International Journal of the Physical Sciences Vol. 6(19), pp. 4579-4588, 16 September, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.983


*Full Length Research Paper*

# Quantum hybrid genetic algorithm based on simulated annealing and its application

**Airong Nie**

Planning and Finance Department, East China JiaoTong University, 330013, Nanchang, Jiangxi, P. R. China.
E-mail: zhourgcs@gmail.com.

Accepted 23 July, 2011

**Quantum genetic algorithm (QGA) is firstly improved for numerical optimization with real coding, where populations are updated by a simple rotation method which inspires a real quantum genetic algorithm (RQGA), then simulated annealing (SA) is reasonably introduced in the optimizing process of RQGA, and a hybrid quantum genetic algorithm (HQGA) is presented, which could not only effectively avoid the premature phenomenon but also accelerate the search efficiency under the introduction of SA. Besides HQGA is applied to numerical optimization and the training of BP neural network, and through a comparison among QGA, RQGA and HQGA, it is obviously shown that HQGA performs better on running speed and optimizing capability.**

**Key words:** Quantum genetic algorithm simulated annealing, hybrid algorithm, real coding.

## INTRODUCTION

After quantum computing was formalized in the late 1980s (Benioff, 1980), the unique calculation method, artificial intelligent techniques (Ozcep et al., 2010) and powerful advantages in application of which had aroused widespread attention and quickly became a hotspot, moreover its combining with the subjects in other areas is an important forefront topic recently. Narayanan and Moore, (1996) firstly introduced quantum multi-universe into genetic algorithm, and compared traditional evolutionary algorithm and quantum inspired genetic algorithm about a well-known mathematic problem of traveling salesperson problem (TSP), which found a precedent of combining quantum computing and evolutionary algorithm. After, Kuk and Jong (2000) formally proposed a genetic quantum algorithm in which chromosome was applied to quantum coding, and they presented a detail quantum gate rotation method for updating chromosome, besides they applied this algorithm to 0/1 knapsack problem. Soon after, based on the paper (Kuk and Jong, 2000), further perfected quantum genetic algorithm in 2002, a Q-gate is introduced as a variation operator to drive the individuals toward better solutions, and the algorithm was renamed as quantum-inspired evolutionary algorithm (Kuk and Jong, 2002, 2004).

But the QGA, as a probabilistic parallel algorithm, always plunges into prematurity and has the shortcomings of poor local search ability. So some scholars have been absorbed to improve the QGA, for example, Yang et al. (2003) proposed a novel multi-universe parallel quantum genetic algorithm (MPQGA) and put forward a new blind source separation method based on the combination of MPQGA and independent component analysis and reversible logic circuit design (Zhou et al., 2010, 2011). Zhang et al. (2003) advanced a novel parallel quantum genetic algorithm applying Q-bit phase comparison approach and hierarchical ring model, which is characterized by rapid convergence and good global search capability. Chen et al. (2004) raised a novel Q-gate updating algorithm called Chaos updating rotated gates quantum-inspired genetic algorithm, and this novel algorithm is more powerful in convergence speed. A multi-objective meta-level GQA in order to determine parameters of QG which will be applicable for a wide variety of optimization problems was proposed (Khorsand and Akbarzadeh, 2005). Also some scholars have started to introduce some other optimizing algorithms into QGA (Zhou, 2010; Zhou and Ding, 2007, 2008), Wang et al. (2005) proposed a hybrid genetic algorithm to achieve better optimization performances by reasonably combining the Q-bit search of quantum algorithm in micro-space and classic genetic search of real-coded GA in macro-space.

In this paper, firstly QGA is improved for numerical optimization in m-dimension space with real coding, and then simulated annealing (SA) (Geng et al., 2007; Jose et al., 2008) is reasonably introduced in this optimizing process. Because real coding is suitable for some problems and SA has the powerful local search ability, HQGA can effectively avoid local optimum and meanwhile accelerate the searching efficiency.

## QUANTUM GENETIC ALGORITHM AND SIMULATED ANNEALING

### Quantum genetic algorithm (QGA)

Each individual in QGA is coded based on the concept and principle of quantum computing, that is, $q_j^t$ $(j = 1, 2, ..., n)$ in the population of $Q(t)=\{q_1^t, q_2^t, ..., q_n^t\}$ is code as follows:

$$q_j^t = \begin{bmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \dots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \dots & \beta_{jm}^t \end{bmatrix} \quad (1)$$

Here *m* is the length of the quantum genes (chromosome).

The procedure of QGA is described in the following:

**Initialization:** $\alpha$ $\beta$ in Formula 1 are all initialized with $1/\sqrt{2}$, that is, every individual is coded with the same probability amplitude.
**Observation:** in order to calculate individuals, population are transformed into $P(0)=\{x_1^0, x_2^0, ..., x_n^0\}$ through observing Q(0), where *x* is the individual with binary coding. And the procedure of observation, produce a rand number r (value in 0~1), and if $r < \alpha^2$, set this bit to 0, else to 1.
**Calculation:** each binary solution is evaluated to give a level of its fitness, and P(0) is stored into B(0) which is a memory space.
**Loop:** if the best solutions in B(t) do not meet the accuracy, quantum gate U in the following is applied to update the population Q(t), and individuals in the new population are also observed and calculated like step 2 and 3. B(t) are get by selecting the best individuals among B(t-1) and P(t), and algorithm is running in the loop till B(t) converge to optimum.

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

### Simulated annealing (SA)

Simulated annealing resembles the cooling process of molten metals through annealing. At high temperature, the atoms in the molten metals generate highly-excited random motion, as the temperature is gradually lowered, the motion of the atoms becomes less violent, and eventually they settle into a global energy state minimum. The distribution of energy states could be described by the Boltzmann distribution.

$$P\{r\} = \frac{1}{Z(T)} \exp\left(-\frac{E(r)}{k_B T}\right)$$

Here $Z(T)$ is the normalization factor and $k_B$ is the Boltzmann constant. SA is based on Metropolis rule, when the state $j$ is generated from state $i$, if $E_j < E_i$, the state $j$ is accepted instead of $i$, else if $P = \exp((E_i - E_j)/k_B T)$ >r (a random number value in 0~1), $j$ is still accepted else the state $i$ is reserved.

The procedure of SA simulates this process to achieve the optimum which is written as follows:

***Begin***
*k=0, t=$t_0$, s=$s_0$;*
*while (terminate condition = false)*
*do*
*while(sampling stability condition = false)*
*do*
*$s_j$ = generate(s);*
*if exp(($E_{sj}$-$E_s$)/$k_B$T)>=randrom[0,1], s= $s_j$ ;*
*end*
*$t_{k+1}$=update($t_k$);*
*k=k+1;*
*end*
*end*

## RQGA FOR NUMERICAL OPTIMIZATION

Numerical optimization in m-dimension can be described as follows (Boumaza et al., 2009):

$$\min f(x) = f(x_1, x_2, ..., x_m) \quad a_i \le x_i \le b_i \quad (i = 1, 2, ..., m)$$

And the fitness function that reflects approximate extent of individuals could be defined in the following:

$$fit(x) = -f(x)$$

For calculating the individuals, the coding fashion in the Formula 1 should be translated into binary coding, even further into real number coding, the calculated amount of which is so enormous and impacts to a great extent the speed of algorithm. Aiming at the numerical optimization, we propose a real coding manner.

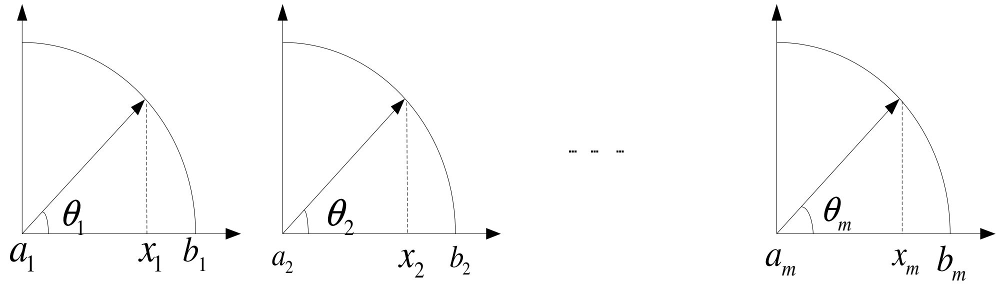



**Figure 1.** Geometric graph of the individual.

### Real coding

For population $P(t) = \{p_1^t, p_2^t, ..., p_n^t\}$, $p_j^t$ $j = 1, 2, ..., n$ is an individual in m-dimension space.

$$p_j^t = \begin{bmatrix} x_{j1}^t & x_{j2}^t & \cdots & x_{jm}^t \end{bmatrix} \tag{4}$$

Here $x_{ji}^t = a_i + (b_i - a_i)\cos\theta_{ji}^t \quad i = 1, 2, ..., m$. The individual in m-dimension space could be shown in the geometric graph fashion as shown in Figure 1.

So searching the optimum individuals is transformed instead of searching for a set of optimum angle $\theta$.

### Population updating

The action of quantum rotation gates in traditional quantum genetic algorithm is actually rotating the angel of Q-bit of the individuals, and inspired by which, we update population applying the angle rotation method. Set an initial rotation angle $\Delta\theta$ and population are updated just by rotating $\theta_i$ $(i = 1, 2, ..., m)$: $\theta_i + \Delta\theta$. And the key problems are defining the size and direction of $\Delta\theta$, which could be made in the following procedure.

Produce a current optimum population B(t), $B(t) = \{b_1^t, b_2^t, ..., b_n^t\}$, here $b_j^t = \{x_{bj1}^t, x_{bj2}^t, ..., x_{bjn}^t\}$. B(t) is made by selecting the best individuals among B(t-1) and $P(t)$, for the individual $p_j^t$ in the current population $P(t)$, if $x_{ji}^t > x_{bji}^t$, the rotation direction of $\theta_{ji}^t$ is positive, that is, $\Delta\theta$ is positive, else $\Delta\theta$ is negative and the size of $\Delta\theta$ is designed in compliance with the application problems.

### Real quantum genetic algorithm (RQGA)

The updating of traditional quantum genetic algorithm is based on probability amplitude updating, and after the quantum coding is replaced by real coding, every updating is for variable in m-dimension space. The rotation direction is defined by comparing the current optimum population and current update population, so the populations evolve always in the direction of better population. The procedure of RQGA is as follows:

**Step 1 initialization:** t=0, all angle $\theta$ in individual coding are initialized with $\pi/3$, that is, $x_i = \dfrac{a_i + b_i}{2}$;

**Step 2** fit calculation: each binary solution is evaluated to give a level of its fitness, and P(0) is stored into B(0) which is a memory space;

Loop:

**Step 3** if the best individual in B(t) does not meet the accuracy, $P(t)$ is updated as follows:

$$\theta_{ji}^{t+1} = \theta_{ji}^t + \Delta\theta_{ji}^t$$

Here $\Delta\theta_{ji}^t = \mathrm{sgn}(x_{ji}^t - x_{bji}^t) \times \Delta\theta$;

**Step 4** individuals in the new population are also calculated like step 2. B(t) are get by selecting the best individuals among B(t-1) and P(t), and algorithm is running in the loop till B(t) converge to optimum. The algorithm can be written as follows:

$t \leftarrow 0$
*initialize* $P(t)$
*evaluate* $P(t)$ *and store the best solutions among* $P(t)$ *to* $B(t)$
  *while* (*not termiation* − *condition*) *do*
    *update* $P(t)$ *under* $\theta$ *rotation*
    *store the best solutions among* $B(t-1)$ *and* $P(t)$ *in* $B(t)$
     $t \leftarrow t+1$
  *end*
*end*

## A NOVEL HYBRID QUANTUM GENETIC ALGORITHM

The evolution algorithms, in which individuals are processed as a group, could accelerate the global searching

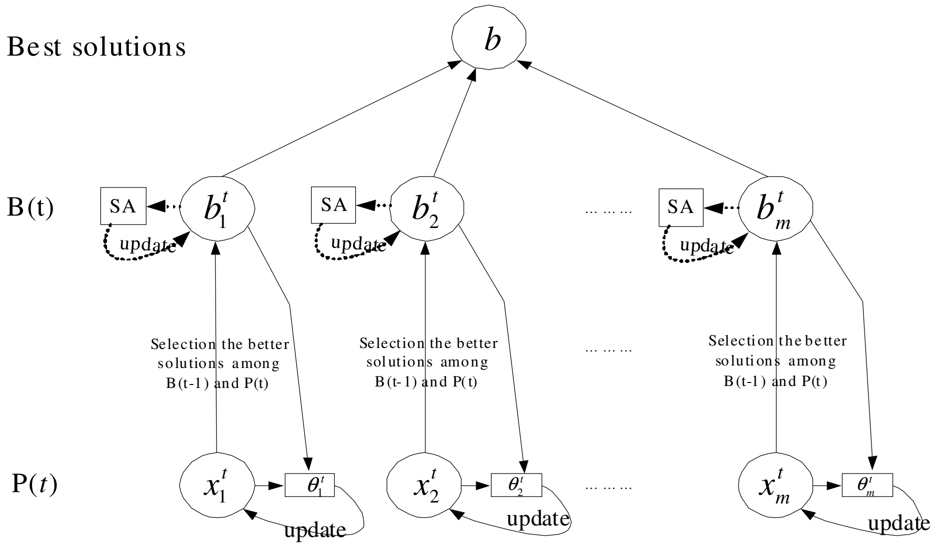



**Figure 2.** The framework of HQGA.

rate against other algorithms with searching individuals one by one, however, it also bring lots of shortcomings. As a novel evolution algorithm, QGA also easily fall into local optimum, that is, B(t) mentioned earlier converge in a population after several generations but it is not the global optimal population. In this paper, simulated annealing is applied to complement optimum searching in the process of RQGA, which makes the algorithm rapidly jump away from the local optimal point. Meanwhile, the powerful local searching ability of SA could capture the updating direction, so the rate of algorithm is advanced. The framework of HQGA is illustrated in Figure 2, where dotted line denotes that the operation works under the certain conditions. The procedure of HQGA is as follows:

**Step 1 initialization:** t=0, all angle $\theta$ in individual coding are initialized with $\pi/3$, that is, $x_i = \frac{a_i + b_i}{2}$;

**Step 2 fit calculation:** each binary solution is evaluated to give a level of its fitness, and P(0) is stored into B(0) which is a memory space;

Loop:

**Step 3** if the best individual in B(t) does not meet the accuracy, $P(t)$ is updated as follows:

$$\theta_{ji}^{t+1} = \theta_{ji}^{t} + \Delta\theta_{ji}^{t} .$$

Here $\Delta\theta_{ji}^{t} = \mathrm{sgn}(x_{ji}^{t} - x_{bji}^{t}) \times \Delta\theta$, and B(t) are get by selecting the best individuals among B(t-1) and P(t);

**Step 4** if the population converge, but the convergence value does not meet the accuracy (the population plunge into local optimum), we apply the SA to jump away from the local extreme point, that is, SA is applied to update each individual in B(t), so the better population is found for the remaining step of RQGA;

**Step 5** the new population $P(t)$ got by step 3 are also calculated like step 2, and algorithm is running in the loop till B(t) converge to optimum.

## SIMULATIONS

### Numerical optimization

In order to validate the effectiveness and feasibility of HQGA, and fix the rotation angle $\theta$, HQGA is applied to solve a series of functions which are shown in the following:

1. Shaffer's F5:

$$f(x_i) = 1/500 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i}^{2} (x_i - a_{ij})^6}$$ , here

$$x_i = (-65.536, 65.536)$$

$$(a_{ij}^k) = (\frac{-32}{-32+16k} \quad \frac{-16}{-32+16k} \quad \frac{0}{-32+16k} \quad \frac{16}{-32+16k} \quad \frac{32}{-32+16k})$$

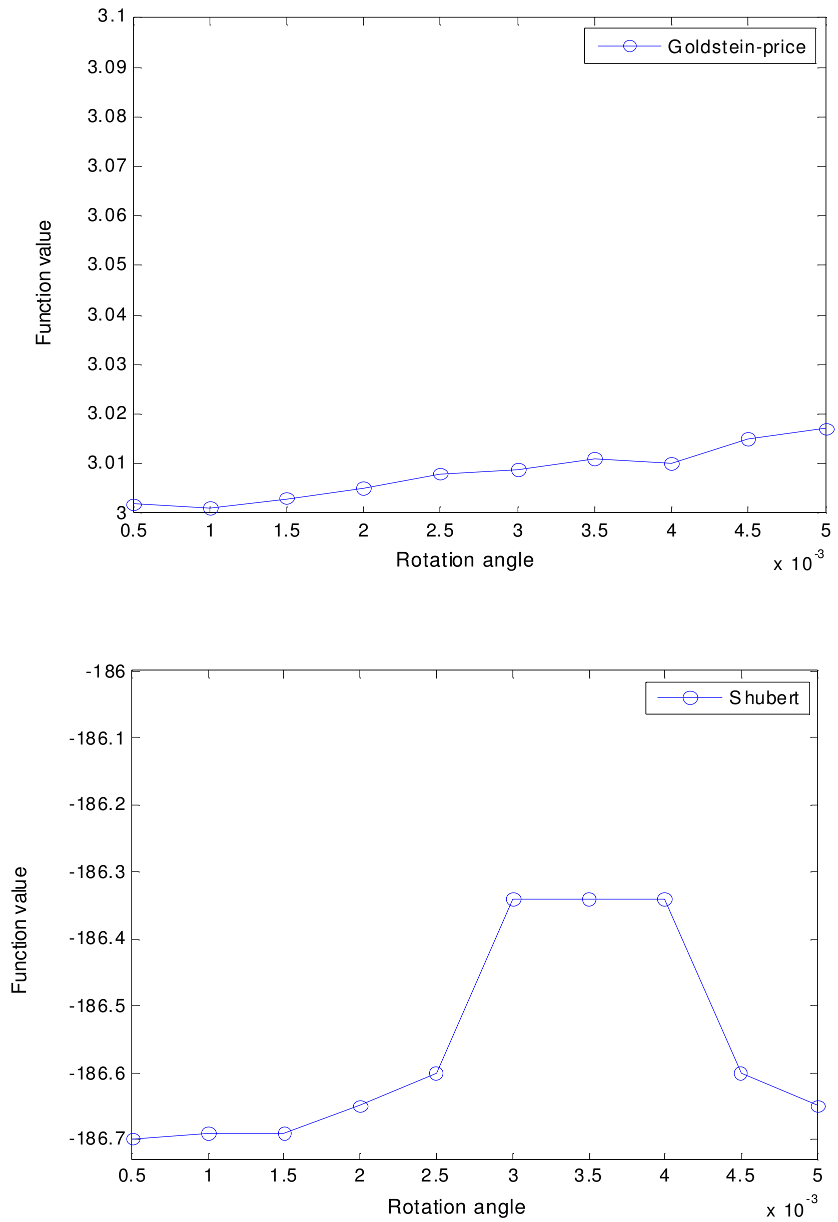
$$(a_{ij}) = (a_{ij}^0 a_{ij}^1 a_{ij}^2 a_{ij}^3 a_{ij}^4) \quad i = 1,2;\ j = 1,2,...,25;\ k = 0,1,...,4$$

The global Maximum is 1.002。

2. Shaffer's F6:

$$f(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2} ,$$

$$x, y \in (-100, 100)$$




C

D

**Figure 3.** The average optimum of 4 functions under different rotation angles.

The global Maximum is 1.

3. Goldstein-price :

$$f(x,y)=[1+(x+y+1)^2(19-14x+3x^2-14y+6xy+3y^2)]\times$$
$$[30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)]$$

$x, y \in [-2, 2]$ , the global Minimum is 3。

4. Shubert:

$$f(x,y)=\{\sum_{i=1}^{5} i\cos[(i+1)x+i]\}\{\sum_{i=1}^{5} i\cos[(i+1)y+i]\}+0.5[(x+1.42513)^2+(y+0.80032)^2]$$

$x, y \in [-10, 10]$ , the global Minimum is -186.73。

Set $\Delta\theta$ to $0.0005\pi$ , $0.0010\pi$ $0.0015\pi$ $0.0020\pi$ $0.0025\pi$ $0.0030\pi$ $0.0035\pi$ $0.0040\pi$ $0.0045\pi$ $0.0050\pi$ respectively, and run the algorithm 50 times under each $\Delta\theta$, the average optimum is shown in the Figure 3.

Here the horizontal axis is the different rotation angles $\Delta\theta$ and the vertical axis is the function optimization averaged over 50 runs. It can be seen from Figure 3, the smaller the rotation angle, the higher the accuracy of optimal solution, however, the more the running time consumes. After considering these two factors, $\Delta\theta = 0.0015\pi$ is set for rotation angle in this paper.

**Table 1.** Running ability Comparison of QGA, RQGA and HQGA.

| | | QGA | RQGA | HQGA |
|---|---|---|---|---|
| | b | 1.002 | 1.000 | 1.002 |
| | m | 0.998 | 0.996 | 1.001 |
| Shaffer's F5 | w | 0.906 | 0.902 | 0.910 |
| | n | 186 | 202 | 129 |
| | t | 0.198 | 0.102 | 0.128 |
| | b | 0.999 | 0.990 | 1.000 |
| | m | 0.991 | 0.981 | 0.989 |
| Shaffer's F6 | w | 0.907 | 0.909 | 0.921 |
| | n | 198 | 246 | 142 |
| | t | 0.246 | 0.298 | 0.205 |
| | b | 3.004 | 3.009 | 3.001 |
| | m | 3.008 | 3.011 | 3.002 |
| Goldstein-price | w | 3.014 | 3.019 | 3.007 |
| | n | 124 | 158 | 117 |
| | t | 0.194 | 0.206 | 0.191 |
| | b | -186.64 | -186.50 | -186.70 |
| | m | -186.48 | -186.41 | -186.61 |
| Shubert | w | -186.39 | -186.32 | -186.54 |
| | n | 189 | 249 | 152 |
| | T | 0.211 | 0.189 | 0.191 |

Applying QGA, RQGA and HQGA to solve aforementioned 4 functions, the running ability is shown is Table 1.

Although RQGA did not perform better than QGA on the solution accuracy and the number of generation, the running time of RQGA is shortened. And HQGA present the best ability among three algorithms on whether the running time, the solution accuracy or the number of generation.

## BP neural network training based on HQGA

### *BP neural network model of earthquake prediction*

Selecting the seismic data of a region in china as the sample data, we do earthquake prediction applying BP neural network based on HQGA. The normalized sample data is shown in Table 2, which contains 8 predictive factors as the input and an earthquake magnitude as the output. Here we apply BP model with a 15-dimension single hide layer, 8-dimension input layer and single-dimension output layer. This BP model is shown in Figure 4.

The connect weight from input to hide layer is $\omega_{ij}$ $i = 1, 2, ..., 8$ ; $j = 1, 2, ..., 15$ , the threshold value of neuron in hide layer is $\theta_j$; weight from hide layer to output is $\omega_{j1}$, the output layer just has a neuron and the threshold of which is $\theta$.

## Training results

Applying QGA, RQGA and HQGA to train aforementioned BP earthquake prediction, and the training results are presented in Figure 5.

The iteration number of QGA, RQGA and HQGA are 225 , 305 and165, respectively, and the error accuracy are 0.000997881, 0.000999126 and 0.000991688, respectively. It is obvious that HQGA performs better than other two algorithms on training rate and convergence accuracy. To further verify the better ability of HQGA against other algorithms, we test the BP net trained by three algorithms through a set of test data. Table 3 gives the test data. The error attained by comparing the prediction amplitude and the actual one is shown in Figure 6, in which the straight line for QGA, dotted line for RQGA model and '+' for HQGA.

## CONCLUSIONS

Aiming at the shortcoming of frequent translations in classical QGA, this paper proposes a real coding quantum

**Table 2.** The normalized sample data.

| Earthquake cumulative frequency | Velocity ratio | Seismic gap | The number of seismic belt | Active phase | Cumulative energy release | b-value | Number of abnormal earthquake swarm | Magnitude |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.16 | 0 | 0 | 0 | 0 | 0.56 | 0 | 0 |
| 0.589 | 0.63 | 0.81 | 0.5 | 1 | 0.459 | 0.79 | 0.5 | 0.648 |
| 0.993 | 0.85 | 0.58 | 1 | 1 | 0.486 | 0.48 | 1 | 0.856 |
| 0.463 | 0.46 | 0.42 | 1 | 1 | 0.894 | 0.98 | 1 | 0.945 |
| 0.286 | 0.56 | 0.86 | 1 | 1 | 0.899 | 0.64 | 1 | 0.528 |
| 0.492 | 0.84 | 0.74 | 0.5 | 0 | 0.104 | 0.88 | 0.5 | 0.465 |
| 0.791 | 0.69 | 0.46 | 0 | 0 | 0.694 | 0.89 | 0 | 0.285 |
| 0.578 | 0.95 | 0.58 | 0.5 | 1 | 0.789 | 0.15 | 0.5 | 0.248 |
| 0.368 | 0.46 | 0.44 | 0.5 | 1 | 0.156 | 0.46 | 0.5 | 0.598 |
| 0.532 | 0.41 | 0.95 | 1 | 1 | 0584 | 0.95 | 0.5 | 0.842 |

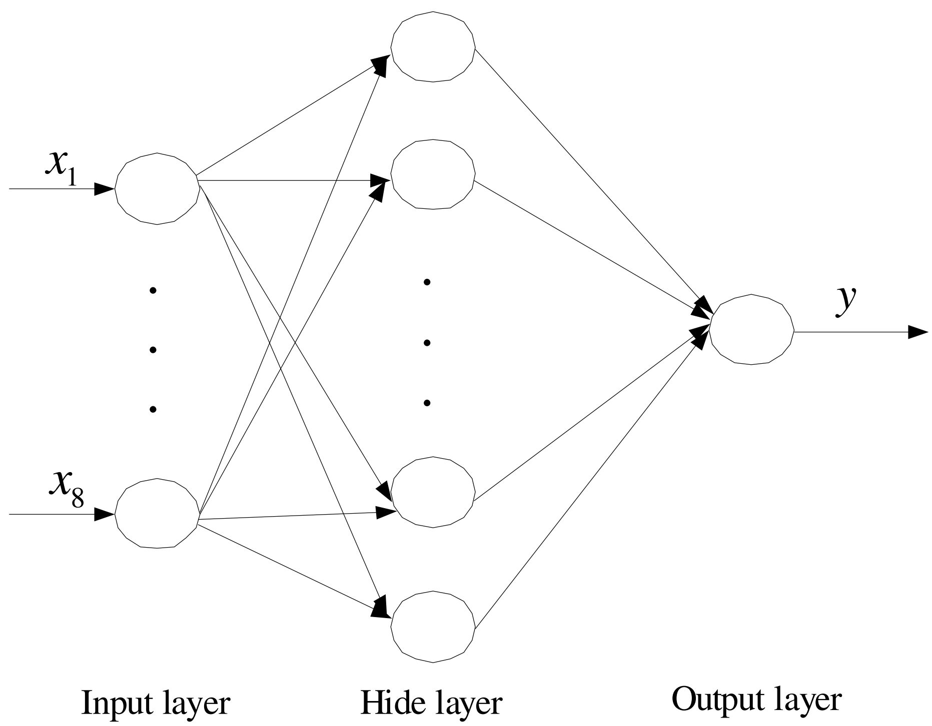
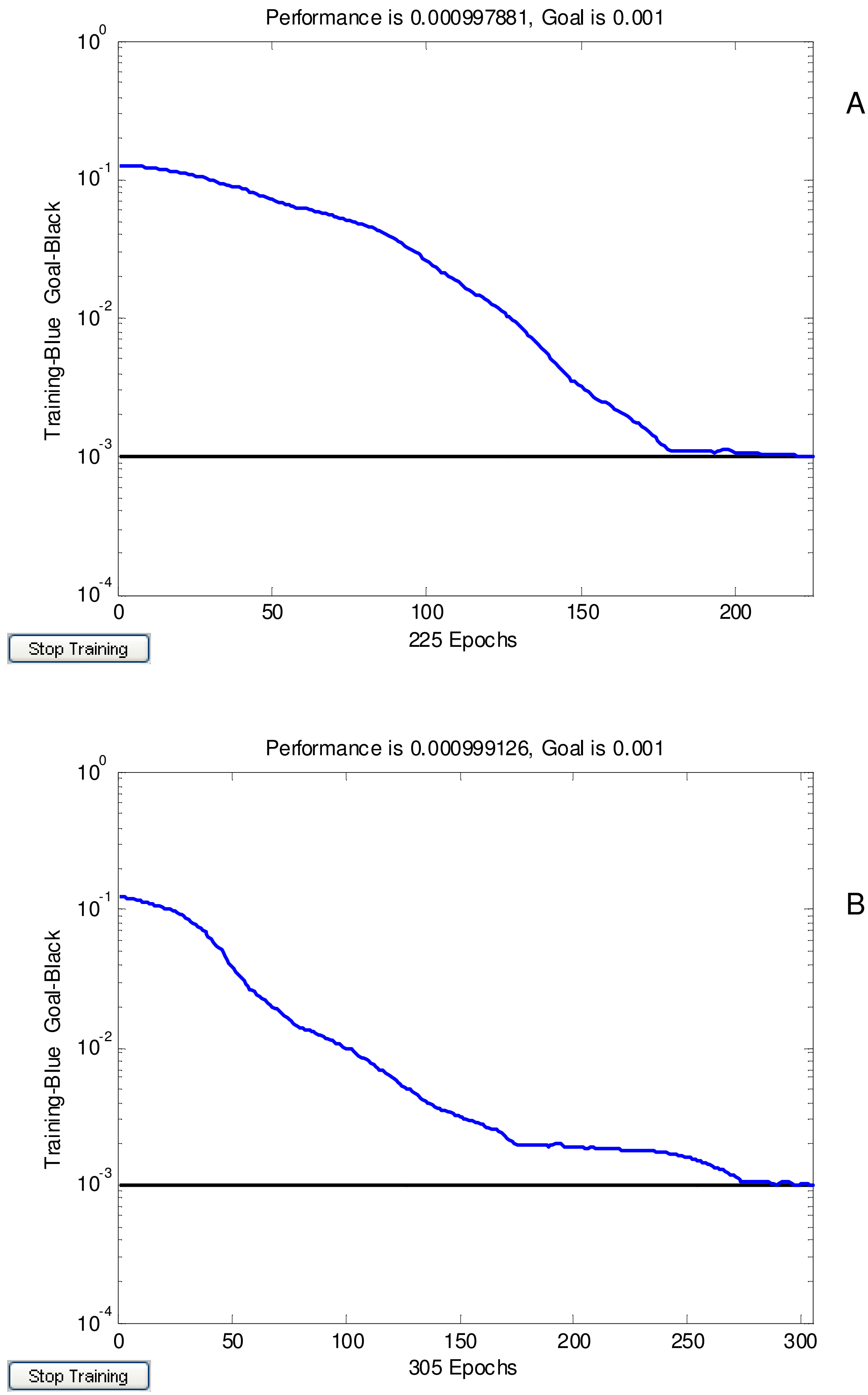



**Figure 4.** The BP earthquake prediction model.

quantum genetic algorithm and introduces the simulated annealing in the optimizing process of RQGA when it plunges into local optimum, as a result the novel HQGA is inspired. Due to the powerful local search of SA, HQGA can not only effectively jump out the local optimum point, but also greatly promote the searching rate. The population update of HQGA is realized by rotating the angle $\theta$ in individual coding, and through solving a set of optimization functions, we attain the optimal rotation angle $\Delta\theta$. Through a comparison among QGA, RQGA and HQGA for numerical optimization, it is obviously seen that RQGA has improved on running time, but QGA still has the better ability on iteration number and convergence accuracy against RQGA, and HQGA is the best on whether running time, iteration number or convergence accuracy. Besides, applying these three algorithms to train the BP neural network of earthquake prediction, HQGA also performs the best in all respect. Our future work is to design other new quantum genetic algorithm using simulated annealing method and apply HQGA to other complex practical problems.


Performance is 0.000997881, Goal is 0.001
Training-Blue Goal-Black
225 Epochs
Stop Training
A
Performance is 0.000999126, Goal is 0.001
Training-Blue Goal-Black
305 Epochs
Stop Training
B

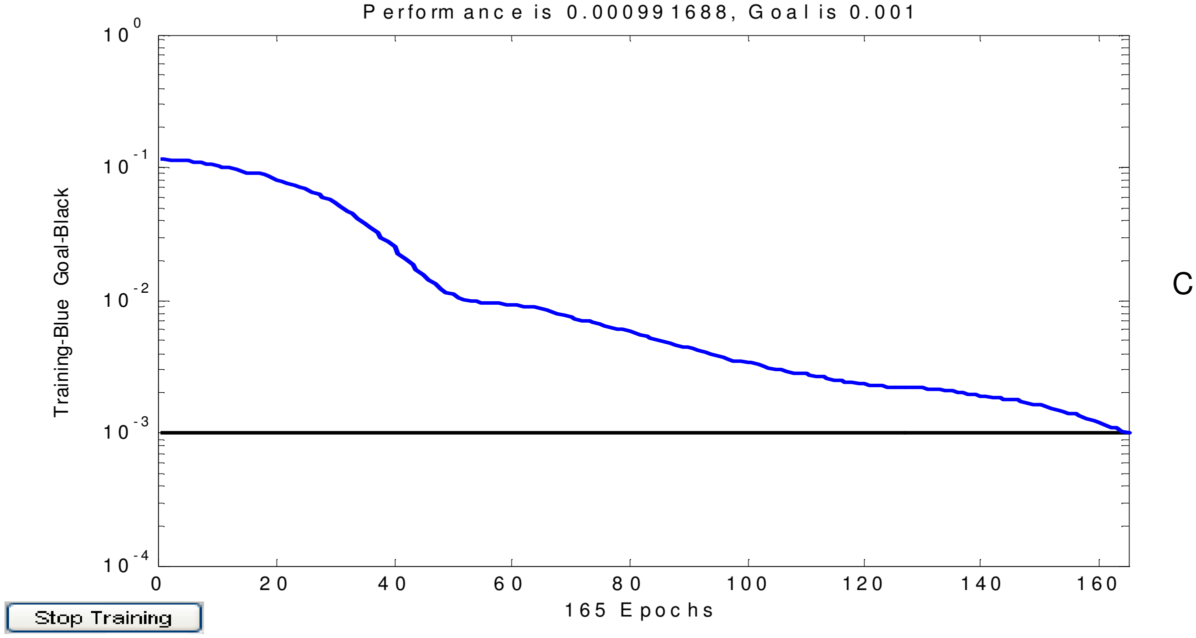



**Figure 5.** The training curve of HQGA(a), QGA(b) and RQGA(c).

**Table 3.** The test data.

| Earthquake cumulative frequency | Velocity ratio | Seismic gap | The number of seismic belt | Active phase | Cumulative energy release | b-value | Number of abnormal earthquake swarm | Magnitude |
|---|---|---|---|---|---|---|---|---|
| 0.259 | 0.15 | 0.21 | 0.5 | 0 | 0.541 | 0.25 | 0 | 0.154 |
| 0.488 | 0.48 | 0.51 | 0.5 | 0 | 0.214 | 0.65 | 0 | 0.216 |
| 0.548 | 0.84 | 0.48 | 0 | 1 | 0.654 | 0.24 | 0.5 | 0.368 |
| 0.654 | 0.21 | 0.41 | 0.5 | 0 | 0.254 | 0.43 | 0.5 | 0.415 |
| 0.851 | 0.19 | 0.16 | 1 | 1 | 0.657 | 0.84 | 1 | 0.512 |
| 0.458 | 0.74 | 0.84 | 1 | 1 | 0.325 | 0.69 | 1 | 0.401 |
| 0.499 | 0.56 | 0.94 | 0.5 | 1 | 0.689 | 0.71 | 1 | 0.463 |

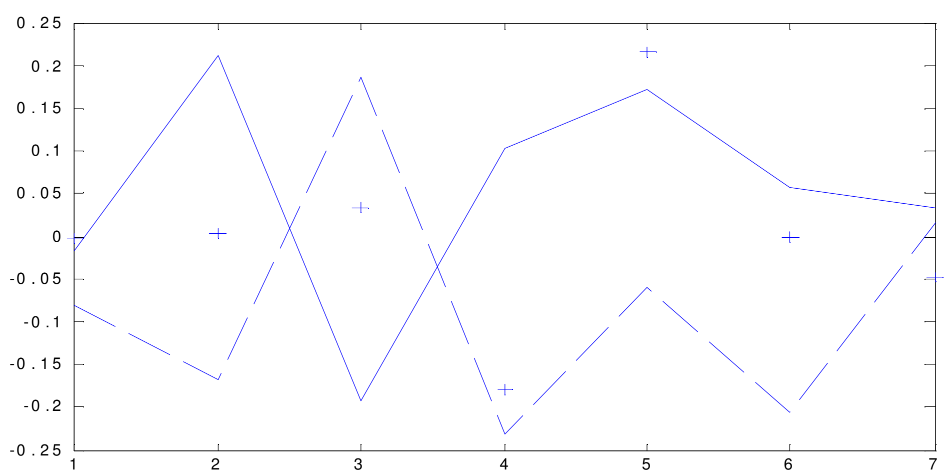


**Figure 6.** The error cure of QGA, RQGA and HQGA.

**REFERENCES**

Benioff P (1980). The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. J. Statist. Phys., 22: 563–591.

Boumaza N, Benouaz T, Chikhaoui A, Heknane A (2009). Numerical simulation of nonlinear pulses propagationin a nonlinear optical directional coupler. Int. J. Phys. Sci., 4(9): 505-513.

Chen H, Zhang JH, Zhang C (2004). Chaos updating rotated gates quantum-inspired genetic algorithm: Proc. Int. Conf. Communications, Circuits Syst., 2: 1108-1112.

Geng XT, Xu J, Xiao JH, Pan LQ (2007). A simple simulated annealing algorithm for the maximum clique problem. Inform. Sci., 177: 5064-5071.

Jose M, Leiva M, Sancho S (2008). A simulated annealing approach to speaker segmentaition in audio databases. Eng. Appl. Artif. Intel., 21: 499-508. ENG APPL ARTIF INTEL.

Khorsand AR, Akbarzadeh TMR (2005). Quantum gate optimization in a meta-level genetic quantum algorithm: 2005 IEEE Int. Conf. Syst. Man Cybernetics, 4: 3055-3062.

Kuk HH, Jong HK (2000). Genetic quantum algorithm and its application to combinational optimization problem: proceeding of the International Congress on Evolutionary Computation. IEEE Press, 1354-1360.

Kuk HH, Jong HK (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comp., 6(6): 580-593.

Kuk HH, Jong HK (2004). Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, $H_\varepsilon$ Gate, and Two-Phase Scheme. IEEE Trans. Evol. Comput., 8(2): 156-169.

Narayanan A, Moore M (1996). Quantum-inspired genetic algorithms: proceeding of IEEE International conference on Evolutionary Computation, Nagoya, Japan, 61-66.

Ozcep F, Yildirim E, Tezel O, Asci M, Karabulut S (2010). Correlation between electrical resistivity and soil-water content based artificial intelligent techniques. Int. J. Phys. Sci., 5(1): 47-56.

Wang L, Tang F, Wu H (2005). Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. Appl. Math. Comput., 171(2): 1141-1156.

Yang JA, Li B, Zhuang ZQ (2003). Multi-universe parallel quantum genetic algorithm and its application to blind-source separation: proc. Int. Conf. Neural Netw. Signal Processing, 1: 393-398.

Zhang GX, Jin WD, Hu LZ (2003). A novel parallel quantum genetic algorithm: proceedings of the fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, 693-697.

Zhou RG (2010). Quantum Competitive Neural Network. Int. J. Theor. Phys., 49(1):110-119

Zhou RG, Ding QL (2007). Quantum M-P Neural Network. Int. J. Theor. Phys., 46(12): 3209-3215

Zhou RG, Ding QL (2008). Quantum Pattern Recognition with Probability of 100%. Int. J. Theor. Phys., 47(5): 1278-1285

Zhou RG, Shi Y, Cao J, Wang HA (2010). Comment on "Efficient approaches for designing reversible Binary Coded Decimal adders". Microelectron. J., 41(5): 308-310.

Zhou RG, Shi Y, Wang HA, Cao J (2011). Transistor realization of reversible “ZS” series gates and reversible array multiplier. Microelectron. J., 42(2): 305-315.