A Comparative Study of Quantum Evolutionary Algorithm and Particle Swarm Optimization for Numerical Optimization Problems

Tinghuai Ma, Qiaoqiao Yan, Wenxia Xuan, Bin Wang


Abstract

Quantum evolutionary algorithm (QEA) and particle swarm optimization (PSO) are two different types of intelligent optimization algorithm. Many efforts on these two algorithms have progressed actively in recent years. In this paper, six typical and complex benchmark testing functions are applied to verify their abilities for dealing with numerical optimization problems. The results show that QEA has better global search capability, which is manifested very outstandingly for the multi peak function optimization, but its convergence speed is slower. Contrarily, PSO has faster convergence speed, and it can solve certain optimization problem rapidly, especially for single peak functions.

Keywords: Evolutionary Algorithm, Quantum Evolutionary Algorithm, Particle Swarm Optimization, Numerical Optimization

1. Introduction

Many practical problems, which originated from different fields such as scientific research, engineering technology, resource allocation, economic management and so on, can be abstracted as the relevant numerical optimization problems. How to effectively solve the numerical optimization problems always is one of the most important research directions of optimization field. In recent years, intelligent optimization algorithms, such as GA (Genetic Algorithm) [1-2], EP (Evolutionary Programming) [3-4], ES (Evolution Strategies) [5-6], SA (Simulated Annealing) [7-8], Tabu (Tabu Search) [9-10], ACO (Ant Colony Optimization) [11-12], IA (Immune Algorithm) [13-14], PSO (Particle Swarm Optimization) [15-16] and QEA (Quantum Evolutionary Algorithm) [17], have received considerable attention in the solution of numerical optimization problems, and have become a type of key methods in this field. Compared with traditional optimization method, such as calculus-based and enumerative strategies, these algorithms are robust, global in operation, not dependent of the analytic properties of functions, and may be applied generally without recourse to domain-specific heuristics although their performance may be affected by these heuristics.

Among those intelligent optimization algorithms mentioned above, PSO and QEA are two different types of optimization methods. QEA [17-20] combines the advantages of both evolutionary computing and quantum computing. Quantum computing includes concepts like quantum mechanical computers and quantum algorithms. By adopting qubit chromosome as a representation, QEA can represent a linear superposition of solutions due to its probability characteristics. It has rapid convergence and good global search compatibility. And quantum gates are applied to produce new thriving chromosomes. PSO [21-22] is a population-based stochastic optimization technique inspired by social behaviors of birds. It contains a swarm of particles in which each particle includes a potential solution. The particles fly through a multidimensional search space in which the position of each particle is adjusted according to its own experience and the experience of its neighbors. PSO system combines local search methods (through self experience) with global search methods (through neighboring experience), attempting to balance exploration and exploitation. As two novel intelligent algorithms, both QEA and PSO have been proved to be more effective than conventional genetic algorithms.
However, little work has been done to compare their performance for numerical optimization problems. In this paper, six typical and complex benchmark testing functions are applied to analyses and compare their abilities of dealing with numerical optimization problems from the views of optimization results and convergent trend.

The remainder of this paper is organized as follows: In Sec. 2 and Sec.3, the basic principles and procedures of QEA and PSO are addressed, respectively. Then six benchmark testing functions and their geometric properties are described in Sec.4, and the experimental results are drawn and analyzed. Finally, conclusions are made in Sec. 5.

2. Quantum Evolutionary Algorithm

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. Its representation and procedure of QEA are presented in this part.

2.1. Representation

Definition 1: A qubit is defined as the smallest unit of information in QEA, which can be described by a pair of numbers \((\alpha, \beta)\) as \(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}\). A qubit may be in the \(|0\rangle\) state, the \(|1\rangle\) state, or in any linear superposition of the two. The state of a qubit can be represented as \(|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle\), where \(|\alpha|^2 + |\beta|^2 = 1\). \(|\alpha|^2\) gives the probability that qubit will be found in the \(|0\rangle\) state and \(|\beta|^2\) gives the probability that qubit will be found in the \(|1\rangle\) state.

Definition 2: A qubit chromosome as a string of m qubits is defined as:

\[
\begin{bmatrix}
\alpha_1, \alpha_2, \cdots, \alpha_{m-1}, \alpha_m \\
\beta_1, \beta_2, \cdots, \beta_{m-1}, \beta_m
\end{bmatrix}
\]

(1)

where \(|\alpha_i|^2 + |\beta_i|^2 = 1\), \(i = 1, 2, \cdots, m\). A qubit chromosome can represent a linear superposition of states.

Definition 3: The angle-distance \(\Delta \theta\) shows the difference between two qubits \(|\varphi\rangle, |\varphi'\rangle\), which is defined as \(\Delta \theta = \arctan(\frac{\alpha}{\beta}) - \arctan(\frac{\alpha'}{\beta'})\), assuming that \(|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle\), \(|\varphi'\rangle = \alpha'|0\rangle + \beta'|1\rangle\).

For example, the angle-distance between \(|\varphi\rangle\) and \(|0\rangle\) is \(\Delta \theta(|\varphi\rangle, |0\rangle) = \arctan(\frac{\alpha}{\beta})\).

2.2 The procedure of quantum evolutionary algorithm

QEA maintains a population of qubit chromosomes \(Q(t) = (q_1, q_2, \cdots, q_n)\) at generation \(t\), where \(n\) is the size of population, and \(q_j\) (\(1 \leq j \leq n\)) is a qubit chromosome defined as:

\[
q_j = \begin{bmatrix}
\alpha_1^j, \alpha_2^j, \cdots, \alpha_{m-1}^j, \alpha_m^j \\
\beta_1^j, \beta_2^j, \cdots, \beta_{m-1}^j, \beta_m^j
\end{bmatrix}
\]

(2)

Its procedure is similar with CGA (Conventional Genetic Algorithm), which can be described as follows:
In the step of \textbf{initialize} $Q(t)$, $\alpha_j$ and $\beta_j$ of $q_j$ are all initialized with $\frac{1}{\sqrt{2}}$ to make all the linear superposition states represented by a qubit chromosome with the same probability:

$$|q_j\rangle = \frac{1}{\sqrt{2^n}}|x_k\rangle$$

where $x_k$ is the k-th state represented by the binary string $(x_1, x_2, \cdots, x_n) \in [0, 1]^n$. The next step makes a set of binary solution $P(t) = \{x'_1, x'_2, \cdots, x'_m\}$ by observing $Q(t)$. A binary solution $x'_j(1 \leq j \leq n)$ is a binary string the same as $x_k$, which is formed by selecting each bit using the probability of corresponding qubit, either $|\alpha_j\rangle$ or $|\beta_j\rangle$. Then each solution $x'_j$ is evaluated to give some measure of fitness and the best solution is stored among $P(t)$.

In the while loop, other steps are similar as before except update $Q(t)$, where a set of qubit chromosomes is produced by applying some appropriate quantum gates. This step can make QEA have fitter states of the qubit chromosome, and its key issue is the selection of quantum gates, which need to be designed in compliance with practical problems.

Rotation gates have been proved to be effective for the knapsack problem and numerical optimization problems [19], such as:

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

where $\theta$ is a rotation angle, the value of $\theta$ has an effect on the speed of convergence, but if is too big, the solution may be diverge or have a premature convergence to a local optimum, the sign of $\theta$ determines the direction of convergence to a global optimum. And the lookup table is usually used as a strategy for the selecting of $\theta$. However, variable angle-distance rotation (VAR) strategy [23] is adopted for its feasibility and effectiveness in this paper.

3 Particle Swarm Optimization

Inspired by social behavior of birds, PSO is similar with other intelligent optimization algorithms. It begins with a population composed of random particles, and then maintains the population by repeated use of certain evolutionary operator. The principles and procedures of PSO are addressed in this section.
3.1. The principles of particle swarm optimization

In a d-dimensional search space, a population contains \( n \) particles, and each particle is associated with two key attributes: location and velocity, both of which can be depicted using a d-dimensional vector. For example, the location and velocity of particle \( i (i = 1, 2, \ldots, n) \) can be denoted by \( X_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \) and \( V_i = (v_{i1}, v_{i2}, \ldots, v_{id}) \), respectively. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviours. Therefore, the particles have a tendency to fly towards the better and better search area over the course of search process.

In practice, a fitness value is assigned to each particle, and the location of particles is a potential solution. According to the objective function designed specially to abstract the application problems, the fitness of particles can be calculated using \( f(X_i) \). Then all the particles are evaluated using their fitness.

In the process of iterations, particles renew their location and velocity by tracing two extreme values \( p_{\text{best}}^i \) and \( g_{\text{best}} \), where \( p_{\text{best}}^i = (p_{i1}, p_{i2}, \ldots, p_{id}) \) represents the best location of particle \( i \), \( g_{\text{best}} = (p_{11}, p_{12}, \ldots, p_{d1}) \) represents the best location of all the particles. The iteration formula of location and velocity is described by the equation (9) and (10):

\[
V(t + 1) = \omega V(t) + c_1 r_1 (p_{\text{best}} - X_i(t)) + c_2 r_2 (g_{\text{best}} - X_i(t)) \tag{9}
\]

\[
X_i(t + 1) = X_i(t) + V(t + 1) \Delta t \tag{10}
\]

where \( \omega \) is the inertia factor, \( r_1 \) and \( r_2 \) are random numbers in the range [0,1], \( c_1 \) and \( c_2 \) are the acceleration constants.

Equation (9) describes how the velocity is dynamically updated and equation (10) the position update of the flying particles. Equation (9) consists of three parts. The first part is the momentum part. The velocity can’t be changed abruptly. It is changed from the current velocity. The second part is the cognitive part which represents private thinking of itself learning from its own flying experience. The third part is the social part which represents the collaboration among particles learning from group flying experience.

3.2. The procedure of particle swarm optimization

Let \( \text{pop}(t) \) represent the swarm of particles, the basic procedure of PSO can be described as follows:

\[
\text{Procedure PSO} \begin{array}{ll}
\text{begin} & \\
\text{t=0} & \\
\text{initialize} \ \text{pop}(t) & \\
\text{evaluate} \ \text{pop}(t) & \\
\text{while} \ (t<\text{MaxGen}) \ \text{do} & \\
\text{begin} & \\
\text{t=t+1} & \\
\text{update the location and velocity of each particle} & \\
\text{evaluate} \ \text{pop}(t) & \\
\text{end} & \\
\text{end} & \\
\end{array}
\]

- 185 -
In the step of initialize \( \text{pop}(t) \), the population size is specified. Meanwhile, the location and velocity of each particle are randomly initialized with numbers in the restricted range. In the next step, the fitness of particles is calculated, and \( p_{\text{best}} \) and \( g_{\text{best}} \) are stored to update the location and velocity of particles according to the equation (9) and (10). The velocity of particles on each dimension is assigned to be \( \pm V_{\text{max}} \), that is, particles’ velocity on each dimension is clamped to a maximum velocity \( V_{\text{max}} \), which is an important parameter. Big \( V_{\text{max}} \) makes particles have the potential to fly far past good solution areas while a small \( V_{\text{max}} \) makes particles have the potential to be trapped into local minima, therefore unable to fly into better solution areas. Usually a fixed constant value is used as the \( V_{\text{max}} \).

4. Experiments and Results

In this section, six numerical optimization problems are introduced firstly, and then some comparative experiments are set to discuss and analyze the performance of QEA and PSO. The following six numerical optimization functions are considered to verify the performance of QEA and PSO for their dealing with numerical optimization problems in this paper. They are: (a) Sphere Function; (b) Ackley Function; (c) Griewank Function; (d) Rastrigin Function; (e) Schwefel Function; (f) Rosenbrock Function.

To minimize the six testing functions mentioned above, QEA, PSO and CGA are tested using the following parameter settings, and the results are presented in Fig.1 and Table 1. The probabilities of crossover and mutation of CGA were fixed as 0.65 and 0.05, respectively. The parameter \( k \) of QEA is set to 8. For PSO, the inertia factor \( \omega \) is 0.5, the acceleration constant \( c_1 \) and \( c_2 \) are equal to 2, and the maximum velocity \( V_{\text{max}} \) is 15 percent of the interval length of variables. The termination condition with the maximum number of generations was used, and the population size is equally set for QEA, PSO and CGA according to the complexity of testing functions.
A Comparative Study of Quantum Evolutionary Algorithm and Particle Swarm Optimization for Numerical Optimization Problems
Tinghuai Ma, Qiaoqiao Yan, Wenxia Xuan, Bin Wang

![Fig 1](image.png)

Fig 1. Comparison among QEA, PSO and CGA on the numerical optimization problems, the vertical axis is the function value, and the horizontal axis is the number of generation. (a), (b), (c), (d), (e), and (f) show the optimization result of Sphere, Ackley, Griewank, Rastrigin, Schwefel, and Rosenbrock function, respectively. All are averaged over 25 runs.

Fig 1 shows the progress of the minimum function value found by QEA, PSO and CGA over 25 runs for Sphere, Ackley, Griewank, Rastrigin, Schwefel, and Rosenbrock function. It should be noted that results of QEA and PSO were significantly better compared with that of CGA for all these six testing functions, for which the main reasons are as follows: (1) CGA is short for the high-dimension optimization problems. For example, CGA can find the minimum value easily when $N \leq 5$ in the case of Sphere function, but it becomes hard with the increasing of $N$; (2) CGA is short for the multi-peak optimization problems. Although CGA can produce new solutions based on crossover and mutation operators, it is hard to find the global optimal for the multi-peak problems. Ackley and Rosenbrock function are single peak functions, the result of CGA for these two functions are relatively better than that for others.

Compared with CGA, QEA and PSO are characterized by high efficiency, rapid speed of convergence and strong capability of global search. However, QEA and PSO have different
optimization ability for different optimization problems. Concretely speaking, QEA has stronger global searching ability than PSO, but PSO is labelled with better local search ability. For the cases of Sphere, Ackley, and Rosenbrock functions, the results of PSO are better than that of QEA in both the final value and the speed of convergence, for which the reason is that Sphere and Rosenbrock are single peak functions, and although Ackley is a multi peak function, seen from Fig.2 that its global minimum point is relatively isolated, so it is easy to be located. However, QEA performs better than PSO in the cases of Griewank, Rastrigin and Schwefel functions, which are complex multi peak functions and have many local optima, so the global search ability is important for those algorithms to be applied. Based on the foregoing analysis, it is clear that PSO performs more outstanding in the exploration and QEA do better in the exploitation.

Table 1. Experimental Results of the Six Numerical Optimization Functions, mean, min, and stdev Represent the Mean Best, Min Best, and Standard Deviation. All Are Averaged over 25 Runs

<table>
<thead>
<tr>
<th>Function Names</th>
<th>QEA</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>mean 6.182E-06</td>
<td>8.066E-51</td>
</tr>
<tr>
<td></td>
<td>min 4.36561E-06</td>
<td>1.70065E-55</td>
</tr>
<tr>
<td></td>
<td>stdev 2.8925E-06</td>
<td>1.97946E-50</td>
</tr>
<tr>
<td>Ackley</td>
<td>mean 0.0019659</td>
<td>7.265E-15</td>
</tr>
<tr>
<td></td>
<td>min 0.001965853</td>
<td>3.9968E-15</td>
</tr>
<tr>
<td></td>
<td>stdev 0</td>
<td>9.83702E-16</td>
</tr>
<tr>
<td>Griewank</td>
<td>mean 0.1542747</td>
<td>0.5097726</td>
</tr>
<tr>
<td></td>
<td>min 0.0008632289</td>
<td>0.137476388</td>
</tr>
<tr>
<td></td>
<td>stdev 0.125089749</td>
<td>0.137476388</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>mean 21.020851</td>
<td>55.251323</td>
</tr>
<tr>
<td></td>
<td>min 7.587766374</td>
<td>22.88403816</td>
</tr>
<tr>
<td></td>
<td>stdev 7.587766374</td>
<td>23.85758909</td>
</tr>
<tr>
<td>Schwefel</td>
<td>mean 486.15517</td>
<td>4038.076</td>
</tr>
<tr>
<td></td>
<td>min 206.2</td>
<td>2961.0</td>
</tr>
<tr>
<td></td>
<td>stdev 194.5821477</td>
<td>637.8615515</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>mean 151.7553274</td>
<td>18012.8255</td>
</tr>
<tr>
<td></td>
<td>min 26.85507258</td>
<td>0.006385856</td>
</tr>
<tr>
<td></td>
<td>stdev 306.598084</td>
<td>36742.09226</td>
</tr>
</tbody>
</table>

In order to make a further analysis of QEA and PSO, Table 1 lists some statistical information about the experimental results of the six numerical optimization functions for QEA and PSO, mainly including the mean best value, standard deviation and minimum best value. From Table I we can see that all the standard deviation of PSO is larger than that of QEA except Sphere function. In particular, for the cases of Ackley and Rosenbrock functions, although the minimum best value of PSO is better than that of QEA, the mean best value is larger. The reason why PSO has larger standard deviation and mean best value is that the success rate of PSO is lower, and once an approximate optimal solution is obtained, it is difficult to be improved even if the number of iterations is increased. That is, PSO has the problem of premature convergence. Correspondingly, QEA is relatively stable but with the lower rapid speed of convergence.

Conclusively, QEA has better global search capability that is manifested very outstandingly for the multi peak function optimization, but its convergence speed is slower. Contrarily, PSO has faster convergence speed, and it can solve certain optimization problem rapidly, especially for single peak functions. But the success rate of PSO is lower, and the premature convergence is a great drawback. It is worth mentioning that many improved algorithms have been proposed based on CGA, QEA, and PSO. Those improved algorithms may be more efficient for specific problems. Some hybrid algorithms such as quantum particle swarms algorithm (QPSO) [24] are also developed, which combines the advantages of QEA and PSO. Here we just do the analysis for their application in the numerical optimization.
5 Conclusions

In this paper, we have made a comparative study of QEA and PSO for the numerical optimization problems. Firstly, the procedures of QEA and PSO are described in detail, and then six testing benchmarks are introduced to analyze and check the performance of QEA and PSO, which consist of both the single peak and multi peak functions. Finally, some experiments are conducted to demonstrate the performance of algorithms based on the former testing benchmarks. For the comparison purpose, CGA is also simulated during the experiments.

The following conclusions are obtained from the experimental results: (1) For all these six testing functions, the optimization results of QEA and PSO are significantly better compared with that of CGA, CGA is short for the high-dimension and multi peak optimization problems. (2) QEA has better global search capability that is manifested very outstandingly for the multi peak function optimization, but its convergence speed is slower. (3) PSO has faster convergence speed, and it can solve certain optimization problem rapidly, especially for single peak functions. But the success rate of PSO is lower, and the premature convergence is a great drawback.

The above conclusions may be useful for fresh researchers as references. In the future, we will make a further study on these two algorithms, and some improved algorithms may be proposed for other application problems.

6. Acknowledgment

This research was partly supported by Natural science fund for colleges and universities in Jiangsu Province (08KJD520018), Natural Science Foundation from Nanjing University of Information & Science Technology (20080300).

7. References