A New Quantum Immune Evolutionary Algorithm and Its Application

ZHANG XIANGXIAN, QU HONGJIAN
Glorious Sun School of Business & Management
Donghua University
Shanghai, 200051, P.R.China
top_power@yeah.net

Abstract: - From recent research on combinatorial optimization of the complex function, quantum-inspired evolutionary algorithms (QEAs) were proved to be better than traditional evolutionary algorithms. However, they are easy to be trapped to prematurity, and the operations in QEAs lack the capability of meeting an actual situation, so that some torpidity often appears when solving problems. In this paper, a new improved quantum immune evolutionary algorithm (QIEA) is proposed to overcome the shortcoming of the conventional QEAs. The new QIEA combines the main mechanisms of immune theory. Experiments on the knapsack problem are compared with other evolutionary algorithms. The result indicates that the performance of new algorithm is superior to the others.

Key-Words: - quantum computing, evolutionary algorithm, qubit, immune, clone, 0/1 knapsack problem

1 Introduction
Evolutionary algorithms are principally a stochastic search and optimization method based on the principles of natural biological evolution. Compared with traditional optimization methods, they are robust, global in operation and so on.

The quantum evolutionary algorithm [1] recently proposed is a new probability optimization method based on quantum computing theory and the thinking of evolutionary algorithms. QEAs have lots of advantages such as better convergence, global search capability, and smaller swarm size than the traditional evolutionary algorithms, so they can triumphantly applied to the optimization of continuous functions of multi-peak [2-4]. However, when QEA is used to deal with the complex functions, it updates qubits based on the mutation of quantum gate, and then builds the binary solutions from the state of qubits. The essence of the whole process is a probabilistic operation, which has a very great randomicity and blindness. So when the qubits have a chance to evolve, this is inevitably to become degradation. At the same time, if we use the QEAs to deal with different kinds of problems, which is included lots of character information. However, QEAs could not make the most of this information, which will make an error frequently.

In this paper, the mechanisms of immune theory will be imported into QEA, and create a new evolutionary algorithm-quantum-inspired immune evolutionary algorithm (QIEA). QIEAs reserve good characteristics of the QEAs, and try to refine the character information from different kinds of problems adaptively, so that the new algorithm can improve its performance. At last, the new QIEA is applied into the knapsack problem, it is testified that this new algorithm improves from two points: search capability and computing time.

2 Overview of QEA
As a new research field, QEA combines the quantum computing with genetic evolutionary algorithm. QC deals with investigations on quantum mechanical computers and quantum mechanics like qubits representation and superposition of states. QC can process huge numbers of quantum states simultaneously in parallel.

QEA relies on a basic concept which is quantum bit or qubit. A qubit can take value 0, 1 or a superposition of the two at the same time. Its state can be defined by:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$  \hspace{1cm} (1)

Where $|0\rangle$ and $|1\rangle$ represent the classical bit
values 0 and 1 respectively; and $\alpha$, $\beta$ are complex numbers that specify the probability amplitudes of the corresponding states. $|\alpha|^2$ gives the probability that the qubit will be found in the 0 state and $|\beta|^2$ gives the probability that the qubit will be found in the 1 state. Normalization of the state to unity guarantees

$$|\alpha|^2 + |\beta|^2 = 1$$

(2)

The state of a qubit can be changed by the operation with a quantum gate. Inspired by the concept of quantum computing, QEA is designed with a novel Q-bit representation, a Q-gate as a variation operator, and an observation process. A Q-bit individual as a string of $n$ Q-bits is defined as

$$q_j = [\alpha_1,j\alpha_2,j\ldots\alpha_m,j]$$

$$[\beta_1,j\beta_2,j\ldots\beta_m,j]$$

$j = 1, 2, \ldots, n$

(3)

The following rotation gate is used as a Q-gate in QEA, such

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

(4)

Where $\theta$ is a rotation angle of each Q-bit toward either 0 or 1 state depending on its sign.

The structure of QIEA implies that most operations of QEA are based on the probabilistic, and QEA can not use any prior knowledge in the whole computing process.

### 3 An Improved Quantum-inspired Immune Evolutionary Algorithm

#### 3.1 The description of the QIEA

Biological immune system has a lot of characters such as diversity, immune memory, self-organization, self-learning, adaptive behavior, and robustness. The immune mechanism contains these concepts:

1. Immune recognition;
2. Immune feedback;
3. Clonal selection;
4. Immune adjustment

In this paper, the QIEA combines quantum computing and immune mechanism, and it is made up of a new immune operator, which operations have two parts: immune recognition and clonal selection. Immune recognition means we can use the prior knowledge in a problem-solving situation, and then refine the basic characteristic information which can help us to improve the performance of QIEA. Clonal selection is to increase the diversity of the swarm in order to prevent the degradation of the swarm. In particular, they are as follows:

1. In this paper, evolutionary strategy with niche was used to initialize quantum swarm, which means the qubits’ probability space will be divided into $N$ sub-population space as follows:

$$\begin{bmatrix}
\alpha_k \\
\beta_k
\end{bmatrix} = \begin{bmatrix}
\frac{\sqrt{i/N}}{\sqrt{1-i/N}} \\
\end{bmatrix}$$

(5)

The whole initialization process will be according to the formula (5), and each sub-population will be initialized into the qubits which have the same probability. The formula (5) describes the initialization value of the $i$ sub-population in the whole N sub-population.

2. Suppose the individual is $X$, when QIEA is in immune recognition, we use the prior knowledge to modify the qubits of $X$ in order to make the individual has more better fitness in a higher probability. Actually, if the swarm is represented as

$$s = (x_1, x_2, \ldots, x_n)$$

It means that we should select some individuals in term of a probability $\alpha$, that is to say $n_s = \alpha n$ individuals will be selected to carry out immune recognition.

3. When the traditional artificial immune clone algorithm is in the phase of clone copy, the scale of cope is fixed. In this paper, the scale of cope is dynamic.

$$N_i' = \text{round}(n \times \frac{\text{fit}(p_i')}{\sum_{j=1}^{n} \text{fit}(p_j')})$$

$$i = 1, 2, \ldots, n$$

(6)

Where $N_i'$ is the scale of clone of each individual in the $t$ iterative time, and the scale of clone of the swarm is:

$$N_C' = \sum_{i=1}^{n} \sum_{j=1}^{n} \text{round}(n \times \frac{\text{fit}(p_i')}{\sum_{j=1}^{n} \text{fit}(p_j')})$$

$$\approx \text{round}(n \times \frac{\sum_{i=1}^{n} \text{fit}(p_i')}{\sum_{j=1}^{n} \text{fit}(p_j')}) = n$$

(7)
When every individual in the swarm is in the phase of clone, the number is not fixation in formula (6), which means that more excellence the father individual is, more bigger the scale of son individual is. At the same time, every sub-swarm actually can represent a larger search space, at the subsequent steps; it becomes mutation in its low bit. All of this can make sub-swarm distribute around father individual’s search space, which can increase the diversity of the solution space of functions and avoid trapping into local peak effectively. So the algorithm can get balance between depth search and breadth search by doing this and improve its ability to solve the function. At the same time, the scale of swarm does not change obviously in formula (7), which can make sure the efficiency of the algorithm.

(4) Before a Q-gate is employed to update a Q-bit individual in this paper, the new algorithm will make mutation in its low bit(multi-point mutation), which can represent a larger search space and increase the diversity of the solution space of functions.

(5) When this new algorithm is on its run, it will identify automatically. If the swarm’s average fitness has a little change in \((t + 1)\) th iteration in contrast with the fitness which was got in \(t\) th iteration, then the new algorithm will create \(round(\beta \times n)\) new individual random and join them in the swarm instead of the individual except the best individual. In this way, the algorithm can make trouble in the swarm in order to make sure the diversity of the search space. In this case, \(\beta\) is an adaptive parameter, which is negatively correlated with \(\varepsilon\). \(n\) is the scale of swarm in \((t + 1)\) iteration.

### 3.2 The procedure of QIEA for the particular questions

Begin
1. \(t = 0\) initialize \(Q(t) = \{q'_1, q'_2, \ldots, q'_n\}\) according to the formula (5), define a empty memory storeroom;
2. Make \(R(t)\) by observing the states of \(Q(t)\);
3. Immune recognition;
4. Clone the individual which included in \(R(t)\), then get a new swarm \(R'(t)\);
5. Each sub-swarm mutation in their low bit;
6. Evaluate \(R(t)\), store the best solutions in memory storeroom.;

7. While the termination is false
   (1) Update the swarm by Q-gate to get the new swarm;
   (2) If \(\varepsilon\) has a little change, update the swarm random;
   (3) \(t = t + 1\) go to 2.

End

When QIEA is in the step of mutation, there are often three mutation methods\(^6\).

1. Traditional mutation;
2. Build the values of the updated Q-bit \((\alpha, \beta)\) random;
3. Use the appropriate Q-gate, by which operation the Q-bit should be updated.

In the formula (4), Q-gate has a parameter \(\theta\), which can get as follows\(^7\):

<table>
<thead>
<tr>
<th>(x_i)</th>
<th>(best)</th>
<th>(f(c_i) &gt; f(best))</th>
<th>(\Delta\theta_i)</th>
<th>(s(\alpha, \beta))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>false</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>true</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>false</td>
<td>0.05π</td>
<td>0.05π</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>true</td>
<td>0.05π</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>false</td>
<td>0.025π</td>
<td>0.025π</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>true</td>
<td>0.01π</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>false</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>true</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In the reference (7), where \(\theta_i = s(\alpha, \beta) \times \Delta\theta_i\), \(x_i\) is the \(i\) th bit of current individual, \(best_i\) is the best individual’s \(i\) th bit in current swarm, \(f(\cdot)\) is fitness function. \(\Delta\theta_i\) is the rotation angle’s value, which can control the convergence scope of the algorithm. \(s(\alpha, \beta)\) is the rotation angle’s direction, which can control the convergence speed of the algorithm.

### 4 Experimental results

0/1 knapsack problem is an effective criterion to verify the performance of all kinds of algorithms. 0/1 knapsack problem is a typical combinatorial optimization problem, it belongs to a NP-complete, we can describe it as follows:

Suppose something related to travel, its number is \(n\), the quality and value of each one is \(w_i(w_i > 0)\), \(c_i(c_i > 0)\), \((i = 1, 2, \ldots, n)\). The capacity of this knapsack is \(V(V > 0)\), then to solve an answer \(x = (x_1, x_2, \ldots, x_n)\) to make sure
the total value of this knapsack which is loaded with a lot of things is the most largest. We can describe this question as follows:

\[
\text{max } f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} c_i x_i
\]

Subject to \( \sum_{i=1}^{n} w_i x_i \leq V \), \( x_i \in \{0,1\} \), \( (i = 1,2, \ldots, n) \),

\( x_i \) is a decision variable, if \( x_i = 1 \), which means res \( i \) has loaded in this knapsack; if \( x_i = 0 \), which means res \( i \) has not loaded in this knapsack.

In order to get a result, all the test data in this experiment have the strong correlation between weight and value\(^[8]\).

\( w_i = \text{random}([1,10]) \) \quad \text{(Equally)}

\( c_i = w_i + 5 \)

And the average capacity of the knapsack:

\[
V = 0.5 \sum_{i=1}^{n} w_i
\]

Before we solve knapsack problem, we have some prior knowledge, which are:

(1) When people begin to install a knapsack, they will choose bigger value of “profit/weight” firstly;

(2) When people begin to eliminate the capacity of knapsack, they will choose smaller value of “profit/weight” to discard firstly.

So this is character information, and we will use them in our new QIEA. In order to compare the new QIEA with others in this paper, we adopt traditional GA, QEA, and QIEA to validate the knapsack problem.

The matlab 7.0 tool was adopted for the implementation of the approach described previously. The number in knapsack problem is 250, and 500 respectively, and the size of population is 100 in GA and QEA. The size of population is 50 in QIEA. The three evolutionary algorithms run 50 times and get their statistical results, respectively. The statistical results are in table 2. We can find that the QIEA’s performance improve obviously compared with GA and QEA.

### Table 2 statistical results

<table>
<thead>
<tr>
<th>Numbe</th>
<th>Value</th>
<th>GA</th>
<th>QEA</th>
<th>QIEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>Best</td>
<td>1653.4</td>
<td>1681.3</td>
<td>1710.6</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>1507.6</td>
<td>1595</td>
<td>1671.5</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>1302.7</td>
<td>1421.4</td>
<td>1578.9</td>
</tr>
<tr>
<td>500</td>
<td>Best</td>
<td>2722.3</td>
<td>2907.2</td>
<td>3123.2</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>2599.4</td>
<td>2767.1</td>
<td>3107.6</td>
</tr>
<tr>
<td></td>
<td>Worst</td>
<td>2278.2</td>
<td>2635.9</td>
<td>2918.5</td>
</tr>
</tbody>
</table>

## 5 Conclusions

This paper proposed a new QIEA, inspired by the concept of immune mechanism. Every individual of each chromosome in this QIEA will make full use of the character information of particular problem, and they recur to the new immune operator which combines immune recognition and clonal selection to solve problems. The experiment results of knapsack problem show QIEA overcomes the shortcoming of the traditional evolutionary algorithm, and can improve the performance evidently.

### References:


