Adaptive Bacterial Foraging Optimization Based on Roulette Strategy

Weifu Cao, Yingsi Tan^(✉), Miaojia Huang, and Yuxi Luo

College of Management, Shenzhen University, Shenzhen 518061, China

Caowave@qq.com, 394461726@qq.com

Abstract. Bacterial foraging optimization has drawn great attention and has been applied widely in various fields. However, BFO performs poorly in convergence when coping with more complex optimization problems, especially multimodal and high dimensional tasks. Aiming to address these issues, we therefore seek to propose a hybrid strategy to improve the BFO algorithm in each stage of the bacteria's foraging behavior. Firstly, a non-linear descending strategy of step size is adopted in the process of flipping, where a larger step size is given to the particle at the very beginning of the iteration, promoting the rapid convergence of the algorithm while later on a smaller step size is given, helping enhance the particles' global search ability. Secondly, an adaptive adjustment strategy of particle aggregation is introduced when calculating step size of the bacteria's swimming behavior. In this way, the particles will adjust the step size according to the degree of crowding to achieve efficient swimming. Thirdly, a roulette strategy is applied to enable the excellent particles to enjoy higher replication probability in the replication step. A linear descent elimination strategy is adopted finally in the elimination process. The experimental results demonstrate that the improved algorithm performs well in both single-peak function and multi-peak function, having strong convergence ability and search ability.

Keywords: Non-linear decreasing methods · Roulette gambling mechanism · Hybrid algorithm

1 Introduction

The past decades have witnessed numerous researchers sought different methods to tackle optimization problems. Mathematics methodology used to be a relatively good technique to solve these problems but its limitations turn out to be obvious as more and more practical optimization problems involve complexity, multi-minimum, strong constraint, non-linearity and a lot of variables and modeling difficulty. The traditional mathematics methodology is incapable of reaching the optimum in a reasonable computation time. Hence, developing new approaches has become one of the most important research directions. In recent years, a number of metaheuristics inspired from nature and

The original version of this chapter was revised: the author name was updated. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-53956-6_64

© Springer Nature Switzerland AG 2020, corrected publication 2020

Y. Tan et al. (Eds.): ICSI 2020, LNCS 12145, pp. 299–311, 2020.

https://doi.org/10.1007/978-3-030-53956-6_27

biology has appealed numerous researches and have developed and come to widely used in different practice problems [1]. For example, genetic algorithm (GA) [2], is a probabilistic search algorithm, mimicking nature selection and biology evolution mechanism; particle swarm optimization (PSO) [3], is developed based on the swarming strategies of bird flocking and fish schooling.

Bacterial foraging optimization (BFO) [4], proposed by Passino is also belongs to this bio-mimetic algorithms group. It mimics the foraging strategy of *Escherichia coli* bacteria, involving four key processes, namely, chemotaxis, swarming, reproduction and elimination-dispersal steps, encouraging BFO to search the global optimum more efficiently than the traditional mathematical methods. BFO algorithm has been effectively applied to solve practical optimization problems in different fields, such as PID controller design [5], dynamic environment optimization [6], image processing [7, 8], vehicle routing [9], and machine learning [10]. Once emerged, BFO has been widely applied to various single-objective optimization problems [11–14] and multi-objective optimization problems [15].

The increasing interest for BFO has revealed that it is a promising, swarm-based optimization algorithm. However, there still exist several limitations in original BFO due to the fixed chemotaxis step-size, the less-efficient search direction for tumbling and the swarming strategy with a delay in approximating the global solution and a poor convergence rate, especially in tackling nonlinear, high-dimensional and multi-modal functions [1, 16–19]. Many scholars proposed corresponding improvement strategies based on these problems, mainly aiming at the two key steps of chemotaxis and replication. Some of the current researches of improving BFO could be described as follows:

With regards to the chemotaxis process, Panda and Naik (2015) [16] proposed a modified BFO algorithm called adaptive crossover BFO algorithm, which incorporated adaptive chemotaxis and inherited the crossover mechanism of genetic algorithm. W. G. Zhao, L. Y. Wang (2016) [1] reported an effective bacterial foraging optimization (EBFO) where a gravitational search strategy is incorporated into the chemotaxis step to adjust its unit length according to the swarm information. L. L. Wanga et al. [17] (2018) developed a bare bones bacterial foraging optimization (BBBFO) algorithm in which a chemotactic strategy based on Gaussian distribution is incorporated into this method through making use of both the historical information of individual and the share information of group. C. C. Yang et al. (2016) designed a new bacterial foraging optimizer using new chemotaxis and conjugation strategies (BFO-CC). Via the new chemotaxis mechanism, each bacterium randomly selects a standard-basis-vector direction for swimming or tumbling. At the same time, the step size of each bacterium is adaptively adjusted based on the evolutionary generations and the information of the globally best individual, which readily makes the algorithm keep a better balance between a local search and global search and significantly improve convergence. B. Pang et al. (2019) [18] improved BFO algorithm (LPBFO) based on the Lévy flight step-size and particle swarm optimization (PSO) operator. During the chemotactic process in LPBFO, each bacterium selects one dimension for tumbling randomly. The step-size of each bacterium is determined by the stochastic flight lengths of the improved Lévy flight which can generate small step-size with high frequency and big step-size occasionally and the stochastic step-size is reduced

adaptively based on the evolutionary generations, which makes the bacteria transform from global search to local search.

In the reproduction step, W. G. Zhao, L.Y. Wang (2016) [1] designed a swarm diversity strategy to enhance the reproduction mode depending on the swarm diversity. The simulation results show that the proposed algorithm is more effective than its competitors and can be extended to other global optimization problems. Liang and H.S. Tian (2016) [19] presents a hybrid algorithm based on the combination of BFO algorithm and PSO algorithm. Due to the random change of the direction of search, and the migration of bacteria by selective replication at a certain probability, results show that the improved hybrid algorithm is better than the basic PSO algorithm and the basic BFO algorithm. Besides, the convergence rate is fast and has good robustness. L.L. Wanga et al. [17] (2018) introduced the swarm diversity in the reproduction strategy to promote the exploration ability of the algorithm. The comparative results reveal that the proposed approach is more superior to its counterparts.

Overall, the current optimization algorithm of bacterial foraging focuses on proposing novel chemotaxis step strategies, but pays less attention to the reproduction process, especially to enhancing the influence of particles that perform well, so as to accelerate the convergence rate and improve the searching capacity of the particles.

To improve the original BFO for tackling complex optimization problems. We incorporate three strategies into the traditional BFO in each stage of the bacteria's foraging behavior, seeking to achieve a valid balance between the exploration and the exploitation during the life cycle of search for global optima. Firstly, to ensure the convergence performance and local mining abilities of the algorithm, a non-linear descending strategy of step size is adopted in the process of chemotaxis, where a larger step size is given to the particle at the very beginning of the iteration, promoting the rapid convergence of the algorithm and a smaller step size is given later on to help enhance the particles' local search ability. Secondly, an adaptive adjustment strategy of particle aggregation is introduced when calculating step size of the bacteria's swimming behavior. In this way, the particles will adjust the step size according to the degree of crowding to achieve efficient swimming. Thirdly, a roulette strategy is applied to enable the excellent particles to enjoy higher replication probability in the replication step. A linear descent elimination strategy is adopted finally in the elimination process, thus avoiding premature and a trap into local extremum. Various benchmark functions are considered to verify the performance of the proposed improved strategies, the comparative results demonstrate that the improved algorithm has a significant enhancement over the original BFO, especially in single-peak function and multi-peak function, having strong convergence ability and search ability.

2 Standard BFO

Bacterial Foraging Optimization (BFO), a new swarm intelligence optimization algorithm proposed by Passino Kevin in 2002. The BFO algorithm simulates the foraging behavior of *Escherichia coli* in the human intestinal tract for mathematical modeling, so as to solve the optimization problem. Because of its simple structure and parallel processing, it has been widely used in many fields. According to the Theory of Bacterial Foraging, bacterial population has a strong tendency to nutrients. And every step

of their movement is towards the position where they can get the maximum energy per unit time under the constraints of their own physiology and surrounding environment. *Escherichia coli* foraging behavior mainly includes four basic steps: Chemotaxis, Swarming, Reproduction, Elimination and dispersal.

2.1 Chemotaxis

Chemotaxis is a process that simulates *Escherichia coli* swimming and flipping. *Escherichia coli* migration can be divided into two processes: turnover and swimming, both of which are realized by flagella. If the flagella are counterclockwise rotation, *Escherichia coli* will be swimming; If the flagella were rotating clockwise, the bacteria would flip around in place looking for direction. After the bacteria find a new swimming direction by flipping, they will swim in that direction for the same length many times.

To define the direction of Chemotaxis, a unit length random direction is generated named $\varphi(j)$. Then in computational Chemotaxis, the movement of the bacterium could be represented by formula (1)

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\varphi(j) \quad (1)$$

where $C(i)$ represents the size of the step taken in the random direction, $\theta^i(j, k, l)$ is the i th bacterium at j th chemotactic, k th reproductive, and l th elimination and dispersal step.

2.2 Swarming

In the process of bacteria approaching the optimal living environment, individuals of the colony release information, attractant and repellent, namely pheromone, to all other bacteria according to their current fitness value. This bacterium accepts the attractor and repulsor released by all other bacteria and modifies its adaptive value under the action of all its pheromones. Make sure to move to a better location for the local environment.

In a colony, the calculation method of pheromone concentration between the above individuals can be expressed as formula (2)

$$\begin{aligned} J_{cc}(\theta, P(j, k, l)) &= \sum_{i=1}^n J_{cc}^i(\theta, \theta^i(j, k, l)) \\ &+ \sum_{i=1}^n [h_{\text{repellent}} \exp(-w_{\text{repellent}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \\ &+ \sum_{i=1}^n [-d_{\text{attract}} \exp(-w_{\text{attract}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \end{aligned} \quad (2)$$

where $J_{cc}(\theta, P(j, k, l))$ is a target function that changes with the population distribution state. When it is superimposed to the actual target function of the problem, it shows the trend of the relative distance between bacteria and the best individual. S is the size of the population; P indicates the dimensions of the optimization problem.

2.3 Reproduction

According to the principle of survival of the fittest, individuals with better performance (fitness) will reproduce, while those with poor performance will die.

First step of Reproduction's algorithm implementation is to calculate the health value, which is used to measure an individual's ability to explore. The health value of bacterium i determined as follows:

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l) \quad (3)$$

Then sorting the health values of the bacteria in descending order. The least half healthy bacteria die and the other half healthiest bacteria each split into two bacteria, which are placed in the same location. During the process of reproduction, the population of bacteria is constant.

2.4 Elimination and Dispersal

The number of bacterial populations changes gradually over time or with changes in the environment, such as nutrient depletion. When an emergency occurs, bacteria in an area may be eliminated or dispersed to new areas. The last step of Standard BFO algorithm is the simulation of this process. Bacterial migration occurs according to a certain probability P_{ed} , when a certain bacteria meets the migration conditions, the individual will be reallocated to another space.

3 Adaptive BFO Based on Roulette Strategy

In order to solve the problem of lack of learning direction and slow convergence speed in traditional bacterial foraging algorithm, this paper improve an adaptive bacterial foraging optimization algorithm based on roulette strategy (ARBFO) which improves BFO from chemotaxis, replication and elimination-diffusion. The ARBFO algorithm adopts roulette strategy and adaptive adjustment formula, which makes the algorithm have the ability of learning and adaptive adjustment to the environment. As a result, it accelerates the convergence speed and improves the search accuracy.

3.1 Chemotaxis: Two Non-linear Decreasing Methods

Chemotaxis step size controls the search range of the population. If the step size is not set properly, it will be trapped in local optimum or unable to converge to the optimal solution.

In the process of microbial foraging, the distance between each colony and the optimal solution in the initial population is larger, and the step size should be larger to help the rapid convergence of the microbial community. With the search, the distance between each colony and the optimal solution decreases, and the step size should be reduced at this time, so as to improve the accuracy of the microbial community search.

In this paper, two non-linear decreasing methods are used to adjust the step sizes of inversion and swimming in chemotaxis. In the process of inversion, with the increase of replication behavior, the newborn bacteria will search for a better range. At this time, the step size should be smaller to improve the search accuracy. The non-linear decline formula of step size should be adjusted:

$$C_i^k = C_{\min} + (C_{\max} - C_{\min})e^{(-4 * (\frac{k}{N_{re}}))^3} \quad (4)$$

Among them, K is the current number of replication, and N_{re} is the largest number of bacterial replication. With the increase of k , the step size will decrease.

In the course of swimming, a strategy of non-linear adaptive adjustment of step size is adopted. Considering the current iteration and the maximum iteration number, the number of chemotactic steps in a given range is changed. With the increase of iteration number, the step size shows a downward trend. This strategy advances according to the number of iterations and the current nutritional status of particles. Adaptive adjustment [21], as follows:

$$C(j) = [(c_{\max} - c_{\min} - k_1 - k_2) * \cos(\frac{iter}{iter_{\max}} * \frac{\pi}{2}) + c_{\min} + k_2] + k_1 * \frac{\pi}{2} * \arctan \frac{J}{J_{avg}} - k_2 * \frac{\min(J, J_t)}{\max(J, J_t)} \quad (5)$$

In this formula, besides the classical adaptive adjustment strategy, the parameters J , J_{avg} , J_t . adjusted according to the microbial nutrition environment are introduced. The swimming step is adjusted by the overall weight, the average microbial nutrition value, and the individual optimal fitness value, which enhances the learning ability of the operator. With the increase of iteration times, the step size is as a whole. The step size decreases with the increase of nutrient value of the bacterial community search, so as to improve the accuracy of the later stage of the search.

Among them, J and J_{avg} represent the current bacterial fitness and the average value of all bacteria respectively. J_t is the optimal fitness value of the previous individual. $K1$ and $K2$ are normal numbers. They are adjusted adaptively according to the average value of bacterial nutrition and the optimal fitness value of the individual. After experiments, they are set to 2,0.4 respectively.

3.2 Duplication: Roulette Gambling Mechanism

The replication strategy in the standard BFO algorithm, which retains half of the bacterial individuals with good fitness and eliminates the other half of the bacterial individuals with poor fitness, has not improved the optimal position of the current bacterial community; at the same time, this replication method reduces the diversity of the population in a half way by sacrificing the diversity of the population. The fast convergence of the algorithm can easily lead to the local optimum of the algorithm in the solution of high-dimensional and multi-peak functions.

In this paper, the roulette mechanism is used to improve the replication link in the flora foraging algorithm, improve the learning ability of the operator and improve the convergence speed. Roulette selection strategy, also known as proportional selection

operator, has the basic idea that the probability of each individual being selected is proportional to the value of its fitness function.

If the population size is N and the fitness of individual X_i is $f(x_i)$, then the selection probability $P(x_i)$ of individual X_i is:

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^N f(x_j)} \quad (6)$$

The roulette selection method can be realized by following process simulation:

- (1) A uniformly distributed random number R is generated in $[0,1]$.
- (2) If $r < q_1$, chromosome X_1 is selected.
- (3) If $q_{k-1} < R < Q_K$ ($2 < K < N$), the chromosome X_K is selected.
- (4) Q_i is called individual X_i ($i = 1, 2, \dots$. The cumulative probability of N) is calculated as follows:

$$q_i = \sum_{j=1}^i P(x_j) \quad (7)$$

In this process, the higher the fitness, the greater the chance that the particle will be selected, and vice versa, the smaller the chance that the particle will be selected. However, regardless of the fitness, the probability of being selected exists, allowing some particles with low nutritional value to survive. It is possible to improve the diversity of particles (Fig. 1).

3.3 Reproduction and Elimination: Linear Decreasing Adaptive Regulation Mechanism

Migration is the behavior that interferes with bacterial colonies and initializes the population randomly according to a certain probability. Migration probability describes the living environment of bacteria. Generally speaking, the change of nutrient value of bacterial colonies is not susceptible to the influence of bacteria, but decreases with the reproduction of bacteria. At the same time, the survival competition among bacteria has become more intense, and the more likely it is to disperse.

In this paper, a linear decreasing adaptive regulation mechanism is adopted. According to the linear decreasing of iteration number, the improved migration step simulates the change of environment during a bacterial foraging process according to probability formula (8). With the increase of iteration number, the migration probability decreases gradually, the chance of executing probability greater than the migration probability increases, and the bacteria are dispersed. The probability increases gradually to improve the diversity of particles and search accuracy.

$$Ped_j = Ped_{\min} + \frac{iter_{\max} - iter}{iter_{\max}} * (Ped_{\max} - Ped_{\min}) \quad (8)$$

Fig. 1. Duplication flow chart after quoting roulette mechanism

Among them, P_{ed} represents the probability of eliminating migration, $ITER$ is the current number of iterations. Through this formula, the probability of eliminating migration can be adjusted adaptively according to the number of iterations to maximize the diversity of particles and search accuracy.

3.4 Benchmark Function

To verify the performance of the improved algorithm (ARBFO), this article introduces Sphere, Schwefel, Rosenbrock, Rotated-Hyper-Ellipse, Beale, General rastrigrin, Ackley, Griewank, Alpine, Three-hump Camel [17, 20] ten typical benchmark functions to conduct simulated contrast experiment. The minimum of ten benchmark functions is 0. Among them, Sphere, Schwefel, Rosenbrock, Rotated-Hyper-Ellipse, Beale are unimodal functions, General rastrigrin, Ackley, Griewank, Alpine, Three-hump Camel are multimodal functions. And the Rotated-Hyper-Ellipse is the only rotated function.

Table 1. Experiment Result

	Function		ASRBFO	BFOLD C	BFONDC	ARBFO
f_1	<i>Sphere</i>	<i>Min</i>	1.0448E+05	2.2529E+04	1.7247E-02	3.1663E-02
		<i>Max</i>	1.0448E+05	1.2522E+05	2.4423E-02	5.0875E-02
		<i>Mean</i>	1.0924E+05	5.8324E+04	2.0858E-02	3.9629E-02
		<i>Std</i>	2.4426E+03	2.8390E+04	2.6759E-03	5.6516E-03
f_2	<i>Schwefel</i>	<i>Min</i>	5.7970E+64	4.6761E+58	4.9436E+46	4.4882
		<i>Max</i>	5.7970E+64	3.5092E+72	4.7399E+52	7.7279
		<i>Mean</i>	3.9266E+67	4.0783E+69	8.5016E+51	7.1735
		<i>Std</i>	3.9535E+67	9.1487E+70	1.5520E+52	9.6905E-01
f_3	<i>Rosenbrock</i>	<i>Min</i>	3.8772E+10	1.5537E+09	5.1282E+01	0
		<i>Max</i>	3.8772E+10	5.5556E+10	1.0079E+03	0
		<i>Mean</i>	4.2359E+10	1.3921E+10	1.8286E+02	0
		<i>Std</i>	2.4723E+09	1.3059E+10	2.9506E+02	0
f_4	<i>Rotated-Hyper-Ellipse</i>	<i>Min</i>	2.5707E+06	4.4197E+05	1.8046	2.2840
		<i>Max</i>	2.5707E+06	3.0662E+06	1.9574E+02	3.8138
		<i>Mean</i>	2.6580E+06	1.3151E+06	9.2511E+01	2.6105
		<i>Std</i>	8.1457E+04	7.1888E+05	7.6209E+01	4.5021E-01
f_5	<i>Beall</i>	<i>Min</i>	3.1277E-02	3.5338E-01	3.8131E-10	4.3665E-11
		<i>Max</i>	3.1277E-02	1.9030E+05	1.9680E-03	2.1881E-08
		<i>Mean</i>	4.0331E-01	2.2927E+02	1.9682E-04	4.8885E-09
		<i>Std</i>	1.5528E-01	4.4009E+03	6.2233E-04	6.1904E-09
f_6	<i>Generalized Rastrigrin</i>	<i>Min</i>	1.0278E+05	3.3723E+04	8.9868E+02	5.00E+01
		<i>Max</i>	1.0278E+05	1.2570E+05	1.2176E+03	5.00E+01
		<i>Mean</i>	1.0826E+05	6.5668E+04	1.0957E+03	5.00E+01
		<i>Std</i>	3.4972E+03	2.5350E+04	1.0088E+02	0
f_7	<i>Ackley</i>	<i>Min</i>	2.1225E+01	2.0015E+01	2.0017E+01	1.9973
		<i>Max</i>	2.1225E+01	2.1457E+01	2.0023E+01	2.6504
		<i>Mean</i>	2.1255E+01	2.0305E+01	2.0019E+01	2.3840
		<i>Std</i>	2.1945E-02	3.7634E-01	2.1066E-03	2.1936E-01
f_8	<i>Griewank</i>	<i>Min</i>	2.7011E+01	6.6180	1.7503E-03	3.5593E-03
		<i>Max</i>	2.7011E+01	3.2305E+01	3.6206E-03	7.8410E-03
		<i>Mean</i>	2.7887E+01	1.5600E+01	2.7847E-03	5.6189E-03
		<i>Std</i>	7.0130E-01	7.1150	4.7405E-04	1.3919E-03
f_9	<i>Alpine</i>	<i>Min</i>	8.5683E+02	9.4282E+01	1.1371E+02	9.6447E-02
		<i>Max</i>	8.5683E+02	1.1429E+03	1.5907E+02	1.6201E-01
		<i>Mean</i>	9.0606E+02	2.2713E+02	1.3368E+02	1.2218E-01
		<i>Std</i>	4.2282E+01	1.7135E+02	1.4184E+01	2.1431E-02
f_{10}	<i>Three-Hump Camel</i>	<i>Min</i>	9.1108E-03	5.9735E-02	2.0558E-10	8.4751E-11
		<i>Max</i>	9.1108E-03	4.4493E+01	6.3831E-09	2.7885E-09
		<i>Mean</i>	3.7530E-01	3.8030E-01	2.6565E-09	1.1598E-09
		<i>Std</i>	4.9494E-01	2.6250	1.7704E-09	9.0145E-10

3.5 Parameter Settings

The experimental environment is Inter(R) Core (TM) i5-7Y54 CPU @1.20 GHz, Windows 10, the running memory is 8BG, and it is completed under Matlab2018b.

To exam the performance of ARBFO, this article choose ASRBFO [21], BFOLDC [22] and BFONDC [23] to conduct the simulated contrast experiment, and record the minimum value, maximum value, average value and standard deviation. Each algorithm independently runs 10 times. Different BFO algorithms are set the same parameters. For BFOLDC, $C_{max} = 0.2$, $C_{min} = 0.01$. For BFONDC, $C_{max} = 0.6$, $C_{min} = 0.01$. For ASRBFO, maximum evaluation number $MaxFEs = 10000$; $Fre = 2000$, i.e. reproduce every 2000 iterations; $Fed = 3300$, i.e. dispel every 3300 iterations; initial step unit $Csz = 0.15$; swimming times $Ns = 10$, $C_{max} = 0.3$, $C_{min} = 0.15$, $k1 = 0.2$, $k2 = 0.15$. For ARBFO, $C_{max} = 0.6$, $C_{min} = 0.01$; $k1 = 0.2$, $k2 = 0.15$.

3.6 Analysis of Algorithm Convergence

Table 1 shows the minimum value (Min), maximum value (Max), average value (Mean) and standard deviation (Std) of four algorithms in 10 benchmark functions. The dimension of the particles is 50, and the optimal results will be marked in bold.

It can be seen from the data in Table 1 that, in terms of benchmark functions, ARBFO algorithm yields a compelling result in the convergence accuracy and stability of both unimodal functions and multimodal functions. Function 1–5 are unimodal functions and function 6–10 are multimodal functions.

Figure 2 shows the fitness curve of four algorithms. Combined Fig. 2 with Table 1, ARBFO outperforms other algorithms in functions 2, 4, 5, 6, 7, 9, and the optimization accuracy and stability of ARBFO algorithm are improved obviously. It is worth noticing that, the fitness curve of Rosenbrock has a breakpoint, because the ARBFO has converged to the minimum value of 0.

Compared with BFONDC algorithm, ARBFO fails to achieve the desired effect in convergence accuracy and stability. Therefore, future research will continue to improve ARBFO to better solve the existing problems on the part of the benchmark functions. The above analysis can prove that the proposed ARBFO algorithm in this paper demonstrates a far better convergence precision and optimal value than the BFO and it is an effective algorithm.

Fig. 2. Convergence curve on 50-Dim

4 Conclusion

In this paper, an adaptive mechanism and roulette strategy designed to improve the capacity of BFO is presented. The adaptive strategy is used to improve the search efficiency and accuracy in the process of tumbling, swimming, elimination and dispersal. Roulette strategy is adopted in the reproduction process to maximize the diversity of the group and avoid the algorithm falling into the local optimal value due to premature development. In tumbling, nonlinear descent strategy is adopted. In the early iterations particles are given larger step length, to prompt the algorithm to fast convergence, later particles will be given smaller step length, to strengthen the particles search precision. In swimming, step length calculation introduces nutritional value adaptive adjustment strategy, the particles will adaptively adjust step length according to the nutrition value, to realize efficient swimming. Roulette strategy is introduced in the reproduction process, therefore, the particles with better fitness value will have higher probability to reproduce, while the ones with worse fitness value have less chances to being selected, which enhances the diversity of particles and increases the search ability. In elimination and dispersal, the linear descending migration strategy is adopted to heighten the diversity of particles.

Ten benchmark functions has been used to test the performance of ARBFO in comparison with ASRBFO, BFOLD and BFONDC. From the analysis and experiments, the result illustrates that this new algorithm outperforms the other four algorithms in convergence speed and accuracy in many cases, and it reach minimum value with Rosenbrock function. It can be concluded that ARBFO expressively improves the performance of BFO and lets out the better results on most optimization problems comparing with other BFO algorithms. However, more benchmark functions and practical application problems must be investigated in the future with ARBFO. The future work will underline more bio-heuristic mechanisms and application problems to entirely improve the performance of ARBFO.

References

1. Zhao, W., Wang, L.: An effective bacterial foraging optimizer for global optimization. *Inf. Sci.* **329**, 719–735 (2016)
2. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman, Publishing Co. Inc., New York (1989)
3. Kennedy, J.F., Eberhart, R.C.: Particle swarm optimization. In: *Proceedings of IEEE International Conference of Neural Network*, Perth, Australia (1995)
4. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**, 52–67 (2002)
5. Kim, D.H., Cho, J.H.: Biomimicry of bacterial foraging for distributed optimization and control. *J. Adv. Comput. Intell. Intell. Inform.* **9**(6), 669–676 (2005)
6. Tang, W.J., Wu, Q.H., Saunders, J.R.: Bacterial foraging algorithm for dynamic environments. In: *IEEE Congress on Computational Intelligence 2006 (CEC 2006)*, Vancouver, Canada, 16–21 July 2006
7. Sanyal, N., Chatterjee, A., Munshi, S.: An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation. *Exp. Syst. Appl.* **38**(12), 15489–15498 (2011)

8. Verma, O.P., Hanmandlu, M., Kumar, P., Chhabra, S., Jindal, A.: A novel bacterial foraging technique for edge detection. *Pattern Recogn. Lett.* **32**(8), 1187–1196 (2011)
9. Tan, L., Lin, F., Wang, H.: Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows. *Neurocomputing* **151**, 1208–1215 (2015)
10. Chana, R.I.: Bacterial foraging based hyper-heuristic for resource scheduling in grid computing. *Future Gener. Comput. Syst. - Int. J. Grid Comput. Esci.* **29**(3), 751–762 (2013)
11. Yi, J., Huang, D., Fu, S., et al.: Optimized relative transformation matrix using bacterial foraging algorithm for process fault detection. *IEEE Trans. Ind. Electron.* **63**(4), 2595–2605 (2016)
12. Verma, O.P., Parihar, A.S.: An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm. *IEEE Trans. Fuzzy Syst.* **25**(1), 114–127 (2017)
13. Mam, M., Leena, G., Saxena, N.S.: Distribution network reconfiguration for power loss minimization using bacterial foraging optimization algorithm. *IJEM-Int. J. Eng. Manuf. (IJEM)* **6**(2), 18–32 (2016)
14. Panda, A., Tripathy, M.: Security constrained optimal power flow solution of wind-thermal generation system using modified bacteria foraging algorithm. *Energy* **93**(8), 816–827 (2015)
15. Lu, Y., Ni, Q.: An improved bacteria foraging optimization algorithm for high dimensional multi-objective optimization problems. In: Tan, Y., Shi, Y., Tang, Q. (eds.) *ICSI 2018. LNCS*, vol. 10941, pp. 540–549. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93815-8_51
16. Panda, R., Naik, M.K.: A novel adaptive crossover bacterial foraging optimization algorithm for linear discriminant analysis based face recognition. *Appl. Soft Comput.* **30**(C) 722–736 (2015)
17. Wang, L.L., Zhao, W.G., Tian, Y.L., Pang, G.Z.: A bare bones bacterial foraging optimization algorithm. *Cogn. Syst. Res.* **52**, 301–311 (2018)
18. Pang, B., Song, Y., Zhang, C., Wang, H., Yang, R.: Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy. *Appl. Intell.* **49**(4), 1283–1305 (2018). <https://doi.org/10.1007/s10489-018-1317-9>
19. Liang, Y., Tian, H.: An improved hybrid algorithm based on bacterial foraging and particle swarm optimization. *Electron. Sci. Technol.* (2017)
20. Tang, K., et al.: Benchmark Functions for the CEC 2008 Special Session and Competition on Large Scale Global Optimization. Technical Report (2007)
21. Jamil, M., Yang, X.S.: A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Model. Numer. Optim.* **4**(2), 150–194 (2013)
22. Tan, L.J., Yi, W.J., Yang, C., Feng, Y.Y.: Adaptive structure-redesigned-based bacterial foraging optimization. In: Huang, D.-S., Jo, K.-H. (eds.) *ICIC 2016. LNCS*, vol. 9772, pp. 897–907. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42294-7_80
23. Niu, B., Fan, Y., Wang, H., et al.: Novel bacterial foraging optimization with time-varying chemotaxis step. *Int. J. Artif. Intell.* **7**(11), 257–273 (2011)