IOP Conference Series: Materials Science and Engineering

Hybrid Bacterial Foraging Sine Cosine Algorithm for Solving Global Optimization Problems

To cite this article: Shuhairie Mohammad *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **917** 012081

International Conference on Technology, Engineering and Sciences (ICTES) 2020 IOP Publishing
IOP Conf. Series: Materials Science and Engineering **917** (2020) 012081 doi:10.1088/1757-899X/917/1/012081

Hybrid Bacterial Foraging Sine Cosine Algorithm for Solving Global Optimization Problems

Shuhairie Mohammad, Ahmad Nor Kasruddin Nasir, Nor Maniha Abdul Ghani, Raja Mohd Taufika Raja Ismail, Ahmad Azwan Abd Razak, Mohd Falfazli Mat Jusof and Nurul Amira Mhd Rizal

Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia.

shuhairie.mohammad@gmail.com

Abstract. This paper proposes a new hybrid algorithm between Bacterial Foraging Algorithm (BFA) and Sine Cosine Algorithm (SCA) called Hybrid Bacterial Foraging Sine Cosine Algorithm (HBFSCA) to solve global optimization problems. The proposed HBFSCA algorithm synergizes the strength of BFA to avoid local optima with the adaptive step-size and highly randomized movement in SCA to achieve higher accuracy compared to its original counterparts. The performances of the proposed algorithm have been investigated on a set of single-objective minimization problems consist of 30 benchmark functions, which include unimodal, multimodal, hybrid, and composite functions. The results obtained from the test functions prove that the proposed algorithm outperforms its original counterparts significantly in terms of accuracy, convergence speed, and local optima avoidance.

1. Introduction

Optimization refers to the process of finding optimal values for the desired parameters of a given system to maximize or minimize its output. The main advantage of metaheuristic optimization algorithms over conventional optimization algorithms is that they treat optimization problems as a black box. It means that the metaheuristic algorithms change their output only based on the input, and they do not require the derivation of mathematical models of the search landscape. This advantage makes them highly flexible to various type of optimization problems and subsequently become dominant in the field of optimization [1].

Natural, biological, and physical phenomena are familiar sources of inspiration for this category of optimization algorithms. Some of the well-known metaheuristic algorithms include Genetic Algorithm (GA) [2], Particle Swarm Optimization (PSO) [3], Bacterial Foraging Algorithm (BFA) [4], Spiral Dynamic Algorithm (SDA) [5], and Sine Cosine Algorithm (SCA) [6]. Nowadays, BFA which has been inspired by foraging behavior of the E. coli bacteria is gaining attention due to its success in solving real-world optimization problems [7]. At the same time, SCA which has been inspired by mathematical sine and cosine functions also seems to have massive potential like other established metaheuristic algorithms.

In optimization, the term search area refers to all the possible values for the design parameters of a given system. A good optimization algorithm must a right balance between the exploration (to find some promising areas) and the exploitation (to find the optima within those areas). From the optimization perspective, chemotaxis is the most critical mechanism in BFA because the bacteria step size during chemotaxis is the main factor that determines the quality of exploration and exploitation processes. Good exploration is associated with a large step size while good exploitation is associated with a small step size. For that reason, a constant step size in the original BFA creates an imbalance between the exploration and exploitation during optimization, which directly affects the performance of the algorithm. Defining a small step size leads to high accuracy, but many more steps are needed to reach the optima. Consequently, the algorithm will have a slow convergence speed and requires high computational time. In contrast, defining a large step size may expedite the convergence speed, but results in low accuracy, and creates oscillation near the global optimum [8].

Since BFA was introduced in 2002, a significant number of alterations have been proposed to improve its performances mainly through adaptive strategies by varying the bacteria step size during chemotaxis. Adaptive chemotaxis allows BFA to dynamically reduce bacteria step size as the algorithm approaching global optimum hence improving its convergence speed and eliminate oscillation about global optimum. BFA is made adaptive by replacing constant step size in original BFA with an adaptive formula [8]–[11], or by hybridizing BFA with other metaheuristic algorithms that employ adaptive step size in its position updating mechanism. Some of the hybrid BFAs available in the literature including BFA-GA [12], [13], BFA-PSO [7], [14], and BFA-SDA [15], [16].

Motivated by the capabilities and achievements of the adaptive and hybrid BFAs in dealing with real-world optimization problems, this paper proposes a new hybrid algorithm between BFA and SCA referred as Hybrid Bacterial Foraging Sine Cosine Algorithm (HBFSCA). The rest of this paper is organized into four sections; section 2, 3, 4, and 5. Section 2 gives a brief literature review of the original BFA and SCA. Section 3 describes the proposed HBFSCA. Section 4 presents the performances of the proposed algorithm compared to its original counterparts. Finally, section 5 provides a conclusion of the work in this paper.

2. Related Works

This section provides a brief description of the optimization problems utilized in this work, as well as the quick overview of the original versions of BFA and SCA.

2.1. Single Objective Minimization Problems

As mentioned in the previous section, optimization can be defined as the process of finding optimal values for the parameters of a given system to maximize or minimize its output. Considering the single objective minimization type of optimization problems, mathematically, an optimization problem can be written as:

$$\min f\left(\mathbf{X}\right),\ \mathbf{X} \in \Re^{D} \tag{1}$$

where $f\left(\mathbf{X}\right)$ is the function of the design vector

$$\mathbf{X} = \left(X^{1}, X^{2}, \ldots, X^{D}\right)^{T}, \tag{2}$$

and the components X^{d} of $\mathbf{X}$ are called design or decision variables.

2.2. Bacterial Foraging Algorithm

E. coli foraging strategy involves three primary phases namely chemotaxis, reproduction, and elimination-dispersal. All these phases involve systematic procedures that are useful for optimization purposes. Chemotaxis refers to the saltatory movements of the bacteria population while searching for food or nutrient inside a chemical substance. An E. coli bacterium can swim in the same direction for some time on a food gradient, or it can tumble to change its direction in a neutral (no food) or noxious substance, and it alternates between these two types of movement for the whole lifespan. Reproduction refers to the process where healthier bacteria split into two to produce their daughters with the same characteristics as their parents, while the weaker bacteria with not enough food will die. An elimination-dispersal event describes the unpredictable situation where some of the bacteria might be killed or dispersed into a new location by uncontrollable external factors. The details of the original BFA are explained in its original paper [4].

In BFA, N_s, N_c, N_{re}, and N_{ed} indicate the maximum number of swimming-length, maximum chemotaxis steps, maximum reproduction cycle, and maximum elimination-dispersal events, respectively, while C refers to the step size of the bacteria population. Consider there is S number of bacteria in the population while j, k, and l indicate the current chemotactic step, reproduction, and elimination-dispersal, respectively. The position of the i-th bacterium in the D-dimensional search space at the j-th chemotactic step, k-th reproduction, and l-th elimination-dispersal can be expressed as:

$$\mathbf{X}_i(j,k,l) = \left\{X_i^1(j,k,l), X_i^2(j,k,l), \ldots, X_i^D(j,k,l)\right\} \tag{3}$$

During the chemotaxis phase, an E. coli bacterium may first tumble and then followed by a swim or by another tumble. The position of the i-th bacterium after a tumble is given by equation (4):

$$\mathbf{X}_i(j+1,k,l) = \mathbf{X}_i(j,k,l) + C(i)\frac{\boldsymbol{\Delta}_i(j,k,l)}{\sqrt{\boldsymbol{\Delta}_i(j,k,l)^T \boldsymbol{\Delta}_i(j,k,l)}} \tag{4}$$

where $C(i)$ represents the step-size of the i-th bacterium and $\boldsymbol{\Delta}_i(j,k,l)$ is a random vector in D-dimensional that describes the movement direction of the i-th bacterium. If the bacterium finds a higher nutrient concentration, it will continue to swim in the same direction as tumble as given by the equation (5):

$$\mathbf{X}_i(j+1,k,l) = \mathbf{X}_i(j+1,k,l) + C(i)\frac{\boldsymbol{\Delta}_i(j,k,l)}{\sqrt{\boldsymbol{\Delta}_i(j,k,l)^T \boldsymbol{\Delta}_i(j,k,l)}} \tag{5}$$

After the entire lifespan of the bacteria population, as indicated by the total number of chemotactic steps N_c, the bacteria population is sorted according to their health for the reproduction of a new generation. The fittest half of the population with higher nutrient concentration will remain, and a copy of them replace the other half, which has lower nutrient. After a few generations, the elimination-dispersal event will take place: the bacteria are dispersed to a whole new location within the search area. Most of the efforts to improve the performances of BFA in the literature modify the chemotaxis operation [7]–[15].

2.3. Sine Cosine Algorithm

Like other population-based metaheuristic algorithms, SCA starts the optimization by initializes S number of random solutions within the search space. After that, the solutions update their positions by using equation (6)

$$X_i^d(t+1) = \begin{cases} X_i^d(t) + r_1 \times \sin(r_2) \times \left|r_3 P^d(t) - X_i^d(t)\right| & r_4 < 0.5 \\ X_i^d(t) + r_1 \times \cos(r_2) \times \left|r_3 P^d(t) - X_i^d(t)\right| & r_4 \geq 0.5 \end{cases} \tag{6}$$

where $X_i^d(t+1)$, $X_i^d(t)$, and $P^d(t)$ are the next position at iteration $t+1$, the current position at iteration t, and the best solution so far at iteration t in the d-th dimension, respectively. r_1 is an adaptive variable given by the equation (7), r_2 is a random number in [0, 2π], r_3 is a random number in [0, 2], r_4 is a random number in [0, 1], and || indicates the absolute value.

$$r_1 = a - t\frac{a}{T} \tag{7}$$

where a is the maximum step size and T is the maximum number of iteration. The algorithm repeats the position updating operation for the maximum number of iteration T, or until other predefined stopping conditions are satisfied, and then returns the best solution so far at the last iteration as the global optimum. Interested readers can find the details of the original SCA in its original paper [6].

3. Proposed Hybrid Bacterial Foraging Sine Cosine Algorithm (HBFSCA)

Despite the drawbacks produced by the constant step size during chemotaxis by the original BFA mentioned in the previous section, it has a unique structure compared to other metaheuristic algorithms. The reproduction in BFA may expedite the exploitation phase, and the elimination-dispersal event can prevent the algorithm from being trapped in local optima. At the same time, while SCA is relatively a high-speed algorithm because it has a simple structure, our investigation on the SCA revealed that it is suffered from local optima stagnation especially on complex multimodal fitness landscapes. The main idea of the proposed algorithm is to synergize the strengths of both BFA and SCA to overcome the drawbacks in original BFA produced by the constant step size during chemotaxis, in order to produce a new hybrid algorithm with a better performance in terms of accuracy, convergence speed, and local optima avoidance.

The bacteria constant step size of BFA $C(i)$ in equation (4) is made adaptive by using adaptive variable r_1 of SCA in equation (7), as expressed by the equation (8):

$$C(i) = a - a\frac{t(j,k,l)}{(N_c \times N_{re} \times N_{ed})} \tag{8}$$

where a, N_c, N_{re}, and N_{ed} are the maximum step size, maximum chemotactic steps, maximum reproduction cycle, and maximum elimination-dispersal event, respectively. $t(j,k,l)$ is the current iteration counter at the j-th chemotactic step, k-th reproduction, and l-th elimination-dispersal phase. The new adaptive chemotaxis allows the algorithm to reduce the step size of the bacterium linearly from a to 0, with respect to the current iteration $t(j,k,l)$ throughout the optimization process. At the same time, equation (9) which has been adopted from the equation (6) in SCA, generates a highly randomized direction for the bacterium movement during chemotaxis because the algorithm produces different random numbers r_2, r_3, and r_4 for each d-th dimension of the i-th bacterium.

$$\Delta_i^d(j,k,l) = \begin{cases} \sin(r_2) \times \left| r_3 P^d(j,k,l) - X_i^d(j,k,l) \right| & r_4 < 0.5 \\ \cos(r_2) \times \left| r_3 P^d(j,k,l) - X_i^d(j,k,l) \right| & r_4 \geq 0.5 \end{cases} \tag{9}$$

where $X_i^d(j,k,l)$ and $P^d(j,k,l)$ are the current position of the i-th bacterium and the best solution so far at the j-th chemotactic step, k-th reproduction, and l-th elimination-dispersal in the d-th dimension, respectively. Similar to those in SCA, r_2 is a random number in [0, 2π], r_3 is a random number in [0, 2], r_4 is a random number in [0, 1], and || indicates the absolute value. The new direction vector of the i-th bacterium at the j-th chemotactic step, k-th reproduction, and l-th elimination-dispersal is expressed as:

$$\mathbf{\Delta}_i(j,k,l) = \left\{ \Delta_i^1(j,k,l), \Delta_i^2(j,k,l), \ldots, \Delta_i^D(j,k,l) \right\} \tag{10}$$

where D is the total dimension of the search area. The position of the i-th bacterium after a tumble is given by equation (11):

$$\mathbf{X}_i(j+1,k,l) = \mathbf{X}_i(j,k,l) + \left(a - a\frac{t(j,k,l)}{(N_c \times N_{re} \times N_{ed})} \right) \left(\frac{\mathbf{\Delta}_i(j,k,l)}{\sqrt{\mathbf{\Delta}_i(j,k,l)^T \mathbf{\Delta}_i(j,k,l)}} \right) \tag{11}$$

If the bacterium finds a higher nutrient concentration, it will continue to swim in the same direction as the tumble given by the equation (12) for the maximum number of swim-length, as long as it swims up the concentration gradient.

$$\mathbf{X}_i(j+1,k,l) = \mathbf{X}_i(j+1,k,l) + \left(a - a\frac{t(j,k,l)}{(N_c \times N_{re} \times N_{ed})} \right) \left(\frac{\mathbf{\Delta}_i(j,k,l)}{\sqrt{\mathbf{\Delta}_i(j,k,l)^T \mathbf{\Delta}_i(j,k,l)}} \right) \tag{12}$$

In contrast, if the bacterium happens to swim down a concentration gradient, it will tumble to another random direction to find a better nutrient concentration. The rest of the proposed algorithm is similar to those described in the original BFA including the reproduction and elimination-dispersal events.

```
 1:   Initialize the population of S number of bacteria into random positions
 2:   Initialize interation counter t=0
 3:   For each elimination-dispersal event
 4:       For each reproduction event
 5:           For each chemotaxis cycle
 6:               For each bacterium
 7:                   Evaluate the nutrient level at the initial position
 8:                   Update iteration counter t
 9:                   Update C(i) using equation (8)
10:                   For each dimension
11:                       Update r2, r3 and r4
12:                       Generate a random direction using equation (9)
13:                   End for
14:                   Update the bacterium's position using equation (11)
15:                   Evaluate the nutrient level at the new position
16:                   If the bacterium finds a better nutrient concentration
17:                       If the maximum swimming length Ns is not yet satisfied
18:                          Update iteration counter t
19:                          Update C(i) using equation (8)
20:                          Update the bacterium's position using equation (12)
21:                          Evaluate the nutrient level at the latest position
22:                          If the bacterium finds a better nutrient level
23:                             Go to step 17
24:                          Else
25:                             Go to the next bacterium
26:                          End if
27:                       End if
28:                   End if
29:               End for
30:           End for
31:           Split the healthier half of the population into two
32:       End for
33:       Reinitialize the position of the whole bacteria population randomly
34:   End for
35:   Return the bacterium with the highest nutrition level as the global optimum
```

Figure 1. Pseudocode of the proposed HBFSCA algorithm

4. Results and Discussion

Researchers in metaheuristics algorithms usually test their algorithms to several test cases called benchmark functions. It is performed to observe and measure the accuracy, convergence speed, and repeatability of their algorithms. Besides, it is also conducted to ensure that their algorithms can deal with various applications and not a specific problem-dependent. The proposed HBFSCA is implemented to 30 single-objective minimization problems from CEC 2014's test suite [17], consist of four categories including unimodal, multimodal, composite, and hybrid functions.

The performances of the proposed HBFSCA is compared with its original counterparts: original BFA and SCA. Table 1 presents the parameters of the algorithms used in the test. All the algorithms are subject to 100 dimensions, 100 search agents, and 10000000 number of function evaluations (FES). Since the results from a single run are not reliable in metaheuristic algorithms due to their stochastic characteristics, statistical results (mean) over 51 runs are gathered. The performances of all three algorithms were ranked by using a non-parametric test called Friedman test. Another statistical test named Wilcoxon Signed Rank test is also conducted to analyze the significance of the ranks provided by the Friedman test. All the statistical results, mean rank from the Friedman test, and p values from the Wilcoxon Signed Rank test are reported in Table 2.

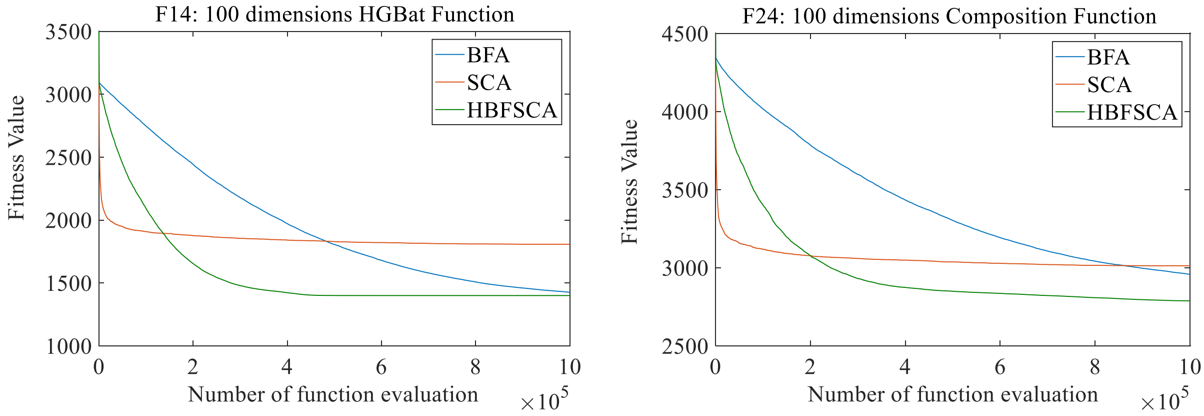
Table 1. Parameters of SCA, BFA, and HBFSCA

Parameter	SCA	BFA	HBFSCA
C		1.7	1.7
N_s		4	4
N_c		500	500
N_{re}		5	5
N_{ed}		3	3
P_{ed}		0.25	0.25
a	0.8		5
T	10000		

Table 2. Statistical results, Friedman test, and Wilcoxon Signed Rank test

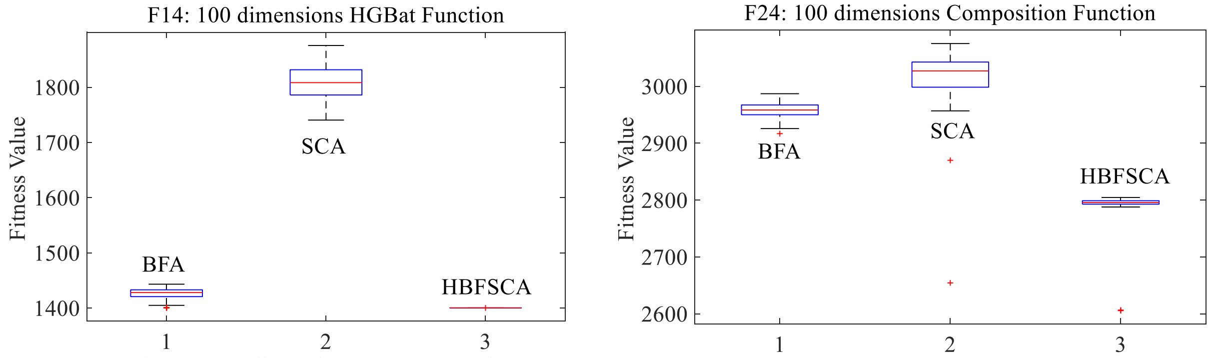
	Statistical results			Friedman test			Wilcoxon Signed Rank test			
No.		Mean			Mean rank		BFA vs HBFSCA		SCA vs HBFSCA	
	BFA	SCA	HBFSCA	BFA	SCA	HBFSCA	Sign rank	p	Sign rank	p
1	2.2130E+08	2.1771E+09	**7.7015E+07**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
2	1.0384E+10	1.5334E+11	**4.8378E+08**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
3	5.5175E+05	**2.0966E+05**	3.8678E+05	3.00	**1.00**	2.00	1326	**0.00**	0	**0.00**
4	1.6661E+03	2.4914E+04	**8.2212E+02**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
5	5.2132E+02	5.2131E+02	**5.2106E+02**	2.57	2.43	**1.00**	1326	**0.00**	1326	**0.00**
6	7.3654E+02	7.4164E+02	**7.2248E+02**	2.12	2.88	**1.00**	1326	**0.00**	1326	**0.00**
7	8.1106E+02	2.1827E+03	**7.0692E+02**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
8	2.0557E+03	1.9963E+03	**1.4370E+03**	2.80	2.20	**1.00**	1326	**0.00**	1326	**0.00**
9	2.4516E+03	2.1963E+03	**1.6401E+03**	3.00	2.00	**1.00**	1326	**0.00**	1326	**0.00**
10	2.0487E+04	2.8235E+04	**1.4043E+04**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
11	2.2747E+04	3.0798E+04	**1.5398E+04**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
12	1.2034E+03	1.2040E+03	**1.2026E+03**	2.02	2.96	**1.02**	1325	**0.00**	1326	**0.00**
13	*1.3005E+03*	1.3063E+03	*1.3005E+03*	1.67	3.00	**1.33**	991	**0.00**	1326	**0.00**
14	1.4258E+03	1.8079E+03	**1.4003E+03**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
15	3.5495E+03	1.3583E+06	**1.6500E+03**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
16	1.6464E+03	1.6462E+03	**1.6449E+03**	2.69	2.31	**1.00**	1326	**0.00**	1326	**0.00**
17	1.0532E+07	1.9816E+08	**3.2902E+06**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
18	3.2028E+07	5.3398E+09	**7.9535E+05**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
19	2.0713E+03	3.0751E+03	**2.0159E+03**	1.98	3.00	**1.02**	1325	**0.00**	1326	**0.00**
20	4.2577E+05	**1.1504E+05**	2.1344E+05	3.00	**1.02**	1.98	1326	**0.00**	7	**0.00**
21	4.5319E+06	6.5644E+07	**1.9866E+06**	1.98	3.00	**1.02**	1322	**0.00**	1326	**0.00**
22	4.2245E+03	7.5511E+03	**3.4040E+03**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
23	2.8017E+03	3.5222E+03	**2.6899E+03**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
24	2.9581E+03	3.0131E+03	**2.7886E+03**	2.06	2.92	**1.02**	1326	**0.00**	1324	**0.00**
25	2.8033E+03	2.7968E+03	**2.7658E+03**	2.55	2.00	**1.45**	1326	**0.00**	913	**0.02**
26	2.8093E+03	2.8461E+03	**2.7985E+03**	1.98	2.98	**1.04**	1276	**0.00**	1326	**0.00**
27	**4.5183E+03**	6.9287E+03	5.1818E+03	**1.51**	2.94	1.55	495	0.12	1326	**0.00**
28	1.9965E+04	2.5751E+04	**1.4487E+04**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
29	3.2464E+06	1.3173E+09	**5.8011E+04**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**
30	8.4727E+05	1.9479E+07	**2.4186E+05**	2.00	3.00	**1.00**	1326	**0.00**	1326	**0.00**

The statistical mean values in Table 2 indicate that the proposed HBFSCA has higher accuracy compared to the original BFA and SCA in all test cases except that for the F3, F13, F20, and F27. For F13, the proposed HBFSCA and BFA perform equally, and both of them outperform SCA. The Friedman test ranks the algorithms from the best to the worst represent by the lowest to the highest rank values. A p-value less than 0.05 in the Wilcoxon Signed Rank test indicates that the difference between the two algorithms tested is significant (5% significant level). The statistical mean values in Table 2 are consistent with the outcomes from the statistical analysis using Friedman and Wilcoxon Signed Rank tests. Figure 2 presents the convergence curves of two selected cases: F14 and F24. It can be observed that the proposed HBFSCA, improves the convergence speed of BFA rapidly, and successfully avoids local optima stagnation as usually occurred to the SCA. Figure 3 shows the Boxplot of the same benchmark functions: F14 and F24. The Boxplot verifies that the proposed algorithm has higher accuracy and repeatability compared to the original BFA and SCA.



(a) F14: 100 dimensions HGBat Function (b) F24: 100 dimensions Composition Function

Figure 2. Convergence curves of the selected functions



(a) F14: 100 dimensions HGBat Function (b) F24: 100 dimensions Composition Function

Figure 3. Box plot of the selected functions

5. Conclusion

This work was devoted to develop a new hybrid metaheuristic optimization algorithm between SCA and BFA and measure its performances against its original counterparts. Generally, the proposed hybridization scheme between BFA and SCA improved the accuracy of BFA significantly. The new proposed HBFSCA has successfully overcome the drawbacks and preserved the strengths of both its original counterparts. Future work would attempt to involve the application of the proposed algorithm to real-world applications in comparison with other types of hybrid BFAs.

Acknowledgments

This work has been supported by the Postgraduate Research Grant of the Universiti Malaysia Pahang (UMP) with the grant reference number PGRS1903145. It has been awarded by the Research and Innovation Department, Universiti Malaysia Pahang, Malaysia.

References

[1] S. Droste, T. Jansen, and I. Wegener, "Upper and Lower Bounds for Randomized Search Heuristics in Black-Box Optimization," *Theory of Computing Systems*, vol. 39. pp. 525–544, 2006.

[2] J. H. Holland and J. S. Reitman, "Cognitive Systems Based on Adaptive Algorithms," in *Pattern-Directed Inference Systems*, Academic Press, 1978, pp. 313–329.

[3] R. Eberhart and James Kennedy, "A New Optimizer Using Particle Swarm Theory," in *Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[4] K. M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," *IEEE Control Syst.*, vol. 22, no. 3, pp. 52–67, 2002.

[5] K. Tamura and K. Yasuda, "Primary Study of Spiral Dynamics Inspired Optimization," *IEEJ Trans. Electr. Electron. Eng.*, vol. 6, no. S1, pp. 2010–2012, 2011.

[6] S. Mirjalili, "SCA: A Sine Cosine Algorithm for Solving Optimization Problems," *Knowledge-Based Syst.*, vol. 96, pp. 120–133, 2016.

[7] S. M. Abd-Elazim and E. S. Ali, "A Hybrid Particle Swarm Optimization and Bacterial Foraging for Optimal Power System Stabilizers Design," *Int. J. Electr. Power Energy Syst.*, vol. 46, no. 1, pp. 334–341, 2013.

[8] S. Dasgupta, S. Das, A. Abraham, and A. Biswas, "Adaptive Computational Chemotaxis in Bacterial," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 919–941, 2009.

[9] Y. Chen and W. Lin, "An Improved Bacterial Foraging Optimization," in *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 2057–2062.

[10] H. Supriyono, "Novel Bacterial Foraging Optimisation Algorithms with Application to Modelling and Control of Flexible Manipulator Systems," The University of Sheffield, 2012.

[11] H. Supriyono and M. O. Tokhi, "Bacterial Foraging Algorithm with Adaptable Chemotactic Step Size," *Proc. - 2nd Int. Conf. Comput. Intell. Commun. Syst. Networks, CICSyN 2010*, pp. 72–77, 2010.

[12] D. H. Kim, "Hybrid GA-BF Based Intelligent PID Controller Tuning for AVR System," *Appl. Soft Comput. J.*, vol. 11, no. 1, pp. 11–22, 2011.

[13] N. A. Okaeme and P. Zanchetta, "Hybrid Bacterial Foraging Optimization Strategy for Automated Experimental Control Design in Electrical Drives," *IEEE Trans. Ind. Informatics*, vol. 9, no. 2, pp. 668–678, 2013.

[14] S. V. R. S. Gollapudi, S. S. Pattnaik, O. P. Bajpai, S. Devi, and K. M. Bakwad, "Velocity Modulated Bacterial Foraging Optimization Technique (VMBFO)," *Appl. Soft Comput. J.*, vol. 11, no. 1, pp. 154–165, 2011.

[15] A. N. K. Nasir, M. O. Tokhi, and N. M. A. Ghani, "Novel Hybrid Bacterial Foraging and Spiral Dynamics Algorithms," in *2013 13th UK Workshop on Computational Intelligence (UKCI)*, 2013, pp. 199–205.

[16] A. N. K. Nasir and M. O. Tokhi, "Novel Metaheuristic Hybrid Spiral-Dynamic Bacteria-Chemotaxis Algorithms for Global Optimisation," *Appl. Soft Comput. J.*, vol. 27, pp. 357–375, 2015.

[17] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," 2013.