IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2017, accepted January 3, 2018, date of publication January 11, 2018, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2791976

A Modified Quantum Bacterial Foraging Algorithm for Parameters Identification of Fractional-Order System

LU LIU[1], LIANG SHAN[1], YUEWEI DAI[1], CHENGLIN LIU[2], AND ZHIDONG QI[1]
[1]School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
[2]Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, Institute of Automation, Jiangnan University, Wuxi 214122, China

Corresponding author: Liang Shan (shanliang@njust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61374153 and Grant 61473138, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20151130, in part by the Six Talent Peaks Project in Jiangsu Province under Grant 2015-DZXX-011, and in part by the China Scholarship Council Fund under Grant 201606845005.

ABSTRACT For many physical systems, the fractional-order system can describe the dynamics more accurately comparing with the integer-order system. However, its parameters, which include not only the identification of model coefficients but also the estimation of fractional orders, are more difficult to be identified. This paper proposes a modified quantum bacterial foraging algorithm (MQBFA) for parameters identification of the fractional-order system, and the MQBFA algorithm is a relatively new issue in recent years with strong engineering backgrounds. Not only does the proposed MQBFA algorithm apply the real number encoding of the single gene to effectively perform encoding, but it also utilizes a continuously varying rotation angle to update the rotation gate. First, a real number encoding of the single gene is designed to improve the searching efficiency of the algorithm. Second, an improved quantum rotation angle is applied to update the rotation angle continuously and adaptively. Third, the modificatory factor of probability amplitude is introduced to enrich the population diversity. In addition, to prevent the generation of invariant solutions in the early stage of quantum evolution, probability of optimal rotation angle is presented. Furthermore, the convergence of the proposed algorithm is analyzed. Based on several benchmark functions and parameters identification of the fractional-order system, simulation results and comparisons demonstrate the effectiveness of the proposed MQBFA algorithm.

INDEX TERMS Quantum bacterial foraging algorithm, fractional-order system, parameter identification, rotation angle, real-coded.

I. INTRODUCTION

A. MOTIVATION

In the past few decades, the fractional calculus theory has attracted attention of many scholars, and the reason lies in the application of the fractional differential equations for many actual systems, such as thermal systems [1], power systems [2], financial systems [3] and hyperchaotic systems [4]. Therefore, it is an important issue to study the fractional-order theory.

The fractional-order model has been widely used in many engineering and science applications because of its lower order, fewer parameters and higher modelling accuracy. Recently, considerable attentions have been paid to study the identification method for the fractional-order structure parameters [5]–[8]. Based on particle swarm optimization (PSO) algorithm, Yuan et al. designed a new method for parameters identification of the fractional-order chaotic delayed systems [5]. Idiou et al. presented a novel time-domain identification scheme by using adjustable fractional-order differentiators to solve a more difficult identification problem [6]. According to the improvements of population initialization, searching equation and the ratio between employed and onlooker bees, Hu et al. introduced a hybrid artificial bee colony (HABC) algorithm to identify the unknown fractional-order chaotic systems [7]. Du et al. investigated an evolutionary optimization approach and applied the method to identify a series of fractional-order chaotic systems with unknown initial values and structure [8].

However, with the increase of the complexity of the fractional-order system, it becomes more and more difficult to ensure the accuracy and stability of the system. So, a more reliable optimization algorithm is reasonable for the design of the fractional-order system.

B. RELATED WORK

Recently, a new bionic algorithm called bacterial foraging algorithm (BFA) was proposed by Passino in 2002 [9]. Compared with other optimization algorithms, BFA includes chemotaxis, reproduction and elimination-dispersal operations, and establishes a relatively complete information sharing mechanism, which is helpful to improve the stability and convergence precision of the algorithm. Panigrahi et al. proposed a multiobjective fuzzy dominance BFA to solve economic emission dispatch problem [10]. Das et al. applied BFA to tuning the fractional-order PID controller in a master-slave chaos synchronization configuration [11]. In [12], BFA was used to generate accurate model for solar photovoltaic (PV) module. Mazouz et al. investigated the robustness of BFA, and combined BFA with PI controller to obtain a good control effect for high-voltage direct current system [13]. In order to solve the problem of fault diagnosis existed in aluminum electrolytic cell, Yi et al. presented an optimized relative transformation matrix using BFA [14]. Based on smallest univalue segment assimilating nucleus principle and BFA, Verma and Parihar designed a fuzzy system for edge detection [15].

Although BFA has made some achievements in practical applications, it's still difficult to resolve the problem of slow convergence rate and large computation, especially in the high dimensional condition. To improve the performance of BFA, many BFA variants have been proposed [16]–[19]. To solve the function optimization problem, Kim *et al.* [16] proposed a hybrid algorithm which was composed of genetic algorithm and bacterial foraging algorithm. To accelerate the optimization speed, Dasgupta *et al.* [17] presented a novel BFA variant with adaptive computational chemotaxis. To make full explore the potential of BFA, Li *et al.* [18] designed a new BFA with varying population. Chen *et al.* [19] put forward a new bacterial colony foraging algorithm based on the combination of colony foraging strategy and BFA. These algorithms are classified into two categories: improve their own parameters and present new algorithms combining with other algorithms. In addition to above improvement strategies, another important improvement is to introduce quantum mechanisms into BFA [20]. Compared with BFA, quantum bacterial foraging algorithm (QBFA) has the quantum characteristics and greatly increases the search space, which can improve the overall performance. Li *et al.* [21] successfully solved the traveling salesman problem using QBFA. In [22], QBFA was used in spectrum sensing, which verified the effectiveness of the algorithm.

C. CONTRIBUTION

The traditional binary coding is adopted in QBFA, but the coding and decoding operations will affect the efficiency of the algorithm. In addition, the rotation angle of the classical QBFA is obtained by look-up table. The rotation angle obtained in this way is discrete and fixed, which can not affect the solution space, and limits the increase of population diversity. Thus, a modified quantum bacterial foraging algorithm (MQBFA) is proposed in this paper. The main contributions can be summarized as follows.

1) A real coding method of single gene is designed to preserve the rich population brought by quantum coding, and eliminate the time of searching optimum.

2) An improved quantum rotation angle is proposed, which can update the rotation angle continuously and adaptively, and enhance the global searching ability of the algorithm.

3) The application of the modificatory factor of probability amplitude enriches the population diversity.

4) A concept called probability of optimal rotation angle is presented to improve the searching performance of the algorithm.

The remainder of this paper is organized as follows. In section 2, the modified quantum bacterial foraging algorithm (MQBFA) is proposed. The convergence of the proposed MQBFA is analyzed in section 3. In section 4, the parameter of MQBFA is analyzed, and the performance of MQBFA is evaluated by seven benchmark functions. In section 5, MQBFA is applied to identify the parameters of fractional-order system. Conclusions are summarized in section 6.

II. MODIFIED QUANTUM BACTERIAL FORAGING ALGORITHM

A. CLASSICAL QBFA ALGORITHM

In the QBFA algorithm, quantum bit is the smallest unit which is also called qubit. A qubit can represent the state 0 and state 1 as well as any superposition of states between the two states. Thus a quantum bit is represented as

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

where, α and β are the probability amplitude of state $|0\rangle$ and state $|1\rangle$, respectively, and meet the requirement: $\alpha^2+\beta^2 = 1$. α^2 and β^2 are the probability of state $|0\rangle$ and state $|1\rangle$, respectively. A qubit is represented by the probability amplitude. For example, a chromosome with qubits is represented as

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_p \\ \beta_1 & \beta_2 & \cdots & \beta_p \end{array}\right], \tag{2}$$

where $|\alpha|^2 + |\beta|^2 = 1, i = 1, 2, \cdots, p$

In the QBFA algorithm, the update process of chromosome state is accomplished by a single quantum rotation gate which is defined as

$$U(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{3}$$

Obviously, $U(\theta)$ is a unitary matrix. The adjustment operation of the quantum rotation gate is shown as follows

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = U(\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}, \tag{4}$$

where the rotation angle is defined as $\theta_i = s(\alpha_i, \beta_i) \cdot \Delta\theta_i$, and its value can be obtained by look-up table from [20].

The QBFA algorithm is a probabilistic algorithm, in which the bacterial population is composed of quantum chromosomes. Suppose that the population of the tth generation is $Q(t) = \{q_1^t, q_2^t, \cdots, q_n^t\}$. The probability amplitude of the jth chromosome is given by

$$q_j^t = \begin{bmatrix} \alpha_1^t & \alpha_2^t & \cdots & \alpha_N^t \\ \beta_1^t & \beta_2^t & \cdots & \beta_N^t \end{bmatrix}, \quad j = 1, 2, \cdots, N, \tag{5}$$

where, n is the population size, and N is the chromosome length.

B. THE PROPOSED MQBFA ALGORITHM

In this paper, there are mainly four improvements in our proposed MQBFA algorithm including encoding mechanism, update strategy, population initialization and the improvement for optimal rotation angle.

1) ENCODING STYLE OF MQBFA ALGORITHM

Compared with binary coding in classical QBFA, an extra storage space exists in the real coding of single gene, and the coding mode is as follows,

$$\left| rx_1 \quad rx_2 \quad \cdots \quad rx_n \right|, \tag{6}$$

$$r = l_{max} - l_{min}, \tag{7}$$

where r is the range of the real domain, l_{max} is the upper limit of the real domain, l_{min} is the lower limit of the real domain, n is the length of chromosome, $x_i (i = 1, 2, \cdots, n)$ is a random number generated from the interval [0, 1], and x_i corresponds with the ith probability amplitude in QBFA algorithm. The form of the corresponding probability amplitude is as follows,

$$\begin{vmatrix} \cos\theta_1 & \cos\theta_2 & \cdots & \cos\theta_n \\ \sin\theta_1 & \sin\theta_2 & \cdots & \sin\theta_n \end{vmatrix} \tag{8}$$

The chromosome with a length of n is coded by the real coding of single gene, and the probability $\cos\theta_i^2$ related to the probability amplitude $\cos\theta_i$ is compared with a random number *rand* generated between 0 and 1. If $\cos\theta_i^2 > rand$, the measured value of the corresponding gene is set as $r * x_i$, otherwise, it is set as $r * rand$. The real number encoding of the single gene is essentially the expansion of the quantum bit encoding.

The new encoding mechanism not only preserves the qubit encoding, but also replaces the binary encoding with real encoding, which retains the rich population brought by quantum encoding and eliminates the time converted from binary form to decimal form.

2) MODIFICATORY FACTOR OF PROBABILITY AMPLITUDE

In order to improve population diversity and searching efficiency of the algorithm, the modificatory factor of probability amplitude (MFB) in [23] is introduced. As the name suggests, the modificatory factor of probability amplitude is used to modify the probability amplitude updated by quantum rotation gate. MFB is defined as follows.

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = \begin{cases} (\sqrt{\gamma}, \sqrt{1-\gamma})^T, & |\alpha_i'|^2 \le \gamma \\ (\sqrt{1-\gamma}, \sqrt{\gamma})^T, & |\alpha_i'|^2 \ge 1-\gamma \\ (\alpha_i', \beta_i')^T, & else \end{cases} \tag{9}$$

where γ is a relatively small positive number, and $[\alpha_i' \ \ \beta_i']^T$ is the probability amplitude updated by quantum gate.

3) IMPROVED QUANTUM ROTATION ANGLE

Some parameters are defined as follows:

θ_i represents the angle of a quantum bit on the unit circle of the current bacterium.

θ_b denotes the angle of a quantum bit on the unit circle of the optimal bacterium.

$[\alpha_i^* \ \ \beta_i^*]^T$ is the probability amplitude related to a quantum bit of the current optimal bacteria.

$[\alpha_i \ \ \beta_i]^T$ is the probability amplitude related to a quantum bit of current bacteria.

θ_0 is the initial value of the rotation angle.

$A_i = \begin{vmatrix} \alpha_0 & \alpha_i \\ \beta_0 & \beta_i \end{vmatrix}$ and $sgn(t) = \begin{cases} 1, & t \ge 0 \\ -1, & t < 0 \end{cases}$ is the *sign* function.

The improved quantum rotation angle is defined as

$$S_1 = sgn|A_i|$$

$$S_2 = sgn(\theta_i - \theta^*), \tag{10}$$

$$\theta = S_1 \theta_0 e^{-(\theta_i - \theta^*)^{S_2}}. \tag{11}$$

S_1 is used to control the direction of the rotation angle, and S_2, θ_0, θ_i and θ^* are used to control the size of the rotation angle. When $m = \theta_i - \theta^* > 0$, we have $S_2 = 1$. Thus, $|\theta'| = \theta_0 e^{-m} < \theta_0$, which implies the value of rotation angle varies towards to the trend of less than θ_0. When $m = \theta_i - \theta^* = 0$, we get $S_2 = 1$. So, $|\theta'| = \theta_0 e^{-m} = \theta_0$, which implies the value of rotation angle varies is the same as the initial value θ_0. When $m = \theta_i - \theta^* < 0$, we have $S_2 = -1$. Therefore, $|\theta'| = \theta_0 e^{-m} > \theta_0$, which implies the value of rotation angle varies towards to the trend of greater than θ_0. The rotation angle obtained by (11) is continuous, and the solution space can be searched comprehensively, which drives the algorithm to find the global optimal solution quickly.

From above analysis, once the value of the initial rotation angle is fixed, the value of the quantum rotation angle is adjusted adaptively around θ_0 until the global optimal rotation angle is obtained. Thus, the value of θ_0 has a great influence on the quantum rotation angle.

4) PROBABILITY OF OPTIMAL ROTATION ANGLE

In the classical QBFA algorithm, the initial values of all probability amplitudes are set as $1/\sqrt{2}$. So, the current optimal rotation angle is the same as the rotation angle of each qubit, which is easy to generate the invariant solutions in the early stage of quantum evolution, and even deteriorates the performance of the algorithm. Motivated from the mutation operation in genetic algorithm, this paper presents a concept called probability of optimal rotation angle. The probability of optimal rotation angle P_θ is adopted to modify the optimal rotation angle, and the corresponding expression is shown below.

$$\theta_b = \begin{cases} \theta' * \mu, & P_\theta < \varepsilon \\ \theta' * (1-\mu), & P_\theta \ge \varepsilon \end{cases} \tag{12}$$

where μ and ε are random numbers obeying (0, 1) distribution. θ_b takes $\theta' * \mu$ or $\theta' * (1-\mu)$ with the same probability. So, the value of P_θ is set as 0.5.

III. CONVERGENCE ANALYSIS

Definition 1: Suppose that the population size is S. Define that f represents the fitness function value, and f_{best} represents the global optimal solution which is defined as $f_{best} = min_{x_i^t \in X^t}\{f(x_i^t), i = 1, 2, \cdots, S\}$. Let $X^t = \{x_1^t, x_2^t, \cdots, x_S^t\}$ represents the position information at the tth iteration.

Definition 2: For a set of random sequences $\tau(n = 1, 2, \cdots)$, where existing a random variable τ, let $p\{\lim\limits_{n\to\infty} \tau = \tau\} = 1$; or for $\forall\delta > 0$, there is $p(\cap_{n=1}^{\infty} \cup_{k\ge n}^{\infty} |\tau_k - \tau| \ge \delta) = 0$, which we call random sequence τ_n converges to random variable τ with probability 1.

Definition 3: Define that $E = \{E_1, E_2, \cdots, E_D\}$ represents the searching space, where $D = \prod\limits_{i=1}^{n} \frac{a_i - b_i}{\varepsilon}$. Suppose that a and b are the upper and lower bounds of the searching space, and ε is a given precision. The state space is defined as:

(1) $A_1 = \{E_i || f_{E_i - f_{best}}| < \varepsilon\}$

(2) $A_2 = \{E_i || f_{E_i - f_{best}}| \ge \varepsilon\}$

Suppose that p_{11} is the transfer probability from A_1 to A_1, p_{21} is the transfer probability from A_2 to A_1, and p_{22} is the transfer probability from A_2 to A_2. Because the strategy of reserving optimum is used for the MQBFA algorithm, we get $p_{11} = 1$.

Lemma 1 [24]: Suppose that $(B_1, B_2, \cdots, B_n)$ is an event sequence of mutual independence in probability space, and $p(B_k)$ is the corresponding probability. Then,

(1) $\sum\limits_{n=1}^{\infty} p(B_k) < \infty, p(\cap_{n=1}^{\infty} \cup_{k=n}^{\infty} B_k) = 0$

(2) $\sum\limits_{n=1}^{\infty} p(B_k) = \infty, p(\cap_{n=1}^{\infty} \cup_{k=n}^{\infty} B_k) = 1$

Lemma 2 [25]: The population sequence $\{Q^t, t \ge 0\}$ of MQBFA is a homogeneous Markov chain with finite state space.

Theorem 1: The optimal solution sequence $\{X^t, t \ge 0\}$ of MQBFA is a homogeneous Markov chain with finite state space.

Proof: The state transfer process in MQBFA is carried out in a finite space, and its operations such as quantum coding, measurement, chemotaxis, reproduction, and elimination-dispersal, are performed in an independent random process. Besides, the whole process of evolution adopts the strategy of reserving the best individual, which means that Q^{t+1} is only related to Q^t and has no relationship with the previous population information. According to Lemma 1, we can complete the proof of Theorem 1. □

Theorem 2: There exists a constant $\mu \in (0, 1)$ in MQBFA, thus, we have $p_{22} < \mu$.

Proof: For a given precision $\varepsilon > 0$, when $\exists l > 0$ and $|X - X_{best}| \le l$, we have $|f(x) - f(x_{best})| < \zeta$, where X_{best} is the optimal solution of the optimization problem, and $0 < \zeta < \varepsilon$. Let $S_1 = \{X||X - X_{best}| \le l\}$, and it is easy to find that $S_1 \subset A_1$.

In the MQBFA algorithm, the operation for modifying the optimal rotation angle is the same as the Gauss mutation. Assume that the individual X in A_2 is implemented with the operation and generate a new individual $X + \lambda$, then we have

$$p\{(X+\lambda) \in S_1\} < p\{(X+\lambda) \in A_1\} = p_{21}. \tag{13}$$

The random variable λ follows Gauss distribution $N(0, \lambda^2)$ whose probability density function is defined as

$$f(x) = \frac{1}{\lambda\sqrt{2\pi}} exp(-\frac{x^2}{2\lambda^2}),$$

Because

$$\begin{aligned} p\{(X+\lambda) \in S_1\} &= \prod_{i=1}^{n} p\{|X_i + \lambda - X_{best}| \le l\} \\ &= \prod_{i=1}^{n} \int_{X_{best} - X_i - l}^{X_{best} - X_i + l} f(x)dx, \end{aligned}$$

we get

$$\prod_{i=1}^{n} \int_{X_{best} - X_i - l}^{X_{best} - X_i + l} f(x)dx < 1 \Rightarrow$$
$$0 < p\{(X+\lambda) \in S_1\} < 1.$$

Because the random variable λ follows Gauss distribution, the variable x related to $p\{(X+\lambda) \in S_1\}$ is continuous. Thus $\exists X' \in E$, we have

$$p\{(X'+\lambda) \in S_1\} = min\{p\{(X+\lambda) \in S_1\}\}. \tag{14}$$

Let $\mu = 1 - p\{(X'+\lambda) \in S_1\}$. According to (13) and (14), we get

$$p\{(X'+\lambda) \in S_1\} \le \{p\{(X+\lambda) \in S_1\}\} < p_{21} = 1 - p_{22} \Rightarrow$$
$$p_{22} < 1 - p\{(X'+\lambda) \in S_1\} \Rightarrow p_{22} < \mu.$$ □

Theorem 3: The MQBFA algorithm is globally convergent.

Proof: For a given precision $\varepsilon > 0$, after evolution from generation t to generation $t+1$, the probability of not meeting the precision is $p' = p_{22} = p\{|f(X) - f(X_{best})| \ge \varepsilon\}$. According to Theorem 2, we have

$$\sum_{t=1}^{\infty} p_t' < \sum_{t=1}^{\infty} \mu. \tag{15}$$

Because $\mu \in (0, 1)$, we can get $\sum\limits_{t=1}^{\infty} \mu = \frac{\mu}{1-\mu}$, and $\sum\limits_{t=1}^{\infty} p_{22} < \frac{\mu}{1-\mu} < \infty$. According to Lemma 1, we have $p(\cap_{n=1}^{\infty} \cup_{k=n}^{\infty} |f_t - f_{best}| \ge \delta) = 0$. Based on Definition 2, we can draw the conclusion that the MQBFA algorithm is globally convergent. □

TABLE 1. Test results with different θ_0.

Algorithm	θ_0	Optimal value	Worst value	Average value	Variance
MQBFA	0.3π	2.8421E-10	1.6768E+00	2.7390E-01	7.4844E-01
	0.5π	2.8421E-10	7.9899E-01	1.8120E-01	5.7107E-01
	0.7π	2.8421E-10	2.4663E-02	2.4680E-03	5.4736E-02
	0.9π	2.8421E-10	6.1665E-06	1.2956E-06	4.3401E-06
	1.1π	2.8421E-10	1.1942E-08	2.6634E-09	2.2653E-13
	1.3π	2.8421E-10	5.6751E-07	1.1951E-07	4.1018E-07
	1.5π	2.8421E-10	1.3106E-03	2.0149E-04	1.6947E-05

IV. PERFORMANCE EVALUATIONS

A. PARAMETER ANALYSIS

According to the analysis for the improved quantum rotation angle, it is clear that the optimal rotation angle is obtained around θ_0, which implies that θ_0 has a great influence on the performance of the algorithm. In order to obtain an excellent performance of the algorithm, the selection for θ_0 is essential. Therefore, a classical function named *Sphere* is used for testing the MQBFA algorithm to get an appropriate θ_0. The function is defined as

$$f_1(x) = \sum_{d=1}^{n} x_d^2. \tag{16}$$

When $x = (0, 0, \cdots, 0)$, the function get the optimum 0. In *Sphere* function, the searching space is set as [-100,100], and the order is set as 2. For the MQBFA algorithm, in order to make the simulation results more convincing, parameters such as S (population size), N_c (number of chemotaxis), N_{re} (number of reproduction), N_{ed} (number of elimination-dispersal), P_{ed} (probability of elimination-dispersal) and *len* (length of chromosome) are set according to [21]. The specific parameters are set as: $S = 40$, $N_c = 50$, $N_{re} = 5$, $N_{ed} = 2$, $P_{ed} = 0.25$ and $len = 44$. Run the algorithm 50 times and the test results with different values of θ_0 are listed in Table 1.

From Table 1, when $\theta_0 = 1.1\pi$, the proposed MQBFA algorithm get the best results regardless of the average value, the worst value or the variance, which implies that the MQBFA algorithm has higher calculation accuracy and stronger stability. It is easy to find that the algorithm can get the same optimal value for different θ_0. The reason for the same optimal value lies in the algorithmic structure.

To test the performance of the proposed method, the MQBFA algorithm is compared with QBFA, QGA and PSO algorithms by seven benchmark functions in Table 2. In Table 2, $f_1 - f_4$ are multimodal functions, and $f_5 - f_7$ are unimodal functions.

B. PARAMETER EVALUATIONS

For the QBFA algorithm, the parameters are set based on [21]: $S = 40$, $N_c = 50$, $N_{re} = 5$, $N_{ed} = 2$, $P_{ed} = 0.25$, $len = 44$. For the MQBFA algorithm, The values of N_c, N_{re}, N_{ed}, *len* and P_{ed} are set as that in QBFA, and other parameters are set as: $\theta_0 = 1.1\pi$ and $P_\theta = 0.5$. For the QGA algorithm, the maximum number of iterations $MaxIter = 500$, crossover probability $P_{cross} = 0.7$ and mutation probability $P_m = 0.15$ [21]. Because MQBFA, QGA and QBFA algorithms are coded by binary mode, PSO algorithm is introduced to highlight the good performance of the proposed scheme. For the PSO algorithm, the population size $S = 40$, the maximum iteration number $MaxIter = 500$, the inertia factor $W = 0.8$, and the learning factor $c_1 = c_2 = 1.5$ [26].

The difficulty of optimization enhances with the increase of dimension. To demonstrate the effectiveness of the proposed MQBFA algorithm, the dimension of the seven benchmark functions is set to 30. Running 30 times, the average value (*Mean*) and the number of iterations (*Iter*) of test results are listed in Table 3.

From Table 3, we note that the MQBFA algorithm has more accurate results than those of QBFA, QGA and PSO algorithms, which indicates that the MQBFA algorithm has higher calculation accuracy. Furthermore, from the aspect of *Iter*, we can easily find that the proposed approach has faster convergence rate than QBFA and QGA algorithms. Besides, the MQBFA algorithm has a relatively smaller convergence rate than PSO. The reason lies in that the MQBFA algorithm adopts binary encoding, which affects the convergence rate of the algorithm.

V. PARAMETER IDENTIFICATION OF FRACTIONAL-ORDER SYSTEM

A. FRACTIONAL CALCULUS THEORY

1) DEFINITION OF THE FRACTIONAL-ORDER DERIVATIVES AND INTEGRALS

Although the fractional calculus has a history of more than 300 years, it is applied in practical fields in recent decades. In the fractional calculus theory, the basic operator is defined as

$${}_aD_t^\alpha = \begin{cases} \dfrac{d^\alpha}{dt^\alpha}, & Re(\alpha) > 0 \\ 1, & Re(\alpha) = 0 \\ \displaystyle\int_a^t (t\tau)^{-\alpha}, & Re(\alpha) < 0, \end{cases} \tag{17}$$

where a and t are the upper and lower bounds of the operational factor. α is the order of calculus, and $Re(\alpha)$ is the real part of α. In the development of the fractional calculus, there are several typical definitions: Caputo, Riemann-Liouville (RL) and Grumwald-Letnikov (GL) definitions [27]. For example, GL is defined as

$${}_aD_t^\alpha f(t) = \lim_{h \to 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-a)/h]} w_j^{(\alpha)} f(t - jh), \tag{18}$$

$$w_j^{(\alpha)} = \frac{(-1)^j \Gamma(\alpha + 1)}{\Gamma(j + 1)(\alpha - j + 1)}, \tag{19}$$

where $\Gamma(\cdot)$ is the Γ function, $[(t - a)/h]$ is the largest integer that is not greater than $(t - a)/h$, and h is the sampling period.

TABLE 2. Basic information of benchmark functions.

Function	Mathematical representation	Range of search	Optimal value
Griewank	$f_1(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	$[-100,100]^D$	0
Ackley	$f_2(x)=20+e-20e^{-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}-e^{-\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$	$[-100,100]^D$	0
Rastrigrin	$f_3(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i^2)+10]$	$[-100,100]^D$	0
Schewefel	$f_4(x)=418.9829D+\sum_{i=1}^{D}(-x_i\sin(\sqrt{x_i}))$	$[-500,500]^D$	0
Rosenbrock	$f_5(x)=\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$[-10,10]^D$	0
Sum Squares	$f_8(x)=\sum_{i=1}^{D}ix_i^2$	$[-100,100]^D$	0
Dixon-Price	$f_9(x)=(x_1-1)^2+\sum_{i=2}^{D}i(2x_i^2-x_{i-1})^2$	$[-10,10]^D$	0

TABLE 3. Performance comparison of MQBFA, QBFA, QGA and PSO algorithms.

Function	MQBFA		QBFA		QGA		PSO	
	Iter	Average value	*Iter*	Average value	*Iter*	Average value	*Iter*	Average value
Griewank	185	2.0675E-01	197	1.3545E+00	332	4.6907E+00	169	1.4752E+00
Ackley	225	1.2485E+00	347	1.0485E+01	316	5.5324E+01	177	2.6374E+01
Rastrigrin	224	2.5657E+01	352	1.1241E+02	359	3.8057E+02	154	3.1428E+03
Schewefel	110	1.6357E+03	125	1.2685E+04	145	1.4651E+04	104	1.7627E+04
Rosenbrock	182	5.5248E+01	246	1.3347E+02	276	1.2348E+02	103	1.8428E+02
Sum Squares	177	2.3852E+00	358	6.2245E+00	357	8.7157E+00	174	1.1175E+01
Dixon-Price	196	4.5874E+01	233	1.5183E+03	254	4.6683E+03	165	8.3671E+03

2) THE FRACTIONAL-ORDER SYSTEM

So far, we have found that many fields such as electrochemistry, control theory and biology, can be accurately described by the fractional difference equations. Under zero initial condition, the corresponding transfer function of the fractional difference equations is given by

$$G(s)=\frac{Y(s)}{U(s)}=\frac{b_m s^{\beta_m}+b_{m-1}s^{\beta_{m-1}}+\cdots+b_0}{a_n s^{\alpha_n}+a_{n-1}s^{\alpha_{n-1}}+\cdots+a_0}, \quad (20)$$

where a_i and b_i are real numbers, α_i and β_i are calculus orders, $u(t)$ and $y(t)$ are the input and output signals of the system, respectively.

With the increase of the fractional order, it is difficult to accurately identify the fractional-order systems. In view of this, a fractional-order system with continuous order is proposed in [28]. And its expression is described as

$$G(s)=\frac{Y(s)}{U(s)}=\frac{b_m s^{m\beta}+b_{m-1}s^{(m-1)\beta}+\cdots+b_0}{a_n s^{n\alpha}+a_{n-1}s^{(n-1)\alpha}+\cdots+a_0}. \quad (21)$$

B. SIMULATIONS

To further prove the effectiveness of the proposed scheme, the MQBFA algorithm as well as QBFA, QGA and PSO algorithms is applied to estimate the parameters of the fractional-order system. The parameter settings of all the algorithms are shown in Sec 4.2.

The simulations are implemented using MATLAB 7.11 on Intel(R) Core(TM) i5-2320 CPU, 3.00 GHz with 4 GB RAM. And the specific identification steps are as follows.

Step1: Determine the parameters to be identified, including model parameters and order.

Step2: Determine the input signal and the fitness function. In this paper, The sum of the squared errors between the output $y(t)$ and the true value $y_0(t)$, i.e., $J=\int_0^T[y(t)-y_0(t)]^2dt$ is used as the fitness function, and the sinusoidal signal is introduced as the input signal.

Step3: Initialize the parameters of the algorithm to generate searching vector.

Step4: The parameters in the fractional-order system are identified with the corresponding algorithm.

Step5: If the deadline condition is satisfied, the algorithm will stop, otherwise, go to step 4.

The structure chart of parameter estimation for the fractional-order system is shown in Figure 1.

1) IDENTIFICATION FOR THE KNOWN FRACTIONAL-ORDER MODEL STRUCTURE

Suppose that the transfer function of the identified system is known, its expression is shown below.

$$G(s)=\frac{1}{2.2S^{2.3}+1.4S^{1.1}+1} \quad (22)$$

For the fractional-order system, the searching range of coefficients is set as [0, 3], and the fluctuation range of order is limited as [−0.05, 0.05]. The identification vector is described as

$$x=[a_1 \quad a_2 \quad a_3 \quad b_1 \quad b_2],$$

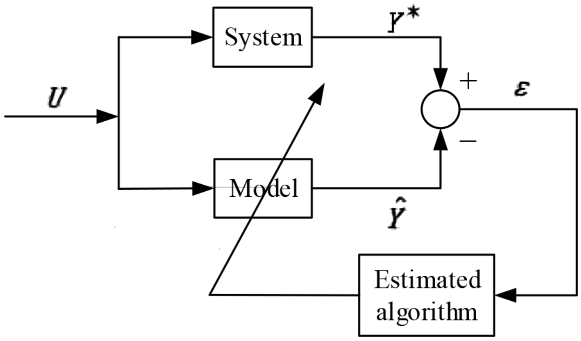

FIGURE 1. Structure chart of parameter estimation for fractional-order system.

TABLE 4. The estimation results of various algorithms.

Algorithm	PSO	QGA	QBFA	MQBFA
$Best$				
a_1	2.1808	2.1846	2.1895	2.1987
a_2	1.3952	1.3970	1.3982	1.3998
a_3	0.9902	0.9911	0.9943	0.9994
b_1	2.2891	2.2898	2.2953	2.2998
b_2	1.0947	1.0954	1.0987	1.0995
J	2.4751E-02	2.1630E-02	8.0712E-03	6.8319E-04
$Mean$				
a_1	2.1748	2.1793	2.1855	2.1981
a_2	1.3712	1.3797	1.3859	1.3935
a_3	0.9826	0.9847	0.9878	0.9988
b_1	2.2745	2.2782	2.2811	2.2943
b_2	1.0809	1.0862	1.0916	1.0982
J	6.4241E-01	5.4221E-01	4.0912E-01	8.8012E-02
$Worst$				
a_1	2.1315	2.1459	2.1515	2.1236
a_2	1.4819	1.4413	1.2801	1.3531
a_3	1.1123	0.8562	0.9501	1.0712
b_1	2.5142	2.6421	2.1411	2.1708
b_2	1.1793	1.1384	0.8368	1.0146
J	2.0846E-01	1.8415E-01	16052E-01	1.4902E-01

where a_1, a_2 and a_3 are the model parameters, and b_1 and b_2 are the orders of the model. Table 4 shows the best, the mean and the worst estimated parameters with over 30 independent runs.

TABLE 5. The estimation results of various algorithms.

Algorithm	PSO	QGA	QBFA	MQBFA
$Best$				
a_1	2.1057	2.1176	2.1227	2.1531
a_2	1.2169	1.2275	1.2337	1.2837
a_3	0.9328	0.9424	0.9458	0.9931
b_1	2.1607	2.1692	2.1965	2.2184
b_2	1.0057	1.0183	1.0395	1.0754
J	2.2181E+01	2.1842E+01	8.3842E-01	1.1273E-01
$Mean$				
a_1	1.9832	2.0157	2.0387	2.1225
a_2	1.1497	1.1572	1.2284	1.2694
a_3	0.8995	0.9107	0.9521	0.9897
b_1	2.1383	2.1422	2.1562	2.2058
b_2	0.9704	0.9721	1.0124	1.0537
J	2.2145E+02	7.2815E+01	1.0574E+01	4.2178E+00
$Worst$				
a_1	1.9354	1.9398	1.9724	2.0571
a_2	1.1298	1.1357	1.1852	1.2464
a_3	0.8934	0.9036	0.9381	0.9537
b_1	2.0813	2.0872	2.0918	2.1574
b_2	0.9187	0.9215	0.9834	1.0281
J	7.1854E+02	1.2721E+02	3.7541E+01	8.9475E+00

From Table 4, an interesting observation to notice that the estimated values obtained by MQBFA are much closer to the true values, which reveals that the proposed method is more accurate than the other three algorithms being compared. In addition, it is easy to find that the MQBFA algorithm can obtain the optimal fitness function values whether in the optimal value, the mean value or the worst value.

In order to verify the search efficiency of MQBFA algorithm, the evolutionary process of the fitness function and the estimated parameters are shown in Fig. 2. Fig. 2 illustrates that the estimated parameters can quickly converge to the true values, and the fitness function value also has a fast convergence rate, which means the MQBFA algorithm has better convergence characteristics.

In order to study the influence of the variation range of order on the identification effect, the variation range of order

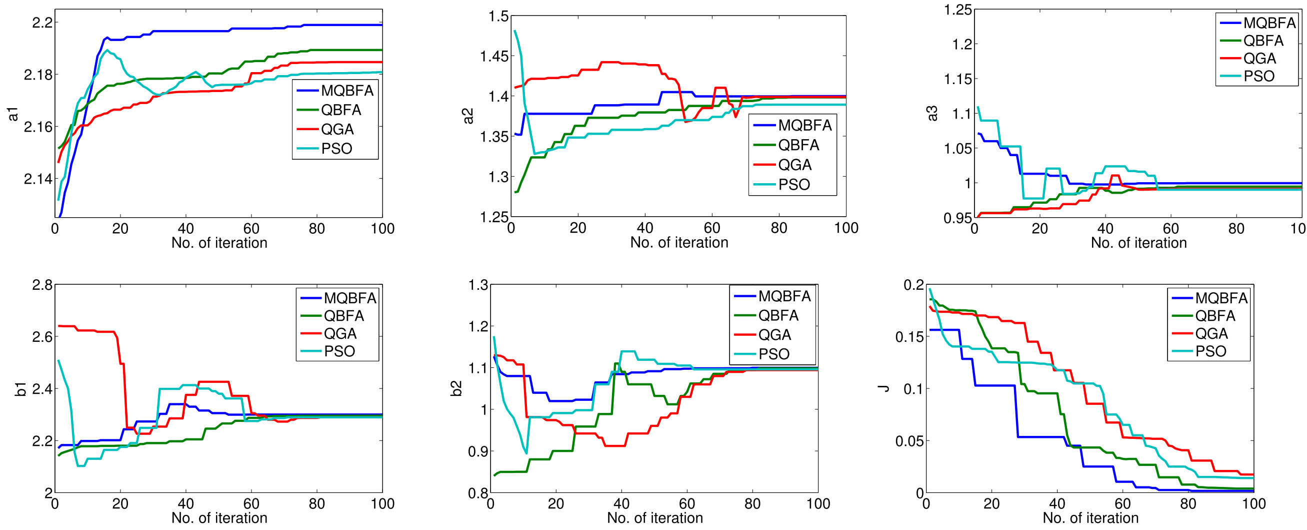

FIGURE 2. Evolutionary curves of estimated parameters and adaptive function values.

TABLE 6. The model structure.

Model	Structure	Parameter
1	$\frac{1}{a_1 S^{b_1} + a_2}$	a_1, a_2, b_1
2	$\frac{1}{a_1 S^{b_1} + a_2 S^{b_2} + a_3}$	a_1, a_2, a_3, b_1, b_2
3	$\frac{1}{a_1 S^{b_1} + a_2 S^{b_2} + a_3 S^{b_3} + a_4}$	$a_1, a_2, a_3, a_4, b_1, b_2, b_3$
4	$\frac{1}{a_1 S^{b_1} + a_2 S^{b_2} + a_3 S^{b_3} + a_4 S^{b_4} + a_5}$	$a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4$

is set as [−0.5, 0.5], and other conditions remain unchanged. The statistical results are listed in Table 5.

From Table 5, it is worth noting that the identification effect can be improved significantly with the decrease of the random number range of order. The main reason is that the objective function value affected by the order changes exponentially, which indicates that there is a large change in the function value with a minor change of order. Thus, the appropriate reduction of the order helps shorten the convergence time of the algorithm and improve search ability.

2) IDENTIFICATION FOR THE UNKNOWN FRACTIONAL-ORDER MODEL STRUCTURE

In the model identification, the model structure is often unknown. When one assumed model structure matches the real model, the measured error will be smaller than that of other models. Usually, the high-order systems rarely occur in the actual system, and the higher order fractional-order system can be reduced to one of the models in Table 6. Therefore, the 4 models in Table 6 are selected for simulation study. The transfer function of the identified object is assumed to be

$$G(s) = \frac{1}{2.5S^{2.3} + 1.4S^{1.4} + 0.8S^{0.6} + 1}. \tag{23}$$

The statistical results are listed in Table 7.

Table 7 shows that MQBFA algorithm outperforms other three algorithms. Besides, we can find that the performance index J of model 3 is significantly smaller than that of other models, so model 3 is closest to the target model. Moreover, taking the J value of model 3 as the dividing point, the J value decreases first and then increases. Furthermore, the difficulty of parameters identification is greatly increased with the increase of the parameters of the system. We can also see that the identification accuracy of a_2, b_1 and b_2 obtained by MQBFA is not the best. We take the parameters obtained by MQBFA as an example to explain the above phenomena. a_2, a_4, b_1 and b_2 are used as independent variables respectively, and the values of independent variables in the test are based on the results obtained by the other three algorithms. The test results are listed in Table 8.

TABLE 7. The estimation results of various algorithms.

Algorithm	PSO	QGA	QBFA	MQBFA
1				
a_1	3.7152	3.6917	2.8572	2.8027
a_2	1.6837	1.5867	1.4831	1.4107
b_1	1.5571	1.4834	1.3725	1.2547
J	1.6844E+02	1.5482E+02	1.2841E+02	8.4417E+01
2				
a_1	3.6027	3.5913	3.4752	3.3472
a_2	1.3217	1.3475	1.3904	1.3947
a_3	1.0381	1.0427	1.0247	1.0934
b_1	2.3708	2.3473	2.3092	2.2579
b_2	0.9215	0.9087	0.8961	0.8566
J	6.5752E+01	9.4283E+01	7.4752E+01	3.5715E+01
3				
a_1	2.8993	2.8942	2.8875	2.8254
a_2	1.2204	1.1821	1.1758	1.1847
a_3	0.9182	0.8917	0.8834	0.8784
a_4	1.1395	1.1084	1.0529	1.0183
b_1	2.3575	2.2824	2.2627	2.2742
b_2	1.3773	1.3624	1.3652	1.3517
b_3	0.6725	0.6717	0.6635	0.6685
J	1.2574E+01	8.3571E+00	4.5724E+00	5.2714E-01
4				
a_1	0.2598	0.3061	0.3454	0.2917
a_2	2.0965	2.3328	2.27233	2.1847
a_3	1.1025	1.0708	1.1057	1.1383
a_4	1.2951	1.2148	1.1639	1.1328
a_5	1.3486	1.1249	1.1126	1.0725
b_1	2.4720	2.2427	2.2035	2.1518
b_2	2.6025	2.6883	2.6243	2.5085
b_3	1.6243	1.5929	1.5074	1.4934
b_4	0.8646	0.8343	0.8084	0.7857
J	1.1824E+02	8.3574E+01	3.7824E+01	8.9257E+00

In Table 8, the data of line 1 represents the results obtained by MQBFA. The values of a_2 obtained by the other three algorithms are used as independent variables, and other variables remain unchanged, and the performance index J are listed in lines 2-4. Similarly, the data of lines 5-13 represents

TABLE 8. Test results using a_2, a_4, b_1 and b_2 as independent variables.

	a_1	a_2	a_3	a_4	b_1	b_2	b_3	J
1	2.8254	1.1847	0.8784	1.0183	2.2742	1.3517	0.6685	0.5271
2	2.8254	1.1758	0.8784	1.0183	2.2742	1.3517	0.6685	0.6872
3	2.8254	1.1821	0.8784	1.0183	2.2742	1.3517	0.6685	0.5345
4	2.8254	1.2204	0.8784	1.0183	2.2742	1.3517	0.6685	0.4814
5	2.8254	1.1847	0.8784	1.0529	2.2742	1.3517	0.6685	1.4591
6	2.8254	1.1847	0.8784	1.1084	2.2742	1.3517	0.6685	8.7106
7	2.8254	1.1847	0.8784	1.1395	2.2742	1.3517	0.6685	15.1293
8	2.8254	1.1847	0.8784	1.0183	2.2627	1.3517	0.6685	0.6374
9	2.8254	1.1847	0.8784	1.0183	2.2824	1.3517	0.6685	0.4723
10	2.8254	1.1847	0.8784	1.0183	2.3575	1.3517	0.6685	1.0487
11	2.8254	1.1847	0.8784	1.0183	2.2742	1.3652	0.6685	0.5721
12	2.8254	1.1847	0.8784	1.0183	2.2742	1.3624	0.6685	0.5624
13	2.8254	1.1847	0.8784	1.0183	2.2742	1.3773	0.6685	0.5028

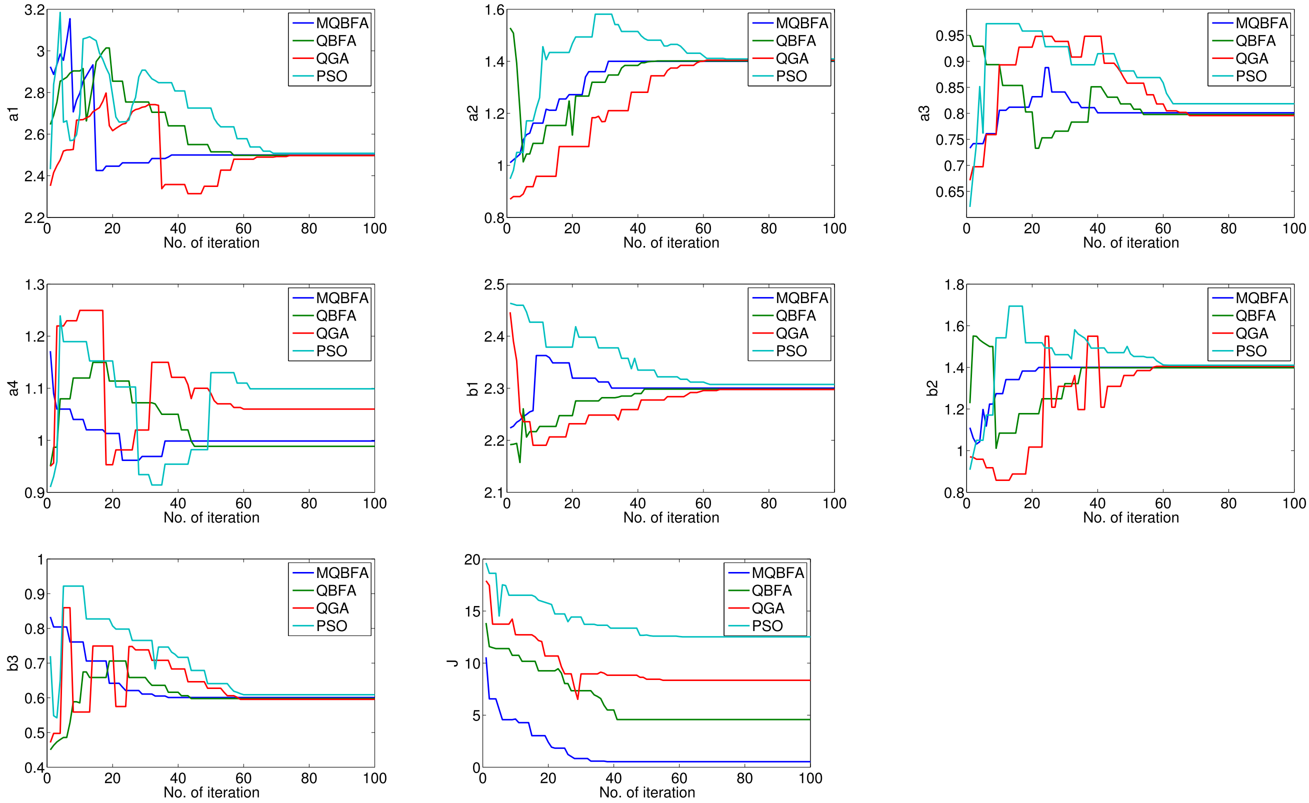

FIGURE 3. Evolutionary curves of estimated parameters and adaptive function values.

the results which are obtained by using a_4, b_1 and b_2 as independent variables respectively. Table 8 reveals that a small change of a_2, b_1 and b_2 has little effect on the performance index J, whereas a slight change of a_4 causes a significant change of the performance index J. In the early iterations of the MQBFA algorithm, J changes from big to small with the tuning of all the parameters. However, in the later iterations of the algorithm, when J becomes smaller mainly due to change of a_4, the algorithm is insensitive to minor changes of other parameters, and the algorithm tends to converge gradually. The evolutionary processes of estimated parameters in model 3 are shown in Fig. 3. Fig. 3 reveals that the proposed MQBFA algorithm has the fastest convergence speed.

VI. CONCLUSIONS

In this paper, based on a modified quantum bacterial foraging algorithm, a new parameter estimation scheme is proposed to identify the parameters of the fractional-order system. A new encoding mechanism is designed to improve the searching efficiency of the algorithm. Besides, a novel update strategy for the quantum rotation gate is presented to further strengthen the convergence rate of the algorithm. In addition, to enrich the diversity of population, a modificatory factor of probability amplitude is introduced in population initialization. Furthermore, the application of probability of optimal rotation angle enhances the searching ability of the algorithm. To verify the correctness of the MQBFA algorithm, seven benchmark functions are adopted to test the performance. In contrast to the other three algorithms, the test results demonstrate the correctness of the proposed scheme. In order to test the validity and convergence rate of the proposed algorithm, the MQBFA algorithm as well as other algorithms is used for parameters identification of a typical fractional-order system. Numerical simulations show that the proposed approach is an effective method with faster convergence rate and higher precision.

For a complex system with multiple parameters (more than 10), if we want to obtain a good performance, the time complexity will be increased due to the complex structure of MQBFA. So, in the future work, we will enhance the performance by improving the structure of algorithm. Furthermore, we will utilize the proposed algorithm to identify the parameters of the actual fractional-order model, such as the Cole impedance model.

REFERENCES

[1] A. Maachou *et al.*, "Thermal system identification using fractional models for high temperature levels around different operating points," *Nonlinear Dyn.*, vol. 70, no. 2, pp. 941–950, 2012.

[2] M. P. Aghababa, A. R. Haghighi, and M. Roohi, "Stabilisation of unknown fractional-order chaotic systems: An adaptive switching control strategy with application to power systems," *IET Generat. Transmiss. Distrib.*, vol. 9, no. 14, pp. 1883–1893, 2015.

[3] B. Xin and J. Zhang, "Finite-time stabilizing a fractional-order chaotic financial system with market confidence," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 1399–1409, 2015.

[4] S. Kassim *et al.*, "A novel secure image transmission scheme based on synchronization of fractional-order discrete-time hyperchaotic systems," *Nonlinear Dyn.*, vol. 88, no. 4, pp. 1–17, 2017.

[5] L. Yuan, Q. Yang, and C. Zeng, "Chaos detection and parameter identification in fractional-order chaotic systems with delay," *Nonlinear Dyn.*, vol. 73, nos. 1–2, pp. 439–448, 2013.

[6] D. Idiou, A. Charef, and A. Djouambi, "Linear fractional order system identification using adjustable fractional order differentiator," *IET Signal Process.*, vol. 8, no. 4, pp. 398–409, 2014.

[7] W. Hu, Y. Yu, and S. Zhang, "A hybrid artificial bee colony algorithm for parameter identification of uncertain fractional-order chaotic systems," *Nonlinear Dyn.*, vol. 82, no. 3, pp. 1–16, 2015.

[8] W. Du *et al.*, "Identification of fractional-order systems with unknown initial values and structure," *Phys. Lett. A*, vol. 381, no. 23, pp. 1943–1949, 2017.

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

[10] B. K. Panigrahi, V. R. Pandi, S. Das, and S. Das, "Multiobjective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem," *Energy*, vol. 35, no. 12, pp. 4761–4770, 2010.

[11] S. Das, I. Pan, S. Das, and A. Gupta, "Master-slave chaos synchronization via optimal fractional order $PI^{\lambda}D^{\mu}$ controller with bacterial foraging algorithm," *Nonlinear Dyn.*, vol. 69, no. 4, pp. 2193–2206, 2012.

[12] N. Rajasekar, N. K. Kumar, and R. Venugopalan, "Bacterial foraging algorithm based solar PV parameter estimation," *Solar Energy*, vol. 97, pp. 255–265, Nov. 2013.

[13] L. Mazouz, S. A. Zidi, S. Saadi, M. Khatir, and T. Benmessaoud, "VSC-HVDC system optimized PI controllers using bacterial foraging algorithm," *Elect. Eng.*, vol. 97, no. 3, pp. 205–212, 2015.

[14] J. Yi *et al.*, "Optimized relative transformation matrix using bacterial foraging algorithm for process fault detection," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2595–2605, Apr. 2016.

[15] O. P. Verma and A. S. Parihar, "An optimal fuzzy system for edge detection in color images using bacterial foraging algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 1, pp. 114–127, Jan. 2017.

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

[17] S. Dasgupta *et al.*, "Adaptive computational chemotaxis in bacterial foraging algorithm," in *Proc. Int. Conf. Complex, Intell. Softw. Intensive Syst.*, Barcelona, Spain, Mar. 2008, pp. 64–71.

[18] M. S. Li, T. Y. Ji, W. J. Tang, Q. H. Wu, and J. R. Saunders, "Bacterial foraging algorithm with varying population," *Biosystems*, vol. 100, no. 3, pp. 185–197, 2010.

[19] H. Chen *et al.*, "Bacterial colony foraging algorithm: Combining chemotaxis, cell-to-cell communication, and self-adaptive strategy," *Inf. Sci.*, vol. 273, no. 3, pp. 73–100, 2014.

[20] S. Huang and G. L. Zhao, "A comparison between quantum inspired bacterial foraging algorithm and GA-like algorithm for global optimization," *Int. J. Comput. Intell. Appl.*, vol. 11, no. 3, pp. 1969–2013, 2012.

[21] F. Li *et al.*, "Quantum bacterial foraging optimization algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, Jul. 2014, pp. 1265–1272.

[22] A. Diamantopoulos *et al.*, "A quantum bacterial foraging optimization algorithm and its application in spectrum sensing," *Int. J. Model. Identificat. Control*, vol. 18, no. 3, pp. 234–242, 2013.

[23] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, $H_{epsi;}$ gate, and two-phase scheme," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 156–169, Apr. 2004.

[24] X. W. Zhang, Y. Liang, *The Mathematical Basis of Genetic Algorithm*. Xi'an, China: Xi'an JiaoTong Univ., 2000, pp. 110–112.

[25] P. C. Li, "Quantum genetic algorithm based on Bloch coordinates of qubits and its application," *Control Theory Appl.*, vol. 25, no. 6, pp. 985–989, 2008.

[26] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Inf. Process. Lett.*, vol. 85, no. 6, pp. 317–325, 2003.

[27] B. Ross, *A Brief History and Exposition of the Fundamental Theory of Fractional Calculus* (Lecture Notes in Mathematics), vol. 457. New York, NY, USA: Springer-Verlag, 1975, pp. 1–36.

[28] T. T. Hartley and C. F. Lorenzo, "Fractional-order system identification based on continuous order-distributions," *Signal Process.*, vol. 83, no. 11, pp. 2287–2300, 2003.

LU LIU received the B.Sc. degree in electrical engineering and automation from the University of Jinan, China, in 2013. He is currently pursuing the Ph.D. degree in control science and engineering with the Nanjing University of Science and Technology. His research interests include servo control system and intelligence control algorithm.

LIANG SHAN received the B.Sc. degree in electrical engineering and the Ph.D. degree in control science and control engineering from the Nanjing University of Science and Technology, China, in 2002 and 2007, respectively. He is currently an Associate Professor with the Nanjing University of Science and Technology. His research interests include intelligence control algorithm, nonlinear system, and control methods of motor servo system.

YUEWEI DAI received the B.Sc. and M.Sc. degrees in system engineering from the East China Engineering Institute, China, in 1984 and 1987, respectively, and the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology, China, in 2002. He is currently a Professor with the Nanjing University of Science and Technology. His research interests include multimedia security, system engineering theory, and network security.

CHENGLIN LIU received the bachelor's degree in electrical engineering and automation from the Nanjing University of Science and Technology, Nanjing, China, in 2003, and the Ph.D. degree in control theory and control engineering from Southeast University, China. Since 2008, he has been with Jiangnan University, Wuxi, China, where he is currently a Professor with the Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, School of Internet of Things Engineering. His current research interests include cyber-physical systems, sensor networks, and coordination control of multi-agent systems.

ZHIDONG QI received the B.Sc. degree in industrial automation from North China Electric Power University, China, in 1999, and the M.Sc. degree in electrical engineering from Jiangsu University, China, in 2002, and the Ph.D. degree in control science and control engineering from Shanghai Jiao Tong University, China, in 2006. He is currently an Associate Professor with the Nanjing University of Science and Technology, China. His research interests include fractional control algorithm, nonlinear system, and modeling and control methods of fuel cell.

• • •