International Journal of Innovative
Computing, Information and Control
Volume **13**, Number **5**, October **2017**

 ISSN 1349-4198
pp. **1649–1667**

NOVEL ADAPTIVE QUANTUM-INSPIRED BACTERIAL FORAGING ALGORITHMS FOR GLOBAL OPTIMIZATION

Sharina Huang[1,2], Guoliang Zhao[1,*] and Minghao Chen[2]

[1]School of Science
Heilongjiang University of Science and Technology
No. 2468, Puyuan Road, Songbei District, Harbin 150022, P. R. China
hsrn1982@163.com; *Corresponding author: ocnzhao@163.com

[2]School of Science
Harbin Institute of Technology
No. 92, Xidazhi Street, Nangang District, Harbin 150001, P. R. China
chenmh130264@aliyun.com

Received January 2017; revised June 2017

Abstract. *In this paper, novel bacterial foraging algorithms (BFA) based on quantum principle, adaptive chemotactic step size and spiral dynamics algorithm (SDA) are presented. Then, an adaptive chemotactic step size scheme based on iteration number is obtained via analytical approach. Furthermore, the adaptive chemotactic step size based on individual bacterium fitness value and the current iteration number is proposed. The chemotactic step size schemes can be more dynamically varied and hence better exploration and exploitation strategies are introduced. Swarming mechanism is removed from BFA and the role is played by SDA. With the combination of different strategies, two versions of quantum-inspired bacterial foraging algorithm (QBFA) are proposed. The performance of the proposed algorithms is tested with seven basic benchmark functions and seven CEC05 test functions, and compared with original QBFA and two other adaptive bacterial foraging algorithms. Based on the experiment results, two tailed t-test, non-parametric Wilcoxon signed rank test and Friedman test are used to check the significant difference in the performance of the algorithms. The results show that the proposed algorithms outperform the reference algorithms in terms of accuracy and convergence speed.*
Keywords: Quantum-inspired bacterial foraging algorithm, Chemotactic step size, Spiral dynamics algorithm, Non-parametric Wilcoxon signed rank test, Friedman test

1. **Introduction.** Bacterial foraging algorithm (BFA) originally proposed by Passino [1] is modelled as a distributed optimization process which mimics the swarming and social behavior found in individual and group behavior of E.Coli bacteria. Since its inception, BFA has been widely used in both theoretical study and practical applications, such as power system [2, 3], circuit design [4], signal processing [5], image registration [6], nonlinear dynamic modelling [7], vehicle routing [8], load forecasting [9], energy auditing and management [10], optimization problems [11, 12, 13, 14, 15, 16, 17, 18, 19], feature selection [20], and Bayesian network structural learning [21].

Chemotaxis is crucial to BFA, which mimics a bacterium takes step to reach regions with high-nutrient content [22]. The constant step size in the original BFA influences the convergence speed of the algorithm and the accuracy of the final solution. If a smaller step size is employed, an accuracy solution can be obtained with a very low speed. On the contrary, a bacterium may approach optimum location with a larger step size, but the accuracy of the final solution is reduced. Thus, many adaptive chemotactic step size

strategies are proposed to BFA, such as Gaussian distribution based bacterial foraging strategy [23], and congestion management based bacterial foraging strategy [24].

Mishra [5] presented a hybrid least square-fuzzy bacterial foraging strategy to estimate voltage/current waveforms' harmonic components that appeared in power system, which was carried out by a Takagi-Sugeno fuzzy inference scheme. However, the resulting algorithm may be too specific to solve other general benchmark function optimization problems. An artificial bacterium's chemotactic movements modelled in continuous time form is derived in [22], the stability and convergence of the mathematical model are studied by some Lyapunov stability theorems. An adaptive efficient forecasting BFA model for stock market indices prediction is introduced in [13], a simple linear form step size varying within $[0, 1]$ based on fitness value at every iteration is adopted for the technique, and experiment results show that the new models outperform GA and PSO based evolutionary computing models on the metrics as computational efficient, accuracy and convergence speed. Dasgupta provided a mathematical model which uses two simple health value based schemes for the computational chemotactic step size in BFA. Based on the variable chemotactic step size [25], Xu and Chen [26] designed a more complex scheme to accelerate the convergence speed. In [27], BFA is improved from three aspects: adaptive chemotaxis step size, cell-to-cell communication and dynamic population, and seven versions of BFA combined by different strategies were proposed. In [28], the constrained optimization problem is solved by BFA with linear/nonlinear decreasing chemotactic step size. In the two algorithms, the chemotactic step size varies from a high value $C_{\max}$ linearly or nonlinearly to $C_{\min}$ at the maximal number of iterations. A variant of BFA with exponential based chemotactic step size is proposed in [8], and the algorithm is applied on vehicle routing problem. Nasir and Tokhi [7] presented a novel adaptive BFA variant for global optimization. The chemotactic step sizes varied dynamically based on three distinct strategies: (1) index of iteration; (2) index of bacteria; (3) the combination of all indexes and fitness value.

Spiral dynamics algorithm (SDA) [29] is inspired from spiral phenomena, such as the spiral of waves and a galaxy. In the literature, there are several techniques proposed by researchers to improve the performance of SDA. A hybrid bacteria-chemotaxis spiral-dynamic algorithm where a BFA is combined with SDA with different dynamic behaviours is presented [30], and the algorithm provides a more comprehensive study complexity. Nasir et al. [31] have proposed an improved version of SDA metaheuristic algorithm, and the algorithm is verified by a twin rotor nonparametric fuzzy logic modelling problem. Unique exploration and exploitation strategies of a SDA is achieved via a spiral model which produces the concrete spiral motion and dynamic step size.

Inspired by the different types of chemotactic step size proposed in the literature, we found almost all of the schemes have the property that a larger step size was adopted in the early stage of searching and a smaller step size was adopted in the final stage or the bacteria near the optima, and the step size was changed dynamically according to the status of the bacterium, such as fitness value or the iteration number. However, all of the above mentioned step size schemes are represented as a specified function by the authors, and the shape of function seems empirically. To solve this problem, we analyze the behaviors of bacteria near the optima firstly and establish a differential equation, and the chemotactic step size based on iteration number is obtained by solving the differential equation. Furthermore, another novel adaptive chemotactic step size based on iteration number and bacterium fitness value is proposed. Bacteria move with a large step size at the early stage or the exploration phase, a small step size at the later stage of the search operation or exploitation stage when the bacteria move around an optimum point. Swarming in original BFA is a cell-to-cell communication mechanism, but the mechanism

affects the real health status of the bacterium; thus it was discarded by many researchers [8, 27, 28, 32]. In our work, swarming step is removed and the role is played by SDA. In the framework of quantum-inspired bacterial foraging algorithm (QBFA), the proposed two novel adaptive chemotactic step size schemes and SDA are adopted. Two versions of adaptive QBFA are proposed. The proposed algorithms are validated with seven basic benchmark functions and seven CEC05 test functions (including shifted benchmark version), and compared with QBFA and two other adaptive bacterial foraging algorithms. Moreover, based on the experiment results, non-parametric Wilcoxon signed rank test and Friedman test are employed to check the significant difference of the algorithms.

The rest of the paper is organized as follows. A brief outline of the classical BFA is provided in Section 2, and two recently proposed chemotactic step size schemes are described explicitly. In Section 3, the proposed adaptive QBFA will be introduced; quantum principle, two new chemotactic step size schemes, and SDA are also presented. With the combination of different strategies described in Section 3, two versions of QBFA are presented, and comparison analyses of the proposed algorithms with its predecessor algorithms on two classes of benchmark functions are provided in Section 4. Finally, conclusions are drawn in Section 5.

2. **Adaptive Chemotactic Step Size Model.**

2.1. **Model of BFA.** BFA includes four phases: chemotaxis, swarming, reproduction, and elimination-dispersal. In one chemotactic step, the movement of the ith bacterium from jth step to $(j+1)$th step can be formulated 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 $\theta^i(j,k,l)$ represents ith bacterium at jth chemotactic step, kth reproductive step and lth elimination-dispersal step. $C(i)$ is the step size of the ith bacterium taken in random direction, and Δ indicates a unit length vector in the random direction. As each bacterium moves, attractant or repellent is released to signal other bacteria. BFA simulates this social behaviors as follows

$$\begin{aligned} J_{cc}(\theta^i,\theta) = \sum_{t=1}^{N_b} J_{cc}^t(\theta^i,\theta) = & \sum_{t=1}^{N_b}\left[-d_{atract}\exp\left(-\omega_{atract}\sum_{m=1}^{N}\left(\theta_m^i-\theta_m^t\right)^2\right)\right] \\ & + \sum_{t=1}^{N_b}\left[-d_{repell}\exp\left(-\omega_{repell}\sum_{m=1}^{N}\left(\theta_m^i-\theta_m^t\right)^2\right)\right], \end{aligned}$$

where $J_{cc}(\theta^i,\theta)$ is the value to be added to the actual objective function to represent a time varying objective function, N_b is the total number of bacteria, N is the dimension of search space, and d_{atract}, ω_{atract}, d_{repell}, ω_{repell} are coefficients that should be chosen properly. After swarming phase, bacteria are classified and selected based on their health and fitness level. It is needed to point out that the population size should be even in this situation. However, it is not very difficult to extend to odd situation. Half bacteria with the lower health and fitness value are eliminated and reborn randomly. To make the foraging speed more faster, an elimination-dispersal phase is introduced to extend the exploitation ability of the BFA. With this strategy, the probability of the bacteria relocated at closer position to the optimum nutrient or food location is prone to increase. The four processes perform in a sequential order and loop continuously until full life cycle of bacteria is reached or maximum iteration steps are met.

2.2. **Adaptive chemotactic step size model of BFA.** Recently, Nasir and Tokhi [7] proposed two adaptive bacterial foraging algorithms, named IBFA and FIBFA. The two chemotactic step size schemes adopted in the two algorithms are presented as follows.

$$C_{IBFA} = \frac{1.5}{0.9 \times Iter^{0.9} + 1}, \tag{2}$$

$$C_{FIBFA} = \frac{2.5}{\dfrac{0.9 \times Iter^{0.9}}{0.2 \times |J(i,j,k,l)|^{0.2}} + 1}, \tag{3}$$

where $Iter = i \times j \times k \times l$ is the product accumulated total iteration of i, j, k and l, while i, j, k and l are index of total number of bacteria in the population, indices of total number of chemotaxis, reproduction and elimination-dispersal, respectively. $J(i,j,k,l)$ is the current bacterium fitness value. The chemotactic step size C_{IBFA} in IBFA is a scheme based on the iteration index, while C_{FIBFA} in FIBFA is a scheme based on the combination of iteration index and individual bacterium fitness value. The two algorithms present good performance compared with the original BFA. Thus, IBFA and FIBFA will be specified as two benchmark algorithms compared with the proposed algorithms in this paper.

3. **Adaptive Quantum-Inspired Bacterial Foraging Algorithm (AQBFA).** In this section, quantum principle and QBFA [32] are introduced first. Then, two novel adaptive chemotactic step sizes are proposed and SDA is described. Finally, adaptive quantum-inspired bacterial foraging algorithm (AQBFA) is proposed and the flowchart of AQBFA is depicted.

3.1. **Quantum principle.** Quantum principle is a mechanism which is used to decode and observe the population in this paper. The basic element in quantum principle is called a quantum bit or Q-bit. In mathematics, a Q-bit is represented by a unit vector in the two-dimensional complex Hilbert space, and can be written in the Dirac notation as follows:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{4}$$

where $|0\rangle$ and $|1\rangle$ are two computational basis states of Q-bit, α and β are called probability amplitudes of the state $|\phi\rangle$ and $\alpha^2 + \beta^2 = 1$. The Q-bit defined as a unit complex vector $(\alpha, \beta)^T$ is the smallest unit of information in quantum principle, $|\alpha|^2$ and $|\beta|^2$ give the probability that the Q-bit will be found in the '0' and '1' states, respectively. Furthermore, a Q-bit position in N-dimensional space can be represented as a string of quantum bits. In BFA, the Q-bit position of the ith bacterium at the jth chemotactic step kth reproduction step and lth elimination-dispersal step is defined as

$$\begin{bmatrix} \alpha_{i1}(j,k,l) & \alpha_{i2}(j,k,l) & \cdots & \alpha_{iN}(j,k,l) \\ \beta_{i1}(j,k,l) & \beta_{i2}(j,k,l) & \cdots & \beta_{iN}(j,k,l) \end{bmatrix}, \tag{5}$$

where $|\alpha_{in}(j,k,l)|^2 + |\beta_{in}(j,k,l)|^2 = 1$, $n = 1, 2, \ldots, N$, and N denotes the dimension of the search space. The Q-bit position can represent 2^N states simultaneously. An example with three-Q-bit is represented as

$$\begin{bmatrix} \dfrac{\sqrt{3}}{2} & \dfrac{1}{3} & \dfrac{\sqrt{2}}{2} \\ \dfrac{1}{2} & \dfrac{2\sqrt{2}}{3} & -\dfrac{\sqrt{2}}{2} \end{bmatrix}.$$

The state can be described as

$$\frac{1}{24}|000\rangle - \frac{1}{24}|001\rangle + \frac{1}{3}|010\rangle - \frac{1}{3}|011\rangle + \frac{1}{72}|100\rangle - \frac{1}{72}|101\rangle + \frac{1}{9}|110\rangle - \frac{1}{9}|111\rangle. \quad (6)$$

This Q-bit representation has the advantage that it is able to represent a linear superposition of states probabilistically which has eight states and each state has a corresponding probability.

In quantum algorithm, quantum gate is used to explore the search space. The new Q-bit position is obtained through quantum rotation gate. The operation used in this paper can be presented as follows.

$$\begin{aligned}\begin{bmatrix} \alpha'_{in} \\ \beta'_{in} \end{bmatrix} &= \begin{bmatrix} \cos(\Delta\theta_{in}) & -\sin(\Delta\theta_{in}) \\ \sin(\Delta\theta_{in}) & \cos(\Delta\theta_{in}) \end{bmatrix} \begin{bmatrix} \alpha_{in} \\ \beta_{in} \end{bmatrix} \\ &= U(\Delta\theta_{in}) \begin{bmatrix} \alpha_{in} \\ \beta_{in} \end{bmatrix}, \end{aligned} \quad (7)$$

where $[\alpha_{in}, \beta_{in}]^T$ represents the current Q-bit position, $U(\Delta\theta_{in})$ denotes a quantum gate, $\Delta\theta_{in}$ is a rotation angle, and $[\alpha'_{in}, \beta'_{in}]^T$ is the new Q-bit position which is obtained by rotation gate operation. In this paper, $\Delta\theta_{in}$ is determined by the current position of ith bacterium and the best position of the population.

In our method, the initial Q-bit position of the population is denoted as follows:

$$P_Q(t) = \{P_{Q_1}(t), P_{Q_2}(t), \ldots, P_{Q_{Nb}}(t)\}^T, \quad (8)$$

where $P_{Q_i}(t)$ is defined as

$$P_{Q_i}(t) = \begin{bmatrix} \cos(\theta_{i1}) & \cos(\theta_{i2}) & \cdots & \cos(\theta_{iN}) \\ \sin(\theta_{i1}) & \sin(\theta_{i2}) & \cdots & \sin(\theta_{iN}) \end{bmatrix}, \quad (9)$$

where $\theta_{in} \in [0, \pi/2]$, $n = 1, 2, \ldots, N$, $i = 1, 2, \ldots, N_b$, and N_b denotes the number of population. Then, the population can be described as follows:

$$P_P(t) = \{P_{P_1}(t), P_{P_2}(t), \ldots, P_{P_{Nb}}(t)\}^T, \quad (10)$$

where $P_{P_i}(t)$ is defined as follows

$$P_{P_i}(t) = \begin{bmatrix} \cos^2(\theta_{i1}) \\ \cos^2(\theta_{i2}) \\ \vdots \\ \cos^2(\theta_{iN}) \end{bmatrix}^T \otimes UB^T + \begin{bmatrix} \sin^2(\theta_{i1}) \\ \sin^2(\theta_{i2}) \\ \vdots \\ \sin^2(\theta_{iN}) \end{bmatrix}^T \otimes LB^T,$$

where $\otimes$ denotes the Hadamard product of two vectors, LB and UB denote the lower and upper bounds of the decision variable, respectively. This representation can guarantee the feasibility and diversity of the population.

3.2. **Quantum-inspired BFA (QBFA).** In what follows we give the step-by-step descriptions of QBFA [32].

[step 0] Initialize parameters N, N_b, $\theta_{i\cdot}$, N_c, N_s, N_{re}, N_{ed}, P_{ed}, and C_{QBFA}. The meanings of the symbols are listed in Table 1.

[step 1] Make an observation of the Q-bit position $P_Q(t)$, and then obtain the initial population $P_P(t)$.

[step 2] Elimination-dispersal loop: For $l = 1, \ldots, N_{ed}$.

[step 3] Reproduction loop: For $k = 1, \ldots, N_{re}$.

[step 4] Chemotaxis loop: For $j = 1, \ldots, N_c$.

(a) For ith bacterium ($i = 1$ initially), update the quantum gate and Q-bit individuals are updated clockwise or anti-clockwise according to the current situation.

TABLE 1. Parameters of QBFA

Vars	Explanation
N	Dimension of the search space
N_b	Total number of bacteria in the population
$\theta_{i.}$	Rotation angle during the life span when bacterium moves to optimal solution
N_c	The number of chemotaxis steps
N_s	Swimming length
N_{re}	The number of reproduction steps
N_{ed}	The number of elimination-dispersal steps
P_{ed}	Elimination-dispersal events
C_{QBFA}	The chemotactic step size used in QBFA

(b) Compute fitness value $J(i, j, k, l)$, and save the fitness value as J_{last}.

(c) Compute chemotactic step size C_{QBFA} and generate a new position as follows

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C_{QBFA} \times \left(\theta^b(j,k,l) - \theta^i(j,k,l)\right), \tag{11}$$

where $\theta^i(j, k, l)$ is the position of ith bacterium at jth chemotaxis step, kth reproduction step and lth elimination-dispersal step, $\theta^b(j, k, l)$ is the global best of the population found so far.

(d) Swim.

i) Set $m = 0$ and calculate $J(i, j + 1, k, l)$.

ii) If $m < N_s$ and $J(i, j + 1, k, l) < J_{last}$, let $J_{last} = J(i, j + 1, k, l)$ and generate a new position along the previous direction as follows

$$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + C_{QBFA} \times \left(\theta^b(j,k,l) - \theta^i(j,k,l)\right). \tag{12}$$

Let $m = m + 1$, and go to ii).

iii) Else, if $J_{last} < J_{best}$, update the J_{best} and P_{best}, where J_{best} is the best fitness value the population found so far, and P_{best} is the corresponding position.

(e) If $i < N_b$, let $i = i + 1$, and go to (a).

[step 5] If $j < N_c$, let $j = j + 1$, and go to step 4.

[step 6] Reproduction.

(a) Let J^i_{health} be the health status of the ith bacterium for given k and l, compute the health status for the ith bacterium ($i = 1, 2, \ldots, N_b$). Sort bacteria in order of ascending health status J_{health} (lower cost means higher health).

(b) The S_r bacteria with better fitness value survive and have the chance to generate the next generation.

[step 7] If $k < N_{re}$, let $k = k + 1$, and go to step 3.

[step 8] Elimination-dispersal: Eliminate and disperse the bacteria for $i = 1, 2, \ldots, N_b$ with probability P_{ed}. That is, if a bacterium is eliminated, simply generate a location on the optimization domain with the quantum operation. If $l < N_{ed}$, let $l = l + 1$ and the algorithm will go to step 2; otherwise, the algorithm will be terminated.

In QBFA, the chemotactic step size adopts the scheme proposed by [25] and

$$C_{QBFA} = \frac{|J(i,j,k,l)|}{|J(i,j,k,l)| + \lambda}, \tag{13}$$

where λ is a positive constant and $\lambda = 4000$ is a suitable parameter.

3.3. **The iteration-based adaptive chemotactic step size design.** Inspired by the law that chemotactic step size in the early chemotaxis phase should be larger than the

chemotactic step size at final stage of chemotaxis phase, a unified framework of adaptive step size is proposed in this section. First, the iteration-based adaptive chemotactic step size function design method is introduced. Consider a function $c : X = [1, +\infty) \to (0, 1)$, where X is the interval of all iterations. We will define the chemotactic step size as

$$C_{IAQBFA} = \frac{c_{\min}}{c(x)}, \tag{14}$$

where $c_{\min}$ is the minimum step size in the algorithm. Let $\lim_{x\to+\infty} c(x) = 1$, Δc is proportional to Δx for the same x. Obviously, Δc should be small if $c \approx 1$. It would be $(1 - c)^\alpha$ if a new order α is introduced to scale the item $(1 - c)$. Then, we will have the following difference equation:

$$\Delta c = k_1(1 - c)^\alpha \, \Delta x/x, \quad \alpha \in \mathbb{R}^+, \tag{15}$$

where k_1 is a positive constant number. Let $\Delta x \to 0$, it holds that

$$\frac{dc}{dx} = k_1(1 - c)^\alpha/x. \tag{16}$$

We can easily solve (16) and have

$$c(x) = \begin{cases} 1 - [c_1 - k_1(1 - \alpha)\ln x]^{\frac{1}{1-\alpha}}, & \alpha \neq 1, \\ 1 - \dfrac{c_1}{x^{k_1}}, & \alpha = 1. \end{cases} \tag{17}$$

With initial condition $c(1) = c_{\min}$ $(0 < c_{\min} \ll 1)$, we can get

$$c_1 = \begin{cases} (1 - c_{\min})^{1-\alpha}, & \alpha \neq 1, \\ 1 - c_{\min}, & \alpha = 1. \end{cases} \tag{18}$$

Then Equation (17) has the following form

$$c(x) = \begin{cases} 1 - [(1 - c_{\min})^{1-\alpha} - k_1(1 - \alpha)\ln x]^{\frac{1}{1-\alpha}}, & \alpha \neq 1, \\ 1 - \dfrac{1 - c_{\min}}{x^{k_1}}, & \alpha = 1, \end{cases} \tag{19}$$

where variable x is assigned to iteration number $Iter = i \times j \times k \times l$. Note that $c(x)$ is a nonlinear increasing function with respect to x, $c(1) = c_{\min}$ and $\lim_{x\to+\infty} c(x) = 1$. Then the chemotactic step size C_{IAQBFA} defined as (14) is a nonlinear decreasing function with respect to x, when $x = 1$, $C_{IAQBFA} = 1$, and $\lim_{x\to+\infty} C_{IAQBFA} = c_{\min}$. It means that the chemotactic step size in the first iteration will be 1, and chemotactic step size approximates $c_{\min}$ as the iteration steps increase. The chemotactic step size in the algorithm will be defined as

$$C_{IAQBFA} = \frac{c_{\min}}{1 - \dfrac{1 - c_{\min}}{x^{k_1}}}, \tag{20}$$

when $\alpha = 1$, this is similar to chemotactic step size (1) which is originally defined in [7]. When $\alpha > 1$, the chemotactic step size in the algorithm will be defined as

$$C_{IAQBFA} = \frac{c_{\min}}{1 - [(1 - c_{\min})^{1-\alpha} - k_1(1 - \alpha)\ln x]^{\frac{1}{1-\alpha}}}. \tag{21}$$

Figure 1 depicts the adaptive chemotactic step size function C_{IAQBFA} with $\alpha = 2$, $c_{\min} =$ 6E-3, $k_1 =$ 1E-2, and chemotactic step size (2). From the figure we can see that the shapes of the two curves are similar and the decreasing speed of the two chemotactic step sizes in the early iteration stage is faster than the later iteration stage, while decreasing speed of chemotactic step size (21) is lager than step size in (1) at early stage of iteration and the situation is changed in the later stage of iteration.

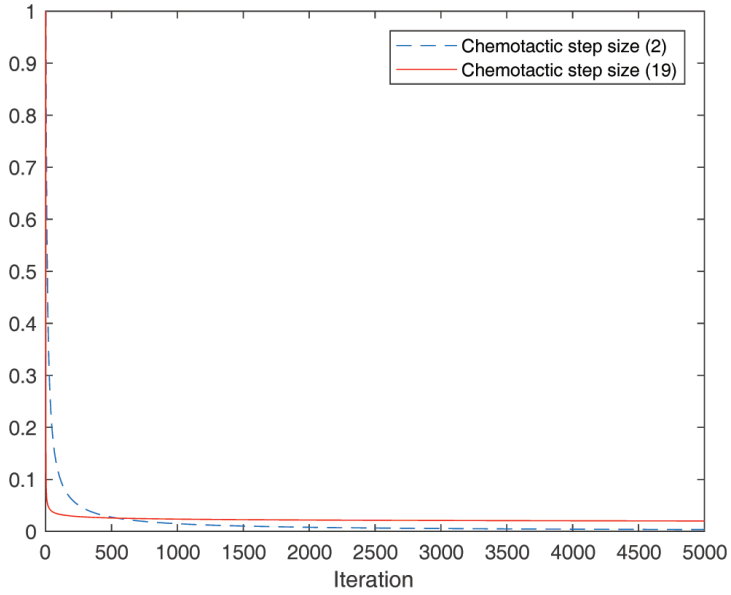

FIGURE 1. Chemotactic step size based on index of iteration

3.4. **The fitness-based adaptive chemotactic step size design.** The analysis in [25] has been revealed that a chemotactic step size defined as the function of the fitness value of a bacterium can lead to better performance as compared to a fixed step size. So we consider the factor of fitness value in chemotactic step size design. The adaptive chemotactic step size function should follow the law that a smaller step size would be adopted when the bacterium is near the optima. In this section, based on the chemotactic step sizes defined as (20) and (21), fitness-based adaptive chemotactic step size function is introduced. We will define the chemotactic step size as

$$C_{FAQBFA} = \begin{cases} \dfrac{c_{\min}}{1 - \left[(1-c_{\min})^{1-\alpha} - k_1(1-\alpha)\dfrac{\ln x}{|J(i,j,k,l)|^{\beta}}\right]^{\frac{1}{1-\alpha}}}, & \alpha \neq 1, \\ \dfrac{c_{\min}}{1-(1-c_{\min})\dfrac{|J(i,j,k,l)|^{\beta}}{x^{k_1}}}, & \alpha = 1, \end{cases} \tag{22}$$

where variable x is assigned to iteration number $Iter = i \times j \times k \times l$, k_1, α, β and $c_{\min}$ are positive constant numbers and $c_{\min}$ is the lower bound of chemotactic step size in the algorithm. C_{FAQBFA} is designed such that C_{FAQBFA} is decreased and approaches $c_{\min}$ when $|J(i,j,k,l)|$ is small or x is large, and C_{FAQBFA} in the first iteration will also be 1 when $\alpha \neq 1$. In this case, C_{FAQBFA} is dependent on both the fitness value of a bacterium and the current iteration number.

3.5. **Spiral dynamics algorithm (SDA).** Inspired by spiral phenomena in nature, Tamura and Yasuda [29] propose a new metaheuristic algorithm which is named SDA. In the algorithm, a center point is specified, other points rotate around the center with an angular and the distance between the center and rotated point decreases in equal ratio. For two dimensional optimization problem and center point is specified as origin, the spiral dynamics model can be depicted as follows.

$$\begin{aligned} \begin{bmatrix} x_1^{s+1} \\ x_2^{s+1} \end{bmatrix} &= \begin{bmatrix} \alpha_1 & -\beta_1 \\ \beta_1 & \alpha_1 \end{bmatrix} \begin{bmatrix} x_1^s \\ x_2^s \end{bmatrix} = r \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_1^s \\ x_2^s \end{bmatrix} \\ &:= A_{spiral}\, \boldsymbol{x}^s, \quad s = 0, 1, \ldots. \end{aligned} \tag{23}$$

where $r = \sqrt{\alpha_1^2 + \beta_1^2}$, and $\theta = \arctan(\beta_1/\alpha_1)$. Conditions $\beta_1 \neq 0$ and $r < 1$ should be guaranteed, where condition $\beta_1 \neq 0$ makes the solution rotated on the $x_1 - x_2$ plane, and condition $r < 1$ can guarantee the series generated by (23) converge to origin for arbitrary initial point. For any values of r and θ, all the search points will settle at a center point of the spiral trajectory at the end of the search process.

For n-dimensional optimization problem and an arbitrary initial point $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ with n set as even number, $\boldsymbol{x}$ should be reshaped as $2 \times n/2$ to perform operation (23), the results can then be catenated as vector when the operation is finished. The spiral model of SDA is defined as

$$\boldsymbol{x}^{s+1} = A_{spiral}\boldsymbol{x}^s - (A_{spiral} - I_n)\boldsymbol{x}^*, \tag{24}$$

where $\boldsymbol{x}^*$ is a center point of spiral.

SDA is well known for its powerful spiral searching ability. In search space, search points employ a spiral motion trajectory for both exploration and exploitation strategies. The diversification and intensity of all the search points are guaranteed in the search operation. In the model, all the search points are motivated and guided towards the optimum location during a few iteration steps. A swarming behaviour is exhibited in SDA. Inspired by this property of SDA, the traditional swarming mechanism in BFA is replaced by SDA in our algorithms.

3.6. **Adaptive quantum-inspired bacterial foraging algorithm (AQBFA).** Based on quantum principle, two proposed adaptive chemotactic step sizes and SDA, we introduce the AQBFA. In the AQBFA, quantum principle is used to encode the solution into the quantum bacterium, adaptive chemotactic step size is adopted and swarming step is replaced with SDA. The flowchart of AQBFA is shown in Figure 2.

4. **Experiment Results with Benchmark Functions.** Based on the two proposed novel adaptive chemotactic step size schemes and AQBFA in Section 3, two novel bacterial foraging algorithms named IAQBFA and FAQBFA are proposed under the AQBFA model. IAQBFA and FAQBFA denote the AQBFA augmented with adaptive chemotactic step size schemes (21) and (22), respectively. This section illustrates comparisons of the proposed IAQBFA, FAQBFA with QBFA [32], IBFA and FIBFA [31]. The common parameters used in these algorithms are set as $N_b = 50$, $N_c = 100$, $N_s = N_{re} = 4$, $N_{ed} = 2$, and $P_{ed} = 0.25$. For chemotactic step size schemes (21) and (22), the parameters are set as $\alpha = 2$, $c_{\min} =$ 6E-3, $k_1 =$1E-2, and $\beta = 0.2$. The parameters of SDA are set as $\alpha_1 = \beta_1 = 0.67$.

4.1. **Benchmark functions.** In order to compare the performance of the above mentioned five algorithms, there are two classes of benchmark functions to be involved in this section, i.e., basic benchmark functions and CEC05 benchmark functions. The basic benchmark functions [33] and the CEC05 benchmark functions [34] are depicted in Table 2 and Table 3, respectively. These two classes of benchmark functions are employed generally in evolutionary computation field to verify fitness accuracy and convergence speed [7, 26, 27].

In Table 2, the first function is Sphere function (f_1) which is a differentiable separable unimodal function. Quadric function (f_2) and Rosenbrock function (f_3) are differentiable unimodal unseparable functions. Quartic function (f_4) is a discontinuous separable unimodal function which contains a random term. Rastrigin function (f_5) and Ackley function (f_6) are differentiable multimodal functions, where f_5 has many local optima, and f_6 has a large quantity of minor local optima and one narrow global optimum basin. Griewank function (f_7) is a differentiable nonseparable multimodal function. In the seven benchmark functions, f_1 is easy to solve and f_7 is difficult to find the global optimum,

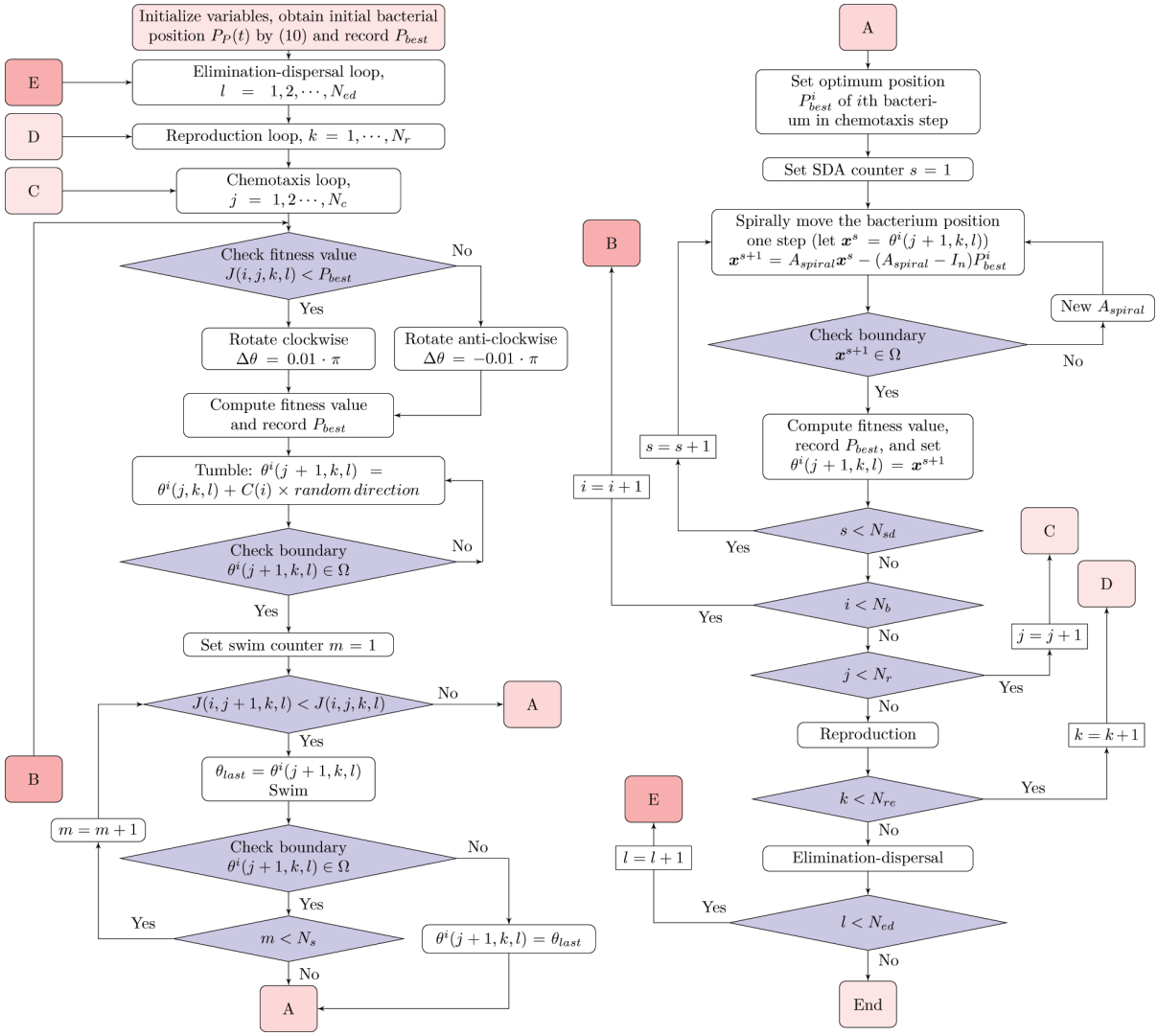

FIGURE 2. Flowchart of the AQBFA

TABLE 2. The basic benchmark functions for comparison [33]

Name	Objective function	Range of search	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^{N} x_i^2$	$[-100, 100]^N$	0
Quadric	$f_2(x) = \sum_{i=1}^{N} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]^N$	0
Rosenbrock	$f_3(x) = \sum_{i=1}^{N-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	$[-100, 100]^N$	0
Quartic	$f_4(x) = \sum_{i=1}^{N} i x_i^4 + rand[0, 1]$	$[-1.28, 1.28]^N$	0
Rastrigin	$f_5(x) = \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i)) + 10$	$[-5.12, 5.12]^N$	0
Ackley	$f_6(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) + 20 + e - \exp\left(\frac{1}{30}\sum_{i=1}^{N}\cos(2\pi x_i)\right)$	$[-32, 32]^N$	0
Griewank	$f_7(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^N$	0

TABLE 3. CEC05 benchmark functions for comparison [34]

Name	Objective function
Shifted Sphere	$f_8(x) = \sum_{i=1}^{N} z_i^2 + f_{bias},\ z = x - \boldsymbol{o}$
Shifted Schwefel's Problem 1.2	$f_9(x) = \sum_{i=1}^{N} \sum_{j=1}^{i} z_j^2 + f_{bias},\ z = x - \boldsymbol{o}$
Shifted Rosenbrock's Function	$f_{10}(x) = \sum_{i=1}^{N-1} (100(z_i^2 - z_{i+1})^2 + (z_i^2 - 1)^2) + f_{bias},$ $z = x - \boldsymbol{o}$
Shifted Rotated Griewank	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^{N} z_i^2 - \prod_{i=1}^{N} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{bias},$ $z = (x - \boldsymbol{o}) * M$
Shifted Rotated Ackley	$f_{12}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} z_i^2}\right) + 20$ $+ e + f_{bias} - \exp\left(\sum_{i=1}^{N} \cos(2\pi z_i)\right),$ $z = (x - \boldsymbol{o}) * M$
Shifted Rastrigin	$f_{13}(x) = \sum_{i=1}^{N} (z_i^2 - 10\cos(2\pi z_i) + 10) + f_{bias},$ $z = x - \boldsymbol{o}$
Shifted Rotated Rastrigin	$f_{14}(x) = (1 + 0.1 \times rand[-1, 1]) \sum_{i=1}^{N} x_i^2 + f_{bias}$

TABLE 4. Parameters of CEC05 benchmark functions ($\boldsymbol{o} = o_1 \cdot I + o_2 \cdot \mathrm{rand}(1, n)$)

Name	o_1	o_2	f_{bias}	Range of search
Shifted Sphere	-100	200	-450	$x \in [-100, 100]^N$
Shifted Schwefel's Problem 1.2	-100	200	-450	$x \in [-100, 100]^N$
Shifted Rosenbrock's Function	-100	200	390	$x \in [-100, 100]^N$
Shifted Rotated Griewank	–	–	-180	No bounds
Shifted Rotated Ackley	-5	10	-140	$x \in [-32, 32]^N$
Shifted Rastrigin	-100	200	-450	$x \in [-5, 5]^N$
Shifted Rotated Rastrigin	–	–	-330	$x \in [-5, 5]^N$

multimodal functions are more difficult to solve than the unimodal function generally. In our experiments, 30, 45 and 60 dimension benchmark functions are adopted. The domain and global optimal value $f_{\min}$ of the functions f_1-f_7 are also presented, and the global optima values of functions f_1-f_7 are zero.

CEC05 benchmark functions f_8-f_{14} listed in Table 3 are shifted and rotated functions, in which f_8-f_{10}, f_{13} are shifted and f_{11}, f_{12}, f_{14} are shifted rotated based on the classical functions. Problems f_8-f_{12} are multimodal functions, f_{13} and f_{14} are unimodal functions. This class of benchmark functions is difficult to solve compared to the basic benchmark functions. In Table 3, M is an orthogonal matrix with a specified constant condition number for a particular problem, $\boldsymbol{o}$ denotes the shifted global optimum, the values of $\boldsymbol{o}$ and f_{bias} are listed in Table 4. The rotated problems are produced by the shifted variable $(x - \boldsymbol{o})$ left multiplied the matrix M.

4.2. **Results for the benchmark functions.** All of the five algorithms are performed 30 times independently on the specified dimension of each benchmark function. The mean and standard deviation (Std) of best fitness values are computed based on the result of

each run. A two tailed t-test is adopted to compare the significant difference of the performance between the best of IAQBFA and FAQBFA and other four algorithms. Because the required conditions for the safe usage of parametric tests may not be guaranteed in tests, the non-parametric statistic techniques should be employed. Hence, Wilcoxon signed rank test and Friedman test are adopted in this paper. The significant level of the three above mentioned tests is set to $\alpha = 0.05$.

For the seven basic benchmark functions, the mean, standard deviation, and t-test values of the 30 runs of the five involved algorithms are presented in Table 5. According to the mean values, the five algorithms are ranked for benchmark functions with a specified dimension. The best mean value and standard deviation among the five algorithms are highlighted in bold font. From Table 5 we can observe that the algorithm FAQBFA has best mean value on f_1, f_2, f_3, f_5 and f_7 benchmark functions for all the three dimensions, the algorithm QBFA has best mean value on f_4 and f_6 benchmark functions for all the three dimensions. The algorithms FAQBFA, IAQBFA and QBFA achieved significantly better results than those of IBFA and FIBFA. The table also shows that the algorithm which has the best mean value always has the best standard deviation for all the benchmark functions except QBFA on the function f_6 with 60-dimension. For function f_6 with 60-dimension, IAQBFA achieves the best standard deviation. In Table 5, t-value denotes the t-test values of the algorithms compared with the best of IAQBFA and FAQBFA, the associated 95% confidential interval and p value are also listed. The p value can provide information about whether a statistic test is significant or not. From Table 5, we can observe that the mean values of FAQBFA are less than IAQBFA's in 101 cases out of 105 cases. In the two tailed t-test, the best of FAQBFA and IAQBFA are compared with other four algorithms, so algorithm FAQBFA is always involved, only in 4 cases other four algorithms are compared with IAQBFA in the t-test. Given a 95% confidence level with 65 samples, the corresponding p-values are listed in Table 5, 60 cases results show that FAQBFA and other four algorithms have significant difference.

Fitness accuracy of the five algorithms is also tested by the non-parametric Wilcoxon signed rank test and Friedman test, significance level of the both tests are set to $\alpha = 0.05$. Analogous to the t-test, Wilcoxon signed rank test aims to test whether there exist significant differences between the given two algorithms. Table 5 presents the results of Wilcoxon signed rank test, in which the proposed algorithms IAQBFA and FAQBFA are compared with other four algorithms, respectively. Table 5 shows that all of the p values are less than 0.05. It means that the differences between FAQBFA and other four algorithms are significant and the differences between IAQBFA and other four algorithms are also significant. Wilcoxon signed rank test results are listed in Table 7. Friedman test based on results in Table 5 are listed in Table 8. Notice that, the lowest ranking highlighted by bold font is the FAQBFA, the ascending order among the other four are QBFA, IAQBFA, FIBFA and IBFA, respectively. In summary, for the seven basic benchmark functions, FAQBFA is the best algorithm according to accuracy criterion. It also illustrates that adaptive step size scheme (22) designed with iteration and fitness value is more efficient than the other schemes.

The convergence speed performance is plotted for the five algorithms. In this paper, only the functions f_1-f_7 with dimension 30 are presented in Figure 3 for IBFA, FIBFA, QBFA, IAQBFA and FAQBFA in terms of the average of the best fitness scale in $\log 10$ versus first 40,000 fitness evaluations. Plots show that IAQBFA and FAQBFA schemes have faster convergence speed in the first 40,000 fitness evaluations than others. For the other three algorithms, the convergence speeds of QBFA and FIBFA are similar, IBFA is the lowest algorithm among the five algorithms generally.

TABLE 5. Comparison of means and standard deviations for f_1-f_7 with different dimensions (averaged over 30 runs)

Fun.	Alg.	Dim	Mean	Std	Rank	t-Value	95% Confidential interval	p
f_1	IBFA	30	3.5413e-04	6.6200e-05	5	29.300	(3.2941e-4, 3.7885e-4)	< 0.05
		45	3.7595e-04	7.0705e-05	5	29.123	(3.4955e-4, 4.0235e-4)	< 0.05
		60	3.7358e-04	6.9194e-05	5	29.572	(3.4774e-4, 3.9942e-4)	< 0.05
	FIBFA	30	8.9029e-05	1.6503e-06	4	295.480	(8.8413e-5, 8.9645e-5)	< 0.05
		45	8.9102e-05	5.3927e-07	4	9049.051	(8.9082e-5, 8.9122e-5)	< 0.05
		60	8.9099e-05	5.3997e-07	4	9036.610	(8.9065e-5, 8.9105e-5)	< 0.05
	QBFA	30	2.3175e-07	3.7625e-07	2	3.373	(9.1234e-8, 3.7223e-7)	< 0.05
		45	5.1032e-07	8.6408e-07	2	3.235	(1.8766e-7, 8.3297e-7)	< 0.05
		60	8.2765e-07	1.7586e-06	3	2.578	(1.7096e-7, 1.4843e-6)	< 0.05
	IAQBFA	30	3.6095e-07	7.3809e-07	3	2.678	(8.5329e-8, 6.3655e-7)	< 0.05
		45	5.7802e-07	1.0229e-07	3	3.095	(1.9607e-7, 9.5995e-7)	< 0.05
		60	7.1193e-07	1.2472e-07	2	3.126	(2.4621e-7, 1.1776e-6)	< 0.05
	FAQBFA	30	**1.5866e-11**	**3.7492e-11**	**1**	–	–	–
		45	**1.0399e-11**	**1.9515e-11**	**1**	–	–	–
		60	**1.0264e-11**	**1.4813e-11**	**1**	–	–	–
f_2	IBFA	30	2.0624e-03	4.6954e-04	4	24.059	(1.8871e-3, 2.3778e-3)	< 0.05
		45	2.1890e-03	5.1702e-04	4	23.190	(1.9959e-3, 2.3821e-3)	< 0.05
		60	2.1848e-03	3.5932e-04	4	33.299	(2.0506e-3, 2.3190e-3)	< 0.05
	FIBFA	30	9.8003e-03	5.8860e-04	5	91.197	(9.5805e-3, 1.0020e-2)	< 0.05
		45	9.6411e-03	5.1859e-04	5	101.828	(9.4475e-3, 9.8347e-3)	< 0.05
		60	9.9143e-02	7.9466e-04	5	68.333	(9.6175e-3, 1.0211e-2)	< 0.05
	QBFA	30	8.1484e-05	1.3448e-04	2	3.319	(3.1265e-5, 1.3170e-4)	< 0.05
		45	7.0798e-04	1.5044e-03	3	2.578	(1.4622e-4, 1.2697e-3)	< 0.05
		60	7.4818e-04	1.1024e-03	3	3.717	(3.3651e-4, 1.1598e-3)	< 0.05
	IAQBFA	30	9.3802e-05	3.0022e-04	3	1.711	(−1.8304e-5, 2.0590e-4)	0.098
		45	2.9594e-04	7.0099e-04	2	2.312	(3.4172e-5, 5.5768e-4)	< 0.05
		60	6.1194e-04	1.1555e-03	2	2.900	(1.8043e-4, 1.0434e-3)	< 0.05
	FAQBFA	30	**1.9573e-09**	**2.4119e-09**	**1**	–	–	–
		45	**1.1519e-08**	**1.6307e-08**	**1**	–	–	–
		60	**3.9203e-08**	**1.0969e-07**	**1**	–	–	–
f_3	IBFA	30	2.9123e+00	7.4076e-03	4	2153.348	(2.9095e+0, 2.9150e+0)	< 0.05
		45	2.9089e+00	6.8679e-03	4	2319.845	(2.9063e+0, 2.9115e+0)	< 0.05
		60	2.1848e-03	6.8679e-04	4	33.304	(2.0506e-3, 2.3190e-3)	< 0.05
	FIBFA	30	3.3996e+00	1.1144e-01	5	167.089	(3.3579e+0, 3.4412e+0)	< 0.05
		45	3.4173e+00	1.5301e-01	5	122.326	(3.3602e+0, 3.4745e+0)	< 0.05
		60	3.3795e+00	1.0354e-01	5	178.774	(3.3409e+0, 3.4182e+0)	< 0.05
	QBFA	30	3.3631e-06	5.7352e-06	3	3.212	(1.2215e-6, 5.5045e-6)	< 0.05
		45	5.2958e-06	1.7822e-05	3	1.628	(−1.3565e-6, 1.1953e-5)	0.114
		60	4.9209e-06	1.4274e-05	2	1.888	(−4.0932e-7, 1.0251e-5)	0.069
	IAQBFA	30	1.5292e-06	1.4311e-06	2	5.852	(9.9473e-7, 2.0635e-6)	< 0.05
		45	2.1302e-06	2.5689e-06	2	4.542	(1.1708e-6, 3.0892e-6)	< 0.05
		60	5.2821e-06	9.1432e-06	3	3.164	(1.8678e-6, 8.6961e-6)	< 0.05
	FAQBFA	30	**7.9035e-11**	**1.5299e-10**	**1**	–	–	–
		45	**2.3508e-10**	**6.2157e-11**	**1**	–	–	–
		60	**1.2115e-10**	**2.7715e-10**	**1**	–	–	–
f_4	IBFA	30	3.3103e-02	2.2559e-02	5	8.028	(2.4602e-2, 4.1425e-2)	< 0.05
		45	2.7716e-02	2.3251e-02	5	6.508	(1.8928e-2, 3.6278e-2)	< 0.05
		60	2.6459e-02	1.5084e-02	4	9.593	(2.0747e-2, 3.1991e-2)	< 0.05
	FIBFA	30	2.6220e-02	2.0288e-02	4	7.070	(1.8572e-2, 3.3690e-2)	< 0.05
		45	2.4181e-02	2.1077e-02	4	6.253	(1.6196e-2, 3.1940e-2)	< 0.05
		60	2.9481e-02	2.1111e-02	5	7.628	(2.1509e-2, 3.7270e-2)	< 0.05

TABLE 5. (continued)

Fun.	Alg.	Dim	Mean	Std	Rank	t-Value	95% Confidential interval	p
	QBFA	30	**4.3413e-05**	**4.6293e-05**	**1**	−2.230	(−8.8632e-5, −3.8403e-6)	< 0.05
		45	**3.2872e-05**	**2.3298e-05**	**1**	−3.318	(−1.2976e-4, −3.0799e-5)	< 0.05
		60	**4.1447e-05**	**3.1898e-05**	**1**	−2.235	(−9.5008e-5, −4.2188e-6)	< 0.05
	IAQBFA	30	9.3498e-05	7.6371e-04	3	0.185	(−3.8663e-5, 4.6360e-5)	0.854
		45	1.1315e-04	1.3038e-04	2	–	–	–
		60	2.7219e-04	3.5023e-04	3	2.571	(3.7051e-5, 3.2520e-4)	< 0.05
	FAQBFA	30	8.9649e-05	9.4650e-05	2	–	–	–
		45	1.3110e-04	1.4823e-04	3	0.496	(−5.6020e-5, 9.1922e-5)	0.623
		60	9.1061e-05	1.1655e-04	2	–	–	–
f_5	IBFA	30	7.1197e-02	1.3745e-02	5	28.372	(6.6064e-2, 7.6329e-2)	< 0.05
		45	7.5477e-02	1.3621e-02	5	30.350	(7.0391e-2, 8.0563e-2)	< 0.05
		60	7.1408e-02	1.3332e-02	5	29.336	(6.6429e-2, 7.6386e-2)	< 0.05
	FIBFA	30	1.2425e-02	4.2583e-04	4	159.810	(1.2266e-2, 1.2584e-2)	< 0.05
		45	1.2520e-02	4.4689e-04	4	153.452	(1.2353e-2, 1.2687e-2)	< 0.05
		60	1.2465e-02	4.4736e-04	4	152.610	(1.2298e-2, 1.2632e-2)	< 0.05
	QBFA	30	2.6115e-07	4.8446e-07	2	2.952	(8.0246e-8, 4.4205e-7)	< 0.05
		45	1.5156e-07	2.2865e-07	2	3.630	(6.6173e-8, 2.3693e-7)	< 0.05
		60	1.1075e-07	1.4497e-07	2	4.184	(5.6612e-8, 1.6488e-7)	< 0.05
	IAQBFA	30	7.8065e-07	1.3103e-06	3	3.263	(2.9139e-7, 1.2699e-6)	< 0.05
		45	6.4217e-07	1.0582e-06	3	3.327	(2.4756e-7, 1.0378e-6)	< 0.05
		60	5.7578e-07	1.1905e-06	3	2.649	(1.3123e-7, 1.0203e-6)	< 0.05
	FAQBFA	30	**3.7961e-12**	**7.6552e-12**	**1**	–	–	–
		45	**5.7661e-12**	**8.0501e-12**	**1**	–	–	–
		60	**1.0303e-11**	**2.6130e-11**	**1**	–	–	–
f_6	IBFA	30	4.9306e-03	4.8296e-04	5	54.045	(4.6265e-3, 4.9905e-3)	< 0.05
		45	4.9734e-03	5.6882e-04	5	47.224	(4.6768e-3, 5.1002e-3)	< 0.05
		60	5.0100e-03	5.0475e-04	5	52.385	(4.7251e-3, 5.1091e-3)	< 0.05
	FIBFA	30	2.2437e-03	1.4948e-05	4	105.882	(2.0805e-3, 2.1625e-3)	< 0.05
		45	2.2489e-03	1.1776e-05	4	143.584	(2.1332e-3, 2.1949e-3)	< 0.05
		60	2.2480e-03	1.4413e-05	4	163.006	(2.1280e-3, 2.1821e-3)	< 0.05
	QBFA	30	**8.1851e-05**	**7.1877e-05**	**1**	−2.318	(−7.5860e-5, −4.746e-6)	< 0.05
		45	**5.6487e-05**	**7.2439e-05**	**1**	−1.326	(−7.2238e-5, 1.5423e-5)	0.195
		60	**7.9332e-05**	8.4479e-05	**1**	−0.829	(−4.7155e-5, 1.9945e-5)	0.414
	IAQBFA	30	1.2215e-04	1.0597e-04	2	–	–	–
		45	8.4895e-05	8.3114e-05	2	–	–	–
		60	9.2937e-05	**6.7836e-05**	2	–	–	–
	FAQBFA	30	9.2354e-01	1.4084e+00	5	3.591	(3.9751e-1, 1.4493e+0)	< 0.05
		45	1.5493e+00	1.6228e+00	5	5.229	(9.4329e-1, 2.1552e+0)	< 0.05
		60	1.9786e+00	1.7065e+00	5	6.350	(1.3413e+0, 2.6157e+0)	< 0.05
f_7	IBFA	30	2.4732e-05	4.0951e-06	5	33.078	(2.3202e-5, 2.6261e-5)	< 0.05
		45	2.5503e-05	3.6765e-06	5	37.994	(2.4130e-5, 2.6876e-5)	< 0.05
		60	2.5941e-05	2.9350e-06	5	48.410	(2.4845e-5, 2.7037e-5)	< 0.05
	FIBFA	30	5.2693e-06	4.4101e-10	4	failed	failed	failed
		45	5.2693e-06	4.9711e-10	4	failed	failed	failed
		60	5.2693e-06	3.8092e-10	4	failed	failed	failed
	QBFA	30	7.3604e-07	1.4991e-06	2	2.689	(1.7624e-7, 1.2958e-6)	< 0.05
		45	1.1326e-06	1.1604e-06	3	3.782	(5.2011e-7, 1.7451e-6)	< 0.05
		60	1.0036e-06	2.1783e-06	3	2.523	(1.9017e-7, 1.8169e-6)	< 0.05
	IAQBFA	30	8.6690e-07	1.9908e-06	3	2.385	(1.2348e-7, 1.6103e-6)	< 0.05
		45	6.4657e-07	1.8035e-06	2	1.964	(−2.6892e-8, 1.3199e-6)	0.059
		60	7.1698e-07	1.3082e-06	2	3.002	(2.2846e-7, 1.2055e-6)	< 0.05
	FAQBFA	30	**2.5345e-11**	**4.8562e-11**	**1**	–	–	–
		45	**1.6706e-11**	**4.2225e-11**	**1**	–	–	–
		60	**1.5648e-11**	**2.8642e-11**	**1**	–	–	–

TABLE 6. Performance of all algorithms on CEC05 benchmark functions (averaged over 30 runs)

Function		IBFA	FIBFA	QBFA	IAQBFA	FAQBFA
f_8	Mean	8909.4279	8650.6850	−449.4366	**−449.5195**	−232.4032
	Std	3.1271	41.8928	0.1476	**0.1374**	50.7970
	Rank	5	4	2	**1**	3
f_9	Mean	8909.6041	8653.9868	−449.3659	**−449.4842**	−225.0580
	Std	2.9858	46.9612	0.1701	**0.1517**	62.7650
	Rank	5	4	2	**1**	3
f_{10}	Mean	8992.9453	8721.3704	390.5751	**390.4915**	650.1049
	Std	3.3611	50.0396	0.1969	**0.1305**	75.5439
	Rank	5	4	2	**1**	3
f_{11}	Mean	8935.4329	8674.7632	6689.7910	**4656.6614**	5911.8439
	Std	2.3743	41.3210	6524.5539	**43.0837**	483.1981
	Rank	5	4	3	**1**	2
f_{12}	Mean	8939.5689	8681.5180	6656.4715	−119.0426	**−119.0299**
	Std	2.7076	46.1616	9923.6429	**0.0447**	0.0953
	Rank	5	4	3	2	**1**
f_{13}	Mean	8920.2839	8676.8111	10283.4198	**−309.3002**	−166.5421
	Std	3.5596	52.9493	13693.1746	**0.0959**	27.6014
	Rank	4	3	5	**1**	2
f_{14}	Mean	8919.5089	8663.0473	9677.1638	**−309.2905**	−164.1773
	Std	3.0473	45.7693	11164.5167	**0.0815**	25.2951
	Rank	4	3	5	**1**	2

TABLE 7. Results of Wilcoxon signed rank test based on results in Table 5 and Table 6 (The results based on Table 6 are listed in bracket.)

Algorithms	R^+	R^-	p value	Significant (Improvement)
IBFA vs. IAQBFA	626 (210)	4 (0)	< 0.05 (< 0.05)	YES (YES)
FIBFA vs. IAQBFA	626 (210)	4 (0)	< 0.05 (< 0.05)	YES (YES)
QBFA vs. IAQBFA	274 (174)	356 (36)	< 0.05 (< 0.05)	YES (YES)
IBFA vs. FAQBFA	552 (210)	78 (0)	< 0.05 (< 0.05)	YES (YES)
FIBFA vs. FAQBFA	549 (210)	81 (0)	< 0.05 (< 0.05)	YES (YES)
QBFA vs. FAQBFA	480 (102)	150 (108)	< 0.05 (< 0.05)	YES (YES)
IAQBFA vs. FAQBFA	507 (11)	123 (199)	< 0.05 (< 0.05)	YES (YES)

For CEC05 benchmark functions, we only perform the experiments on 30-dimension functions. Table 6 provides the mean and standard deviation of the 30 runs of the five involved algorithms. In terms of fitness accuracy, IAQBFA achieved the best performance on f_8-f_{11}, and f_{14} out of the seven functions, FAQBFA achieved the best performance on f_{12}. It is clearly visible that algorithm IAQBFA achieved more competitive performance on CEC05 test functions than its performance on basic benchmarks. We only carry out the non-parametric Wilcoxon signed rank test and Friedman test for seven CEC05 benchmark functions based on the results of Table 6. The results of Wilcoxon signed rank test and Friedman test are listed in the bracket of Table 7 and Table 8, respectively. Table 7 shows

TABLE 8. Results of Friedman test based on results in Table 5 and Table 6 (The results based on Table 6 are listed in bracket.)

Algorithm	Mean rank	χ^2	p value
IBFA	4.50 (4.98)	1627.91 (735.02)	< 0.05 (< 0.05)
FIBFA	4.21 (3.98)		
QBFA	2.21 (2.35)		
IAQBFA	2.38 (**1.22**)		
FAQBFA	**1.69** (2.46)		

that all of the p values are less than 0.05. It means that the differences between FAQBFA and other four algorithms are significant and the difference between IAQBFA and other four algorithms are also significant. In Table 8, the lowest ranking highlighted by bold font belongs to IAQBFA, which is followed by QBFA, FAQBFA, FIBFA and IBFA. Obviously, for the CEC05 benchmark functions, IAQBFA is the best algorithm according to fitness accuracy, algorithm FAQBFA performs not so good as the previous test functions. It is may be due to the fact that the global optimum of CEC05 benchmark functions are not 0 any more, the algorithms with iteration based chemotactic step size (21) is more efficient than the algorithms with iteration and fitness value based chemotactic step size.

5. **Conclusions.** Novel bacterial foraging algorithms based on quantum mechanism, adaptive chemotactic step size, and SDA have been presented in this paper. Firstly, a differential equation was established based on the analysis of bacteria motion pattern near the optima, and the chemotactic step size based on iteration index was obtained by solving the differential equation. Furthermore, the adaptive chemotactic step size based on individual bacterium fitness value and current iteration number was proposed. The chemotactic step size schemes varied dynamically, hence achieved better exploration and exploitation strategies. The traditional swarming mechanism in BFA was replaced by SDA. With the combination of different strategies, two versions of QBFA have been proposed. The proposed two QBFA variants have been compared with QBFA, IBFA and FIBFA on seven basic benchmark functions and seven CEC05 benchmark functions. The performance of the algorithms has been statistically evaluated with fitness accuracy and convergence speed. Two tailed t-test and non-parametric Wilcoxon signed rank test have been used to assess the significant difference between the proposed two algorithms and the three algorithms in references. Friedman test has been used to rank the performance of the five algorithms. The numerical and graphical results show that FAQBFA significantly outperformed QBFA, IAQBFA, IBFA and FIBFA on both the fitness accuracy and convergence speed for basic benchmark functions. While for CEC05 benchmark functions, IAQBFA is the best algorithm. In the future, we will extend application of IAQBFA and FAQBFA to multi-objective discrete optimization problems and complex real-world problems.

Acknowledgements. The authors are grateful for the support of the National Natural Science Foundation of China (Grant Nos. 61603126 and 11471088), and this work is also supported by the Natural Science Foundation of Heilongjiang Province (No. QC2016094).

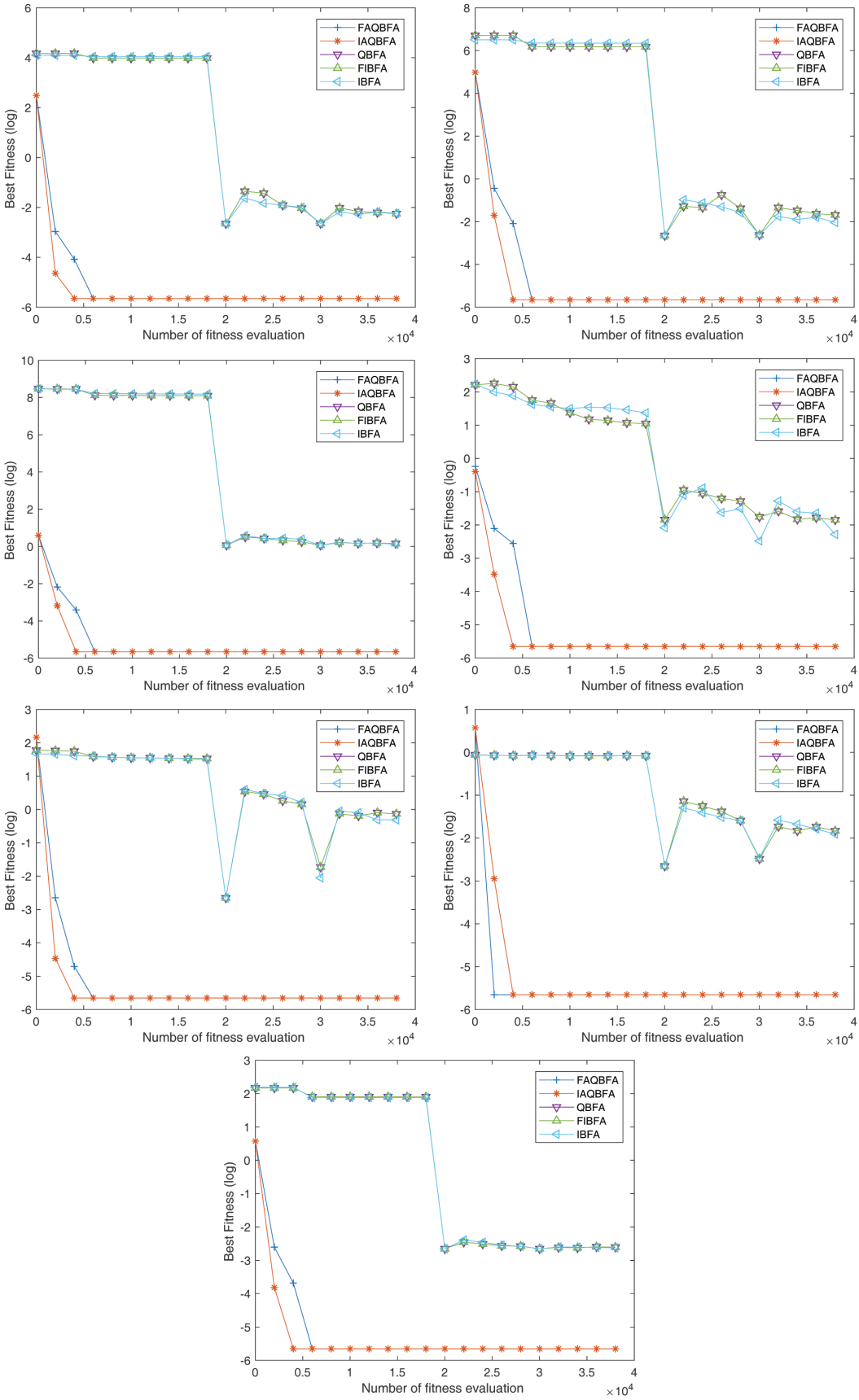

FIGURE 3. Comparison of convergence plots of f_1-f_7 for IBFA, FIBFA, QBFA, IAQBFA and FAQBFA

REFERENCES

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

[2] S. Devi and M. Geethanjali, Application of modified bacterial foraging optimization algorithm for optimal placement and sizing of distributed generation, *Expert Systems with Applications*, vol.41, no.6, pp.2772-2781, 2014.

[3] M. Eslamian, S. H. Hosseinian and B. Vahidi, Bacterial foraging-based solution to the unit-commitment problem, *IEEE Trans. Power Systems*, vol.24, no.3, pp.1478-1488, 2009.

[4] S. Tabatabaei and B. Vahidi, Bacterial foraging solution based fuzzy logic decision for optimal capacitor allocation in radial distribution system, *Electric Power Systems Research*, vol.81, no.4, pp.1045-1050, 2011.

[5] S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, *IEEE Trans. Evolutionary Computation*, vol.9, no.1, pp.61-73, 2005.

[6] E. Bermejo, O. Cordón, S. Damas and J. Santamaría, A comparative study on the application of advanced bacterial foraging models to image registration, *Information Sciences*, vol.295, pp.160-181, 2015.

[7] A. N. K. Nasir and M. O. Tokhi, An improved spiral dynamic optimization algorithm with engineering application, *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.45, no.6, pp.943-954, 2015.

[8] 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.

[9] M. Ulagammai, P. Venkatesh, P. Kannan and N. P. Padhy, Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting, *Neurocomputing*, vol.70, no.16, pp.2659-2667, 2007.

[10] V. P. Sakthivel, R. Bhuvaneswari and S. Subramanian, Non-intrusive efficiency estimation method for energy auditing and management of in-service induction motor using bacterial foraging algorithm, *IET Electric Power Applications*, vol.4, no.8, pp.579-590, 2010.

[11] A. Biswas, S. Dasgupta, S. Das and A. Abraham, A synergy of differential evolution and bacterial foraging optimization for global optimization, *Neural Network World*, vol.17, no.6, pp.607-626, 2007.

[12] S. Gholami-Boroujeny and M. Eshghi, Non-linear active noise cancellation using a bacterial foraging optimisation algorithm, *IET Signal Processing*, vol.6, no.4, pp.364-373, 2012.

[13] R. Majhi, G. Panda, B. Majhi and G. Sahoo, Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques, *Electric Power Systems Research*, vol.36, no.6, pp.10097-10104, 2009.

[14] T. J. Hsieh, A bacterial gene recombination algorithm for solving constrained optimization problems, *Applied Mathematics and Computation*, vol.231, pp.187-204, 2014.

[15] 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.

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

[17] H. Chen, Y. Zhu and K. Hu, Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning, *Applied Soft Computing*, vol.10, no.2, pp.539-547, 2010.

[18] W. G. Zhao and L. Y. Wang, An effective bacterial foraging optimizer for global optimization, *Information Sciences*, vol.329, pp.719-735, 2016.

[19] S. R. N. Huang and M. H. Chen, Constructing optimized interval type-2 TSK neuro-fuzzy systems with noise reduction property by quantum inspired BFA, *Neurocomputing*, vol.173, pp.1839-1850, 2016.

[20] Y. P. Chen, Y. Li, G. Wang and Y. F. Zheng, A novel bacterial foraging optimization algorithm for feature selection, *Expert Systems with Applications*, vol.83, pp.1-17, 2017.

[21] C. C. Yang, J. Z. Ji, J. M. Liu, J. D. Liu and B. C. Yin, Structural learning of Bayesian networks by bacterial foraging optimization, *International Journal of Approximate Reasoning*, vol.69, pp.147-167, 2016.

[22] S. Das, S. Dasgupta, A. Biswas, A. Abraham and A. Konar, On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm, *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol.39, no.3, pp.670-679, 2009.

[23] L. dos Santos Coelho, C. da Costa Silveira, C. Sierakowski and P. Alotto, Improved bacterial foraging strategy applied to team workshop benchmark problem, *IEEE Trans. Magnetics*, vol.46, no.8, pp.2903-2906, 2010.

[24] C. Venkaiah and D. M. V. Kumar, Fuzzy adaptive bacterial foraging congestion management using sensitivity based optimal active power re-scheduling of generators, *Applied Soft Computing*, vol.11, no.8, pp.4921-4930, 2011.

[25] S. Dasgupta, S. Das, A. Abraham and A. Biswas, Adaptive computational chemotaxis in bacterial foraging optimization: An analysis, *IEEE Trans. Evolutionary Computation*, vol.13, no.4, pp.919-941, 2009.

[26] X. Xu and H. L. Chen, Adaptive computational chemotaxis based on field in bacterial foraging optimization, *Soft Computing*, vol.18, no.4, pp.797-807, 2014.

[27] H. Chen, Y. Zhu, K. Hu and L. Ma, Bacterial colony foraging algorithm: Combining chemotaxis, cell-to-cell communication, and self-adaptive strategy, *Information Sciences*, vol.273, pp.73-100, 2014.

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

[29] K. Tamura and K. Yasuda, Primary study of spiral dynamics inspired optimization, *IEEJ Trans. Electrical and Electronic Engineering*, vol.6, pp.98-100, 2011.

[30] A. N. K. Nasir, M. O. Tokhi and N. Ghani, Novel adaptive bacteria foraging algorithms for global optimization, *Applied Computational Intelligence & Soft Computing*, vol.2014, pp.1-7, 2014.

[31] A. N. K. Nasir, M. O. Tokhi and N. M. A. Ghani, Novel adaptive bacterial foraging algorithms for global optimisation with application to modelling of a TRS, *Expert Systems with Applications*, vol.42, no.3, pp.1513-1530, 2015.

[32] S. R. N. Huang and L. G. Zhao, A comparison between quantum inspired bacterial foraging algorithm and GA-like algorithm for global optimization, *International Journal of Computational Intelligence and Applications*, vol.11, no.3, pp.1969-2013, 2012.

[33] X. Yao, Y. Liu and G. M. Lin, Evolutionary programming made faster, *IEEE Trans. Evol. Comput.*, vol.3, no.2, pp.82-102, 1999.

[34] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger and S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, *KanGAL Report 2005005*, 2005.