



BSO20: efficient brain storm optimization for real-parameter numerical optimization

Peilan Xu¹ · Wenjian Luo² · Xin Lin¹ · Shi Cheng³ · Yuhui Shi⁴

Received: 19 November 2020 / Accepted: 18 May 2021 / Published online: 17 June 2021
© The Author(s) 2021

Abstract

Brain storm optimization (BSO) is an emerging global optimization algorithm. The primary idea is to divide the population into different clusters, and offspring are generated within a cluster or between two clusters. However, the problems of inefficient clustering strategy and insufficient exploration exist in BSO. In this paper, a novel and efficient BSO is proposed, called BSO20 (proposed in 2020). BSO20 pays attention to both the clustering strategy and the mutation strategy. First, we propose a hybrid clustering strategy, which combines two clustering strategies, i.e., nearest-better clustering and random grouping strategy. The size of the subpopulation clustered by two strategies is dynamically adjusted as the population evolves. Second, a modified mutation strategy is used in BSO20 to share information within a cluster or among multiple clusters to enhance the ability of exploration. BSO20 is tested on the problems of the 2017 IEEE Congress on Evolutionary Computation competition on real parameter numerical optimization. BSO20 is compared with several variants of BSO and two variants of particle swarm optimization, and the experimental results show that BSO20 is competitive.

Keywords Real-parameter numerical optimization · Brain storm optimization · Nearest-better clustering

Introduction

In human society, brainstorming is an effective method when a group of people is facing a difficult problem. People with different backgrounds are gathered together to constantly propose new ideas in the process of cooperation and mutual inspiration and finally solve the problem. The brain storm optimization (BSO) [29] is a novel swarm intelligence (SI) for global optimization inspired by this process, which has received more and more attention in recent years [21,25,26,34]. The primary idea of BSO is to divide the population into different clusters, while new solutions are generated within a cluster or between two clusters. Therefore, the key steps affecting the performance of BSO are the solution clustering strategy and the solution mutation strategy.

k -means is the clustering strategy used in the classic BSO. However, the role of k -means for BSO is worth discussing. First, k -means is suitable for spherical clustering, and cannot deal with the clusters with arbitrary shapes. Second, the expensive time complexity of k -means affects the efficiency of BSO. Especially in the early stage of evolution, it is meaningless to cluster by k -means when the individuals are scattered. Third, for the problem with D -dimensional solution space and single objective space, k -means only con-

✉ Wenjian Luo
luowenjian@hit.edu.cn

Peilan Xu
xpl@mail.ustc.edu.cn

Xin Lin
iskcal@mail.ustc.edu.cn

Shi Cheng
cheng@snnu.edu.cn

Yuhui Shi
shiyh@sustech.edu.cn

¹ School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, China

² School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, Guangdong, China

³ School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

⁴ Shenzhen Key Lab of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

siders clustering the population in the solution space but not using the information of the objective space.

There has been a lot of work trying to improve the clustering strategy of BSO in different ways. First, researchers have tried to improve the time efficiency of BSO from two perspectives. On the one hand, different grouping algorithms were proposed to replace k -means, such as simple grouping method (SGM) [40], random grouping strategy (RGS) [4]. Besides, a class of objective space-based grouping algorithms [11,19,30] could greatly reduce the time cost, which was first used in BSO-OS (BSO in objective space) proposed by Shi [30]. On the other hand, some work tried to reduce the times of calling for k -means [3,5] or controlled the number of iterations of k -means [43]. Second, some researchers have considered the fitness and the position of solutions simultaneously and proposed new clustering strategies, such as fitness-guided clustering strategy [41], and the role-playing strategy [6]. Third, some well-known clustering strategies were also used in BSO, such as affinity propagation [8] and agglomerative hierarchical clustering [7].

The mutation strategy is also the focus of BSO. In the classic BSO, the Gaussian mutation is performed on the selected individual or combined individual, and a transfer function is used to update the step size of the mutation. However, one of the obvious defects of this mutation strategy is that the mutation step size is a fixed schema. It does not depend on the problem and the current population distribution.

The improvements of the mutation strategy mainly include three categories. The first is based on the mutation strategy of the classic BSO, such as the improvements of the update strategy of step size [4,10,24,42], the mutation strategy adaptive selection [38]. The second contains several new mutation strategies, such as modified BSO (MBSO) [40], advanced discussion mechanism-based BSO (ADMBSO) [35], BSO with learning strategy (BSOLS) [32], adaptive BSO with multiple strategies (AMBOS) [9], active learning BSO (ALBSO) [5]. The third is hybrid algorithms formed by BSO and existing algorithms, such as BSO with a chaotic operation (BSO-CO) [36], BSO with differential evolution [2], hybrid BSO and simulate annealing [16], and hybrid covariance matrix adaptive evolution strategy and global-best BSO [13].

In this paper, we propose a novel and efficient BSO, termed BSO20 (proposed in 2020). BSO20 improves the classic BSO from two aspects. First, regarding the clustering strategy, an efficient hybrid clustering strategy is designed. Second, in terms of the mutation strategy, a modified mutation strategy is proposed to improve the exploration efficiency. In summary, the main contributions of this paper are listed as follows.

1. We propose a hybrid clustering strategy, which combines nearest-better clustering (NBC) [27,28] and RGS. NBC is a competitive clustering strategy based on the objective

space and the solution space, which could better reveal the information of the landscape. Especially, NBC is suitable for clusters with arbitrary shapes. Moreover, RGS is used to cluster the scattered population in the early stage of the optimization, as well as poor individuals in the evolution process. By RGS, the hybrid clustering strategy could reduce the time cost of the algorithm and increase the diversity within a cluster.

2. We propose a modified mutation strategy for intra-cluster and inter-cluster mutation. The primary idea is to enhance the ability of individuals to share information and to improve the exploration efficiency of the population. In the intra-cluster, each selected individual shares information with one better individual, and in the inter-cluster, each selected individual shares information with two individuals from the different clusters.

BSO20 is tested on the problems of the 2017 IEEE Congress on Evolutionary Computation competition on real parameter numerical optimization (CEC'17 RPNO) and compared with several up-to-date BSOs as well as two variants of particle swarm optimization (PSO). The experimental results show that the performance of BSO20 is competitive.

The organization of this paper is as follows. The next section introduces the related work. The details of the proposed BSO20 algorithm are described in the following section. Next section demonstrates the experimental results and the parameter analysis. Following section discusses the CPU time and the convergence of BSO20. Finally, the last section concludes this paper and introduces the future work.

Related work

In this section, BSO and some of its clustering strategies are reviewed. Then, a competitive clustering strategy NBC is introduced.

Brain storm optimization

BSO was proposed by Shi [29], which is inspired by human being's brainstorming process. The core operations of an iteration of BSO include the following three parts.

1. Solution clustering: The cluster strategy is used for dividing the population, and NP individuals in the population are clustered into k clusters. In order to avoid the premature convergence of BSO, the replacing operator is used to control the initialization of a random cluster center with the probability p_{init} .
2. Solution generation: In the classic BSO, a new solution x_{new} is generated according to Formula (1), where y is

a base individual, d is the dimension index, $\mathcal{N}^d(0, 1)$ is a random number obeying standard normal distribution, and $\xi(t)$ is the step size at t -iteration updated according to Formula (2).

$$x_{\text{new}}^d = y^d + \xi^d(t) \cdot \mathcal{N}^d(0, 1), \quad (1)$$

$$\xi^d(t) = \text{logsig}\left(\frac{0.5 \times T - t}{20}\right) \cdot \text{rand}^d. \quad (2)$$

In Formula (2), T is the maximum number of iterations, t is the current number of iterations, rand^d is a random number between 0 to 1 following the uniform distribution, and the transfer function “ $\text{logsig}(\cdot)$ ” is described by Formula (3).

$$\text{logsig}(a) = \frac{1}{1 + \exp(-a)}. \quad (3)$$

There are two mutations in BSO, namely intra-cluster mutation and inter-cluster mutation, where the probability of the intra-cluster mutation is $p_{\text{one_cluster}}$, and the probability of the inter-cluster mutation is $(1 - p_{\text{one_cluster}})$. In the intra-cluster mutation, an individual is selected from a random cluster as the base individual y , where y is the center of the selected cluster with the probability $p_{\text{one_best}}$. In the inter-cluster mutation, the base individual y is generated according to Formula (4).

$$y = r \cdot x_{i_1} + (1 - r) \cdot x_{i_2}. \quad (4)$$

where r is a random number between 0 to 1, x_{i_1} and x_{i_2} are two different individuals coming from two randomly clusters, as well as x_{i_1} and x_{i_2} are the centers of the respective clusters with the probability $p_{\text{two_best}}$.

3. Solution selection: The new solution is compared with the old solution with the same index, and the better solution is retained in the population.

Clustering strategies in BSO

The solution clustering strategy is one of the primary steps of BSO, which has many classification methods. In this section, the existing clustering strategies are divided into four categories according to the difference of clustering space: the solution space-based clustering strategies represented by k -means [29], the objective space-based clustering strategies represented by BSO-OS [30], the clustering strategy based on both the solution space and the objective space, such as fitness-guided clustering [41], and other clustering strategies.

Solution space-based clustering strategies

The solution space-based clustering strategies are the mainstream clustering strategies in BSO, which is to divide the similar individuals into one cluster according to the distance.

k -means is a popular clustering strategy used in BSO, and its idea is as follows. First, k cluster centers are randomly generated in the solution space, and each individual is grouped to the nearest cluster center. Then, the new cluster centers are recalculated, and the population is grouped again around the new cluster centers. The above steps are repeated until clusters tend to be stable. However, k -means requires multiple iterations to update clusters, and the number of iterations is related to the choice of the initial cluster centers. In order to reduce the time cost brought by k -means in the classic BSO, efforts were made in two aspects. On the one hand, the number of calling for k -means in BSO is reduced, such as k -means is called with probability p_c in [3] and [5], where p_c is a constant value in [5], whereas is dynamically adjusted in [3]. On the other hand, the number of iterations of k -means is reduced. For example, in [43], the median is used in place of the mean in k -means, which stabilizes the cluster centers to reduce the number of iterations.

In addition to k -means, affinity propagation clustering [8] and agglomerative hierarchical clustering [7] have also been adopted in BSO. Compared with k -means, the time complexity $O(n^2 \log n)$ of these are high.

Moreover, some new clustering algorithms have been proposed for BSO. For example, Zhan et al. proposed SGM in [40]. SGM first randomly selects M different individuals from the population as cluster centers at first, and then each individual is grouped into the nearest cluster center.

Objective space-based clustering strategies

On the single objective optimization problems, the objective space-based clustering strategies group individuals according to fitness. In [30], Shi first proposed this strategy called BSO-OS. The solutions are divided into two clusters in BSO-OS. One cluster of individuals with better fitness is called elite solutions, and the other cluster is called normal solutions. There is no need to calculate the distance between individuals in BSO-OS, which greatly reduces the computational burden. Similarly, in [19], Li et al. proposed the BSO based on a competition mechanism, which designed two competing groups in the same way.

In [11], Mohammed proposed another objective space-based clustering strategy called fitness-based grouping, which ensured that the good and the poor individuals in the population are evenly distributed in each cluster.

Clustering strategy based on solution and objective space

There was a little work that considers clustering individuals in both the solution space and the objective space. In [41], Zhang et al. proposed the fitness-guided clustering strategy, and the clusters are formed as follows. First, all individuals in the population are labeled as unprocessed individuals. Then, the best individual x_b among unprocessed individuals is selected as the cluster center. Next, n individuals nearest to x_b are selected from unprocessed individuals and are grouped into a new cluster with x_b . These individuals are labeled as processed individuals. The above steps are repeated for unprocessed individuals until all individuals are processed.

In [6], Chen et al. proposed an enhanced BSO with a role-playing strategy (PRBSO). The solutions are grouped into three clusters according to the fitness difference and distance between all solutions and the best solution. Among them, good solutions closer to the best solution are called innovative ideas, good solutions far from the best solution are called conservative ideas, and others are called ordinary ideas.

Other clustering strategy

In [4], Cao et al. proposed a random grouping BSO (RGBSO), in which RGS is adopted as the clustering strategy. RGS randomly divides the population into k clusters of the same size. The best individual in each cluster is the cluster center. Compared with other clustering strategies, the time complexity of RGS is lowest, because there is no need to calculate distances and sort the population. Moreover, the diversity of each group is stronger in RGS. In [33], Wang et al. proposed to use orthogonal experiment design to improve the distribution of individuals in clusters.

Nearest-better clustering

Nearest-better clustering (NBC) [20,22,27,28] is an excellent clustering strategy based on the solution space and objective space for the multimodal optimization. NBC relies upon the assumption that the distance between the optimal solutions is greater than the weighted average distance of all individuals to their nearest better neighbors. The details of NBC are as follows.

First, the population P is sorted according to the fitness value from the superior to the poor, and the distance of each pair of individuals is calculated. Then, an empty tree is created, and the best individual becomes the root node. For the remaining individuals, each individual is connected to its nearest better neighbor, and the length of the edge is the distance between two individuals. In [20], the parent node is called the leader, and the child node is called a follower. Next, each edge in the tree is traversed, and all edges whose length of edge greater than the threshold ε are cut off. Generally, the

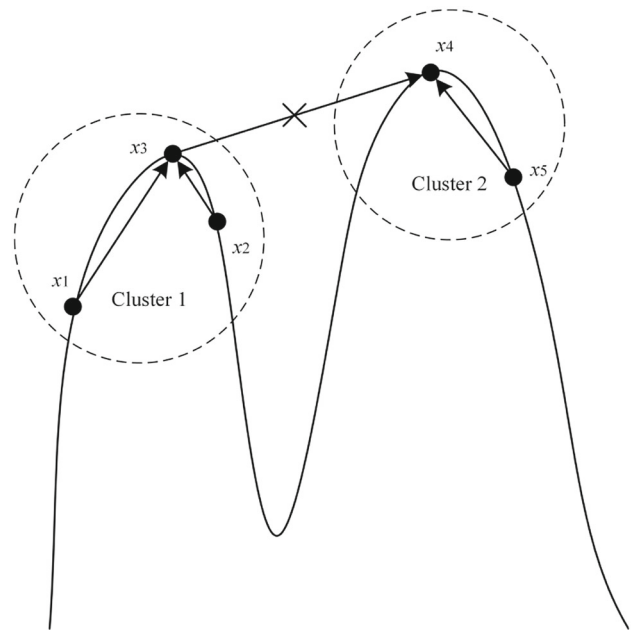


Fig. 1 The schematic of NBC

threshold ε is set to the weighted average of the length of all edges. Finally, each connected component forms a cluster.

Figure 1 illustrates an example of clustering the population $P = \{x_1, x_2, x_3, x_4, x_5\}$ using NBC. In the figure, each individual looks for its nearest better neighbor to form a tree. Then, the edge $e(x_3, x_4)$ is cut off according to the threshold ε , and finally, the population P is divided into two clusters.

The proposed algorithm: BSO20

Motivation

The core parts of the classic BSO are the clustering strategy and the mutation strategy. However, there are some deficiencies in the performance of clustering and exploration that need to be improved.

First, in the classic BSO, the population is clustered by k -means before each iteration, and the higher time complexity of k -means affects the efficiency of the algorithm. Moreover, k -means only makes use of the information of the solution space, whereas the information of the objective space is ignored. Although there have been some studies on the clustering strategy, the performance still could be improved. Consequently, there is a need to introduce a novel and efficient clustering strategy for BSO, and NBC is considered in this paper.

The steps of NBC include three parts: sorting with time complexity $O(n \log n)$, distance calculation with time complexity $O(n^2)$, and cluster partition with complexity $O(n)$. Therefore the time complexity of NBC is $O(n^2)$, where n

is the population size. NBC uses the distance information of the solution space and the fitness information of the objective space, which is promising for obtaining more knowledge about the landscape. However, Luo et al. pointed out that the “long-tail phenomenon” exists in NBC [23], which means that the NBC may be unreliable to deal with poor individuals in some cases. Therefore, in this paper, NBC is considered for clustering better individuals, while the clustering strategy RGS is considered for clustering poor individuals.

Second, for the mutation strategy of the classic BSO, the primary defect is that the update of the step size is independent of the problem and the state of the current population, which affects the exploration performance of BSO. For this reason, an improved mutation strategy is proposed, and the generation of the base individual is modified to improve the exploration efficiency.

In summary, a novel and efficient variant of BSO, named BSO20, is proposed in this paper. In the rest of this section, the clustering strategy and the mutation strategy of BSO20 are detailed, and the algorithm flow of BSO20 is described at the end of this section.

Clustering strategy of BSO20

A new hybrid clustering strategy combining NBC and RGS is used in BSO20. The primary idea is that NBC is used to cluster better individuals to exploit the potential of promising areas, while RGS is used to cluster poorer individuals to increase the diversity within the cluster, and to further reduce the time cost of BSO20. In the early stage of evolution, individuals are scattered randomly in the search space, and RGS is the main clustering strategy at this time. Then, NBC is used to cluster better individuals when the promising areas could be found, and the other individuals are clustered by RGS to explore the search space. As the population evolves, the number of individuals using NBC for clustering is gradually increased, and finally, all individuals are clustered by NBC.

The number and the size of clusters need to be considered when designing a clustering strategy because a smaller number of clusters could affect the performance of inter-cluster mutation, and a smaller size of clusters could affect the performance of intra-cluster mutation. In this paper, the size of clusters generated by RGS is set to S_r in BSO20, and the total number of clusters is set to k in the hybrid clustering strategy (Formula (5)).

$$k = \frac{NP}{S_r} \quad (5)$$

where NP is the population size.

During the population evolution, the number of clusters generated by RGS decreases linearly with the number of iterations (Formula (6)).

$$k_r = \left\lceil k \cdot \left(1 - \frac{t}{T}\right) \right\rceil \quad (6)$$

where T is the maximum number of iterations, t is the current number of iterations.

Therefore, the number of individuals clustered by NBC can be calculated as Formula (7).

$$N_{nbc} = NP - k_r \cdot S_r \quad (7)$$

In order to ensure the total number of clusters is k , the number of clusters formed by NBC, i.e., k_n , should be set to $k - k_r$. Therefore, the partition condition of NBC should be modified, that is, the longest $k_n - 1$ edges are cut off to generate k_n clusters.

Overall, the pseudocode of the clustering strategy of BSO20 is given as Algorithm 1. Among them, lines 3–7 describe the clustering process of NBC, and lines 9–12 describe the clustering process of RGS.

Algorithm 1 Clustering Strategy of BSO20

```

1: Record the index of the all individuals as  $\Lambda$ ;
2: Calculate the number of clusters  $k_r$  and  $k_n$ ;
3: if  $k_n > 0$  then
4:   Calculate  $N_{nbc}$  according to Formula (7);
5:   Select the best  $N_{nbc}$  individuals from the population  $P$  and form  $k_n$  clusters by NBC;
6:   Remove the index of best  $N_{nbc}$  individuals from  $\Lambda$ ;
7: end if
8: Shuffle the index of remaining individuals  $\Lambda$ ;
9: for  $j \leftarrow k_n + 1$  to  $k$  do
10:   Form  $j$ -th cluster with the top  $S_r$  index in  $\Lambda$ ;
11:   Remove the top  $S_r$  index from  $\Lambda$ ;
12: end for

```

Mutation strategy of BSO20

The mutation strategy of BSO20 is an improved version based on the classic BSO. The main idea of the modified mutation strategy is to promote the efficiency of information exchange between individuals within a cluster or among multiple clusters to improve the exploration performance of BSO.

Before describing the modified mutation strategy, a concept of the leader set needs to be briefly introduced. In the clusters formed by NBC, the leader set of an individual x includes its ancestor nodes. In the clusters formed by RGS, the leader set of an individual x includes the individuals

which are better than x . Particularly, if the leader set of an individual x is empty, then the leader of x is itself.

In the intra-cluster mutation, each selected individual shares information with its random leader to generate a base individual y (Formula (8)). In the inter-cluster mutation, each selected individual shares information with two individuals from two different clusters (Formula (9)).

$$y^d = (1 - r) \cdot x_s^d + r \cdot \text{leader}_s^d, \quad (8)$$

$$y^d = (1 - r_1 - r_2) \cdot x_s^d + r_1 \cdot x_{i_1}^d + r_2 \cdot x_{i_2}^d. \quad (9)$$

where r , r_1 and r_2 are random numbers uniformly distributed between 0 and 1, x_s is an individual randomly selected from the population, leader_s is a leader individual randomly selected from the leader set of x_s , x_{i_1} and x_{i_2} are randomly selected from two randomly selected clusters, and d denotes the dimension index.

The mutation of the base individual y is the same as that in the classic BSO, except that $\mathcal{N}_b^d(0, 1)$ is used instead $\mathcal{N}^d(0, 1)$ in BSO20 (Formula (10)). The random numbers generated by Gaussian distribution are limited between lb and ub .

$$x_{\text{new}}^d = y^d + \xi^d(t) \cdot \mathcal{N}_b^d(0, 1),$$

$$\mathcal{N}_b^d(0, 1) = \begin{cases} ub, & \text{if } \mathcal{N}^d(0, 1) > ub, \\ lb, & \text{if } \mathcal{N}^d(0, 1) < lb, \\ \mathcal{N}^d(0, 1), & \text{otherwise,} \end{cases} \quad (10)$$

where ub and lb are the upper bound and lower bound of the search space, respectively, and $\xi^d(t)$ is updated as Formulas (2) and (3).

BSO20

In the above parts, the clustering strategy and the mutation strategy of BSO20 are introduced. Algorithm 2 shows the pseudocode of BSO20, and the details are described as follows.

First, after the population and parameters are initialized (i.e., lines 1–2), the population P is clustered by the hybrid clustering strategy, and the leader set of each individual is obtained (i.e., lines 4–5).

Then, from line 7 to line 16, an individual x_s is randomly selected from the population to generate a base individual y . The intra-cluster mutation is performed with the probability $p_{\text{one_cluster}}$, and the inter-cluster mutation is performed with $1 - p_{\text{one_cluster}}$.

Finally, from line 17 to line 22, a new individual x_{new} is obtained from the base individual y by Gaussian mutation, and the population is updated.

In BSO20, the replacing operator is not considered, due to its contributing less to the algorithm, and this was verified by

Algorithm 2 BSO20

```

1: Initialize parameters;
2: Initialize and evaluate the population  $P$ ;
3: while the terminal condition is not satisfied do
4:   Cluster population  $P$  by Algorithm 1;
5:   Get the leader set of each individual;
6:   for  $i \leftarrow 1$  to  $NP$  do
7:     Randomly select  $x_s$  from population  $P$ ;
8:     if  $\text{rand} < p_{\text{one\_cluster}}$  then
9:       Randomly select  $\text{leader}_s$  from leader set of  $x_s$ ;
10:      Generate the base individual  $y$  using Formula (8);
11:    else
12:      Randomly select two clusters  $C_{i_1}$  and  $C_{i_2}$ ;
13:      Randomly select  $x_{i_1}$  from  $C_{i_1}$ ;
14:      Randomly select  $x_{i_2}$  from  $C_{i_2}$ ;
15:      Generate base individual  $y$  using Formula (9);
16:    end if
17:    Calculate the update step size  $\xi$ ;
18:    Generate the new individual  $x_{\text{new}}$  using Formula (10);
19:    Evaluate the new individual  $x_{\text{new}}$ ;
20:    if  $x_{\text{new}}$  is better than  $x_i$  then
21:      Update individual  $x_i$  by  $x_{\text{new}}$ ;
22:    end if
23:  end for
24: end while

```

the experiment in [39]. Moreover, the selection probabilities of cluster center $p_{\text{one_best}}$ and $p_{\text{two_best}}$ are not considered. In the intra-cluster mutation, the selected individual moves towards its leader, so there is a chance to explore near the cluster center. In the inter-cluster mutation, the selected individual shares information within two clusters. BSO20 only retains one parameter of the classic BSO, i.e., the probability of the intra-cluster mutation $p_{\text{one_cluster}}$, and adds one parameter S_r , i.e., the size of clusters generated by RGS, which reduces the hassle of tuning many parameters. The source codes of BSO20 are available at <https://github.com/squirreler-xu/BSO20>.

Experiments

In this section, BSO20 is tested on the CEC'17 RPNO, and the parameters of BSO20 are analyzed at the end of the section.

Experimental setting

In this paper, 29 global optimization problems in CEC'17 RPNO [1] are used as the benchmark, which includes four types of functions: unimodal functions (F_1 , F_3), simple multimodal functions (F_4 – F_{10}), hybrid functions (F_{11} – F_{20}), and composition functions (F_{21} – F_{30}). The problem F_2 has been excluded according to the recommendation of [1].

The dimensions D of the benchmark include $30D$, $50D$, and $100D$. For each function, this benchmark suggests that the algorithm runs 51 times independently to calculate the

mean as well as the standard deviation of 51 results [1]. The maximum number of evaluations for one run is $10^4 \times D$.

In order to verify the performance of BSO20, the classic BSO [29] and several variants are used as compared algorithms in this experiment, namely BSO-OS [30], RGBSO [4], BSO with chaotic local search (CBSO) [37], active learning BSO (ALBSO) [5], and bare-bones global-best BSO (BBGBO) [12]. BSO-NBC is also adopted for comparisons, which is similar to the classic BSO, except that NBC is used as a clustering strategy instead of k -means. The clustering strategies of the classic BSO and BSO-OS are based on the solution space and objective space, respectively. The clustering strategy of RGBSO is RGS. Therefore, the above four algorithms are used to compare the effect of the clustering strategy on the performance of the proposed algorithm. BSO20 is also compared with the latest variants of BSO, such as CBSO, ALBSO, and BBGBO, to demonstrate its competitiveness. Moreover, two variants of particle swarm optimization (PSO), i.e., scout particle swarm optimization (ScPSO) [17] and phasor particle swarm optimization (PPSO) [15], are also added as compared algorithms for comparison experiments.

BSO20 only includes two parameters, which are set as follows. The size of the cluster formed by RGS S_r is set to 20, and the probability of the intra-cluster mutation $p_{\text{one_cluster}}$ is set to 0.1. In contrast, the compared algorithms have more parameters than BSO20. For example, the classic BSO has 5 main parameters, namely the number of clusters k , the probability of replacing operator p_{init} , the intra-cluster mutation probability $p_{\text{one_cluster}}$, as well as the cluster center selection probabilities $p_{\text{one_best}}$ and $p_{\text{two_best}}$. In addition, the number of main parameters of BSO-OS, RGBSO, CBSO, ALBSO, and BBGBO are 3, 4, 5, 5, and 8, respectively. The parameters of the above compared algorithms all adopt the parameter settings recommended in the original literature [4,5,12,29,30,37]. Moreover, the population size of all algorithms is set to $4D$.

For each algorithm, the mean and standard deviation of each function in 51 runs are counted as experimental results. Moreover, the Wilcoxon signed-rank test is used for verifying significant differences between the statistical results of BSO20 and the compared algorithms.

Experimental results

The experimental results on $30D$, $50D$, $100D$ functions are listed in Tables 1, 2, 3, 4, 5, 6, and In each line, the bolded term is the best mean or standard deviation of the fitness values found by algorithms. In the last row of Tables 2, 4, and 6, the comparison results of BSO20 on different dimensions with each compared algorithm are given, where the symbol “+/-” indicates that BSO20 is better/worse than the compared algorithms.

The statistical results show that the performance of BSO20 remains stable on different dimensions. Comparing BSO20 with the classic BSO, there are 23, 24, 25 functions better than the classic BSO on $30D$, $50D$, and $100D$ problems, respectively, which can also be seen at the bottom of the second column of Tables 2, 4, and 6. The experimental results indicate that the performance of the classic BSO decreases as the dimension increases, while BSO20 can maintain better performance. Similarly, the comparison results of “+/-” between BSO20 and BSO-OS are “25/4,” “25/4,” and “26/3,” which can also be seen at the bottom of the third column of Tables 2, 4, and 6.

Then, from the results of RGBSO and BSO-NBC, it can be seen that RGS is competitive on higher dimensional problems, and NBC is also more valuable than k -means for BSO. Furthermore, as shown at the bottom of the fourth and fifth columns of Tables 2, 4, and 6, BSO20 achieves better results in different dimensions, outperforming RGBSO and BSO-NBC on more than 20 functions, which demonstrates the superiority of the hybrid clustering strategy.

From the sixth columns to the eighth columns of Tables 1–6, comparisons between BSO20 against the state-of-the-art variants of BSO show that BSO20 is the best performer overall. BSO20 has great superiority when comparing with CBSO and ALBSO. In particular, it outperforms ALBSO on all functions. Moreover, compared with BBGBO, BSO20 also performed well on $30D$ problems, but as the dimension increases, the competitiveness of BSO20 gradually decreases, especially on the composition functions.

Finally, the last two columns of the Tables 1, 2, 3, 4, 5, 6 list the results of the performance comparison between BSO20 and two variants of PSO. The experimental results indicate that the performance of BSO20 is inferior to ScPSO on $30D$ problems, but is better than ScPSO on $100D$ problems. Moreover, compared with PPSO, BSO20 performs better on different dimensional problems, and the comparison results of “+/-” between BSO20 and PPSO are “19/10,” “18/11,” and “17/12.”

BSO20 performs better on the multimodal problems than the unimodal problems, which is determined by the feature of BSO. Furthermore, BSO20 also performs well on the unimodal problems, such as F_1 . On the multimodal problems, the modified mutation strategy improves the exploration ability of BSO20 and enhances the chance of BSO20 to find better solutions. Moreover, on the hybrid functions, BSO20 achieves better results on most functions, while it has general performance on the composition functions.

Wilcoxon signed-rank test

In this section, Wilcoxon signed-rank test is carried to further analyze the experimental results. The results of the Wilcoxon signed-rank test between BSO20 and all compared algo-

Table 1 Experimental results on 30- D functions F_1, F_3-F_{16}

Function		BSO20	Classic BSO	BSO-OS	RGBSO	BSO-NBC	CBSO	ALBSO	BGBSO	ScPSO	PPSO
F1	Mean	1.84E+03	2.65E+03	2.59E+03	2.39E+03	2.07E+03	2.83E+03	3.76E+10	2.30E+04	5.40E+03	1.42E+03
	Std.	1.15E+03	2.39E+03	2.65E+03	2.37E+03	1.41E+03	2.47E+03	8.52E+09	7.11E+03	6.48E+03	1.78E+03
F3	Mean	1.63E+04	2.41E+01	1.33E+04	1.49E+01	5.80E+00	8.46E+02	8.05E+04	6.02E+04	1.71E+04	9.21E+03
	Std.	5.14E+03	6.59E+01	4.77E+03	1.05E+02	3.63E+00	9.21E+02	1.20E+04	7.56E+03	4.12E+03	1.39E+02
F4	Mean	1.03E+02	8.90E+01	7.28E+01	7.89E+01	9.54E+01	4.88E+02	8.55E+03	9.46E+01	8.78E+01	8.13E+01
	Std.	1.64E+01	2.81E+01	1.50E+01	2.60E+01	2.36E+01	2.22E+01	2.78E+03	1.83E+01	1.36E+01	2.28E+01
F5	Mean	7.24E+01	1.79E+02	1.75E+02	1.99E+02	1.60E+02	6.88E+02	3.21E+02	1.35E+02	7.68E+01	1.87E+02
	Std.	1.12E+01	2.95E+01	2.70E+01	4.35E+01	2.15E+01	3.10E+01	3.37E+01	4.31E+01	2.09E+01	3.71E+01
F6	Mean	7.34E+00	5.23E+01	4.96E+01	5.34E+01	4.69E+01	6.53E+02	6.93E+01	9.41E+00	2.16E+01	3.13E+01
	Std.	2.84E+00	6.91E+00	6.42E+00	7.83E+00	4.97E+00	7.03E+00	8.66E+00	6.26E+00	2.62E+01	1.22E+01
F7	Mean	9.88E+01	4.85E+02	4.05E+02	5.25E+02	3.04E+02	1.17E+03	5.61E+02	1.27E+02	1.27E+02	3.96E+02
	Std.	1.83E+01	9.60E+01	6.95E+01	8.00E+01	5.76E+01	1.00E+02	7.04E+01	2.31E+01	2.65E+01	6.78E+01
F8	Mean	5.27E+01	1.34E+02	1.25E+02	1.54E+02	1.10E+02	9.35E+02	2.67E+02	1.27E+02	7.74E+01	1.49E+02
	Std.	1.10E+01	2.57E+01	2.45E+01	4.04E+01	1.44E+01	2.60E+01	2.03E+01	2.91E+01	1.84E+01	2.97E+01
F9	Mean	4.13E+02	3.21E+03	2.84E+03	3.60E+03	2.46E+03	4.02E+03	6.59E+03	1.15E+02	2.99E+02	3.21E+03
	Std.	1.59E+02	7.62E+02	5.71E+02	8.84E+02	4.41E+02	7.49E+02	1.63E+03	3.01E+02	3.49E+02	1.03E+03
F10	Mean	3.72E+03	4.26E+03	4.23E+03	4.56E+03	3.75E+03	5.19E+03	7.11E+03	3.53E+03	3.40E+03	4.04E+03
	Std.	7.01E+02	4.78E+02	6.15E+02	6.80E+02	5.82E+02	5.41E+02	5.19E+02	5.54E+02	6.92E+02	8.46E+02
F11	Mean	1.00E+02	1.36E+02	1.28E+02	1.45E+02	1.10E+02	1.24E+03	6.41E+03	1.47E+02	8.51E+01	1.18E+02
	Std.	3.22E+01	5.39E+01	4.87E+01	5.52E+01	3.57E+01	5.23E+01	2.40E+03	4.31E+01	3.91E+01	4.04E+01
F12	Mean	1.64E+06	1.65E+06	1.87E+06	8.96E+05	1.52E+06	1.92E+06	5.09E+09	2.45E+06	8.57E+05	6.01E+05
	Std.	5.57E+05	7.69E+05	1.06E+06	5.22E+05	6.90E+05	8.62E+05	2.15E+09	1.59E+06	7.84E+05	3.14E+06
F13	Mean	2.70E+04	5.53E+04	5.02E+04	6.50E+04	4.10E+04	4.88E+04	1.69E+09	7.97E+05	1.55E+04	1.78E+05
	Std.	6.98E+03	3.92E+04	2.88E+04	3.68E+04	1.95E+04	3.01E+04	1.33E+09	4.76E+06	1.78E+04	8.71E+05
F14	Mean	4.28E+03	4.04E+03	7.17E+03	3.55E+03	2.50E+03	7.45E+03	4.02E+05	3.20E+04	2.47E+04	1.26E+03
	Std.	3.11E+03	3.04E+03	7.20E+03	3.13E+03	2.04E+03	4.43E+03	6.90E+05	2.70E+04	2.29E+04	5.23E+03
F15	Mean	6.87E+03	2.68E+04	2.60E+04	3.46E+04	1.59E+04	2.10E+04	5.28E+07	3.75E+04	1.65E+04	1.36E+03
	Std.	3.54E+03	1.90E+04	1.83E+04	1.88E+04	6.46E+03	1.35E+04	7.68E+07	2.97E+04	1.13E+04	6.31E+03
F16	Mean	8.81E+02	1.45E+03	1.40E+03	1.47E+03	1.12E+03	3.04E+03	2.23E+03	1.09E+03	5.75E+02	1.16E+03
	Std.	2.39E+02	3.43E+02	3.08E+02	4.77E+02	2.96E+02	3.19E+02	3.97E+02	3.57E+02	2.29E+02	3.42E+02

Table 2 Experimental results on 30-D functions F_{17} – F_{30}

Function		BSO20	Classic BSO	BSO-OS	RGBSO	BSO-NBC	CBSO	ALBSO	BBGSO	ScPSO	PPSO
F17	Mean	2.67E+02	7.44E+02	7.86E+02	8.37E+02	5.20E+02	2.39E+03	8.64E+02	5.26E+02	1.88E+02	5.80E+02
	Std.	1.41E+02	2.40E+02	2.73E+02	2.85E+02	1.63E+02	2.61E+02	2.49E+02	2.66E+02	1.10E+02	1.83E+02
F18	Mean	1.96E+05	1.14E+05	1.16E+05	1.08E+05	9.29E+04	1.18E+05	3.24E+06	2.85E+05	4.12E+05	2.58E+04
	Std.	1.09E+05	8.14E+04	1.02E+05	8.85E+04	4.73E+04	8.47E+04	4.48E+06	2.25E+05	3.08E+05	4.66E+04
F19	Mean	1.46E+04	1.36E+05	1.23E+05	5.70E+04	1.26E+05	5.23E+04	1.08E+08	3.91E+05	1.77E+04	2.09E+02
	Std.	1.88E+04	5.71E+04	7.45E+04	2.82E+04	4.90E+04	4.40E+04	1.49E+08	2.67E+05	1.61E+04	2.94E+02
F20	Mean	2.77E+02	6.67E+02	7.18E+02	7.91E+02	3.66E+02	2.69E+03	6.70E+02	4.87E+02	2.17E+02	4.04E+02
	Std.	1.00E+02	1.80E+02	2.23E+02	2.20E+02	1.19E+02	1.77E+02	1.64E+02	1.73E+02	1.47E+02	1.54E+02
F21	Mean	2.62E+02	3.97E+02	3.78E+02	3.99E+02	3.76E+02	2.50E+03	4.97E+02	3.05E+02	2.81E+02	3.76E+02
	Std.	1.17E+01	3.46E+01	3.37E+01	3.97E+01	3.40E+01	3.63E+01	3.35E+01	2.76E+01	1.91E+01	4.06E+01
F22	Mean	1.00E+02	4.14E+03	3.34E+03	3.49E+03	2.68E+03	5.80E+03	5.72E+03	1.01E+02	1.27E+03	1.86E+03
	Std.	1.11E-13	1.79E+03	2.03E+03	2.27E+03	1.90E+03	1.86E+03	1.18E+03	6.81E-01	1.66E+03	2.15E+03
F23	Mean	5.02E+02	9.84E+02	9.67E+02	9.84E+02	9.46E+02	3.26E+03	8.41E+02	4.76E+02	4.21E+02	6.18E+02
	Std.	2.63E+01	1.19E+02	1.22E+02	1.17E+02	1.13E+02	1.00E+02	9.17E+01	6.14E+01	1.67E+01	6.87E+01
F24	Mean	5.90E+02	1.09E+03	1.11E+03	1.18E+03	1.10E+03	3.23E+03	8.61E+02	5.25E+02	4.94E+02	7.29E+02
	Std.	3.46E+01	1.11E+02	1.07E+02	1.31E+02	1.01E+02	1.39E+02	8.15E+01	3.19E+01	1.95E+01	9.52E+01
F25	Mean	3.90E+02	3.88E+02	3.86E+02	3.89E+02	3.88E+02	2.89E+03	2.13E+03	3.92E+02	3.88E+02	4.04E+02
	Std.	8.83E+00	4.28E+00	9.36E+00	8.98E+00	9.13E+00	6.28E+00	5.93E+02	1.14E+01	5.48E+00	2.21E+01
F26	Mean	3.20E+03	5.20E+03	5.40E+03	5.05E+03	3.97E+03	7.79E+03	6.46E+03	2.83E+02	1.88E+03	3.69E+03
	Std.	1.33E+03	1.73E+03	1.23E+03	2.07E+03	2.14E+03	1.73E+03	1.02E+03	3.76E+01	1.82E+02	1.34E+03
F27	Mean	6.49E+02	1.21E+03	1.14E+03	1.24E+03	1.23E+03	3.69E+03	9.42E+02	5.29E+02	5.10E+02	6.00E+02
	Std.	2.79E+01	2.37E+02	2.62E+02	2.68E+02	1.90E+02	1.81E+02	1.45E+02	2.19E+01	1.03E+01	8.10E+01
F28	Mean	4.03E+02	3.73E+02	4.08E+02	3.36E+02	3.66E+02	3.20E+03	3.20E+03	4.13E+02	4.36E+02	4.05E+02
	Std.	4.99E+00	4.88E+01	1.94E+01	4.81E+01	4.84E+01	3.26E+01	6.69E+02	2.04E+01	2.53E+01	5.24E+01
F29	Mean	8.21E+02	1.52E+03	1.45E+03	1.65E+03	1.29E+03	4.41E+03	2.48E+03	1.03E+03	6.60E+02	1.06E+03
	Std.	1.76E+02	3.26E+02	3.01E+02	3.13E+02	2.90E+02	3.50E+02	6.36E+02	2.50E+02	1.62E+02	2.52E+02
F30	Mean	1.21E+05	4.91E+05	6.68E+05	2.01E+05	4.92E+05	6.24E+05	1.53E+08	1.55E+06	9.49E+03	9.38E+04
	Std.	7.21E+04	2.71E+05	4.71E+05	1.46E+05	2.83E+05	3.96E+05	1.10E+08	1.07E+06	4.02E+03	5.93E+05
“+/-”			23/6	25/4	23/6	22/7	27/2	29/0	22/7	12/17	19/10

Table 3 Experimental results on 50-*D* functions F_1, F_3-F_{16}

Function		BSO20	Classic BSO	BSO-OS	RGBSO	BSO-NBC	CBSO	ALBSO	BGBSO	ScPSO	PPSO
F1	Mean	7.83E+02	1.65E+03	2.11E+03	1.74E+03	1.24E+03	1.77E+03	7.92E+10	5.89E+04	1.39E+04	3.78E+03
	Std.	8.77E+02	2.30E+03	2.18E+03	2.17E+03	1.72E+03	2.47E+03	8.90E+09	1.33E+04	9.68E+03	4.54E+03
F3	Mean	5.15E+04	5.47E+03	7.69E+04	3.32E+03	1.12E+04	1.77E+04	1.91E+05	1.63E+05	7.86E+04	7.39E+00
	Std.	7.71E+03	5.51E+03	1.44E+04	2.73E+03	3.27E+03	4.90E+03	3.20E+04	1.68E+04	1.24E+04	4.10E+00
F4	Mean	1.50E+02	1.62E+02	1.60E+02	1.56E+02	1.60E+02	5.69E+02	2.23E+04	1.45E+02	1.75E+02	1.06E+02
	Std.	3.16E+01	5.24E+01	3.94E+01	5.54E+01	4.64E+01	4.92E+01	4.85E+03	4.65E+01	3.80E+01	5.56E+01
F5	Mean	1.35E+02	3.03E+02	2.76E+02	3.21E+02	2.53E+02	8.11E+02	5.84E+02	2.74E+02	1.65E+02	3.29E+02
	Std.	1.58E+01	4.10E+01	4.09E+01	4.04E+01	2.09E+01	4.42E+01	3.36E+01	4.60E+01	3.38E+01	5.22E+01
F6	Mean	1.64E+01	5.67E+01	5.20E+01	5.77E+01	5.18E+01	6.57E+02	8.27E+01	2.33E+01	2.65E+00	4.00E+01
	Std.	2.89E+00	4.94E+00	4.12E+00	4.53E+00	3.15E+00	5.20E+00	6.98E+00	8.41E+00	1.64E+00	9.62E+00
F7	Mean	2.13E+02	9.65E+02	7.23E+02	1.06E+03	6.34E+02	1.65E+03	1.11E+03	2.60E+02	3.11E+02	7.81E+02
	Std.	2.45E+01	1.66E+02	9.60E+01	1.38E+02	7.03E+01	1.44E+02	9.27E+01	4.47E+01	6.05E+01	1.05E+02
F8	Mean	1.42E+02	3.08E+02	2.79E+02	3.38E+02	2.63E+02	1.12E+03	5.86E+02	2.47E+02	1.70E+02	3.50E+02
	Std.	1.38E+01	3.93E+01	3.15E+01	4.83E+01	2.78E+01	3.84E+01	3.81E+01	5.60E+01	3.58E+01	4.81E+01
F9	Mean	2.75E+03	9.41E+03	8.03E+03	1.02E+04	9.14E+03	1.04E+04	2.62E+04	1.65E+03	1.89E+03	1.03E+04
	Std.	4.18E+02	1.37E+03	9.14E+02	1.38E+03	1.04E+03	1.51E+03	4.04E+03	1.79E+03	1.53E+03	2.05E+03
F10	Mean	6.27E+03	7.30E+03	7.06E+03	7.61E+03	6.31E+03	8.24E+03	1.32E+04	5.49E+03	6.61E+03	6.73E+03
	Std.	1.10E+03	6.98E+02	7.73E+02	8.34E+02	8.77E+02	7.85E+02	7.39E+02	8.96E+02	1.37E+03	9.18E+02
F11	Mean	1.65E+02	2.05E+02	2.25E+02	1.87E+02	1.58E+02	1.31E+03	2.04E+04	3.20E+02	2.23E+02	2.11E+02
	Std.	1.92E+01	4.07E+01	3.64E+01	5.34E+01	2.87E+01	4.18E+01	4.66E+03	5.63E+01	7.58E+01	6.02E+01
F12	Mean	6.96E+06	1.17E+07	1.19E+07	5.04E+06	8.31E+06	1.30E+07	3.20E+10	1.69E+07	7.00E+06	2.12E+07
	Std.	2.48E+06	6.64E+06	8.23E+06	2.67E+06	4.33E+06	7.99E+06	9.58E+09	1.09E+07	3.89E+06	1.35E+08
F13	Mean	2.50E+04	5.65E+04	4.98E+04	5.81E+04	4.80E+04	5.51E+04	1.12E+10	1.36E+05	1.34E+04	1.09E+04
	Std.	7.73E+03	2.96E+04	2.24E+04	3.30E+04	2.44E+04	2.63E+04	6.05E+09	7.65E+04	1.16E+04	8.92E+03
F14	Mean	4.86E+04	3.11E+04	3.60E+04	2.38E+04	3.34E+04	4.04E+04	4.76E+06	1.02E+05	1.22E+05	2.37E+04
	Std.	2.52E+04	2.36E+04	2.48E+04	1.95E+04	2.18E+04	2.85E+04	5.64E+06	8.40E+04	8.12E+04	1.38E+05
F15	Mean	1.57E+04	2.34E+04	2.34E+04	2.84E+04	1.91E+04	2.35E+04	2.34E+09	5.51E+04	9.53E+03	2.75E+03
	Std.	3.84E+03	1.31E+04	1.53E+04	1.92E+04	9.30E+03	1.13E+04	1.81E+09	2.69E+04	8.94E+03	4.43E+03
F16	Mean	1.01E+03	2.14E+03	2.13E+03	2.32E+03	1.64E+03	3.85E+03	4.22E+03	1.59E+03	1.46E+03	2.08E+03
	Std.	2.78E+02	5.33E+02	4.86E+02	5.56E+02	4.29E+02	5.40E+02	5.95E+02	5.43E+02	4.57E+02	4.71E+02

Table 4 Experimental results on 50-D functions F_{17} – F_{30}

Function		BSO20	Classic BSO	BSO-OS	RGBSO	BSO-NBC	CBSO	ALBSO	BBGBSO	ScPSO	PPSO
F17	Mean	1.08E+03	1.84E+03	1.67E+03	1.95E+03	1.25E+03	3.48E+03	2.67E+03	1.22E+03	1.13E+03	1.57E+03
	Std.	2.92E+02	3.60E+02	3.50E+02	3.83E+02	2.28E+02	3.72E+02	5.87E+02	3.14E+02	2.86E+02	3.45E+02
F18	Mean	1.14E+06	3.16E+05	4.25E+05	1.35E+05	2.83E+05	3.95E+05	4.15E+07	7.51E+05	2.27E+06	1.34E+05
	Std.	2.94E+05	1.26E+05	2.08E+05	6.11E+04	7.98E+04	1.69E+05	3.51E+07	6.53E+05	1.97E+06	6.03E+05
F19	Mean	6.05E+04	4.41E+05	3.56E+05	1.85E+05	4.13E+05	2.33E+05	6.63E+08	7.74E+05	9.54E+03	2.28E+03
	Std.	4.04E+04	2.34E+05	1.89E+05	6.03E+04	2.13E+05	2.09E+05	5.25E+08	4.47E+05	1.27E+04	3.61E+03
F20	Mean	9.09E+02	1.42E+03	1.39E+03	1.63E+03	7.37E+02	3.45E+03	1.63E+03	1.07E+03	8.22E+02	1.02E+03
	Std.	2.32E+02	3.32E+02	3.39E+02	3.91E+02	1.78E+02	3.03E+02	3.12E+02	3.05E+02	2.56E+02	2.82E+02
F21	Mean	3.21E+02	5.99E+02	5.79E+02	6.24E+02	5.66E+02	2.70E+03	8.34E+02	3.98E+02	3.70E+02	5.76E+02
	Std.	1.47E+01	7.16E+01	5.80E+01	6.37E+01	5.15E+01	7.31E+01	5.69E+01	3.71E+01	3.45E+01	5.99E+01
F22	Mean	7.86E+02	8.04E+03	7.61E+03	8.21E+03	7.43E+03	1.01E+04	1.37E+04	1.04E+02	7.69E+03	7.30E+03
	Std.	1.94E+03	9.07E+02	8.90E+02	7.96E+02	9.05E+02	7.79E+02	6.37E+02	6.86E+00	1.58E+03	1.41E+03
F23	Mean	7.34E+02	1.61E+03	1.59E+03	1.61E+03	1.56E+03	3.90E+03	1.46E+03	6.68E+02	5.89E+02	1.06E+03
	Std.	3.28E+01	1.60E+02	1.76E+02	1.72E+02	1.56E+02	1.79E+02	1.27E+02	9.34E+01	3.16E+01	1.41E+02
F24	Mean	9.37E+02	1.68E+03	1.69E+03	1.78E+03	1.67E+03	3.85E+03	1.44E+03	7.25E+02	6.36E+02	1.16E+03
	Std.	5.56E+01	1.78E+02	1.36E+02	1.82E+02	2.10E+02	2.99E+02	1.25E+02	7.71E+01	3.41E+01	1.39E+02
F25	Mean	5.91E+02	5.68E+02	5.02E+02	5.59E+02	5.79E+02	3.07E+03	1.03E+04	5.22E+02	5.43E+02	5.42E+02
	Std.	1.32E+01	3.52E+01	4.36E+01	3.42E+01	2.46E+01	2.95E+01	1.51E+03	3.03E+01	2.77E+01	4.06E+01
F26	Mean	6.69E+03	1.04E+04	9.42E+03	1.05E+04	8.97E+03	1.30E+04	1.32E+04	3.03E+02	2.86E+03	6.95E+03
	Std.	4.76E+02	9.30E+02	1.91E+03	1.15E+03	1.46E+03	9.44E+02	9.93E+02	9.73E+00	3.57E+02	2.67E+03
F27	Mean	1.24E+03	2.53E+03	2.49E+03	2.52E+03	2.53E+03	5.20E+03	2.20E+03	6.42E+02	5.97E+02	1.04E+03
	Std.	8.08E+01	5.00E+02	4.72E+02	5.57E+02	4.51E+02	5.23E+02	4.37E+02	1.02E+02	3.56E+01	2.14E+02
F28	Mean	5.31E+02	5.13E+02	4.97E+02	5.08E+02	5.09E+02	3.31E+03	7.73E+03	4.78E+02	4.99E+02	5.25E+02
	Std.	2.22E+01	2.31E+01	1.64E+01	1.72E+01	2.27E+01	2.09E+01	1.02E+03	2.41E+01	2.51E+01	9.55E+01
F29	Mean	1.52E+03	2.85E+03	2.74E+03	2.73E+03	2.54E+03	5.70E+03	7.97E+03	1.71E+03	9.70E+02	2.02E+03
	Std.	2.79E+02	5.19E+02	4.18E+02	4.53E+02	5.18E+02	4.68E+02	2.47E+03	6.00E+02	2.67E+02	4.26E+02
F30	Mean	1.39E+07	1.89E+07	2.60E+07	1.09E+07	1.77E+07	2.08E+07	1.28E+09	2.75E+07	1.32E+06	1.64E+06
	Std.	1.79E+06	2.81E+06	2.70E+06	1.69E+06	1.55E+06	2.62E+06	6.06E+08	5.64E+06	4.07E+05	1.07E+06
“+/-”			24/5	25/4	23/6	22/7	26/3	29/0	18/11	15/14	18/11

Table 5 Experimental results on 100- D functions F_1, F_3-F_{16}

Function		BSO20	Classic BSO	BSO-OS	RGBSO	BSO-NBC	CBSO	ALBSO	BGBSO	ScPSO	PPSO
F1	Mean	3.92E+03	8.87E+04	8.13E+04	4.73E+03	3.51E+04	4.44E+04	2.04E+11	1.64E+07	1.06E+07	8.42E+03
	Std.	1.46E+03	7.06E+04	2.42E+04	4.90E+03	1.40E+04	1.87E+04	1.54E+10	7.44E+07	4.85E+06	9.69E+03
F3	Mean	2.14E+05	9.33E+04	2.72E+05	7.54E+04	1.88E+05	1.53E+05	3.75E+05	3.43E+05	3.49E+05	2.85E+03
	Std.	2.84E+04	3.94E+04	2.18E+04	1.68E+04	1.53E+04	2.56E+04	4.74E+04	1.21E+04	3.52E+04	1.03E+03
F4	Mean	2.82E+02	2.98E+02	3.49E+02	2.67E+02	2.90E+02	6.94E+02	5.44E+04	2.45E+02	3.48E+02	2.93E+02
	Std.	2.14E+01	4.50E+01	4.21E+01	4.42E+01	5.55E+01	5.06E+01	8.44E+03	3.27E+01	5.38E+01	6.44E+01
F5	Mean	3.93E+02	7.31E+02	6.57E+02	7.59E+02	6.53E+02	1.23E+03	1.38E+03	6.06E+02	4.49E+02	7.98E+02
	Std.	2.32E+01	6.02E+01	5.30E+01	7.67E+01	3.61E+01	5.69E+01	5.72E+01	8.00E+01	7.25E+01	8.01E+01
F6	Mean	3.19E+01	6.00E+01	5.55E+01	6.04E+01	5.62E+01	6.62E+02	9.48E+01	4.82E+01	1.84E+01	4.77E+01
	Std.	1.95E+00	3.90E+00	3.55E+00	4.13E+00	2.14E+00	4.10E+00	5.47E+00	8.79E+00	4.71E+00	6.74E+00
F7	Mean	7.96E+02	2.49E+03	1.95E+03	2.62E+03	1.82E+03	3.25E+03	2.86E+03	7.32E+02	9.37E+02	2.14E+03
	Std.	4.67E+01	2.88E+02	1.70E+02	3.00E+02	1.12E+02	3.02E+02	1.50E+02	1.56E+02	1.29E+02	1.86E+02
F8	Mean	4.35E+02	8.28E+02	7.57E+02	8.64E+02	7.70E+02	1.63E+03	1.49E+03	5.44E+02	4.58E+02	8.84E+02
	Std.	2.18E+01	7.20E+01	6.68E+01	7.76E+01	4.55E+01	7.41E+01	5.50E+01	8.71E+01	6.93E+01	9.28E+01
F9	Mean	1.11E+04	2.27E+04	1.96E+04	2.18E+04	2.48E+04	2.32E+04	6.31E+04	2.35E+04	1.78E+04	2.10E+04
	Std.	1.09E+03	2.12E+03	1.51E+03	2.14E+03	2.43E+03	1.55E+03	5.06E+03	4.86E+02	7.06E+03	2.07E+03
F10	Mean	1.40E+04	1.50E+04	1.47E+04	1.54E+04	1.49E+04	1.63E+04	2.96E+04	1.14E+04	2.16E+04	1.45E+04
	Std.	1.20E+03	1.32E+03	1.29E+03	1.42E+03	1.52E+03	1.38E+03	1.00E+03	1.57E+03	3.48E+03	1.25E+03
F11	Mean	8.72E+03	1.35E+03	5.13E+03	1.17E+03	1.11E+03	2.47E+03	1.79E+05	2.14E+04	1.28E+04	1.20E+03
	Std.	1.20E+03	1.77E+02	1.07E+03	1.46E+02	1.02E+02	1.99E+02	3.38E+04	6.01E+03	3.19E+03	2.05E+02
F12	Mean	4.74E+07	9.51E+07	1.60E+08	1.99E+07	6.67E+07	1.03E+08	1.10E+11	6.80E+07	6.57E+07	4.30E+07
	Std.	5.67E+06	2.16E+07	3.13E+07	8.33E+06	1.23E+07	1.88E+07	1.70E+10	3.06E+07	3.31E+07	1.60E+08
F13	Mean	2.33E+04	3.73E+04	3.26E+04	3.72E+04	3.07E+04	3.95E+04	2.15E+10	1.64E+06	2.04E+04	1.65E+04
	Std.	3.87E+03	1.18E+04	6.62E+03	1.11E+04	5.53E+03	1.10E+04	5.56E+09	1.11E+07	2.11E+04	8.55E+03
F14	Mean	7.80E+05	3.52E+05	3.69E+05	1.53E+05	3.21E+05	3.81E+05	2.60E+07	4.09E+05	2.60E+06	2.79E+04
	Std.	1.03E+05	1.24E+05	1.21E+05	6.89E+04	9.13E+04	1.30E+05	1.39E+07	3.86E+05	1.23E+06	2.37E+04
F15	Mean	1.56E+04	3.21E+04	2.65E+04	3.24E+04	2.36E+04	3.31E+04	8.13E+09	7.16E+04	7.29E+03	6.03E+03
	Std.	3.07E+03	1.44E+04	8.86E+03	1.29E+04	6.68E+03	1.26E+04	2.32E+09	2.96E+04	6.69E+03	4.66E+03
F16	Mean	3.67E+03	5.51E+03	4.99E+03	4.97E+03	4.71E+03	7.14E+03	1.27E+04	3.47E+03	3.86E+03	4.53E+03
	Std.	4.19E+02	8.22E+02	6.23E+02	7.34E+02	5.61E+02	7.96E+02	1.77E+03	7.67E+02	5.93E+02	6.20E+02

Table 6 Experimental results on 100-D functions F_{17} – F_{30}

Function		BSO20	Classic BSO	BSO-OS	RGBSO	BSO-NBC	CBSO	ALBSO	BBGBSO	ScPSO	PPSO
F17	Mean	2.70E+03	3.96E+03	3.64E+03	3.91E+03	3.79E+03	5.73E+03	9.61E+04	2.74E+03	3.20E+03	4.10E+03
	Std.	3.53E+02	5.64E+02	5.67E+02	5.42E+02	5.80E+02	6.26E+02	1.56E+05	5.00E+02	5.89E+02	5.54E+02
F18	Mean	1.29E+06	6.54E+05	6.37E+05	2.53E+05	5.47E+05	6.70E+05	5.44E+07	8.39E+05	6.39E+06	9.41E+04
	Std.	1.68E+05	2.37E+05	1.89E+05	7.67E+04	1.31E+05	2.19E+05	3.78E+07	1.77E+06	3.70E+06	3.17E+04
F19	Mean	3.05E+05	2.64E+06	2.17E+06	9.17E+05	1.79E+06	1.72E+06	7.61E+09	2.44E+06	7.59E+03	6.02E+03
	Std.	1.72E+05	1.23E+06	1.09E+06	2.98E+05	8.64E+05	1.30E+06	2.55E+09	1.10E+06	7.83E+03	1.07E+04
F20	Mean	2.72E+03	3.45E+03	3.38E+03	3.63E+03	2.51E+03	5.48E+03	4.53E+03	2.84E+03	3.18E+03	3.32E+03
	Std.	4.71E+02	4.71E+02	4.12E+02	4.73E+02	3.76E+02	5.09E+02	4.27E+02	8.23E+02	5.71E+02	4.89E+02
F21	Mean	6.54E+02	1.70E+03	1.64E+03	1.72E+03	1.65E+03	3.80E+03	1.98E+03	7.51E+02	6.93E+02	1.36E+03
	Std.	2.61E+01	1.50E+02	1.25E+02	1.52E+02	1.51E+02	1.53E+02	1.03E+02	1.09E+02	7.11E+01	1.78E+02
F22	Mean	1.23E+04	1.66E+04	1.66E+04	1.70E+04	1.63E+04	1.91E+04	3.08E+04	5.72E+03	2.23E+04	1.69E+04
	Std.	5.61E+03	1.33E+03	1.33E+03	1.37E+03	1.22E+03	1.30E+03	1.04E+03	7.36E+03	3.41E+03	1.94E+03
F23	Mean	1.38E+03	2.99E+03	3.02E+03	3.04E+03	3.02E+03	5.35E+03	2.72E+03	1.16E+03	8.97E+02	1.92E+03
	Std.	6.83E+01	2.50E+02	2.54E+02	2.70E+02	2.75E+02	2.61E+02	2.38E+02	1.61E+02	6.35E+01	2.27E+02
F24	Mean	2.63E+03	3.62E+03	3.94E+03	4.14E+03	3.85E+03	5.97E+03	4.30E+03	1.48E+03	1.31E+03	3.31E+03
	Std.	1.19E+02	3.97E+02	5.07E+02	5.53E+02	7.21E+02	6.87E+02	4.09E+02	1.96E+02	6.32E+01	4.84E+02
F25	Mean	8.32E+02	8.45E+02	8.61E+02	7.91E+02	8.30E+02	3.35E+03	1.94E+04	7.26E+02	9.80E+02	8.06E+02
	Std.	5.50E+01	6.49E+01	3.34E+01	5.88E+01	5.69E+01	5.95E+01	1.76E+03	6.26E+01	8.42E+01	5.61E+01
F26	Mean	2.03E+04	2.81E+04	2.69E+04	2.94E+04	2.35E+04	3.19E+04	4.13E+04	1.26E+03	8.08E+03	1.97E+04
	Std.	6.68E+02	1.74E+03	1.55E+03	1.63E+03	1.95E+03	2.01E+03	4.25E+03	2.66E+03	7.77E+02	4.54E+03
F27	Mean	1.97E+03	4.42E+03	4.71E+03	4.33E+03	4.40E+03	7.26E+03	5.00E+03	6.89E+02	7.54E+02	1.41E+03
	Std.	1.52E+02	9.99E+02	1.01E+03	1.15E+03	1.04E+03	1.07E+03	8.01E+02	4.71E+01	4.73E+01	3.57E+02
F28	Mean	6.39E+02	6.40E+02	7.12E+02	6.03E+02	6.18E+02	3.44E+03	2.55E+04	5.82E+02	7.10E+02	6.47E+02
	Std.	2.95E+01	3.80E+01	3.35E+01	3.95E+01	3.57E+01	3.70E+01	2.37E+03	4.62E+01	3.90E+01	6.52E+01
F29	Mean	4.66E+03	6.34E+03	6.45E+03	5.83E+03	6.22E+03	9.32E+03	3.31E+04	3.82E+03	3.58E+03	4.73E+03
	Std.	4.36E+02	6.46E+02	6.21E+02	5.53E+02	5.63E+02	7.11E+02	1.74E+04	6.45E+02	5.05E+02	6.62E+02
F30	Mean	5.07E+06	1.28E+07	1.62E+07	4.49E+06	1.08E+07	1.42E+07	1.56E+10	1.36E+07	4.10E+04	7.95E+05
	Std.	1.11E+06	3.18E+06	4.00E+06	1.41E+06	2.49E+06	3.69E+06	4.89E+09	5.05E+06	2.16E+04	1.02E+06
“+/-”			25/4	26/3	20/9	22/7	25/4	29/0	15/14	19/10	17/12

rithms on 30D, 50D, and 100D functions are shown in Table 7, which are obtained by the open source software KEEL 3.0 [31]. The details of related statistical indicators can be found in [14]. In each pair of comparisons of Table 7, the bolded R^+ values emphasize that BSO20 outperforms the compared algorithm with a level of significance of 0.05.

As can be seen from Table 7, BSO20 significantly outperforms the classic BSO, BSO-OS, BSO-NBC, CBSO, and ALBSO with a level of significance of 0.05 on different dimensions. Compared with RGBSO and BBGSO, BSO20 has significant superiority on 30D problems, but not significant on 50D and 100D. Nevertheless, BSO20 is still better than RGBSO and BBGSO from the value of R^+ . Therefore, the statistical results demonstrate the competitive performance of BSO20. From the last two rows of Table 7, the values of signed rank R^+ and R^- demonstrate that BSO20 is better than ScPSO on 50D and 100D problems, and better than PPSO on 30D and 50D problems. However, the experimental results also indicate that the performance difference between BSO20 and SCPSO as well as PPSO is not significant at a level of significance of 0.05.

Parameter analysis

BSO20 contains only two parameters, the intra-cluster mutation probability $p_{\text{one_cluster}}$, and the cluster size S_r . In this section, the impact of the parameters $p_{\text{one_cluster}}$ and S_r on the performance of BSO20 are analyzed. For the convenience of presentation, the logarithm of the mean fitness, i.e., $\log_{10}(\text{fit})$, is used as the ordinate.

Parameter analysis of $p_{\text{one_cluster}}$

Figures 2 and 3 show the mean values of experimental results of functions F_1 , F_3 – F_{30} with different intra-cluster mutation probability $p_{\text{one_cluster}}$, where $p_{\text{one_cluster}}$ is set to 0, 0.1, ..., 0.9, 1, and S_r is set to 20.

From Figs. 2 and 3, it can be seen that, for the same function in different dimensions (i.e., 30D and 50D), the trend of the mean changing with the parameter $p_{\text{one_cluster}}$ are similar. The experimental results also indicate that for most functions, inter-cluster mutation with greater probability could achieve better results, especially on functions F_{19} , F_{22} and F_{30} . However, there are some functions that are not sensitive to the parameter $p_{\text{one_cluster}}$, such as F_{11} – F_{13} , F_{26} – F_{27} . Moreover, BSO20 with greater probability of the intra-cluster mutation is more effective on some functions, such as F_3 , F_{14} and F_{18} , where F_3 is a unimodal function, and F_{14} , F_{18} are hybrid functions including unimodal components.

From the experimental results, the following conclusion could be drawn. For most multimodal functions, the inter-cluster mutation has more effect on the performance of BSO20, while for some unimodal functions, the inter-cluster

mutation may not be significant. One of the reasons for this situation is that for global optimization problems, the inter-cluster mutation helps share information between multiple clusters, and the clusters are gathered slowly, which is beneficial to further exploitation of promising areas for the algorithm.

Parameter analysis of S_r

The parameter S_r determines the total number of clusters k in BSO20, and k determines whether BSO20 prefers exploitation or exploration. If k is small, the clusters gather faster, and BSO20 tends to exploit. Conversely, BSO20 tends to explore if there are many clusters. In order to ensure that k is an integer, S_r is set to {5, 10, 20, 24, 30, 40} for 30D problems, and S_r is set to {5, 10, 20, 25, 40, 50} for 50D problems. Moreover, $p_{\text{one_cluster}}$ is set to 0.1.

Figures 4 and 5 show the mean values of experimental results of functions F_1 , F_3 – F_{30} with different clusters size S_r . There are trends in Figs. 4 and 5 to suggest that S_r has a little impact on the performance of the algorithm on most functions, except for 50D function F_{22} . In other words, BSO20 is not sensitive enough to the parameter S_r on most functions. Moreover, the results also reveal that the effect of S_r on the performance of the algorithm is related to the landscape of functions. For example, BSO20 prefers a larger S_r for functions F_{20} and F_{22} , while a smaller S_r for function F_3 and F_{30} .

Discussion

BSO20 includes two improvements, i.e., the clustering strategy and the mutation strategy. In this section, the impact of the hybrid clustering strategy and the modified mutation strategy of BSO20 in terms of CPU time consumption and convergence are discussed.

CPU time consumption

The clustering strategy is one of the crucial parts that affects the efficiency and performance of BSO. In this paper, the hybrid clustering strategy combining NBC and RGS was proposed. Compared with the classic BSO, the proposed hybrid strategy improves the efficiency and performance of BSO. In this section, CPU time consumption is used as an indicator to compare the efficiency of BSO20 and the classic BSO.

The details of the platform we used are Intel(R) Xeon(R) CPU E5-2640 v4 2.4 GHz processor, 32GB RAM, MATLAB R2017b, and Windows Server 2012 R2. The experiments open up a parallel pool, which processes 29 functions in parallel, and the algorithm runs independently on each function 51 times.

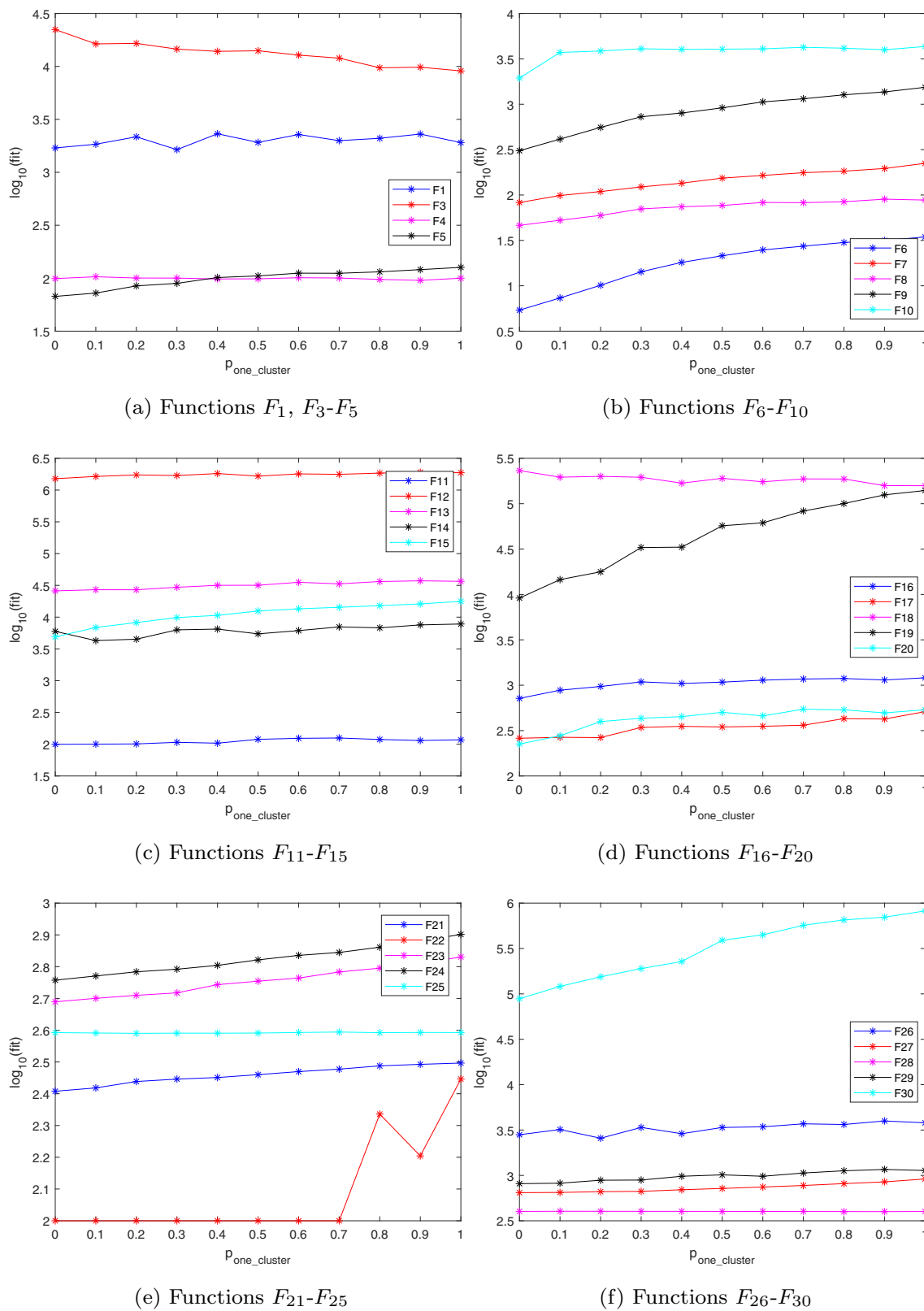


Fig. 2 The effect of parameter $p_{\text{one_cluster}}$ on 30-D functions (F_1, F_3-F_{30})

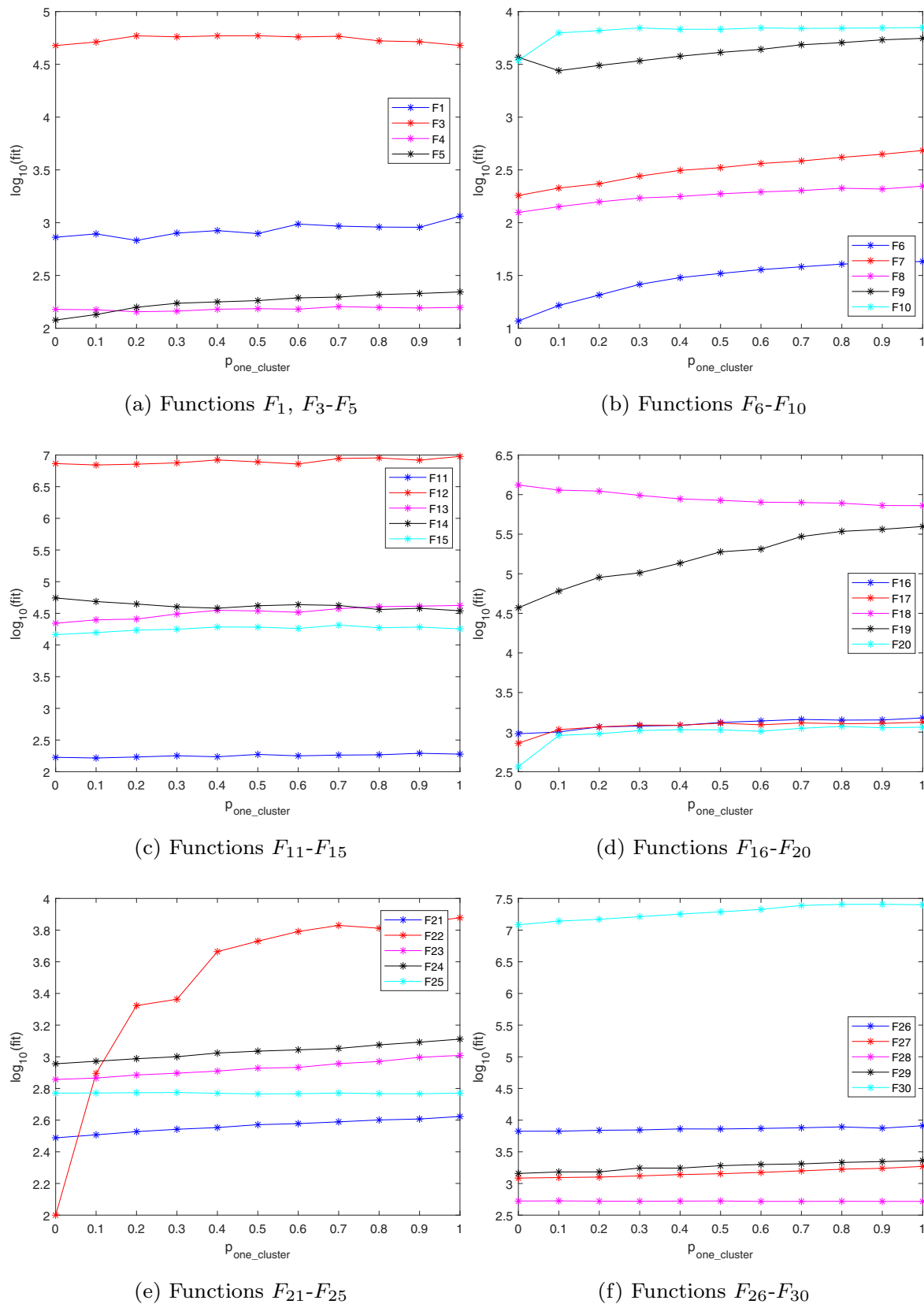


Fig. 3 The effect of parameter $p_{\text{one_cluster}}$ on 50-D functions (F_1, F_3-F_{30})

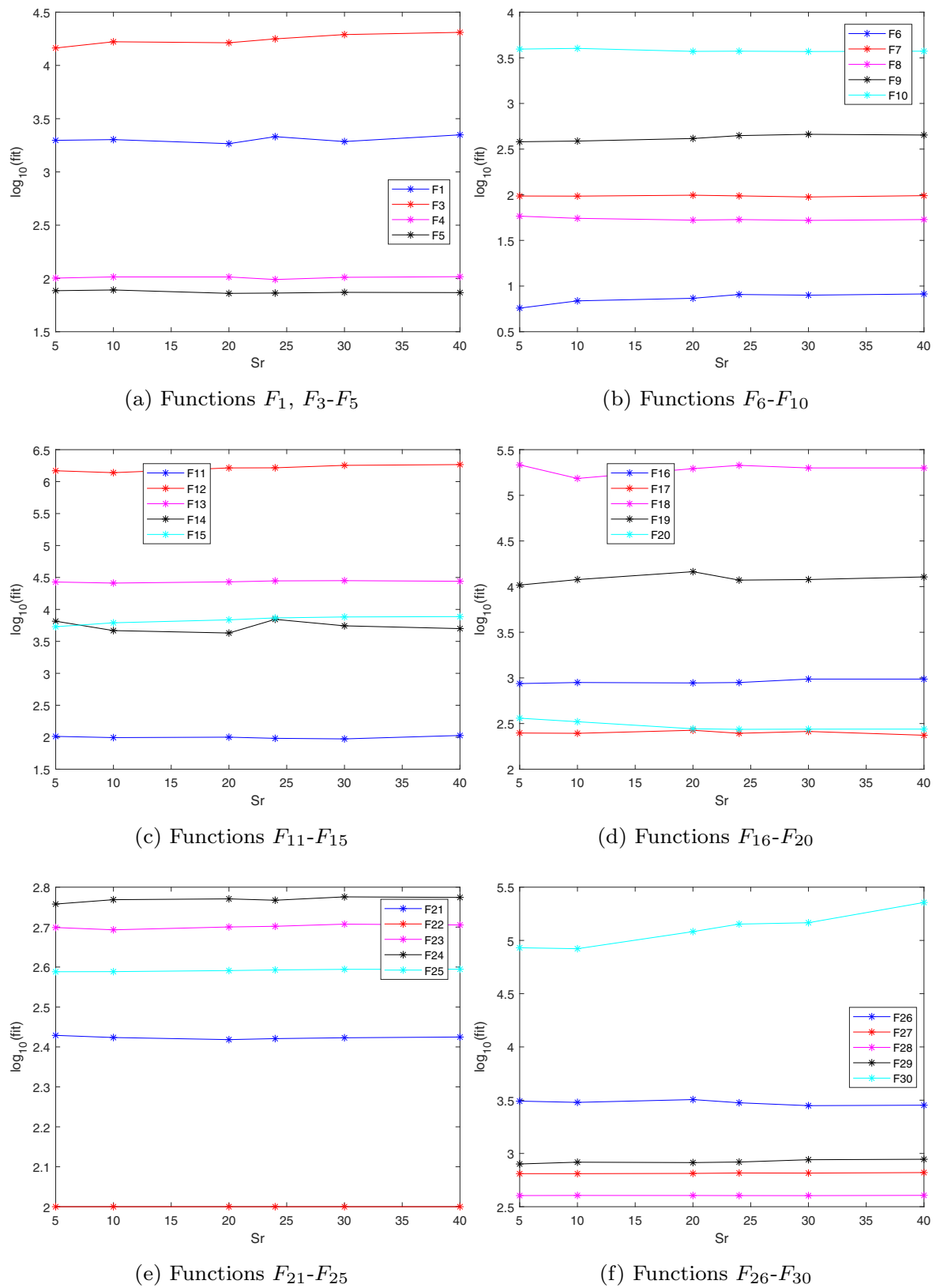


Fig. 4 The effect of parameter S_r on 30- D functions (F_1, F_3-F_{30})

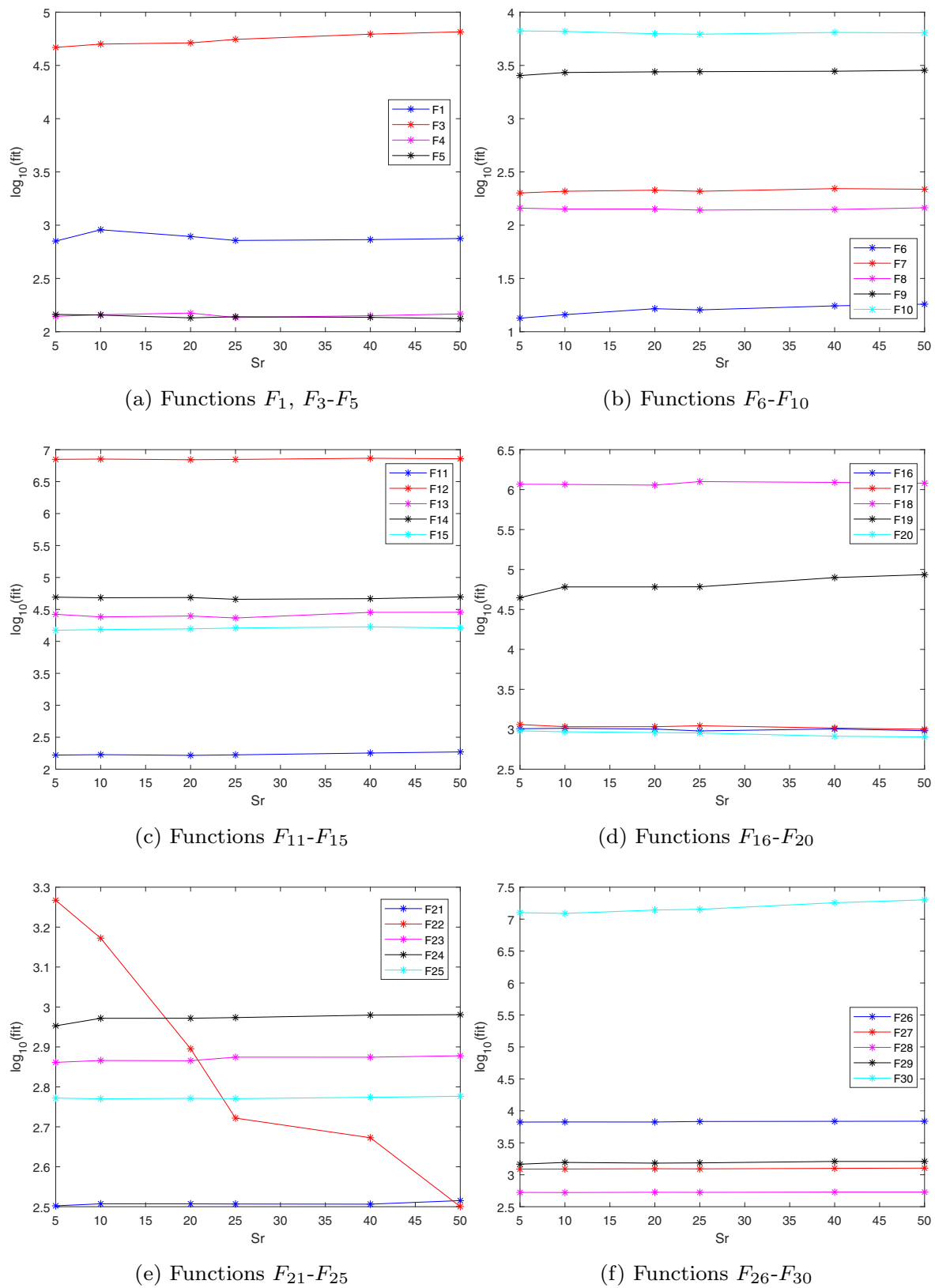
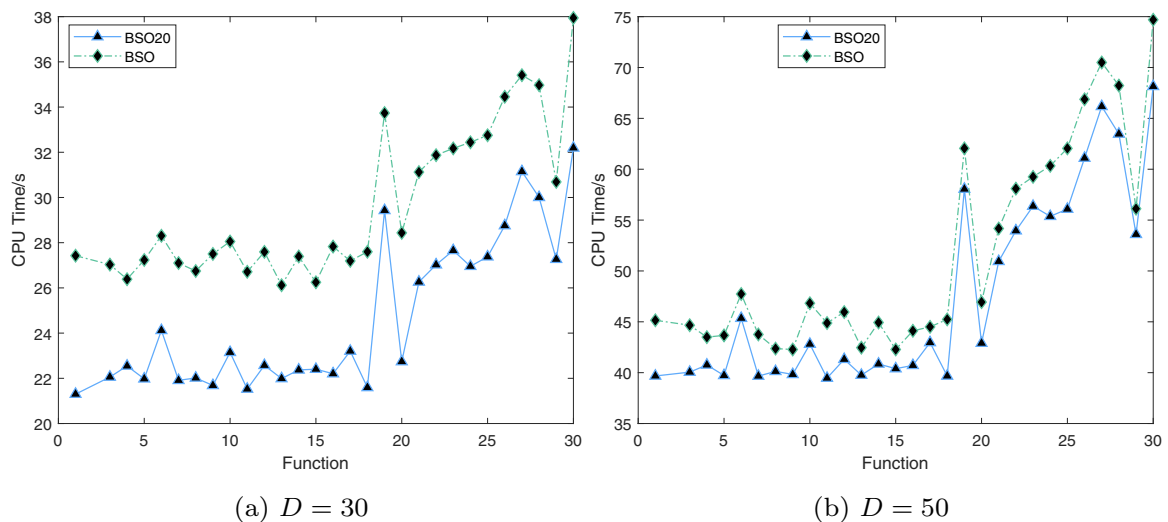
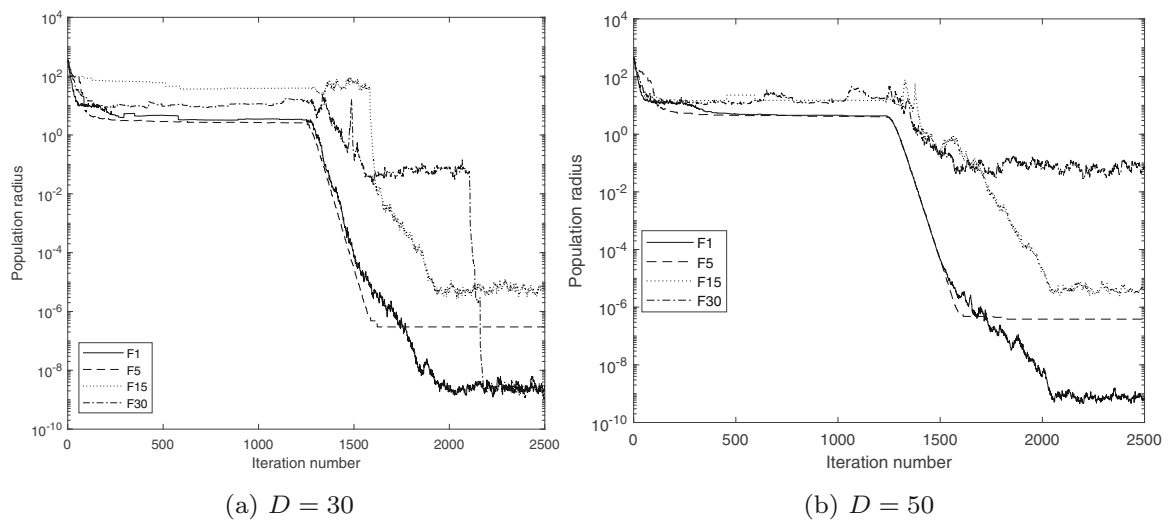


Fig. 5 The effect of parameter S_r on 50- D functions (F_1, F_3-F_{30})

Table 7 Results obtained by the Wilcoxon signed-rank test for BSO20

VS	30D			50D			100D		
	R^+	R^-	p value	R^+	R^-	p value	R^+	R^-	p value
Classic BSO	369.0	66.0	0.001	355.0	80.0	0.002	342.0	93.0	0.006
BSO-OS	382.0	53.0	0.000	378.0	57.0	0.000	367.0	68.0	0.001
RGBSO	330.0	105.0	0.014	296.0	139.0	0.092	275.0	160.0	0.210
BSO-NBC	328.0	107.0	0.016	343.0	92.0	0.006	332.0	103.0	0.012
CBSO	384.0	51.0	0.000	361.0	74.0	0.001	343.0	92.0	0.006
ALBSO	435.0	0.0	0.000	435.0	0.0	0.000	435.0	0.0	0.000
BBGBSO	348.0	87.0	0.004	289.0	146.0	0.126	252.0	183.0	0.449
ScPSO	194.0	241.0	1.000	221.0	214.0	0.931	273.0	162.0	0.226
PPSO	230.0	205.0	0.779	243.0	192.0	0.574	187.5	247.5	1.000

**Fig. 6** The mean CPU time on 29 functions**Fig. 7** The changing trends of the population radius of BSO20 on F_1 , F_5 , F_{15} , and F_{30} functions

The Fig. 6 shows the experimental results on 30D and 50D functions. The CPU time consumption of BSO20 on each function is less than the classic BSO. It can be concluded that the clustering strategy adopted by BSO20 is more efficient than k -means adopted by classic BSO. Therefore, experimental results of the algorithm performance and the CPU time consumption both support that the hybrid clustering strategy is superior to k -means adopted by the classic BSO.

Convergence analysis

BSO20 is a variant of BSO, which has the advantage of BSO exploring multiple promising regions in parallel. The novel mutation strategy increases the step size of each mutation and shares information among multiple clusters to improve the exploration efficiency.

In this section, the functions F_1 , F_5 , F_{15} and F_{30} are selected as the representatives of unimodal functions, simple multimodal, hybrid functions, and composition functions. The population radius could be used to analyze the convergence of the algorithm on these functions [18]. In this paper, the population radius is calculated as Formula (11).

$$r = \max_{1 \leq i \leq n} \{\|x_i - x_{\text{mean}}\|\}, \quad (11)$$

where x_i is the i th individual in the population, x_{mean} represents the center of the population, which is set to the mean of all individuals, $\|\cdot\|$ represents the Euclidean distance, and n is the population size. There is one difference from [18], we adopt the maximum value of the distances between all individuals and the center of the population as the population radius instead of the average.

Figure 7 shows the changing trends of the population radius on 30D and 50D functions during evolution, where the ordinate is the population radius and the abscissa is the number of iterations. The experimental results reveal that the population radius decreases significantly as the iteration number increases. The changing trends of the population radius shown in Fig. 7 could be divided into three stages. In the early stage of the algorithm, multiple clusters are distributed in different regions, and the population radius is large. Subsequently, about 1200–1500 iterations, multiple clusters gather, BSO20 focuses on exploiting the local regions. Finally, the population will converge. However, population convergence cannot always be guaranteed. For example, the population radius of 50D function F_{30} eventually tends to 10^{-1} . The reason for this situation is that there are multiple clusters localized to local peaks.

Conclusions

In this paper, we have proposed BSO20 as the improvement of the classic BSO for real-parameter numerical optimization problems.

The major changes of BSO20 include the clustering strategy and the mutation strategy. First, a hybrid clustering strategy was proposed in this paper for BSO20, which combines NBC and RGS, and the subpopulation size clustered by two strategies was dynamically adjusted as the population evolves. The new clustering strategy not only has superiority in time efficiency but also achieves good results in experiments. Second, a modified mutation strategy was also proposed for BSO20 to enhance the exploration ability. Overall, BSO20 improves the time efficiency and the search efficiency of BSO. Moreover, BSO20 reduces the number of parameters of the classic BSO to ease the difficulty of tuning parameters. BSO20 has been tested on CEC'17 RPNO and compared with several other BSOs as well as two PSOs. The experimental results have demonstrated the competitiveness of the proposed algorithm.

In the future work, we will further improve the BSO20, especially the adaptive strategy to adjust the parameters dynamically to balance the convergence and diversity of the population. Additionally, we will draw on the successful experience of differential evolution (DE) and particle swarm optimization (PSO), and strive to improve the performance of BSO20 to achieve more highly-competitive results.

Acknowledgements This study is supported by the National Key Research and Development Program of China (Grant No. 2020YFB21-04003), the Guangdong Provincial Key Laboratory (Grant No. 2020B-121201001) and the National Natural Science Foundation of China (Grant No. 61573327).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Awad NH, Ali MZ, Liang JJ, Qu BY, Suganthan PN (2016) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. Technical Report, Nanyang Technological University, Singapore
2. Cao Z, Hei X, Wang L, Shi Y, Rong X (2015) An improved brain storm optimization with differential evolution strategy for applications of ANNs. Math Probl Eng 2015:923698
3. Cao Z, Rong X, Du Z. (2017) An improved brain storm optimization with dynamic clustering strategy. In: Proceedings of the MATEC web of conferences, EDP sciences, p 19002

4. Cao Z, Shi Y, Rong X, Liu B, Du Z, Yang B (2015) Random grouping brain storm optimization algorithm with a new dynamically changing step size. In: Proceedings of the international conference in swarm intelligence. Springer, pp 357–364
5. Cao Z, Wang L (2019) An active learning brain storm optimization algorithm with a dynamically changing cluster cycle for global optimization. *Cluster Comput* 22:1413–1429
6. Chen J, Deng C, Peng H, Tan Y, Zhou X, Wang F (2019) Enhanced brain storm optimization with role-playing strategy. In: Proceedings of the 2019 IEEE congress on evolutionary computation (CEC). IEEE, pp 1132–1139
7. Chen J, Wang J, Cheng S, Shi Y (2016) Brain storm optimization with agglomerative hierarchical clustering analysis. In: Proceedings of the international conference on swarm intelligence. Springer, pp 115–122
8. Chen J, Xie Y, Ni J, (2014) Brain storm optimization model based on uncertainty information. In: Proceedings of the 2014 tenth international conference on computational intelligence and security. IEEE, pp 99–103
9. Chu X, Chen J, Cai F, Li L, Qin Q (2018) Adaptive brainstorm optimisation with multiple strategies. *Memet Comput* 10:383–396
10. El-Abd M (2016) Brain storm optimization algorithm with re-initialized ideas and adaptive step size. In: Proceedings of the 2016 IEEE congress on evolutionary computation (CEC). IEEE, pp 2682–2686
11. El-Abd M (2017) Global-best brain storm optimization algorithm. *Swarm Evolut Comput* 37:27–44
12. El-Abd M (2019) Gaussian bare-bones brain storm optimization algorithm. In: Proceedings of the 2019 IEEE congress on evolutionary computation (CEC). IEEE, pp 227–233
13. Elsayed S, El-Abd M, Sallam K (2019) Enhancing the local search ability of the brain storm optimization algorithm by covariance matrix adaptation. In: Brain storm optimization algorithms. Springer, pp 105–122
14. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *J Heuristics* 15:617–644
15. Ghasemi M, Akbari E, Rahimnejad A, Razavi SE, Ghavidel S, Li L (2019) Phasor particle swarm optimization: a simple and efficient variant of PSO. *Soft Comput* 23:9701–9718
16. Jia Z, Duan H, Shi Y (2016) Hybrid brain storm optimisation and simulated annealing algorithm for continuous optimisation problems. *Int J Bio-Inspired Comput* 8:109–121
17. Koyuncu H, Ceylan R (2019) A PSO based approach: scout particle swarm algorithm for continuous global optimization problems. *J Comput Des Eng* 6:129–142
18. Li C, Yang S (2009) A clustering particle swarm optimizer for dynamic optimization. In: Proceedings of the 2009 IEEE congress on evolutionary computation (CEC). IEEE, pp 439–446
19. Li F, Peng H, Deng C, Zeng Y, Tan Y, Wang W (2019) Brain storm optimization algorithm based on competition mechanism. In: Proceedings of the 2019 IEEE congress on evolutionary computation (CEC). IEEE, pp 1140–1147
20. Lin X, Luo W, Xu P (2021) Differential evolution for multimodal optimization with species by nearest-better clustering. *IEEE Trans Cybern* 51:970–983
21. Lu H, Zhou R, Cheng S, Shi Y (2019) Brain storm optimization for test task scheduling problem. In: Brain storm optimization algorithms. Springer, pp 35–59
22. Luo W, Qiao Y, Lin X, Xu P, Preuss M (2020) Hybridizing niching, particle swarm optimization, and evolution strategy for multimodal optimization. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2020.3032995>
23. Luo W, Sun J, Bu C, Yi R (2018) Identifying species for particle swarm optimization under dynamic environments. In: Proceedings of the 2018 IEEE symposium series on computational intelligence. IEEE, pp 1921–1928
24. Peng H, Deng C, Wu Z (2019) SPBSO: self-adaptive brain storm optimization algorithm with pbest guided step-size. *J Intell Fuzzy Syst* 36:5423–5434
25. Pourpanah F, Lim CP, Wang X, Tan CJ, Seera M, Shi Y (2019) A hybrid model of fuzzy min-max and brain storm optimization for feature selection and data classification. *Neurocomputing* 333:440–451
26. Pourpanah F, Shi Y, Lim CP, Hao Q, Tan CJ (2019) Feature selection based on brain storm optimization for data classification. *Appl Soft Comput* 80:761–775
27. Preuss M (2010) Niching the CMA-ES via nearest-better clustering. In: Proceedings of the 12th annual conference companion on genetic and evolutionary computation. ACM, pp 1711–1718
28. Preuss M, Schönmann L, Emmerich M (2005) Counteracting genetic drift and disruptive recombination in (μ^+, λ) -EA on multimodal fitness landscapes. In: Proceedings of the 7th annual conference on genetic and evolutionary computation. ACM, pp 865–872
29. Shi Y (2011) Brain storm optimization algorithm. In: Proceedings of the international conference in swarm intelligence. Springer, pp 303–309
30. Shi Y (2015) Brain storm optimization algorithm in objective space. In: Proceedings of the 2015 IEEE congress on evolutionary computation (CEC). IEEE, pp 1227–1234
31. Triguero I, González S, Moyano JM, García S, Alcalá-Fdez J, Luengo J, Fernández A, del Jesús MJ, Sánchez L, Herrera F (2017) KEEL 3.0: an open source software for multi-stage analysis in data mining. *Int J Comput Intell Syst* 10:1238–1249
32. Wang H, Liu J, Yi W, Niu B, Baek J (2017) An improved brain storm optimization with learning strategy. In: Proceedings of the international conference on swarm intelligence. Springer, pp 511–518
33. Wang R, Ma L, Zhang T, Cheng S, Shi Y (2019) Brain storm optimization algorithm based on improved clustering approach using orthogonal experimental design. In: Proceedings of the 2019 IEEE congress on evolutionary computation (CEC). IEEE, pp 262–270
34. Wu X (2019) Brain storm optimization algorithms for flexible job shop scheduling problem. In: Brain storm optimization algorithms. Springer, pp 245–271
35. Yang Y, Shi Y, Xia S (2015) Advanced discussion mechanism-based brain storm optimization algorithm. *Soft Comput* 19:2997–3007
36. Yang Z, Shi Y (2015) Brain storm optimization with chaotic operation. In: Proceedings of the 2015 seventh international conference on advanced computational intelligence. IEEE, pp 111–115
37. Yu Y, Gao S, Cheng S, Wang Y, Song S, Yuan F (2018) CBSO: a memetic brain storm optimization with chaotic local search. *Memet Comput* 10:353–367
38. Yu Y, Wu L, Yu H, Li S, Wang S, Gao S (2017) Brain storm optimization with adaptive search radius for optimization. In: Proceedings of the 2017 international conference on progress in informatics and computing. IEEE, pp 394–398
39. Zhan Z, Chen W, Lin Y, Gong Y, Li Y, Zhang J (2013) Parameter investigation in brain storm optimization. In: Proceedings of the 2013 IEEE symposium on swarm intelligence. IEEE, pp 103–110
40. Zhan Z, Zhang J, Shi Y, Liu H (2012) A modified brain storm optimization. In: Proceedings of the 2012 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
41. Zhang W, Zhang Y, Peng C (2019) Brain storm optimization for feature selection using new individual clustering and updating mechanism. *Appl Intell* 49:4294–4302
42. Zhou D, Shi Y, Cheng S (2012) Brain storm optimization algorithm with modified step-size and individual generation. In: Proceedings

- of the international conference in swarm intelligence. Springer, pp 243–252
43. Zhu H, Shi Y (2015) Brain storm optimization algorithms with k -medians clustering algorithms. In: Proceedings of the 2015 seventh international conference on advanced computational intelligence. IEEE, pp 107–110

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.