@gdpu.edu.cn

Yongming Cai
cym@gdpu.edu.cn

Shoubin Dong
sbdong@scut.edu.cn

1 School of Medical Information and Engineering, Guangdong Pharmaceutical University, Guangzhou 510006, People's Republic of China

2 School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, People's Republic of China

3 Department of Information Management Engineering, School of Management, Guangdong University of Technology, Guangzhou 510520, People's Republic of China

4 Guangdong Province Precise Medicine Big Data of Traditional Chinese Medicine Engineering Technology Research Center, Guangzhou 510006, People's Republic of China

Published online: 16 October 2019

characteristics of problems. In other words, an optimization problem can be considered a black box, which minimizes the complication of problems. Over the past few decades, MHAs have achieved great success and can be divided into three categories: swarm intelligence (SI), evolutionary algorithm (EA) and physical phenomena algorithm (PHA).

Swarm intelligence algorithms are inspired by the collective behaviors of natural species such as insects, animals, microorganisms and humans. The typical SIs have been developed, such as particle swarm optimization (PSO) [3], ant colony optimization (ACO) [4–6], shuffled frog leaping algorithm (SFLA) [7] and artificial bee colony (ABC) algorithm [8]. Algorithms such as cuckoo search (CS) algorithm [9], biogeography-based optimization (BBO) algorithm [10] and teaching–learning-based optimization (TLBO) [11] are proposed. Evolutionary algorithms are inspired by the Darwinian principles of nature's capability to evolve living beings well adapted to their environment. The key is to imitate the individual's evolution through mutation, selection, crossover operation, thus resulting in a better solution. Typical evolutionary algorithms are genetic algorithms [12], evolution strategies [13] and evolutionary programming [14]. Fused the mutation mechanism of PSO, differential evolution (DE) and its improved versions are proposed and have been used in many fields for continuous optimization problems. It includes the differential evolution (DE) [15], adaptive differential evolution (jDE, SaDE and JADE) [16–18], etc. Physical phenomena algorithms mimic physical rules due to some physical phenomenon. In contrast to SIs or EAs, each individual in population moves and communicates in search space according to physical rules. Typical PPAs are gravitational search algorithm (GSA) [19], big bang–big crunch (BB-BC) [20], ray optimization (RO) [21] algorithm and small-world optimization algorithm (SWOA) [22], etc.

In recent years, more nature-inspired meta-heuristic algorithms have been developed by the different operators and interactions among the individuals. Sine–cosine algorithm (SCA) was first proposed by Mirjalili et al. [23]. The two operators were achieved by the sine search style and cosine search style, and two individuals are interacted in each operator [23]. Wu et al. proposed an across neighborhood search (ANS) method. The center-based operator was realized by a Gaussian distribution function, and an individual was interacted with the superior individual or the random individual [24]. Inspired by the social hierarchy and hunting behavior of gray wolves in nature, a gray wolf optimizer (GWO) was proposed. A superior individual-based operator like the ANS was achieved in GWO, and an individual can search guided by the superior individual [25]. Kaveh et al. proposed a thermal exchange optimization (TEO) algorithm inspired by the heat transfer mechanism. An individual (cooling objective) can be guided by the better individual (environment objective), and interaction is completed by the cooling group and the environment groups [26]. Inspired by black hole phenomenon, a black hole algorithm was proposed. A global best guided operator was achieved, and an individual can be guided by the global individual [27]. Uymaz et al. proposed an artificial algae algorithm (AAA) inspired by the algae growth mechanism. A spiral search operator was achieved in AAA, and an individual (an algae) was guided by the better individual (the better algae) [28]. Inspired by the physical phenomenon, a lightning attachment procedure optimization (LAPO) algorithm was proposed by Foroughi Nematollahi et al. It simulates the lightning attachment process, and two center-based operators were realized by the opposite directions and an individual interacts with the center of population [29]. Mirjalili et al. proposed an ant lion optimizer (ALO), which mimics the hunting mechanism of ant lion in nature. In ALO, the operator was achieved by the interaction of individuals between different subpopulations (ant subpopulation and ant lion subpopulation) [30]. Inspired by the growth of trees in forest, forest optimization algorithm (FOA) was proposed by Ghaemi et al. Two dimension selection-based operations were performed according to the age of a tree [31]. Mirjalili et al. proposed the whale optimization algorithm (WOA) [32], which was inspired by the social behavior of humpback whales. In WOA, the two operations are performed like GWO [25] and SCA [23]. Inspired by the behavior of spotted hyenas, spotted hyena optimizer (SHO) [33] was proposed by Dhiman et al. The two operators of SHO were similar to those of GWO [25]. Saremi et al. proposed grasshopper optimization algorithm (GOA) [34], which mimics the behavior of grasshopper swarms in nature. The main operator was achieved by the interactions of all the individuals among a population. Inspired by the swarming behavior of salp when navigating and foraging in oceans, salp swarm algorithm (SSA) was proposed by Mirjalili et al. [35]. In SSA, historical information-based operation was performed for each individual. Tang et al. proposed the invasive tumor growth optimization (ITGO), in which multi-subpopulation and multi-operators were achieved for search of each individual [36].

As mentioned above, a nature-inspired meta-heuristic approach is based on a basic mode. Details are shown in Fig. 1. To design a new algorithm, three main components are necessary. First, an object of seeking (foods, survival, etc.) in nature should be mapped to the object of optimization problem (best fitness value). Second, 'individual set' in nature should be mapped to the 'solution sets' for the object of optimization problem. These steps are easy to be achieved. Third, 'the behaviors set' should be mapped to 'the search operator set' for optimization. And 'the learning

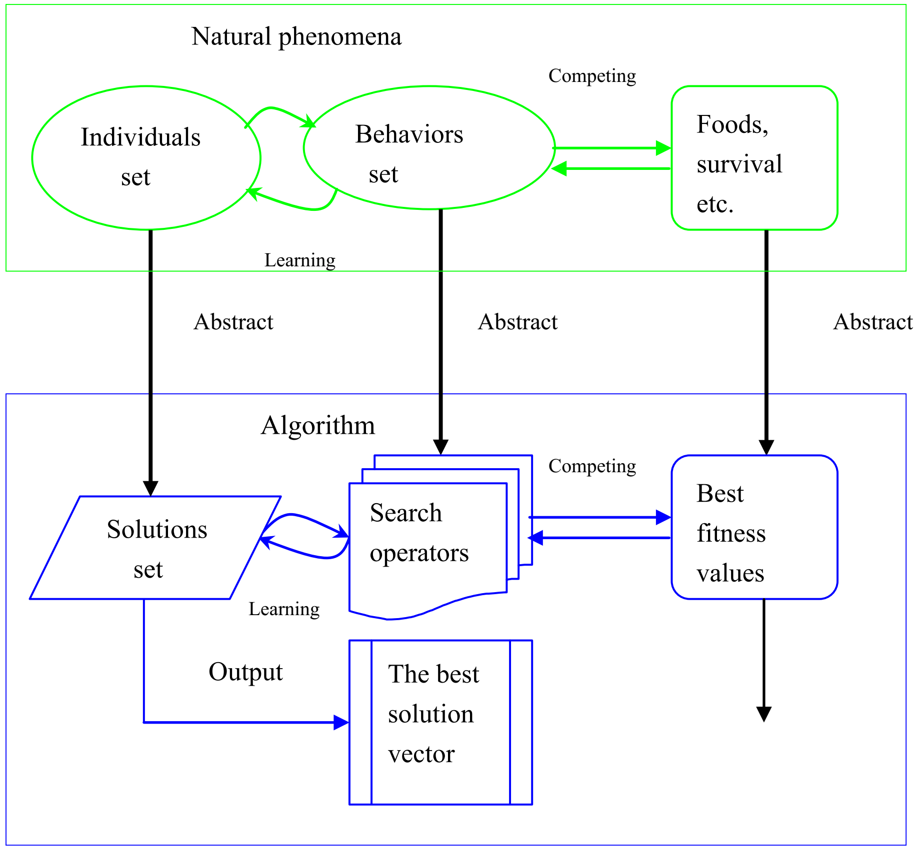


Fig. 1 Nature-inspired meta-heuristic mode

between individuals' should be mapped to 'the learning between different solutions.' The 'competing for the object of seeking' should be mapped to 'the competing for object of optimization problem (best fitness).' We know that the third step is the most important component for mathematical modeling. Figure 1 shows that search operators, learning and competing are the core problems. First, how to select one or more individuals for one or more operators by the learning and competing strategy? In other words, how to select individuals to achieve the interactions between different individuals? Second, how to design a search style for a search operator? Owning to the two problems, we attempt to design a new meta-heuristic approach to solve continuous optimization problems.

In this paper, we propose a spherical search style contrast to the basic search styles of other approaches. More importantly, we propose a spherical search optimizer (SSO) based on spherical search style. Theoretical analysis is provided. Experimental results validate that it is a simple yet efficient meta-heuristics approach. The rest part of the paper is organized as follows. In Sect. 2, related works are reviewed and discussed. Spherical search optimizer is proposed in Sect. 3. In Sect. 4, the analysis of SSO is represented. Experimental results and analysis are shown in Sect. 5. Finally, conclusions and further discussions are given in Sect. 6.

2 Related works

2.1 Basic hypercube search style

Search style is important for a basic unit of an operator in a nature-inspired meta-heuristic algorithm. The typical search style is the hypercube search style. It can be represented as Eq. (1).

$$A_i^{\text{new}} = \sum_{k=1}^{D} A_{i,k} + \text{rand.}(B_{j,k} - A_{i,k}) \tag{1}$$

where A_i and B_j denote the two solutions corresponding to the two individuals in the population. D denotes the dimension, and rand $\sim$ (0,1) means the random number of uniform distribution. Generally speaking, B_j can be considered as an 'attractor,' which has better fitness value. Equation (1) can be visualized in the three-dimensional space, when D = 3. Figure 2a shows an instance for the search process of individual A in the three-dimensional space. The two blue lines denote the two individuals (two vectors) A and B. According to Eq. (1), individual A searches in a cube region by a diagonal line with vertex A and B. A can walk to arbitrary position in this cube region using the uniform distribution. The red line in Fig. 2a gives a possible search trajectory and it is a broken line. It can be

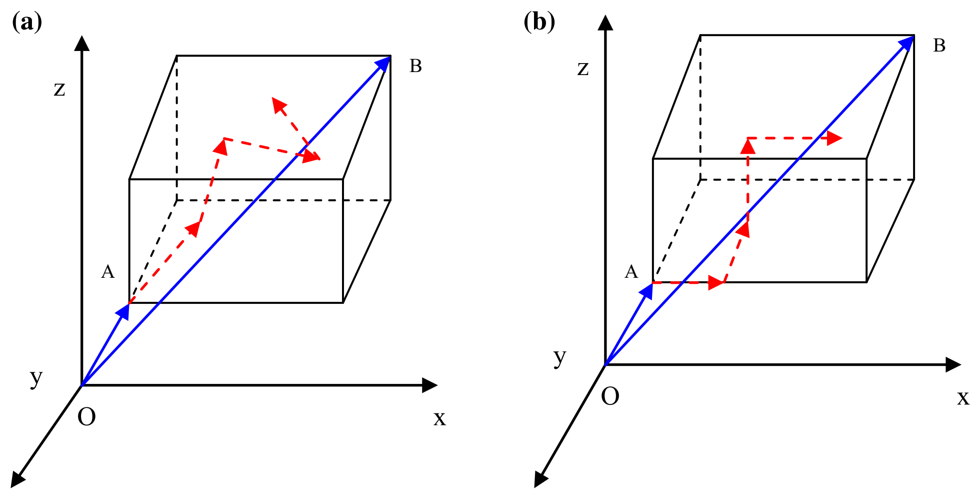


Fig. 2 **a** Hypercube search style, **b** reduced hypercube search style. The A denotes the original individual, and the B denotes the 'guided individual' for the search process

observed that there is no section of the broken line, which stands perpendicularly to a plane because three dimensions of A change synchronously according to Eq. (1). It is the basic characteristic of the cube search style. The hypercube search style has been used widely for many nature-inspired meta-heuristic algorithms. The typical algorithm is the particle swarm optimization (PSO) [3]. The operator is combined by the two units (the global guided unit and the historical information guided unit) with the hypercube search style. Indeed, the improved versions of PSO are using the hypercube search styles for the operators, as given in [37–39].

2.2 Basic reduced hypercube search style

Another search style is the expanded hypercube search style, which reduces the dimension for search. It can be represented as Eq. (2).

$$A_i^{\text{new}} = \begin{cases} A_{i,k}, & k \notin M \\ A_{i,k} + \text{rand}.(B_{j,k} - A_{i,k}), & k \in M \end{cases} \tag{2}$$

where A_i and B_j denote the two solutions corresponding to the two individuals in the population. rand $\sim$ (0,1) means the random number of uniform distribution. M is a set of the selected dimension, and each element in M is the number (index) of some dimensions. Generally speaking, the dimension is selected randomly. The length of M is less than the max dimension of A_i and greater than 0. If the length of M is equal to the max dimension of $A_{i,}$ the dimension is not reduced and it is the basic hypercube search style as Eq. (1). Figure 2b shows a possible search process of the reduced hypercube search style in a three-dimensional space. In this instance, the length of M is equals to 1. The two blue lines in Fig. 2b denote the two individuals A and B, and the red line denotes the search process. In the first step, the individual A can only moves in the x-dimension. (The y-dimension and z-dimension are reduced.) The moving direction is vertical to the plane (y, z). In the second step, the individual A can only moves in the y-dimension. (The x-dimension and z-dimension is reduced.) The moving direction is vertical to the plane (x, z). The third and fourth step is similar to the first or second step. It can be observed that the individual A often moves vertically to some plane in the three-dimensional space. It indicates that individual A will often move vertically to some hyperplane in the high-dimensional space, which is the characteristic of the reduced hypercube search style. More meta-heuristic algorithms have used this search style; the typical one is the differential evolution (DE). Almost all the improved versions of DE such as [16–18] use the reduced hypercube search style. Each operator can be achieved by the two important parameters (scaling factor F and crossover rate CR). F controls the length of search, and CR controls the dimension selection. In other words, F replaces the rand in Eq. (1) and CR controls the number of M in Eq. (2). The randomly selected set M (dimension selection) increases the diversity of moving direction for the individual. In addition, another state-of-the-art algorithm, named artificial bee colony (ABC) [8], also uses the reduced hypercube search style. It is a special case of the reduced hypercube search style by one dimension selection method. In other words, the length of M is equals to 1.

2.3 Motivations of this work

Search style is a basis for operators in meta-heuristic approach, which directly influences the performance of search. An operator can be achieved by a search style and an interaction among selected individuals. Many state-of-the-art algorithms (such as PSOs, DEs and ABCs) have used mainly the hypercube search style and the reduced hypercube search style. The advantage of them is that it is easy to be achieved for search. However, the search is limited in a hypercube region by a diagonal line with vertex A and B, which reduces the diversity of the search. For instance, the individual A probably cannot obtain the better solution if the global optimum is located in the region around A. Due to this reason, the helical search style [28] and sine–cosine search style [24] were proposed. These styles use the sine function and cosine function to search. However, the searches are still limited in the hypercube region similar to the hypercube search style and the reduced hypercube search style. Therefore, we attempt to design a new search style which can aid the individual to search by a different way.

3 Spherical search optimizer

To design a nature-inspired meta-heuristic approach, operators and interactions among individuals must be considered. Almost all the operators are achieved by combining one or more basic units with some search style. In this paper, we propose a new operator based on the spherical search style. Figure 3 shows an instance of the spherical search style in three-dimensional space. A and B denote the two individuals, and the two blue lines (OA and OB) are the two vectors. The individual A can search in a spherical space by A as the center of a sphere, and the length of $|AB|$ is the radius of a sphere. It can be observed that the vector AB can search in the whole sphere if the angles and the radius are changed. Against to the basic cube search style, spherical search style can search in more wide space with different walking ways. Based on this style, the spherical search operator can be represented as Eqs. (3), (4) and (5).

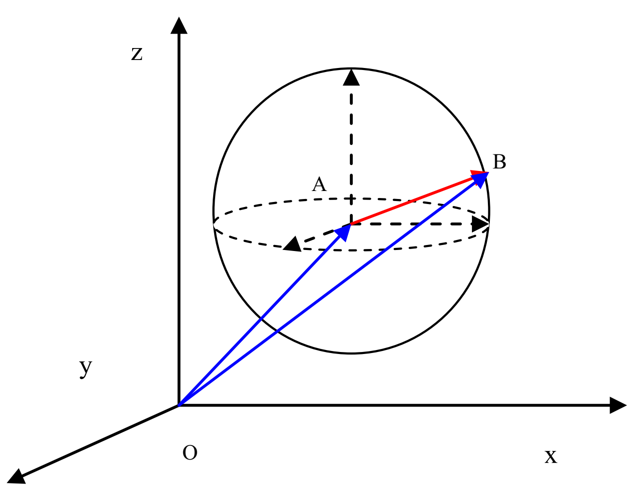


Fig. 3 Spherical search style; the A denotes the original individual, and the B denotes the 'guided individual' for the search process

$$A_i^{new} = F \cdot ||A_p - B_p||_2.\mathrm{Cos}(\theta) \quad (3)$$

$$A_j^{new} = F \cdot ||A_p - B_p||_2 \cdot \mathrm{Sin}(\theta) \cdot \mathrm{Sin}(\omega) \quad (4)$$

$$A_k^{new} = F \cdot ||A_p - B_p||_2 \cdot \mathrm{Sin}(\theta) \cdot \mathrm{Cos}(\omega) \quad (5)$$

$$A_i^{new} = F \cdot ||A_p - B_p||_2.\mathrm{Sin}(\theta) \quad (6)$$

where A and B are the two vectors corresponding to the two individuals in the population. And i, j, k ($i \neq j \neq k$) are integers randomly selected that denote the number of dimension selection corresponding to the vector A and B. p is a set of integers ($p = \{i, j, k\}$). $||A_p - B_p||_2$ denotes the Euclidean distance between A_p and B_p. F is the scaling factor and F = randn(m, s). It means a random number of normal distribution as mean value (for instance, $m = 0.5$) and standard deviation (for instance, $s = 0.03$). θ is the angle between vector A and Z axis, and ω is the angle in x–y plane. θ is a random number of uniform distribution in $[0, \pi]$, and ω is a random number of uniform distribution in $[0, 2\pi]$. In addition, the spherical search can be converted to circular search as Eqs. (3) and (6) in the two-dimensional space.

The second core problem is how to select the individuals for the interaction among them. In this paper, the interaction between individual A and individual B is the key. To choose individual B in whole population, the tournament selection method [28] is adopted. Pseudocode of the proposed spherical search optimizer is shown in Fig. 4.

To explain the search mechanism of SSO, we assume a multimodal optimization problem with two local optima and one global optimum in two-dimensional space scene as shown in Fig. 5. In Fig. 5, four green balls with vertical lines are the initial positions of four individuals. And four blue balls with horizontal lines are the positions of individuals produced by the tournament selection method. General speaking, four blue balls have better position than four green balls according to the survival of the fittest principle of tournament selection method. There are four circles (A, B, C, D), which represent the search regions of four green balls. In circle A, it can be observed that the green ball can search in a large region because the position of green ball is the circular center and the distance between the green ball and the blue ball is the radius of circular A. It is obvious that the green ball in circle A is easy to escape from the local optimum (purple region). However, the blue ball in circle A is near to the optimum in purple region. The characteristic of circular search (like the spherical search) means that it has the stronger exploration ability especially for the multimodal optimization problems.

To verify the conclusion of the analysis as above, we test a simple experiment for a multimodal optimization problem in two-dimensional space. It is the F9 Shifted Rastrigin's Function in CEC2005. We know that it is a

Algorithm SSO( )

/*FES denotes the number of fitness evaluations. The terminal condition is depended on the *Max_FES* and it is the max number of fitness evaluations. *LB, UP* denotes the search boundary. The *popsize* denotes the population size and the *Dim* denotes the dimension of the optimization problem. The *rand* represents the random number of uniform distribution *rand ~U(0,1)*. And the *normal(m, s)* denotes the random number of normal distribution, such as the mean *m=0.5*, standard deviation *s=0.09* etc.

*/

FES=0;//The number of fitness evaluation

For *i=1:popsize*

// Generates an individual(a vector) X_i and evaluate fitness values $f(X_i)$;

X_i *=LB+rand(1,Dim).*(UP-LB);*

FES=FES+1;

End For

While *FES<=Max_FES*

  For *i=1:popsize*

  // *c*, *d* are used for the tournement selection process

  Randomly generates the two different integers *c*, *d* from the set {1,2, …, *popsize*};

  If $f(X_{c,1:Dim}) < f(X_{d,1:Dim})$

   b= *c* **;**

  Else

   b= *d* **;**

  End If

// The tournement selection process is end

  If *Dim*= **=2** // for the two-dimensional optimization problems

   //*u*, *v* denotes the two integers in the set {1,2}

   $X_{i,u}^{new} = X_{i,u} + normal(0.5,0.09) * \sqrt{(X_{b,u}-X_{i,u})^2+(X_{b,v}-X_{i,v})^2} * \cos(\theta)$

   $X_{i,v}^{new} = X_{i,v} + normal(0.5,0.09) * \sqrt{(X_{b,u}-X_{i,u})^2+(X_{b,v}-X_{i,v})^2} * \sin(\theta)$

  ElseIf *Dim*>**=3** // for the high dimensional optimization problems

   Randomly generates the three different integers *u*, *v*, *r* from the set {1,2, …, *Dim*};

   $X_{i,u}^{new} = X_{i,u} + normal(0.5,0.09) * \sqrt{(X_{b,u}-X_{i,u})^2+(X_{b,v}-X_{i,v})^2+(X_{b,r}-X_{i,r})^2} * \cos(\theta)$

   $X_{i,v}^{new} = X_{i,v} + normal(0.5,0.09) * \sqrt{(X_{b,v}-X_{i,v})^2+(X_{b,v}-X_{i,v})^2+(X_{b,r}-X_{i,r})^2} * \sin(\theta) * \sin(\omega)$

   $X_{i,r}^{new} = X_{i,r} + normal(0.5,0.09) * \sqrt{(X_{b,u}-X_{i,u})^2+(X_{b,v}-X_{i,v})^2+(X_{b,r}-X_{i,r})^2} * \sin(\theta) * \cos(\omega)$

  End If

  If $f(X_{i,1:Dim}^{new}) < f(X_{i,1:Dim})$

   $X_{i,1:Dim} = X_{i,1:Dim}^{new}$;

  End If

 End For

End While

Output the best solution;

Fig. 4 Pseudocode of the proposed SSO algorithm

multimodal optimization problem with much more optima as shown in Fig. 6a. Population size is 20, and the max fitness evaluation (MAX_FES) is set as 10,000. Figure 6b–d shows the three fragments in the whole search process of SSO. Figure 6b shows the positions (red diamond) of 20 individuals, which are produced by an uniform distribution. (The initial FES value is equal to 0.) And the black diamond denotes the global optimum. SSO is run, and we take the picture (Fig. 6c) of fragment of the search process when FES = 51. It can be observed that some individuals

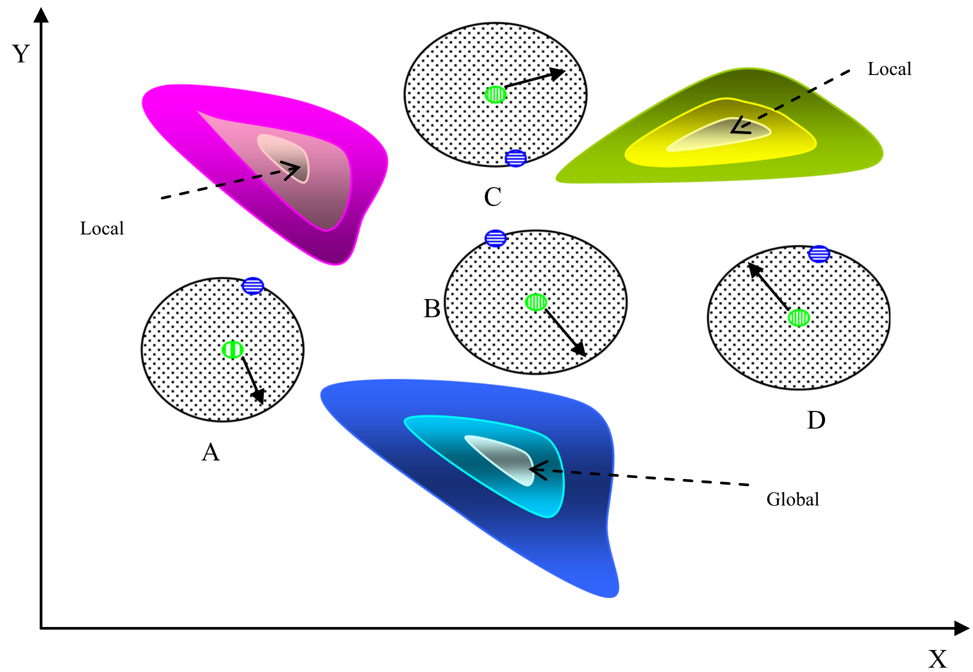


Fig. 5 Search process of SSO

have been clustered in some different optima. In Fig. 6d, SSO has obtained the global optimum ($X = -1.56440000095487$, $Y = 1.90049999172369$ and fitness $= -330$) when FES = 1537. It can be observed that the 20 individuals have clustered into the eight different local optima groups, which ensures the diversity of population. The source code of SSO can be downloaded at the Web site (https://github.com/scutdy/SSO).

4 Analysis of SSO

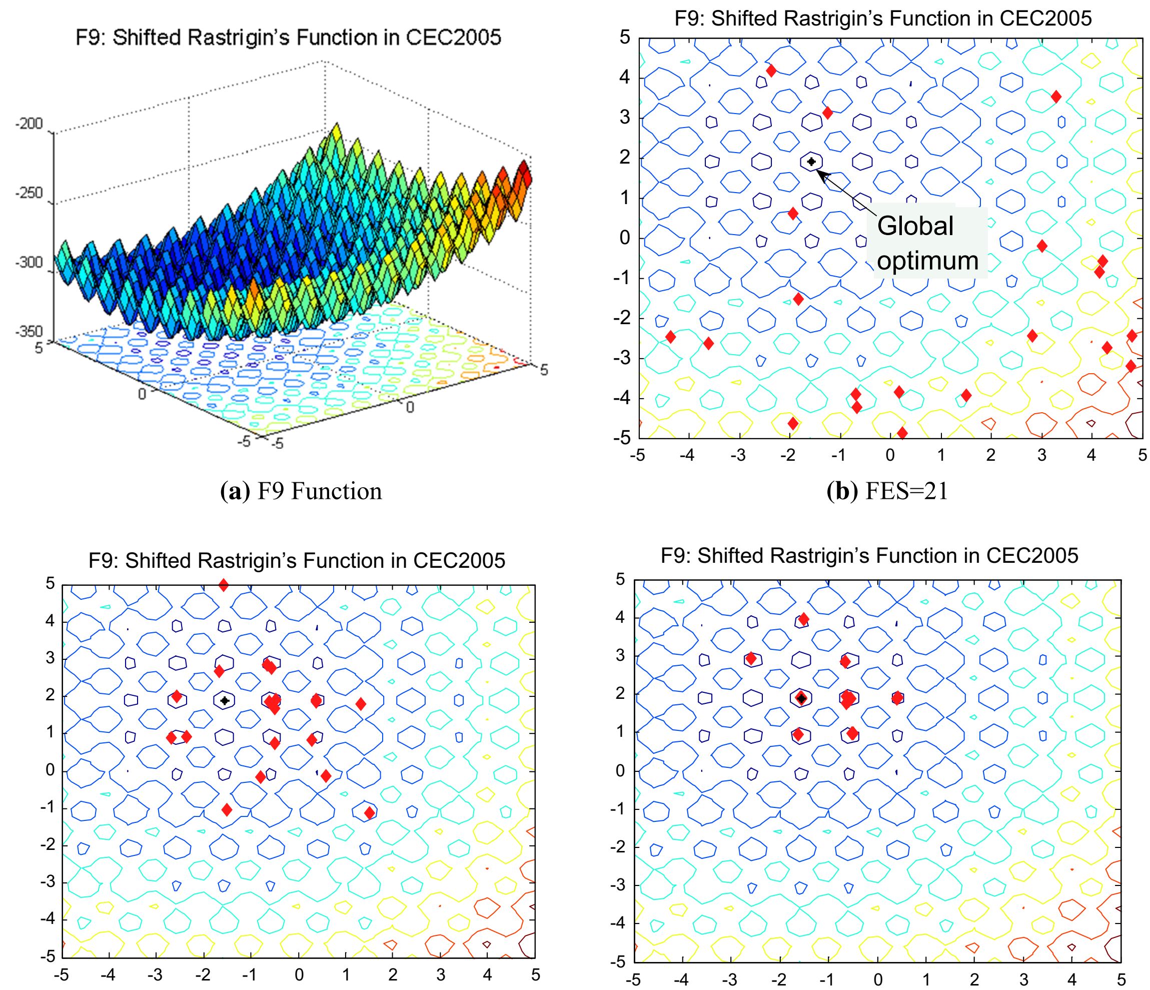
4.1 Balance between exploitation and exploration

The balance between exploitation and exploration is a core problem in meta-heuristic approach. The proposed approach obtains the good performance due to the balance between exploitation and exploration. As we known, if an individual is guided by the global best individual, it will search rapidly to a local optimum. It can increase the ability of exploitation but reduce the ability of exploration. In another case, if an individual is guided by the individual randomly chosen from the population, it can search in a wide region. It can increase the ability of exploration but reduce the ability of exploitation. Therefore, the balance between exploitation and exploration must be considered. In our method, we adopt the tournament selection method to choose an individual as an 'attractor' for guidance. The selected individual by the tournament selection method can ensure that the 'attractor' is not the only global best one but more better individuals in the population. It can achieve the balance between exploitation and exploration. In addition, the individual search by a spherical search style can also aid in the balance between exploitation and exploration. As shown in Fig. 3, we can find that individual A can search not only in the region around the 'attractor' B but also in the region around itself. Generally speaking, the local optimum is located in the region around B and more bad solutions are located in the region around A for unimodal optimization problems. However, it is not always true for the multimodal optimization problem because there are more local optima in the search space. In other words, it is possible that the global optimum is located in the region around A although there is a local optimum in the region around B. The search in the region around B can be considered as exploitation process, and the search in the region around A can be considered as exploration process. The individual A searches synchronously in the region around A and B, which achieves the balance between exploitation and exploration. In fact, we adopt the reduced hypersphere search style (dimension = 3) not the real hypersphere search style because hypersphere search style has too wide search space, which may cause the search space too sparse.

4.2 Theoretical analysis

4.2.1 Analysis of the search style

What iff a search operator by the spherical search style obtain the same position (solution) produced by the same search operator by the hypercube search style or the



(a) F9 Function **(b)** FES=21

(c) FES=513 **(d)** FES=1537

Fig. 6 An example of SSO in the two-dimensional space for F9 in CEC2005, and FES denotes the numbers of fitness evaluation

reduced hypercube search style? What if the spherical search style has a larger search space than other search styles? Take these problems, theoretical analysis is as follows:

Definition 1 Search space can be represented as:

$$S^d = S \times S \times \cdots \times S = \{(x_1, x_2, \ldots, x_d) | x_i \in [u, v], \\ u \in R, v \in R, i = 1, 2, \ldots, d\} \tag{7}$$

where d denotes the dimension, $d \geq 2$, $u < v$.

Definition 2 Suppose $\vec{a}, \vec{b} \in S^d$, individuals $\vec{a}$, $\vec{b}$ search by a hypercube search style in t_k time can be represented as

$$\vec{a}(t_k) = \vec{a}(t_{k-1}) + \vec{r} \cdot (\vec{b}(t_{k-1}) - \vec{a}(t_{k-1})) \tag{8}$$

where $\vec{r} = \{r_j | r_j \sim \text{rand}[0,1], j \in M\}$, rand[0, 1] represents an uniformly distributed random variable within the range [0,1].

$$\vec{a} = \{a_j | j \in M\} \tag{9}$$

$$\vec{b} = \{b_j | j \in M\} \tag{10}$$

$$M = \{c_i | i = 1, 2, \ldots, m, m \leq d\} \tag{11}$$

where c_i is an index (integer) randomly selected from 1 to d, and d denotes the dimension. In other words, c_i represents a discrete uniformly distributed random variable within the set $T = \{1, 2, \ldots, d\}$ and $M \subseteq T$. If m is equal to d, formula (8) represents the hypercube search style. Otherwise, if m is less to d, formula (8) represents the reduced hypercube search style.

Definition 3 Suppose $\vec{a}, \vec{b} \in S^d$, individuals $\vec{a}$, $\vec{b}$ search by a spherical search style in t_k time can be represented as

$$a_i = ||\vec{a}_p - \vec{b}_p||_2.\text{Cos}(\theta) \tag{12}$$

$$a_j = ||\vec{a}_p - \vec{b}_p||_2.\text{Sin}(\theta) \cdot \text{Sin}(\omega) \tag{13}$$

$$a_k = ||\vec{a}_p - \vec{b}_p||_2.\text{Sin}(\theta) \cdot \text{Cos}(\omega) \tag{14}$$

where θ denotes a random number of uniform distribution in $[0, \pi]$, ω is a random number of uniform distribution in $[0, 2\pi]$ and $p = \{i, j, k\}$. Variable i, j, k $(i \neq j \neq k)$ denotes three integers randomly selected from 1 to d. In other words, they are chosen by a discrete uniform distribution from 1 to d, and d denotes the dimension.

Theorem 1 *Individual $\vec{a}(t_k)$ can obtain the same position (solution) whether it uses the hypercube search style or the reduced hypercube search style* [40].

Proof Suppose an individual $\vec{a}$ is updated by a hypercube search style and obtains a position (solution) in t_k time as follows:

$$\vec{a}' = \vec{a}'_1 + \vec{a}'_2 + \cdots + \vec{a}'_i + \cdots + \vec{a}'_d \tag{15}$$

where $\vec{a}'_i = (0, 0, \ldots, 0, a'_i, 0, \ldots, 0)$; a'_i denotes a component of vector $\vec{a}'$.

$$a'_i(t_k) = a'_i(t_{k-1}) + r'_i(t_{k-1}) \cdot (b'_i(t_{k-1}) - a'_i(t_{k-1})) \tag{16}$$

In reduced hypercube search style, m $(m < d)$ components randomly selecting from individual $\vec{a}'$ can be updated in each time. It is a multi-step search method. Hence, m components of $\vec{a}'$ are chosen by a discrete uniform distribution in {1,2,…,d} and r_i represents an uniformly distributed random variable within the range [0,1]. It is obvious that all components of $\vec{a}'$ can be chosen and the same a'_i value can be obtained in a countless times sampling. That is to say, the reduced hypercube search can produce the same position $\vec{a}'$ produced by hypercube search style in a countless times sampling. □

Theorem 2 *Individual $\vec{a}(t_k)$ can obtain the same position (solution) $\vec{a}'$ whether it uses the hypercube search style or the spherical search style.*

Proof Suppose an individual $\vec{a}$ is updated by the hypercube search style and obtains a position (solution) $\vec{a}'$ in t_k time as formulas (15), (16). Then,

$$\theta^* = \arccos\left(\frac{\vec{a}'((t_k))}{||\vec{a}'_p(t_{k-1}) - \vec{b}'_p(t_{k-1})||_2}\right) \tag{17}$$

$$\omega^* = \arcsin\left(\frac{\vec{a}'(t_k)}{||\vec{a}'_p(t_{k-1}) - \vec{b}'_p(t_{k-1})||_2 \mathrm{Sin}(\theta^*)}\right) \tag{18}$$

Due to three components, i, j, k of $\vec{a}'$ are chosen randomly by a discrete uniform distribution from 1 to d and $\theta \sim \mathrm{rand}(0, \pi)$ and $\omega \sim \mathrm{rand}(0, 2\pi)$ are produced by a continuous uniform distribution. It is obvious that all components of $\vec{a}'$ can be chosen and the angles θ^* and ω^* can be obtained in a countless times sampling. Therefore, the same a'_i value in all the components of $\vec{a}'$ produced by hypercube search style can also be obtained by the spherical search style. □

Theorem 3 *The scale of each component a'_i in an individual $\vec{a}(t_k)$ by the spherical search style is larger than that of the hypercube search style.*

Proof The max length of a component a'_i in an individual $\vec{a}(t_k)$ for the hypercube search style (hc) and the spherical search style (hr) can be computed as:

$$hc = |a'_i - b'_i| \tag{19}$$

$$\begin{aligned} hr &= ||\vec{a}_p - \vec{b}_p||_2 \\ &= \sqrt{((a'_i - b'_i)^2 + (a'_j - b'_j)^2 + (a'_k - b'_k)^2)} \end{aligned} \tag{20}$$

It is obvious that

$$hr > \sqrt{((a'_i - b'_i)^2} = hc \tag{21}$$

where hc denotes the length in one-dimensional space by the hypercube search style and hr denotes the length in one-dimensional space by the spherical search style. It means that the spherical search style has larger search space than the hypercube search style.

4.2.2 Convergence analysis of SSO

The SSO algorithm modifies each individual by adjusting the two angles randomly in a spherical region. Its initial position is taken as the center of the sphere, and the Euclidean distance between it and an elite individual is taken as the radius. The elite individual is produced by a tournament selection method, and it is selected from an elite population. It is obvious that the elite population is a subpopulation from the original population. If the population traps in a local optimum set, then the elite population also traps in the local optimum set. The radius of the pair of the individual vectors from the population is equal to 0. It means that SSO cannot escape from the local optima. This is one of the reasons resulting in the fact that the SSO does not converge to the global optimum with probability 1. In turn, this section will theoretically prove that the SSO cannot converge in probability to the global optimum by Markov chain analysis method.

Definition 4 *(Convergence in Probability* [41]*).* Let $\{x(t), t = 0, 1, 2,\ldots\}$ be a population sequence generated by a population-based stochastic algorithm, the stochastic sequence $\{x(t)\}$ weakly converges in probability to the global optimum, if and only if:

$$\lim_{t \to \infty} p\{x(t) \cap B^* \neq \phi\} = 1$$

where B^* is the set of the global optima of an optimization problem. That is, the algorithm holds with convergence in probability. Otherwise, the sequence $\{x(t)\}$ or the algorithm is called no convergence in probability.

Due to the limitations of the numerical calculation accuracy in computer, we can map the continuous search space Ψ to a finite discrete set Φ.

$$\Phi^{ps} = \underbrace{\Phi \times \Phi \times \cdots \Phi}_{ps} \tag{22}$$

Let $x(t)$ denote the tth generation population of the SSO, then search process of the SSO can be described as a stochastic process $\{x(t), t = 0, 1, 2, \ldots\}$ that takes on a finite number of possible values. Unless otherwise mentioned, $p\{x(t+1) = I | x(t) = X\}$ denote the probability that the process will, when in state X at time t, next make a transition into state I.

Let M^0 and I^0 denote the mutation operator and improvement operator, respectively. Then $p\{M^0(X^m) = Y^m\}$ is the probability that X^m changes into Y^m at t generation by the mutation operator of SSO ($m = 3$, m is the number of the selected dimensions). $p\{I^0(Y^m, X^m) = I\}$ is the probability that X^m is improved by Y^m into I at t generation by the improvement operator.

$$\begin{aligned} & p\{x(t+1) = I | x(t) = X\} \\ & = \sum_{X^m, Y^m \in \Phi^d} p\{M^0(X^m) = Y^m\} \cdot p\{I^0(Y^m, X^m) = I\}\} \end{aligned} \tag{23}$$

Let Ω denote a local optima or premature solutions set. That is, Ω is a subset of the state space Φ^{ps}, and every vector (individual) of the population of Ω is equal and is not the global optimum.

Lemma 1 *Let $\{x(t), t = 0, 1, 2, \ldots\}$ be the population sequence generated by the SSO. In the SSO, if an arbitrary population $x(t)$ traps into a local optimum or premature set, it cannot escape. That is, suppose that there exists a time t_k such that the population $x(t_k) \in \Omega$, then*

$$\lim_{t \to \infty} p\{x(t) = x(t_k) | x(t_k) \in \Omega\} = 1 \tag{24}$$

where

$$\Omega = \{X = \underbrace{(\vec{b}, \vec{b}, \cdots \vec{b})}\} : \vec{b} \in \Phi, \vec{b} \notin B^*\} \tag{25}$$

Proof All the vectors of the population $x(t_k)$ are equal when the state $x(t_k) \in \Omega$ at time t_k, so the radius of the sphere by two arbitrary vectors is 0. By formulas (3)–(5), we can get that

$$p\{M^0(x^m(t_k)) = x^m(t_k)\} = 1 \tag{26}$$

So, for an arbitrary positive integer t, we can get that

$$\begin{aligned} P\{x(t+1) &= X | x(t) = X \in \Omega\} \\ &= \sum_{X^m, Y^m \in \Phi^d} p\{M^0(X^m) = Y^m\} \cdot p\{I^0(Y^m, X^m) = I\}\} \\ &= p\{M^0(X^m) = X^m\} \cdot p\{I^0(X^m, X^m) = I\}\} \\ &= 1 \end{aligned} \tag{27}$$

Hence, for a given positive integer t_k, if the population $x(t_k) \in \Omega$, then

$$\lim_{t \to \infty} p\{x(t) = x(t_k) | x(t_k) \in \Omega\} = 1$$

□

Theorem 4 *The population sequence generated by the SSO $\{x(t), t = 1, 2, \ldots\}$ cannot converge in probability to the global optima set. That is,*

$$\lim_{t \to \infty} p\{x(t) \cap B^* \neq \phi\} < 1 \tag{28}$$

Proof Let $x(0)$ denote the initial population of the SSO. The individuals of the $x(0)$ are uniformly distributed random vectors lying the discrete space. The probability generating every individual of $x(0)$ is equal. So, the probability that the initial population of the SSO traps into a local optima or premature solutions set is obviously greater than 0. That is,

$$p\{x(0) \in \Omega\} > 0 \tag{29}$$

From the Lemma 1, when the $x(0) \in \Omega$, we have

$$\lim_{t \to \infty} p\{x(t) = x(0) | x(0) \in \Omega\} = 1 \tag{30}$$

implying that

$$\lim_{t \to \infty} p\{x(t) \cap B^* = \phi | x(0) \in \Omega\} = 1 \tag{31}$$

Hence, we can get

$$\begin{aligned} \lim_{t \to \infty} p\{x(t) \cap B^* \neq \phi\} &= 1 - \lim_{t \to \infty} p\{x(t) \cap B^* = \phi\} \\ &\leq 1 - \lim_{t \to \infty} p\{x(t) \cap B^* = \phi, x(0) \in \Omega\} \\ &\leq 1 - \lim_{t \to \infty} p\{x(t) \cap B^* = \phi, x(0) \in \Omega\} \\ &\quad \cdot p\{x(0) \in \Omega\} \\ &= 1 - p\{x(0) \in \Omega\} \end{aligned} \tag{32}$$

According to formula (21),

$$1 - p\{x(0) \in \Omega\} < 1 \tag{33}$$

implying that

$$\lim_{t \to \infty} p\{x(t) \cap B^* \neq \phi\} < 1 \tag{34}$$

□

According to Definition 4, the sequence $x(t)$ generated by the SSO holds no convergence in probability. No convergence in probability implies that the limitation (if exists) of the convergent probability is less than 1. It does not imply that the algorithm must not converge to the global optimum [41]. In fact, the algorithm with no convergence in probability may hold with a high convergence ratio. However, an algorithm with convergence in probability is generally more robust than one with no convergence in probability. It means that the improved version of SSO in future can consider the improvement method for convergence in probability of it, such as enhancing the diversity of population.

4.3 Discussion about SSO and other nature-inspired evolutionary algorithms

As mentioned above, nature-inspired evolutionary algorithm establishes the mathematical model inspired by nature to solve optimization problem [42, 43] and it is also a meta-heuristic algorithm. The key point is how to understand the behavior, phenomenon or others of nature and how to build the relation between optimization problems and them. Furthermore, researchers should build the mathematical model for the search operators and the individual's selection in the population to solve the optimization problem. So far, more researchers focus on the study of the behavior, phenomenon or others of nature but ignore the study of the search style for search operators. Almost all the nature-inspired evolutionary algorithms use the hypercube search style. In contrast to them, SSO as a meta-heuristic algorithm focuses on the search style related to the search operator. Indeed, the spherical search style is proposed and it can also be used to other nature-inspired evolutionary algorithms. In other words, researchers can propose new nature-inspired evolutionary algorithm inspired by nature using the spherical search style.

5 Experimental results and analysis

To validate the performance of our approach, we adopt the 21 benchmark optimization problems from CEC2013 suite [44], CEC2014 suite [45], CEC2015 suite [46] and CEC2017 suite [47] to compare SSO against the nine state-of-the-art algorithms. It includes artificial bee colony (ABC) [8], cuckoo search (CS) algorithm [9], teaching–learning-based optimization (TLBO) [11], gray wolf optimizer (GWO) [25], artificial algae algorithm (AAA) [28], backtracking search optimization algorithm (BSA) [48], self-adaptive differential evolution (SaDE) algorithm [17], adaptive differential evolution (JADE) [18] and self-adapting control parameters in differential evolution (jDE) [16]. In addition, the proposed approach is used to solve the data clustering optimization problem. Then, we compare the SSO against the six algorithms. It includes covariance matrix adaptation evolution strategy (CMA-ES) (parameter setting given in [49]), gravitational search algorithm (GSA) [19], gray wolf optimizer (GWO) [25], teaching–learning-based optimization (TLBO) [11], big bang–big crunch (BB-BC) algorithm [20] and black hole (BH) algorithm [27].

For a fair comparison, each benchmark problem is run 30 times by a different initial population each time to avoid any negative effects of the structure of the initial population during the tests. All the algorithms for comparison set the same population size as $ps = 20$. The number of the fitness evaluations (FES) is used to compare all the algorithms, and the max fitness evaluation (MAX_FES) is set as D*1E4 recommended as CEC suites. The mean values, standard deviation and median values of error (fitness value-optimum value) are computed as a result of the test for the detailed statistical analysis. The t test [38, 39] and the Friedman's test [50] are used to statistic the comparison results. Table 1 shows the parameter setting of all the algorithms for comparison, and Table 2 shows the details of the 21 benchmark problems.

5.1 Experimental results and analysis on CEC suites

In these tests, we adopt only the two unimodal functions (F1, F2 in Table 2) from CEC2013 and nineteen multimodal functions (F3–F21) from CEC suites. The unimodal functions are used to test the local search ability of the proposed approach, and multimodal functions are used to test the global search ability. The nineteen multimodal functions with different characteristics are rotated, shifted or non-separable, which are difficult to be solved for many methods. In fact, the real-world optimization problems are more complex to be solved. Only one benchmark CEC suite (such as CEC2013 or other) can only simulate small scenes for the real-world optimization problem. Then, more improved or expanded versions of CEC are proposed. Therefore, we choose more hybrid functions and composition functions from the different CEC suites to better simulate the real-world optimization problems. We only use the 21 functions due to the space limited in this paper.

In addition, almost all algorithms are sensitive to the dimension of the optimization functions. The performance of a meta-heuristic algorithm will be declined due to more optima when the dimension of an optimization problem is increased. The experiments are arranged as the two groups with the low dimensions (Dim = 30) and high dimensions (Dim = 50 and 100). The experimental results are given in Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17.

Table 1 Parameter setting

No.	Algorithm	Parameter setting
1.	ABC	SN = 10, limit = SN * 2 * dimension
2.	CS	Beta = 1.5, pa = 0.25
3.	TLBO	–
4.	GWO	–
5.	AAA	k = 2, Le = 0.3, ap = 0.5
6.	BSA	F = 3
7.	SaDE	–
8.	jDE	$t1$ = $t2$ = 0.1, F = 0.5, CR = 0.9
9.	JADE	c = 1/10, p = 0.05, CRm = 0.5, Fm = 0.5
10.	SSO	s = 0.03

1. Experiment on the low dimension (D = 30)

In Table 3, the letter 'R' in the first line means the rank of different algorithms. Table 3 presents the multimodal optimization problems. The SSO obtains the best performance against ABC, CS, TLBO, GWO and AAA for Shifted and Rotated Griewank's Function (F4), Rotated Weierstrass Function (F5), Hybrid Function 6 (N = 4) (F12), Composition Function 2 (F19) and Composition Function 7 (N = 6) (F21). Table 4 also shows the comparison results of SSO and BSA, SaDE, jDE and JADE. It can be observed that SSO obtains the best performance against BSA, SaDE, jDE and JADE for the eight functions including Rotated Ackley's Function (F3), Shifted and Rotated HGBat Function (F7), Hybrid Function 2 (N = 4) (F9), Composition Function 7 (F14), Composition Function 2 (F15), Composition Function 5 (F17), Composition Function 2 (F19) and Composition Function 5 (F20). For the unimodal functions, only SaDE and CS obtain the global optimum for Sphere Function (F1) and Different Power Function (F2). It means that they have stronger local search ability. In other words, they search quickly to find the local optimum. However, they are weak to solve the multimodal problems. The comparison results of SaDE and CS for multimodal problems (F3–F21) confirm our deduction (Tables 3, 4). Although SSO is not the best one for the unimodal optimization problems, it can solve more multimodal optimization problems better. It illustrates that SSO achieves the balance between exploitation and exploration better. Table 5 shows the average rank test and

Table 2 CEC suites

Func	CEC number	Function name	Range	Optimum
F1	CEC2013_1	Sphere Function	[– 100, 100]	– 1400
F2	CEC2013_5	Different Power Function	[– 100, 100]	– 1000
F3	CEC2013_8	Rotated Ackley's Function	[– 100, 100]	– 700
F4	CEC2014_7	Shifted and Rotated Griewank's Function	[– 100, 100]	700
F5	CEC2013_9	Rotated Weierstrass Function	[– 100, 100]	– 600
F6	CEC2017_6	Shifted and Rotated Expanded Scaffer's F6 Function	[– 100, 100]	600
F7	CEC2014_14	Shifted and Rotated HGBat Function	[– 100, 100]	1400
F8	CEC2015_7	Shifted and Rotated HGBat Function, HGBat Function	[– 100, 100]	700
F9	CEC2015_11	Hybrid Function 2 (N = 4), Griewank's Function, Weierstrass Function, Rosenbrock's Function	[– 100, 100]	1100
F10	CEC2015_12	Hybrid Function 3 (N = 5), Katsuura Function, HappyCat Function, Griewank's Function, Rosenbrock's Function, Schwefel's Function, Ackley's Function 1200	[– 100, 100]	1200
F11	CEC2017_11	Hybrid Function 1 (N = 3)	[– 100, 100]	1100
F12	CEC2017_16	Hybrid Function 6 (N = 4)	[– 100, 100]	1600
F13	CEC2013_26	Composition Function 6 (n = 5, rotated) 1200	[– 100, 100]	1200
F14	CEC2013_27	Composition Function 7 (n = 5, rotated)	[– 100, 100]	1300
F15	CEC2014_24	Composition Function 2 (N = 3)	[– 100, 100]	2400
F16	CEC2014_25	Composition Function 3 (N = 3)	[– 100, 100]	2500
F17	CEC2014_27	Composition Function 5 (N = 5)	[– 100, 100]	2700
F18	CEC2014_28	Composition Function 6 (N = 5)	[– 100, 100]	2800
F19	CEC2015_14	Composition Function 2 (N = 3), Schwefel's Function, Rastrigin's Function, High Conditioned Elliptic Function	[– 100, 100]	1400
F20	CEC2017_25	Composition Function 5 (N = 5)	[– 100, 100]	2500
F21	CEC2017_27	Composition Function 7 (N = 6)	[– 100, 100]	2700

Table 3 Comparison results of SSO and ABC, CS, TLBO, GWO, AAA (D = 30); the letter 'R' in the first line denotes the rank of different algorithms, and each algorithm is run 30 times for each problem

Func	ABC	*R*	CS	*R*	TLBO	*R*	GWO	*R*	AAA	*R*	SSO	*R*
F1 Median	4.5475e−013	**4**	**0.0000e000**	**1**	1.2960e−011	**5**	6.5157e+002	**6**	2.2737e−013	**2**	4.5475e−013	**3**
Means	4.3959e−013		**0.0000e000**		2.1433e−010		6.9381e+002		2.8043e−013		4.3201e−013	
Std	5.7687e−014		**0.0000e000**		5.9234e−010		5.7328e+002		1.1460e−013		6.9378e−014	
F2 Median	7.9581e−013	**3**	**0.0000e000**	**1**	2.3192e−011	**5**	3.2511e+002	**6**	3.4106e−013	**2**	2.0486e−010	**4**
Means	7.2002e−013		**4.9264e−014**		2.5080e−009		3.7197e+002		3.1074e−013		2.0182e−010	
Std	9.1210e−014		5.7299e−014		1.2838e−008		2.3467e+002		**5.1134e−014**		6.3684e−011	
F3 Median	2.0976e+001	**6**	**2.0954e+001**	**3**	2.0955e+001	**4**	2.0968e+001	**1**	2.0958e+001	**5**	2.0960e+001	**2**
Means	2.0961e+001		2.0946e+001		2.0948e+001		**2.0941e+001**		2.0958e+001		2.0942e+001	
Std	**4.6334e−002**		6.7855e−002		5.1169e−002		6.4080e−002		7.1068e−002		5.0463e−002	
F4 Median	4.7703e−010	**3**	2.0464e−012	**4**	4.1881e−002	**5**	6.2819e+000	**6**	3.4106e−013	**2**	**5.6843e−013**	**1**
Means	1.4571e−003		3.1340e−003		1.7970e+000		1.1916e+001		3.2858e−004		**1.2472e−010**	
Std	6.7930e−003		6.6759e−003		7.0523e+000		1.4402e+001		1.7997e−003		**6.1375e−010**	
F5 Median	3.1164e+001	**5**	3.0770e+001	**4**	3.1280e+001	**6**	2.5752e+001	**3**	2.8004e+001	**2**	**2.7378e+001**	**1**
Means	3.0719e+001		3.0131e+001		3.1257e+001		3.0068e+001		2.7249e+001		**2.6938e+001**	
Std	2.1181e+000		2.8310e+000		3.1265e+000		9.4421e+000		3.0424e+000		**2.2168e+000**	
F6 Median	**0.0000e000**	**1**	1.8166e+001	**5**	2.2256e+001	**6**	5.0838e+000	**4**	3.4106e−013	**2**	1.6492e−003	**3**
Means	**1.1369e−014**		1.9451e+001		2.3385e+001		5.7904e+000		4.5627e−009		3.2703e−003	
Std	**3.4689e−014**		1.2875e+001		7.5548e+000		2.2344e+000		2.4990e−008		4.2778e−003	
F7 Median	**1.9429e−001**	**1**	2.5762e−001	**3**	2.8651e−001	**4**	8.2144e−001	**6**	3.0231e−001	**5**	2.5207e−001	**2**
Means	**1.9542e−001**		2.5298e−001		3.0515e−001		7.7142e−001		3.0636e−001		2.4789e−001	
Std	**1.5406e−002**		3.5345e−002		9.6705e−002		6.7212e−001		4.8630e−002		3.7072e−002	
F8 Median	7.5282e+000	**2**	1.0185e+001	**4**	1.1599e+001	**5**	1.9144e+001	**6**	**6.9656e+000**	**1**	8.5454e+000	**3**
Means	7.5854e+000		1.0024e+001		1.2933e+001		1.9780e+001		**6.9291e+000**		8.7662e+000	
Std	1.3732e+000		1.4268e+000		4.9219e+000		4.6030e+000		**1.2366e+000**		1.2735e+000	
F9 Median	3.2231e+002	**1**	**3.0662e+002**	**3**	1.0373e+003	**6**	7.9290e+002	**5**	3.1373e+002	**4**	3.2521e+002	**2**
Means	**3.2433e+002**		3.4973e+002		8.8550e+002		7.8044e+002		4.6838e+002		3.2665e+002	
Std	1.2771e+001		1.6413e+002		3.5040e+002		6.6245e+001		2.3385e+002		**1.1650e+001**	
F10 Median	**1.0619e+002**	**1**	1.0741e+002	**3**	1.0963e+002	**6**	1.0741e+002	**5**	1.0734e+002	**2**	1.0784e+002	**4**
Means	**1.0614e+002**		1.0762e+002		1.0968e+002		1.0803e+002		1.0737e+002		1.0781e+002	
Std	**3.6729e−001**		7.7213e−001		2.3350e+000		2.5105e+000		9.5487e−001		4.6854e−001	
F11 Median	1.7539e+002	**6**	6.1895e+001	**1**	1.2671e+002	**4**	1.7564e+002	**5**	**5.5856e+001**	**2**	7.4516e+001	**3**
Means	2.8447e+002		**6.7988e+001**		1.4044e+002		1.8158e+002		6.8828e+001		7.8403e+001	
Std	3.3901e+002		**2.3525e+001**		6.2471e+001		3.9130e+001		3.3190e+001		2.7020e+001	
F12 Median	7.6368e+002	**2**	8.4273e+002	**4**	8.2323e+002	**3**	7.7562e+002	**5**	8.6814e+002	**6**	**6.9116e+002**	**1**
Means	7.5754e+002		8.0540e+002		8.0051e+002		8.1780e+002		8.5008e+002		**6.7204e+002**	
Std	1.9504e+002		1.6326e+002		2.4907e+002		3.4683e+002		2.0985e+002		**1.3125e+002**	
F13 Median	2.0046e+002	**2**	**2.0015e+002**	**1**	2.0014e+002	**6**	2.0167e+002	**5**	2.0009e+002	**4**	2.0187e+002	**3**
Means	2.0052e+002		**2.0016e+002**		2.7588e+002		2.6286e+002		2.0583e+002		2.0188e+002	
Std	1.8156e−001		**7.9934e−002**		8.8398e+001		8.3453e+001		3.1415e+001		4.9226e−001	
F14 Median	**4.0001e+002**	**1**	1.1394e+003	**6**	1.0532e+003	**4**	9.7927e+002	**3**	1.0610e+003	**5**	4.8603e+002	**2**
Means	**4.7444e+002**		1.1248e+003		1.0553e+003		1.0082e+003		1.0570e+003		6.2436e+002	
Std	**2.2631e+002**		5.5251e+001		7.6661e+001		2.1133e+002		7.9121e+001		2.7358e+002	
F15 Median	2.2762e+002	**6**	2.2531e+002	**4**	**2.0000e+002**	**1**	2.0001e+002	**2**	2.2648e+002	**5**	2.2645e+002	**3**

Table 3 (continued)

Func	ABC	*R*	CS	*R*	TLBO	*R*	GWO	*R*	AAA	*R*	SSO	*R*
Means	2.2788e+002		2.2530e+002		**2.0000e+002**		2.0001e+002		2.2720e+002		2.2502e+002	
Std	1.1833e+000		1.1715e+000		**6.0892e−004**		2.6845e−003		3.1202e+000		5.4651e+000	
F16 Median	2.0745e+002	**5**	2.0464e+002	**2**	**2.0000e+002**	**1**	2.1001e+002	**6**	2.0595e+002	**3**	2.0669e+002	**4**
Means	2.0780e+002		2.0472e+002		**2.0058e+002**		2.1031e+002		2.0631e+002		2.0656e+002	
Std	1.5420e+000		1.3177e+000		**1.8122e+000**		2.6590e+000		2.1655e+000		9.5899e−001	
F17 Median	4.0868e+002	**2**	4.0322e+002	**1**	9.0915e+002	**6**	6.7410e+002	**5**	**4.0311e+002**	**4**	4.1048e+002	**3**
Means	4.0936e+002		**4.0450e+002**		8.0046e+002		6.7704e+002		4.6014e+002		4.1029e+002	
Std	4.2832e+000		4.4900e+000		2.2397e+002		6.9965e+001		1.3398e+002		**3.6486e+000**	
F18 Median	1.0358e+003	**5**	1.0013e+003	**3**	1.6318e+003	**6**	9.9035e+002	**4**	9.1104e+002	**2**	**8.9983e+002**	**2**
Means	1.0496e+003		1.0300e+003		1.6152e+003		1.0387e+003		9.1039e+002		**8.9381e+002**	
Std	1.1964e+002		9.3502e+001		3.7670e+002		1.3404e+002		3.9612e+001		**4.0186e+001**	
F19 Median	3.1420e+004	**2**	3.3001e+004	**3**	3.6359e+004	**6**	3.5413e+004	**5**	3.2985e+004	**4**	**3.1310e+004**	**1**
Means	3.1445e+004		3.1544e+004		3.5726e+004		3.5467e+004		3.2793e+004		**3.1361e+004**	
Std	2.5434e+002		6.0158e+003		2.3631e+003		1.4764e+003		1.0645e+003		**2.2459e+002**	
F20 Median	**3.8381e+002**	**1**	3.8688e+002	**3**	4.1162e+002	**5**	4.3929e+002	**6**	3.8712e+002	**4**	3.8576e+002	**2**
Means	**3.8397e+002**		3.8588e+002		4.1140e+002		4.4091e+002		3.8726e+002		3.8571e+002	
Std	**5.8736e−001**		1.5736e+000		2.2655e+001		1.7335e+001		1.4189e+000		1.3931e+000	
F21 Median	5.1625e+002	**2**	5.1936e+002	**4**	5.6996e+002	**6**	5.2638e+002	**5**	5.1841e+002	**3**	**5.1034e+002**	**1**
Means	5.1489e+002		5.2086e+002		5.6880e+002		5.2938e+002		5.1537e+002		**5.0820e+002**	
Std	6.8498e+000		8.1950e+000		3.0365e+001		1.4286e+001		1.0976e+001		**7.2901e+000**	

statistical values by the Friedman (p value = 0.05). It confirms that SSO is the winner for the 21 problems against other state-of-the-art algorithms.

The t test method is used to compare the difference between SSO and the other nine algorithms for the 21 optimization problems. A two-tailed test with significance level of 0.05 is adopted, and the p value and t value are recorded and shown in Table 6. The results of SSO are highlighted in bold when it performs the better performance. The 'B' and 'W' refer to that SSO is significantly better/significantly worse than other algorithms, respectively. Table 6 shows that the average, excellent and good rate between SSO and other algorithms for the 21 functions in terms of the t test is 49.73% (Eq. (35). And the average performance of SSO is good, although it is not always the best for all functions with respect to the nine algorithms.

$$\text{total} = \sum_{i=1}^{9}\sum_{j=1}^{21}(B_{ij})/(21*9) \tag{35}$$

2. Experiment on the high dimension (D = 50 and D = 100)

To test the performance of SSO for high-dimensional optimization problems, the dimension is increased from 30 to 50. All the algorithms are run in the same parameter setting as above. Mean values, standard deviation and median values and the rank are given in Tables 7 and 8. Table 7 shows that SSO obtains the best performance against ABC, CS, TLBO, GWO and AAA for eight functions (F3, F4, F5, F9, F12, F17, F19 and F21). Compared to Table 3, it is not difficult to find that the number of SSO as winner (R = 1) is increased from 5 to 8 when the dimension is increased from 30 to 50. In addition, SSO becomes the winner for Rotated Ackley's Function (F3), Hybrid Function 2 (N = 4) (F9) and Composition Function 5 (N = 5) (F17) when the dimension is increased from 30 to 50. Table 8 shows that SSO obtains the best performance against BSA, SaDE, JADE and jDE for 11 functions (F3, F4, F7, F8, F9, F13, F15, F16, F17, F19 and F20). Compared to Table 4, the number of SSO as winner (R = 1) is increased from 8 to 11 when the dimension is increased from 30 to 50. Table 9 shows the average rank test and statistical values by the Friedman. It confirms that SSO is the winner for the 21 problems against other state-of-the-art algorithms. This test shows that SSO is robust in high-

Table 4 Comparison results of SSO and BSA, SaDE, JADE, jDE algorithm (D = 30); the letter 'R' in the first line denotes the rank of different algorithms, and each algorithm is run 30 times for each problem

Func	BSA	***R***	SaDE	***R***	JADE	***R***	jDE	***R***	SSO	***R***
F1 Median	2.2737e−013	**3**	**0.0000e000**	**1**	2.2737e−013	**2**	2.2737e−013	**5**	4.5475e−013	**4**
Means	2.4253e−013		**0.0000e000**		1.2885e−013		5.6086e−013		4.3201e−013	
Std	5.7687e−014		**0.0000e000**		1.1460e−013		1.9198e−012		6.9378e−014	
F2 Median	3.4106e−013	**3**	**0.0000e000**	**1**	1.1369e−013	**2**	6.8212e−013	**4**	2.0486e−010	**5**
Means	3.1453e−013		**0.0000e000**		1.2127e−013		3.6077e−012		2.0182e−010	
Std	1.3578e−013		**0.0000e000**		2.8843e−014		9.8873e−012		6.3684e−011	
F3 Median	2.0966e+001	**5**	**2.0940e+001**	**1**	2.0951e+001	**2**	2.0960e+001	**4**	2.0960e+001	**3**
Means	2.0965e+001		**2.0932e+001**		2.0937e+001		2.0949e+001		2.0942e+001	
Std	4.2410e−002		**5.0600e−002**		5.0188e−002		4.7155e−002		5.0463e−002	
F4 Median	**3.4106e−013**	**2**	9.8573e−003	**5**	2.2737e−013	**3**	3960e−003	**4**	5.6843e−013	**1**
Means	2.4603e−003		3.1983e−002		6.1558e−003		2.7994e−002		**1.2472e−010**	
Std	8.5170e−003		5.3861e−002		8.9196e−003		5.5292e−002		**6.1375e−010**	
F5 Median	2.8508e+001	**4**	**2.3935e+001**	**1**	2.9583e+001	**5**	2.4156e+001	**2**	2.7378e+001	**3**
Means	2.8220e+001		**2.4128e+001**		2.9589e+001		2.4371e+001		2.6938e+001	
Std	2.5131e+000		**3.4528e+000**		1.8907e+000		3.8056e+000		2.2168e+000	
F6 Median	**2.2737e−013**	**1**	3.2639e−002	**5**	1.1369e−013	**3**	1.0718e−004	**4**	1.6492e−003	**2**
Means	**2.3116e−013**		7.4918e−002		4.5401e−003		1.6047e−002		3.2703e−003	
Std	**5.5722e−014**		1.5260e−001		1.6122e−002		5.5511e−002		4.2778e−003	
F7 Median	**2.4570e−001**	**2**	2.5553e−001	**3**	2.6522e−001	**5**	2.7956e−001	**4**	2.5207e−001	**1**
Means	2.5530e−001		2.5955e−001		2.8781e−001		2.7698e−001		**2.4789e−001**	
Std	4.2005e−002		4.0982e−002		1.0595e−001		4.9406e−002		**3.7072e−002**	
F8 Median	7.6701e+000	**3**	7.5339e+000	**2**	8.5602e+000	**5**	**6.7720e+000**	**1**	8.5454e+000	**4**
Means	7.6504e+000		7.2552e+000		8.6353e+000		**6.5088e+000**		8.7662e+000	
Std	1.1514e+000		1.6240e+000		1.2501e+000		1.7111e+000		1.2735e+000	
F9 Median	**3.0649e+002**	**2**	6.5435e+002	**4**	6.1256e+002	**5**	5.6218e+002	**3**	3.2521e+002	**1**
Means	4.7430e+002		5.4164e+002		5.7046e+002		4.9809e+002		**3.2665e+002**	
Std	1.9044e+002		2.2615e+002		1.6564e+002		1.4388e+002		**1.1650e+001**	
F10 Median	1.0652e+002	**2**	1.0667e+002	**3**	1.0654e+002	**3**	**1.0620e+002**	**1**	1.0784e+002	**4**
Means	1.0650e+002		1.0666e+002		1.0666e+002		**1.0626e+002**		1.0781e+002	
Std	5.8201e−001		1.1571e+000		1.1711e+000		**9.5475e−001**		4.6854e−001	
F11 Median	**2.3587e+001**	**1**	9.5607e+001	**4**	1.0646e+002	**5**	4.1808e+001	**2**	7.4516e+001	**3**
Means	**2.9390e+001**		1.0893e+002		1.1202e+002		5.0202e+001		7.8403e+001	
Std	**2.0852e+001**		4.6373e+001		3.8283e+001		3.1611e+001		2.7020e+001	
F12 Median	6.1768e+002	**4**	5.7260e+002	**3**	5.5313e+002	**2**	**5.0322e+002**	**1**	6.9116e+002	**5**
Means	5.7126e+002		5.3967e+002		5.0761e+002		**4.5421e+002**		6.7204e+002	
Std	1.7004e+002		2.2584e+002		2.3344e+002		2.0267e+002		**1.3125e+002**	
F13 Median	**2.0005e+002**	**1**	2.0006e+002	**4**	2.0018e+002	**5**	2.0001e+002	**3**	2.0187e+002	**2**
Means	**2.0005e+002**		2.5450e+002		2.5626e+002		2.4503e+002		2.0188e+002	
Std	**1.8886e−002**		7.2916e+001		7.5682e+001		7.0077e+001		4.9226e−001	
F14 Median	1.0062e+003	**5**	8.3441e+002	**2**	9.8645e+002	**4**	8.4296e+002	**3**	**4.8603e+002**	**1**
Means	9.9379e+002		8.3446e+002		9.5304e+002		8.4475e+002		**6.2436e+002**	
Std	1.2241e+002		8.6952e+001		1.3312e+002		9.6565e+001		**2.7358e+002**	
F15 Median	2.2646e+002	**2**	2.3651e+002	**3**	2.4032e+002	**4**	2.4011e+002	**5**	**2.2645e+002**	**1**

Table 4 (continued)

Func	BSA	*R*	SaDE	*R*	JADE	*R*	jDE	*R*	SSO	*R*
Means	2.2653e+002		2.3580e+002		2.3742e+002		2.3865e+002		**2.2502e+002**	
Std	2.4764e+000		7.1775e+000		7.4156e+000		7.2718e+000		**5.4651e+000**	
F16 Median	**2.0491e+002**	**1**	2.1345e+002	**5**	2.1103e+002	**4**	2.0564e+002	**3**	2.0669e+002	**2**
Means	**2.0522e+002**		2.1365e+002		2.1102e+002		2.0660e+002		2.0656e+002	
Std	**2.2514e+000**		2.7141e+000		4.2808e+000		3.5436e+000		9.5899e−001	
F17 Median	**4.0191e+002**	**2**	5.8165e+002	**5**	5.1870e+002	**4**	4.2790e+002	**3**	4.1048e+002	**1**
Means	4.1713e+002		5.7826e+002		5.0874e+002		4.5399e+002		**4.1029e+002**	
Std	5.7989e+001		1.2083e+002		8.5217e+001		8.5797e+001		**3.6486e+000**	
F18 Median	**8.5275e+002**	**1**	9.7217e+002	**5**	8.8230e+002	**4**	8.8167e+002	**2**	8.9983e+002	**3**
Means	**8.5152e+002**		9.6852e+002		9.4776e+002		8.8683e+002		8.9381e+002	
Std	**2.3585e+001**		7.7254e+001		2.5261e+002		8.6417e+001		4.0186e+001	
F19 Median	3.3315e+004	**2**	3.4197e+004	**5**	3.3001e+004	**3**	3.3470e+004	**4**	**3.1310e+004**	**1**
Means	3.3057e+004		3.3812e+004		3.3218e+004		3.3220e+004		**3.1361e+004**	
Std	8.4716e+002		1.2637e+003		1.6035e+003		9.5478e+002		**2.2459e+002**	
F20 Median	3.8727e+002	**2**	3.8956e+002	**5**	3.8759e+002	**3**	3.8897e+002	**4**	**3.8576e+002**	**1**
Means	3.8717e+002		4.0335e+002		3.8802e+002		3.9396e+002		**3.8571e+002**	
Std	1.3483e+000		2.0349e+001		2.3307e+000		1.3302e+001		**1.3931e+000**	
F21 Median	**5.0662e+002**	**1**	5.3635e+002	**5**	5.1691e+002	**3**	5.2063e+002	**4**	5.1034e+002	**2**
Means	**5.0710e+002**		5.4095e+002		5.1604e+002		5.2537e+002		5.0820e+002	
Std	**7.6042e+000**		1.8572e+001		1.4901e+001		1.6123e+001		7.2901e+000	

Table 5 Friedman test for SSO and other algorithms (D = 30)

Order	Algorithm	Average ranks	Statistical value	*p* value
1	**SSO**	**3.81**	52.35	3.86E−8
2	BSA	3.90		
3	ABC	4.43		
4	CS	4.79		
5	AAA	5.14		
6	jDE	5.19		
7	SaDE	5.48		
8	JADE	5.69		
9	GWO	8.29		
9	TLBO	8.29		

dimensional optimization problems. In addition, we compare the performance of SSO against other algorithms when the dimension is increased to 100. For unimodal optimization problems, it can be observed that the performance of some meta-heuristic algorithms such as CS and SaDE has been degraded drastically with the increase in the problem scale. For instance, CS and SaDE obtain the global optimum for Sphere Function (F1) when the dimension is 30 as shown in Table 3. However, the performance of CS and SaDE has been degraded drastically when the dimension is increased to 100 as shown in Tables 10 and 11. By comparison, the means value and media values of SSO have a little change when the dimension is increased from 30 to 100. For multimodal optimization problems, the number of SSO as the winner

Table 6 *T* test for comparison

Func	ABC	CS	TLBO	GWO	AAA	BSA	SaDE	JADE	jDE
F1 *p* value	6.6236e−001	5.8829e−025	5.7512e−002	**2.8840e−007**	1.5045e−006	8.3298e−013	5.8829e−025	6.4065e−014	7.1518e−001
t value	4.4117e−001	−3.4106e+001	1.9779e+000	**6.6287e+000**	−6.0208e+000	−1.2042e+001	−3.4106e+001	−1.3359e+001	3.6849e−001
F2 *p* value	7.9556e−017	7.3979e−017	3.3300e−001	**1.4711e−009**	7.6463e−017	7.5715e−017	7.3532e−017	7.4609e−017	9.4182e−017
t value	−1.7306e+001	−1.7354e+001	9.8454e−001	**8.6818e+000**	−1.7332e+001	−1.7338e+001	−1.7358e+001	−1.7348e+001	−1.7196e+001
F3 *p* value	2.0772e−001	8.2333e−001	6.6846e−001	9.3879e−001	3.4539e−001	**6.4156e−002**	5.2254e−001	6.4716e−001	5.6707e−001
t value	1.2886e+000	2.2530e−001	4.3266e−001	−7.7460e−002	9.5919e−001	**1.9245e+000**	−6.4729e−001	−4.6252e−001	5.7899e−001
F4 *p* value	2.4960e−001	**1.5525e−002**	1.7341e−001	**9.3040e−005**	3.2558e−001	1.2445e−001	**2.9015e−003**	**7.2439e−004**	**9.6047e−003**
t value	1.1749e+000	**2.5713e+000**	1.3957e+000	**4.5316e+000**	1.0000e+000	1.5822e+000	**3.2524e+000**	**3.7801e+000**	**2.7731e+000**
F5 *p* value	**1.6649e−007**	**8.0819e−005**	**7.0701e−007**	9.9656e−002	6.5230e−001	5.2530e−002	**3.6016e−004**	**1.1272e−005**	**4.8162e−003**
t value	**6.8338e+000**	**4.5825e+000**	**6.2973e+000**	1.7009e+000	4.5528e−001	2.0216e+000	**−4.0387e+000**	**5.2919e+000**	**−3.0530e+000**
F6 *p* value	2.4005e−004	**4.0375e−009**	**1.3711e−016**	**1.3819e−014**	2.4006e−004	**2.4005e−004**	1.5686e−002	6.8756e−001	2.2051e−001
t value	−4.1873e+000	**8.2728e+000**	**1.6954e+001**	**1.4195e+001**	−4.1873e+000	**−4.1873e+000**	2.5669e+000	4.0622e−001	1.2522e+000
F7 *p* value	4.3802e−008	5.7977e−001	**4.1465e−003**	**1.6688e−004**	**5.9597e−005**	4.9489e−001	1.7716e−001	6.9773e−002	**1.5460e−002**
t value	−7.3394e+000	5.6001e−001	**3.1124e+000**	**4.3198e+000**	**4.6924e+000**	6.9128e−001	1.3832e+000	1.8830e+000	**2.5731e+000**
F8 *p* value	2.2931e−003	**6.3346e−004**	**6.3520e−005**	**6.2798e−014**	1.6877e−006	2.7743e−003	5.0040e−004	7.4469e−001	5.6339e−006
t value	−3.3436e+000	**3.8300e+000**	**4.6694e+000**	**1.3370e+001**	−5.9789e+000	−3.2699e+000	−3.9174e+000	−3.2878e−001	−5.5419e+000
F9 *p* value	4.0771e−001	4.5154e−001	**1.0627e−009**	**1.9272e−026**	**2.3310e−003**	**1.9454e−004**	**1.5932e−005**	**4.8461e−009**	**2.5719e−007**
t value	−8.4013e−001	7.6315e−001	**8.8155e+000**	**3.8470e+001**	**3.3373e+000**	**4.2640e+000**	**5.1673e+000**	**8.1998e+000**	**6.6713e+000**
F10 *p* value	2.4597e−016	2.6530e−001	**2.8979e−004**	6.4738e−001	3.0510e−002	6.7428e−012	2.3600e−005	9.9108e−005	2.7947e−008
t value	−1.6582e+001	−1.1359e+000	**4.1184e+000**	4.6221e−001	−2.2746e+000	−1.1035e+001	−5.0260e+000	−4.5088e+000	−7.5122e+000
F11 *p* value	**2.8124e−003**	1.0354e−001	**1.2904e−005**	**2.2089e−012**	2.0409e−001	2.7701e−010	**5.7960e−003**	**5.6590e−004**	1.2123e−003
t value	**3.2645e+000**	−1.6808e+000	**5.2432e+000**	**1.1565e+001**	−1.2993e+000	−9.3788e+000	**2.9789e+000**	**3.8719e+000**	−3.5869e+000
F12 *p* value	6.6267e−002	**2.0046e−003**	**8.2118e−003**	**3.7384e−002**	**4.0892e−004**	1.6799e−002	2.3893e−003	6.6900e−004	3.7902e−005
t value	1.9085e+000	**3.3953e+000**	**2.8376e+000**	**2.1817e+000**	**3.9919e+000**	−2.5375e+000	−3.3277e+000	−3.8097e+000	−4.8555e+000
F13 *p* value	2.3958e−014	4.5650e−018	**8.0114e−005**	**3.9703e−004**	4.9806e−001	1.1027e−018	**4.4333e−004**	**4.7184e−004**	**2.1318e−003**
t value	−1.3891e+001	−1.9258e+001	**4.5857e+000**	**4.0028e+000**	6.8616e−001	−2.0296e+001	**3.9621e+000**	**3.9391e+000**	**3.3717e+000**
F14 *p* value	2.5472e−002	1.5527e−010	**4.7494e−009**	**2.2565e−006**	**3.0813e−009**	**2.0690e−007**	**5.3719e−004**	**1.7735e−006**	**3.6271e−004**
t value	−2.3555e+000	9.6269e+000	**8.2078e+000**	**5.8732e+000**	**8.3814e+000**	**6.7525e+000**	**3.8912e+000**	**5.9608e+000**	**4.0361e+000**
F15 *p* value	**6.8085e−003**	7.7051e−001	3.3287e−021	3.3584e−021	5.3088e−002	2.7301e−001	**2.2753e−007**	**7.1058e−008**	**9.9501e−009**
t value	**2.9139e+000**	2.9445e−001	−2.5068e+001	−2.5060e+001	2.0165e+000	1.1174e+000	**6.7170e+000**	**7.1549e+000**	**7.9146e+000**
F16 *p* value	**1.5967e−003**	2.3774e−007	1.1352e−015	**4.3208e−008**	5.6587e−001	1.5835e−003	**2.6595e−013**	**1.1183e−005**	9.4909e−001
t value	**3.4824e+000**	−6.7006e+000	−1.5641e+001	**7.3447e+000**	−5.8080e−001	−3.4855e+000	**1.2616e+001**	**5.2947e+000**	6.4404e−002
F17 *p* value	2.9855e−001	3.0262e−006	**1.7333e−010**	**5.0995e−019**	5.1311e−002	5.2986e−001	**2.1385e−008**	**5.5428e−007**	**9.2077e−003**
t value	−1.0585e+000	−5.7667e+000	**9.5795e+000**	**2.0879e+001**	2.0329e+000	6.3586e−001	**7.6157e+000**	**6.3869e+000**	**2.7905e+000**
F18 *p* value	**2.9445e−008**	**4.6344e−009**	**2.3683e−011**	**1.2257e−006**	1.1551e−001	2.3521e−005	**4.5332e−005**	2.5669e−001	7.0485e−001
t value	**7.4920e+000**	**8.2176e+000**	**1.0458e+001**	**6.0956e+000**	1.6225e+000	−5.0272e+000	**4.7910e+000**	1.1571e+000	−3.8255e−001

Table 6 (continued)

Func	ABC	CS	TLBO	GWO	AAA	BSA	SaDE	JADE	jDE
F19 *p* value	2.0660e−001	8.6791e−001	**7.5668e−011**	**3.2158e−015**	**3.9212e−008**	**5.0656e−012**	**4.4853e−011**	**9.3849e−007**	**1.0532e−011**
t value	1.2919e+000	1.6779e−001	**9.9398e+000**	**1.5025e+001**	**7.3819e+000**	**1.1170e+001**	**1.0171e+001**	**6.1933e+000**	**1.0828e+001**
F20 *p* value	8.3475e−007	6.1518e−001	**9.7095e−007**	**8.3669e−017**	**4.2775e−004**	**2.1932e−004**	**6.1782e−005**	**1.9862e−005**	**1.4715e−003**
t value	−6.2363e+000	5.0817e−001	**6.1809e+000**	**1.7273e+001**	**3.9753e+000**	**4.2203e+000**	**4.6795e+000**	**5.0880e+000**	**3.5134e+000**
F21 *p* value	**3.2642e−003**	**1.4193e−006**	**2.0565e−011**	**8.0045e−009**	**8.7741e−003**	5.4279e−001	**7.6732e−010**	**1.0246e−002**	**6.8714e−006**
t value	**3.2064e+000**	**6.0421e+000**	**1.0522e+001**	**8.0004e+000**	**2.8104e+000**	−6.1587e−001	**8.9503e+000**	**2.7463e+000**	**5.4702e+000**

(R = 1) is increased from five functions (F4, F5, F 12, F19 and F21) to seven functions (F5, F9, F12, F17, F18, F19, F21) when the dimension is increased from 30 to 100 for ABC, CS, TLBO, GWO and AAA. For BSA, SaDE, JADE and jDE, the number of SSO as the winner (R = 1) is the same as eight different functions (F4, F9, F13, F15, F16, F19, F20 and F21) when the dimension is increased from 30 to 100. Although BSA, SaDE, JADE and jDE have better stable performances against ABC, CS, TLBO, GWO and AAA, the performance of them cannot outperform SSO in more optimization problems. Table 12 shows the average ranks of the 10 algorithms and the statistical values by the Friedman test. It is obvious that the performance of SSO outperforms that of the other nine algorithms significantly.

Then, we compare the convergence speed of SSO against other methods. Figure 7 shows the convergence cure (dimension size = 30, 50 and 100). It can be observed that SSO runs with the faster convergence speed than other nine methods for Shifted and Rotated Griewank's Function (F4) when the dimension is 30. For Composition Function 2 (N = 3) (F19), SSO has almost the same convergent speed when the dimension is 30. However, SSO obtains the better accuracy than other methods for F19. It also can be observed that SSO runs with the faster convergence speed than other nine methods for Hybrid Function 2 (N = 4) (F9) and Composition Function 7 (N = 6) (F21) when the dimension is 50. The dimension is increased from 50 to 100 for Hybrid Function 2 (N = 4) (F9), and SSO can also obtain better accuracy and convergent speed than other nine functions. For Composition Function 7 (N = 6) (F21), AAA runs with the faster convergence speed than SSO in the early stage (FES <= 100,000). However, SSO converges faster than AAA after the early stage (FES > 100,000).

5.2 Parameter tuning

In our proposed approach, scaling factor is achieved by the normal distribution with mean value (m) as 0.5 and standard deviation (s) should be tuned. The object of scaling factor for SSO is to change the length of the radius in order to search in the whole sphere. The mean value must be set as 0.5 because it can ensure that the search can be initiated from the center of the radius. Parameter s controls the scaling rate of radius. The larger s makes the search out from the sphere, which is not better for convergence. Therefore, we set the search range as [0.01, 0.1] for parameter s. To tune the parameter s, we choose five multimodal functions (F6, F11, F16, F25 and F27) from CEC2017. Other parameter setting such as population size for this test is same as in Sect. 5.1. Experimental results are given in Table 13, and the bold fonts mean the winners. It

Table 7 Comparison results of SSO and ABC, CS, TLBO, GWO, AAA (D = 50); the letter 'R' in the first line denotes the rank of different algorithms, and each algorithm is run 30 times for each problem

Func	ABC	***R***	CS	***R***	TLBO	***R***	GWO	***R***	AAA	***R***	SSO	***R***
F1 Median	9.0949e−013	**4**	**0.0000e+000**	**1**	3.9131e−010	**5**	2.1735e+003	**6**	6.8212e−013	**2**	9.0949e−013	**3**
Means	8.8676e−013		**0.0000e+000**		1.9716e−008		2.2121e+003		6.2907e−013		8.0339e−013	
Std	9.1536e−014		**0.0000e+000**		7.4194e−008		8.5120e+002		1.2922e−013		1.1537e−013	
F2 Median	1.3642e−012	**2**	**1.1369e−013**	**1**	4.0518e−010	**5**	4.2802e+002	**6**	5.6843e−013	**4**	2.2521e−010	**3**
Means	1.3908e−012		**1.1369e−013**		5.2948e−009		4.6530e+002		5.4570e−013		2.2731e−010	
Std	1.3246e−013		**0.0000e+000**		1.1199e−008		2.5597e+002		9.1536e−014		5.8756e−011	
F3 Median	2.1147e+001	**4**	2.1134e+001	**2**	2.1136e+001	**3**	**2.1131e+001**	**1**	2.1170e+001	**5**	2.1137e+001	**1**
Means	2.1140e+001		2.1129e+001		2.1130e+001		**2.1126e+001**		2.1156e+001		**2.1126e+001**	
Std	3.4076e−002		3.2811e−002		3.3258e−002		**3.5133e−002**		5.2121e−002		4.7476e−002	
F4 Median	2.3065e−005	**3**	**1.1369e−013**	**4**	4.1719e−002	**5**	3.2496e+001	**6**	1.1369e−012	**2**	1.0805e−008	**1**
Means	2.3993e−003		4.9954e−003		2.3888e−001		3.6822e+001		7.3960e−004		**4.9419e−007**	
Std	4.7021e−003		1.1399e−002		6.5981e−001		2.4198e+001		2.2567e−003		**2.5368e−006**	
F5 Median	6.0546e+001	**4**	6.0364e+001	**3**	6.0642e+001	**5**	7.2684e+001	**6**	5.7550e+001	**2**	**5.4002e+001**	**1**
Means	6.0012e+001		5.9807e+001		6.0093e+001		6.4579e+001		5.6905e+001		**5.3366e+001**	
Std	2.7242e+000		2.5236e+000		4.3592e+000		1.4351e+001**2**		4.3403e+000		**2.9335e+000**	
F6 Median	**0.0000e+000**	**1**	3.6867e+001	**5**	4.1715e+001	**6**	1.0957e+001	**4**	5.6843e−013	**2**	1.0767e−002	**3**
Means	**7.5791e−015**		3.8207e+001		3.9920e+001		1.0887e+001		5.3433e−013		1.0861e−002	
Std	**2.8843e−014**		1.0462e+001		7.2507e+000		2.4219e+000		7.4040e−014		5.9965e−003	
F7 Median	**2.4416e−001**	**1**	3.0115e−001	**3**	3.1614e−001	**5**	1.1242e+000	**6**	3.2326e−001	**4**	2.9639e−001	**2**
Means	**2.4557e−001**		2.9519e−001		3.4905e−001		4.5346e+000		3.2659e−001		2.8768e−001	
Std	**2.1183e−002**		3.4254e−002		1.6898e−001		6.8726e+000		4.2095e−002		2.6285e−002	
F8 Median	**1.6089e+001**	**1**	1.6143e+001	**3**	3.0968e+001	**5**	6.9480e+001	**6**	1.9398e+001	**4**	1.8699e+001	**2**
Means	**1.8116e+001**		2.9346e+001		4.3056e+001		7.7588e+001		3.0295e+001		2.2345e+001	
Std	**8.9538e+000**		1.8628e+001		2.6477e+001		1.7909e+001		1.7575e+001		1.0594e+001	
F9 Median	3.2090e+002	**2**	3.5560e+002	**3**	1.7923e+003	**6**	1.1786e+003	**5**	1.2645e+003	**4**	**3.4413e+002**	**1**
Means	5.3763e+002		7.6342e+002		1.7634e+003		1.1735e+003		1.0020e+003		**4.1316e+002**	
Std	3.8862e+002		5.5836e+002		1.0235e+002		8.6093e+001		4.6601e+002		**2.1819e+002**	
F10 Median	**1.0848e+002**	**1**	1.1171e+002	**4**	1.1227e+002	**6**	1.1259e+002	5	1.0975e+002	**2**	1.1082e+002	**3**
Means	**1.0850e+002**		1.1186e+002		1.1284e+002		1.1273e+002		1.0990e+002		1.1066e+002	
Std	**4.0498e−001**		1.4007e+000		2.1508e+000		2.4044e+000		8.6674e−001		7.1334e−001	
F11 Median	9.0580e+002	**6**	1.8324e+002	**2**	2.1422e+002	**3**	3.9428e+002	**4**	**1.3569e+002**	**1**	3.7023e+002	**5**
Means	1.1183e+003		1.8550e+002		2.3337e+002		4.3073e+002		**1.4203e+002**		5.1088e+002	
Std	7.9895e+002		3.8017e+001		7.1353e+001		1.1593e+002		**4.4047e+001**		3.4683e+002	
F12 Median	1.6538e+003	**5**	1.6396e+003	**2**	1.6705e+003	**4**	1.4814e+003	**6**	1.6212e+003	**3**	**1.4440e+003**	**1**
Means	1.6491e+003		1.5966e+003		1.6215e+003		1.7232e+003		1.6120e+003		**1.3760e+003**	
Std	3.4032e+002		2.3952e+002		4.2926e+002		6.6976e+002		2.9095e+002		**2.0086e+002**	
F13 Median	2.0119e+002	**2**	2.0090e+002	**4**	4.4800e+002	**5**	4.1600e+002	**6**	**2.0020e+002**	**1**	2.0557e+002	**3**
Means	2.0123e+002		2.0935e+002		4.1533e+002		4.1564e+002		**2.0021e+002**		2.0582e+002	
Std	2.9961e−001		4.6139e+001		8.6269e+001		6.7325e+001		**6.6987e−002**		1.5892e+000	
F14 Median	2.0076e+003	**3**	1.9085e+003	**6**	1.8906e+003	**5**	**1.4547e+003**	**1**	1.8867e+003	**4**	1.8330e+003	**2**
Means	1.7279e+003		1.9035e+003		1.8882e+003		**1.6475e+003**		1.8720e+003		1.6893e+003	
Std	6.1137e+002		6.6357e+001		9.6322e+001		**3.7576e+002**		1.2244e+002		4.1657e+002	
F15 Median	2.6426e+002	**2**	2.6950e+002	**4**	2.**0000e+002**	**1**	**2.0001e+002**	**1**	2.6828e+002	**3**	2.6799e+002	**5**

Table 7 (continued)

Func	ABC	*R*	CS	*R*	TLBO	*R*	GWO	*R*	AAA	*R*	SSO	*R*
Means	2.6382e+002		2.6748e+002		**2.0000e+002**		**2.0001e+002**		2.6723e+002		2.6691e+002	
Std	4.5764e+000		7.1114e+000		**2.6962e**−004		**2.5940e−003**		3.6739e+000		3.7057e+000	
F16 Median	2.1519e+002	**5**	2.1503e+002	**4**	**2.0000e+002**	**1**	2.2503e+002	**6**	2.1156e+002	**2**	2.1279e+002	**3**
Means	2.1527e+002		2.1452e+002		**2.0000e+002**		2.2504e+002		2.1249e+002		2.1293e+002	
Std	1.6981e+000		4.1955e+000		**2.3017e−0**13		4.9733e+000		4.1704e+000		1.2290e+000	
F17 Median	1.2227e+003	**3**	1.4246e+003	**5**	1.6864e+003	**6**	1.1283e+003	**2**	1.2076e+003	**4**	**5.1777e+002**	**1**
Means	1.1253e+003		1.2291e+003		1.6840e+003		1.1081e+003		1.1744e+003		**7.0296e+002**	
Std	2.7084e+002		4.5354e+002		8.4909e+001		9.2182e+001		2.3186e+002		**3.2077e+002**	
F18 Median	1.9349e+003	**5**	1.5649e+003	**3**	3.0189e+003	**6**	1.5837e+003	**4**	**1.3926e+003**	**1**	1.4378e+003	**2**
Means	1.9967e+003		1.6559e+003		2.9991e+003		1.6572e+003		**1.4218e+003**		1.4482e+003	
Std	3.7142e+002		3.5379e+002		7.9820e+002		1.9263e+002		1.3719e+002		**9.6766e+001**	
F19 Median	4.9533e+004	**2**	5.9205e+004	**3**	7.3277e+004	**5**	7.5510e+004	**6**	5.9201e+004	**4**	4.9537e+004	**1**
Means	5.0192e+004		5.8927e+004		6.9338e+004		7.2463e+004		5.8996e+004		**4.9663e+004**	
Std	2.4500e+003		7.4290e+003		1.1701e+004		5.8326e+003		6.9809e+003		**3.0839e+002**	
F20 Median	**4.8566e+002**	**1**	5.3011e+002	**4**	5.7396e+002	**5**	7.9627e+002	**6**	5.2732e+002	**3**	5.2525e+002	**2**
Means	**4.9069e+002**		5.3696e+002		5.6306e+002		7.7156e+002		5.2155e+002		5.1969e+002	
Std	**2.7760e+001**		2.7575e+001		3.5671e+001		9.5922e+001		4.0402e+001		1.8342e+001	
F21 Median	7.1300e+002	**3**	7.3667e+002	**4**	1.0409e+003	**6**	7.2567e+002	**5**	6.5818e+002	**2**	**6.2315e+002**	**1**
Means	7.0465e+002		7.1810e+002		1.0491e+003		7.3868e+002		6.5047e+002		**6.2315e+002**	
Std	4.9285e+001		9.4877e+001		1.4550e+002		6.2228e+001		4.7057e+001		**3.3899e+001**	

can be observed that SSO can obtain the best performance for Shifted and Rotated Expanded Scaffer's F6 Function (CEC2017_6) and Hybrid Function 6 ($N = 4$) (CEC2017_16) when the parameter is set as $s = 0.04$. SSO obtains the best performance for CEC2017_11, CEC2017_25 and CEC2017_27 when the parameter is set as $s = 0.02$, $s = 0.07$ and $s = 0.09$. Table 14 shows the average ranks of SSO with the different parameter values by the Friedman test. It means that the parameter s could be set as $s = 0.09$ for many optimization problems. Although the difference is not significant according to the result given in Table 14, the recommended parameter setting of s can be considered for other optimization problems.

5.3 Experiment on the real-world optimization problems

Data clustering is an important and popular tool in data mining techniques. It has been widely used in many fields such as document clustering, costumer analysis, pattern recognition and image segmentation. Clustering can be considered as a classification process without any supervision, in which a set of data objects is divided into some different clusters. In particular, a data object in a cluster must have the great similarity and the data object of different clusters must have high dissimilarity.

Table 8 Comparison results of SSO and BSA, SaDE, JADE, jDE algorithm (D = 50); the letter 'R' in the first line denotes the rank of different algorithms, and each algorithm is run 30 times for each problem

Func	BSA	***R***	SaDE	***R***	JADE	***R***	jDE	***R***	SSO	***R***
F1 Median	6.8212e−013	**3**	**0.0000e+000**	**1**	2.2737e−013	**2**	9.0949e−013	**5**	9.0949e−013	**4**
Means	7.3517e−013		**0.0000e+000**		3.8654e−013		6.7833e−012		8.0339e−013	
Std	1.2922e−013		**0.0000e+000**		3.2239e−013		2.1101e−011		1.1537e−013	
F2 Median	1.1369e−012	**2**	1.1369e−013	**5**	**2.2737e−013**	**1**	1.2847e−011	**4**	2.2521e−010	**3**
Means	1.9137e−012		4.7031e−004		**3.1074e−013**		3.8971e−009		2.2731e−010	
Std	2.4629e−012		1.8326e−003		**1.6334e−013**		2.1109e−008		5.8756e−011	
F3 Median	2.1150e+001	**5**	2.1139e+001	**3**	2.1139e+001	**4**	2.1139e+001	**2**	**2.1137e+001**	**1**
Means	2.1141e+001		2.1133e+001		2.1137e+001		2.1132e+001		**2.1126e+001**	
Std	3.7868e−002		3.2544e−002		2.9356e−002		3.5244e−002		4.7476e−002	
F4 Median	9.8647e−003	**2**	1.9690e−002	**5 s**	7.3960e−003	**3**	3.4335e−002	**4**	**1.0805e−008**	**1**
Means	1.1332e−002		1.4153e−001		1.4000e−002		1.1773e−001		**4.9419e−007**	
Std	9.3538e−003		2.7492e−001		2.3074e−002		2.5174e−001		**2.5368e−006**	
F5 Median	5.5284e+001	**5**	**4.8613e+001**	**1**	5.8524e+001	**4**	5.3264e+001	**2**	5.4002e+001	**3**
Means	5.5863e+001		**4.8548e+001**		5.8547e+001		5.0179e+001		5.3366e+001	
Std	3.8629e+000		**3.7885e+000**		1.9128e+000		6.7211e+000		2.9335e+000	
F6 Median	**1.1988e−008**	**1**	4.3902e−001	**5**	2.2737e−013	**3**	1.2703e−002	**4**	1.0767e−002	**2**
Means	**1.1426e−006**		5.9878e−001		1.3559e−002		7.3941e−002		1.0861e−002	
Std	**5.6020e−006**		6.0582e−001		4.3404e−002		1.4759e−001		5.9965e−003	
F7 Median	3.1192e−001	**3**	2.9602e−001	**2**	3.1955e−001	**4**	3.4870e−001	**5**	**2.9639e−001**	**1**
Means	3.1911e−001		2.9668e−001		3.9303e−001		4.3308e−001		**2.8768e−001**	
Std	2.9805e−002		3.7815e−002		1.9808e−001		1.7733e−001		**2.6285e−002**	
F8 Median	4.2812e+001	**4**	**1.5800e+001**	**2**	4.5782e+001	**5**	4.3253e+001	**3**	1.8699e+001	**1**
Means	4.0405e+001		3.1592e+001		4.2493e+001		4.0009e+001		**2.2345e+001**	
Std	2.1585e+001		2.6794e+001		1.9051e+001		1.0191e+001		**1.0594e+001**	
F9 Median	9.2618e+002	**3**	1.2114e+003	**5**	9.8472e+002	**4**	7.6788e+002	**2**	**3.4413e+002**	**1**
Means	8.7821e+002		1.2119e+003		9.5169e+002		7.7079e+002		**4.1316e+002**	
Std	2.6719e+002		7.4716e+001		1.6830e+002		1.5732e+002		**2.1819e+002**	
F10 Median	1.0971e+002	**2**	1.1219e+002	**5**	1.1113e+002	**4**	**1.0933e+002**	**1**	1.1082e+002	**3**
Means	1.0980e+002		1.1211e+002		1.1099e+002		**1.0963e+002**		1.1066e+002	
Std	1.0656e+000		1.5551e+000		1.3985e+000		**1.2632e+000**		7.1334e−001	
F11 Median	**6.2220e+001**	**1**	1.2069e+002	**2**	1.9601e+002	**4**	1.2341e+002	**3**	3.7023e+002	**5**
Means	**6.9989e+001**		1.1821e+002		1.8797e+002		1.3276e+002		5.1088e+002	
Std	**2.2727e+001**		3.5210e+001		6.2836e+001		5.0933e+001		3.4683e+002	
F12 Median	1.1781e+003	**4**	1.1904e+003	**3**	1.1613e+003	**2**	**1.0252e+003**	**1**	1.4440e+003	**5**
Means	1.2049e+003		1.1164e+003		1.0992e+003		**1.0303e+003**		1.3760e+003	
Std	2.1987e+002		3.0795e+002		2.4449e+002		**2.7057e+002**		2.0086e+002	
F13 Median	**2.0028e+002**	**2**	4.0945e+002	**3**	4.4803e+002	**5**	4.0674e+002	**4**	2.0557e+002	**1**
Means	2.3974e+002		3.3738e+002		4.1207e+002		3.3872e+002		**2.0582e+002**	
Std	8.9801e+001		1.0637e+002		8.5137e+001		1.0790e+002		**1.5892e+000**	
F14 Median	1.7848e+003	**5**	1.5639e+003	**2**	1.7615e+003	**4**	**1.4695e+003**	**1**	1.8330e+003	**3**
Means	1.7600e+003		1.5693e+003		1.7224e+003		**1.4378e+003**		1.6893e+003	
Std	1.4460e+002		9.4777e+001		1.3775e+002		**1.5003e+002**		4.1657e+002	
F15 Median	2.6981e+002	**2**	2.8995e+002	**5**	2.8706e+002	**4**	2.8268e+002	**3**	**2.6799e+002**	**1**

Table 8 (continued)

Func	BSA	*R*	SaDE	*R*	JADE	*R*	jDE	*R*	SSO	*R*
Means	2.6917e+002		2.9018e+002		2.8692e+002		2.8349e+002		**2.6691e+002**	
Std	4.5421e+000		5.9725e+000		4.9958e+000		5.7052e+000		**3.7057e+000**	
F16 Median	2.1569e+002	**2**	2.3308e+002	**5**	2.2702e+002	**4**	2.1943e+002	**3**	**2.1279e+002**	**1**
Means	2.1600e+002		2.3304e+002		2.2603e+002		2.1963e+002		**2.1293e+002**	
Std	4.5842e+000		5.1426e+000		1.0940e+001		9.3803e+000		**1.2290e+000**	
F17 Median	9.2742e+002	**3**	1.1318e+003	**5**	9.6140e+002	**4**	8.4286e+002	**2**	**5.1777e+002**	**1**
Means	9.3889e+002		1.1122e+003		9.6384e+002		8.2929e+002		**7.0296e+002**	
Std	1.3525e+002		1.0118e+002		1.1667e+002		1.0377e+002		**3.2077e+002**	
F18 Median	**1.3163e+003**	**1**	1.7734e+003	**5**	1.5654e+003	**4**	1.3642e+003	**2**	1.4378e+003	**3**
Means	**1.3129e+003**		1.8325e+003		1.7222e+003		1.3864e+003		1.4482e+003	
Std	**5.9923e+001**		2.9402e+002		4.7544e+002		1.8280e+002		**9.6766e+001**	
F19 Median	7.0794e+004	**4**	5.0123e+004	**2**	7.2396e+004	**3**	7.4893e+004		**4.9537e+004**	**1**
Means	6.5534e+004		6.1179e+004		6.5429e+004		7.0140e+004		**4.9663e+004**	
Std	8.1325e+003		1.2945e+004		1.0754e+004		9.3658e+003		**3.0839e+002**	
F20 Median	5.6375e+002	**4**	5.7785e+002	**5**	5.3497e+002	**3**	5.3096e+002	**2**	**5.2525e+002**	**1**
Means	5.5077e+002		5.6464e+002		5.3370e+002		5.3186e+002		**5.1969e+002**	
Std	4.0716e+001		3.9011e+001		4.3409e+001		4.0030e+001		**1.8342e+001**	
F21 Median	**6.0007e+002**	**1**	9.0162e+002	**5**	6.7106e+002	**3**	6.8332e+002	**4**	6.2315e+002	**2**
Means	**6.1416e+002**		9.2315e+002		6.6372e+002		7.0223e+002		6.2315e+002	
Std	**4.1198e+001**		9.7179e+001		5.8237e+001		8.3873e+001		3.3899e+001	

Table 9 Friedman test for SSO and other algorithms (D = 50)

Order	Algorithm	Average ranks	Statistical value	p value
1	**SSO**	**3.31**	44.52	1.13E−6
2	AAA	4.52		
3	ABC	4.57		
4	BSA	4.76		
5	jDE	5.00		
6	CS	5.07		
7	JADE	5.81		
8	SaDE	6.21		
9	TLBO	7.86		
10	GWO	7.88		

Strictly speaking, clustering problem is to determine a partition $G = \{C_1, C_2, \ldots, C_k\}$, $\forall k C_k \neq \emptyset$ and $\forall h \neq k, C_h \cap C_k = \emptyset$ such that data objects belonging to the same cluster are as similar as possible. At the same time, data objects belonging to the different clusters are as dissimilar as possible. One object can only belong to one cluster. To achieve this task, a mathematical model can be described as follows:

$$\operatorname{Min} J = \sum_{i=1}^{n} \sum_{j=1}^{k} d_{ij}(X_i - C_j)^2 \tag{36}$$

where n is the number of data objects, k is the number of clusters, X_i denotes the data object i and C_j denotes the cluster center j. $d_{ij}(X_i - C_j)$ means the Euclidean distance

Table 10 Comparison results of SSO and ABC, CS, TLBO, GWO, AAA (D = 100); the letter 'R' in the first line denotes the rank of different algorithms, and each algorithm is run 30 times for each problem

Func	ABC	***R***	CS	***R***	TLBO	***R***	GWO	***R***	AAA	***R***	SSO	***R***
F1 Median	2.0464e−012	**3**	**2.2737e−013**	**1**	5.6233e−008	**5**	1.7542e+004	**6**	1.3642e−012	**2**	1.8190e−012	**4**
Means	2.0994e−012		**1.6674e−013**		1.1316e−005		1.7287e+004		1.4476e−012		1.7887e−012	
Std	2.5810e−013		**1.0227e−013**		3.8588e−005		5.1988e+003		1.7392e−013		1.7646e−013	
F2 Median	3.1832e−012	**2**	5.3433e−012	**3**	5.4750e−004	**5**	1.5754e+003	**6**	**1.2506e−012**	**1**	3.4379e−010	**4**
Means	3.2098e−012		6.9804e−012		4.8257e+001		1.9438e+003		**1.2581e−012**		3.5138e−010	
Std	2.0399e−013		5.6820e−012		1.0992e+002		9.6823e+002		**1.3330e−013**		5.5610e−011	
F3 Median	2.1297e+001	**4**	**2.1259e+001**	**1**	2.1286e+001	**3**	2.1280e+001	**2**	2.1303e+001	**5**	2.1311e+001	**6**
Means	2.1291e+001		**2.1259e+001**		2.1280e+001		2.1269e+001		2.1295e+001		2.1304e+001	
Std	2.8038e−002		3.0356e−002		3.5153e−002		4.6151e−002		5.2875e−002		**2.5728e−002**	
F4 Median	6.3043e−006	**2**	7.3960e−003	**4**	6.1416e−002	**5**	2.4603e+002	**6**	**2.6148e−012**	**1**	2.7119e−005	**3**
Means	1.4711e−005		1.3174e−002		3.6907e−001		2.4649e+002		**2.6830e−012**		4.3804e−005	
Std	2.1411e−005		2.7105e−002		7.1866e−001		6.3812e+001		**5.8627e−013**		3.6992e−005	
F5 Median	1.4683e+002	**5**	1.3404e+002	2	1.4370e+002	**4**	1.5999e+002	**6**	1.3858e+002	**3**	**1.3108e+002**	**1**
Means	1.4486e+002		1.3339e+002		1.4383e+002		1.4514e+002		1.3721e+002		**1.3048e+002**	
Std	4.9353e+000		4.6702e+000		5.0002e+000		2.6189e+001		6.2412e+000		**4.3822e+000**	
F6 Median	**0.0000e+000**	**1**	6.1744e+001	**6**	5.2324e+001	**5**	2.4226e+001	**4**	1.2506e−012	**2**	4.6525e−002	**3**
Means	**1.8948e−014**		5.9368e+001		5.2254e+001		2.4118e+001		4.7738e−008		5.1257e−002	
Std	**5.2425e−014**		1.0127e+001		4.4644e+000		2.2276e+000		2.2512e−007		3.2688e−002	
F7 Median	**2.6695e−001**		3.0529e−001		3.3531e−001		7.4498e+001		3.3626e−001		3.1744e−001	
Means	**2.6574e−001**		3.0764e−001		3.4559e−001		7.7696e+001		3.3216e−001		3.1915e−001	
Std	**2.1759e−002**	**1**	2.4093e−002	**2**	1.1798e−001	**5**	2.4898e+001	**6**	2.6591e−002	4	1.6015e−002	**3**
F8 Median	**1.0444e+002**	**1**	1.1443e+002	**4**	1.6154e+002	**5**	2.2260e+002	**6**	1.1172e+002	**2**	1.1179e+002	**3**
Means	**9.2784e+001**		1.2130e+002		1.4190e+002		2.4148e+002		9.8499e+001		1.0450e+002	
Std	**2.4490e+001**		3.9906e+001		4.4359e+001		4.0371e+001		3.6008e+001		2.3013e+001	
F9 Median	4.2982e+002	**2**	3.2897e+003	**5**	4.0090e+003	**6**	2.5493e+003	**4**	2.6724e+003	**3**	**4.4172e+002**	**1**
Means	1.4184e+003		2.8231e+003		3.7788e+003		2.5536e+003		2.3279e+003		**8.4227e+002**	
Std	1.1627e+003		1.0958e+003		8.8358e+002		1.6879e+002		9.3047e+002		**8.1148e+002**	
F10 Median	**1.1533e+002**	**1**	1.2055e+002	**4**	1.2540e+002	**5**	1.3095e+002	6	1.1712e+002	**2**	1.1804e+002	**3**
Means	**1.1531e+002**		1.2044e+002		1.2844e+002		1.3375e+002		1.1709e+002		1.1786e+002	
Std	**3.9812e−001**		9.8353e−001		1.1433e+001		8.0200e+000		5.8635e−001		5.9105e−001	
F11 Median	7.6563e+004	**6**	**1.2288e+003**	**1**	1.2776e+003	**2**	5.8779e+003	**3**	5.5242e+003	**4**	2.4851e+004	**5**
Means	7.3332e+004		**1.1941e+003**		1.3150e+003		6.0667e+003		7.0314e+003		2.3755e+004	
Std	1.9247e+004		**1.5687e+002**		2.4908e+002		1.5534e+003		5.4330e+003		8.3711e+003	
F12 Median	3.9367e+003	**3**	4.3126e+003	**6**	4.0704e+003	**5**	3.8968e+003	**4**	3.6544e+003	**2**	**3.4503e+003**	**1**
Means	3.8490e+003		4.2858e+003		4.0836e+003		4.0609e+003		3.7579e+003		**3.3709e+003**	
Std	3.6847e+002		3.8961e+002		7.8834e+002		1.2722e+003		6.6979e+002		**4.2151e+002**	
F13 Median	**2.0266e+002**	**1**	6.3200e+002	**4**	6.4260e+002	**6**	5.5809e+002	**5**	2.0056e+002	**3**	2.1640e+002	**2**
Means	**2.1312e+002**		4.9731e+002		6.4346e+002		5.8190e+002		3.0708e+002		2.5808e+002	
Std	**5.3617e+001**		2.0919e+002		1.2992e+001		5.9161e+001		1.9663e+002		1.3087e+002	
F14 Median	4.3944e+003	**5**	4.0407e+003	**3**	4.1225e+003	**4**	**3.1018e+003**	**1**	4.1635e+003	**6**	3.9323e+003	**2**
Means	4.1131e+003		3.9953e+003		4.1102e+003		**3.3266e+003**		4.1525e+003		3.7355e+003	
Std	1.0160e+003		1.4547e+002		1.9231e+002		**5.8861e+002**		1.1980e+002		8.4723e+002	
F15 Median	3.6459e+002	**3**	3.8814e+002	**6**	**2.0000e+002**	**1**	**2.0001e+002**	**2**	3.6872e+002	**5**	3.6837e+002	**4**

Table 10 (continued)

Func	ABC	*R*	CS	*R*	TLBO	*R*	GWO	*R*	AAA	*R*	SSO	*R*
Means	3.6431e+002		3.8743e+002		**2.0000e+002**		**2.0001e+002**		3.6878e+002		3.6794e+002	
Std	2.8547e+000		8.0600e+000		1.5014e−004		**3.8466e−003**		3.4150e+000		3.0710e+000	
F16 Median	2.4920e+002	**3**	2.6464e+002	**6**	**2.0000e+002**	**1**	2.8755e+002	**5**	2.4636e+002	**2**	2.4988e+002	**4**
Means	2.4857e+002		2.6060e+002		**2.0000e+002**		2.5609e+002		2.4665e+002		2.4942e+002	
Std	5.0972e+000		1.6121e+001		**1.7345e−013**		4.6885e+001		1.3001e+001		3.5920e+000	
F17 Median	2.5233e+003	**2**	3.1433e+003	**5**	3.7690e+003	**6**	2.3582e+003	**3**	2.6257e+003	**4**	**2.2818e+003**	**1**
Means	2.3403e+003		2.8571e+003		3.7517e+003		2.3566e+003		2.6261e+003		**2.2267e+003**	
Std	6.4589e+002		8.1824e+002		2.0683e+002		1.6144e+002		1.7745e+002		**3.2310e+002**	
F18 Median	7.1127e+003	**5**	5.8236e+003	**4**	8.8959e+003	**6**	5.0211e+003	**3**	4.1398e+003	**2**	**3.7170e+003**	**1**
Means	6.9120e+003		5.7278e+003		8.8114e+003		4.9455e+003		3.9193e+003		**3.6734e+003**	
Std	8.5452e+002		1.4489e+003		1.3702e+003		7.8070e+002		7.7612e+002		**5.0804e+002**	
F19 Median	1.0889e+005	**2**	1.0892e+005	**3**	1.3119e+005	**4**	1.5643e+005	**5**	1.0889e+005	**2**	**1.0888e+005**	**1**
Means	1.0889e+005		1.1006e+005		1.3833e+005		1.5516e+005		1.0889e+005		**1.0888e+005**	
Std	1.0769e+001		6.1291e+003		2.0294e+004		7.8999e+003		1.3421e+001		**1.5702e+001**	
F20 Median	**6.9723e+002**	**1**	8.2532e+002	**4**	8.4195e+002	**5**	2.5668e+003	**6**	7.6962e+002	**2**	7.6825e+002	**3**
Means	**6.9360e+002**		8.0551e+002		8.2103e+002		2.6230e+003		7.6135e+002		7.6320e+002	
Std	4.8699e+001		6.5525e+001		8.2117e+001		4.3063e+002		6.6890e+001		**4.2440e+001**	
F21 Median	8.0476e+002	**3**	9.5985e+002	**4**	1.8051e+003	**6**	1.0240e+003	**5**	7.7410e+002	**2**	**7.2929e+002**	**1**
Means	8.0164e+002		9.5919e+002		1.8080e+003		1.0313e+003		7.7620e+002		**7.3572e+002**	
Std	3.7017e+001		1.0050e+002		3.3959e+002		8.8076e+001		3.3114e+001		**2.8815e+001**	

between the data object X_i and the cluster center C_j. To minimize J in Eq. (36), we must find the best cluster centers. It is an NP optimization problem, and many meta-heuristic methods have been used to solve it [51–56].

In this test, we choose six benchmark datasets with different dimensions from the repository of the machine learning databases [57]. They are named Iris, Wine, Glass, Wisconsin Breast Cancer, Vowel and Contraceptive Method Choice (CMC). The details of six datasets are shown in Table 15. Six methods including CMA-ES, GSA, GWO, BB-BC, BH and TLBO are used to compare the SSO. The population size is set as $ps = 20$, and the max fitness evaluation (MAX_FES) is set as D*1E4. (D denotes the dimension.) The parameters of CMA-ES and GSA are adopted in the literature [19, 49]. The parameter setting of GWO and TLBO is same as above in this paper. Each algorithm except BB-BC and BH is run 20 times, and the min values, max values, means values and standard deviations are recorded. The experimental results of BB-BC and BH are cited from the literature [27] for comparison. Table 16 shows that the performance of SSO outperforms the other six meta-heuristic methods obviously. For the cancer dataset, SSO and CMA-ES obtain almost the same performance. For CMC dataset, SSO and TLBO obtain almost the same performance. Table 17 shows the average ranks by Friedman test and the statistic value. It validates

Table 11 Comparison results of SSO and BSA, SaDE, JADE, jDE, AAA (D = 100); the letter 'R' in the first line denotes the rank of different algorithms, and each algorithm is run 30 times for each problem

Func	BSA	***R***	SaDE	***R***	JADE	***R***	jDE	***R***	SSO	***R***
F1 Median	2.9559e−012	**3**	**2.2737e−013**	**1**	2.5011e−012	**4**	5.8890e−011	**5**	1.8190e−012	**2**
Means	3.3272e−012		**2.2737e−013**		1.1611e−011		5.8907e−010		1.7887e−012	
Std	1.1415e−012		**0.0000e+000**		3.9716e−011		1.1728e−009		1.7646e−013	
F2 Median	3.2219e−009	**3**	1.1369e−013	**5**	**1.2506e−012**	**1**	5.0909e−010	**4**	3.4379e−010	**2**
Means	1.0027e−008		1.4061e−002		**1.9440e−012**		2.2642e−006		3.5138e−010	
Std	1.9932e−008		5.1572e−002		**1.6285e−012**		1.0917e−005		5.5610e−011	
F3 Median	**2.1263e+001**	**1**	2.1309e+001	**3**	2.1302e+001	**4**	2.1293e+001	**5**	2.1311e+001	**2**
Means	**2.1257e+001**		2.1309e+001		2.1287e+001		2.1290e+001		2.1304e+001	
Std	3.3586e−002		1.9152e−002		3.6010e−002		2.3729e−002		**2.5728e−002**	
F4 Median	5.3433e−012	**2**	9.8573e−003	**4**	1.2756e−010	**3**	1.4772e−002	**5**	**2.7119e−005**	**1**
Means	6.7211e−003		5.5025e−002		9.8186e−003		1.9177e−001		**4.3804e−005**	
Std	1.0639e−002		1.4320e−001		1.5191e−002		4.5982e−001		**3.6992e−005**	
F5 Median	1.3707e+002	**4**	**1.2974e+002**	**2**	1.4086e+002	**5**	1.2976e+002	**1**	1.3108e+002	**3**
Means	1.3600e+002		1.2826e+002		1.4090e+002		**1.2561e+002**		1.3048e+002	
Std	6.3049e+000		**6.6040e+000**		3.9810e+000		1.0633e+001		4.3822e+000	
F6 Median	**4.3783e−005**	**1**	2.5319e+000	**5**	1.1369e−012	**2**	2.1573e−001	**4**	4.6525e−002	**3**
Means	**9.2393e−005**		2.9684e+000		3.8327e−002		3.7635e−001		5.1257e−002	
Std	**1.5222e−004**		1.9513e+000		1.6663e−001		5.7102e−001		3.2688e−002	
F7 Median	3.1808e−001	**2**	**3.0786e−001**	**1**	3.2862e−001	**4**	3.4486e−001	**5**	3.1744e−001	**3**
Means	3.1636e−001		**3.0802e−001**		3.6385e−001		3.6421e−001		3.1915e−001	
Std	2.0594e−002		**2.6805e−002**		1.3746e−001		1.3731e−001		1.6015e−002	
F8 Median	**9.7652e+001**	**1**	1.0483e+002	**3**	1.1734e+002	**4**	1.1374e+002	**5**	1.1179e+002	**2**
Means	**9.8495e+001**		1.0875e+002		1.1845e+002		1.2231e+002		1.0450e+002	
Std	**4.4767e+001**		4.3131e+001		3.9487e+001		1.8243e+001		2.3013e+001	
F9 Median	2.1517e+003	**3**	2.8903e+003	**5**	2.4602e+003	**4**	2.0469e+003	**2**	**4.4172e+002**	**1**
Means	2.1066e+003		2.7450e+003		2.4619e+003		2.0312e+003		**8.4227e+002**	
Std	2.5703e+002		6.8062e+002		2.1168e+002		1.8244e+002		**8.1148e+002**	
F10 Median	**1.1682e+002**	**1**	1.2070e+002	**5**	1.1977e+002	**4**	1.1859e+002	**3**	1.1804e+002	**2**
Means	**1.1686e+002**		1.2041e+002		1.1977e+002		1.1890e+002		1.1786e+002	
Std	**6.8572e−001**		1.6347e+000		1.5376e+000		1.5577e+000		5.9105e−001	
F11 Median	**2.6438e+002**	**1**	7.0939e+002	**3**	1.2465e+003	**4**	6.3014e+002	**2**	2.4851e+004	**5**
Means	**2.7124e+002**		7.0650e+002		1.1385e+003		6.2168e+002		2.3755e+004	
Std	**7.0528e+001**		1.3613e+002		4.1988e+002		1.4048e+002		8.3711e+003	
F12 Median	3.0992e+003	**3**	3.3096e+003	**4**	**2.9123e+003**	**1**	2.9271e+003	**2**	3.4503e+003	**5**
Means	3.0785e+003		3.3256e+003		**2.8690e+003**		2.8802e+003		3.3709e+003	
Std	3.4215e+002		5.6298e+002		**3.5469e+002**		4.7754e+002		4.2151e+002	
F13 Median	6.5166e+002	**4**	5.9207e+002	**3**	6.4068e+002	**5**	5.7055e+002	**2**	**2.1640e+002**	**1**
Means	6.1597e+002		5.8969e+002		6.3221e+002		5.8324e+002		**2.5808e+002**	
Std	1.1435e+002		1.9703e+001		2.8494e+001		3.1705e+001		**1.3087e+002**	
F14 Median	3.6405e+003	**3**	3.4612e+003	**2**	3.7808e+003	**4**	**3.2177e+003**	**1**	3.9323e+003	**5**
Means	3.5984e+003		3.4600e+003		3.7027e+003		**3.2593e+003**		3.7355e+003	
Std	3.4682e+002		1.5980e+002		2.5202e+002		**2.3497e+002**		8.4723e+002	
F15 Median	3.8225e+002	**2**	4.2912e+002	**3**	4.4178e+002	**5**	4.3081e+002	**4**	**3.6837e+002**	**1**

Table 11 (continued)

Func	BSA	*R*	SaDE	*R*	JADE	*R*	jDE	*R*	SSO	*R*
Means	3.8419e+002		4.2804e+002		4.4196e+002		4.2912e+002		**3.6794e+002**	
Std	8.5780e+000		1.1130e+001		1.1878e+001		1.5749e+001		**3.0710e+000**	
F16 Median	2.6685e+002	**2**	2.8060e+002	**3**	3.0877e+002	**5**	3.0246e+002	**4**	**2.4988e+002**	**1**
Means	2.6556e+002		2.7949e+002		3.1155e+002		2.9726e+002		**2.4942e+002**	
Std	7.2251e+000		1.4265e+001		1.4720e+001		1.8061e+001		**3.5920e+000**	
F17 Median	1.9635e+003	**2**	2.5218e+003	**5**	2.4378e+003	**4**	**1.8724e+003**	**1**	2.2818e+003	**3**
Means	1.9450e+003		2.5184e+003		2.4237e+003		**1.8632e+003**		2.2267e+003	
Std	3.1311e+002		1.4766e+002		2.1060e+002		**1.9811e+002**		3.2310e+002	
F18 Median	**2.6176e+003**	**1**	4.7201e+003	**5**	4.4384e+003	**4**	3.8257e+003	**3**	3.7170e+003	**2**
Means	**2.7713e+003**		4.6723e+003		4.3907e+003		3.8290e+003		3.6734e+003	
Std	**4.0448e+002**		7.1006e+002		9.4626e+002		8.8647e+002		5.0804e+002	
F19 Median	1.0895e+005	**2**	1.5897e+005	**4**	1.0894e+005	**5**	1.0894e+005	**3**	**1.0888e+005**	**1**
Means	1.0944e+005		1.3879e+005		1.1516e+005		1.1050e+005		**1.0888e+005**	
Std	2.6813e+003		2.6857e+004		1.4266e+004		8.5511e+003		**1.5702e+001**	
F20 Median	8.4316e+002	**5**	8.3514e+002	**4**	7.9791e+002	**2**	8.0728e+002	**3**	**7.6825e+002**	**1**
Means	8.2283e+002		8.1831e+002		7.8953e+002		7.9017e+002		**7.6320e+002**	
Std	6.5051e+001		7.4125e+001		8.2660e+001		6.3669e+001		**4.2440e+001**	
F21 Median	7.4332e+002	**2**	1.4147e+003	**5**	9.7435e+002	**4**	8.7597e+002	**3**	**7.2929e+002**	**1**
Means	7.5403e+002		1.3895e+003		9.6799e+002		8.8912e+002		**7.3572e+002**	
Std	4.2268e+001		1.6366e+002		1.0316e+002		8.7566e+001		**2.8815e+001**	

Table 12 Friedman test for SSO and other algorithms (D = 100)

Order	Algorithm	Average ranks	Statistical value	*p* value
1	**SSO**	**3.95**	41.68	3.75E−6
2	BSA	4.00		
3	ABC	4.07		
4	AAA	4.36		
5	jDE	5.24		
6	JADE	5.90		
7	CS	6.00		
8	SaDE	6.19		
9	GWO	7.48		
10	TLBO	7.81		

that the performance of SSO is better than other methods. In addition, Fig. 8 shows the results of data clustering in three-dimensional space. It can be observed that the six datasets have been divided into the different clusters better according to Eq. (25).

6 Conclusions

In this paper, we study the mechanism of meta-heuristic approach in these years. We find that search style and individual selection method are the two core problems for design. We research the mechanism of the basic hypercube search style and the basic reduced hypercube search style.

Table 13 Parameter analysis

Parameter	Method	CEC2017_6	CEC2017_11	CEC2017_16	CEC2017_25	CEC2017_27
S = 0.1	Means	6.0329e−003	5.6863e+002	1.6691e+003	2.2860e+003	2.6080e+003
	Std.	7.1885e−003	1.9699e+001	1.8931e+002	1.3759e+000	7.5889e+000
S = 0.2	Means	2.9274e−003	5.6638e+002	1.6956e+003	**2.2849e+003**	2.6093e+003
	Std.	3.1720e−003	2.1519e+001	1.3459e+002	**1.4514e+000**	5.6917e+000
S = 0.3	Means	4.1100e−003	5.6768e+002	1.6906e+003	2.2856e+003	2.6083e+003
	Std.	4.1355e−003	2.1136e+001	1.1629e+002	1.5766e+000	5.7455e+000
S = 0.4	Means	**2.0498e−003**	5.7720e+002	**1.6609e+003**	2.2854e+003	2.6095e+003
	Std.	**1.8156e−003**	2.4302e+001	**2.0056e+002**	1.5195e+000	5.5165e+000
S = 0.5	Means	3.2051e−003	5.6222e+002	1.7016e+003	2.2856e+003	2.6088e+003
	Std.	3.1340e−003	2.4237e+001	1.4585e+002	1.4134e+000	5.3047e+000
S = 0.6	Means	3.2785e−003	5.6399e+002	1.6908e+003	2.2859e+003	2.6095e+003
	Std.	4.9721e−003	1.7234e+001	2.0996e+002	1.6164e+000	6.6756e+000
S = 0.7	Means	3.0064e−003	**5.6153e+002**	1.7113e+003	2.2855e+003	2.6072e+003
	Std.	3.2968e−003	**1.7924e+001**	1.5028e+002	1.3547e+000	6.3901e+000
S = 0.8	Means	3.2564e−003	5.6865e+002	1.6691e+003	2.2855e+003	2.6078e+003
	Std.	4.4996e−003	2.1306e+001	1.3075e+002	1.4260e+000	8.0445e+000
S = 0.9	Means	2.2522e−003	5.6934e+002	1.6730e+003	2.2854e+003	**2.6071e+003**
	Std.	2.6624e−003	2.3333e+001	1.6890e+002	1.5703e+000	**4.8341e+000**
S = 1.0	Means	2.3001e−003	5.7292e+002	1.6618e+003	2.2859e+003	2.6096e+003
	Std.	2.3358e−003	2.8961e+001	1.4811e+002	1.4127e+000	7.8222e+000

Table 14 Friedman test for SSO parameters (D = 30)

Order	Algorithm	Average ranks	Statistical value	p value
1	**s = 0.09**	**3.70**	6.198	0.72
2	s = 0.07	4.50		
3	s = 0.04	4.60		
4	s = 0.02	4.80		
5	s = 0.08	5.00		
6	s = 0.05	5.9		
7	s = 0.03	6.30		
8	s = 0.10	6.50		
9	s = 0.01	6.70		
9	s = 0.06	7.00		

These basic styles are easy to be achieved but are limited in the hypercube cube regions, which limits the diversity of the search.

Therefore, we propose the simple spherical search style and design a spherical search optimizer (SSO) to solve continuous optimization problems. The individual is selected by the traditional tournament selection method. Then, theoretical analysis confirms that the search by the simple spherical search style in three-dimensional space can also obtain the same solution like the basic hypercube search style. The convergence analysis demonstrates that the SSO algorithm is no convergence in probability. It means that the diversity of the population should be considered for the improved version of SSO in the future. At the end, the proposed approach is compared with the 15

Table 15 Details of six dataset

No.	Dataset	Number of cluster	Dimension	Distribution (sample numbers of each class)
1	Iris	3	4	150 (50, 50, 50)
2	Wine	3	13	178 (59, 71, 48)
3	Glass	6	9	214 (70, 76, 17, 13, 9, 29)
4	Cancer	2	9	683 (444, 239)
5	Vowel	6	3	871 (72, 89, 172, 151, 207, 180)
6	CMC	3	9	1473 (629, 334, 510)

Table 16 Comparison results of SSO and other state-of-the-art algorithms for clustering problem, and each algorithm is run 20 times for each problem

	CMA-ES	GSA	GWO	BB-BC	BH	TLBO	SSO
1 Min	**96.6554**	**96.6554**	96.7041	97.42865	96.66306	**96.6554**	**96.6554**
Max	127.6676	114.0822	125.4239	96.67648	96.65589	97.1967	**96.6554**
Means	100.9600	99.4998	105.0347	96.76537	96.65681	96.7676	**96.6554**
Std	10.5933	5.0037	11.1449	0.20456	0.00173	0.2201	**2.5876 e−014**
2 Min	**16,292.1846**	17,578.1931	16,305.3804	16,310.11354	16,300.22613	**16,292.1846**	**16,292.1846**
Max	16,293.8142	23,765.9908	16,510.4391	16,298.67356	16,293.41995	16,294.1704	**16,292.2324**
Means	16,292.5888	19,413.1843	16,334.7371	16,303.41207	16,294.31763	16,293.5504	**16,292.1925**
Std	0.4796	1469.7893	42.3188	2.66198	1.65127	0.6966	**0.015083**
3 Min	210.4286	211.1662	263.8064	243.20883	213.95689	210.4507	**210.4287**
Max	246.6827	251.0337	337.7799	223.89410	210.51549	246.6827	**210.5024**
Means	218.6723	222.5004	299.1684	231.23058	211.49860	230.0070	**210.4401**
Std	10.23917	11.6018	19.51238	4.65013	1.18230	13.7858	**0.0166954**
4 Min	**2964.3869**	**2964.3869**	2964.4375	2964.38902	2964.45074	**2964.3869**	**2964.3869**
Max	**2964.3869**	2964.5215	2964.5778	2964.38753	2964.38878	2969.4348	**2964.3869**
Means	**2964.3869**	2964.4475	2964.4877	2964.38798	2964.39539	2964.6393	**2964.3869**
Std	**1.3761 e−012**	0.0686	0.0464	0.00048	0.00921	1.1287460	1.5890 e−012
5 Min	**148,967.2408**	219,806.2499	149,223.7265	153,090.44077	153,058.98663	149,041.1312	148,987.9948
Max	168,340.2569	289,496.3018	161,879.8795	149,038.51683	148,985.61373	150,852.8749	**149,259.2519**
Means	152,379.6293	255,941.8163	153,694.8208	151,010.03392	149,848.18144	149,755.6462	**149,094.9870**
Std	6485.2149	19,516.6776	4388.9611	1859.32353	1306.95375	704.6323900	**70.80506894**
6 Min	5532.1847	5532.4679	5586.1561	5644.70264	5534.77738	**5532.1847**	**5532.1847**
Max	5532.1847	5551.8629	6022.1975	5534.09483	5532.88323	**5532.1847**	**5532.1847**
Means	5532.1847	5534.3340	5786.5272	5574.75174	5533.63122	**5532.1847**	**5532.1847**
Std	9.7532 e−012	4.313871	137.7478	39.43494	0.59940	1.2455 e−009	**3.9808 e−012**

Table 17 Friedman test for SSO parameters (D = 100)

Order	Algorithm	Average ranks	Statistical value	p value
1	**SSO**	**1.92**	20.87	0.002
2	BH	2.42		
3	BB-BC	3.50		
4	CMA-ES	3.58		
5	TLBO	4.50		
6	GSA	5.50		
7	GWO	6.50		

state-of-the-art algorithms for the different CEC benchmark suites and a real-world optimization problem. Experiments verify the effectiveness of the proposed approach.

In the future, other works can be considered. First, we only focus on the search style and do not research deeply for the individual selection method. New individual selection method inspired by nature can be proposed. Second, two angles of the sphere are produced by the uniform distribution and the radius of the sphere is adjusted by a normal distribution. Adjustment method of angles and the radius of spherical search style can be researched.

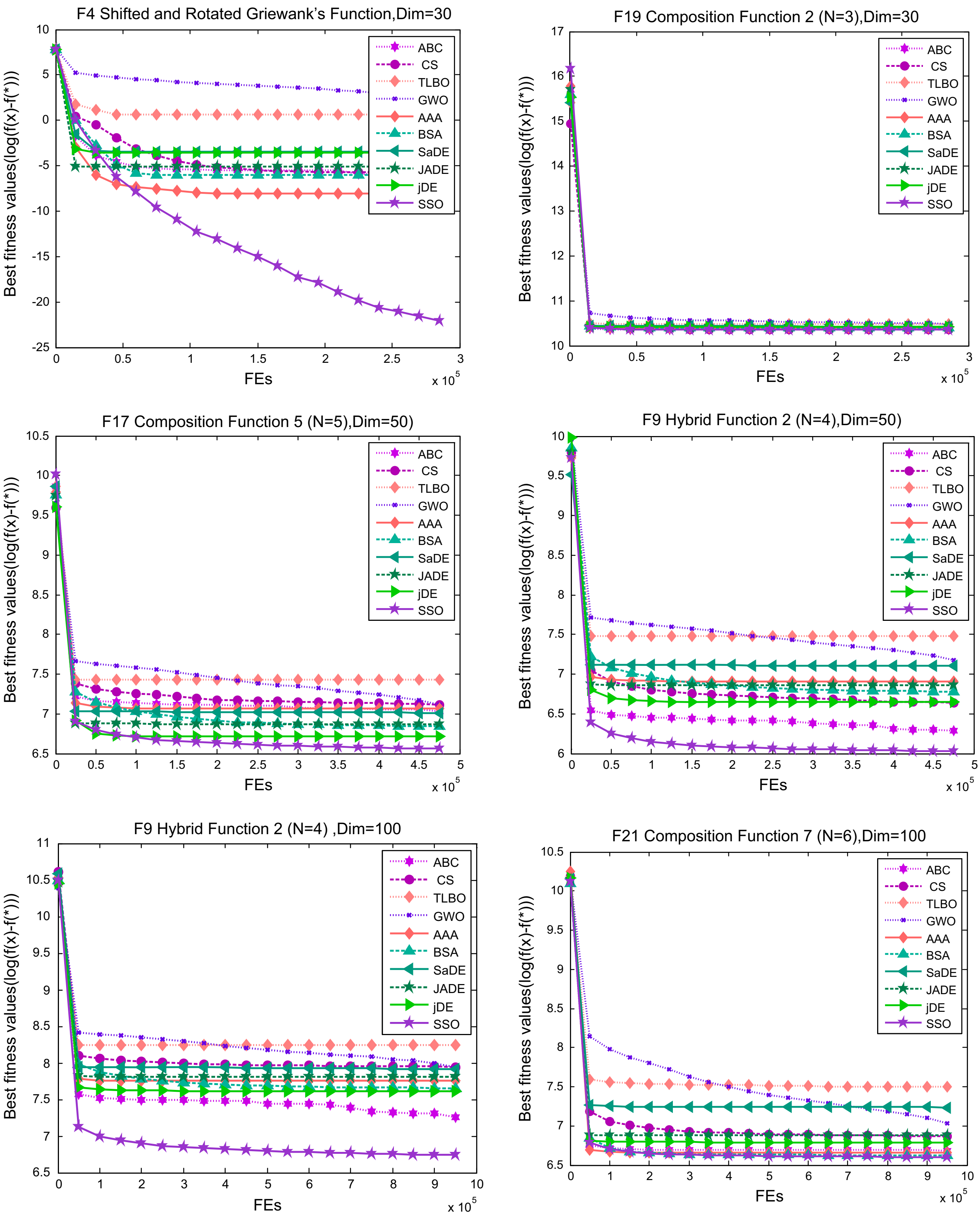


Fig. 7 Clustering results of SSO in three-dimensional space

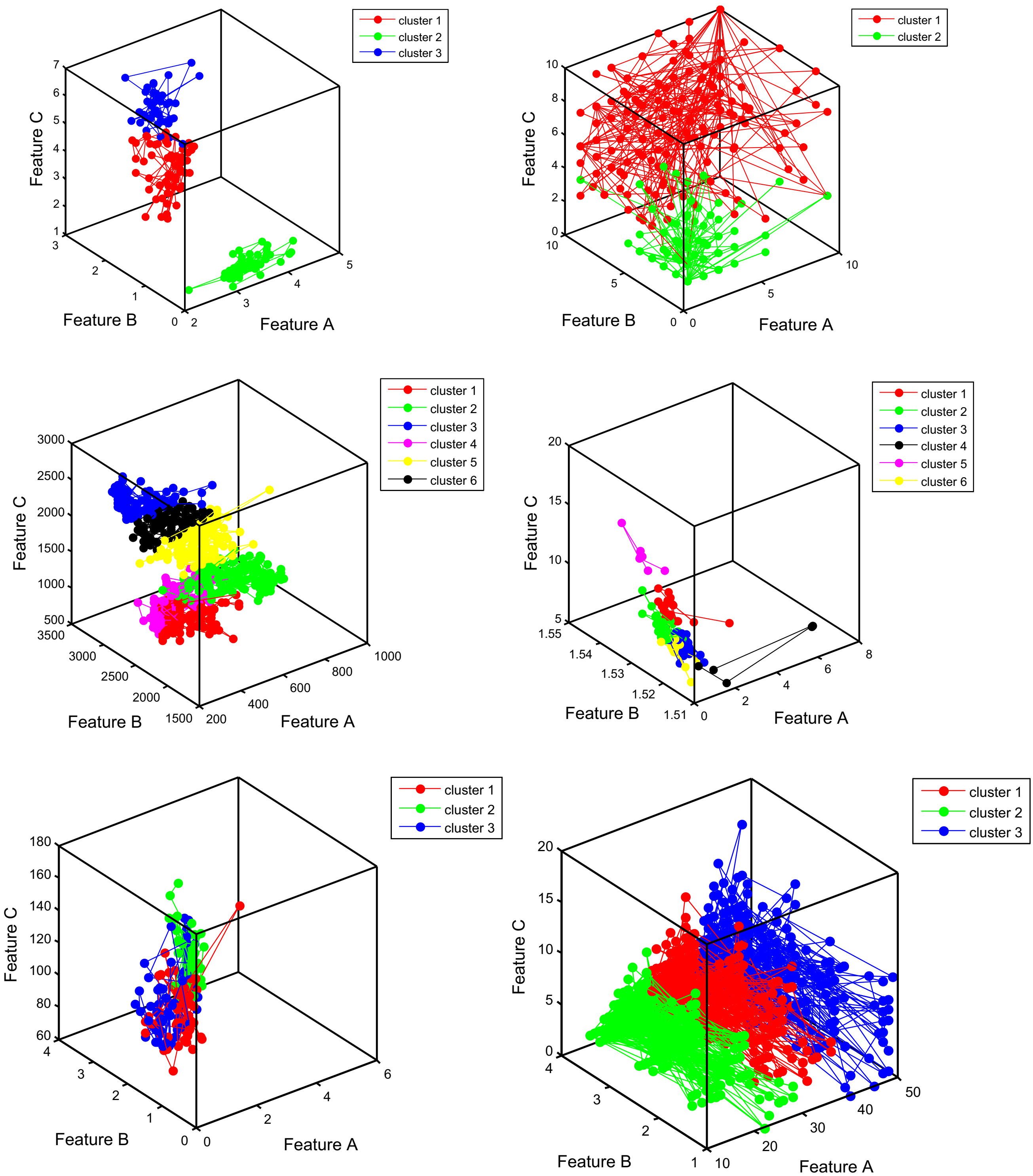


Fig. 8 Best clustering results of SSO in three-dimensional space

Acknowledgements This work is supported by the Guang Dong Provincial Natural Fund Project (2016A030310300); the National Natural Science Foundation of China (71871069, 71401045, 61976239); the Ministry of Education in China Project of Humanities and Social Sciences (18YJAZH137); the Guangdong Provincial Natural Fund Project (2017A030313394); the major scientific research projects of Guangdong (2017WTSCX021); the planning project of the 13th Five-Year in Philosophy and Social Sciences of Guangzhou (2018GZGJ48); the Ministry of Education Science and Technology Development Center (2017A11001); and the Guangdong University Engineering Technology Research Center (2016GCZX004). This research was funded by the Guangdong Natural Science Foundation (Grant No. 2015A030308017) and the Guangdong Science and Technology Key Project (Grant No. 2015B010131009).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Rizk-Allah RM, Hassanien AE, Elhoseny M, Gunasekaran M (2019) A new binary salp swarm algorithm: development and application for optimization tasks. Neural Comput Appl 31:1641–1663
2. Boveiri HR, Elhoseny M (2019) A-COA: an adaptive cuckoo optimization algorithm for continuous and combinatorial optimization. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3928-9
3. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceeding IEEE international conference neural network, Perth, Western Australia, pp 1942–1948
4. Dorigo M, Birattari M, Stützle T, Libre U, Bruxelles D, Roosevelt AFD (2006) Ant colony optimization -artificial ants as a computational intelligence technique. IEEE Comput Intell Mag 1:28–39
5. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 26:29–41
6. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theoret Comput Sci 344(2):243–278
7. Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manag 129(3):210–225
8. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, vol 200. Erciyes University, Engineering Faculty, Computer Engineering Department, pp 1–10
9. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: Proceeings of world congress on nature and biologically inspired computing. IEEE Publications, USA, pp 210–214
10. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12:702–713
11. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183:1–15
12. Grefenstette JJ (1986) Optimization of control parameters for genetic algorithms. IEEE Trans Syst Man Cybern 16(1):122–128
13. Rechenberg I (1973) Evolution strategies: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog, Stuttgart
14. Yao X, Liu Y (1996) Fast evolutionary programming. Evolut Program 3:451–460
15. Storn RM, Price KV (1997) Differential evolution -a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359
16. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evolut Comput 10(6):646–657
17. Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: Proceedings of IEEE congress on evolutionary computation, vol 2. pp 1785–179
18. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evolut Comput 13(5):945–958
19. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
20. Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37:106–111
21. Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294
22. Du H, Wu X, Zhuang J (2006) Small-world optimization algorithm for function optimization. In: International conference on natural computation. Springer, Berlin, Heidelberg, pp 264–273
23. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133
24. Wu G (2016) Across neighborhood search for numerical optimization. Inf Sci 329:597–618
25. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
26. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84
27. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci 222:175–184
28. Uymaz SA, Tezel G, Yel E (2015) Artificial algae algorithm (AAA) for nonlinear global optimization. Appl Soft Comput 31:153–171
29. Nematollahi F, Rahiminejad A, Vahidi B (2017) A novel physical based meta-heuristic optimization method known as lightning attachment procedure optimization. Appl Soft Comput 59:596–621
30. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
31. Ghaemi M, Feizi-Derakhshi M-R (2014) Forest optimization algorithm. Expert Syst Appl 41:6676–6687
32. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
33. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70
34. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
35. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
36. Tang D, Dong S, Jiang Y, Li H, Huang Y (2015) ITGO: invasive tumor growth optimization algorithm. Appl Soft Comput 36:670–698
37. Gao Y, Zhang G, Lu J, Wee HM (2011) Particle swarm optimization for bi-level pricing problems in supply chains. J Glob Optim 51:245–254
38. Zhan ZH, Zhang J, Li Y, Chung HSH (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B (Cybern) 39(6):1362–1381

39. Wang GG, Gandomi AH, Yang XS (2014) A novel improved accelerated particle swarm optimization algorithm for global numerical optimization. Eng Comput 31(7):1198–1220
40. Tang D (2019) Spherical evolution for solving continuous optimization problems. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.105499
41. Hu ZB, Xiong SW, Su QH, Fang ZX (2014) Finite Markov chain analysis of classical differential evolution algorithm. J Comput Appl Math 268:121–134
42. Zhang H, Cao X, Ho JK, Chow TW (2016) Object-level video advertising: an optimization framework. IEEE Trans Ind Inf 13(2):520–531
43. Milner S, Davis C, Zhang H, Llorca J (2012) Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks. IEEE Trans Mob Comput 11(7):1207–1222
44. Liang JJ, Qu BY, Suganthan PN, Hernández-Díaz AG (2013) Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 201212(34), pp 281–295
45. Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 635
46. Liang JJ, Qu BY, Suganthan PN, Chen Q (2014) Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Technical Report 201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, vol 29, pp 625–640
47. Awad NH, Ali MZ, Liang JJ, Qu BY, Suganthan PN (2017) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. Nanyang Technological University, Singapore, Jordan University of Science and Technology, Jordan and Zhengzhou University, Zhengzhou China, Technical Report 2017
48. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219:8121–8144
49. Civicioglu P (2013) Artificial cooperative search algorithm for numerical optimization problems. Inf Sci 229:58–76
50. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1:3–18
51. Das S, Abraham A, Konar A (2009) Automatic hard clustering using improved differential evolution algorithm. In: Metaheuristic clustering. Springer, Berlin, Heidelberg, pp 137–174
52. Fathian M, Amiri B, Maroosi A (2007) Application of honey-bee mating optimization algorithm on clustering. Appl Math Comput 190:1502–1513
53. Hatamlou A, Abdullah S, Nezamabadi-Pour H (2011) Application of gravitational search algorithm on data clustering. In: International conference on rough sets and knowledge technology. Springer, Berlin, Heidelberg, pp 337–346
54. Hatamlou A, Abdullah S, Nezamabadi-pour H (2012) A combined approach for clustering based on K-means and gravitational search algorithms. Swarm Evolut Comput 6:47–52
55. Hatamlou A, Abdullah S, Hatamlou M (2011) Data clustering using big bang–big crunch algorithm. In: International conference on innovative computing technology. Springer, Berlin, Heidelberg, pp 383–388
56. Satapathy SC, Naik A (2011) Data clustering based on teaching-learning-based optimization. In: International conference on swarm, evolutionary, and memetic computing. Springer, Berlin, Heidelberg, pp 148–156
57. Blake CL, Merz CJ (1998) UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Sciences. http://www.ics.uci.edu/mlearn/MLRepository.html