Simplified Bacterial Foraging Optimization Based on Reverse Chemotaxis Strategy

Jun YU
Institute of Science and Technology
Niigata University
Niigata, Japan
email: yujun(AT)ie.niigata-u.ac.jp

Ben NIU
College of Management
Shenzhen University
Shenzhen, China
email: drniuben(AT)gmail.com

***Abstract*—We propose a reverse chemotaxis strategy to guide bacterial individuals to quickly converge to potential areas by using local fitness information efficiently. When bacterial individuals tumble or swim to worthless areas (i.e., poor fitness), the proposed strategy reverses and expands the current search directions in hopes of finding other effective directions to escape the current dilemma. We then use random noise generated by a Gaussian distribution to perturb these reversed directions to prevent bacterial individuals from oscillating in original directions. Besides, we also propose a structural simplification strategy to greatly simplify the optimization framework of the standard bacterial foraging optimization (BFO) by integrating the elimination-dispersal operation into the reproduction operation. In other words, there are two ways to replace unhealthy bacterial individuals in our proposed strategy, one is copied by healthy bacterial individuals and the other is generated by the elimination-dispersal operation. We also use Gaussian mutations to perturb these offspring individuals that exactly replicate their parents to increase the diversity of the population. Finally, the three-level nested BFO optimization framework can be reduced to a single-level loop. To evaluate the performance of our proposal, we run (the standard BFO + two proposed strategies) and the standard BFO on 28 benchmark functions from CEC 2013 test suite, and each function is run 30 times independently on three different dimensions. The experimental results confirmed that our proposed strategies can speed up the BFO search and jump out of local areas effectively.**

***Index Terms*—evolutionary computation, bacterial foraging optimization, simplified BFO, reverse chemotaxis strategy, structural simplification strategy**

I. INTRODUCTION

Optimization has always been one of the hottest topics, and practitioners are committed to developing various techniques to solve increasingly complex real-world problems. Since Newton invented calculus and proposed the concept of the extremum in the 17th century, various classical optimization methods have been continuously proposed, such as the steepest descent method [1], linear programming [2], [3], and nonlinear programming [4], [5]. Unfortunately, these methods are difficult to achieve satisfactory results on modern industrial problems that usually have many characteristics, such as non-differentiability, discontinuity, constraints, and large-scale.

Some researchers borrowed the idea of natural selection and survival of the fittest to develop population-based evolutionary computation (EC) algorithms for solving such complicated problems [6]. After decades of development, many powerful EC algorithms have been proposed, e.g., genetic algorithm [7], differential evolution [8], particle swarm optimization [9] and others [10]–[12]. At the same time, many researchers focus on introducing various novel search strategies into these standard EC algorithms to further improve their performance [13]–[15]. The good news is that these EC algorithms and their variants solve many real-world problems successfully, such as feature selection [16], power minimization [17], fashion design [18], and dry gear hobbing [19].

Bacterial foraging optimization (BFO) [20], as an important branch of the EC community, has attracted wide attention since it was proposed. Many practitioners developed lots of BFO's variants to solve different types of optimization problems, such as multi-objective optimization [21], high-dimensional optimization [22], and constrained optimization [23]. Besides, Muñoz et al. simplified the structure of the standard BFO and further improved its performance [24]; Kim et al. hybridized the genetic algorithm and the BFO to obtain a more powerful performance by integrating the advantages of both algorithms [25]. Although these modifications improve the performance of the BFO to some extent, there is still room to perfect the BFO by overcoming remaining shortcomings, such as, slow convergence and high calculation cost. That is why this paper came into being.

The first objective of this paper is to propose a reverse chemotaxis strategy to guide bacterial individuals to quickly find current potential directions by making full and efficient use of local fitness information. The second objective is to significantly simplify the overall optimization framework of the standard BFO by integrating the *elimination-dispersal* operation and the *reproduction* operation together. We finally analyze the effectiveness of our proposal and give some open topics for discussion.

Following this introduction Section, we briefly summarize the optimization principles of the standard BFO in Section II. The proposed two strategies are comprehensively described in Section III, and a set of controlled experiments is designed to evaluate the performance of our proposal in Section IV. We

The work described in this paper is supported by the National Natural Science Foundation of China of China (71971143).

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

Algorithm 1 The general optimization framework of the standard BFO. N_{ed}, N_{re}, and N_c are preset parameters for controlling the step size of the iteration.

```
Initialize the population randomly.
for $j = 0; j < N_{ed}; j++$ do
    for $k = 0; k < N_{re}; k++$ do
        for $l = 0; l < N_c; l++$ do
            Perform chemotaxis operations;
        end for
        Perform reproduction operations;
    end for
    Perform elimination and dispersal operations;
end for
Output the optimal solution found.
```

finally analyze the experimental results and conclude our work in Sections V and VI, respectively.

II. BACTERIAL FORAGING OPTIMIZATION

Inspired by the social foraging behavior of *E. coli* bacteria, the BFO repeatedly mimics the behavior of swallowing food to find the global optimum. Similar to most EC algorithms, a bacterial individual represents a candidate solution, and the BFO randomly generates multiple bacterial individuals to form an initial population. Then, all bacterial individuals make their next foraging decisions to search for nutrition areas by sharing information with others. After all bacterial individuals undergo the above operations (i.e., chemotaxis operations) many times, the propagation process of survival of the fittest and random migration is triggered to update the current population. The bacterial individuals who survive to the next generation start their foraging behavior again, and the above operations are performed repeatedly until a termination condition is satisfied. Algorithm 1 gives the optimization framework of the standard BFO that mainly consists of three nested operations: *chemotaxis*, *reproduction*, and *elimination and dispersal*.

Chemotaxis operation is the core operation of the BFO that directly determines the BFO performance by simulating the foraging behavior of *E. coli* bacteria. There are two movement patterns in the original *chemotaxis* operation: tumble and swim. Moving a unit step in a randomly selected direction is defined as tumble. When bacterial individuals complete a tumble and only their fitness is improved, they will continue to move in the current direction until their fitness no longer improves or reaches a predetermined maximum number of movement steps. Otherwise, they will stop moving and stay in their tumbled positions for the selection of the next generation. This process is defined as swim. Bacterial individuals experience multiple aforementioned movement patterns to gradually move towards their preferred nutrient gradient and avoid entering harmful environments.

Reproduction operation simulates asexual binary fission of *E. coli* bacteria to generate offspring individuals. Generally speaking, bacteria can generate a large number of offspring after a period of growth. If nutrition is sufficient, bacteria can show exponential growth. Since the propagation process of bacteria also follows the principle of "survival of the fittest", taking the cumulative value of the fitness of bacterial individuals during the *chemotaxis* operation as the selection criterion. The last half of the bacterial individuals are directly abandoned according to the accumulated fitness value, and the first half of the bacterial individuals split into two exactly same offspring individuals to form a new population.

Elimination and dispersal operation is designed to increase the diversity of the population by dispersing bacterial individuals to any position in the search space with a low trigger probability. This inspiration comes from the observation of real bacteria, we note that the environment in which bacteria live may change gradually or suddenly for various reasons. For example, a significant local temperature rise may kill a group of bacteria currently living in an area with a high concentration of nutrient gradient. These phenomena thus are introduced into the BFO framework to simulate mutations.

III. REVERSE CHEMOTAXIS STRATEGY AND STRUCTURAL SIMPLIFICATION STRATEGY

The BFO has become a hot topic in the community of meta-heuristic algorithms thanks to its many advantages, e.g., parallel search, strong robustness, and wide applicability. However, it is undeniable that the BFO performance depends heavily on numerous parameter settings that usually need to be carefully tuned manually. Besides, compared with other EC algorithms, the BFO has a relatively complex structure and needs a lot of fitness evaluations to finally converge to the global optimum. We thus proposed two effective strategies to overcome these found defects while further improving the BFO performance. The first strategy (i.e., reverse chemotaxis strategy) uses local fitness information to quickly find potential areas rather than the random search; the second strategy (i.e., structural simplification strategy) combines *reproduction* operation and *elimination and dispersal* operation together to simplify the optimization structure and reduce the number of required parameters significantly.

A. Reverse Chemotaxis Strategy

The original *chemotaxis* operation consists of two patterns: tumble and swim, and the tumble directly determines the direction of the subsequent swim operations. Unfortunately, once bacterial individuals converge to poor local areas, they can only keep tumbling until they find a new potential direction, and then start a new round of swim operations in that direction. Although the original *chemotaxis* operation can ensure that bacterial individuals constantly swim to better areas, it may reduce the convergence speed greatly. Besides, it is difficult for bacterial individuals to jump out of the trapped local areas. The first strategy thus helps bacterial individuals to accelerate convergence to potential areas and quickly jump out of local areas rather than random tumbling repeatedly.

When we only consider a tiny local area, it can be regarded as a smooth hyperplane. Based on this hypothesis, we can infer that the reverse area of the worthless direction may have a high possibility of potential, and bacterial individuals need to spend a high cost, i.e., a large number of fitness evaluations, to reach the area by random tumbling or swimming. As a preliminary attempt, we simply use the opposite of the current poor direction as the new direction for the next swim, and the swimming step is doubled to help escape the current predicament. Besides, we also use Gaussian mutations to perturb every dimension of the new direction, which can avoid oscillations in the original direction. Note that even if bacterial individuals' fitness is not improved, they do not stop but continue to swim in the new direction until the maximum number of swims is reached, which can help bacterial individuals find a better direction quickly.

$$Dir_{new} = -2.0 \times Dir_{current} + Gaussian(0, \sigma) \tag{1}$$

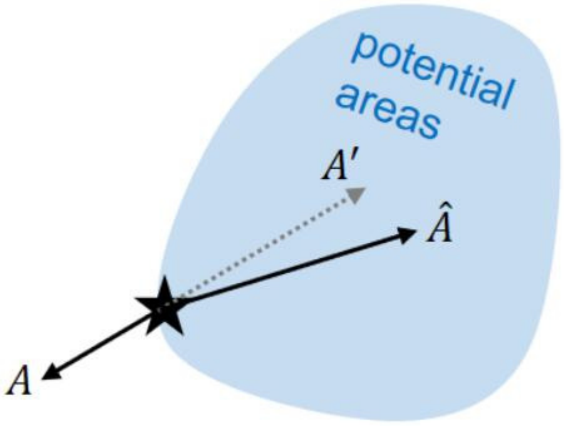

Fig. 1. The chemotactic effect of our proposed strategy. A means that a bacterial individual tumbles or swims to a poor area, and $A^{'}$ is the opposite direction of A and the swimming step is doubled. $\hat{A}$ is the next swimming direction after $A^{'}$ experiences Gaussian mutation disturbance.

In short, we can use Eq. (1) to summarize the new proposed chemotaxis strategy, where bacterial individuals can swim to better areas without interruption, i.e., they do not need to stop foraging behavior even when they move to poor areas. Here, we set σ equal to the swimming step in our following experimental evaluations, and Fig. 1 demonstrates the reverse effect of our proposed chemotaxis strategy.

B. Structural Simplification Strategy

Since the standard BFO uses three nested loops, it not only increases the complexity of the BFO structure, but also makes the minimum number of fitness evaluations required not to be ignored. At the same time, we notice that the *reproduction* operation can retain half of the potential bacterial individuals, but reduces the diversity of the population dramatically. On the contrary, although the *elimination and dispersal* operation can increase the diversity by randomly spreading bacterial individuals, it does not hesitate to abandon excellent bacterial individuals that have been painstakingly found. We thus realize that these two operations are complementary and could be merged together to avoid their defects.

Algorithm 2 The general optimization framework of the standard BFO combined with our proposed two strategies. Note that all bacterial individuals are sorted according to their health. PS: population size, P_{ed}: eliminate probability. Steps 4-11 are the implementation details of our proposed structural simplification strategy.

```
1:  Initialize the population randomly.
2:  while A termination condition is not satisfied do
3:      Perform the proposed reverse chemotaxis operation;
4:      for $i = (PS/2); i < PS; i++$ do
5:          if $rand(0,1) < P_{ed}$ then
6:              Replace the $i$-th unhealthy using the original elimination and dispersal operation;
7:          else
8:              Replace the $i$-th unhealthy with $(i - (PS/2))$-th bacterial individual using the original reproduction operation;
9:              Use Gaussian mutations to perturb the new $i$-th bacterial individual.
10:         end if
11:     end for
12: end while
13: Output the optimal solution found.
```

The second strategy, i.e., structural simplification strategy, focuses on achieving the balance between the elite selection and the diversity of the population well by simplifying the BFO's optimization structure. Since the original *reproduction* operation copy half of the healthy bacterial individuals directly to cover unselected bacterial individuals, our proposal focuses on reforming the sources of these unhealthy bacterial individuals replaced.

As an attempt, we integrate the *elimination and dispersal* operation into the *reproduction* operation as a new means to replace unhealthy bacterial individuals, i.e., randomly generated bacterial individuals may replace unhealthy bacterial individuals. Thus, there are two ways to update unhealthy bacterial individuals; one is the original *reproduction* operation, and the other is the original *elimination and dispersal* operation. To avoid having exactly the same bacterial individuals and further increasing the diversity of the population, we also use Gaussian mutations to perturb new bacterial individuals when they are copied from healthy bacterial individuals. In general, this strategy can not only prevent potential bacterial individuals from being replaced, but also increase the diversity of the population greatly. Finally, Algorithm 2 gives the combination framework of the standard BFO and our proposed two strategies.

IV. EXPERIMENTAL EVALUATIONS

We use 28 benchmark functions from CEC 2013 test suites [26] to evaluate the performance of our proposed strategies, and Table II gives their variable ranges, optimum fitness, and various characteristics, e.g., shifted, rotated, unimodal, and multi-modal. We select the standard BFO as the baseline algorithm and combine it with our proposed two strategies, then independently run these two algorithms (i.e., (the BFO + two proposed strategies) and the BFO) on three different dimensions (i.e., 2-D, 10-D, and 30-D) of each benchmark function with 30 trial runs to avoid contingency. Table I shows the parameter configuration of the BFO used in our evaluation experiments.

TABLE I
THE PARAMETER SETTINGS OF THE STANDARD BFO ALGORITHM USED IN OUR EXPERIMENTS.

population size for 2-D, 10-D, and 30-D search	20
ss: step size	0.6
N_{ed}: number of elimination-dispersal steps	1
N_{re}: number of reproduction steps	2
N_{ch}: number of chemotaxis steps	10
N_{sl}: swim length	4
P_{ed}: eliminate probability	0.25
d_{attr}: depth of the attractant	0.1
w_{attr}: width of the attractant signal	0.2
h_{rep}: height of the repellant effect	0.1
w_{rep}: height of the repellant effect	10.0
MAX_{NFC}: max. # of fitness evaluations for search	$5,000 \times D$

We use the number of fitness evaluations instead of generations to terminate evaluation experiments for fair comparisons, and apply the Wilcoxon signed-rank test at the stop condition (i.e., the maximum number of fitness evaluations) to check significant differences between (the BFO + two proposed strategies) and the BFO. The detailed results of the statistical tests are summarized in Table III.

TABLE II
BENCHMARK FUNCTIONS: UNI=UNIMODAL, MULTI=MULTIMODAL, COMP.=COMPOSITION

No.	Types	Characteristics	Optimum fitness
F_1	Uni	Sphere function	-1400
F_2		Rotated high conditioned elliptic function	-1300
F_3		rotated Bent Cigar function	-1200
F_4		Rotated discus function	-1100
F_5		different powers function	-1000
F_6	Multi	Rotated Rosenbrock's function	-900
F_7		Rotated Schaffers function	-800
F_8		Rotated Ackley's function	-700
F_9		Rotated Weierstrass function	-600
F_{10}		Rotated Griewank's function	-500
F_{11}		Rastrigin's function	-400
F_{12}		Rotated Rastrigin's function	-300
F_{13}		Non-continuous rotated Rastrigin's function	-200
F_{14}		Schwefel's function	-100
F_{15}		Rotated Schwefel's function	100
F_{16}		Rotated Katsuura function	200
F_{17}		Lunacek BiRastrigin function	300
F_{18}		Rotated Lunacek BiRastrigin function	400
F_{19}		Expanded Griewank's plus Rosenbrock's function	500
F_{20}		Expanded Scaffer's F_6 function	600
F_{21}	Comp.	Composition Function 1 (n=5,Rotated)	700
F_{22}		Composition Function 2 (n=3,Unrotated)	800
F_{23}		Composition Function 3 (n=3,Rotated)	900
F_{24}		Composition Function 4 (n=3,Rotated)	1000
F_{25}		Composition Function 5 (n=3,Rotated)	1100
F_{26}		Composition Function 6 (n=5,Rotated)	1200
F_{27}		Composition Function 7 (n=5,Rotated)	1300
F_{28}		Composition Function 8 (n=5,Rotated)	1400

TABLE III
THE STATISTICAL TEST RESULTS OF THE WILCOXON SIGNED-RANK TEST FOR AVERAGE FITNESS OF 30 TRIAL RUNS BETWEEN (THE BFO + OUR PROPOSED TWO STRATEGIES) AND THE BFO. $A \gg B$ AND $A > B$ MEAN THAT A IS SIGNIFICANT BETTER THAN B WITH SIGNIFICANT LEVELS OF 1% AND 5%, RESPECTIVELY. $A \approx B$ MEANS THAT ALTHOUGH A IS BETTER THAN B, THERE IS NO SIGNIFICANT DIFFERENCE BETWEEN THEM. SIMPLIFIEDBFO: (THE BFO + OUR PROPOSED TWO STRATEGIES).

	2D	10D	30D
F_1	BFO $\approx$ SimplifiedBFO	BFO $\gg$ SimplifiedBFO	BFO $\gg$ SimplifiedBFO
F_2	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\gg$ BFO
F_3	SimplifiedBFO $>$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_4	SimplifiedBFO $>$ BFO	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\approx$ BFO
F_5	SimplifiedBFO $\gg$ BFO	BFO $\approx$ SimplifiedBFO	SimplifiedBFO $\approx$ BFO
F_6	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\gg$ BFO
F_7	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_8	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_9	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{10}	SimplifiedBFO $\gg$ BFO	BFO $\gg$ SimplifiedBFO	BFO $\gg$ SimplifiedBFO
F_{11}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{12}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{13}	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{14}	SimplifiedBFO $>$ BFO	SimplifiedBFO $\approx$ BFO	BFO $\gg$ SimplifiedBFO
F_{15}	SimplifiedBFO $>$ BFO	BFO $\approx$ SimplifiedBFO	BFO $\gg$ SimplifiedBFO
F_{16}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{17}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{18}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{19}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{20}	BFO $\approx$ SimplifiedBFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{21}	SimplifiedBFO $\gg$ BFO	BFO $\approx$ SimplifiedBFO	SimplifiedBFO $>$ BFO
F_{22}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\approx$ BFO	BFO $\approx$ SimplifiedBFO
F_{23}	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $>$ BFO	BFO $>$ SimplifiedBFO
F_{24}	SimplifiedBFO $>$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{25}	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{26}	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{27}	SimplifiedBFO $\approx$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO
F_{28}	SimplifiedBFO $>$ BFO	SimplifiedBFO $\gg$ BFO	SimplifiedBFO $\gg$ BFO

V. DISCUSSIONS

We start the discussion from the perspective of the superiority of our proposed two strategies. The original *chemotaxis* operation uses the tumble pattern to find promising directions and then uses the swim pattern to track these found directions, which can ensure that bacterial individuals continue to converge to better areas. Unfortunately, bacterial individuals need a large number of resources (fitness evaluations) to get out of trouble when they fall into local areas because a random search strategy is employed to simulate the tumble pattern. We thus propose a reverse chemotaxis strategy to guide the evolution of bacterial individuals even if they swim to poor areas by making full use of the information of the local fitness landscape. The core idea of this strategy is to avoid searching for found poor directions and to favor unexplored potential areas. We simply use the opposite directions of poor directions and Gaussian perturbation to realize our idea. There is no doubt that other implementation methods are also acceptable. For example, we can give different search probability to the surrounding areas of bacterial individuals to suppress the exploration of poor areas. In addition, the proposed strategy does not need to introduce any new parameters, and the additional number of fitness evaluations is also acceptable from the point of view of performance improvement. We can say that the reverse chemotaxis strategy is effective and easy to use.

The benefit of the other strategy (i.e., structural simplification strategy) is to simplify the BFO structure greatly and reduce the number of necessary parameters. Although the *elimination-dispersal* and *reproduction* operations can bring performance improvements, they also bring some new defects mentioned above. Besides, all parameters from these two operations need to be carefully tuned when the BFO is applied to different optimization scenarios. Once the parameter configuration is unreasonable, the performance will be severely degraded or even hinder its practicability. Fortunately, we observe that there is a complementary relationship between these two operations by analyzing their working principles. As a possible attempt, we integrate the *elimination-dispersal* operation as part of the *reproduction* operation to provide a new way to update the population. In other words, this strategy can not only provide diverse bacterial individuals in the next generation, but also prevent potential bacterial individuals from being randomly replaced. In addition, the proposed strategy makes the standard three-layer nested loop degenerate into a single-layer loop, and some parameters (e.g. N_{ed}, N_{re}, and N_{ch}) can also be discarded. Finally, this strategy also does not need to introduce any additional parameters and fitness evaluations. We can say that the structural simplification strategy is a *low cost, high return* strategy.

Secondly, we would like to discuss the scalability of our proposed two strategies. Not limited to the standard BFO, these two strategies can also be applied to other BFO's variants. We do not need to modify their optimization framework greatly and just replace the corresponding operations with our proposed strategies. In addition, these two strategies are also separable, which means we can use either instead of both strategies to combine with the BFO. Actually, we also encourage other ways to implement our proposed strategies. For example, we can construct guiding vectors [27], [28] by using local fitness information to avoid inefficient search and accelerate the convergence; we can also use other mutations e.g., lévy flights [29] and chaotic mutation [30], to replace the random mutation in the original *elimination-dispersal* operation. In short, our proposed two strategies have strong applicability and scalability.

Next, we offer several potential topics for discussions. Since every coin has two sides, our proposal can simplify the optimization structure and improve the BFO performance, but it also causes new problems. The biggest problem is that although our proposal can accelerate the convergence speed in the early stage obviously, the convergence precision in the later stage needs to be further improved. In other words, how to ensure convergence precision in the later stage is a topic worthy of further study. Here, we also give some possible approaches that have not been experimentally proved. For example, we can develop an adaptive strategy for tuning these parameters to gradually emphasize the exploitation ability as the population converges; we can also build multiple approximate fitness models for optimization problems through different regression methods, and then extract hidden information from them to avoid searching in the area without potential and speed up the search process. Anyway, our proposed two strategies still have a lot of room for improvement.

Finally, we applied the Wilcoxon signed-rank test to check the significant differences between (the BFO + two proposed strategies) and the BFO. The results of the statistical test confirmed that our proposal can improve the BFO performance significantly, especially for complex problems both on low-dimensional and high-dimensional. However, the acceleration effect of our proposed strategies on high-dimensional unimodal problems is not obvious, and even worse on $F1$. This may be because our proposed strategies sacrifice convergence speed in exchange for the diversity of the population, and experimental results also support our opinions. It also enlightens us to emphasize the balance between different search capabilities (i.e., exploration and exploitation) for different optimization problems. For F_{10}, F_{14}, and F_{15}, our proposal does not achieve the expected performance in high dimensions, and the real reasons need further analysis in our future work.

VI. CONCLUSION

We proposed two novel strategies to further improve the BFO performance while simplifying its optimization structure. The first strategy (i.e., reverse chemotaxis strategy) uses local fitness information to quickly guide bacterial individuals to converge to better areas instead of random search; the second strategy (i.e., structural simplification strategy) can achieve a better balance between the elite selection and the diversity by merging the original two operations. The experimental results confirmed that our proposed strategies are effective and promising, especially can accelerate the convergence of complex problems.

In our future work, we will observe more survival mechanisms of real bacteria and introduce them into the BFO to develop more powerful variants. Besides, we also try to use historical information collected from the evolutionary search to enhance the search efficiency of the BFO, and propose an intelligent control strategy to maintain high performance by tuning parameters in real-time.

REFERENCES

[1] A. L. Cauchy, "Méthode générale pour la résolution des systémesd' équations simultanées," *Comptes Rendus Hebd. Séances Acad. Sci*, vol. 25, pp. 536–538, 1847.

[2] A. Charnes and C. E. Lemke, "Computational problems of linear programing," in *the 1952 ACM National Meeting (Pittsburgh)*, 1952, pp. 97–98.

[3] D. F. Votaw, "Methods of solving some personnel-classification problems," *Psychometrika*, vol. 17, no. 3, pp. 255–266, 1952.

[4] P. Wolfe, "Recent developments in nonlinear programming," vol. 3, no. C, pp. 155–187, 1962.

[5] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, ser. Nonlinear Programming: Theory and Algorithms, 2005, pp. 1–853.

[6] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

[7] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

[8] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341– 359, 1997.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 4, Perth, Western Australia, 1995, pp. 1942–1948.

[10] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[11] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," ser. Lecture Notes in Computer Science, vol. 6145 LNCS, no. PART 1, Beijing, China, June 2010, pp. 355–364.

[12] X. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[13] J. Yu and H. Takagi, "Acceleration for fireworks algorithm based on amplitude reduction strategy and local optima-based selection strategy," in *8th International Conference on Swarm Intelligence*, 2017, pp. 477–484.

[14] J. Yu, Y. Pei, and H. Takagi, "Competitive strategies for differential evolution," in *The 2018 IEEE International Conference on Systems, Man, and Cybernetics*, 2018, pp. 268–273.

[15] J. Yu, Y. Tan, and H. Takagi, "Accelerating fireworks algorithm with an estimated convergence point," in *9th International Conference on Swarm Intelligence*, 2018, pp. 263–272.

[16] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[17] A. A. A. Esmin, G. Lambert-Torres, and A. C. Zambroni de Souza, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 859–866, 2005.

[18] H. . Kim and S. . Cho, "Application of interactive genetic algorithm to fashion design," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 6, pp. 635–644, 2000.

[19] Q. Xiao, C. Li, Y. Tang, J. Pan, J. Yu, and X. Chen, "Multi-component energy modeling and optimization for sustainable dry gear hobbing," *Energy*, vol. 187, 2019.

[20] K. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[21] B. Niu, H. Wang, J. Wang, and L. Tan, "Multi-objective bacterial foraging optimization," *Neurocomputing*, vol. 116, pp. 336–345, 2013.

[22] S. Dasgupta, A. Biswas, S. Das, B. K. Panigrahi, and A. Abraham, "A micro-bacterial foraging algorithm for high-dimensional optimization," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 785–792.

[23] B. Niu, J. Wang, and H. Wang, "Bacterial-inspired algorithms for solving constrained optimization problems," *Neurocomputing*, vol. 148, pp. 54–62, 2015.

[24] M. A. Muñoz, S. K. Halgamuge, W. Alfonso, and E. F. Caicedo, "Simplifying the bacteria foraging optimization algorithm," in *2010 IEEE Congress on Evolutionary Computation*, 2010, pp. 4095–4101.

[25] D. H. Kim, A. Abraham, and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Information Sciences*, vol. 177, no. 18, pp. 3918–3937, 2007.

[26] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," 2013. [Online]. Available: http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf

[27] J. Li, S. Zheng, and Y. Tan, "The effect of information utilization: Introducing a novel guiding spark in the fireworks algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 153–166, 2017.

[28] Y. H. Li, J. Yu, H. Takagi, and Y. Tan, "Accelerating fireworks algorithm with weight-based guiding sparks," in *10th International Conference on Swarm Intelligence*, 2019, pp. 257–266.

[29] X. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World Congress on Nature and Biologically Inspired Computing*, 2009, pp. 210–214.

[30] S. Q. Liu, Y. Zhou, and W. P. Yan, *Research on chaos theory improved evolutionary algorithm*, ser. Advanced Materials Research, 2014, vol. 926-930.