A Retroactive Quantum-inspired Evolutionary Algorithm

Zakaria Laboudi∗ - Salim Chikhi†

∗SCAL group, MISC Laboratory
† University of Constantine – Route de Ain El Bey 25017 Constantine. Algeria.
laboudizak@yahoo.fr, slchikhi@yahoo.com

ABSTRACT
This study outlines some weaknesses of existing Quantum-inspired Evolutionary Algorithms (QEA) by explaining how a bad choice of the rotation angle of qubit quantum gates can slow down optimal solutions discovery. A new algorithm, called Retroactive Quantum inspired Evolutionary Algorithm (rQEA), is proposed. With rQEA the rotation of individual’s amplitudes is performed by quantum gates according to a retroactive strategy making the algorithm more adaptive and thus leading to a good balance between intensification and diversification. Our algorithm was tested and compared to Classical Genetic Algorithm (CGA) and to QEA on two benchmark problems. Experiments have shown that rQEA performs better than both CGA and QEA in terms of genericity, adaptability and accuracy.

Keywords: Quantum-inspired Evolutionary Algorithm, Quantum Computing, Knapsack Problem, Onemax Problem, Adaptive Memory Programming, Retroaction.

1. INTRODUCTION
Quantum computation is a newly emerging interdisciplinary science of information science and quantum science. The first quantum algorithm was proposed by Shor in 1994 [1], for number factorization. Grover [2] also proposed a quantum algorithm for random search in databases, the complexity of its algorithm was reduced to be of the order of \( N^{1/2} \). More recently, quantum computation has attracted a wide attention, and it becomes a very interest research field.

Quantum-inspired evolutionary algorithm (QEA) is a combination between conventional genetic algorithms (CGA) and quantum computing. There were some efforts to use QEA for exploring search spaces; we quote for example [3] where a QEA was used to solve the knapsack problem and [4] who proposed a parallel version of QEA. In [5], a QEA was also used to solve the binary decision diagram ordering problem. More recently, QEA where used to evolve cellular automata rules (CA) [6] [7] to solve the density classification problem.

In this work, we propose a new retroactive strategy for the rotation of individual’s amplitudes in the QEA process to extract some computational abilities to perform processing in an effective and an efficient manner. We have considered the classic 0/1 knapsack problem and the maxOnes problem.

This paper is organized as follows. Section 2 describes the basic concept of quantum computing and QEA principles. A brief description of the used benchmark problems is presented in section 3. Section 4 tackles the problem of choosing rotation angle. A full description of the proposed algorithm (rQEA) is presented in section 5. In section 6 we summarize and analyze the experimental results. We finish the paper by concluding remarks follow and some perspectives in section 7.

2. QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM (QEA)
2.1. QUANTUM COMPUTING
In quantum computing, the smallest unit of information storage is the quantum bit (qubit) [3]. A qubit can be in the state 1, in the state 0 or in a superposition of both. The state of a qubit can be represented as [3]:

\[
|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle \tag{1}
\]

Where \( |0\rangle \) and \( |1\rangle \) represent the values of classical bits 0 and 1 respectively, \( \alpha \) and \( \beta \) are complex numbers satisfying:

\[
|\alpha|^2 + |\beta|^2 = 1 \tag{2}
\]

\( |\alpha|^2 \) is the probability where a qubit is in state 0 and \( |\beta|^2 \) represents the probability where a qubit is in state 1. A quantum register of \( m \) qubits can represent \( 2^m \) values simultaneously. However, when the 'measure' is taken, the superposition is destroyed and only one of the values becomes available for use. That's why we think that quantum computers can be used mainly in applications involving a choice among a multitude of alternatives.

In general, a quantum algorithm has less complexity than its classic equivalent algorithm through the concept of quantum superposition. Among the most famous quantum algorithms we quote Shore’s algorithm for number factorization (1994) [1] and Grover’s algorithm for research in a non sorted database (1996) [2]. Both algorithms have solved problems which their complexity was reduced.

2.2. QEA PRINCIPLES
QEs are a combination between CGA and quantum computing. They are mainly based on qubits and states...
superposition of quantum mechanics. Unlike the classical representation of chromosomes (binary string for instance), here they are represented by vectors of qubits (quantum register). Thus, a chromosome can represent the superposition of all possible states. The structure of a QEA is illustrated in figure 1 [3]:

2.2.1. Structure of quantum chromosomes
A chromosome is simply a string of m qubits that forms a quantum register. Figure 2 shows the structure of a quantum chromosome.

2.2.2. Initializing the population
The easiest way to create the initial population is to initialize all the amplitudes of qubits by the value 1 / (2 \(^\frac{1}{2}\)) [3]. This means that a chromosome represents all quantum superposition states with equal probability.

2.2.3. Evaluation of individuals
The role of this phase is quantifying the quality of each quantum chromosome in the population to make a reproduction. The evaluation is based on an objective function that corresponds to each individual, after measuring, an adaptation value. It permits to mark individuals in the population.

2.2.4. Quantum genetic operations
1. Measuring chromosomes: In order to exploit effectively superposed states of qubits, we have to observe each qubit. This leads us to extract a classic chromosome. The aim is to enable the evaluation of each quantum chromosome.

![Figure 4. Measured chromosome.](image)

A simple way to implement this function is given by the following pseudo code:

```
Function measure ()
begin
    r := get r in [0,1] ;
    if (r > \(\alpha\))
        return 1 ;
    else
        return 0 ;
    end if
end
```

2. Interference: This operation allows modifying the amplitudes of individuals in order to improve performance. It mainly consists of moving the state of each qubit in the sense of the value of the best solution. This is useful for intensifying the search around the best solution. It can be performed using a unit transformation that allows a rotation whose angle is a function of the amplitudes (\(a_i, b_i\)) and the value of the corresponding bit in the reference solution. The value of the rotation angle \(\delta\theta\) has to be chosen so that to avoid premature convergence. It is often empirically determined and its direction is determined as a function of the values of \(a_i, b_i\) and the value of the qubit located at the position \(i\) in the individual being modified [5].

3. Qubit rotation gates strategy: The rotation of individual’s amplitudes is performed by quantum gates. Quantum gates can also be designed in accordance with the present problem. The population \(Q(t)\) is updated with a quantum gates rotation of qubits constituting individuals. The rotation strategy adopted is given by the following formula:
The knapsack problem is one of the most combinatorial. Selecting from among various items those which are most profitable, given that the knapsack has a limited capacity. There are many types of knapsack problem, so the simplest one is called 0/1 knapsack problem. It is described as: given a set of \( m \) items and a knapsack, find a subset of the items so as to maximize the profit \( f(x) \) as shown in formula 4:

\[
f(x) = \sum_{i=1}^{m} p_i x_i
\]

Subject to:

\[
f(x) = \sum_{i=1}^{m} w_i x_i \leq C
\]

Where \( x = (x_1, x_2, \ldots, x_m) \), \( x_i \) is 0 or 1, \( p_i \) and \( w_i \) are the profit and the weight of the \( i \)-th item. \( C \) is the capacity of the knapsack.

3.1. THE KNAPSACK PROBLEM

The knapsack problem is one of the most combinatorial algorithms. The knapsack problem can be described as selecting from among various items those which are most profitable, given that the knapsack has a limited capacity. There are many types of knapsack problem, so the simplest one is called 0/1 knapsack problem. It is described as: given a set of \( m \) items and a knapsack, select a subset of the items so as to maximize the profit \( f(x) \) as shown in formula 4:

\[
f(x) = \sum_{i=1}^{m} p_i x_i
\]

Subject to:

\[
f(x) = \sum_{i=1}^{m} w_i x_i \leq C
\]

Where \( x = (x_1, x_2, \ldots, x_m) \), \( x_i \) is 0 or 1, \( p_i \) and \( w_i \) are the profit and the weight of the \( i \)-th item. \( C \) is the capacity of the knapsack.

3.2. THE ONEMAX PROBLEM

The OneMax Problem consists of maximizing the number of ones of a bit string and the global optimum is often obtained from a lookup table having the following format:

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( b_i )</th>
<th>( f(x) \geq f(b) )</th>
<th>( \Delta \theta_i )</th>
<th>( s(\alpha_i \beta_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \Delta \theta_1 )</td>
<td>( \xi )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( \Delta \theta_2 )</td>
<td>( \xi )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( \Delta \theta_3 )</td>
<td>( \xi )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \Delta \theta_4 )</td>
<td>( \xi )</td>
</tr>
</tbody>
</table>

Table 2. Look-up table for quantum gates rotation

\( x_i \) and \( b_i \) are the \( i \)-th bits of \( x \) and \( b \) (the best solution), respectively. \( f \) is the fitness function and \( s(\alpha_i \beta_i) \) is the sign of the rotation angle \( \theta_i \).

One can ask how to set \( \Delta \theta \) values? Really it is a very hard task. It has been some efforts to avoid this problem by [9] but to date we have no knowledge of a systematic mechanism that can determine them. In literature, the analysis of most of papers that deal QEA shows that the determination of \( \Delta \theta \) values is performed empirically i.e. intuitively by testing the algorithm for many times until achieving the best values. Following a suggestion from [9], if it was ambiguous to select a positive or a negative number for the values of the angle parameters, it is recommended to set the values to 0. In [9], for the knapsack problem, \( \Delta \theta_3 = 0.01 \pi \), \( \Delta \theta_5 = -0.01 \pi \), and 0 for the rest were used. The magnitude of \( \Delta \theta_i \) has an effect on the speed of convergence, but if it is too big, the solutions may diverge or converge prematurely to a local optimum. The values from 0.001\( \pi \) to 0.05\( \pi \) are recommended for the magnitude of \( \Delta \theta_i \), although they depend on the problems. The sign of \( \Delta \theta_i \) determines the direction of convergence [9].

For this reason, we have thought to invent a generic rotation gates strategy that can be applied for a large wide of problems.

5. RETROACTIVE QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

In this section we present our improved novel algorithm called Retroactive Quantum-inspired Evolutionary Algorithm (rQEA).

5.1. ADAPTIVE MEMORY PROGRAMMING

The concept of adaptive memory programming [10] is invented by observing characteristics similarity of the best methods of solving combinatorial optimization problems. It focuses on three basic concepts: memory, intensification and diversification. The memory here is the information collected by the algorithm, on which it intensifies and diversifies. The memory here is the information collected by the algorithm, on which it depends on. The intensification is the use of available information to improve its relevance. Diversification is the search for new information, in order to increase knowledge of the problem [11].

5.2. DESCRIPTION OF THE ALGORITHM

In order to overcome the problem of choosing rotation angle, we introduce a new rotation gates strategy based on retroaction. The amplitudes of quantum chromosomes are updated according to the look-up table 3.

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( b_i )</th>
<th>( f(x) \geq f(b) )</th>
<th>( \Delta \theta_i )</th>
<th>( s(\alpha_i \beta_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \Delta \theta_1 )</td>
<td>( \xi )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( \Delta \theta_2 )</td>
<td>( \xi )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( \Delta \theta_3 )</td>
<td>( \xi )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \Delta \theta_4 )</td>
<td>( \xi )</td>
</tr>
</tbody>
</table>

Table 3. Look-up table for retroactive quantum gates rotation
According to the lookup table, one can easily remark that this strategy improves, for each individual, the amplitudes of qubits that are bad according to an angle $\delta \theta_1$ while it decreases those that are good according to an angle $\delta \theta_2$. The modification of the amplitudes of qubits is done according to the signs of the amplitudes, the best solution and the solution extracted by the individual container. It is natural that $\delta \theta_1 > \delta \theta_2$ because decreasing amplitudes serves only to correct stochastic errors to avoid a genetic drift and to ensure a genetic diversity in the population. RQEA is:

1. **Adaptive:** here the memory is the quantum chromosomes. The intensification is ensured by the interference operation while the diversification is ensured by the measure operation and the rotation gates strategy.
2. **Retroactive:** the adopted rotation gates strategy leads the quantum chromosomes to degradation in order to restart the search process again. This is very similar to what happen for optimization by ant colonies inspired algorithms where the pheromone vaporizes [12].
3. **Simple:** Instead of searching of $(\delta \theta_i , +/ -)$ values, $i = 1..8$, rQEA turn the problem to find only two parameters $\delta \theta_1$ and $\delta \theta_2$. More formally, if we set $\delta \theta_2 = \alpha \times \delta \theta_1$ where $\alpha \in [0, 1]$ then, we have only to determine empirically $\alpha$ and $\delta \theta_1$ which is relatively simple.
4. **Generic:** rQEA is generic in that it has only to modify the $\alpha$ and $\delta \theta_1$ parameters depending on the present problem.
5. **Powerful:** rQEA is tested on several problems and show a high accuracy comparing to other evolutionary algorithms.

### 6. EXPERIMENTAL RESULTS

We have executed both algorithms (QEA & rQEA) over 25 runs to solve the two benchmark problems, at each one the concerned algorithm was iterated for a maximum number = 1000 generations. Table 4 and table 5 show the experimental results of the knapsack problem (with 100, 250, 500 and 1000 items) and the oneMax (with $\lambda =100$, $\lambda =250$, $\lambda =500$ and $\lambda =5000$). We note here that executing both algorithms more than 25 runs doesn’t make difference.

### 6.1. PARAMETERS VALUES

For both algorithms (rQEA & QEA) the population size was fixed to be 100. For rQEA, the parameter values are represented in table 2. Experimental results are shown in tables 6, 7 and 8.

<table>
<thead>
<tr>
<th>No. of Items</th>
<th>Method</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>QEA</td>
<td>m b 612.3 628.2</td>
</tr>
<tr>
<td></td>
<td>rQEA</td>
<td>m b 609.0 628.2</td>
</tr>
<tr>
<td>250</td>
<td>QEA</td>
<td>m b 1531.3 1551.9</td>
</tr>
<tr>
<td></td>
<td>rQEA</td>
<td>m b 1529.3 1546.4</td>
</tr>
<tr>
<td>500</td>
<td>QEA</td>
<td>m b 3056.5 3088.5</td>
</tr>
<tr>
<td></td>
<td>rQEA</td>
<td>m b 3063.6 3097.3</td>
</tr>
<tr>
<td>1000</td>
<td>QEA</td>
<td>m b 6031.9 6095.5</td>
</tr>
<tr>
<td></td>
<td>rQEA</td>
<td>m b 6083.4 6140.3</td>
</tr>
</tbody>
</table>

### 6.2. DISCUSSION

Table 6 shows the experimental results of the knapsack problem where the number of items was 100, 250, 500 and 1000. The hardware and software configuration for
both problems was as follows: Intel Pentium 4 (core 2duo 2.0 GHz), 3 Go MB of memory, Windows Vista OS, Java Programming language (JDK 6). In the case of 100 and 250 items, both algorithms QEA and rQEA were equivalent. This because the number of items is so small (we have $2^{100}$ possibility). However, augmenting the number of items (500 and 1000 items where we have $2^{1000}$ possibility) leads rQGA to behave better than QGA.

The oneMax problem can be considered as a good measure of quantum gates rotation strategy power. This is because QEA and rQEA are baesd on qubit representation which is inderministic. For instance, measuring a quantum chromosome twice doesn’t lead to the same result. This is why we think that the maxOne problem is a good way to measure how quantum chromosomes can keep a quantum representation that lead to an appropriate value. The same remarks are noted as those related to the knapsack problem results.

One possible interpretation for those results: when the size of quantum chromosomes is small, the search space is also small. This implies that stochastic errors were not occurred frequently. In the other hand, when the size of quantum chromosomes is big, the search space is also tending to be more and more difficult to explore. This implies that stochastic errors will occurred frequently. This is exactly what happens when the size of quantum chromosomes is increased. We can then judge that QEA are not able to correct stochastic errors. For example, if a qubit amplitude of an current optimal solution has been rotated to a wrong sense, the algorithm continues running without being able to correct this error. This is why also rQEA has behaved better than QEA for big size chromosomes. The retroaction mechanism allows to the quantum chromosome after a number of generations to return to it’s initial form, enabling it’s revaluation and perhaps to correct an error that was occurred.

To check thses issues we have extended the number of generations in the case of oneMax problem to become 2000, the obtained fitness values are presented in table 8. Our algorithm has continued to correct errors contrarily to QEA that has never exceeded the optimal values obtained by rQEA.

7. CONCLUSION

In this study we have tackled the problem of choosing angle rotation of quantum gates by introducing a new retroactive quantum-inspired evolutionary algorithm (rQEA) based on a retroactive rotation gates strategy making the algorithm highly adaptive. Our experimental results have shown that is possible to replace the set of rotation angles by two rotation angles only. rQEA can then be a very promising tool for exploring large search spaces, preserving adaptability, retroaction, efficiency, performance, simplicity and finally genericity. Our future work will focus on how to determine automatically the best values of rQEA parameters. Another perspective of this work is to study the possibility of integrating rQEA into other versions of QEA (as parallel quantum-inspired genetic algorithm (QGA) [13] or versatile quantum-inspired evolutionary algorithm vQEA [8]).

8. REFERENCES
