Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 620325, 16 pages
http://dx.doi.org/10.1155/2014/620325

Research Article

Quantum-Inspired Evolutionary Algorithm for Continuous Space Optimization Based on Multiple Chains Encoding Method of Quantum Bits

Rui Zhang, Zhiteng Wang, and Hongjun Zhang

PLA University of Science and Technology, Nanjing 210007, China

Correspondence should be addressed to Zhiteng Wang; zhitengwang@sina.com

Received 3 April 2014; Accepted 14 June 2014; Published 23 July 2014

Academic Editor: Jingjing Zhou

This study proposes a novel quantum evolutionary algorithm called four-chain quantum-inspired evolutionary algorithm (FCQIEA) based on the four gene chains encoding method. In FCQIEA, a chromosome comprises four gene chains to expand the search space effectively and promote the evolutionary rate. Different parameters, including rotational angle and mutation probability, have been analyzed for better optimization. Performance comparison with other quantum-inspired evolutionary algorithms (QIEAs), evolutionary algorithms, and different chains of QIEA demonstrates the effectiveness and efficiency of FCQIEA.

1. Introduction

Quantum computation is based on the principal concepts of the quantum theory [1, 2]. Numerous researchers have devoted increasing interests to quantum computation, a novel interdisciplinary field that covers quantum mechanics and information science. Quantum-inspired evolutionary algorithms (QIEA) focus on generating new evolutionary algorithms (EA) in accordance with some concepts and principles of quantum computation, such as standing waves [3]. In QIEA, the quantum bits (qubits) that encode chromosomes are used and quantum gate (Q-gate) has been proposed as a tool that can be used to update the chromosome to achieve evolutionary search. The binary observation QIEA (bQIEA), based on a probability optimization algorithm according to quantum computation concept and theory, was initially proposed in 2002 by Han and Kim to solve combinatorial optimization problems [4]. Since then, variants of the bQIEA have been developed, including bQIEA with different operators, such as crossover and mutation operators [5–8], bQIEA with a novel update method for Q-gates [9–12], and hybrid bQIEA that focuses on the interactions between bQIEA and common genetic algorithm (CGA), immune algorithms, and particle swarm optimization (PSO) [12–31]. The combination of quantum computation and genetic algorithm has been widely studied [16, 32–43]. QIEA has also been proposed as a method that can be applied to solve some combinatorial optimization problems [44–66], such as traveling salesman problem [32], knapsack problem [67], and filter design [68], as well as design optimizations of electromagnetic devices [69] and analog test point selection [70]. QIEA has numerous advantages, such as the use of a small population size, fast convergence, and an excellent global-search capability compared with conventional EAs.

However, several studies have demonstrated that bQIEA is not suitable for numerical optimization problems [16, 71], such as function extremum optimization. Several problems have caused bQIEA to have slow converging speed and prematurity. For instance, frequent binary encoding and quantum chromosome decoding cause randomness and blindness. The magnitude and the direction of the rotational angle of quantum rotation gates cannot be determined during optimization. To solve global numerical optimization problems with continuous variables, studies have proposed a real-observation QIEA (rQIEA) [72, 73], which is characterized by the modified Q-bit representation to represent a Q-bit

Table 1: Query direction table of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$.

$\Delta\beta$	$A \neq 0$	$-\operatorname{sgn}(A)\lvert\Delta\beta\rvert$
	$A = 0$	0
$\Delta\varphi$	$B \neq 0$	$-\operatorname{sgn}(B)\lvert\Delta\varphi\rvert$
	$B = 0$	0
$\Delta\theta$	$C \neq 0$	$-\operatorname{sgn}(C)\lvert\Delta\theta\rvert$
	$C = 0$	0

individual, the use of real observation to generate real-valued solutions from Q-bit individuals, and the modified Q-gate to guide the individuals toward better solutions. S. Li and P. Li [74] further proposed a convenient method to determine the direction of the rotational angle; in particular, a real qubits encoding method can be used to avoid binary coding and the double-chains quantum evolutionary algorithm, which is characterized by each qubit that is regarded as two coordinate genes. Each chromosome contains two gene chains, and each gene chain represents an optimization result. P. Li and S. Li [75] studied the appropriate rotational angle size between two rotational angles to obtain the best optimization effects and proposed the three-chain quantum evolutionary algorithm, which is characterized by bloch coordinates as the gene chains, such that each chromosome has three gene chains.

The optimized effects of three-chain quantum evolutionary algorithm is more efficient than the double-chains quantum evolutionary algorithm in solving continuous space optimization problems because the three-chain quantum evolutionary algorithm can accelerate the evolution speed and expand the search space of QIEA by increasing the number of gene chains. The current study presents a novel QIEA called four-chain quantum-inspired evolutionary algorithm (FCQIEA) based on expanded chains encoding to improve the QIEA performance further. FCQIEA integrates the characteristics of the double-chains encoding method and the three-chain encoding method to describe each qubit as four paratactic genes. Our results demonstrate that FCQIEA is more effective and feasible based on the optimization experiments that involve typical function extremum. We also compared our results with common quantum-inspired evolutionary algorithm (CQIEA), double-chains quantum-inspired evolutionary algorithm (DCQIEA), and three-chain quantum-inspired evolutionary algorithm (TCQIEA). In addition, we compared the performance of the proposed FCQIEA with other QIEAs and other EAs.

2. The Principle of FCQIEA

2.1. General Description of Continuous Optimization. In general, a continuous space optimization problem can be described as follows:

$$\begin{aligned} &\min \quad f(x) = f(x_1, x_2, \ldots, x_n) \\ &\text{s.t.} \quad a_i \le x_i \le b_i; \quad i = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

In (1), n-dimensional continuous space optimization solutions are regarded as points or vectors. a_i and b_i are the lower and the upper bounds for the design variable x_i, respectively. Hence, restrained conditions are bounded as closed set Ω in n-dimension continuous space, and all points in Ω are regarded as approximate solutions.

2.2. Expanded Encoding Method for Quantum Chromosome. In quantum computation, the basic unit of information is described by a qubit, which can be expressed as follows:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2}$$

where α and β satisfy the following normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1. \tag{3}$$

A pair of complex numbers α and β in (2) and (3) is called qubit probability amplitude. Therefore, the qubit can also be described by using the probability amplitude as $[\alpha, \beta]^T$. According to the nature of the probability amplitude, a qubit can be expressed as in Figure 1.

Thus, $\alpha = \cos\theta$ and $\beta = \sin\theta$; then, qubit $|\varphi\rangle$ can be expressed as follows:

$$|\varphi\rangle = [\cos\theta, \sin\theta]^T. \tag{4}$$

Qubits are directly regarded as genes on a chromosome; thus, we can use (4) as the encoding method. The principle of the double-chains encoding method [43] is applied and described as follows:

$$\begin{vmatrix} \cos\theta_{i1} \\ \sin\theta_{i1} \end{vmatrix} \begin{vmatrix} \cos\theta_{i2} \\ \sin\theta_{i2} \end{vmatrix} \begin{vmatrix} \cdots \\ \cdots \end{vmatrix} \begin{vmatrix} \cos\theta_{ij} \\ \sin\theta_{ij} \end{vmatrix}, \tag{5}$$

where $\theta_{ij} = 2\pi \times \text{rand}$, rand $\in [0, 1]$, and rand is a random number from 0 to 1; $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, \ldots, n$; m describes the population size; n is the number of qubits.

We can consider points $[0, 0.5]$ and $[0, -0.05]$ as the center of two semicircles, such as in the Chinese Taiji (Figure 2).

Figure 2 shows that the vector of qubit $|\varphi\rangle$ intersects the semicircle at point B, which can be connected to point C. We can draw the vertical line segment BD from point B to line segment AC. We can infer that angle $\angle ABC = \pi/2$ forms a semicircle and angle $\angle ACB = \theta$. Thus, the length of the line segment $AB = AC \times \sin\theta = \sin\theta$. Similarly, $CB = AC \times \cos\theta = \cos\theta$. Thus, we can obtain the following equations:

$$\begin{aligned} BD &= BC \times \sin\theta = \cos\theta \times \sin\theta, \\ CD &= BC \times \cos\theta = \cos\theta \times \cos\theta. \end{aligned} \tag{6}$$

Similarly, we obtain the following:

$$\begin{aligned} AD &= AB \times \sin\theta = \sin\theta \times \sin\theta, \\ CD &= AB \times \cos\theta = \sin\theta \times \cos\theta. \end{aligned} \tag{7}$$

Equations (6) and (7) show that each vector in a qubit can be a hypotenuse in a right triangle so that it can be separated into two vectors, in which the sum of squares is equal to the square of the vector. Hence, we can determine vector β as follows:

$$\beta^2 = \sin\theta^2 = (\sin\theta\cos\theta)^2 + (\sin\theta\sin\theta)^2. \tag{8}$$

Table 2: Comparison of the optimization results on function extremum (10 optimizations).

Algorithm	Goldstein-Price function					Shubert function				
	Optimal result	Worst result	Average result	Convergence times	Average times (s)	Optimal result	Worst result	Average result	Convergence result	Average times (s)
CQIEA	3.1312	3.2852	3.2142	0	5.0205	9.9828	9.8937	9.9270	0	4.8763
DCQIEA	3.0047	3.1360	3.0418	4	1.3126	9.9980	9.9378	9.9739	1	0.7487
TCQIEA	3.0070	3.0152	3.0070	10	2.2693	9.9997	9.9643	9.9916	6	1.0335
FCQIEA	3.0000	3.0059	3.0016	10	2.24042	10	10	10	10	1.1191

Table 3: Benchmark continuous functions.

Function	Function expression	Solution space	$F_{\min}$
Ackley	$F_{\text{Ack}} = 20 + e - 20e^{-0.2\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}} - e^{(1/n)\sum_{i=1}^{n}\cos 2\pi x_i}$	$x_i \in (-32.768, 32.768)$	0
Griewank	$F_{\text{Gri}} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}} + 1$	$x_i \in (-600, 600)$	0
Rastrigin	$F_{\text{Ras}} = \sum_{i=1}^{n}\left[x_i^2 - 10\cos 2\pi x_i + 10\right]$	$x_i \in (-5.12, 5.12)$	0
Schwefel	$F_{\text{Sch}} = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$x_i \in (-10, 10)$	0
Rosenbrock	$F_{\text{Ros}} = \sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^2\right) + \left(x_i - 1\right)^2\right]$	$x_i \in (-30, 30)$	0

Table 4: Comparisons of bQIEA, rQIEA, and FCQIEA using five functions.

Problems		F1	F2	F3	F4	F5
NoFE		$1.5e+5$	$2.0e+5$	$5.0e+5$	$7.0e+5$	$2.0e+6$
bQIEA	Mean	$2.6e-3$	$3.5e-2$	$3.9e-2$	**$3.9e-4$**	$1.3e+1$
	Std.	$8.4e-4$	$3.3e-2$	$1.8e-1$	**$3.0e-9$**	$1.8e+1$
rQIEA	Mean	$2.4e-4$	$1.8e-15$	$1.5e-15$	$3.2e+3$	$1.3e-3$
	Std.	$1.0e-8$	$1.1e-14$	$7.4e-15$	$5.2e+2$	$3.5e+0$
FCQIEA	Mean	**$2.5e-9$**	**$1.9e-15$**	**$1.7e-15$**	$3.6e+3$	**$1.6e-3$**
	Std.	**$1.1e-8$**	**$1.4e-14$**	**$7.4e-15$**	$5.3e+2$	**$3.8e-4$**

Equation (8) is consistent with the following normalization condition and thus is equivalent to

$$\beta = [\cos\theta \times \sin\theta, \sin\theta \times \sin\theta]^T, \quad a = [\cos\theta]^T. \tag{9}$$

We can also substitute (4) in (10) as follows:

$$|\varphi\rangle = [\cos\theta, \sin\theta]^T = [\cos\theta, \cos\theta\sin\theta, \sin\theta\sin\theta]^T. \tag{10}$$

To describe the quantum dynamics behavior objectively, comprehensively, and unambiguously, we can use a new angle φ $(0 < \varphi < \pi)$, which is called "supporting role," to replace θ and obtain vector $\sin\theta$ as follows:

$$[\sin\varphi \times \sin\theta, \cos\varphi \times \sin\theta, \cos\theta]^T. \tag{11}$$

Equation (11) also satisfies the normalization condition; hence, (11) is correct. In fact, (11) also corresponds to the three-chain encoding method [75]:

$$\begin{vmatrix} \sin\varphi_{i1}\sin\theta_{i1} \\ \cos\varphi_{i1}\sin\theta_{i1} \\ \cos\theta_{i1} \end{vmatrix} \begin{matrix} \cdots \\ \cdots \\ \cdots \end{matrix} \begin{vmatrix} \sin\varphi_{in}\sin\theta_{ij} \\ \cos\varphi_{in}\sin\theta_{ij} \\ \cos\theta_{ij} \end{vmatrix}. \tag{12}$$

Likewise, we can obtain vector $\sin\varphi\sin\theta$ by adding the "supporting role" β to form the four-chain encoding method as follows:

$$[\cos\beta\cos\varphi \times \sin\theta, \sin\beta\sin\varphi \times \sin\theta, \cos\varphi\sin\theta, \cos\theta]^T,$$

$$\begin{vmatrix} \cos\beta_{i1}\sin\varphi_{i1}\sin\theta_{i1} \\ \sin\beta_{i1}\sin\varphi_{i1}\sin\theta_{i1} \\ \cos\varphi_{i1}\sin\theta_{i1} \\ \cos\theta_{i1} \end{vmatrix} \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} \begin{vmatrix} \cos\beta_{ij}\sin\varphi_{ij}\sin\theta_{ij} \\ \sin\beta_{ij}\sin\varphi_{ij}\sin\theta_{ij} \\ \cos\varphi_{ij}\sin\theta_{ij} \\ \cos\theta_{ij} \end{vmatrix}. \tag{13}$$

We can obtain four optimal solutions, which are expressed below:

$$\begin{aligned} p_{i1} &= \left(\cos\beta_{i1}\sin\varphi_{i1}\sin\theta_{i1}, \ldots, \cos\beta_{in}\sin\varphi_{in}\sin\theta_{in}\right), \\ p_{i2} &= \left(\sin\beta_{i1}\sin\varphi_{i1}\sin\theta_{i1}, \ldots, \sin\beta_{in}\sin\varphi_{in}\sin\theta_{in}\right), \\ p_{i3} &= \left(\cos\varphi_{i1}\sin\theta_{i1}, \ldots, \cos\varphi_{in}\sin\theta_{in}\right), \\ p_{i4} &= \left(\cos\theta_{i1}, \ldots, \cos\theta_{in}\right). \end{aligned} \tag{14}$$

We define p_{i1}, p_{i2}, p_{i3}, and p_{i4} as solutions $X1$, $X2$, $X3$, and $X4$, respectively. The encoded chromosomal structure is shown in Figure 3.

Table 5: Comparison results for GA, PSO, ABC, and FCQIEA.

Fun.	Dim.	G	GA			PSO			ABC			FCQIEA		
			Suc. (%d)	Mean	Time (s)	Suc. (%d)	Mean	Time (s)	Suc. (%d)	Mean	Time (s)	Suc. (%d)	Mean	Time (s)
F1	5	100	15	0.5	0.01	24	0.09	0.04	69	0.05	0.05	98	0.001	0.3
		500	80	0.01	0.02	96	0.001	0.12	99	0.003	0.15	100	0	0.4
	30	500	0	40.13	—	0	2.63	—	0	2.25	—	95	0.003	6.22
		1000	0	30.46	—	0	2.58	—	0	2.25	—	100	0	6.25
F2	20	20	96	0.0234	0.35	97	0.030565	0.32	99	0.00087	0.4	100	0	0.26
	30	100	81	1.2342	0.45	98	0.011151	0.52	99	$2.87E-09$	0.7	100	0	2.16
F3	5	500	10	1.3212	0.04	45	0.93	0.05	100	0	0.6	100	0	0.26
		1000	10	0.7213	0.05	45	0.45	0.06	100	0	0.8	100	0	0.39
	50	500	0	49	—	0	41.73	—	0	46	—	100	0	9.58
		10000	0	40	—	0	31.43	—	0	25	—	100	0	9.82
F4	5	1000	70	0.6	0.4	87	0.02	0.18	87	0.03	0.31	100	0	0.04
		2000	53	0.34	0.5	78	0.011	0.19	91	0.02	0.41	100	0	0.06
	50	10000	0	10.45	—	0	3.21	—	0	3.16	—	100	0	11.80
		20000	0	49.67	—	0	4.83	—	0	4.98	—	100	0	17.46
F5	20	1000	0	24.78	—	0	12.26	—	8	0.01	2.12	100	0	1.95
		2000	0	28.42	—	0	11.14		10	0.01	3.28	100	0	2.48
	40	10000	0	51.21	—	0	45.32	—	0	44.21	—	71	0.02	342
		20000	0	49.38	—	0	42.17	—	0	43.71	—	79	0.02	425

```
Procedure update (p_i)
Begin
 J = 0
   While (J < N) Do
    Begin
    J = J + 1
    Determine Δβ, Δφ, and Δθ with the query direction (Table 1) and obtain q_ij' as follows:
    q_ij' = U(Δβ, Δφ, Δθ)q_ij
   End
   p_i = p_i'
End
```

Pseudocode 1

```
Procedure Mutate (p_i)
Begin
   If p_i is not the current optimum chromosome,
    Begin
    J = 0
    While (J < N)
    Begin
    J = J + 1
    Generate Random Number rnd in range 0 to 1
    If rnd < p_m, then obtain q_ij' from q_ij' = Vq_ij
    End
     p_i = p_i'
   End
End
```

Pseudocode 2

Figure 3 shows that each chromosome can be separated into four gene chains, and each gene chain can have infinite sets of qubits that greatly expand the number of the global optimal solution and significantly increase the probability to obtain the global optimal solution.

2.3. Solution Space Transformation. In the quantum evolution progress, all qubits have limited values within −1 to 1. Thus, we need to transform all qubit values from unit space $I^n = [-1, 1]^n$ to space Ω of the continuous optimization problem (1) by using linear transformation. Each gene value corresponds to an optimization variable in the solution space. If the jth qubit on chromosome p_i is $[x_{ij}^4, x_{ij}^3, x_{ij}^2, x_{ij}^1]$, then the

```
Begin
t = 0
(1) Initialize the colony Q(t) = (p_i1, p_i2, p_i3, p_i4).
(2) Construct X(t) = (X_i1^t, X_i2^t, X_i3^t, X_i4^t) by solution space transformation.
(3) Obtain the optimum solution of the current BestX among X(t) and the current optimum chromosome BestC among Q(t)
    by evaluating X(t).
(4) Store BestX into GX and BestC into GC.
     While (not termination-condition)
        Begin
        t = t + 1
(5) Calculate Q(t) by updating and mutating the states of Q(t − 1).
    Determine X(t) = (X_i1^t, X_i2^t, X_i3^t, X_i4^t) by solution space transformation.
(6) Obtain the optimum solution of the current BX among X(t) and the current optimum chromosome BC among Q(t)
     by evaluating X(t).
(7) If fit(BX) < fit(GX)
        BestX = GX; BestC = GC
        Otherwise
        GX = BestX; GC = BestC
        End
    End
End
```

(8) The procedure of FCQIEA can be summarized as follows:

Step 1. Initialize the population. Let the current generation $t = 0$; generate an initial population $Q(t) = \{q_1^t, q_2^t, \ldots, q_m^t\}$, which has m individual qubits. Set the magnitude of the rotational angle $|\Delta\beta| = \beta_0$, $|\Delta\varphi| = \varphi_0$, and $|\Delta\theta| = \theta_0$, respectively. Set p_m as the mutation and Max_gen as the maximum generation.

Step 2. Transform the solution space. Four approximate solutions in each chromosome are transformed from the unit space $I^n = [-1, 1]^n$ to the solution space Ω of the continuous optimization problem (1); thus, the set of approximate solution $X(t)$ can be obtained.

Step 3. Compute the fitness. By computing the fitness of $4m$ approximate solutions, obtain the best solution $BestX$ in the current solution and the best chromosome $BestC$ in the current chromosome. Store $BestX$ as the global optimum solution GX and store $BestC$ as the global optimum chromosome GC.

Step 4. Set $t = t + 1$. Update and mutate $Q(t - 1)$. Calculate the new population $Q(t)$.

Step 5. Transform the solution space again and obtain a set of approximate solution $X(t)$.

Step 6. By computing the fitness of $X(t)$, determine the current optimum solution $BestX$ and the current optimum. chromosome $BestC$ If $fit(BestX) < fit(GX)$, then update the current optimum solution $BestX = GX$; at the same time, update the current optimum chromosome $BestC = GC$ to avoid population degradation. Otherwise, let $GX = BestX$ and $GC = BestC$ so that the algorithm approaches the global optimum solution.

Step 7. If the algorithm does not converge and if $t < Max\_gen$, then go back to Step 4 until the algorithm becomes convergent or until $t > Max\_gen$.

Pseudocode 3

corresponding variables in the solution space are computed as follows:

$$\begin{aligned} X_{i1}^j &= 0.5 \times \left[b_j\left(1 + x_{ij}\right) + a_j\left(1 - x_{ij}\right)\right], \\ X_{i2}^j &= 0.5 \times \left[b_j\left(1 + x_{ij}\right) + a_j\left(1 - x_{ij}\right)\right], \\ X_{i3}^j &= 0.5 \times \left[b_j\left(1 + x_{ij}\right) + a_j\left(1 - x_{ij}\right)\right], \\ X_{i4}^j &= 0.5 \times \left[b_j\left(1 + x_{ij}\right) + a_j\left(1 - x_{ij}\right)\right], \end{aligned} \tag{15}$$

where $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, \ldots, n$. Thus, each chromosome maps out four approximate solutions of the optimization problem.

2.4. Quantum Chromosome Update. Considering that m quantum chromosomes are present in the colony and we can obtain $4m$ approximate solutions by solution space transformation, we can then compute the fitness of these approximate solutions and define the solution with the maximum fitness as the current optimum solution in the quantum evolution progress. The chromosome corresponds to the current optimum solution called the optimum chromosome. By computing the fitness, we can obtain both optimum solution and optimum chromosome and subsequently update the colony by using the quantum rotation gate to obtain the optimal solution. In this updated process, the new optimum chromosome can be produced such that the colony can likely

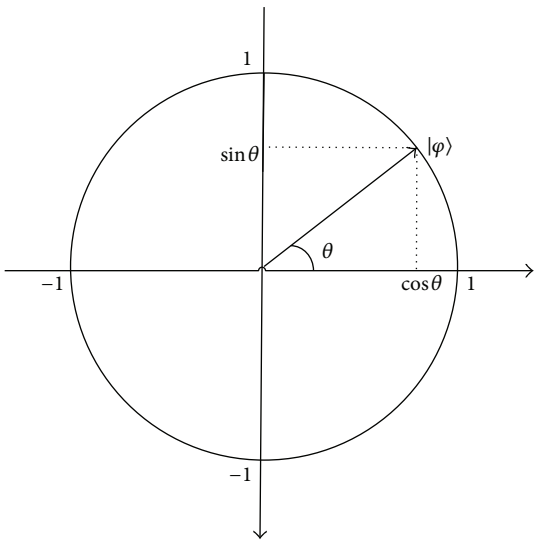

Figure 1: Schematic diagram of the qubit probability amplitude.

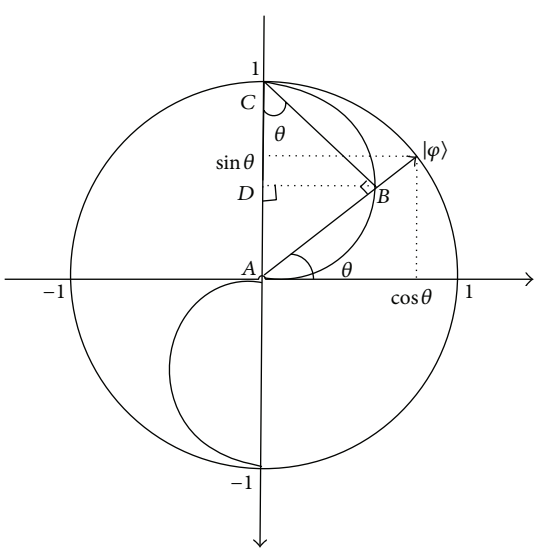

Figure 2: Schematic diagram of the qubit probability amplitude decomposition.

evolve. The present study proposes the quantum rotation gate U to update the individual qubit as follows:

$$
\begin{aligned}
U &= \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{bmatrix}, \\
u_{11} &= \left[\cos\Delta\beta\cos\Delta\varphi\left(\cos\Delta\theta - \frac{\sin\Delta\theta\cos\theta}{\sin\theta}\right)\right]^T, \\
u_{12} &= \left[\sin\Delta\beta\cos\Delta\varphi\left(\frac{\sin\Delta\theta\cos\theta}{\sin\theta} - \cos\Delta\theta\right)\right]^T, \\
u_{13} &= \left[\sin\Delta\varphi\cos\Delta\theta\left(\sin\beta\sin\Delta\beta - \cos\beta\cos\Delta\beta\right)\right]^T, \\
u_{14} &= \left[\cos\varphi\sin\Delta\varphi\cos\Delta\theta(\cos\beta\cos\Delta\beta - \sin\beta\sin\Delta\beta)\right]^T, \\
u_{21} &= \left[-\sin\Delta\beta\sin\Delta\varphi\left(\cos\Delta\theta + \frac{\sin\Delta\theta\cos\theta}{\sin\theta}\right)\right]^T, \\
u_{22} &= \left[-\cos\Delta\beta\sin\Delta\varphi\left(\cos\Delta\theta + \frac{\sin\Delta\theta\cos\theta}{\sin\theta}\right)\right]^T, \\
u_{23} &= \left[\cos\Delta\varphi\cos\Delta\varphi\left(\sin\beta\cos\Delta\beta + \cos\beta\sin\Delta\beta\right)\right]^T, \\
u_{24} &= \left[\cos\varphi\cos\Delta\varphi\cos\Delta\varphi(\sin\beta\cos\Delta\beta + \cos\beta\sin\Delta\beta)\right]^T, \\
u_{31} &= -\frac{\sin\Delta\varphi}{\cos\beta}, \qquad u_{32} = 0, \\
u_{33} &= \cos\Delta\varphi\cos\Delta\theta, \\
u_{34} &= \left[\sin\Delta\theta\left(\cos\varphi\cos\Delta\varphi - \sin\varphi\sin\Delta\varphi\right)\right]^T, \\
u_{41} &= u_{42} = 0, \qquad u_{43} = \frac{\sin\Delta\theta}{\cos\varphi}, \qquad u_{44} = \cos\Delta\theta, \\
U &\begin{bmatrix} \cos\beta\sin\varphi\sin\theta \\ \sin\beta\sin\varphi\sin\theta \\ \cos\varphi\sin\theta \\ \cos\theta \end{bmatrix} \\
&= \begin{bmatrix} \cos(\beta+\Delta\beta)\sin(\varphi+\Delta\varphi)\sin(\theta+\Delta\theta) \\ \sin(\beta+\Delta\beta)\sin(\varphi+\Delta\varphi)\sin(\theta+\Delta\theta) \\ \cos(\varphi+\Delta\varphi)\sin(\theta+\Delta\theta) \\ \cos(\theta+\Delta\theta) \end{bmatrix}.
\end{aligned}
\tag{16}
$$

These equations show that we can obtain the phase rotation of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$ from U. $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$ are important factors that directly influence the convergence speed and the efficiency of quantum evolution algorithm. In general, a query table is constructed [4] to determine the size and the direction of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$. In this way, all of the possible instances are considered when making a decision, but it is very complicated to lower the algorithm efficiency. To determine the rule that considers the size and the direction of the rotational angle by using a simple method, we can deduce the following conclusions.

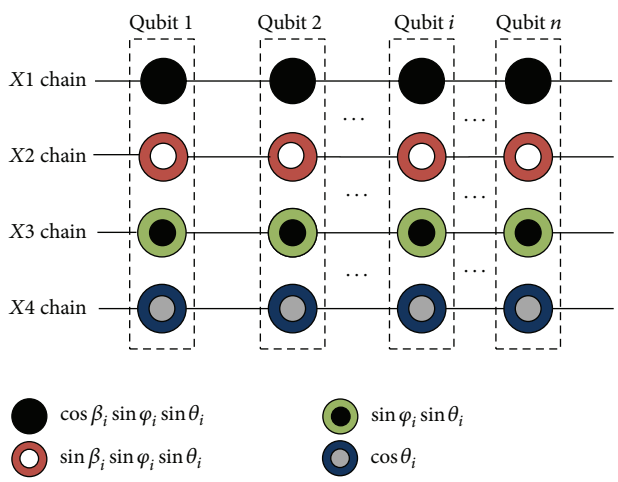

Figure 3: Structure of the quantum chromosome.

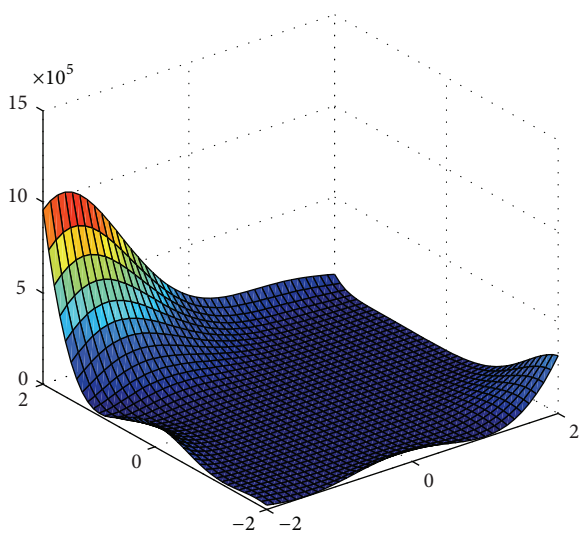

Figure 4: Goldstein-Price function.

Theorem 1. *Suppose $q_{0j}(x_{0j}^4, x_{0j}^3, x_{0j}^2, x_{0j}^1)$ is the jth qubit in the current optimum chromosome of the four gene chains encoding method; then, $q_{ij}(x_{ij}^4, x_{ij}^3, x_{ij}^2, x_{ij}^1)$ is the jth qubit in the ith chromosome, where $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, \ldots, n$.*

Let

$$A = \begin{vmatrix} x_{0j}^4 & x_{ij}^4 \\ x_{0j}^3 & x_{ij}^3 \end{vmatrix}, \tag{17}$$

$$B = \begin{vmatrix} x_{0j}^2 & x_{ij}^2 \end{vmatrix}, \tag{18}$$

$$C = \begin{vmatrix} x_{0j}^1 & x_{ij}^1 \end{vmatrix}. \tag{19}$$

(1) The direction of $\Delta\beta$ is determined by the following rules. If $A \neq 0$, then $\operatorname{sgn}(\Delta\beta) = -\operatorname{sgn}(A)$; if $A = 0$, then the direction of φ may be positive or negative at random.

(2) The direction of $\Delta\varphi$ is determined by the following rules. If $B \neq 0$, then $\operatorname{sgn}(\Delta\varphi) = -\operatorname{sgn}(B)$; if $B = 0$, then the direction of φ may be positive or negative at random.

(3) The direction of $\Delta\theta$ is determined by the following rules. If $C \neq 0$, then $\operatorname{sgn}(\Delta\theta) = -\operatorname{sgn}(C)$; if $C = 0$, then the direction of θ may be positive or negative at random.

Proof. Consider the following equations:

$$q_{0j} = \begin{bmatrix} \cos\beta_0 \sin\varphi_0 \sin\theta_0 \\ \sin\beta_0 \sin\varphi_0 \sin\theta_0 \\ \cos\varphi_0 \sin\theta_0 \\ \cos\theta_0 \end{bmatrix}, \qquad q_{ij} = \begin{bmatrix} \cos\beta \sin\varphi \sin\theta \\ \sin\beta \sin\varphi \sin\theta \\ \cos\varphi \sin\theta \\ \cos\theta \end{bmatrix}. \tag{20}$$

□

To determine the direction of $\Delta\beta$, A is expressed in the form of a trigonometric function as follows:

$$\begin{aligned} A &= \begin{bmatrix} \cos\beta_0 \sin\varphi_0 \sin\theta_0 & \cos\beta \sin\varphi \sin\theta \\ \sin\beta_0 \sin\varphi_0 \sin\theta_0 & \sin\beta \sin\varphi \sin\theta \end{bmatrix} \\ &= \sin\varphi \sin\theta \sin\varphi_0 \sin\theta_0 \sin(\beta - \beta_0). \end{aligned} \tag{21}$$

We can obtain $\sin\varphi \sin\theta \sin\varphi_0 \sin\theta_0 > 0$ from $\varphi_0 \in [0, \pi]$, $\varphi \in [0, \pi]$, $\theta_0 \in [0, \pi]$, and $\theta \in [0, \pi]$.

(1) If $A \neq 0$, then

$$\begin{aligned}\operatorname{sgn}(\Delta\beta) &= \operatorname{sgn}(\beta - \beta_0) \\ &= -\operatorname{sgn}(\sin\varphi \sin\theta \sin\varphi_0 \sin\theta_0 \sin(\beta - \beta_0)) \\ &= -\operatorname{sgn}(A).\end{aligned} \tag{22}$$

If $A = 0$, then $\sin\varphi \sin\theta \sin\varphi_0 \sin\theta_0 = 0$ or $\sin(\beta - \beta_0) = 0$.

If $\sin\varphi \sin\theta \sin\varphi_0 \sin\theta_0 = 0$, then at least one angle is zero; β and β_0 can have arbitrary values. Therefore, the direction of $\operatorname{sgn}(\Delta\beta)$ is arbitrary. If $\sin(\beta - \beta_0) = 0$, then $\beta = \beta_0$ or $|\beta - \beta_0| = \pi$; therefore, both positive and negative directions have the same result. Thus, the direction of $\operatorname{sgn}(\Delta\beta)$ is arbitrary when $A = 0$.

(2) If $B \neq 0$, then

$$\begin{aligned}\operatorname{sgn}(\Delta\varphi) &= \operatorname{sgn}(\varphi - \varphi_0) \\ &= -\operatorname{sgn}(\cos\varphi_0 \sin\theta_0 - \cos\varphi \sin\theta) \\ &= -\operatorname{sgn}\left(x_{0j}^2 - x_{ij}^2\right) \\ &= -\operatorname{sgn}(B).\end{aligned} \tag{23}$$

If $B = 0$, then $\varphi_0 = \varphi$ or $|\varphi - \varphi_0| = \pi$; both positive and negative directions have the same result. Therefore, the direction of $\operatorname{sgn}(\Delta\varphi)$ is arbitrary.

(3) If $C \neq 0$, then

$$\begin{aligned}\operatorname{sgn}(\Delta\theta) &= \operatorname{sgn}(\theta - \theta_0) \\ &= -\operatorname{sgn}(\cos\theta_0 - \cos\theta) \\ &= -\operatorname{sgn}\left(x_{0j}^1 - x_{ij}^1\right) \\ &= -\operatorname{sgn}(C).\end{aligned} \tag{24}$$

If $C = 0$, then $\theta_0 = \theta$ or $|\theta - \theta_0| = \pi$; both positive and negative directions have the same result. Therefore, the direction of $\operatorname{sgn}(\Delta\theta)$ is arbitrary.

The magnitudes of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$ have an important effect on the convergence speed. A decreased efficiency is observed in the optimization process when the magnitudes of $\Delta\theta$, $\Delta\varphi$, and $\Delta\theta$ are very small. By contrast, a premature convergence likely occurs in the optimization process when the magnitudes of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$ are very high.

The values from 0.005π to 0.05π are recommended for the magnitudes of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$, although they depend on the problems [75].

Based on our analysis, a simplified query table (Table 1) is constructed to determine the direction of $\Delta\beta$, $\Delta\varphi$, and $\Delta\theta$.

Let chromosome p_i $(i = 1, 2, \ldots, m)$ be q_{ij} $(j = 1, 2, \ldots, n)$. Based on Table 1, the update progress of p_i is described by a pseudocode as shown in Pseudocode 1.

2.5. Mutation Operation. Quantum nongate is applied to exchange the probability amplitudes to avoid local optimal solution in a certain qubit as follows:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \begin{bmatrix} \sin\theta \\ \cos\theta \end{bmatrix}. \tag{25}$$

Such influence as expressed in (18) can be considered as the phase mutation of a qubit, in which θ is mutated to $\pi/2 - \theta$. In this case, a quantum nongate V is proposed to mutate the quantum as follows:

$$V = \begin{bmatrix} \tan\beta \cot\varphi \cot\theta & 0 & 0 & 0 \\ 0 & \tan\beta \cot\varphi \cot\theta & 0 & 0 \\ 0 & 0 & \tan\varphi \cot\theta & 0 \\ 0 & 0 & 0 & \tan\theta \end{bmatrix},$$

$$V \begin{bmatrix} \cos\beta \sin\varphi \sin\theta \\ \sin\beta \sin\varphi \sin\theta \\ \cos\varphi \sin\theta \\ \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\pi}{2} - \beta\right) \sin\left(\frac{\pi}{2} - \varphi\right) \sin\left(\frac{\pi}{2} - \theta\right) \\ \sin\left(\frac{\pi}{2} - \beta\right) \sin\left(\frac{\pi}{2} - \varphi\right) \sin\left(\frac{\pi}{2} - \theta\right) \\ \cos\left(\frac{\pi}{2} - \varphi\right) \sin\left(\frac{\pi}{2} - \theta\right) \\ \cos\left(\frac{\pi}{2} - \theta\right) \end{bmatrix}. \tag{26}$$

The quantum nongate rotation can enhance population diversity. The mutation operation does not compare with the current optimum chromosome. The rotation magnitudes are usually great, which are $\pi/2 - 2\beta$, $\pi/2 - 2\varphi$, and $\pi/2 - 2\theta$.

Let the qubits in the chromosome p_i be q_{ij}, where $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, \ldots, n$; p_m is the mutation probability. Based on our analysis, the mutation progress p_i is described by another pseudocode as shown in Pseudocode 2.

3. Algorithm Description

The structure of FCQIEA is shown by the pseudocode under the section of the procedure for FCQIEA (see Pseudocode 3).

4. Convergence Analysis

Considering that the tth population is $Q(t) = \{q_1^t, q_2^t, \ldots, q_m^t\}$, the ith quantum chromosome q_i^t is defined as follows:

$$\begin{vmatrix} \cos\beta_{i1}^t \sin\varphi_{i1}^t \sin\theta_{i1}^t \\ \sin\beta_{i1}^t \sin\varphi_{i1}^t \sin\theta_{i1}^t \\ \cos\varphi_{i1}^t \sin\theta_{i1}^t \\ \cos\theta_{i1}^t \end{vmatrix} \begin{matrix} \cdots \\ \cdots \end{matrix} \begin{vmatrix} \cos\beta_{in}^t \sin\varphi_{in}^t \sin\theta_{in}^t \\ \sin\beta_{in}^t \sin\varphi_{in}^t \sin\theta_{in}^t \\ \cos\varphi_{in}^t \sin\theta_{in}^t \\ \cos\theta_{in}^t \end{vmatrix}. \tag{27}$$

Lemma 2. *The sequence of iteration in FCQIEA $\{Q_t, t \geq 0\}$ is a finite Markov chain.*

Proof. Given that the value of $Q(t)$ is continuous, the state space of the population is theoretically infinite, but Q_t is finite in accuracy in the progress of computation. Considering that the value of Q_t is v and v^{nm} represents the magnitude of the state space, the population becomes finite. The chromosome is then updated and mutated to show the relationship between Q_{t+1} and Q_t; thus, the iteration sequence of $\{Q_t, t \geq 0\}$ is a finite Markov chain (end). □

Lemma 3. *In the Markov chain sequence, which is generated by the iteration of FCQIEA, the sequence of the best fitness in the population* $fit(Q_t) = \max_{q_i^t \in Q_t}\{fit(q_i^t) : i = 1, 2, \ldots, n\}$ *is not continuously decreasing. Thus, the equation is always* $fit(Q_{t+1}) \geq fit(Q_t)$ *when* $t \geq 0$.

Proof. By the retention strategy of storing the best chromosome in FCQIEA, the population is not degenerated in each population generation. The population is expressed as $\text{fit}(Q_{t+1}) \geq \text{fit}(Q_t)$ when $t \geq 0$. □

Lemma 4. *FCQIEA is converged with probability 1.*

Proof. Assume that n qubits are present in each chromosome in a population with m chromosomes, which can be described as point $[-1, 1]^{nm}$ in state space. The state space of the population is defined as v^{nm} when the accuracy of Q_t is finite and the value is v. Consider $bq_k = \max_{q_i^t \in Q_t}\{\text{fit}(q_i^t) : i = 1, 2, \ldots, n\}$ as the best individual in Q_k from the kth population; then, $S^* = \{bq \mid \max_{1 \leq k \leq v^{nm}} \text{fit}(bq_k) = \text{fit}^*\}$ is defined as the global optimum solution set and fit^* as the global best fitness. Suppose $I = \{i \mid bq_i \cap s^* = \varphi\}$ and Q_t^i represents the ith state after t times iteration ($i = 1, 2, \ldots, v^{nm}$); then, the transition probability $P_i(i \to j) = P(Q_t^i \to Q_{t+1}^j)$ can be calculated in the random progress $\{Q_t, t \geq 0\}$ after further iteration from the ith state to the jth state.

(1) If $i \notin I$ and $j \in I$, by Lemma 3 and $\text{fit}(Q_{t+1}^j) \geq \text{fit}(Q_t^j)$, then $P_i(i \to j) = 0$.

(2) If $i \in I$ and $j \notin I$, by Lemma 3 and $\text{fit}(Q_{t+1}^j) \geq \text{fit}(Q_t^j)$, then $P_i(i \to j) \geq 0$.

Considering that $P_t(i)$ is the probability with the i state in Q_t population and $P_t = \sum_{i \in I} P_t$, the probability of Q_{t+1} with the state of $j \in I$ can be derived using the Markov chain as follows:

$$P_{t+1} = \sum_{i \in I}\sum_{j \in I} P_t(i) P_t(i \longrightarrow j) + \sum_{i \notin I}\sum_{j \in I} P_t(i) P_t(i \longrightarrow j),$$
$$P_t = \sum_{i \in I}\sum_{j \in I} P_t(i) P_t(i \longrightarrow j) + \sum_{i \in I}\sum_{j \notin I} P_t(i) P_t(i \longrightarrow j). \tag{28}$$

The probability of Q_{t+1} can be further derived as follows:

$$\sum_{i \in I}\sum_{j \in I} P_t(i) P_t(i \longrightarrow j) = P_t - \sum_{i \in I}\sum_{j \notin I} P_t(i) P_t(i \longrightarrow j),$$
$$0 \leq P_{t+1} = P_t - \sum_{i \in I}\sum_{j \notin I} P_t(i) P_t(i \longrightarrow j) + \sum_{i \notin I}\sum_{j \in I} P_t(i) P_t(i \longrightarrow j) \leq P_t. \tag{29}$$

Thus,

$$\lim_{t \to m} P_t = 0,$$
$$\lim_{t \to \infty} P(\text{fit}(bq_t) = \text{fit}^*) = 1 - \lim_{t \to \infty}\sum_{i \in I} P_i(i) = 1 - \lim_{t \to \infty} P_t = 1. \tag{30}$$

□

5. Experimental Result

We tested the FCQIEA performance by using different parameters to achieve the best performance for FCQIEA. Afterward, the performance of FCQIEA was compared with CQIEA, DCQIEA, and TCQIEA, as well as with bQIEA and rQIEA. The FCQIEA performance was also compared with the standard genetic algorithm (GA), particle swarm optimization (PSO), and artificial bee colony (ABC).

5.1. Parameter Analysis. In this section, different parameters, which include rotational angle and mutation probability, were analyzed by conducting a simulation test with two application examples of extremum optimizing complex functions (Goldstein-Price function and Shubert function) to achieve the best performance of FCQIEA.

(1) The Goldstein-Price function is expressed as follows:

$$f(x, y) = \left[1 + (x + y + 1)^2 \left(19 - 14 + 3x^2 - 14y + 6xy + 3y^2\right)\right] \times \left[30 + (2x - 3y)^2 \left(18 - 32x + 12x^2 + 48y - 36xy + 27y^2\right)\right], \tag{31}$$

where $|x| \leq 2$ and $|y| \leq 2$. The minimum local point is expressed as follows:

$(1.2, 0.8)$, $(1.2, 0.2)$, $(-0.6, -0.4)$, and $(0, -1)$.

The global minimum is 3. Optimum results <3.005 can be the result of the algorithm convergence. Goldstein-Price function is shown in Figure 4.

(2) The Shubert function is expressed as follows:

$$f(x, y) = 10\cos(2\pi x) + 10\sin(2\pi y) - x^2 - y^2 - 10, \tag{32}$$

where $x, y \in (-5.12, 5.12)$. This function is a typical multipeak function, in which the maximum extremum value

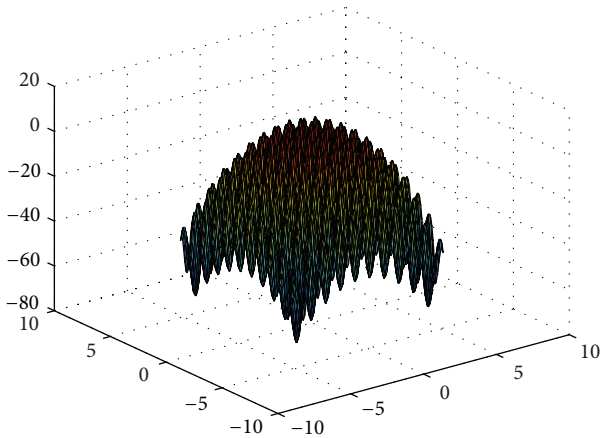

Figure 5: The Shubert function.

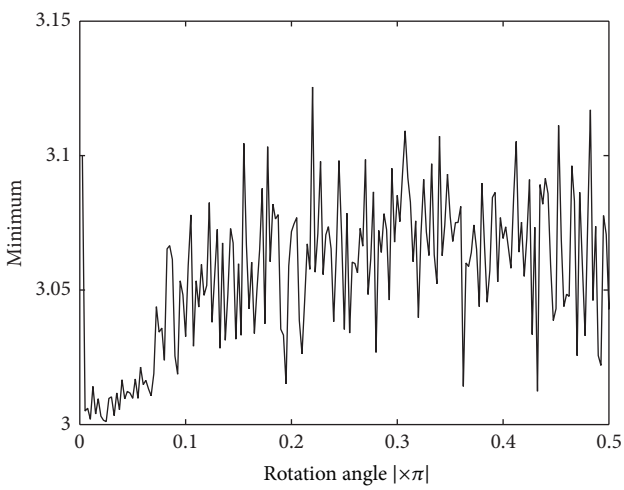

Figure 6: Effect of $|\Delta\beta|$, $|\Delta\varphi|$, and $|\Delta\theta|$ on the Goldstein-Price function optimization.

point is $(0, 0)$ that corresponds to the global optimum value of 10. When the optimum results reach 9.995, the algorithm is considered as convergent. Shubert function is shown in Figure 5.

The range of $|\Delta\beta|$, $|\Delta\varphi|$, and $|\Delta\theta|$ was studied by applying Goldstein-Price and Shubert functions. The population size, the maximum number of epochs, and the mutation probability are set at 10, 100, and 0, respectively.

The optimum result can be expressed as shown in Figures 6 and 7 when $|\Delta\beta| = |\Delta\varphi| = |\Delta\theta| = \{0.001\pi, 0.002\pi, \ldots, 0.5\pi\}$.

Figures 6 and 7 show that the optimization results are optimal when $0.005\pi < |\Delta\beta| = |\Delta\varphi| = |\Delta\theta| < 0.05\pi$.

The relationship between $|\Delta\beta|$, $|\Delta\varphi|$, and $|\Delta\theta|$ was also investigated. The population size, the maximum number of epochs, and the mutation probability are set at 10, 100, and 0, respectively.

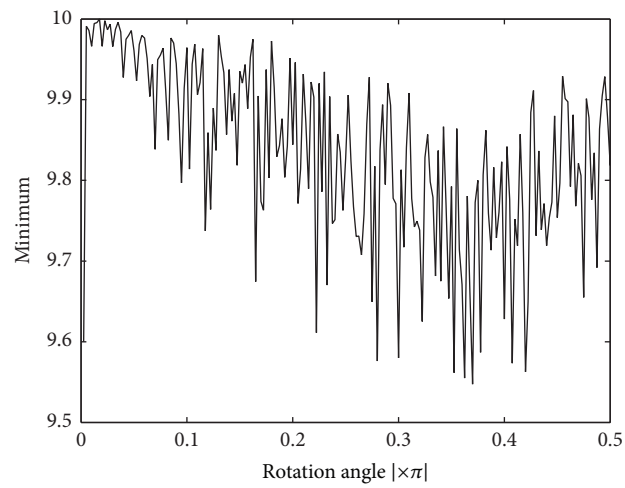

Figure 7: Effect of $|\Delta\beta|$, $|\Delta\varphi|$, and $|\Delta\theta|$ on the Shubert function optimization.

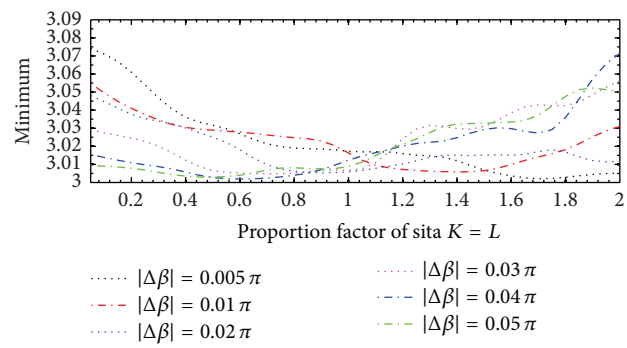

Figure 8: Effect of the proportion factors K and L on the Goldstein-Price function optimization.

Let

$$|\Delta\beta| = \{0.005\pi, 0.01\pi, 0.02\pi, 0.03\pi, 0.04\pi, 0.05\pi\},$$
$$|\Delta\beta| = K|\Delta\varphi|, \tag{33}$$
$$|\Delta\varphi| = L|\Delta\theta|.$$

The optimization results are shown in Figures 8 and 9 when $K = L = \{0.1, 0.2, 0.3, \ldots, 1.8, 1.9, 2.0\}$.

Figures 8 and 9 show that the value of $|\Delta\beta|$, K, and L considers the following six cases when we obtain the optimum result:

(1) $|\Delta\beta| = 0.005\pi$, $K = L \approx 1.8$;

(2) $|\Delta\beta| = 0.01\pi$, $K = L \approx 1.22$;

(3) $|\Delta\beta| = 0.02\pi$, $K = L \approx 0.86$;

(4) $|\Delta\beta| = 0.03\pi$, $K = L \approx 0.71$;

(5) $|\Delta\beta| = 0.04\pi$, $K = L \approx 0.61$;

(6) $|\Delta\beta| = 0.05\pi$, $K = L \approx 0.55$.

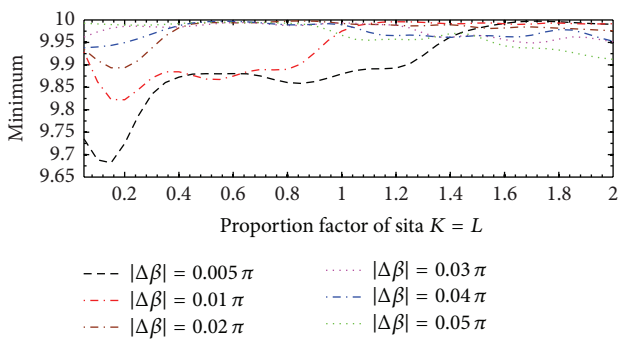

Figure 9: Effect of the proportion factors K and L on the Shubert function optimization.

Based on our analysis, the optimization result is optimal when $KL = 0.015\pi/|\Delta\beta|$. Thus, we can obtain

$$|\Delta\theta| = KL\left|\Delta\beta\right| \approx 0.015\pi,$$
$$|\Delta\theta| = L\left|\Delta\varphi\right| \approx 0.015\pi. \tag{34}$$

The influence of the mutation probability on the optimization performance was investigated. The chromosome size and the maximum number of epochs are set at 10 and 100, respectively. Let $K = L = 1$, such that $|\Delta\beta| = |\Delta\varphi| = |\Delta\theta|$.

The magnitude of the rotational angle $\Delta\theta$ and the mutation probability Pm is set as follows:

$$\Delta\theta = \{0.005\pi, 0.01\pi, 0.015\pi, 0.02\pi, 0.025\pi, 0.03\pi, 0.035\pi, 0.04\pi, 0.045\pi, 0.05\pi\},$$
$$P_m = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}. \tag{35}$$

To determine the relationship between mutation probability and rotational angle as well as to avoid the influence of random factors, we performed the calculation 10 times by using different mutation probabilities in each rotational angle from 0.005π to 0.05π and obtained the average value as the optimization result. The optimization result is shown in Figures 10 and 11.

Figures 10 and 11 show that the mutation operator can accelerate the optimization process when $0.05 < P_m < 0.70$. The optimal optimization effect can be obtained when $P_m = 0.05$. In general, the range of Pm can be selected from 0.01 to 0.5.

By synthesizing the description, the optimum parameters for FCQIEA can be expressed as follows:

(1) $0.005\pi < |\Delta\beta| < 0.05\pi$;

(2) $0.005\pi < |\Delta\varphi| < 0.05\pi$;

(3) $|\Delta\theta| = 0.015\pi$;

(4) $0.01 < P_m < 0.5$.

5.2. Comparison of the Different Chains of QIEA. In this section, we tested the performance of the proposed FCQIEA by using two application examples of the extremum optimizing complex functions (the Goldstein-Price and the Shubert functions). We also compared FCQIEA with CQIEA, DCQIEA, and TCQIEA in the experiments.

In the following section, we evaluated the optimization performance of FCQIEA and compared it with that of CQIEA, DCQIEA, and TCQIEA. In these three algorithms, the number of chromosomes, the mutation probability, and the maximum number of epochs are set at 20, 0.04, and 100, respectively. In FCQIEA, $|\Delta\beta|$, $|\Delta\varphi|$, and $|\Delta\theta|$ are 0.015π. In TCQIEA, $|\Delta\varphi|$ and $|\Delta\theta|$ are 0.01π [50]. In DCQIEA, $|\Delta\theta|$ is 0.01π. In CQIEA, each variable of the Shubert function is described by 25 binary digits, where the first bit represents the sign, the second to fifth bits represent the integral part of the variable, and the sixth to twenty-fifth bits represent the fraction part of the variable. In this case, each chromosome consists of 50 qubits. For the Goldstein-Price function, each variable is described by 18 binary digits, where the first bit represents the sign, the second and third bits represent the integral part of the variable, and the fourth to eighteenth bits represent the fraction part of the variable [50]. Therefore, each chromosome consists of 36 qubits. Two functions are optimized 10 times by FCQIEA, TCQIEA, DCQIEA, and CQIEA. The optimization results are shown in Figures 12 and 13 as well as in Table 2.

Table 2 and Figures 12 and 13 show that FCQIEA is the most efficient algorithm and the optimized result is considered optimal. TCQIEA and DCQIEA are also considered efficient algorithms. By contrast, our results revealed that CQIEA is the most inefficient of the three algorithms.

Table 2 and Figures 12 and 13 also show that FCQIEA is the most efficient and the optimized result is optimal. TCQIEA and DCQIEA are also considered efficient algorithms. By contrast, our results revealed that CQIEA is the most inefficient of the three algorithms. In FCQIEA, four-chain encoding method enhances the search capability compared with TCQIEA and DCQIEA. The application of mutation operator avoids running into the local minimum. The TCQIEA efficiency is lower than FCQIEA because FCQIEA have four gene chains, but TCQIEA and DCQIEA have less than three gene chains. For the same reason, the efficiency of DCQIEA is lower than that of TCQIEA. The running time of CQIEA is the greatest because of frequently encoding and decoding the binary number. Given that the optimization solution is obtained by probability, CQIEA is not the best choice to solve continuous optimization problems. Thus, we can improve the QEA efficiency by adding gene chains.

5.3. Comparison of Other QIEAs. Five benchmark numerical optimization problems, including Ackley (F1), Griewank (F2), Rastrigin (F3), Schwefel (F4), and Rosenbrock (F5), are described in Table 3.

The multimodal functions include Ackley, Griewank, Rastrigin, and Schwefel. The Ackley function has a surface with numerous local optima because of its exponential term. The variables of Griewank function exhibit interdependence because the function has a product term. Similarly, Rastrigin function has numerous local minima. The surface of Schwefel function is composed of numerous peaks and valleys. The

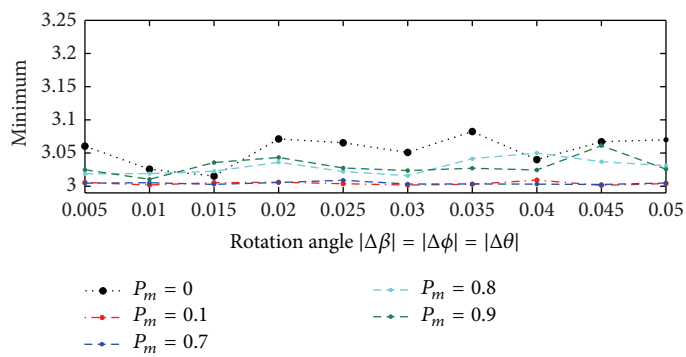

Figure 10: Effect of the mutation probability P_m on the Goldstein-Price function optimization.

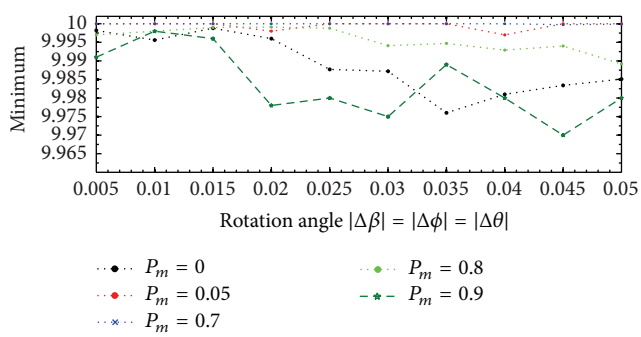

Figure 11: Effect of the mutation probability P_m on the Shubert function optimization.

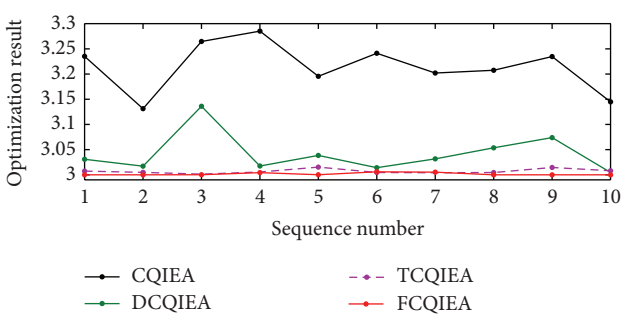

Figure 12: Comparison of the optimization results on the Goldstein-Price function (10 optimizations).

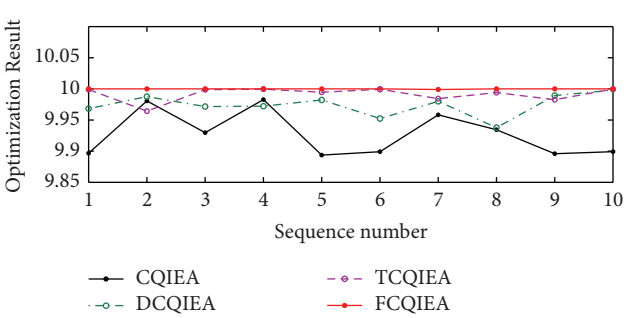

Figure 13: Comparison of the optimization results on the Shubert function (10 optimizations).

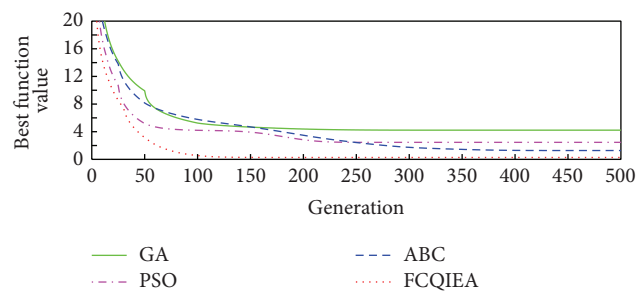

Figure 14: Comparison of the optimization results on the Rastrigin function.

best minimum among the functions is located far from the global minimum, which is found near the boundaries of the search domain [48, 76]. Rosenbrock is a continuous, convex, and unimodal function. Given that the global optimum is inside a long, narrow, parabolic-shaped flat valley and the variables are dependent, the gradients generally do not point toward the optimum and these gradients are difficult to converge at the global optimum. The functions are used to compare the different QIEA performances, including bQIEA and rQIEA, by using 30 dimensions. The population size is set at 50 for FCQIEA. The stopping criterion is the number of function evaluations. Each test function is performed with 50 independent runs. The mean best values and the standard deviations are recorded for each test function. Table 3 lists the simulation test results, in which FCQIEA provides better results than the two other versions of QIEA, which are used to search for optimal solutions and maintain robustness. The number of runs is 50. Mean, Std., and NoFE represent the best mean, the standard deviation, and the number of function evaluations, respectively. The bold style highlights the best result for each function (Table 4).

5.4. Comparison with Other EAs. In this paper, we tested the set of five widely used benchmark continuous functions (Table 3) with high complexity to compare the performance of the FCQIEA with other EAs. The FCQIEA algorithms are also compared with the standard GA, PSO, and ABC to search for the global minima in the solution space.

In the experiments, the population size of ABC, PSO, GA, and FCQIEA is set at 50 and the maximum number of function evaluations is set at 30,000; the other parameters used were in accordance with the parameter settings in a previous study [77]. More than 100 independent experiments were conducted for each function and performance indexes, such as the success ratio and the average minimum function value. The results were subsequently recorded. Table 5 shows the results of five test functions, in which the standard GA, PSO, ABC, and FCQIEA were used.

The Ackley function (F1) contains many peaks and valleys. Considering the five-dimensional Ackley function optimization, the global minimum was obtained by using all of the algorithms, but FCQIEA had the best performance among the four algorithms. FCQIEA can be used to find the global minimum with approximately 100% success rate in several hundredths of a second. We increased the dimension of the function to 30 and found that the global optimum was still obtained at a success rate of approximately 100% based on FCQIEA. By contrast, the other three algorithms failed to converge. Similar results were obtained after we tested the performance of the Rastrigin function (F3; Figure 14).

The Griewank function is a multimodal function (F2) with numerous regularly distributed local minima. The global minimum was satisfactorily obtained based on the four algorithms when the dimension was set at 20 and 30. Throughout the experiment, FCQIEA exhibited the best performance among the four algorithms. The performances of the four algorithms for the Schwefel function (F4) are similar to those in the Ackley function. The Rosenbrock function (F5) is a unimodal optimization problem, but it is very difficult to solve. Convergence is also difficult to achieve at the global optimum. Table 5 shows that both GA and PSO fail to find the global optimum. At the function dimension of 20, the optimum can be obtained via ABC but with a very low success ratio. However, the results of FCQIEA are very good, in which the global minimum can be rapidly determined with 100% success rate for dimension equal to 20. However, if the dimension was increased to 40, the success ratio decreased to 70% with an increased run time.

6. Conclusion

This paper presents the expanded encoding method for QIEA by determining the qubit vector to improve the stability and the global search ability for high-dimensional continuous space optimization. Based on the expanded encoding method, this paper presents the four-chain encoding method. Moreover, the optimum parameter, which considers the rotational angle and the mutation probability, is studied experimentally for FCQIEA. FCQIEA is shown as a robust algorithm that can be used to solve high-dimensional continuous space optimization problems compared with CQIEA, DCQIEA, and TCQIEA with extremum optimizing complex functions (the Goldstein-Price and the Shubert functions) as well as with the standard bQIEA, rQIEA, GA, PSO, and ABC for five widely used benchmark examples. The experimental results indicate that the expanded encoding method can improve the optimization efficiency of QIEA.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant no. 70791137).

References

[1] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 1st edition, 2000.

[3] M. Moore and A. Narayanan, "Quantum-inspired computing," Tech. Rep., Department of Computer Science, University Exeter, Exeter, UK, 1995.

[4] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, 2002.

[5] N. Li, P. Du, and H. Zhao, "Independent component analysis based on improved quantum genetic algorithm: application in hyperspectral images," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '05)*, pp. 4323–4326, July 2005.

[6] J. J. Xu, H. J. Chen, Y. H. Cheng, and R. Luo, "Blind signal separation based on quantum genetic algorithm," *Journal of Communication and Computer*, vol. 2, no. 9, pp. 62–66, 2005.

[7] S. Yang, M. Wang, and L. Jiao, "A novel quantum evolutionary algorithm and its application," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 1, pp. 820–826, June 2004.

[8] H. Talbi, M. Batouche, and A. Draa, "A quantum-inspired genetic algorithm for multi-source affine image registration," in *Image Analysis and Recognition*, vol. 3211 of *Lecture Notes in Computer Science*, pp. 147–154, Springer, Berlin, Germany, 2004.

[9] G. X. Zhang, N. Li, W. D. Jin, and L. Z. Hu, "Novel quantum genetic algorithm and its applications," *Frontiers of Electrical and Electronic Engineering in China*, vol. 1, no. 1, pp. 31–36, 2006.

[10] G. X. Zhang, L. Z. Hu, and W. D. Jin, "Quantum computing based machine learning method and its application in radar emitter signal recognition," in *Modeling Decisions for Artificial Intelligence*, vol. 3131 of *Lecture Notes in Artificial Intelligence*, pp. 92–103, Springer, Berlin, Germany, 2004.

[11] G. X. Zhang, L. Z. Hu, and W. D. Jin, "Resemblance coefficient and a quantum genetic algorithm for feature selection," in *Discovery Science*, vol. 3245 of *Lecture Notes in Computer Science*, pp. 155–168, Springer, Berlin, Germany, 2004.

[12] X. Wang, Y. Yang, and J. Xiao, "Application of quantum genetic algorithm in logistics distribution planning," in *Proceedings of the 26th Chinese Control Conference (CCC '07)*, pp. 759–762, Hunan, China, July 2007.

[13] Y. Li, L. Jiao, and F. Liu, "Self-adaptive chaos quantum clonal evolutionary programming," in *Proceedings of the International Conference on Signal Processing (ICSP '04)*, vol. 2, pp. 1550–1553, 2004.

[14] Y. Li, Y. N. Zhang, R. C. Zhao, and L. C. Jiao, "The immune quantum-inspired evolutionary algorithm," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (ICSMC '04)*, pp. 3301–3305, October 2004.

[15] Y. Li, Y. Zhang, Y. Cheng, X. Jiang, and R. Zhao, "A novel immune quantum-inspired genetic algorithm," in *Advances in Natural Computation*, vol. 3612 of *Lecture Notes in Computer Science*, pp. 215–218, Springer, Berlin, Germany, 2005.

[16] L. Wang, F. Tang, and H. Wu, "Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation," *Applied Mathematics and Computation*, vol. 171, no. 2, pp. 1141–1156, 2005.

[17] L. Wang, H. Wu, F. Tang, and D. Z. Zheng, "A hybrid quantum-inspired genetic algorithm for flow shop scheduling," in *Advances in Intelligent Computing*, vol. 3645 of *Lecture Notes in Computer Science*, pp. 636–644, Springer, Berlin, Germany, 2005.

[18] L. Wang, Q. Niu, and M. R. Fei, "A novel quantum ant colony optimization algorithm," in *Bio-Inspired Computational Intelligence and Applications*, vol. 4688 of *Lecture Notes in Computer Science*, pp. 277–286, 2007.

[19] Y. Wang, X. Y. Feng, Y. X. Huang et al., "A novel quantum swarm evolutionary algorithm and its applications," *Neurocomputing*, vol. 70, no. 4–6, pp. 633–640, 2007.

[20] X. You, Y. Liu, and D. Shuai, "On parallel immune quantum evolutionary algorithm based on learning mech-anism and its convergence," in *Advances in Natural Computation*, vol. 4221 of *Lecture Notes in Computer Science*, pp. 903–912, Springer, Berlin, Germany, 2006.

[21] X. You, D. Shuai, and S. Liu, "Research and implementation of quantum evolution algorithm based on immune theory," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, pp. 3410–3414, June 2006.

[22] X. You, S. Liu, and D. Shuai, "Studying the performance of quantum evolutionary algorithm based on immune theory," in *Computational Science—ICCS 2007*, vol. 4490 of *Lecture Notes in Computer Science*, pp. 1068–1075, 2007.

[23] X. M. You, S. Liu, and D. X. Shuai, "On improved parallel immune quantum evolutionary algorithm based on learning mechanism," in *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA '06)*, pp. 908–913, Jinan, China, 2006.

[24] Y. Li and L. Jiao, "Quantum-inspired immune clonal algorithm," in *Artificial Immune Systems*, vol. 3627 of *Lecture Notes in Computer Science*, pp. 304–317, Springer, Berlin, Germany, 2005.

[25] Y. Li and L. Jiao, "Quantum-inspired immune clonal multiobjective optimization algorithm. in," in *Advances in Knowledge Discovery and Data Mining*, vol. 4426 of *Lecture Notes in Artificial Intelligence*, pp. 672–679, 2007.

[26] Y. Li and F. Liu, "A novel immune clonal algorithm," in *Advances in Natural Computation*, vol. 4222 of *Lecture Notes in Computer Science*, pp. 31–40, 2006.

[27] L. Jiao and Y. Li, "Quantum-inspired immune clonal optimization," in *Proceedings of the International Conference on Neural Networks and Brain (ICNN&B '05)*, vol. 1, pp. 461–468, Beijing, China, October 2005.

[28] Y. Huang, C. Tang, and S. Wang, "Quantum-inspired swarm evolution algorithm," in *Proceedings of the International Conference on Computational Intelligence and Security Workshops (CISW '07)*, pp. 208–211, Harbin, China, December 2007.

[29] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum genetic optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 231–241, 2008.

[30] G. Pan, K. Xia, Y. Dong, and J. Shi, "An improved LS-SVM based on quantum PSO algorithm and its application," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, vol. 2, pp. 606–610, Haikou, China, August 2007.

[31] L. Wang and L. P. Li, "An effective hybrid quantum-inspired evolutionary algorithm for parameter estimation of chaotic systems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1279–1285, 2010.

[32] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 61–66, Nagoya, Japan, May 1996.

[33] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, Hε gate, and two-phase scheme," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 156–169, 2004.

[34] B. Li and L. Wang, "A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 37, no. 3, pp. 576–591, 2007.

[35] Y. J. Lv and N. X. Liu, "Application of quantum genetic algorithm on finding minimal reduct," in *Proceedings of the 2007 IEEE International Conference on Granular Computing (GrC '07)*, pp. 728–733, November 2007.

[36] Z. Zhao, S. Zheng, J. Shang, and X. Kong, "Study of cognitive radio decision engine based on quantum genetic algorithm," *Acta Physica Sinica*, vol. 56, no. 11, pp. 6760–6766, 2007 (Chinese).

[37] G. Zhang, Y. Gu, L. Hu, and W. Jin, "A novel genetic algorithm and its application todigital filter design," in *Proceedings of the IEEE Intelligent Transportation Systems*, vol. 2, pp. 1600–1605, Shanghai, China, 2003.

[38] H. Xing, L. Bai, and Y. Ji, "QoS multicast routing scheme using QGA in IP/DWDM networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 15, no. 4, pp. 95–100, 2008.

[39] J. Xiao, Y. Yan, J. Zhang, and Y. Tang, "A quantum-inspired genetic algorithm for k-means clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4966–4973, 2010.

[40] J. Gu, X. Gu, and M. Gu, "A novel parallel quantum genetic algorithm for stochastic job shop scheduling," *Journal of Mathematical Analysis and Applications*, vol. 355, no. 1, pp. 63–81, 2009.

[41] H. Xing, X. Liu, X. Jin, L. Bai, and Y. Ji, "A multi-granularity evolution based Quantum Genetic Algorithm for QoS multicast routing problem in WDM networks," *Computer Communications*, vol. 32, no. 2, pp. 386–393, 2009.

[42] J. G. Vlachogiannis and J. Østergaard, "Reactive power and voltage control based on general quantum genetic algorithms," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6118–6126, 2009.

[43] S. Zhao, G. Xu, T. Tao, and L. Liang, "Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks," *Computers and Mathematics with Applications*, vol. 57, no. 11-12, pp. 2009–2015, 2009.

[44] B. Akay and D. Karaboga, "A modified Artificial Bee Colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120–142, 2012.

[45] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, vol. 2, pp. 1769–1776, September 2005.

[46] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter evolution," *Evolutionary Computation Journal*, vol. 10, no. 4, pp. 371–395, 2002.

[47] W. Gao and S. Liu, "A modified artificial bee colony algorithm," *Computers and Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.

[48] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[49] N. Hansen, "Compilation of results on the CEC benchmark function set," Technical Report, Institute of Computational Science, ETH Zurich, Zürich, Switzerland, 2005, http://www.ntu.edu.sg/home/epnsugan/.

[50] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: operators and tools for the behavioral analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, 1998.

[51] W. B. Langdon and R. Poli, "Evolving problems to learn about particle swarm optimizers and other search algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 561–578, 2007.

[52] G. Li, P. Niu, and X. Xiao, "Development and investigation of efficient artificial bee colony algorithm for numerical function optimization," *Applied Soft Computing Journal*, vol. 12, no. 1, pp. 320–332, 2012.

[53] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[54] E. Mininno, F. Neri, F. Cupertino, and D. Naso, "Compact differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 32–54, 2011.

[55] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera, "Memetic algorithms for continuous optimisation based on local search chains," *Evolutionary Computation*, vol. 18, no. 1, pp. 27–63, 2010.

[56] D. Molina, M. Lozano, and F. Herrera, "MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization," in *Proceedings of the 6th IEEE World Congress on Computational Intelligence (WCCI '10), IEEE Congress on Evolutionary Computation (CEC '10)*, Barcelona, Spain, July 2010.

[57] M. A. Montes de Oca, T. Stützle, M. Birattari, and M. Dorigo, "Frankenstein's PSO: a composite particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, 2009.

[58] F. Neri and V. Tirronen, "Scale factor local search in differential evolution," *Memetic Computing*, vol. 1, no. 2, pp. 153–171, 2009.

[59] Q. H. Nguyen, Y.-S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.

[60] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 107–125, 2008.

[61] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution*, A Practical Approach to Global Optimization, Springer, Berlin, Germany, 2005.

[62] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[63] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[64] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep. Nanyang Technological University, Singapore and KanGAL Report 2005005, IIT Kanpur, Kanpur, India, 2005, http://www3.ntu.edu.sg/home/EPNSugan.

[65] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243–259, 2009.

[66] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[67] K. H. Han and J. H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 1354–1360, San Diego, Calif, USA, July 2000.

[68] G. X. Zhang, N. Li, W. D. Jin, and L. Z. Hu, "Novel quantum genetic algorithm and its application," *Acta Electronica Sinica*, vol. 32, no. 3, pp. 476–479, 2004.

[69] W. Zhang, H. Xu, Y. Bai, and S. Yang, "An quantum-inspired evolutionary algorithm applied to design optimizations of electromagnetic devices," *International Journal of Applied Electromagnetics and Mechanics*, vol. 39, no. 1–4, pp. 89–95, 2012.

[70] H. Lei and K. Qin, "Quantum-inspired evolutionary algorithm for analog test point selection," *Analog Integrated Circuits and Signal Processing*, vol. 75, no. 3, pp. 491–498, 2013.

[71] H. Qu, D. Zhao, and F. Zhou, "A new quantum clone evolutionary algorithm for multi-objective optimization," in *Proceedings of the 2008 International Seminar on Business and Information Management (ISBIM '08)*, pp. 23–25, December 2008.

[72] G. X. Zhang and H. N. Rong, "Real-observation quantum-inspired evolutionary algorithm for a class of numerical optimization problems," in *Computational Science-ICCS 2007*, vol. 4490 of *Lecture Notes in Computer Science*, pp. 989–996, 2007.

[73] M. Fiasché, "A quantum-inspired evolutionary algorithm for optimization numerical problems," in *Proceedings of the 19th International Conference (ICONIP '12), Part III*, T. Huang et al., Ed., vol. 7665 of *Lecture Notes in Computer Science*, pp. 686–693, Springer, 2012.

[74] S. Li and P. Li, "Quantum genetic algorithm based on real encoding and gradient information of object function," *Journal of Harbin Institute of Technology*, vol. 38, no. 8, pp. 1216–1223, 2006.

[75] P. Li and S. Li, "Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits," *Neurocomputing*, vol. 72, no. 1—3, pp. 581–585, 2008.

[76] O. B. Domingo, C. Hervás-Martínez, and N. García-Pedrajas, "Improving crossover operator for real-coded genetic algorithms using virtual parents," *Journal of Heuristics*, vol. 13, no. 3, pp. 265–314, 2007.

[77] D. Srinivasan and T. H. Seow, "Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, vol. 4, pp. 2292–2297, Canberra, Australia, December 2003.