Higher-Order Quantum Genetic Algorithm and Its Application on Non-Linear Equations

Lei Jin¹*, Zheng Shan², Chunchao Hu¹ and Le Xu¹

¹School of Information Engineering, Zhengzhou University, Zhengzhou, Henan, 450000, China
²State Key Laboratory of Mathematical Engineering and Advanced Computing, Information Engineering University, Zhengzhou, Henan, 450000, China

*Corresponding author’s e-mail: jl@gs.zzu.edu.cn

Abstract. The high efficiency of quantum algorithms is caused by the quantum parallelism of the superposition principle and the quantum entangled, but traditional quantum genetic algorithms only use the quantum superposition principle. In order to further improve the performance of the algorithm, this paper proposes a new higher-order quantum genetic algorithm, which adds quantum entanglement properties on the basis of the principle of quantum superposition. Finally, the traditional quantum genetic algorithm and the higher-order quantum genetic algorithm are used to test a set of nonlinear equations many times. The results show that compared with the traditional quantum genetic algorithm, the evolutionary update operation of the higher-order quantum genetic algorithm does not involve multiple judgment conditions of the traditional quantum gate, requires less evolutionary algebra, and can converge quickly.

1. Introduction
Quantum Genetic Algorithm [1] (QGA) is a probabilistic evolutionary algorithm with quantum characteristics that was born by adding quantum computing theory to the classical genetic algorithm [2] (GA). It uses qubit chromosomes with superposition characteristics for encoding, and updates through quantum revolving gates to achieve the optimal solution of the goal. Its unique computing performance has attracted widespread attention in the scientific community.

Quantum computing is different from the definite value in classical physics. Quantum motion is governed by statistical laws. We call this magical and variable characteristic of microscopic particles as quantum state characteristics. It uses the superposition, coherence and entanglement [3] of different bits in the quantum state to realize quantum computing. In quantum computing, the system is not in a constant state, it has a certain probability, and the state probability vector corresponds to different state possibilities. Due to quantum characteristics are different from classical physics, quantum computing has the ability to make up for the shortcomings of classical algorithms in some problems, improve algorithm performance, and achieve the purpose of improving algorithms.

The genetic algorithm with quantum characteristics has enriched the diversity of the population, and the search ability and convergence speed of the algorithm have been improved. The high efficiency of quantum computing is caused by the quantum parallelism of the superposition principle and the quantum entangled, but the current quantum genetic algorithm only uses the characteristic of quantum superposition, and does not use quantum entanglement. In order to further improve the
performance of the algorithm, this paper proposes a new higher-order quantum genetic algorithm based on the principle of superposition, adding quantum entanglement characteristics to improve the search efficiency of the algorithm's optimal solution. Taking the nonlinear equation system as an example of the optimization problem, and the nonlinear equation system is tested many times through traditional quantum genetic algorithm and higher-order quantum genetic algorithm. The results show that the convergence speed of higher-order quantum genetic algorithms has been further improved.

2. Theory of higher-order quantum genetic algorithm

2.1. Structure of Quantum chromosome

Traditional QGA uses qubits to encode individuals, and each qubit is represented by a pair of complex numbers \((\alpha, \beta)\), defined as:

\[
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}
\]  

(1)

We often write Equation (1)’ as a quantum mechanical expression:

\[ |q\rangle = \alpha |0\rangle + \beta |1\rangle \]  

(2)

Where \(\alpha\) and \(\beta\) represent the probability amplitude of the corresponding state. In quantum theory, \(|\alpha|^2\) and \(|\beta|^2\) represent the probability that the qubit collapse to the \(|0\rangle\) and \(|1\rangle\), respectively, and satisfy the condition that \(|\alpha|^2 + |\beta|^2 = 1\) [4]. Each quantum bit can be probabilistically represented as a binary bit of 0 or 1 by the process of observation.

It only uses the principle of quantum superposition, in which an individual is made up of a set of independent quantum bits. A system with m-bit qubits can be expressed as:

\[ |q_1, q_2, \cdots, q_m\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle + \cdots + \alpha_m|1\rangle \]

with a normalization constraint:

\[ |\alpha_1|^2 + |\alpha_2|^2 + \cdots + |\alpha_m|^2 = 1 (i = 1, 2, \cdots, m) \]

The system can represent \(2^m\) states at the same time, after passing through an observation process, it will collapse into a single definite state.

On the basis of the principle of quantum superposition, higher-order QGA adds the characteristic of quantum entanglement, that is, entanglement of arbitrary qubits forms a quantum register, several quantum registers form a quantum chromosome, and finally multiple quantum chromosomes form a population. To facilitate discussion, we assume that all registers in a quantum chromosome are entangled by the same number of qubits. If each r qubits are entangled together, the system of m qubits can be expressed as:

\[
\begin{array}{cccc}
R_1 & R_2 & \cdots & R_{\frac{m}{r}} \\
|q_1\rangle & |q_r\rangle & \cdots & |q_{r+1}\rangle & |q_{2r}\rangle & \cdots & |q_{m-r}\rangle & \cdots & |q_m\rangle
\end{array}
\]

Each r qubits are entangled to form a quantum register, and the m-bit qubit system has \(k = \frac{m}{r}\) registers. Quantum registers also follow the principle of superposition. A quantum register can get \(n = 2^r\) states.
During the implementation of the algorithm, it is convenient for the quantum chromosome to be represented by the following structure. As shown in Table 1, it consists of $k=N/r$ ($r=2$) quantum registers.

**Table 1. Two-qubit entanglement ($r=2$) chromosome structure representation.**

| $R_1$ | $R_2$ | $R_3$ | $\ldots$ | $R_j$ | $\ldots$ | $R_k$ |
|-------|-------|-------|----------|-------|----------|-------|
| $\alpha_0^0$ | $\alpha_0^1$ | $\alpha_0^2$ | $\ldots$ | $\alpha_0^j$ | $\ldots$ | $\alpha_0^k$ |
| $\alpha_1^0$ | $\alpha_1^1$ | $\alpha_1^2$ | $\ldots$ | $\alpha_1^j$ | $\ldots$ | $\alpha_1^k$ |
| $\alpha_2^0$ | $\alpha_2^1$ | $\alpha_2^2$ | $\ldots$ | $\alpha_2^j$ | $\ldots$ | $\alpha_2^k$ |
| $\alpha_3^0$ | $\alpha_3^1$ | $\alpha_3^2$ | $\ldots$ | $\alpha_3^j$ | $\ldots$ | $\alpha_3^k$ |

Just as the qubits of traditional QGA can express the linear superposition of states in the form of probability, the register composed of entangled bits of higher-order QGA can also be expressed by the linear combination of these ground states. Such as $r=2$:

$$|q\rangle = \alpha_0^0|00\rangle + \alpha_1^0|01\rangle + \alpha_2^0|10\rangle + \alpha_3^0|11\rangle$$

Here $\alpha_0^2, \alpha_1^2, \alpha_2^2, \alpha_3^2$ represents the magnitude of the register probability size, which still needs to meet the normalization condition: $\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1$. In order to ensure that any state appears with the same probability [5], the probability amplitude needs to be equal in the initialization phase: $\alpha_0^2 = \alpha_1^2 = \alpha_2^2 = \frac{1}{4}$. If $r=1$, the higher-order QGA becomes a traditional QGA, indicating the applicability of the higher-order QGA. If $r=N$, it becomes a real quantum algorithm, that is, a chromosome is a register that contains all the states, but the cost of simulation also increases exponentially.

### 2.2 Quantum chromosome measurement

Just as the chromosome of the traditional QGA is collapsed into a set of classical binary representation after measuring qubits, the result of the chromosome measurement register of the higher-order QGA is also a binary representation. The quantum measurement symbol is implemented according to [6].

The algorithm 1 gives the pseudo code of the quantum chromosome measurement process composed of $k$ quantum registers of size $r$:

**Algorithm 1. Quantum register state measurement.**

```plaintext
For $i \in 1, \ldots, k$ do
    rand $\leftarrow$ random number in the area $[0,1]$
    Sum $\leftarrow$ 0
    For $j \in 1, \ldots, 2^r - 1$ do
        Sum $\leftarrow$ Sum $+ \left| \alpha_j^i \right|^2$
        if rand $<$ Sum then
            $p \leftarrow W_j$
        end if
    end for
end for
```

Where the $i$ index is responsible for traversing all registers, index $j$ is used to determine the state of the quantum register, $W$ is the auxiliary matrix corresponding to the classical binary representation of the state index number $j$. If $r=2$, then $W$ is shown in Table 2. In this case, the classic representation of the register is the binary representation corresponding to the index number $j$. 

---

During the implementation of the algorithm, it is convenient for the quantum chromosome to be represented by the following structure. As shown in Table 1, it consists of $k=N/r$ ($r=2$) quantum registers.
Table 2. Auxiliary matrix.

| State register index number | Classical representation of the register |
|-----------------------------|------------------------------------------|
| 0                           | 00                                       |
| 1                           | 01                                       |
| 2                           | 10                                       |
| 3                           | 11                                       |

2.3. Quantum revolving gate

Traditional QGA uses the unique quantum revolving gate in quantum theory [7], through the mutual interference of each quantum superposition state, the probability amplitude of the quantum state is changed, thereby updating the population and enriching the diversity of the population. However, the rotation angle and direction of the traditional QGA revolving gate are determined according to the look-up table, which involves the judgment of multiple conditions, which affects the efficiency of the algorithm. The offspring individuals of higher-order QGA are also determined by the optimal individuals of the previous generation and their probability amplitude states. By continuously applying quantum operators to change the linear superposition state of quantum registers, the state of the register to be updated corresponds to the optimal individual. The status change of the register does not involve multiple judgment conditions of the lookup table, which improves the algorithm efficiency.

The new update operator is applied to each register in two stages, and its working mode is shown in Figure 1.

The new update operator is applied to each register in two stages, and its working mode is shown in Figure 1.

\[
\begin{align*}
R_b & \in \{ 00, 01, 10, 11 \} \\
|R_x\rangle & = [\alpha_0, \alpha_1, \alpha_2, \alpha_3] \\
\end{align*}
\]

\[
R_{b_i} \in \{ 00, 01, 10, 11 \}
\]

\[
|R_x\rangle = [\alpha_0, \alpha_1, \alpha_2, \alpha_3]
\]

Figure 1. Higher-order QGA (r=2) evolutionary update process.

\[
\begin{align*}
R_{b_i} & \text{ the optimal chromosome of the previous generation with length N, and } R_x \text{ is the chromosome to be updated with length N. Each r=2 qubits are entangled to form } k = \frac{N}{2} \text{ quantum registers. Each register } R_{x_i} \text{ of chromosome } R_x \text{ to be updated evolves with the state of register } R_{b_i} \text{ of the optimal chromosome } R_b \text{ of the previous generation. As shown in the right half of the picture above: The length of the vertical bar represents the size of the probability amplitude of the register to be updated. Assume that the } i-th \text{ register } R_{x_i} \text{ of the individual to be updated corresponds to the } i-th \text{ register } R_{b_i} \text{ of the optimal individual, and the } b \text{ index number of the optimal representation } 10 \text{ of the register } R_{b_i} \text{ is 2. In the first stage, the probability amplitude } |\alpha_{x_i}'|^2 \text{ of } R_{x_i} \text{ needs to be increased:}
\end{align*}
\]

\[
|\alpha_{x_i}'|^2 = \sqrt{|\alpha_{b_i}'|^2 + \mu(1-|\alpha_{b_i}'|^2)}
\]

\[
(3)
\]

Where \( \mu \) is a parameter in the range \([0,1]\). In the second stage, in order to meet the normalization condition, other probability amplitudes of registers need to be reduced. These two stages make register
\( R_{ij} \) approach the best individual register \( R_{ib} \). Finally, the individual to be updated approaches the best individual of the previous generation.

The evolutionary update operation of quantum chromosome \( R_{xi} \) composed of \( k \) quantum registers of size \( r \) is shown in Algorithm 2:

**Algorithm 2. Quantum gate operator.**

```
For \( i \in 1, \cdots, k \) do
    bestamp \( \leftarrow \) the index \( j \in 0, \cdots, 2^r - 1 \) of best binary genes in \( R_{ib} \)
    Sum \( \leftarrow 1 - \left| \alpha_{bestamp}^i \right|^2 \)
    \( \alpha_{bestamp}^i = \sqrt{ \left| \alpha_{bestamp}^i \right|^2 + \mu(1 - \alpha_{bestamp}^i) } \)
    \( M \leftarrow \sqrt{1 - \left| \alpha_{bestamp}^i \right|^2 \over Sum} \)
    for \( amp \in \{0,1,\cdots,2^r - 1\} \) do
        if \( amp \neq bestamp \) then
            \( \alpha_{amp}^i = M \cdot \alpha_{amp}^i \)
        end if
    end for
end for
```

Where \( bestamp \) is the index number of the optimal representation of the quantum register \( R_{ib} \) of the optimal individual, index \( i \) traverses all quantum registers, and \( amp \) traverses all index numbers of the quantum register probability amplitude.

3. **Higher-order quantum genetic algorithm in solving equations**

The simulation test in this article is based on AMD CPU 2.40GHz and memory 8.0GB. The algorithm written and compiled in MATLAB.

3.1. **Problem description**

Suppose the system of nonlinear equations is:

\[
f_j (x_1, x_2, \cdots, x_n) = 0
\]

Where \( j = 1, 2, \cdots, m \), \( X = (x_1, x_2, \cdots, x_n) \in D \subset R^n \), \( D = \{(x_1, x_2, \cdots, x_n) | a_i \leq x_i \leq b_i, i=1, 2, \cdots, n \} \).

Assuming that the nonlinear equation system has a unique solution in region D, and it is known that genetic algorithm can be applied to function optimization, the constructor function:

\[
F(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{m} f_i (x_1, x_2, \cdots, x_n)
\]

In this way, seeking the solution of the nonlinear equations in the region is transformed into a function optimization problem:

\[
\min F(x_1, x_2, \cdots, x_n) \quad s.t. (x_1, x_2, \cdots, x_n) \subset D
\]

Obviously, the optimal solution of function optimization is the solution of the nonlinear equation system, and it satisfies \( F = 0 \).
3.2. Experimental simulation
In order to facilitate the experiment, this paper uses a two-qubit entangled higher-order QGA (r=2) for the experiment. Through classical genetic algorithm (GA), traditional QGA and higher-order QGA (r=2), a set of nonlinear equations are tested multiple times, and then analyzed and compared. The nonlinear equations are:

\[
\begin{align*}
    x^y + y^z - 5xyz - 85 &= 0 \\
    x^3 - y^z - z^x - 60 &= 0 \\
    x^2 + z^y - y - 2 &= 0
\end{align*}
\]

The parameters are set as follows: the evolutionary generations of GA, traditional QGA and higher-order QGA (r=2) are all 200 generations, and the population size is 40. Among them, the crossover probability of GA is 0.7, the probability of mutation is 0.01, the rotation angle of traditional QGA is fixed at 0.01 \( \pi \), and the higher-order QGA (r=2)\( \mu = 0.9918 \). The objective function is the fitness function, and 9 simulation experiments were carried out for each equation system with GA, traditional QGA and higher-order QGA (r=2).

The results of the classical genetic algorithm (GA) are shown in Table 3, the data in Table 3 are counted, and the statistical results are shown in Table 4. Figure 2 shows the result of the 200-generation evolution process of the traditional quantum genetic algorithm.

Table 3. The results of GA.

| Algorithm | The optimal result | The worst result | The biggest evolution generation | The smallest evolution generation | The average evolution generation |
|-----------|--------------------|------------------|----------------------------------|----------------------------------|--------------------------------|
| CQGA      | (3.9861, 2.9547, 0.9790, -0.182) | (3.8986, 3.2844, 1.1237, -4.258) | 148                              | 90                               | 117                            |

Figure 2. Evolution of GA.
Figure 3. Evolution of CQGA.

It can be seen from the results of 9 runs of the above GA that, due to the randomness of the algorithm itself, the results of each run are different, but the solution obtained by the 9 runs is close to the optimal value. But GA is affected by the length of the binary code, that is, the greater the accuracy, the greater the code length. GA algorithm is affected by the initial population, and the early evolution speed may be fast, but it is also affected by the population size. The evolutionary algebra is generally over 100 generations, and the algorithm efficiency is not high.

The results of the traditional quantum genetic algorithm are shown in Table 5, the data in Table 5 are counted, and the statistical results are shown in Table 6. Figure 3 shows the result of the 200-generation evolution process of the traditional quantum genetic algorithm.

Table 5. The results of CQGA.

| Times | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|-------|----|----|----|----|----|----|----|----|----|
| x     | 3.9919 | 3.9688 | 4.008 | 3.9764 | 4.003 | 4.009 | 3.9898 | 3.9907 | 3.984 |
| y     | 2.9687 | 2.8281 | 3.004 | 2.875 | 2.9687 | 2.9921 | 2.9686 | 2.9849 | 2.967 |
| z     | 0.8944 | 0.4686 | 1.0128 | 0.6094 | 0.9187 | 1.001 | 0.890 | 0.9374 | 0.873 |
| The optimal value | -0.84 | -8.83 | -0.017 | -6.454 | -0.959 | -0.272 | -0.875 | -0.308 | -1.07 |
| Evolutionary algebra | 50 | 60 | 40 | 80 | 60 | 70 | 80 | 50 | 70 |

Table 6. The statistical results of CQGA.

| Algorithm | The optimal result | The worst result | The biggest evolution generation | The smallest evolution generation | The average evolution generation |
|-----------|--------------------|------------------|----------------------------------|----------------------------------|---------------------------------|
| CQGA      | (4.008, 3.004, 1.0128, -0.017) | (3.9688, 2.8281, 0.4686, -8.83) | 80 | 40 | 63 |

It can be seen from the above results that running the traditional quantum genetic algorithm 9 times, the solutions of the above equations are all close to x=4, y=3, z=1. According to the evolution process diagram of the traditional quantum genetic algorithm, the algorithm is Convergent, but due to the randomness of the quantum genetic algorithm itself, the results are different each time. The results of the second and fourth runs were too different from those of the 7 groups. We can speculate that these two runs may fall into the local optimal solution. The traditional quantum genetic algorithm needs to determine the value and direction of the quantum revolving gate based on a look-up table. Because the look-up table involves the judgment of multiple conditions, it affects the efficiency of the algorithm,
resulting in the optimal solution around the 70th generation, and the optimal algebra is 40 generations, and the evolution process is slower between 20 and 50 generations.

The experimental results of higher-order QGA (r=2) are shown in Table 7. A general survey of the experimental data in Table 7 is carried out, and the statistical results are shown in Table 8. Figure 4 shows the results of the 200-generation evolutionary process of higher-order QGA (r=2).

Table 7. The results of higher-order QGA (r=2).

| Times | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| X1    | 3.9963| 3.9901| 3.9689| 3.9078| 3.9984| 4.010 | 3.9798| 4.1053| 3.9975|
| X2    | 2.9847| 2.8431| 2.9215| 2.9117| 3.007 | 2.9687| 2.9486| 2.8643| 2.8754|
| X3    | 0.9089| 0.9395| 0.9247| 0.9164| 1.006 | 0.8790| 0.9497| 0.9258| 0.9748|

The optimal value: -0.269, -0.119, -0.247, -0.165, -0.007, -0.297, -0.251, -0.412, -0.217

Evolutionary algebra: 35, 30, 45, 40, 30, 50, 45, 50, 40

Table 8. The statistical results of higher-order QGA (r=2).

| Algorithm | The optimal result | The worst result | The biggest evolution generation | The smallest evolution generation | The average evolution generation |
|-----------|--------------------|------------------|----------------------------------|----------------------------------|---------------------------------|
| QGA       | (3.9984, 3.007, 1.006, -0.007) | (4.1053, 2.8643, 0.9258, -0.412) | 50                               | 30                               | 40.5                            |

It can be seen from the above results that the solution of the system of equations obtained by the 9th order of the higher-order QGA (r=2) is closer to $x=4, y=3, z=1$, indicating that the higher-order QGA (r=2) is also convergent. The randomness of the quantum genetic algorithm still exists, and it did not make the improved results better than the optimal solution of the previous algorithm, and also makes the results of each run different, but the results of 9 runs are more concentrated than the results of traditional QGA, the difference between the optimal values is small. The higher-order QGA (r=2) evolutionary update operation does not involve multiple conditional judgments of the traditional revolving gate look-up table. The algorithm obtains the optimal solution in about 40 generations, which requires fewer generations and shorter running time than traditional quantum genetic algorithms. The higher-order QGA (r=2) proposed in this paper embodies a powerful solution ability in solving nonlinear equations, and can achieve the purpose of rapid convergence to the optimal solution.
4. Conclusions
This article introduces the basic theories and methods of higher-order QGA. Based on the superposition principle, quantum entanglement property is added to the algorithm, which improves the efficiency of searching the optimal solution. This paper takes classic genetic algorithm, traditional quantum genetic algorithm and high-order quantum genetic algorithm (r = 2) as examples, and performs 9 tests on the nonlinear equations.

Simulation experiments show that the higher-order QGA (r=2) evolutionary update operation does not involve multiple conditional judgments of the traditional revolving gate look-up table, and requires less algebra and shorter running time than traditional quantum genetic algorithms, reflecting powerful solutions ability to converge quickly.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (No. 61972413, No.61701539).

References
[1] Roy, U., Roy, S., & Nayek, S. (2014). Optimization with quantum genetic algorithm. International Journal of Computer Applications, 102(16), 1-7.
[2] Holland JH. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence[M],U Michigan Press,1975.
[3] S. Y. Yang, F. Liu, and L. C. Jiao, “Novel genetic algorithm based on the quantum chromosome,” Journal of Xidian University, vol. 31, no. 1, pp. 76–81, 2004.
[4] Han, K. H., & Kim, J. H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE transactions on evolutionary computation, 6(6), 580-593.
[5] Zhen-Quan, Y. J. A. Z. (2003). Actuality of Research on Quantum Genetic Algorithm [J]. Computer Science, 11.
[6] Nowotniak, R. and Kucharski, J. (2014), “Higher-Order Quantum-Inspired Genetic Algorithms”, Federated Conference on Annals of Computer Science and Information Systems, Vol. 2, pp. 465-470, DOI: 10.15439/2014F99
[7] Abbasy, N. H., & Ismail, H. M. (2009). A unified approach for the optimal PMU location for power system state estimation. IEEE Transactions on power systems, 24(2), 806-813.