1. Introduction

Component assignment problem (CAP) is critical for the research on system reliability, which was firstly put forward by Derman [1]. Many scholars have performed comprehensive studies on CAP with different algorithms.

Since importance measure can find the weakness of the system, the highest reliable components are probably assigned into most important position to optimize the system reliability. By analyzing the error terms, Lin and Kuo [2] established a LK algorithm for lin/con/k/n system optimization, which can assign components one by one into the system based on the Birnbaum Importance value.

According to the characteristics of Genetic algorithm (GA), GA can break through the limit of local optimal solution and find the global optimal or approximate optimal solution in a large scaled search space. And the concurrency of quantum computing can improve the computing speed largely. With the rapid development of computer technology, combining importance measure with quantum computing and GA is meaningful to improve the efficiency and accuracy for CAP.

In this paper, we thus introduce a Birnbaum importance based quantum genetic algorithm (BIQGA) to solve the CAP. And then the performance of the BIQGA and general quantum genetic algorithm (QGA) is compared.

2. Birnbaum importance based quantum genetic algorithm

2.1 Process of BIQGA algorithm

The process of the BIQGA algorithm is shown as follows:

(1) Determine the objective function and solution space of the practical CAP pattern.

(2) Determine the coding method and code space.

(3) Generate an initial population Q(t) with N individuals, and All the probability amplitude can be initialized with \((1/\sqrt{2},1/\sqrt{2})\), and every state is equiprobable, and then set \(t=0\).

(4) Perform measurement on initial population; and get the state \(P(t)\) and its rotation matrix \(U(t)\).

(5) Evaluate the fitness of each state \(P(t)\).


(7) After the local search, measure the state and record the optimal state, its fitness value and the corresponding rotation matrix \(B(t)\).

(8) If the current population satisfies the termination condition, stop and output the optimal state and its fitness value.

(9) If the termination condition is not satisfied, save the optimal state and its fitness value, and perform rotation \(B(t)\) on \(Q(t)\) to get the \(Q(t+1)\).

(10) Perform measurement on \(Q(t+1)\) and the state \(P(t+1)\).

(11) Evaluate the fitness of each state in \(P(t+1)\).

(12) Perform the BI-based local search on the optimal state in \(P(t+1)\), if the result of search is superior to the saved optimal state, the relevant value of saved optimal state will be replaced by the current optimal state; otherwise, the optimal state won't be updated, and perform the atavism, then set \(Q(t+1)=Q(t)\), \(B(t+1)=B(t)\).

(13) Set \(t=t+1\), and return to step 7.

Fig. 1 Process of the BIQGA algorithm

3. Simulation study

3.1 experiment design

The reliability of a system depends not only on its inherent structure, but also on the reliabilities of components and the assigned positions in the system. We choose lin/con/k/n system as the research object. The algorithm should be required to test 3 types of components – low, high, arbitrary
reliable components. The reliabilities of components are randomly distributed on \([0.01, 0.2]\), \([0.8, 0.99]\) and \([0.01, 0.99]\), respectively. Considering a single instance may have deviation, we use 100 instances and regard them as an experiment for the system with one type of components. The symbols of the small system and large system are shown in the Table 1 and Table 2 respectively.

### Table 1 Symbol of small system

<table>
<thead>
<tr>
<th>F system symbol</th>
<th>G system symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>lin/con/3/7</td>
<td>g1</td>
</tr>
<tr>
<td>lin/con/3/8</td>
<td>g2</td>
</tr>
</tbody>
</table>

### Table 2 Symbol of large system

<table>
<thead>
<tr>
<th>F system symbol</th>
<th>G system symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>lin/con/3/20</td>
<td>G1</td>
</tr>
<tr>
<td>lin/con/4/20</td>
<td>G2</td>
</tr>
<tr>
<td>lin/con/3/30</td>
<td>G3</td>
</tr>
<tr>
<td>lin/con/4/30</td>
<td>G4</td>
</tr>
<tr>
<td>lin/con/5/30</td>
<td>G5</td>
</tr>
<tr>
<td>lin/con/3/50</td>
<td>G6</td>
</tr>
<tr>
<td>lin/con/4/50</td>
<td>G7</td>
</tr>
<tr>
<td>lin/con/5/50</td>
<td>G8</td>
</tr>
</tbody>
</table>

The numerical experiment is implemented to compare the performance of the BIQGA and QGA. For the small system, we get the optimal solution by enumeration method, and record the comparative indices of BIQGA and QGA, such as computing time, mean standardized system reliability (MSSR) and achievement times. For the large system, we choose 1000 assignments by stochastic method for the 18 systems respectively, and regard the assignment with maximum system reliability as the approximate optimal solution. If the system reliability is bigger than the approximate optimal solution, the achievement times will increase 1. We will record the time of two algorithms, MSSR and the achievement times.

### 3.2 Simulation results

For the small system, because of the time of QGA is larger than that of BIQGA, so we compare the MSSR and the achievement times of the two algorithms, as shown in the Figure 2 and 3.

For the large system, the achievement times for BQA and BIQGA are nearly 100, so we compare the percentage of time difference and the difference value of MSSR, as shown in Figure 4 and 5.

![Fig. 2 The achievement times of QGA and BIQGA](image)

![Fig. 3 The MSSR of QGA and BIQGA](image)

![Fig. 4 MSSR difference for large systems](image)

![Fig. 5 Percentage of time difference for large systems](image)

### 4. Conclusions

Through the simulation experiment, the accuracy and efficiency of BIQGA is superior to the QGA as a whole. Compared with QGA, the performance of BIQGA is better in the high reliable-f system and the low reliable-g system.

**Acknowledgment**

The authors gratefully acknowledge the financial supports for this research from the National Natural Science Foundation of China (Grant No. 71471147) and the 111 Project (Grant No. B13044).

**References**

