



Hybrid Bacterial Forging Optimization Based on Artificial Fish Swarm Algorithm and Gaussian Disturbance

Ruozhen Zheng¹, Zhiqin Feng¹, Jiaqi Shi^{2(✉)}, Shukun Jiang^{3(✉)},
and Lijing Tan⁴

¹ College of Management, Shenzhen University, Shenzhen, China

² College of Economy, Shenzhen University, Shenzhen, China
szdsjq@163.com

³ College of Software, Shenzhen University, Shenzhen, China
15625562736@163.com

⁴ College of Management, Shenzhen Institute of Information Technology,
Shenzhen, China

Abstract. Traditional Bacterial Forging Optimization (BFO) has poor convergence speed and is easily trapped in the local optimum while dealing with some complex problems. Facing these disadvantages, a new hybrid algorithm for BFO based on Artificial Fish Swarm Algorithm (AFSA) and Gaussian disturbance is proposed, abbreviated as AF-GBFO. The algorithm combines following and swarming behaviors in AFSA with the chemotaxis part of BFO so that bacteria can update positions by evaluating the value of their own and others positions. The convergence speed can be improved in this way. The algorithm also combines Gaussian disturbance to change bacteria's positions by adding a number following Gaussian distribution. In that case, if all bacteria gather around the local optimum, they still have chance to get out of it. Meanwhile the elimination-dispersal way has been changed to have half of the bacteria eliminated and keep the positions with good values so that the convergence speed is increased. Compared with original BFO, GA, BFOLIW and BFONIW, AF-GBFO outperforms in most cases especially for the multimodal functions.

Keywords: Bacterial Forging Optimization · Artificial Fish Swarm Algorithm · Gaussian disturbance

1 Introduction

In 2002, Passino [9] proposed an optimization algorithm for bacterial foraging based on the behavior of *E. coli* in human intestinal tract to devour food. The algorithm simulates three kinds of behavior, the chemotaxis of *E. coli*, reproduction and elimination-dispersal. It holds great global searching ability and can search in parallel. Because of its efficiency, it is widely used in engineering and

science, such as PID controller design [7], power network [8] and others. However, it has poor convergence rate and is easily trapped in local minimum while dealing with some complex problems [10].

To address problem of poor convergence, researchers have developed many optimization algorithms. For example, Gupta [6] combined PSO algorithm with BFO algorithm to improve the convergence of mixed BF-PSO algorithm. Chen [2] proposed the BCF algorithm and added adaptive and cooperative modes to BFO. Wang [12] proposed the BBBFO algorithm, which adopted convergent strategy of gaussian distribution used the historical information of individuals and the shared information to promote the convergence, and enhanced the diversity of replicated groups. Feng [5] proposed ES-ABFO algorithm, which adopts adaptive chemotaxis. Dass [3] adopts adaptive step size and reproductive mechanism with different birth locations of clones. Dasgupta [4] proposed two schemes for adaptive chemotaxis step level. However, with convergence speed increasing, premature convergence occurs. Previous researches have failed to consider original global searching ability as they try to improve convergence performance. Therefore, the purpose of this paper is to enhance the convergence of BFO while keeping the global searching ability.

AFSA has a fast convergence speed while BFO has a strong global search ability. Inspired by BFO-AFSA proposed by Teng [11] that adopt the chemotaxis of BFO to AFSA to settle the problem of being caught in the local optimum in AFSA, AFSA clustering mechanism is used in new algorithm. The distinct features of AF-GBFO can be listed as follows: (1) Following and swarming behaviors as essential information mechanisms are introduced to the chemotaxis part. Bacteria change their current positions by the best position information in following behavior and central position information in swarming behavior so that they converge more quickly. (2) Gaussian disturbance is added to adjust former positions randomly to alleviate the premature convergence caused by convergence speed improving and balance the strategies.

This paper is organized as follows: The original BFO and AFSA are described in Sects. 2 and 3 specifies the improved algorithm. Following Sect. 4 shows the experimental results. Finally, Sect. 5 summarizes the whole paper.

2 Related Work

2.1 Bacterial Foraging Optimization

The bacterial foraging optimization algorithm was firstly proposed by Passino [9] in 2002. It simulates the behavior of *E. coli* searching for food in the human intestine. This process contains three main behavioral patterns: chemotaxis, reproduction, elimination and dispersal. The specific process is as follows:

Chemotaxis. *E. coli* has two behavioral patterns flagellum during chemotaxis: runs and tumble. First calculate the fitness value $J_{health}(j, k, l)$ of the bacteria, store the bacteria i as the best value of the current fitness, rotate and compare

the fitness value before and after the rotation. If it is better, then continue to swim a unit in this direction, otherwise, jump out directly Chemotactic cycle.

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T \Delta(i)}} \quad (1)$$

where: $\theta^i = [\theta_1^i, \theta_2^i, \dots, \theta_d^i]$ is location information of bacteria. j is step of chemotaxis, k is step of reproduction and l is step of elimination and dispersal. $C(i)$ is the unit length of each run, $\Delta(i)$ is the direction angle of j^{th} step.

Reproduction. During the breeding process, the bacteria follow the principle of survival of the fittest, and the bacteria are arranged in ascending order according to J_{health}^i . Only top 50 of the bacteria can survive and the copying operation will keep the population size unchanged. A new generation of individuals enter the next cycle:

$$J_{health}^i = \sum_{j=1}^{Nc+1} J(i, j, k, l) \quad (2)$$

where: J_{health}^i , Nc and J are value, chemotaxis steps and fitness function.

Elimination and Dispersal. Replication will reduce the diversity of the population. In order to avoid premature convergence caused by reproduction, the BFO algorithm adds the dispelling behavior. Some individuals in are randomly regenerated in space, and a new generation of individuals enters a new chemotactic loop. Dispersal or not is determined by a given possibility P_{ed} .

2.2 Artificial Fish Swarm Algorithm

Based on imitations of natural fishes' foraging behaviors, Artificial Fish Swarm Algorithm (AFSA) was proposed by Xiaolei et al. [13] in 2002. In water, each fish in the swarm is swimming through in search of their group's food. Similarly, each fish in AFSA with stochastic initial setting such as $X = (x_1, x_2, x_3, \dots, x_n)$ is a potential solution to the problem to be solved and refers to the position of fish in certain dimensions; $Y = f(X)$ indicates the food concentration of local position; Y is the final optimal value. There are other parameters, such as d_{ij} (the distance between two fishes), step (the biggest movement of fish swimming), T (the numbers of attempts) and δ (a crowd factor). In AFSA, fishes search for optimization goal with three major behaviors preying, swimming and swarming. The description is as follows:

Preying Behavior. It is one of behaviors of fishes looking for food. Firstly, randomly initialize a fish's position X_i and choose another position X_j as well within the fish's visual scope with the fitness function. Then compute the food concentration of these two positions (Y_j and Y_i). If Y_j is larger than Y_i , fish will move one step toward X_j with formula (4). Otherwise, repeat selecting and comparing until T times. If Y_i is still larger than Y_j , X_i will be replaced by a random position within the visual scope using formula (3)

$$X_j = X_i + Visual * Rand. \quad (3)$$

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{||X_j - X_i^t||} * Step * Rand \quad (4)$$

where: *Visual* represents the field of view. *Rand* is a random number between -1 to 1. *Step* is the size of fish's step.

Swarming Behavior. In the process of swarming, fish searches other fishes and get the central position X_c . If $Y_c/n_f > \delta Y_i$ (n_f refers to the numbers of fishes inside the visual scope of bacteria (i) is satisfied, the fish will take a step to X_c with formula as follows. Otherwise, still preying.

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{||X_c - X_i^t||} * Step * Rand \quad (5)$$

where: *Step* is the size of step. *Rand* is a random number between -1 to 1.

Following Behavior. Fish looks for the fish j with best fitness value in the its sight. If the condition $Y_c/n_f > \delta Y_i$ (n_f is met, it will move one step to the position of the best fish X_i or else it continues preying behavior.

Artificial Fish Swarm Algorithm performs well in solving optimization problems with fast convergence speed and easy-adjusted parameters [1, 13]. And it is applied to machine learning [14] and other fields.

3 Based on AFSA and Gaussian Disturbance

This study incorporates swarming and following behaviors of AFSA in BFO to improve BFO searching performance with Gaussian disturbance and new elimination-dispersal way. The modified BFO is named AF-GBFO. Pseudo codes for AF-GBFO are shown in Algorithm 1.

3.1 Following and Swarming Behaviors

Chemotaxis is the most important part of BFO. In original BFO, there is no information sharing or learning during chemotaxis. In that case, each bacterium just searches for food positions randomly or by its own experience, which limits bacteria's performance. In order to improve the original BFO, AF-GBFO follows the principle of cohesion in following behavior: finds the position of the best bacteria within its visual scope. As shown in Fig. 1(a), in following part, according to the fitness value, the bacteria finds the position of best bacteria within its visual scope and takes one step tumble toward it along with uncertain steps of swimming. The formula can be described as:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + \frac{\theta^p(j, k, l) - \theta^i(j, k, l)}{\|\theta^p(j, k, l) - \theta^i(j, k, l)\|} * C(i) \quad (6)$$

where: $\theta^i(j, k, l)$ refers to the position of bacteria i after j^{th} chemotaxis, k^{th} reproduction and l^{th} dispersal. And $\theta^p(j, k, l)$ is the position of best bacteria in the visual scope. C^i refers to the length of step.

Meanwhile, theory of swarming behavior is added to chemotaxis. As shown in Fig. 1(b), after tumbling and swimming, the bacteria calculates the distances again to find the positions of bacteria whose fitness is better and then get the central position of this colony. In this method, another parameter is introduced to ensure the central position not too crowded. The equation is as follows.

$$\theta^i(j+1, k, l) = \theta^i(j+1, k, l) + \frac{\theta^c(j, k, l) - \theta^i(j+1, k, l)}{\|\theta^c(j, k, l) - \theta^i(j+1, k, l)\|} * C(i) \quad (7)$$

where: $\theta^c(j, k, l)$ is the central position of good bacteria colony.

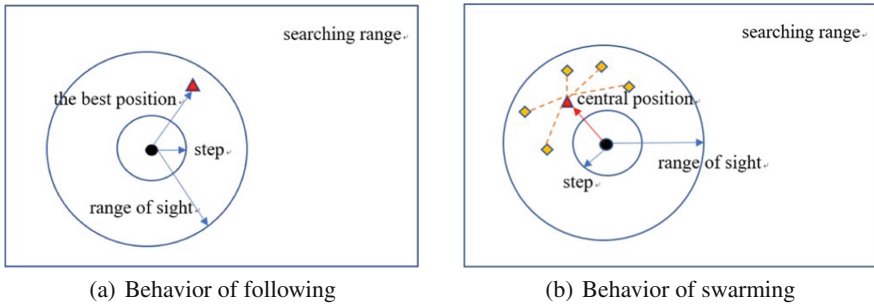


Fig. 1. Following and Swarming Behaviors

3.2 Gaussian Disturbance

Swarming and following improve the convergence speed greatly but the problem of bacteria trapped in local optima remains. In original BFO, if there is a relatively good bacteria in the colony, it will attract the rest of bacteria to approach to it, which lowers the diversity of population. In this paper, we proposed a random number following Gaussian distribution to change the former position using $\theta^{ii} = \theta^i * (1 + N(0.1))$. If the new position is better than the old one, the position will be updated by new one. The equation can be defined as:

$$\theta^{best} = \begin{cases} \theta^{ii} & f(\theta^{ii}) < f(\theta^i) \\ \theta^i & otherwise \end{cases} \quad (8)$$

where: θ refers to the position of bacteria. f represents the fitness value.

In this way, diversity gets improved and bacteria have stronger searching ability.

3.3 Elimination and Dispersal

In original method, elimination-dispersal is determined by a probability p , meaning both the good individuals and bad ones hold the same opportunity to be eliminated. However, it is hard to keep the useful position information if good bacteria are eliminated. In that case, a new elimination dispersal step is adopted. The colony of health bacteria are split into two equal groups. And one of them are eliminated according the fitness value. If the fitness in new position is better than the fitness before eliminating, bacteria locate in the new positions, otherwise back to the former positions.

In the process of modifying algorithm, the algorithm with following and swarming behaviors outperforms the algorithms with only Gaussian disturbance and only elimination-dispersal change, proving that following and swarming mechanism is the most effective one among the three individual strategies. But the combination of strategies is much inferior. It is concluded that with the following and swarming behaviors, each bacterium is able to self-adaptively adjust its positions according to others' information in the most important chemotaxis mechanism and Gaussian disturbance strategy helps to escape local optima while following and swarming improve the convergence speed.

Algorithm 1. Pseudo code for AF-GBFO

```

1: Initial positions and parameters: S,Nc,Nre,Ned...(S is the number of bacteria,, Nre
   is the max number of reproduction, Ned is the chances of elimination and dispersal.)
2: for  $l = 1 \rightarrow Ned$  do
3:   for  $k = 1 \rightarrow Nre$  do
4:     for  $j = 1 \rightarrow Nc$  do
5:       Compute all the fitness function
6:       for  $i = 1 \rightarrow S$  do
7:         1.Compute the distance between the bacteria and others and find the
8:         best one within visual scope.2.update the position using Eq. (6).
9:         3.Swim for N steps and Compute the central position
10:        if  $Y_c/n_f > \delta Y_i$  then
11:          Update the position using Eq. (7)
12:        end if
13:      end for
14:    end for
15:    1.Change the current position with a random number following the Gaussian
16:    distribution. 2.Set the position of the particle equal to the boundary value
17:    if it crosses the search range 3.Update the position if the new one is better.
18:    4.Compute the health values, sort the bacteria with health values and then
19:    copy half of population with better health values
20:  end for
21:  Eliminate bacteria by randomly regenerated
22: end for

```

4 Experiments and Results

4.1 Benchmark Functions

To measure the effectiveness of the novel AF-GBFO algorithm, we adopted 12 benchmark functions to test it in different situations. The search range and the global optimum position X^* are listed in Table 1, g is dimension.

4.2 Experimental Settings

The performance of standard BFO, BFOLIW, BFONIW and GA are compared with AF-GBFO. In BFO, BFOLIW, BFONIW and AF-GBFO, the parameter we set for all the functions include: $S = 50$, Dimension = 15 or 2, $Nc = 1000$, $Nre = 5$, $Ned = 2$, and $Ns = 4$, which ensures a fair comparison. The chemotaxis step length in BFO and AF-GBFO is set 0.01 for all functions. In BFOLIW and BFONIW, a linearly decreasing and nonlinear chemotaxis step length are adopted respectively. The coefficient is 0.2 and initial step length is 0.01 in BFOLIW while the coefficient of BFONIW is 0.6 and step length is 0.001. In GA, the parametric setup is recommended in former researches. The number of group population, generations that max value remains constant, mutation children (Gaussian), mutation children (random) and elitism children are set 50, 20, 20, 20, 2 respectively.

Table 1. Benchmark functions

Function		Search	X*
f1	Rosenbrock	$[-5, 10]$	$[1, 1, \dots, 1]$
f2	Dixon-Price	$[-10, 10]$	$2\hat{g}((2 - 2^g)/2^g), g = 1, \dots, 15$
f3	Styblinski-Tang	$[-5, 5]$	$[-2.90, -2.90, \dots, -2.90]$
f4	Ackley	$[-32.768, 32.768]$	$[0, 0, \dots, 0]$
f5	Levy	$[-10, 10]$	$[1, 1, \dots, 1]$
f6	Rastrigin	$[-5.12, 5.12]$	$[0, 0, \dots, 0]$
f7	Eggholder	$[-512, 512]$	$[512, 404.23]$
f8	Michalewicz10	$[-10, 10]$	$[2.20, 1.57]$
f9	Michalewicz20	$[-10, 10]$	$[2.20, 1.57]$
f10	Beale	$[-4.5, 4.5]$	$[3, 0.5]$
f11	Levy N.13	$[-10, 10]$	$[1, 1]$
f12	Drop-Wave	$[-5.12, 5.12]$	$[0, 0]$

Table 2. Functions results

Function		BFOLIW	BFO	GA	BFONIW	AFGBFO	p-value
f1	Mean	8.02E+00	1.87E+01	5.73E+01	6.49E+00	4.40E+00	1.27E-01
	Std.	8.99E+00	2.05E+01	2.67E+02	8.28E+00	1.32E-02	
f2	Mean	6.76E-01	1.38E+00	2.09E+00	6.56E-01	6.67E-01	1.80E-02
	Std.	2.64E-06	2.13E-02	1.10E+00	4.97E-03	1.16E-12	
f3	Mean	-5.59E+02	-5.44E+02	-5.44E+02	-4.79E+02	-5.78E+02	0.00E+00
	Std.	2.00E+02	5.16E-02	3.96E+02	7.27E+02	2.66E+02	
f4	Mean	1.57E+01	1.56E+01	1.97E+00	4.47E-02	8.88E-16	1.40E-02
	Std.	9.98E-01	1.78E+00	4.43E-01	5.52E-05	0.00E+00	
f5	Mean	8.84E+00	1.13E+01	3.50E-01	4.01E-01	4.83E-01	1.49E-01
	Std.	2.49E+00	5.92E+00	1.68E-02	5.24E-01	9.26E-03	
f6	Mean	2.56E+01	4.14E+01	1.21E+01	5.34E+01	0.00E+00	4.10E-02
	Std.	2.28E+01	1.62E+01	1.62E+01	1.36E+02	0.00E+00	
f7	Mean	-9.35E+02	-9.35E+02	-9.59E+02	-9.67E+02	-5.15E+04	3.30E-02
	Std.	5.57E-16	2.12E-13	1.48E+00	4.92E+03	1.13E+09	
f8	Mean	-5.58E+00	-4.69E+00	-7.74E+00	-5.19E+00	-9.62E+00	2.00E-03
	Std.	2.15E-01	8.91E-02	1.91E+00	8.64E-02	7.48E-01	
f9	Mean	-5.08E+00	-3.95E+00	-6.96E+00	-4.88E+00	-8.87E+00	2.00E-03
	Std.	1.12E-01	3.33E-02	7.83E-01	3.40E-01	7.07E-01	
f10	Mean	6.11E-07	9.09E-06	4.99E-09	1.32E-06	7.56E-12	2.70E-02
	Std.	6.21E-13	1.03E-10	2.49E-16	1.60E-12	4.31E-23	
f11	Mean	3.93E-06	2.28E-04	1.35E-31	3.13E-06	1.85E-10	3.58E-01
	Std.	1.28E-11	3.89E-08	0.00E+00	5.23E-12	3.26E-20	
f12	Mean	1.78E+03	-1.00E+00	1.92E+03	-9.94E-01	-1.00E+00	1.79E-01
	Std.	1.53E-12	0.00E+00	1.62E+04	1.13E-05	0.00E+00	

After adopting the AFSA into BFO, its essential to value the visual scope affected by the search range and dimensions. We take different rate of searching range as the visual scope. According to experiments, $C = 0.3$ should be adopted.

4.3 Experimental Results

In this part, 12 functions are adopted to measure the global searching ability and the convergence rate of algorithms. Table 2 shows the best averaged results of ten independent runs. From the Table 2, we can see that AF-GBFO excels other algorithm at most cases according to mean value and standard deviation.

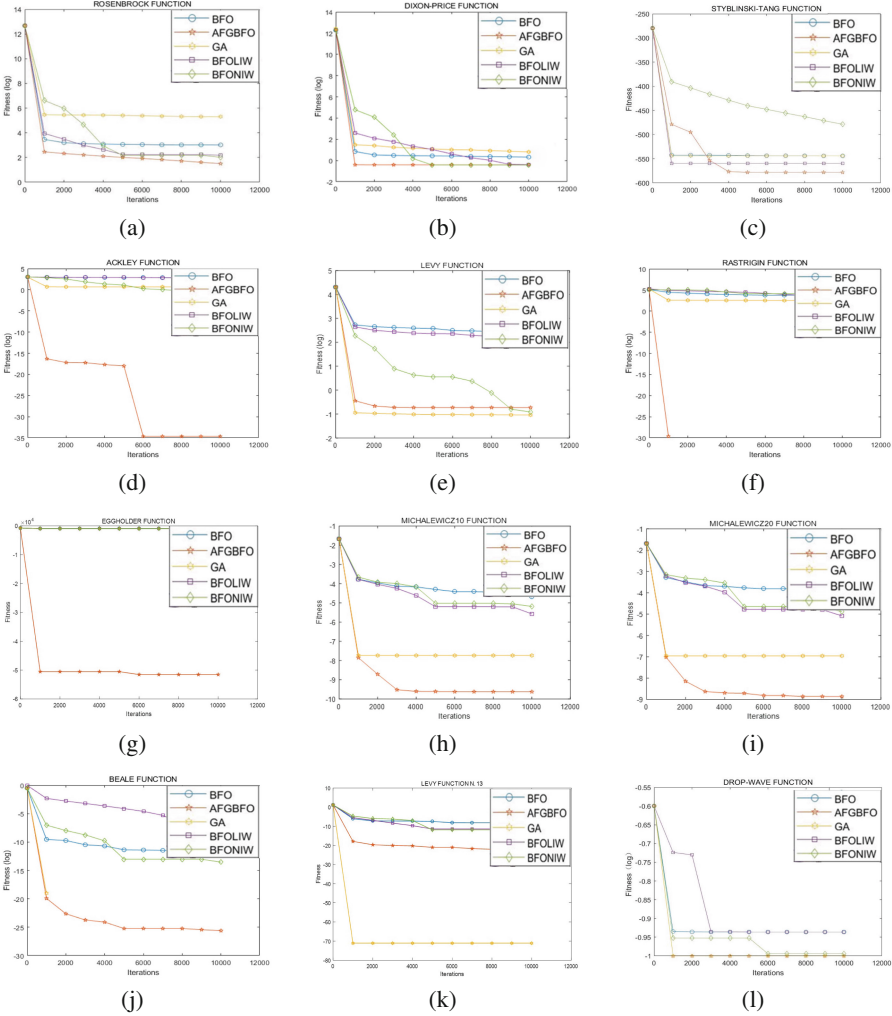


Fig. 2. Convergence curve of function

Figure 2 shows the comparison results of 5 algorithms. The functions shown in Fig. 2(a)–(b) are unimodal functions, Rosenbrock function and Dixon-price function are referred to as the valley. In such shape, convergence to the minimum

is difficult. According to experiments, AF-GBFO finds relatively good solutions and shows fast convergence speed.

The functions shown in Fig. 2(c)–(l) are multimodal functions which exist many local optimum. For functions (f) and (l), AF-GBFO finds the global optimum directly. In function (g), we can see that bacteria in AG-BFO finds the fitness value compared with others. According to the figure of function (j), AF-GBFO doesn't return the best solution but its convergence is still outstanding. When it comes to the function (h) and (i), the Michalewicz function has steep valleys and ridges leading to a more difficult search. According to the figure of (h) and (i), AF-GBFO locates the best fitness value, showing the strong ability of jumping local optimum. But when it comes to functions (e) and (k), GA algorithm excels AF-GBFO in results and in function (k) the disparity becomes obvious. Function (k) is a two-dimensional function. The result may be that in multi-dimensional functions AF-GBFO can exert its searching ability. In order to distinguish whether the results achieved by AF-GBFO were statistically significantly better than other algorithms', T-tests were introduced and the confidence level was set to 95%. From the Table 2 p-value is usually less than 5%, which means that the experimental results from AF-GBFO were significantly different from other results. In conclusion, the AF-GBFO is an effective and competitive algorithm especially in solving multimodal and multi-dimensional functions.

5 Conclusion

Based on BFO and AFSA, a new algorithm AF-GBFO is proposed. This algorithm proposes a new chemotaxis mode that gives the bacteria virtual vision, approaching to the bacteria with the highest concentration of nutrients in the sight, and the convergence speed is greatly accelerated. A number based on Gaussian perturbation is also added to the learning process to avoid bacteria trapping into local optimum. To test the new modified algorithms' effectiveness, AF-GBFO and other four algorithms BFO, BFO-LIW, BFO-NIW, GA, are tested over 12 benchmark functions. Experiment results show that AF-GBFO is usually superior to other algorithms especially in multimodal problems, which greatly proves its fast convergence and strong global searching ability. Now, AF-GBFO is just tested to the benchmark functions. In the future, we will do more to improve this algorithm and apply it to real problems.

Acknowledgment. This work is partially supported by the Natural Science Foundation of Guangdong Province (2018A030310575), Natural Science Foundation of Shenzhen University (8530 3/00000155), Project supported by Innovation and Entrepreneurship Research Center of Guangdong University Student (2018A073825), Research Cultivation Project from Shenzhen Institute of Information Technology (ZY201717) and Innovating and Upgrading Institute Project from Department of Education of Guangdong Province (2017GWT SCX038). Ruozhen Zheng and Zhiqin Feng are first authors. They contributed equally to this paper.

References

1. Azizi, R.: Empirical study of artificial fish swarm algorithm. *Comput. Sci.* **17**(6), 626–641 (2014)
2. Chen, H., Niu, B., Ma, L., Su, W., Zhu, Y.: Bacterial colony foraging optimization. *Neurocomputing* **137**, 268–284 (2014)
3. Daas, M.S., Chikhi, S., Batouche, M.: Bacterial foraging optimization with double role of reproduction and step adaptation. In: *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, p. 71. ACM (2015)
4. Dasgupta, S., Das, S., Abraham, A., Biswas, A.: Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. *IEEE Trans. Evol. Comput.* **13**(4), 919–941 (2009)
5. Feng, X.H., He, Y.Y., Yu, J.: Economic load dispatch using bacterial foraging optimization algorithm based on evolution strategies. In: *Advanced Materials Research*, vol. 860, pp. 2040–2045. Trans Tech Publ. (2014)
6. Gupta, N., Saxena, J., Bhatia, K.S.: Optimized metamaterial-loaded fractal antenna using modified hybrid BF-PSO algorithm. *Neural Comput. Appl.*, 1–17 (2019). <https://doi.org/10.1007/s00521-019-04202-z>
7. Kou, P.G., Zhou, J.Z., Yao-Yao, H.E., Xiang, X.Q., Chao-Shun, L.I.: Optimal PID governor tuning of hydraulic turbine generators with bacterial foraging particle swarm optimization algorithm. *Proc. CSEE* **29**(26), 101–106 (2009)
8. Mishra, S.: Bacteria foraging based solution to optimize both real power loss and voltage stability limit. In: *Power Engineering Society General Meeting* (2007)
9. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**(3), 52–67 (2002)
10. Tan, L., Lin, F., Hong, W.: Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows. *Neurocomputing* **151**(3), 1208–1215 (2015)
11. Teng, F., Zhang, L.: Application of BFO-AFSA to location of distribution centre. *Cluster Comput.* **20**(3), 3459–3474 (2017). <https://doi.org/10.1007/s10586-017-1144-5>
12. Wang, L., Zhao, W., Tian, Y., Pan, G.: A bare bones bacterial foraging optimization algorithm. *Cogn. Syst. Res.* **52**, 301–311 (2018)
13. Xiaolei, L.I., Shao, Z., Qian, J.: An optimizing method based on autonomous animals: fish-swarm algorithm. *Syst. Eng.-Theory Pract.* **22**, 32–38 (2002)
14. Yazdani, D., Golyari, S., Meybodi, M.R.: A new hybrid algorithm for optimization based on artificial fish swarm algorithm and cellular learning automata. In: *International Symposium on Telecommunications* (2010)