Multi-indicators Brain Storm Optimization Algorithm for Multi-objective Optimization Problems

Xuan ZHANG, Guang-ming YANG, Tian ZHANG* and Ming-li SHI

College of Software, Northeastern University, Shenyang, China

*Corresponding author

Keywords: Multi-objective optimization, Brain storm optimization, Indicator.

Abstract. The brains storming optimization (BSO) is a novel swarm intelligence algorithm, which is inspired from the intelligence of the human brainstorming process. In BSO, the convergent operation and divergent operation are implemented to drive the evolution of individuals towards a global optimal region in the single-objective space. In order to extend BSO to resolve multi-objective optimization problems, this paper presents a multi-objective brainstorming algorithm based on multiple indicators, called MIBSO. In MIBSO, the two indicators are used to measure the dispersion and convergence degrees of the obtained solutions, respectively. Experimental results show the effectiveness and efficiency of the proposed algorithm.

Introduction

With the development of artificial intelligence technology, swarm intelligence (SI) has obtained a surge of attentions. A number of swarm intelligence algorithms have been proposed and developed for real-world applications. Essentially, these algorithms belong to a kind of probability-based search paradigm. Generally, in these algorithms, each individual represents a solution of the problem, also regarded as a data sample from the search space. In the family of SI algorithms, e.g., genetic algorithm (GA) [1], artificial bee colony algorithm (ABC) [2-7], bacterial foraging optimization algorithm (BFO) [8][9], and particle swarm algorithm (PSO) [10], all individuals should exchange information and then generate position perturbation via learning from the elite individuals among the population. To be stressed, the motivations of most SI algorithms are usually based on the intelligent behaviors of simple insects or animals. Different from the above, the brain storm optimization (BSO) algorithm is an excellent SI paradigm, which is based on the emerging intelligence of human brainstorming process, more complex than the behaviors of simple insects. In BSO, the individuals are grouped into several clusters, and the operations mainly involve the convergent operator and divergent operator. The new individuals are generated based on mutation operation [11-13]. However, current researches regarding BSO mainly focus on the performance improvements on single-objective optimization problems. In most real-world optimization scenarios, there are usually more than one objective [14]. For example, in the production activities, multiple factors such as material cost, the freight, and loss rate would affect the production efficiency. Thus, the multi-objective BSO algorithms need to be designed deliberately for these multi-objective optimization problems (MOPs). Note that, there is usually not a single solution but a set of optimal solutions in MOPs, also referred as non-dominated solutions or Pareto optimal solutions. Therefore, in this paper, we propose several multi-objective search strategies to extend the original BSO to be a multi-objective version, which can obtain a good performance on multi-objective optimization problem.

Original BSO and Its Variants

BSO is a relatively newly developing swarm intelligence optimization algorithm, which was proposed by Professor Shi Yuhui at the Second International Conference on Community Intelligence (ICSI11) in 2011 [15]. The basic idea of most existing SI algorithms is to simulate the social behaviors of natural creatures, such as bird flock search for food, bee colony search for honey source [16-20] to construct a stochastic optimization paradigm. Note that, human’s brain is the most intelligent creature, it is a
straightforward and effective way to mimic the process of the human’s “brain storm meeting” session for the complex optimization. The brain storm meeting is one of the group discussion modes, where the problem by a group of people is discussed under the organization of the question-master. There are two kinds of people in the brain storm meeting process: one is a question-master, and the others are a group of participators. The question-master, which is responsible for presiding over the meeting, must be good at motivating members to think, and ensure that there is no prejudice in the whole process. The participators should have divergent thinking as much as possible under the guide of the question-master, and the better ideas will be kept by the question-master. Using a brainstorming session method for a topic, all participants can freely present their ideas and are inspired by each other. This can induce more creative ideas and inspiration [21]. In the human brainstorming session, more and more new ideas are put forward by the people who attend the meeting. People in the brainstorming group will need to be open-minded as much as possible. Generally, the BSO imitates the process of human brainstorming sessions to solve the complex problems.

There are three main processes in original BSO, including the clustering, the generation and mutation, and the selection process [22].

The clustering step is to divide the individuals into a set of groups. The k-means algorithm is a simple cluster method used in BSO. In each cluster, the individual with the best fitness is considered as the cluster center, which represents a local optimal solution in a group. New individuals are generated by learning from the clustering center or other individuals, which can provide a search direction towards the global optimal region. After clustering, in order to increase the randomness of BSO, a disturbance strategy is added on an existing idea. This also can reduce the probability of being trapped into the local optima. The key step is the generation process in BSO, in each iteration, a randomly selected cluster center is replaced by an individual from the population. Then, new individuals are generated via four different ways, i.e., one cluster center from one cluster, one ordinary idea from one cluster, two randomly selected cluster center from two clusters, and two randomly selected ordinary idea from two clusters. When a generation choice is made, the new individual will be generated from the current ones by a mutation operation, where the Gaussian mutation is usually adopted due to its promising feather, formulated as follows.

$$x_{new}^{i} = x_{current}^{i} + \xi^{i} \times N(\mu, \sigma)$$  \tag{1}$$
$$\xi^{i} = \log \text{sig}(\frac{T - t}{K}) \times \text{rand}(\cdot)$$  \tag{2}$$

where, the represents the i-th dimension of the newly generated individual, the $x_{current}^{i}$ represents the i-th dimension of the current individual which is selected to be prepared for the mutation operation. The $x_{new}^{i}$ is the new individual generated by $x_{current}^{i}$ by Gaussian disturbance Gaussian perturbation ($N(\mu, \sigma)$) with steps. The parameter $\xi^{i}$ represents the step size, it weights the contribution of the Gaussian random value to the new generated value, the value of $\xi^{i}$ varies with the current iteration number to adjust search range. In formula (2), the T represent the maximum iteration, the t is the current iteration. The coefficient K is used to change the scale of function. The main procedure of BSO is showed in Algorithm. 1.

Accordingly, based on the above procedures, there are many BSO variants to improve the performance. For examples, the predator–prey brain storm optimization (PPBSO) is developed [23], where the cluster centers perform as predators, with a trend towards a better and better position; whereas the other ideas perform as preys, escaping from their nearest predators. Furthermore, a chaotic operator is incorporated into PPBSO [24], which utilizes the performance of chaotic operation in helping individuals jumping out of stagnation. In [25] a quantum-behaved BSO (QBSO) algorithm is proposed to tackle Loney’s solenoid problem by incorporating a quantum mechanism into each idea to improve population diversity and avoid the local optima.
**Algorithm 1.** Procedure of the original BSO

|   |   |
|---|---|
| **Input:** | Population: n, Maxiterations : Max, number of clusters:m |
| **Output:** | Globally optimal individual of each iterations. |
|   | 1. Generate n individuals randomly. |
|   | 2. Evaluate the fitness of n individuals. |
|   | 3. While inter Max do |
|   | 4. cluster the n individuals into m clusters, choose the centers of each cluster. |
|   | 5. For each individuals: |
|   | 6. If rand() < P_one then |
|   | 7. If rand() < P_center |
|   | 8. Select a center for updating |
|   | 9. Else |
|   | 10. Select a normal individual for updating |
|   | 11. Else |
|   | 12. If rand() < P_centers |
|   | 13. Select two centers for updating |
|   | 14. Else |
|   | 15. Select two normal individuals for updating |
|   | 16. End |
|   | 17. The offspring individuals are compared with parents individuals, the better individuals will be reserved. |
|   | 18. **Return:** Best individuals. |

Due to the promising performance, BSO and its variants have been utilized to resolve a variety of complex real-world problems [22] [26], such as economic dispatch considering wind power problem, optimal FACTS devices, electric power dispatch problems, and optimal power flow solution. However, there are few works regarding However, there are few works developed to handle the MOPs, thus, in this paper, we need to extend the original BSO to be a multi-objective version.

**Multi-Objective Brain Storm Optimization**

**Basic Idea**

In BSO, there is only one objective function to be optimized, as a result, only one solution will be found, which is a relatively easy task. Different from the above, there are multiple optimal solutions in multi-objective optimization, which represents the tradeoff between objectives to be optimized. These final solutions, termed as Pareto optimal solutions, need to be obtained in one run. In multi-objective optimization, there are two goals to be considered, i.e., the convergence and diversity. However, traditional Pareto-based approaches would become ineffective in complex MOPs, especially with the increasing number of objectives. Thus, we incorporate another effective approach, i.e., multi-indicators based approach, into BSO to resolve complex MOPs. In addition, in order to make it be suitable for multi-objective problems, some other improvements should be done for the operations of BSO, including the convergent operation and divergent operation [27].

**Convergent operation:** It is realized by the k-means method. Like most of SI algorithms, BSO needs to find better results via exploiting around the optimal solutions or exploring towards the optimal solutions. Accordingly, the convergent operation is to exploit better results around current optimal solutions. When solving the MOPs, the k-means method is not very effective to cluster the population, because it consumes a large computation cost, which is exponentially increased with the increase of the objective number. In addition, the number of dimensions of a solution in MOPs is usually much larger than the number of objectives, which would also deteriorate the effect of clustering. In this paper, we use the objective-space clustering approach in [27] to reduce the computation complexity.
Divergent operation: It is used to generate new individuals with a better diversity. Before generating a new individual, an individual need to be selected randomly as a parent idea. There are four strategies to choose this parent individual from one cluster or two clusters: 1) selecting a cluster center, 2) selecting an ordinary individual from a cluster, 3) selecting two cluster centers, 4) selecting two ordinary individuals from two clusters. The first two strategies choose one individual to search an optimal solution near a promising region, which is an effective method to improve exploitation ability of algorithm. The last two strategies use the combination of two intra-cluster individuals to construct the parent individual can find the optimal solution far from the current individuals, which can improving the population diversity and enhance exploration ability.

Proposed Algorithm

In this paper, an improved multi-objective brain storm optimization (MIBSO) is proposed to solve MOPs. Different indicators have different evaluations of solutions. For instance, in the IBEA algorithm [28], the indicator $I_{e+}$ is used to maintain the convergence of solutions. In order to obtain optimal solutions with good diversity and convergence, MIBSO adopts two different indicators to guide the evolution of population. Using indicators $I_{e+}$ can obtain solutions close to the Pareto front, which means the final solution set will perform better in convergence. Another indicator $I_d$ is based on diversity, it can make the solution set more dispersed [29]. The two indicators favor the diversity and convergence, respectively. The detailed introduction of the two indicators is in section 3.6. The main flow chart of the MIBSO is showed in Algorithm 2.

| Algorithm 2 | Procedure of the MIBSO |
|-------------|------------------------|
| 1. **Initialization operation:** Randomly generate n individuals, evaluate the n individuals; |
| 2. **While** termination not satisfied **do:** |
| 3. **Disturbance step:** Randomly replace a individual depend on the probability. |
| 4. **Clustering step:** All of the individuals are sorted according to Pareto dominance and crowding distance. Select the first n individuals as elite sets, the others belong to normal set. |
| 5. **New individual generated step:** |
| randomly generate a probability value $P_1$. |
| If $P_1 < p_a$ then |
| Generate a probability value $P_2$; |
| If $P_2 < p_b$ then |
| Choose a randomly individual from normal set as the selected one. |
| Else then |
| Choose a randomly individual from elite sets as the selected one. |
| End |
| Else |
| generate a probability value $P_3$ |
| If $P_3 < p_c$ |
| Randomly Choose the combination of two more individuals in the normal set as the selected one. |
| Else |
| Randomly Choose the combination of two more individuals in the elite set as the selected one. |
| End |
| End |
| 5. **Re-initialize:** When the number of individuals stagnation updates reaches a threshold, Reinitialize the individual. |
| 6. **Update step:** Reorder the collection of parent individuals and offspring individuals using two sets of indicators. Reserve better solutions. |
| 7. **Archive updating step:** when all individuals have been generated, each new non-dominated solution obtained in current iteration will be compared with all members in the Archive. The non dominated solution will be reserved. |
Clustering Strategy

The k-means clustering algorithm [30] is used in the original BSO, however, it is just a simple clustering strategy used in grouping operation. Especially, the k-means clustering algorithm has some disadvantages: firstly, the k value needs to be given in advance, and the number of clusters has a greater impact on the clustering result; secondly, the choice of the initial clustering center is randomly selected, and once the initial clustering center is not suitable, it will have a big impact on the clustering results; thirdly, it is sensitive to the noise point. Each clustering result may be affected by the noise point. In order to address these drawbacks, we employ the objective-space clustering approach [27]. Because the main aim of clustering is to simulate the new ideas generation process, and to find the local optimum for each cluster, we can cluster them in the objective space, which can significantly save the computation time.

After the clustering operation, the individuals are divided into two clusters, i.e., the elite set and the normal set. The new individual will be generated from the elite sets or the normal sets randomly.

Disturbance Operation and Re-initialize

In each interaction, in order to simulate the random process and prevent the algorithm from falling into the local optimum prematurely, the disturbance strategy is added on the parent ideas to yield better new individuals. By the probability-based selection, the disturbance operation determines whether it will directly replace an elite individual. If done, the elite set will be changed. As a result, it is easier to jump out of the local optimum.

Another solution to prevent algorithm from stagnation prematurely is the re-initialize mechanism. Each individual will correspond to a counter during the update process, when the count reaches a threshold, the currently updated individual will be randomly initialized [31].

Generation and mutation

New individuals are generated from an old one by using Eq. (1) (2). Four different steps are used in the generation process, as showed in Algorithm 2.

Different from original BSO, after the elite-normal clustering, there are two types of clusters in the objective space, i.e., the elite sets and the normal sets. The new individual will be generated by one or more selected individuals from the elite sets or the normal sets. To increase the interaction of the individuals, more than two individuals are selected to update the whole population [32]. There are three predetermined probability in the selection choice process: \( p_a, p_b, p_c \). The assignment of the probability may have an affection on the final search results. In this paper, \( p_a \) is set as 0.8, \( p_b \) is 0.5, \( p_c \) is 0.5.

When the individuals are selected from the current interaction, the mutation operations will be performed to the selected individual. Instead of using the Gaussian random function, the Cauchy random function is adopted in MIBSO. Compared with Gauss random function, the Cauchy random function has a larger mutation scope, which lead to a greater possibility to find a set of optimal solutions. The mutation formula is shown in the Eq. (3). In which, the \( C(\mu^i, \sigma^i) \) is Cauchy random function. \( \xi^i \) is given in formula 2.

\[
x_{new}^i = x_{current}^i + \xi^i \times C(\mu^i, \sigma^i)
\]  

(3)

Environment Selection Based on Multiple Indicators

When the offspring individuals have been generated from the parent individuals, all of the individuals will be selected to maintain the population. The environment selection operation is based on two indicators, which are used to measure the dispersion and convergence of the whole population [33][34].
**Archive for Convergence based indicator:** To make individuals to be sorted based on convergence, a quality indicator $I_{\varepsilon}$ is used in the clustering operation to improve the convergence of individuals. Suppose there are two solution sets A and B [35], then $I_{\varepsilon}$ is defined as follows.

$$I_{\varepsilon}(M, N) = \min \{ \forall y \in M, \exists x \in N : f_i(x) - f_i(y) < \varepsilon, \text{for } i \in \{1, \ldots, m\} \}$$

(4)

Where the $x$ and $y$ are solutions in solution space, $m$ represents $m$ objectives in the objective space, it represents the minimum shift quantity of each dimension of the objectives. Then the final index coefficient is calculated as follows.

$$V_{I_{\varepsilon}}(x) = \sum_{y \in F(x)} e^{-I_{\varepsilon}(x, y)/0.05}$$

(5)

**Archive for Diversity based indicator:** To make individuals to be sorted based on diversity, the quality indicator $I_{\sigma}$ is used to make the individuals sorted according to the diversity. The definition is as follows.

$$I_{\sigma}(x, y) = \sqrt{\sum_{i \in \Omega} sd(f_i(x), f_i(y))^2}$$

(6)

$$I_{\sigma}(x) = \min_{y \prec x, y \text{ precedes } x} \{ I_{\sigma}(x, y) \}$$

(7)

$$sd(f_i(x), f_i(y)) = \begin{cases} f_i(y) - f_i(x) & \text{if } f_i(x) < f_i(y) \\ 0 & \text{else} \end{cases}$$

(8)

The two indicators collaborate on environmental selection through a Stochastic Ranking strategy [36]. After the environment selection operation, the individuals with better dispersion and convergence will be retained to the next generation. An external archive is used to reserve the non-dominated solutions.

**Experiments and Discussions**

**Test Functions and Experimental Setup**

The comparison result of functions is shown in Figure 1. For almost of problems, the MIBSO perform slightly better or comparable to others, and can be clearly observed from the figure that the distribution of solutions is more uniform than others, which shows the Multi-indicators mechanism works for keeping the balance of diversity and convergence.

| Problem | Value | MIBSO | NSGA II | MMOPSO | IBEA | SMPSO |
|---------|-------|-------|---------|--------|------|-------|
| ZDT1    | Mean  | 5.189e-03 (5.88e-04) | 4.754e-03 (3.94e-04) | 4.787e-03 (1.40e-04) | 4.610e-03 (1.32e-04) | 4.733e-03 (5.15e-05) |
| ZDT2    | Mean  | 4.192e-03 (7.29e-05) | 4.770e-03 (5.01e-04) | 5.077e-03 (2.18e-04) | 8.815e-03 (2.27e-04) | 5.232e-03 (1.05e-03) |
| ZDT3    | Mean  | 5.234e-03 (1.28e-04) | 5.397e-03 (3.70e-05) | 5.938e-03 (1.64e-04) | 3.454e-02 (1.41e-02) | 5.062e-03 (1.01e-04) |
| ZDT4    | Mean  | 1.113e-02 (1.89e-03) | 1.158e-02 (6.56e-04) | 7.662e-02 (9.72e-02) | 1.881e-01 (1.96e-01) | 1.032e+01 (8.75e+0) |

In order to verify the performance of proposed multi-objective BSO algorithm, some comparative experiments were set. The algorithm for comparison is: NSGA-II [1], MMOPSO, IBEA, SMPSO.
Four benchmark test problems are used to validate the performance: ZDT1, ZDT2, ZDT3 and ZDT4. Each problem has two objectives. The Inverse Generational Distance (IGD) metric is used as the testing standard [37] [38]. The iteration in one run is 1000; Each test function was run 30 times, and the average values and standard deviation of the test function were recorded in the Table 1.

![Figure 2. The obtained Pareto fronts contrast curve of all algorithm in ZDT1,ZDT2,ZDT3,ZDT4.](image)

**Results and Analysis**

The mean and standard deviation values of IGD by the four algorithms over 30 independent runs are given in Table 1. From the table, it is clearly observed that MIBSO performs more powerfully than most of other peer algorithms, mainly due to its adopted multi-indicator cooperation strategies.

From the Table 1, it can be found MIBSO have a better result on most benchmark functions. MIBSO performs best on ZDT2 and ZDT4, SMPSO shows the best performance on ZDT3, MIBSO slightly worse than it, but also better than the others. IBEA is a classic indicator-based algorithm, only one indicator $I_\epsilon$ is applied to the algorithm, which prefers convergence to diversity. It performs better than others on ZDT1, but do bad on other problems. Especially, the diversity of the solutions is worse than MIBSO, which means the MIBSO do well on balance the diversity and convergence.

**Summary**

In this paper, a new proposed multi-objective brainstorm optimization based on two indicators called MIBSO is developed to solve the MOPs. Multiple indicators are used to select better individuals, which can enhance the diversity and convergence of the algorithm synergistically. The use of clustering in objective space can reduce computational complexity and make individuals move toward the better results and finally get better individuals. Experimental results show that the MIBSO is relatively prominent or at least comparable to some of the contrast algorithm in terms of IGD. As can be seen MIBSO can be a reliable algorithm to solve multi-objective problem.
Acknowledgement
This work is supported by the National Natural Science Foundation of China under Grant No. 61773103 and 61503373; and Fundamental Research Funds for the Central Universities No. N161705001.

References
[1] Horn J, Nafpliotis N, Goldberg D E. A Niched Pareto Genetic Algorithm for Multiobjective Optimization[M]// A niched Pareto genetic algorithm for multiobjective optimization. 1994:82-87 vol.1.

[2] L. Ma, S. Cheng, X. Wang, M. Huang, H. Shen, X. He, Y. Shi. Cooperative Two-engine Multi-objective Bee Foraging Algorithm with Reinforcement Learning, Knowledge-Based Systems, 133(2017):278-293, 2017.

[3] L. Ma, K. Hu, Y. Zhu, H. Chen. Cooperative Artificial Bee Colony Algorithm for Multi-objective RFID Network Planning, Journal of network and computer applications, 42(2014)143-162.

[4] L. Ma, K. Hu, Y. Zhu, H. Chen, A Hybrid Artificial Bee Colony Optimizer by combining with life-cycle, Powell’s search and crossover, Applied Mathematics and Computation, vol.252, pp.133–154,2015.

[5] L. Ma, X. Wang, M. Huang, Z. Lin, L. Tian, H. Chen. Two-level Master-slave RFID Networks Planning via Hybrid Multi-objective Artificial Bee Colony Optimizer, IEEE Transactions on Systems, Man, and Cybernetics: Systems, PP (99):1-20. 2017.

[6] L. Ma, Y. Zhu, D. Zhang, B. Niu. A hybrid approach to artificial bee colony algorithm, Neural Computing & Applications, 2016, 27(2), 387-409.

[7] Chen H, Ma L, He M, et al. Artificial Bee Colony Optimizer Based on Bee Life-Cycle for Stationary and Dynamic Optimization[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2017, 47(2):327-346.

[8] L. Ma, K. Hu, Y. Zhu, H. Chen. Computational evolution of social behavior in bacterial colony optimization model. Journal of pure and applied microbiology, 2013, 7: 487-493.

[9] H. Jiang, W. Dong, L. Ma, R Wang. Bacterial Foraging Algorithm Based on Reinforcement Learning for Continuous Optimizations, International Symposium on Intelligence Computation and Applications, 41-52, 2017

[10] Kennedy J, Eberhart R. Particle swarm optimization[C]// IEEE International Conference on Neural Networks, 1995. Proceedings. IEEE, 2002:1942-1948 vol.4.

[11] L. Ma, K. Hu, Y. Zhu, B. Niu, H. Chen. Discrete and Continuous Optimization Based on Hierarchical Artificial Bee Colony Optimizer, Journal of applied mathematics, Vol.2014, Article ID 402616, 2014.

[12] L. Ma, H. Chen, Y. Zhu, K. Hu. Hierarchical Artificial Bee Colony Algorithm for RFID Network Planning Optimization, Scientific World Journal, vol. 2014, Article ID 941532,2014.

[13] Y. Liu, L. Ma, G. Yang, A Survey of Artificial Bee Colony Algorithm,2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 2017 :1510-1515

[14] Coello C A C. Evolutionary multi-objective optimization: a historical view of the field[M]. IEEE Press, 2006.
[15] Shi Y. Brain Storm Optimization Algorithm[ C ]//International Conference in Swarm Intelligence. Springer, Berlin, Heidelberg, 2011:303-309.

[16] L. Ma, Y. Zhu, D. Zhang, B. Niu. A hybrid approach to artificial bee colony algorithm, Neural Computing & Applications, 2016, 27(2), 387-409.

[17] H. Chen, L. Ma*, M. He, X. Wang*, Artificial Bee Colony Optimizer Based on Bee Life-cycle for Stationary and Dynamic Optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016:1-20.

[18] L. Ma, K. Hu, Y. Zhu, H. Chen. Improved Multi-objective Artificial Bee Colony Algorithm for optimal power flow problem, Journal of Central South University, 2014, 21 (11): 4220-4227.

[19] L. Ma, H. Chen, Y. Zhu, K. Hu. Hierarchical Artificial Bee Colony Algorithm for RFID Network Planning Optimization, Scientific World Journal, vol. 2014, Article ID 941532, 2014.

[20] L. Ma, K. Hu, Y. Zhu, B. Niu, H. Chen. Discrete and Continuous Optimization Based on Hierarchical Artificial Bee Colony Optimizer, Journal of applied mathematics, Vol.2014, Article ID 402616, 2014.

[21] Shi Y. An Optimization Algorithm Based on Brainstorming Process[ C ]//International Symposium on Intelligence Computation & Applications. Huazhong University of Science and Technology, 2008:35-62.

[22] Cheng S, Qin Q, Chen J, et al. Brain storm optimization algorithm: a review[J]. Artificial Intelligence Review, 2016, 6728(3):1-14.

[23] H.B. Duan, S.T. Li and Y.H. Shi, Predator–prey brain storm optimization for DC brushless motor, IEEE Trans. Magn., vol. 49, no. 10, pp. 53368-5340, Oct. 2013.

[24] Z.W. Zhou, H.B. Duan, P. Li and B. Di, Chaotic differential evolution approach for 3D trajectory planning of unmanned aerial vehicle,” in Proc. IEEE Int. Conf. Control Autom., Hangzhou, China, Jun. 12-14, 2013, pp. 368-372.

[25] Duan H, Li C. Quantum-Behaved Brain Storm Optimization Approach to Solving Loney’s Solenoid Problem[J]. Magnetics IEEE Transactions on, 2015, 51(1):1-7.

[26] Cheng S , Sun Y , Chen J , et al. A Comprehensive Survey of Brain Storm Optimization Algorithms[C]// Evolutionary Computation. IEEE, 2017.

[27] Shi Y. Brain storm optimization algorithm in objective space[C]// Evolutionary Computation. IEEE, 2015:303-309.

[28] Yao X, Burke E K, Lozano, José A, et al. [Lecture Notes in Computer Science] Parallel Problem Solving from Nature - PPSN VIII Volume 3242 || Indicator-Based Selection in Multiobjective Search[J]. 2004, 10.1007/b100601 (Chapter 84):832-842.

[29] M. Li, S. Yang, and X. Liu, “Shift-based density estimation for Pareto-based algorithms in many-objective optimization,” IEEE Trans. Evol. Comput., vol. 18, no. 3, pp. 348–365, Jun. 2014.

[30] Lewis R J, Krawiec M, Confer E, et al. Data clustering: 50 years beyond K-means[J]. Pattern Recognition Letters, 2014, 5211(8):3-4.

[31] El-Abd M. Brain storm optimization algorithm with re-initialized ideas and adaptive step size[C]// Evolutionary Computation. IEEE, 2016.

[32] Xue J, Wu Y, Shi Y, et al. Brain storm optimization algorithm for multi-objective optimization problems[C]// International Conference on Advances in Swarm Intelligence. Springer-Verlag, 2012:513-519.
[33] L. Ma, Li X, Gao T, et al. Indicator-Based Multi-objective Bacterial Foraging Algorithm with Adaptive Searching Mechanism[C]//Bio-Inspired Computing-Theories and Applications. Springer Singapore, 2016:271-277.

[34] Chen S, Wang R, Ma L, et al. A Novel Many-Objective Bacterial Foraging Optimizer Based on Multi-engine Cooperation Framework[C]//International Conference on Swarm Intelligence. Springer, Cham, 2018.

[35] Wang R, Chen S, Ma L, et al. Multi-indicator Bacterial Foraging Algorithm with Kriging Model for Many-Objective Optimization[M]//Advances in Swarm Intelligence. 2018.

[36] Li B, Tang K, Li J, et al. Stochastic Ranking Algorithm for Many-Objective Optimization Based on Multiple Indicators[J]. IEEE Transactions on Evolutionary Computation, 2016, PP(99):1-1.

[37] L. Ma, X. Wang, M. Huang, H. Zhang, H. Chen. A Novel Evolutionary Root System Growth Algorithm for Solving Multi-objective Optimization Problems, Applied Soft Computing, 2017, 57:379-398.

[38] L. Ma, Y. Zhu, Y. Liu, L. Tian, et al. A novel bionic algorithm inspired by plant root foraging behaviors, Applied Soft Computing, 37(2015):95-133.