# An Effective Bacterial Foraging Optimization Based on Conjugation and Novel Step-Size Strategies

Ming Chen, Xiaojun Qiu and Hong Wang

August 7, 2019

# An Effective Bacterial Foraging Optimization Based on Conjugation and Novel Step-Size Strategies

Ming Chen
College of Economics
Shenzhen University
Shenzhen, Guangdong, China, 518060

Xiaojun Qiu and Hong Wang*
College of Management
Shenzhen University
Shenzhen, Guangdong, China, 518060
Email: ms.hongwang@gmail.com

***Abstract*—Bacterial Foraging Optimization (BFO) is an effective metaheuristic algorithm that has been widely applied to the real world. Despite outstanding computing functionality, BFO algorithms can barely avoid premature convergence induced by easy trapping in local optima. To improve the computing functionality of BFO algorithm, this paper presents an improved BFO algorithm that employs a novel step-length evolution strategy. Also, the improved BFO algorithm adopts Lévy flight strategy proposed in LPBFO and the conjugation strategy proposed in BFO-CC. By combining the three strategies associatedlly, the proposed Conjugated Novel Step-size BFO algorithm(CNS-BFO) strikes an outstanding balance between exploitation and exploration, effectively mitigating the problem of premature convergence in BFO algorithm. Experimental results comparing with several similar algorithms on 8 benchmark functions are conducted to demonstrate the efficiency of the proposed CNS-BFO algorithm.**

***Keywords*—BFO; Lévy flight; conjugation; adaptive step size**

## I. Introduction

Metaheuristic algorithm, the so-called nature-inspired algorithm, has drawn considerable attention since inception. Because of its efficacy in tackling optimization and computation problems (especially multi-minimum and multi-constraints problems), metaheuristic algorithm is among the most popular algorithms for researchers. Compared to traditional algorithms, metaheuristic algorithm is much more capable of solving discontinuous and multi-modal problems[1]. Due to its advantages in solving realistic problems, metaheuristic algorithm is incessantly studied and modified to get better performance.

Metaheuristic algorithms are generally divided into single-solution metaheuristic algorithms and population-based metaheuristic algorithms in terms of the way they search for best solutions[2]. single-solution metaheuristic algorithms center on improving single candidate solution. A typical example of single-solution metaheuristic algorithms is simulated annealing algorithm(SA)[3], which simulates the heating and cooling process to control temperature and energy in order to find approximate global optima. Population-based metaheuristic algorithms center on improving multiple candidate solutions. The most popular population-based metaheuristic algorithms include particle swarm optimization algorithm (PSOA)[4], bacterial foraging optimization algorithm (BFOA)[5], ant colony optimization algorithm (ACOA)[6], genetic algorithm (GA)[7], differential evolution algorithm (DEA)[8] and so on. The nature-inspired algorithms are derived from natural phenomena and biological behaviors. For example, particle swarm optimization algorithm[4] is inspired by swarm characteristics of birds seeking foods, and genetic algorithm[7] simulates the process of natural selection and genetic construction.

Performance of population-based metaheuristic algorithms is mostly determined by the process of exploitation and exploration[9], which focus on converging to best candidate solution and adding diversification separately. Exploitation is introduced to perform local search so as to speed up convergence, while usually leading to premature convergence in which the algorithm can get stuck in local optima. In contrast to exploitation, exploration centers on seeking various solutions by broadening search area to provide sufficient candidates for the solution, while usually slowing down the convergence and wasting computational efforts[10]. Lopsided emphasis on either search process can lead to the inefficacy of an algorithm, and hence researchers have exerted enormous efforts in balancing the exploration and exploitation in proposed metaheuristic algorithms[11].

Bacterial foraging optimization algorithm is proposed by Passino in 2002[5]. Under observation, researchers found that Escherichia coli approach nutrients by steps and that the bacteria in unhealthy conditions can be starved and die. Based on the observation, Passino constructed four basic process-chemotaxis, swarming, reproduction, and elimination in the original BFO algorithm. Since the inception of BFO algorithm, it is broadly employed in all sorts of fields, including robotic cells[12], power generation[13], sensor network[14], image segmentation[15], and so on.

Numerous novel BFO algorithms have been proposed in recent years. Hybridization of BFO algorithm with other metaheuristic algorithm attracts much attention from researchers. Researchers proposed hybrid BFO algorithms combined with all sorts of algorithms, such as GABFO[16], BSO[17], and DEBFO[18]. By taking full advantages of other algorithms, hybrid algorithm shows extraordinary advantages in specific fields, such as PID controller tunings[19], economic dispatch problems[20] and so on.

Structure modifications of BFO are also proposed by researchers. To reduce computational time, researchers have re-designed construction of BFO algorithm, such as structure-redesign-based BFO (SRBFO) proposed by Niu et al. in 2014[21], and IMBFOA proposed by Ocaña in 2016[22]. Recently, coevolutionary structure-redesigned-based BFO is modeled by Niu et al. and achieve satisfactory performance[23]. The redesigns of algorithm structures effectively save computational efforts of BFO algorithm.

Apart from hybridization of other algorithms and the modification of structure, researchers integrate communication mechanism in BFO algorithms. Social learning mechanism is incorporated in classical BFO algorithm by Yan et al. in 2012[24]; adaptive comprehensive learning mechanism (ALCBFO)[25] is introduced by L. Tan et al. in 2015 in order to keep balance between exploitation and exploration; novel chemotaxis and conjugation strategies (BFO-CC) are employed by Yang et al. in 2016[26] and contribute to the improvement of BFO algorithm. Recently, Pang B et al. proposed LPBFO[27], which incorporates Lévy flight to the reconstruction of BFO's step-size, making progress in addressing the limitation of premature convergence in BFO algorithm.

Besides, chemotaxis step improvement is thoroughly considered by researchers. Adaptive bacterial foraging optimization algorithm (ABFO) is proposed by Dasgupta et al. in 2009[28]. Linear decreasing BFO (BFO-LDC) and non-linear decreasing BFO (BFO-NDC) are modeled by Niu et al. in 2011[29]. Gravitation search strategy is combined with chemotactic step by Zhao et al. in the proposed effective BFO (EBFO)[30].

Despite huge efforts exerted by researchers, deficiencies of premature convergence and slow convergence speed remain unsolved. In order to further the improvement of BFO algorithm, the paper proposes a Conjugated Novel Step-size BFO algorithm(CNS-BFO) which employs a novel chemotaxis step-size evolution strategy. Moreover, the proposed CNS-BFO algorithm employes conjugation strategy proposed in BFO-CC algorithm and Lévy flight strategy proposed in LPBFO, significantly ameliorating the deficiency of premature convergence, improving the computational efficiency of BFO algorithm.

The rest of the contents is organized as the followings. Classical BFO algorithm is illustrated in section two. In section three, the paper introduces the proposed Conjugated Novel Step-size BFO algorithm. Experiments are conducted to test the performance of the CNS-BFO algorithm in section four. Finally, section five concludes the proposed CNS-BFO algorithm.

## II. Classical BFO Algorithm

Bacterial foraging optimization proposed by Passino[5] in 2002 simulates behaviors of *E. coli* bacteria. To summarize, the classical bacterial foraging optimization algorithm consists of three basic behaviors: chemotaxis, reproduction, and elimination-dispersal. Subsections written below will demonstrate how the three behaviors work.

### A. Chemotaxis

*E. coli* bacteria move towards places of abundant nutrients by rotating flagella. The rotation of flagella can be in either clockwise direction or counterclockwise direction. Counterclockwise direction rotation pushes the bacteria to move towards nutrients, while clockwise direction leads to bacteria' s tumbling away from the current position for searching nutrients. By combining both actions, bacteria perform chemotaxis to approach nutrient.

Suppose bacterium $i$ is denoted as $\theta^i(j,k,l)$, in which $j$ denotes the $j$-th chemotaxis, $k$ denotes the $k$-th reproduction, and $l$ denotes the $l$-th elinimation& dispersal, the behaviors of the $i$-th bacterium can be demonstrated as follows:

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

where $C(i)$ denotes the step size of chemotaxis and $\Delta(i)$ denotes random direction vector in which elements are within $[-1,1]$ .

If the health status gets better after tumbling, bacteria will keep on swimming for several steps, in order to further approach optimal solutions.

### B. Reproduction

After the stage of chemotaxis, each bacterium is assigned with different health status. The health status is evaluated by the summation of the fitness of each bacterium. The lower the total fitness value, the healthier the bacterium. The function for calculating health status can be demonstrated as follows:

$$J_{health}=\sum_{j=1}^{N_c} J(i,j,k,l) \tag{2}$$

where $N_c$ denotes the number of chemotactic behaviors and $J$ is the evaluated value of the position of the $i$ bacterium at the $k$ reproduction and the $l$ ellimination. In classical BFO algorithm, only the first half bacteria ranked by health status can survive, with their offsprings supplant the other half of bacteria. Reproduction is designed to strenghthen local search ability of BFO while maintaining stable population of bacteria population. The function for reproduction can be demonstrated as follows:

$$\theta^{i+Sr}(j,k,l)=\theta^i(j,k,l) \tag{3}$$

where $Sr$ is half of the population size.

### C. Elimination & Dispersal

For fortifying the ability of global search, classical PSO algorithm introduces the mechanism of elimination & dispersal. After stages of chemotaxis and reproduction, bacteria will encounter death induced by adverse environment by a possibility $P_{ed}$. Bacteria will be dispersed to random positions within the search area. By introducing stage of elimination & dispersal, the diversity of bacteria increases and results in stronger global search ability.

## III. Conjugated Novel Step-size BFO Algorithm

### A. Lévy Flight Step Length Strategy

Levy distribution, proposed by Lévy Paul[31], is a special inverse-gamma distribution, whose formula can be expressed as $P_s = s^{-\lambda}$, in which $s$ is a step size and $\lambda$ is between

1 and 3. Subject to Lévy distribution, Lévy flight is a sort of random moving which can generate frequent small step sizes and occasional larger step sizes for bacteria. The traits of Lévy flight makes bacteria move with ununiformed step size, strengthening BFO's ability of exploitation and exploration at the same time: on the one hand, frequent small step sizes fortify the local search ability of bacteria, whereby the exploitation ability of the algorithm can be improved. On the other hand, occasional larger steps prevent bacteria from over concentrating on exploiting the current position, guiding bacteria to a broader search area, whereby the exploration ability of BFO can be strengthened.

The paper adopts the Lévy flight strategy modified by Bao Pang et al. in 2018[27]. The formula of the adaptive step-size is as follows:

$$C^{'}(i) = \frac{\alpha}{t(i)}|s| \tag{4}$$

where $\alpha$ is a parameter for controlling the strength of change in step-size, $t(i)$ is the number of chemotaxis the $i$ bacterium has gone through. As bacterium experiences more chemotaxis, the step size will dwindle as $t(i)$ gets larger, and hence bacteria can become more concentrated on exploiting the current position. $s$ is calculated by:

$$s = \frac{u}{|v|^{\frac{1}{\beta}}} \tag{5}$$

in which $\beta$ is a random number between 0.3 and 1.99. $u$ and $v$ are two normal random variables with means equal to zero and with variances set as $\sigma_u$ and $\sigma_v$ respectively. $\sigma_u$ and $\sigma_v$ is calculated as follows:

$$\sigma_u = \frac{\Gamma(1+\beta)sin(\frac{\pi\beta}{2})}{\Gamma[\frac{(1+\beta)}{2}]2^{\frac{\beta-1}{2}}\beta} \tag{6}$$

$$\sigma_v = 1 \tag{7}$$

### B. Novel Step Evolution Strategy

In the original BFO algorithm, chemotaxis step length is set fixed for each bacterium. Fixed step size saps bacterium's strength to exploit, resulting in a greater possibility of falling into local optima. To address the limitation, the other step-size evolution strategy is applied here. After bacteria and their step sizes are ranked and reproduced, step lengths of bacteria will be updated by the following equation:

$$C^{''}(i, k+1) = C(i, k+1)\boldsymbol{b} \tag{8}$$

$$C(i, k+1) = [C^{''}(i, k+1) - C(i, k+1)] \cdot (1 - \frac{1}{lk+1}) \tag{9}$$

where $\boldsymbol{b}$ denotes a $p \times 1$ vector, $p$ is the dimension of bacteria.

This novel step-size evolution strategy is introduced to make bacteria move adaptively and efficiently. As moving step sizes significantly decide the ability of bacteria to search glocal optimum, careful moving can dramatically improve the odds of finding the global optimum. The novel step length evolution strategy allows bacteria to move with longer step sizes in the beginning, and make them budge at later stage. Combined with Levy-flight step length strategy, the adaptive moving strategy efficiently mitigates the problem of trapping into local optima.

### C. Conjugation Strategy

Conjugation strategy proposed in BFO-CC algorithm by Cuicui Yang et al. in 2016[26] is a novel strategy that simulates the process of transferring genetic material among creatures. The proposed BFO-CC algorithm introduces conjugation mechanism to bacteria, constructing a message exchange mechanism by which diversity of bacteria can be improved. By this way, conjugation strategy enhances bacteria's capability of searching for glocal best. In BFO-CC algorithm, each bacterium chooses another bacterium $\theta^{i^{'}}$ and a conjugation point $pt$ randomly. The formula for conjugation mechanism is given as follows:

$$\theta^{i}_{new} = \theta^{i}(j,k,l) + \boldsymbol{a} \cdot [\theta^{i^{'}}(j,k,l) - \theta^{i}(j,k,l)] \tag{10}$$

where $\boldsymbol{a}$ is a p-dimension random vector in which the values of the components $pt$ to $pt+L-1$ are random numbers uniformly distributed from 0 to 1, with the rest of components being 0. $L$ indicates the length of exchange of dimensions. By taking the conjugation strategy, the bacterium is mutated to learn from other bacteria, whereby avoid premature convergence.

### D. Process of The Improved BFO Algorithm

Pseudocode for the improved BFO algorithm is shown in algorithm 1.

## IV. Experiments And Analysis

To prove the efficiency of the proposed CNS-BFO algorithm, it is compared with other sorts of heuristic algorithms: GA[7] and PSO[4]. Besides, as a variant of BFO algorithm, CNS-BFO is compared with standard BFO[5] and several popular BFO variants: BFO-LDC[29], BFO-NDC[29], and BCO[32] algorithms. To make the comparison fair, algorithms are assigned with the same initialization positions and recommended parameters are adopted for comparison algorithms. The population sizes for all BFO algorithms are set to 50. The total number of $FEs$ is set to 10,000 for 15-D functions. All the algorithms are tested for ten times in all the benchmark functions. Parameters of BFO algorithms are set as follows: $N_c$ = 1,000, $N_{re}$ = 10, $N_{ed}$ = 2, $P_{ed}$ = 0.25, $Sr$ = $\frac{S}{2}$, $n$ = 30, $s$ = 2, and $g$ = 2. For CNS-BFO algorithm, additional parameters are set as follows: $\alpha$ = 1,000, $\beta$=1.5, $p_{con}$=0.2, $N_n$=30, $L$=2. To demonstrate the effectiveness of CNS-BFO, 8 popular benchmark functions are employed to test its performance. Benchmark functions include one basic unimodal function($f_1$), 6 basic multimodal function($f_2 - f_7$), and 1 2D multimodal function($f_8$). Benchmark functions used in evaluating algorithms are shown in Table.I. The test results are demonstrated in Fig.1 and Table.II. Note that the best results have been marked with boldface. Also, the standard deviations of the best value in each run are bracketed.

**Algorithm 1** Pseudocode of CNS-BFO Algorithm

```
Initialize parameters $d, S, N_s, N_c, N_{re}, N_{ed}, N_n, Sr, p_{ed}, p_{con}, \alpha, \beta, \delta_v, \delta_u, u, v, e, L$
for $l$ = 1,2,...,$N_{ed}$ do
  for $k$ = 1,2,...,$N_{re}$ do
    for $j$ = 1,2,...,$N_c$ do
      for $i$ = 1,2,...,$S$ do
        Set $t = j + (k-1)N_c + (l-1) \cdot rand(p, 1)$
        Compute fitness function $J^i$
        Set $J_{last}$ = $J^i$
        Tumble with Levy-Flight step size computed by equation (4)
        Move. Compute $J^i$
        Swim:
        while $m < N_s$ do
          if $J^i$ is smaller than $J_{last}$ then
            Update $\theta^i$
          else
            Stop
          end if
        end while
        Conjugate:
        Select a bacterium randomly
        if $rand < p_{con}$ then
          Choose a conjugation point $pt$, in which $1 \leq pt \leq pt - L + 1$
          Compute $\theta_{new}$ using equation (10)
          if $J^i < J_{last}$ then
            $\theta^i = \theta_{new}$
          end if
        end if
      end for
    end for
    Compute the health status $J^i$ for the $i$-th bacterium
    Sort bacteria by health status
    for $j$ =1 to $Sr$ do
      split
      for $e$ = 1 to $N_n$ do
        Update $C_{i,k+1}$ using equation (8) and (9)
      end for
    end for
  end for
  for m = 1,2,...,S do
    Eliminate each bacterium if $rand < p_{ed}$
  end for
end for
```

According to the test graphs, it is noticeable that the proposed CNS-BFO algorithm tremendously outperforms other algorithms on nearly all the benchmark functions. In the all test functions, CNS-BFO algorithm seeks out optimal global considerably faster than all the other algorithms. Moreover, global optima found by CNS-BFO algorithm is significantly better than other comparison algorithms. In the last test function, the performance of CNS-BFO algorithm is as good as that of PSO algorithm. Overall, CNS-BFO algorithm effectively mitigates the problem of premature convergence, whereby dramatically improves the efficiency of BFO algorithm.

TABLE I: Benchmark Functions

| Function | Name | n | SD | $f_{opt}$ |
|---|---|---|---|---|
| $f_1$ | Sphere | 15 | $[-100, 100]^n$ | 0 |
| $f_2$ | Rastrigin | 15 | $[-5.12, 5.12]^n$ | 0 |
| $f_3$ | NCRastrigin | 15 | $[-5.12, 5.13]^n$ | 0 |
| $f_4$ | Schwefel 1.2 | 15 | $[-100, 100]^n$ | 0 |
| $f_5$ | Schwefel 2.22 | 15 | $[-500, 500]^n$ | 0 |
| $f_6$ | Ackley | 15 | $[-32.768, 32.768]^n$ | 0 |
| $f_7$ | Griewank | 15 | $[-600, 600]^n$ | 0 |
| $f_8$ | Brain2D | 2 | $[-500, 500]^n$ | 0.397887 |

## V. Conclusion

Conjugated Novel Step-size BFO algorithm(CNS-BFO) is proposed to improve the performance of BFO algorithm. Improvements lie in the way of constructing a novel step evolution strategy and incorporating both Lévy flight step-size strategy and conjugation strategy. By modifying the moving strategy and introducing a learning mechanism, CNS-BFO strikes a balance between exploitation and exploration, whereby efficaciously improves the limitation of premature convergence in BFO algorithms. Benchmark functions tests are implemented to compare the performance of the CNS-BFO algorithm with those of other algorithms. The experimental results indicate that the proposed CNS-BFO algorithm significantly enhances the performance of BFO algorithm in both avoiding premature convergence and accelerating convergence. As a result, the proposed CNS-BFO algorithm will contribute to the perfection of the bacterial foraging optimization algorithm and even other metaheuristic algorithms.

## VI. Acknowledgements

This work is partially supported by the Natural Science Foundation of Guangdong Province (2016A030310074, 2018A030310575), Natural Science Foundation of Shenzhen University (85303/00000155), 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 (2017GWTSCX038).

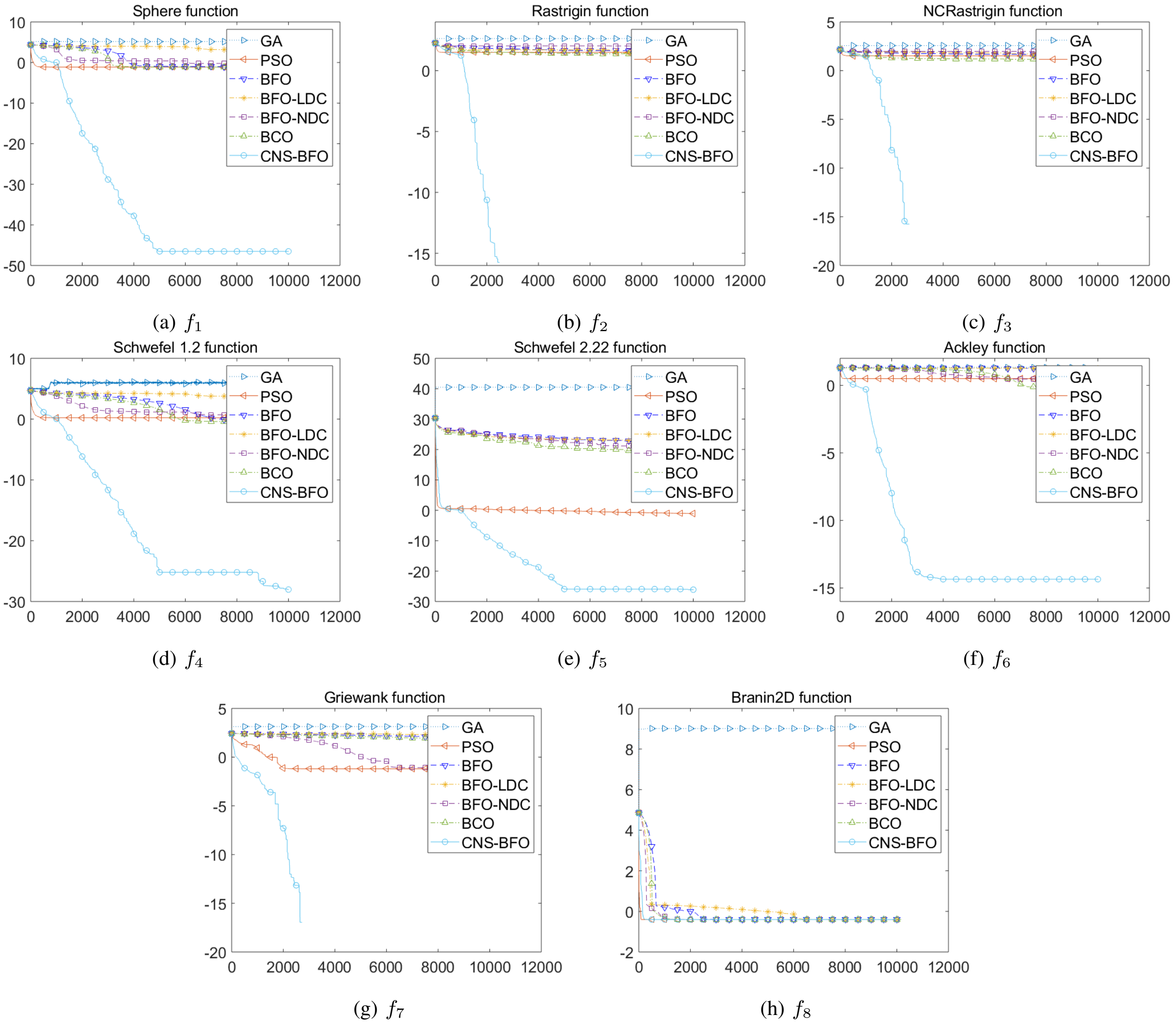
## References

[1] A. Kaveh and A. Zolghadr, “A novel meta-heuristic algorithm: tug of war optimization,” *Iran University of Science & Technology*, vol. 6, no. 4, pp. 469–492, 2016.

[2] Z. Beheshti and S. M. H. Shamsuddin, “A review of population-based meta-heuristic algorithms,” *Int. J. Adv. Soft Comput. Appl*, vol. 5, no. 1, pp. 1–35, 2013.

[3] P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing,” in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.

[4] J. Kennedy, “Particle swarm optimization,” *Encyclopedia of machine learning*, pp. 760–766, 2010.



(a) $f_1$

(b) $f_2$

(c) $f_3$

(d) $f_4$

(e) $f_5$

(f) $f_6$

(g) $f_7$

(h) $f_8$

Fig. 1: The Test Graphs on Benchmark Functions.

[5] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE control systems magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[6] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 2. IEEE, 1999, pp. 1470–1477.

[7] J. H. Holland *et al.*, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 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] X.-S. Yang, *Nature-inspired optimization algorithms*. Elsevier, 2014.

[10] M. A. Sahib, A. R. Abdulnabi, and M. A. Mohammed, "Improving bacterial foraging algorithm using non-uniform elimination-dispersal probability distribution," *Alexandria engineering journal*, vol. 57, no. 4, pp. 3341–3349, 2018.

[11] X.-S. Yang, S. Deb, and S. Fong, "Metaheuristic algorithms: optimal balance of intensification and diversification," *Applied Mathematics & Information Sciences*, vol. 8, no. 3, p. 977, 2014.

[12] A. Majumder, D. Laha, and P. N. Suganthan, "Bacterial foraging optimization algorithm in robotic cells with sequence-dependent setup times," *Knowledge-Based Systems*, vol. 172, pp. 104–122, 2019.

[13] B. Hernández-Ocaña, J. Hernández-Torruco, O. Chávez-Bosquez, M. B. Calva-Yáñez, and E. A. Portilla-Flores, "Bacterial foraging-based algorithm for optimizing the power generation of an isolated microgrid," *Applied Sciences*, vol. 9, no. 6, p. 1261, 2019.

[14] S. Deepa and D. Rekha, "Bacterial foraging optimization-based clustering in wireless sensor network by preventing left-out nodes," in *Intelligent Computing Paradigm: Recent Trends*. Springer, 2020, pp. 43–58.

[15] E. Wang, D. Tu, Y. Chen, and F. Zhang, "An improved bacterial foraging strategy for image segmentation," in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. IEEE, 2019, pp. 544–547.

[16] 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.

[17] A. Biswas, S. Dasgupta, S. Das, and A. Abraham, "Synergy of pso

TABLE II: Mean Values and Standard Deviation of the Eight Algorithms on Benchmark Function $f_1 - f_8$

| No. | GA | PSO | BFO | BFO-LDC | BFO-NDC | BCO | CNS-BFO |
|---|---|---|---|---|---|---|---|
| $f_1$ | 7.00E-02<br>(2.24E-02) | 2.24E-02<br>(5.29E-02) | 8.72E-02<br>(1.58E-02) | 7.98E+02<br>(1.23E+02) | 6.11E-01<br>(1.17E-01) | 7.63E-02<br>(9.00E-03) | **(2.34E-48)**<br>**(7.27E-48)** |
| $f_2$ | 4.50E+01<br>(3.36E+00) | 3.17E+01<br>(1.18E+01) | 4.23E+01<br>(3.55E+00) | 2.82E+01<br>(3.47E+00) | 1.12E+02<br>(1.67E+01) | 2.49E+01<br>(2.53E+00) | **0.00E+00**<br>**(0.00E+00)** |
| $f_3$ | 4.18E+01<br>(6.65E+00) | 3.19E+01<br>(8.82E+00) | 4.27E+01<br>(9.17E+00) | 3.76E+01<br>(1.27E+01) | 9.56E+01<br>(1.31E+01) | 1.28E+01<br>(2.72E+00) | **0.00E+00**<br>**(0.00E+00)** |
| $f_4$ | 4.88E-01<br>(1.08E-01) | 7.37E+02<br>(2.33E+03) | 5.16E-01<br>(9.23E-02) | 4.94E+03<br>(2.32E+02) | 4.90E+00<br>(9.59E-01) | 3.38E-01<br>(5.65E-02) | **8.62E-26**<br>**(2.55E-25)** |
| $f_5$ | 3.08E+22<br>(5.71E+22) | 7.62E+22<br>(1.24E+23) | 7.06E+21<br>(7.64E+21) | 5.96E+22<br>(8.36E+2)2 | 7.92E+19<br>(1.17E+20) | 7.16E+20<br>(2.26E+21) | **3.14E-27**<br>**(6.50E-27)** |
| $f_6$ | 1.74E+01<br>(5.23E-01) | 3.13E+00<br>(8.01E-01) | 1.75E+01<br>(4.51E-01) | 1.85E+01<br>(3.06E-01) | 2.75E+00<br>(2.08E-01) | 6.81E-01<br>(7.94E-02) | **4.09E-15**<br>**(1.12E-15)** |
| $f_7$ | 9.20E+01<br>(1.46E+00) | 6.72E+01<br>(9.60E+00) | 9.12E+01<br>(1.98E+00) | 2.09E+02<br>(9.14E-01) | 8.08E-02<br>(1.87E-02) | 5.18E+01<br>(7.06E+00) | **0.00E+00**<br>**(0.00E+00)** |
| $f_8$ | 3.98E-01<br>(6.24E-07) | **3.98E-01**<br>**(0.00E+00)** | 3.98E-01<br>(4.67E-07) | 3.98E-01<br>(6.46E-07) | 3.98E-01<br>(1.35E-05) | 3.98E-01<br>(1.38E-06) | **3.98E-01**<br>**(0.00E+00)** |

and bacterial foraging optimization—a comparative study on numerical benchmarks," in *Innovations in hybrid intelligent systems*. Springer, 2007, pp. 255–263.

[18] Biswas, Arijit and Dasgupta, Sambarta and Das, Swagatam and Abraham, Ajith, "A synergy of differential evolution and bacterial foraging optimization for global optimization," *Neural Network World*, vol. 17, no. 6, p. 607, 2007.

[19] H. Abdul-Ghaffar, E. Ebrahim, and M. Azzam, "Design of pid controller for power system stabilization using hybrid particle swarm-bacteria foraging optimization," *WSEAS Transactions on Power Systems*, vol. 8, no. 1, pp. 12–23, 2013.

[20] A. Y. Saber, "Economic dispatch using particle swarm optimization with bacterial foraging effect," *International Journal of Electrical Power & Energy Systems*, vol. 34, no. 1, pp. 38–46, 2012.

[21] B. Niu, Y. Bi, and T. Xie, "Structure-redesign-based bacterial foraging optimization for portfolio selection," in *International Conference on Intelligent Computing*. Springer, 2014, pp. 424–430.

[22] B. Hernández-Ocana, M. D. P. Pozos-Parra, and E. Mezura-Montes, "Improved modified bacterial foraging optimization algorithm to solve constrained numerical optimization problems," *Applied Mathematics and Information Sciences*, vol. 10, no. 2, pp. 607–622, 2016.

[23] B. Niu, J. Liu, T. Wu, X. Chu, Z. Wang, and Y. Liu, "Coevolutionary structure-redesigned-based bacterial foraging optimization," *IEEE/ACM transactions on computational biology and bioinformatics*, 2017.

[24] X. Yan, Y. Zhu, H. Zhang, H. Chen, and B. Niu, "An adaptive bacterial foraging optimization algorithm with lifecycle and social learning," *Discrete Dynamics in Nature and Society*, vol. 2012, 2012.

[25] L. Tan, F. Lin, and H. Wang, "Adaptive comprehensive learning bacterial foraging optimization and its application on vehicle routing problem with time windows," *Neurocomputing*, vol. 151, pp. 1208–1215, 2015.

[26] C. Yang, J. Ji, J. Liu, and B. Yin, "Bacterial foraging optimization using novel chemotaxis and conjugation strategies," *Information Sciences*, vol. 363, pp. 72–95, 2016.

[27] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy," *Applied Intelligence*, pp. 1–23, 2018.

[28] S. Dasgupta, S. Das, A. Abraham, and A. Biswas, "Adaptive computational chemotaxis in bacterial foraging optimization: an analysis," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 919–941, 2009.

[29] B. Niu, Y. Fan, H. Wang, L. Li, and X. Wang, "Novel bacterial foraging optimization with time-varying chemotaxis step," *International Journal of Artificial Intelligence*, vol. 7, no. A11, pp. 257–273, 2011.

[30] W. Zhao and L. Wang, "An effective bacterial foraging optimizer for global optimization," *Information Sciences*, vol. 329, pp. 719–735, 2016.

[31] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of levy stable stochastic processes," *Physical Review E*, vol. 49, no. 5, p. 4677, 1994.

[32] B. Niu and H. Wang, "Bacterial colony optimization," *Discrete Dynamics in Nature and Society*, vol. 2012, 2012.