Research Article
An Allele Real-Coded Quantum Evolutionary Algorithm Based on Hybrid Updating Strategy

Yu-Xian Zhang,¹ Xiao-Yi Qian,² Hui-Deng Peng,¹ and Jian-Hui Wang³

¹School of Electrical Engineering, Shenyang University of Technology, Shenyang 110870, China
²School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China
³College of Information Science and Engineering, Northeastern University, Shenyang 110004, China

Correspondence should be addressed to Yu-Xian Zhang; yuxian524524@163.com

Received 3 October 2015; Revised 16 December 2015; Accepted 17 December 2015

Academic Editor: José David Martín-Guerrero

Copyright © 2016 Yu-Xian Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For improving convergence rate and preventing prematurity in quantum evolutionary algorithm, an allele real-coded quantum evolutionary algorithm based on hybrid updating strategy is presented. The real variables are coded with probability superposition of allele. A hybrid updating strategy balancing the global search and local search is presented in which the superior allele is defined. On the basis of superior allele and inferior allele, a guided evolutionary process as well as updating allele with variable scale contraction is adopted. And $H_\varepsilon$ gate is introduced to prevent prematurity. Furthermore, the global convergence of proposed algorithm is proved by Markov chain. Finally, the proposed algorithm is compared with genetic algorithm, quantum evolutionary algorithm, and double chains quantum genetic algorithm in solving continuous optimization problem, and the experimental results verify the advantages on convergence rate and search accuracy.

1. Introduction

Quantum computing is a new class of computing algorithms based on the concepts and principles of quantum theory, such as superposition of quantum states, entanglement, and intervention. Quantum evolutionary algorithm (QEA) is proposed by combining quantum theory with evolutionary theory. It is a kind of evolutionary algorithm with the form of quantum chromosome; the random observation which simulates the quantum collapse can bring diverse individuals [1]. In QEA, the quantum bits (Q-bits) and quantum gate (Q-gate) have been used to update the individuals to realize evolutionary search. QEA has advantages, including the diversity of population, global search performance, and convergence rate, compared with other evolutionary algorithms due to Q-bits coding and the updating of rotation gate. Since then, QEA were developed, including different operators [2–5] and novel update strategy for Q-gates [6–9]. QEA has been applied in many optimization problems [10, 11]. However, frequent encoding and decoding in QEA will slow down convergence rate. In addition, the magnitude and the direction of the rotational angle of quantum rotation gates cannot be determined during optimization.

Recently, numerous literatures on QEA have been reported. Han and Kim [12] proposed a quantum-inspired evolutionary algorithm with $H_\varepsilon$ gate in which the probability amplitudes of rotation gate converge to $\varepsilon$ and $1 - \varepsilon$. Chen et al. [13] proposed a rotation gate with chaos in QEA. It applied generated chaotic sequence beforehand at the updating of rotation gate to improve the computational speed. Yang et al. [14] proposed a rotation gate based on arithmetic guidance. In the proposed algorithm, the current optimal individuals transfer to next generation. S. Y. Li and P. C. Li [15] proposed the double chains quantum genetic algorithm (DCQGA) which mapped the real variables to the probability amplitude on Q-bit by transforming the solution space; in particular, a real Q-bit encoding can be used to avoid binary coding. da Cruz et al. [16] proposed a real-coded quantum evolutionary algorithm in which the solution space is expressed by rectangular region. Its outstanding performance in solving high dimensional continuous optimization problems was verified. P. Li and S. Li [17] proposed a Bloch spherical coding for QEA...
which can avoid local optimization in solving continuous optimization problems. Gao et al. [18] proposed a real-coded quantum evolutionary algorithm based on triploid coding. Zhang et al. [19] proposed four-chain quantum-inspired evolutionary algorithm in which a chromosome contains four gene chains. The above quantum evolutionary algorithms introduce Q-bit encoding and increase diversity of population but spend much time on encoding and decoding; meanwhile, the search process may fall into local minimum and premature. So it is very important for us to employ a novel updating strategy in order to improve convergence rate and prevent the premature.

In this paper, we presented an allele real-coded quantum evolutionary algorithm (ARQEA) based on hybrid updating strategy. The proposed algorithm employs a real-coding to maintain the diversity of allele and introduces a hybrid updating strategy to balance performance between global search and local search in order to improve the convergence rate. The convergence of ARQEA is verified by Markov chain. The numerical examples verify the advantages of ARQEA on convergence rate and search accuracy.

The remainder of this paper is organized as follows: in Section 2, an allele real-coded quantum evolutionary algorithm is proposed in which the real coding as well as hybrid update strategy is presented; in Section 3, the global convergence of proposed algorithm is verified by Markov chain. Finally, the numerical experiments are demonstrated and the advantages are verified in Section 4.

2. An Allele Real-Coded Quantum Evolutionary Algorithm

In ARQEA, the real variables are coding with probability superposition of allele. A hybrid evolutionary strategy balancing the global search and local search is presented in which the superior allele is defined. The supervised evolution as well as variable scale contraction is adopted, and $H_e$ gate is introduced to prevent the premature convergence.

2.1. Real Coding for Allele. We present an allele real coding which will generate two continuous variables $x_i$ and $x_i'$ ($x_i$ and $x_i'$ are defined as the $i$th alleles) to replace 0 and 1 gene of Q-bit coding in QEA. The real coding is

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_i & \cdots & x_n \\ x_1' & x_2' & \cdots & x_i' & \cdots & x_n' \end{pmatrix},$$

(1)

where $n$ is the number of variables.

The probability amplitudes of allele are expressed as trigonometric functions $\cos \theta_i$ and $\sin \theta_i$:

$$\begin{pmatrix} \cos \theta_1 & \cos \theta_2 & \cdots & \cos \theta_i & \cdots & \cos \theta_n \\ \sin \theta_1 & \sin \theta_2 & \cdots & \sin \theta_i & \cdots & \sin \theta_n \end{pmatrix},$$

(2)

where $\cos \theta_i$ and $\sin \theta_i$ are the probability amplitudes of $x_i$ and $x_i'$; the initial value of $\theta_i$ is $\pi/4, i = 1, \ldots, n$.

Then the observed probability of $x_i$ and $x_i'$ is $\cos^2 \theta_i$ and $\sin^2 \theta_i$, respectively. So

$$\cos^2 \theta_i + \sin^2 \theta_i = 1.$$  (3)

We employ the probability superposition to maintain the diversity for individual in real coding. And the proposed real coding prevents overlong chromosome length and saves time spent on encoding and decoding.

2.2. Hybrid Updating Strategy. In ARQEA, we adopt hybrid updating strategy to update allele and its probability amplitudes.

2.2.1. Updating Alleles. Firstly, we encode initial population on the basis of allele real coding scheme, compute the fitness of chromosomes, and recode the superior individual $x_i^*$. Then we compare the distances between the $i$th alleles $x_i$ and $x_i'$ and define the allele corresponding to the shorter distance with $x_i^*$ as “superior allele,” the other one as “inferior allele.” Figure 1 shows the superior and inferior alleles, in which $x_i$ and $x_i'$ are the $i$th alleles of chromosome, $x_i^*$ is the $i$th allele of the current optimal individual, and then we define $x_i$ as the superior allele and $x_i'$ as the inferior allele.

Next, we will update the alleles. The updating strategy is described as follows:

(a) For the superior allele, we employ the updating strategy, such as reasonable choice of the initial step length and dynamic adjustment of the search step length, to guide superior allele to search optimal solution.

In the early search stage, the updating strategy is accelerated by directing the search direction

$$x_{i+1} = x_i + \text{sign}(x_i^* - x_i) \cdot \Delta d_1,$$  (4)

where $x_i$ is a superior allele, $x_{i+1}$ is the updated allele, $\Delta d_1 = \lambda \cdot |x_i^* - x_i|$ is the search step length, usually $\lambda = 0.5$, and sign() is a sign function; it decides the search direction of superior allele.

When $|x_i^* - x_i|/(x_{\text{max}} - x_{\text{min}})$ is less than $\eta$, we adopt a variable scale contraction to search the optimal solution

$$x_{i+1} = x_i + U(-1, 1) \cdot \left(1 - \arctan\left(\frac{r}{g}\right)\right)^4 \cdot \Delta d_2,$$  (5)

where $r$ is the number of current generation, $g$ is the number of max generation, $U(-1, 1)$ is a random distribution between $-1$ and $1, (1 - \arctan(r/g))^4$ is a change from 1 to 0 with the number of generation $r$, and $\Delta d$ is the range of contraction. When $U(-1, 1)$ is a negative number, $\Delta d_2 = x_i - x_{\text{min}}$.

In this search stage, the superior allele will randomly search the optimal solution in a certain region, and the searching scale will change with the number of generation $r$. 

![Figure 1: The superior and inferior alleles.](image-url)
(b) For the inferior allele \( x_i' \), the allele \( x_i \) will be substituted to \( x_i' \) in (4) and (5) and alternate operation every two generations using (4) and (5).

However, considering the randomness of the evolutionary algorithm, the inferior allele has the ability to escape from local optimum and approach the global optimum in the evolution stage.

2.2.2. Updating Probability Amplitudes. After updating the allele, \( H_z \) gate is used to update the corresponding probability amplitudes [20, 21]. We employ \( H_z \) gate operation to prevent the premature convergence. The rotation angle of \( H_z \) gate is

\[
\theta_{t+1} = \theta_t \pm \left(1 - \arctan \left( \frac{r}{g} \right) \right) \Delta t, \quad (6)
\]

where \( \Delta t \) is the range of \( \theta_t \) and its direction depends on the superior allele.

If the superior allele corresponds to \( \cos \theta_t \), then (6) is subtracting. Instead, it is adding. The \( H_z \) gate operation will increase the observed probability for the superior allele and decrease the observed probability for the inferior allele.

The operation rule of \( H_z \) gate is the following:

(a) if \(|\alpha_i'|^2 \leq \varepsilon \) and \(|\beta_i'|^2 \geq 1 - \varepsilon \), then \([\alpha_i'' \beta_i'']^T = \left[ \sqrt{\varepsilon} \ \sqrt{1 - \varepsilon} \right]^T \);

(b) if \(|\alpha_i'|^2 \geq 1 - \varepsilon \) and \(|\beta_i'|^2 \leq \varepsilon \), then \([\alpha_i'' \beta_i'']^T = \left[ \sqrt{1 - \varepsilon} \ \sqrt{\varepsilon} \right]^T \);

(c) else \([\alpha_i'' \beta_i'']^T = [\alpha_i' \beta_i']^T \),

where \( \varepsilon \) is a real number between 0 and 1, \([\alpha_i' \beta_i']^T \) are the probability amplitudes after the \( H_z \) gate operation, and \([\alpha_i'' \beta_i'']^T \) are the probability amplitudes after the \( H_z \) gate operation.

The steps of hybrid evolutionary strategy are described as follows.

Step 1. Judge the relative superior allele.

Step 2. For the superior allele, adopt supervised evolution by (4) to find current optimal allele \( x_i^* \), and employ variable scale contraction by (5) approach to local search when \(|x_i^* - x_i|/(x_{max} - x_{min}) \) is less than \( \eta \).

Step 3. For the inferior allele, search alternately by (4) and (5) in every two generations.

Step 4. Update the probability amplitudes of alleles by (6).

In hybrid updating strategy, we employ different evolutionary mode for superior and inferior allele in order to balance global search and local search. The superior allele affects local search stage, and inferior allele plays important role in global search by the variable scale contraction to search unknown space and maintain the diversity of population. As generation goes on, superior allele and inferior allele exchange position in next generation dynamically. The hybrid updating strategy proposed has two mechanisms for superior and inferior allele to maintain diversity of population. In addition, the hybrid updating strategy in ARQEA balances the global search and local search by dynamic updating strategy of alleles.

2.3. Process of ARQEA. ARQEA employs allele real coding with probability superposition which generates a random number \( r_i \) between 0 and 1 and compares \( r_i \) with the probability amplitudes for each allele and adopts hybrid updating strategy to balance global search and local search. At the early search stage, the scale of mutation is large, the step size in search is long, and the algorithm approaches to the global search. With the increasing of generations, the scale of mutation becomes smaller and the performance of local search is strengthened. The process of ARQEA is shown in Figure 2.

3. Convergence Analyses

In this section, we analyze the global convergence of ARQEA by Markov chain. The ARQEA is described as follows.

Consider ARQEA:

\[
(\Omega, G, D, P, n, X(0), X'(t), X''(t), X(t), F_{min}, X_{best}, F_{best}, \epsilon),
\]

where \( \Omega \) is search space, \( G \) is the evolutionary operator of superior allele, \( D \) is the evolutionary operator of inferior allele, \( P \) is the operator of probability selection, \( n \) is the dimension of \( \Omega \), \( X(0) \) is initial population, \( X'(t) \), \( X''(t) \), and \( X(t) \) are the superior allele, the inferior allele, and the observed allele in \( t \)th generation, respectively, \( F_{min} \) is the fitness function, \( X_{best} \) is the optimal solution and \( F_{best} \) is optimal value of the fitness function, and \( \epsilon \) is the termination criterion.

According to the hybrid update strategy, the population space of ARQEA can be expressed as follows:

\[
X(t+1) = PG(X'(t)) + (1-P)D(X''(t)).
\] (8)

In (8), \( X(t+1) \) relates to the state of generations \( t \). Then, the state space of ARQEA can be expressed as

\[
\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{max}\},
\] (9)

where max = \( \prod_{i=1}^{n}(\pm (\epsilon / \delta) \sqrt{\Omega}) \). \( X'^{u} \) and \( X'^{d} \) are the upper bound and lower bound of the \( i \)th variable. So ARQEA can be regarded as a finite Markov process.

Then the state space \( \Omega \) is divided into two search spaces according to the termination criteria:

(a) \( S_1 = \{\Omega_i | |F_{\Omega_i} - F_{best}| < \epsilon\} \);

(b) \( S_2 = \{\Omega_i | |F_{\Omega_i} - F_{best}| \geq \epsilon\} \).

We define \( p_{1 \rightarrow 1} \) as the transition probability from \( S_1 \) to \( S_1 \), \( p_{2 \rightarrow 1} \) as the transition probability from \( S_2 \) to \( S_1 \), and \( p_{2 \rightarrow 2} \) as the transition probability from \( S_2 \) to \( S_2 \). And \( p_{1 \rightarrow 1} = 1 \).
**Lemma A** (see [22]). Assume $A_1, A_2, \ldots$ are the event sequence in probability space, and let $p^k = p(A_k)$; if $\sum_{k=1}^{\infty} p^k < \infty$, then $p(\bigcap_{i=1}^{\infty} \bigcup_{k \geq i} A_k) = 0$; if $\sum_{k=1}^{\infty} p^k = \infty$ and all $A_k$ are mutually independent, then $p(\bigcap_{i=1}^{\infty} \bigcup_{k \geq i} A_k) = 1$.

**Theorem A.** $\exists \phi \in (0, 1)$, make $p_{2 \rightarrow 2} < \phi$ in ARQEA.

**Proof.** $\exists d > 0$, when $\epsilon > 0$ and $|X(t) - X_{\text{best}}| \leq d$, then $|F(X(t)) - F_{\text{best}}| < \lambda$, in which $0 < \lambda < \epsilon$; if $C_1 = \{X(t) \mid |X(t) - X_{\text{best}}| \leq d\}$, then $C_1 \subset S_1$.

The variable scale contraction in ARQEA is Gaussian distribution. If the individual $X$ in $S_2$ performs variable scale contraction operation and obtained $X + \delta$, then

$$p \{ (X + \delta) \in C_1 \} < p \{ (X + \delta) \in S_1 \} = p_{2 \rightarrow 1}. \quad (10)$$

Since $\delta$ is Gaussian distribution $N(0, \delta^2)$, the probability density is

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-x^2/2\sigma^2}. \quad (11)$$

Because

$$p \{ (X + \delta) \in C_1 \} = \prod_{i=1}^{n} p \{ ||X_i + \delta - X_{\text{best}}|| \leq d \} = \prod_{i=1}^{n} \int_{X_{\text{best}} - X_i - d}^{X_{\text{best}} - X_i + d} f(x) \, dx, \quad (12)$$
Table 1: The test functions.

| Number | Function expression | Search space | \( f_{\text{min}} \) |
|--------|----------------------|--------------|----------------|
| 1      | \( f_1(x_1, x_2) = 0.5 + \sin^2\sqrt{x_1^2 + x_2^2} - 0.5 \) | \( x_i \in [-10, 10] \) | 0 |
| 2      | \( f_2(x_1, x_2) = 100(x_1 - x_2^2)^2 + (x_1 - 1)^2 \) | \( x_i \in [-10, 10] \) | 0 |
| 3      | \( f_3(x_1, x_2) = -20 \exp\left(-0.2 \sqrt{x_1^2 + x_2^2}\right) - \exp\left(\frac{\cos(2\pi x_1) + \cos(2\pi x_2)}{2}\right) + \exp(1) \) | \( x_i \in [-10, 10] \) | 0 |
| 4      | \( f_4(x) = 418.9829 x_1 + \sum_{i=1}^{n} x_i \sin\left(\sqrt{|x_i|}\right), n = 10 \) | \( x_i \in [-500, 500] \) | 0 |
| 5      | \( f_5(x) = \sum_{i=1}^{n} \frac{i+1}{10} (\exp(x_i) - x_i) - 5.5, n = 10 \) | \( x_i \in [-5.12, 5.12] \) | 0 |
| 6      | \( f_6(x) = \frac{1}{4000} \sum_{i=1}^{n} (x_i^2 - \sum_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i+1}}\right)) + 1, n = 20 \) | \( x_i \in [-100, 100] \) | 0 |
| 7      | \( f_7(x) = n \sum_{i=1}^{n} x_i^2, n = 30 \) | \( x_i \in [-10, 10] \) | 0 |
| 8      | \( f_8(x) = 10n + \sum_{i=1}^{n} x_i^2 - 10 \sum_{i=1}^{n} \cos(2\pi x_i), n = 30 \) | \( x_i \in [-10, 10] \) | 0 |

then

\[
0 < \prod_{i=1}^{n} \int_{X_{\text{min}}-X_{\text{var}}}^{X_{\text{max}}-X_{\text{var}}} f(x) dx < 1 \implies 0 < p \{(X + \delta) \in C_1\} < 1. \tag{13}
\]

Because \( \delta \) depends on Gaussian distribution and \( p\{(X + \delta) \in C_1\} \) is continuous, then

\[
p\{(X' + \delta) \in S_1\} = \min \{ p\{(X + \delta) \in S_1\}\}. \tag{14}
\]

Let \( \varphi = 1 - p\{(X' + \delta) \in S_1\}; \) we can obtain following equation by (11) and (14):  

\[
p\{(X' + \delta) \in S_1\} \leq p\{(X + \delta) \in S_1\} < p_{2-1}
\]

\[
= 1 - p_{2-2} \implies p_{2-2} \leq \varphi. \tag{15}
\]

\( p_{2-2} < \varphi \) \;

\( \Box \)

**Theorem B.** If one-step transition probability in ARQEA \( p_{2-2} < \varphi \), then ARQEA is global convergence.

**Proof.** For the termination criterion \( \varepsilon > 0 \), when the probability \( p'_{2-2} = p\{|F(X(t)) - F(X_{\text{best}})| \geq \varepsilon\} \), \( X(t) \) cannot satisfy the termination criterion \( \varepsilon > 0 \) after \( t \) generations; we can obtain (16) by Theorem A:

\[
\sum_{t=1}^{\infty} p'_t < \sum_{t=1}^{\infty} \varphi \tag{16}
\]

and \( 0 < \varphi < 1 \), so \( \sum_{t=1}^{\infty} \varphi = \varphi/(1 - \varphi) \); we can get (17) by Lemma A:

\[
p \left\{ \bigcap_{t=1}^{\infty} \bigcup_{t \in S_1} \{ F(X(t)) - F(X_{\text{best}}) \geq \varepsilon \} \right\} = 0. \tag{17}
\]

So \( \lim_{t \to \infty} F(X(t)) = F(X_{\text{best}}) \). That is, ARQEA can converge to the optimal solution with probability 1. \( \Box \)

**4. Experiments and Results Analysis**

In this section, we investigate the search ability of ARQEA by eight test functions and compare the results with GA, QEA, and DCQGA. And the above optimization algorithms are encoded in MATLAB 2010b and implemented on computer with Intel T3200 CPU and 2 G DDR3 memory using Windows XP.

The test functions are listed in Table 1. The test functions contain trigonometric function, polynomial function, and exponential function in the list. And \( f_1, f_2, \) and \( f_3 \) are two-dimension multiple hump functions; \( f_4, f_5, f_6, f_7, \) and \( f_8 \) are high dimension functions. Each test function has only one optimal solution in feasible region, in which the minimum of \( f_1 \) and \( f_2 \) is 0 at \([0, 0]\), the minimum of \( f_3 \) is 0 at \([1, 1]\), and the minimum of test functions \( f_4, f_5, f_6, f_7, \) and \( f_8 \) is 0 at \([0, 0, \ldots, 0]\). However, the test functions with high dimension will bring more difficulty in global and local search process.

In this experiment, the population size is 50 for GA, QEA, DCQGA, and ARQEA; the max generation is 500 for two-dimension test functions \( f_1, f_2, \) and \( f_3 \), and 3000 for high dimension test functions \( f_4, f_5, f_6, f_7, \) and \( f_8 \). And the coding for GA, QEA, DCQGA, and ARQEA is implemented, respectively, as follows:

(a) GA adopts real coding. The real coding is adapted to every variable in GA. The selection operation employs
Figure 3: Continued.
Computational Intelligence and Neuroscience

Generations
ARQEA
DCQGA
QEA
GA

(g) The convergence curve of \( f_7 \)

(h) The convergence curve of \( f_8 \)

Figure 3: Comparison of convergence for GA, QEA, DCQGA, and ARQEA.

5. Conclusions

In this paper, we present an allele real-coded quantum evolutionary algorithm. The proposed algorithm employs real coding based on probability superposition to maintain the diversity of population and prevent overlong length of chromosome. In hybrid updating strategy, we employ different evolutionary mode for superior and inferior allele in order to balance global search and local search. By comparing with different optimization algorithms, the search accuracy and convergence rate of ARQEA are obviously superior to GAs, QEAs, and DCQGAs in solving continuous optimization problem. For the future, we will analyze the influence
Table 2: Comparison with different optimization algorithms.

| Test function | Evolutionary algorithms | Best value   | Average value | Worst value | Running time (sec.) |
|---------------|-------------------------|--------------|---------------|-------------|---------------------|
|               | QEA                     | 3.4533e – 06 | 4.5658e – 04 | 1.3433e – 03 | 9.543e + 01         |
|               | DCQGA                   | 6.8996e – 08 | 4.3543e – 06 | 7.8732e – 04 | 6.526               |
|               | GA                      | 2.6704e – 11 | 6.3684e – 03 | 9.7032e – 03 | 8.654               |
|               | ARQEA                   | 3.0738e – 15 | 6.0451e – 15 | 2.0368e – 14 | 3.481               |
| $f_1$         | QEA                     | 6.3275e – 04 | 5.2164e – 03 | 1.6233e – 02 | 5.569e + 01         |
|               | DCQGA                   | 5.5437e – 08 | 1.2654e – 06 | 4.3251e – 04 | 5.658               |
|               | GA                      | 1.5684e – 05 | 9.1982e – 02 | 4.0532e – 01 | 6.423               |
|               | ARQEA                   | 2.1055e – 13 | 4.5264e – 13 | 7.3382e – 13 | 1.291               |
| $f_2$         | QEA                     | 1.6048e – 02 | 3.2418e – 02 | 5.2529e – 02 | 4.1324e + 01        |
|               | DCQGA                   | 1.4323e – 05 | 5.2189e – 02 | 1.5917e – 01 | 6.4383              |
|               | GA                      | 1.2181e – 06 | 3.7907e – 04 | 6.3203e – 03 | 2.2957              |
|               | ARQEA                   | 2.5346e – 16 | 4.8919e – 13 | 2.1526e – 09 | 4.4252              |
| $f_3$         | QEA                     | 5.0412e + 01 | 1.6542e + 02 | 3.4774e + 02 | 4.8342e + 03        |
|               | DCQGA                   | 2.4342       | 7.0424e + 01 | 2.4292e + 02 | 5.3226e + 01        |
|               | GA                      | 7.0136e – 04 | 2.0751e + 01 | 2.3687e + 01 | 3.5501e + 01        |
|               | ARQEA                   | 1.2728e – 16 | 3.2573e – 16 | 8.8818e – 16 | 5.0032e + 01        |
| $f_4$         | QEA                     | 4.9056e – 04 | 1.3872e – 02 | 3.4314e – 02 | 3.5231e + 02        |
|               | DCQGA                   | 7.3977e – 06 | 1.8576e – 05 | 3.9103e – 03 | 6.3342e + 01        |
|               | GA                      | 5.0090e – 08 | 1.4547e – 05 | 6.9581e – 05 | 4.7454e + 01        |
|               | ARQEA                   | 1.7982e – 16 | 3.2573e – 16 | 8.8818e – 16 | 5.0032e + 01        |
| $f_5$         | QEA                     | 2.2067e – 01 | 3.3742        | 7.4107       | 3.3703e + 03        |
|               | DCQGA                   | 3.5241       | 2.3371        | 5.7223       | 1.0399e + 02        |
|               | GA                      | 2.9913e – 03 | 2.3544e – 11 | 6.6274e – 01 | 7.3251e + 01        |
|               | ARQEA                   | 9.9678e + 03 | 2.0576e – 01 | 2.7094e – 02 | 4.812e + 01         |
| $f_6$         | QEA                     | 3.6544e – 02 | 4.7685e – 01 | 1.3276       | 3.586e + 03         |
|               | DCQGA                   | 8.9768e – 04 | 3.4765e – 03 | 5.5876e – 02 | 1.6352e + 02        |
|               | GA                      | 6.0000e – 03 | 3.7400e – 02 | 0.2193       | 6.3324e + 01        |
|               | ARQEA                   | 4.9223e – 14 | 7.2357e – 12 | 8.5674e – 11 | 5.01e + 01          |
| $f_7$         | QEA                     | 6.3267e – 03 | 2.6832e – 01 | 1.9519       | 1.2778e + 04        |
|               | DCQGA                   | 3.5257e – 05 | 4.2347e – 04 | 7.0863e – 01 | 2.8593e + 03        |
|               | GA                      | 2.6432e – 02 | 1.4242        | 3.0274       | 8.07e + 01          |
|               | ARQEA                   | 4.1248e – 11 | 5.3442e – 10 | 9.3327e – 08 | 6.158e + 01         |

on initial population to search accuracy and relationship between parameters in updating strategy and convergence rate.

Conflict of Interests

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

References

[1] 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.

[2] 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, Republic of Korea, July 2005.

[3] 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.

[4] 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, IEEE, June 2004.

[5] 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.

[6] G.-X. Zhang, N. Li, W.-D. Jin, and L.-Z. Hu, "Novel quantum genetic algorithm and its applications," in Modeling Decisions for Artificial Intelligence, vol. 3131 of Lecture Notes in Artificial Intelligence, pp. 92–103, Springer, Berlin, Germany, 2004.
[8] 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.

[9] 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, IEEE, Hunan, China, July 2007.

[10] T. Du, Y. Hu, and X. Ke, “Improved quantum artificial fish algorithm application to distributed network considering distributed generation,” Computational Intelligence and Neuroscience, vol. 2015, Article ID 851863, 13 pages, 2015.

[11] R. Jiao, B. Li, Y. Li, and L. Zhu, “An enhanced quantum-behaved particle swarm algorithm for reactive power optimization considering distributed generation penetration,” Mathematical Problems in Engineering, vol. 2015, Article ID 342080, 9 pages, 2015.

[12] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithms with a new termination criterion, $H_1$ gate, and two-phase scheme,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 2, pp. 156–169, 2004.

[13] H. Chen, J. Zhang, and C. Zhang, “Chaos updating rotated gates quantum-inspired genetic algorithm,” in Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS '04), vol. 2, pp. 1108–1112, IEEE, Chengdu, China, June 2004.

[14] 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), pp. 820–826, IEEE, Vancouver, Canada, June 2004.

[15] S. Y. Li and P. C. Li, “Quantum evolutionary algorithm based on real coding and gradient information of object function,” Journal of Harbin Institute of Technology, vol. 38, no. 8, pp. 1216–1219, 2006.

[16] A. V. A. da Cruz, M. M. B. R. Vellasco, and M. A. C. Pacheco, "Quantum-inspired evolutionary algorithm for numerical optimization," in Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06), pp. 2630–2637, IEEE, Vancouver, Canada, July 2006.

[17] 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. 581–591, 2008.

[18] H. Gao, G. H. Xu, R. Zhang, and Z. R. Wang, "Real-coded quantum evolutionary algorithm," Control and Decision, vol. 23, no. 1, pp. 87–90, 2008.

[19] R. Zhang, Z. T. Wang, and H. J. Zhang, "Quantum-inspired evolutionary algorithm for continuous space optimization based on multiple chains encoding method of quantum bits," Mathematical Problems in Engineering, vol. 2014, Article ID 620325, 15 pages, 2014.

[20] H. Chen, J. Zhang, and C. Zhang, "Chaos updating rotated gates quantum-inspired genetic algorithm," in Proceedings of the IEEE International Conference on Communications, Circuits and Systems (ICCCAS '04), vol. 2, pp. 1108–1112, Chengdu, China, June 2004.

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

[22] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in Proceedings of the IEEE World Congress on Computation Intelligence, pp. 69–73, Anchorage, Alaska, USA, May 1998.