BSO-CLS: Brain Storm Optimization Algorithm with Cooperative Learning Strategy

Liang Qu¹, Qiqi Duan¹, Jian Yang¹, Shi Cheng², Ruiqi Zheng¹,
and Yuhui Shi¹(✉)

¹ Department of Computer Science and Engineering,
Southern University of Science and Technology, Guangdong 518055, China
qul@mail.sustech.edu.cn, shiyh@sustech.edu.cn

² School of Computer Science, Shaanxi Normal University, Xi'an 710062, China

Abstract. Brain storm optimization algorithms (BSO) have shown great potential in many global black-box optimization problems. However, the existing BSO variants can suffer from three problems: (1) large-scale optimization problem; (2) hyperparameter optimization problem; (3) high computational cost of the clustering operations. To address these problems, in this paper, we propose a simple yet effective BSO variant named Brain Storm Optimization Algorithm with Cooperative Learning Strategy (BSO-CLS). It is inspired by the new ideas generating process of brain storm in which the participators propose their own ideas by cooperatively learning other participators' ideas. Thus, BSO-CLS iteratively updates the candidate solutions by linearly combining other solutions with the weights deriving from the fitness values of other solutions. To validate the effectiveness of the proposed method, we test it on 6 benchmark functions with the 1000 dimensions. The experimental results show that BSO-CLS can outperform the vanilla BSO and the other BSO variant with the learning strategy.

Keywords: Cooperatively learning · Brain storm optimization · Large scale optimization

1 Introduction

Large scale black-box optimization problems play a very important role in many scientific and engineering applications. It can be formulated as follows [12]:

$$x^* = \arg \min_{x \in R^D} f(x) \quad (1)$$

where x^* is the global minimizer to be searched and $f : R^D \rightarrow R$ is the evaluation function. In this paper, we focus on the large scale black-box optimization problem [11], i.e., the evaluation function f typically has three properties:

- **Black-box:** When a candidate solution $x \in R^D$ is evaluated by f , we can only obtain the function evaluation value $f(x)$. Any other information of f , such as gradient information, is unavailable.
- **High-dimension:** The dimension D of the candidate solution x is typically larger than or equal to 1000 (i.e. $D \geq 1000$).
- **Multimodal:** The evaluation function f can have many local optima and global optima solutions.

Recently, swarm intelligence (SI) [6, 7] has been a promising technique to solve the large scale black-box optimization problems [2, 8]. Most existing SI algorithms such as particle swarm optimization (PSO) [14] and ant colony optimization (ACO) [5] are inspired by the swarm behaviours of the simple creatures in nature. However, BSO [3, 4, 15], a young and promising SI algorithm, is inspired by the brainstorming process of the high-level human being, which utilizes the clustering operation to simulate the brainstorming process and generates the new candidate solutions by learning other solutions from intercluster or intracluster randomly. In general, the existing BSO algorithms can suffer from three problems: (1) cannot obtain good performance on the high-dimension optimization problems; (2) several hyperparameters of algorithm itself need to be optimized; (3) the clustering operation (e.g. k-means [10]) is high computational cost.

To address the aforementioned problems, in this paper, we propose a novel BSO variant named BSO-CLS. It is inspired by the participators' ideas generating process of brain storm, i.e., each new idea is generated by cooperatively learning the ideas from other participators. Thus, for each candidate solution of BSO-CLS, it iteratively updates itself by linearly combining other solutions with the weights deriving from the function evaluation values of other solutions.

The contributions of this paper are summarized as follows:

- We introduce a novel cooperative learning strategy to simulate the new ideas generating process of brain storm. The analysis and experiment show that the novel learning strategy is effective and efficient.
- In order to validate the effectiveness of the proposed method, we test it on 6 benchmark functions with the 1000 dimensions. The empirical results show that BSO-CLS can outperform the vanilla BSO and the other BSO variant with the learning strategy.

The rest of this paper is organized as follows. Section 2 introduces the representative BSO variants. Section 3 introduces BSO-CLS in detail, and Sect. 4 presents and discusses the experimental setting and results. The conclusions are drawn in Sect. 5.

2 Related Work

In this section, we will introduce some representative BSO variants. In order to reduce the high computational cost of clustering operations, BSO-OS [13]

Fig. 1. The processes of the new ideas proposed in the brain storm.

proposed to cluster individuals in the objective space instead of in the solution space. MBSO [17] proposed a modified BSO which introduced the simple grouping method (SGM) with the idea difference strategy (IDS). RGBSO [1] proposed to utilize the random grouping strategy in clustering.

Recently, some learning based new solutions generating strategies have been proposed. BSOLS [16] proposed a novel learning strategy after updating operator to improve the diversity of population. OLBBSO [9] introduced an orthogonal learning (OL) framework and incorporated it with BSO to improve the performance.

3 Proposed Method

In this section, we will introduce the proposed method in detail. The key idea of BSO-CLS is that it utilizes a novel method to simulate the new ideas generating process of the brain storm. Specifically, as shown in Fig. 1, during the brain storm process, the participants generally go through three processes to cooperatively learn other participants' ideas before proposing their own ideas. Taking participant $P0$ as an example, the first process is to listen to other participants' ideas and select the ideas that inspire him. The second process is to analyze the selected ideas. Some of them have a greater impact on him, and others have a smaller impact on him. The third process is to come up with his own ideas based on his analysis and share them with other participants. Thus, other participants can follow these three processes to come up with their own ideas until the brain storm is finished.

Based on the processes introduced above, the BSO-CLS can be implemented as follows:

Step 1: Randomly initialize N candidate solutions $X \in R^{P \times D}$ with D dimension.

Step 2: Evaluate the fitness $Y = f(X)$ using the evaluation function f .

Step 3.1: For each candidate solution, randomly select m other candidate solutions $x_1, x_2, \dots, x_m$ as the learning objects.

Step 3.2: For each learning object x_i , calculate the learning weight w_i using the formula (2) by normalizing the fitness values $f(x_1), f(x_2), \dots, f(x_m)$ of the learning objects.

$$w_i = \frac{f(x_i) - \mu}{\sigma} \quad (2)$$

where μ and σ are the mean and standard deviation of the fitness values $f(x_1), f(x_2), \dots, f(x_m)$ respectively.

Step 3.3: Generate the new candidate solution x_{new} using the formula (3) and evaluate its fitness value.

$$x_{new} = \sum_{i=1}^n w_i x_i \quad (3)$$

Step 3.4: If N new solutions have not been generated, go to Step 3.1.

Step 4: If the stop condition is not met, return Step 3.1.

The procedures of BSO-CLS have been given in the Algorithm 1.

Algorithm 1. BSO-CLS

- 1: Randomly initialize N candidate solutions $X \in R^{P \times D}$ with D dimension
 - 2: Evaluate the fitness $Y = f(X)$ using the evaluation function f
 - 3: **while** not stopping condition **do**
 - 4: **for** $i = 1$ to P **do**
 - 5: Randomly select m candidate solutions $x_1, x_2, \dots, x_m$ as learning objects and their fitness values $f(x_1), f(x_2), \dots, f(x_m)$.
 - 6: Calculate the learning weights $w_1, w_2, \dots, w_m$ using formula (2)
 - 7: Generate the new candidate solution according to the formula (3)
 - 8: **end for**
 - 9: **end while**
-

4 Experiments

In this section, we will present and discuss the experimental settings and results.

4.1 Benchmark Functions

To validate the effectiveness of the BSO-CLS, we test it on the 6 benchmark functions including the unimodal function (Sphere) and the multimodal functions (Rosenbrock, Ackley, Rastrigin, Schwefel, and Ellipsoid). Furthermore, in order to test the proposed method's performance on the various dimensions, we vary the dimensions from 10 to 1000 for all benchmark functions. The information of the benchmark functions are summarized in Table 1.

Table 1. The information of benchmark functions

Benchmark function	Function	Search space
Sphere	$f(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$[-5.12, 5.12]^D$
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^n [10(x_{i+1} - x_i^2) + (1 - x_i)^2]$	$[-5, 10]^D$
Ackley	$f(\mathbf{x}) = -20 \cdot \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + \exp(1)$	$[-32, 32]^D$
Rastrigin	$f(\mathbf{x}) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$	$[-5.12, 5.12]^D$
Schwefel	$f(\mathbf{x}) = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	$[-500, 500]^D$
Ellipsoid	$f(\mathbf{x}) = \sum_{i=1}^n 10^{6 \frac{i-1}{n-1}} x_i^2$	$[-100, 100]^D$

4.2 Comparison Method

This paper aims to propose a novel learning mechanism to generate the new candidate solutions for BSO algorithms. Therefore, we choose the most representative SI algorithm PSO [14], the original BSO algorithm [15] and the other BSO variant BSOLS with learning strategy [16]. The hyperparameters setting for each algorithm is given in Table 2.

Table 2. The hyperparameters setting for each algorithm

Algorithm	Hyperparameters setting
BSO	$k = 20$, $\mu = 0$, $\sigma = 1$, $p_{replace} = 0.2$, $p_{one} = 0.2$, $N = 100$, $M = 5$, $p_{onecluster} = 0.4$, $p_{twocluster} = 0.5$
PSO	$w = 0.9 \sim 0.4$, $c_1 = c_2 = 2.0$, $V_{MAXd} = 0.2 \times \text{Range}$, $N = 40$, global version
BSO-LS	$N = 100$, $m = 5$, $P_{replace} = 0.2$, $P_{one} = 0.8$, $P_{onecluster} = 0.4$, $P_{twocluster} = 0.5$, $P_{dynamic} = 0.5$
BSO-CL	$m = 10$

4.3 Experimental Setting

In order to fairly compare all algorithms, they are tested with the same population size 100 and the same maximum number of function evaluations $1E4 \times D$ in each run for each test function, where D is the dimension. In order to reduce the statistical error, each algorithm is tested 30 times independently, and the mean and standard deviation (std) are reported for comparison.

4.4 Results and Discussion

The experimental results are given in Table 3, and the best results are marked with **bold**.

Table 3. Solutions accuracy (mean and std) comparisons.

Funtions		BSO		PSO		BSO-LS		BSO-CLS	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
Sphere	10	3.66e-34	7.12e-34	5.90e-42	5.87e-42	0.00e+00	0.00e+00	1.74e-06	2.67e-06
	50	4.12e-12	9.06e-12	2.18e-13	8.22e-13	6.61e-12	5.12e-12	3.21e-05	4.89e-05
	100	1.45e+01	2.56e+01	5.23e+01	8.11e+01	2.11e+00	5.61e+00	6.42e-05	8.82e-05
	500	3.80e+03	4.77e+04	5.12e+03	9.01e+03	7.24e+03	6.24e+03	1.60e-02	3.17e-02
	1000	7.25e+04	3.21e+04	7.13e+04	8.23e+04	5.15e+04	8.14e+04	1.83e-01	2.84e-01
Rosenbrock	10	2.21+00	3.78+00	1.21e+00	2.42e+00	9.66e+00	2.22e+00	1.10e-05	2.52e-05
	50	4.56+01	5.21+e01	7.92e+01	9.21e+01	8.89e+01	6.12e+01	1.95e-06	3.69e-06
	100	6.13e+02	4.45e+01	2.98e+02	8.87e+01	9.55e+02	6.73e+02	2.18e-07	2.94e-07
	500	2.38e+04	5.02e+02	6.66+e04	7.78e+02	3.91e+04	6.67e+04	1.17e-07	1.71e-07
Ackley	1000	8.32e+05	4.18e+03	6.33e+05	9.12e+03	7.43e+05	1.10e+03	3.81e-08	5.76e-08
	10	5.71e-05	7.98e-05	8.75e-05	2.20e-05	9.13e-06	4.32e-06	7.31e-03	6.49e-03
	50	3.22e+00	2.18e+00	7.11e+00	9.10e+00	6.28e+00	8.13e+00	1.71e-01	1.67e-01
	100	5.13e+02	6.33e+02	9.12e+02	6.23e+02	1.22e+02	5.11e+01	3.17e-01	2.48e-01
	500	7.13e+03	6.12e+03	5.55e+03	8.12e+03	9.12e+03	7.18e+03	1.15e+00	1.95e-01
Rastrigin	1000	2.26e+04	5.21e+02	7.71e+04	7.91e+02	6.32e+04	8.12e+01	1.17e+00	2.57e-01
	10	4.22e-01	5.99e-01	4.21e+00	1.10e+00	8.32e-01	7.22e-01	1.43e-03	2.86e-03
	50	8.81e+01	7.21e+01	2.27e+01	4.32e+01	7.21e+01	3.33e+01	1.45e-01	2.25e-01
	100	6.44e+03	4.21e+02	5.07e+03	8.89e+02	9.22e+03	8.84e+02	7.78e-01	1.33e+00
	500	3.21e+04	7.21e+02	3.37e+04	1.09e+02	8.50e+04	7.79e+02	4.24e+01	2.15e+01
Schwefel	1000	7.71e+06	5.22e+04	5.03e+06	9.91e+05	7.68e+06	3.28e+04	8.68e+01	3.47e+01
	10	7.21e-02	8.84e-02	5.53e-02	4.77e-02	9.01e-03	4.32e-03	8.98e-04	1.41e-03
	50	6.11e+00	3.25e+00	5.88e+00	4.32e+00	7.76e+00	2.43e+00	2.74e-03	3.86e-03
	100	1.13e+02	6.61e+01	7.87e+01	4.11e+02	5.78e+01	6.54e+01	4.37e-03	4.93e-03
	500	4.28e+03	5.55e+02	7.14e+03	5.27e+02	9.01e+03	2.34e+02	2.45e-02	3.70e-02
Ellipsoid	1000	1.98e+04	9.31e+04	3.16e+04	2.19e+03	8.13e+02	5.26e+02	6.48e-02	7.57e-02
	10	1.45e+00	2.03e+00	5.54e+01	3.22e+00	5.17e+00	6.63e+00	9.73e-01	1.45e+00
	50	7.36e+02	2.54e+02	8.94e+02	6.58e+02	4.13e+02	3.69e+02	8.02e+00	1.22e+00
	100	6.62e+03	5.17e+02	7.96e+02	3.88e+03	5.25e+02	7.13e+02	1.56e+01	2.90e+01
	500	4.13e+04	8.83e+03	1.48e+04	9.22e+04	3.66e+03	7.87e+03	2.51e+02	3.09e+02
	1000	5.16e+05	2.86e+04	5.22e+05	1.18e+03	8.96e+05	7.41e+04	1.11e+03	1.13e+03

As shown in Table 3, we can clearly observe that: (1) The proposed method can obtain competitive performance on the low dimension problems against other methods. (2) The proposed method outperforms all the comparison methods on the high dimension problems. (3) With the dimension increasing, the proposed method obtains better results on the Rosenbrock test function.

5 Conclusion

In this paper, we proposed a novel BSO variant named BSO-CLS, which introduced a simple yet effective cooperative learning mechanism. The key concept is inspired by the new ideas generating process of brain storm which the participants propose their own ideas by cooperatively learning other participants'

ideas. Thus, BSO-CLS iteratively updates the candidate solutions by linearly combining other solutions with the weights deriving from the fitness values of other solutions. The effectiveness of the proposed method was validated on 6 benchmark functions for both low-dimension and high-dimension settings. The empirical results showed that BSO-CLS can outperform the vanilla BSO and the other BSO variant with the learning strategy especially for the large-scale optimization problems.

In the future, we plan to validate the proposed method on more complex problems, such as the rotation and shifted benchmark functions. We intend to explore the reasons behind the better performance on Rosenbrock function with the dimension increasing. Furthermore, we will test the influence of different normalization functions.

Acknowledgement. This work is supported by the National Science Foundation of China under the Grant No. 61761136008, the Shenzhen Peacock Plan under the Grant No. KQTD2016112514355531, the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under the Grant No. 2017ZT07X386, the Science and Technology Innovation Committee Foundation of Shenzhen under the Grant No. ZDSYS201703031748284, Guangdong Provincial Key Laboratory under Grant No. 2020B121201001.

References

1. Cao, Z., Shi, Y., Rong, X., Liu, B., Du, Z., Yang, B.: Random grouping brain storm optimization algorithm with a new dynamically changing step size. In: Tan, Y., Shi, Y., Buarque, F., Gelbukh, A., Das, S., Engelbrecht, A. (eds.) ICSI 2015. LNCS, vol. 9140, pp. 357–364. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20466-6_38
2. Cheng, R., Jin, Y.: A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybern.* **45**(2), 191–204 (2014)
3. Cheng, S., Qin, Q., Chen, J., Shi, Y.: Brain storm optimization algorithm: a review. *Artif. Intell. Rev.* **46**(4), 445–458 (2016). <https://doi.org/10.1007/s10462-016-9471-0>
4. Cheng, S., Shi, Y. (eds.): Brain Storm Optimization Algorithms. ALO, vol. 23. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-15070-9>
5. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**(4), 28–39 (2006)
6. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence. Elsevier (2001)
7. Kennedy, J.: Swarm intelligence. In: Zomaya, A.Y. (ed.) Handbook of Nature-Inspired and Innovative Computing, pp. 187–219. Springer, Boston (2006). https://doi.org/10.1007/0-387-27705-6_6
8. Li, X., Yao, X.: Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans. Evol. Comput.* **16**(2), 210–224 (2011)
9. Ma, L., Cheng, S., Shi, Y.: Enhancing learning efficiency of brain storm optimization via orthogonal learning design. *IEEE Trans. Syst. Man Cybern. Syst.* (2020)
10. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, vol. 1, pp. 281–297 (1967)

11. Omidvar, M.N., Li, X., Mei, Y., Yao, X.: Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Trans. Evol. Comput.* **18**(3), 378–393 (2013)
12. Qian, H., Hu, Y.Q., Yu, Y.: Derivative-free optimization of high-dimensional non-convex functions by sequential random embeddings. In: *IJCAI*, pp. 1946–1952 (2016)
13. Shi, Y.: Brain storm optimization algorithm in objective space. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1227–1234, May 2015. <https://doi.org/10.1109/CEC.2015.7257029>
14. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), pp. 69–73, May 1998. <https://doi.org/10.1109/ICEC.1998.699146>
15. Shi, Y.: Brain storm optimization algorithm. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) *ICSI 2011*. LNCS, vol. 6728, pp. 303–309. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21515-5_36
16. Wang, H., Liu, J., Yi, W., Niu, B., Baek, J.: An improved brain storm optimization with learning strategy. In: Tan, Y., Takagi, H., Shi, Y. (eds.) *ICSI 2017*. LNCS, vol. 10385, pp. 511–518. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61824-1_56
17. Zhan, Z., Zhang, J., Shi, Y., Liu, H.: A modified brain storm optimization. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8. IEEE (2012)