Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2007) 633–640
NEUROCOMPUTING
www.elsevier.com/locate/neucom

A novel quantum swarm evolutionary algorithm and its applications

Yan Wang[a,*], Xiao-Yue Feng[a], Yan-Xin Huang[a], Dong-Bing Pu[b], Wen-Gang Zhou[a], Yan-Chun Liang[a], Chun-Guang Zhou[a]

[a]College of Computer Science and Technology, Jilin University, Key Laboratory for Symbol Computation and Knowledge Engineering of the National Education Ministry, Changchun 130021, China
[b]Department of Computer Science, Northeast Normal University, Changchun 130024, China

Available online 13 October 2006

Abstract

In this paper, a novel quantum swarm evolutionary algorithm (QSE) is presented based on the quantum-inspired evolutionary algorithm (QEA). A new definition of Q-bit expression called quantum angle is proposed, and an improved particle swarm optimization (PSO) is employed to update the quantum angles automatically. The simulated results in solving 0–1 knapsack problem show that QSE is superior to traditional QEA. In addition, the comparison experiments show that QSE is better than many traditional heuristic algorithms, such as climb hill algorithm, simulation anneal algorithm and taboo search algorithm. Meanwhile, the experimental results of 14 cities traveling salesman problem (TSP) show that it is feasible and effective for small-scale TSPs, which indicates a promising novel approach for solving TSPs.

Keywords: Quantum swarm evolutionary algorithm; Quantum-inspired evolutionary algorithm; Particle swarm optimization; Knapsack problem; Traveling salesman problem

1. Introduction

Quantum computing was proposed by Benioff and Feynman [2,4] in the early 1980s. It was declared that quantum computing could solve many difficult problems in the field of classical computation, which was based on the concepts and principles of quantum theory, such as superposition of quantum states, entanglement and intervention. Because of its unique computational performance, there has been a great interest in the application of the quantum computing [5,14]. Han [6] proposed the quantum-inspired evolutionary algorithm (QEA), which was inspired by the concept of quantum computing. In QEA, the smallest unit of information is called a Q-bit, which is defined as $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$. Besides, a Q-gate was introduced as a variation operator to promote the optimization of the individuals Q-bit. Han and Yang (2003, 2004) [7,8,11,21] have applied the QEA to some optimization problems and applications, such as function optimization, face verification, blind source separation, etc.. The performance of the QEA shows that it is better than the traditional evolutionary algorithms, such as the conventional genetic algorithm (GA) [13], in many fields. Although the step size delta involved in QEA is always a constant and should be designed in compliance with application problems, it has not had the theoretical basis till now.

Meanwhile, particle swarm optimization (PSO), which is a population-based optimization strategy introduced by Kennedy and Eberhart [12], demonstrates good performance in many function optimization problems and parameter optimization problems in recent years. It is initialized with a group of random particles and then updates their velocities and positions with following formulae:

$$\begin{aligned} v(t+1) &= v(t) + c_1 * \mathrm{rand}() * (p\mathrm{best}(t) - \mathrm{present}(t)) \\ &\quad + c_2 * \mathrm{rand}() * (g\mathrm{best}(t) - \mathrm{present}(t)), \\ \mathrm{present}(t+1) &= \mathrm{present}(t) + v(t+1). \end{aligned}$$

Wang and Pang [18] have applied the PSO to solve the traveling salesman problem (TSP). In addition, Feng and

*Corresponding author.
E-mail addresses: wy6868@hotmail.com (Y. Wang), cgzhou@jlu.edu.cn (C.-G. Zhou).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.10.001

Wang [3] proposed a hybrid algorithm AFTER–PSO for combining forecasting. Wang and Huang (2004, 2005) also applied the same strategy to establish two neural network systems: a fuzzy neural network system based on generalized class cover and a minimal uncertainty neural networks based on Bayesian theorem. The new systems have been successfully applied to identify the taste signals of tea [10,19,20].

A novel quantum swarm evolutionary algorithm (QSE) is proposed, which is based on QEA, in this article. The proposed algorithm employs a novel quantum bit expression mechanism called quantum angle and adopts the improved PSO to update the Q-bit automatically. The simulated results show that QSE is superior to QEA and many traditional heuristic algorithms, such as climb hill algorithm, simulation anneal algorithm and taboo search algorithm, in solving 0–1 knapsack problem. In addition, the test result of 14 cities TSP problem shows the feasibility to apply the QSE to solve the TSP.

The original QEA and PSO are introduced in Section 2, and then some drawbacks of those methods are discussed respectively. The definition of the quantum angle and the procedure of quantum swarm evolutionary are illustrated in Section 3. Experimental results of 0–1 knapsack problem and TSP are shown and discussed in Section 4. Conclusions are given in Section 5.

2. Original QEA and PSO

2.1. Quantum-inspired evolutionary algorithm

QEA was proposed by Han [6], which was inspired by the concept of quantum computing. In QEA, the smallest unit of information is called a Q-bit, which is defined as $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$, where α and β are complex numbers that specify the probability amplitudes of the corresponding states. The moduli $|\alpha|^2$ and $|\beta|^2$ are the probabilities that the Q-bit exists in state "0" and state "1", respectively, which satisfy that $|\alpha|^2+|\beta|^2=1$. And an m-Q-bits is defined as $\left[\begin{array}{c|c|c|c}\alpha_1 & \alpha_2 & \dots & \alpha_m\\ \beta_1 & \beta_2 & \dots & \beta_m\end{array}\right]$, where $|\alpha_i|^2+|\beta_i|^2=1 \quad (i=1,2,\dots,m)$ and m is the number of Q-bits [6].

The procedure of QEA is described as follows:

Procedure of QEA
 Begin
  Initialize $\mathbf{Q}(0)$ at $t=0$
  Make $\mathbf{P}(0)$ by observing the state of $\mathbf{Q}(0)$
  Repair $\mathbf{P}(0)$
  Evaluate $f(\mathbf{X}_j^0)$
  Store the best solutions among $\mathbf{P}(0)$ into $\mathbf{B}^0$ and $f(\mathbf{B}^0)$
  While (not termination condition) do
   Begin
    $t=t+1$
    Make $\mathbf{P}(t)$ by observing the state of $\mathbf{Q}(t)$
    Repair $\mathbf{P}(t)$
    Evaluate $f(\mathbf{X}_j^t)$
    Update $\mathbf{Q}(t)$ using Q-gate $U(t)$
    Store the best solutions among $\mathbf{P}(t)$ into $\mathbf{B}^t$ and $f(\mathbf{B}^t)$
   End
 End

where $\mathbf{Q}(t)=\{q_1^t,q_2^t,\dots,q_n^t\}$, $q_j^t=\left[\begin{array}{c|c|c|c}\alpha_{j1}^t & \alpha_{j2}^t & \dots & \alpha_{jm}^t\\ \beta_{j1}^t & \beta_{j2}^t & \dots & \beta_{jm}^t\end{array}\right]$, $P(t)=\{X_1^t,X_2^t,\dots,X_n^t\}$, $\mathbf{B}^t\in\mathbf{X}_j$, $\mathbf{X}_j^t=\{x_{j1}^t,x_{j2}^t,\dots,x_{jm}^t\}$, $i=1,2,\dots,m$, $j=1,2,\dots,n$, n is the size of the population.

In the step of "**initialize** $\mathbf{Q}(0)$ at $t=0$", $\begin{bmatrix}\alpha_{ji}^0\\ \beta_{ji}^0\end{bmatrix}$ of all q_j^0 in $\mathbf{Q}(0)$ are initialized with $1/\sqrt{2}$. It means that in each m-Q-bits, q_j^0 represents the linear superposition of all possible states with the same probability [6].

To obtain the binary string, the step of "**make** $\mathbf{P}(t)$ by observing the state of $\mathbf{Q}(t)$" can be implemented for each Q-bit individual as follows. When observing the state of $\mathbf{Q}(t)$, the value $x_{ji}^t=0$ or 1 of $\mathbf{P}(t)$ is determined by the probability $|\alpha_{ji}^t|^2$ or $|\beta_{ji}^t|^2$ [8].

Procedure make $\mathbf{P}(t)$
 Begin
  $j=0$;
  While $(j<n)$ do
   $j=j+1$;
   $i=0$;
   While $(i<m)$ do
    $i=i+1$;
    If random $[0,1]>|\alpha_{ji}^t|^2$
    Then $x_{ji}^t=1$
    Else $x_{ji}^t=0$
   End
  End
 End

The steps of "**repair** $\mathbf{P}(t)$" and "**evaluate** $f(\mathbf{X}_j^t)$" are according to the problems, where $f(\mathbf{X})$ is the fitness function.

The **update** procedure of Q-bits is introduced as follows.

Procedure update $\mathbf{Q}(t)$
 Begin
  $j=0$;
  While $(j<n)$ do
   $j=j+1$;
   $i=0$;
   While $(i<m)$ do
    $i=i+1$;

Determine $\Delta\theta_{ji}$ with the lookup table

Obtain $\begin{bmatrix} \alpha_{ji}^t \\ \beta_{ji}^t \end{bmatrix}$ as:

$$\begin{bmatrix} \alpha_{ji}^t \\ \beta_{ji}^t \end{bmatrix} = U(t) \begin{bmatrix} \alpha_{ji}^{t-1} \\ \beta_{ji}^{t-1} \end{bmatrix}$$

End

End

End

Quantum gate (Q-gate) $U(t)$ is a variable operator of QEA. It can be chosen according to the problem. A modified rotation gate used in QEA is as follows [21]:

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = \begin{vmatrix} \cos(\xi(\Delta\theta_i)) & -\sin(\xi(\Delta\theta_i)) \\ \sin(\xi(\Delta\theta_i)) & \cos(\xi(\Delta\theta_i)) \end{vmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix},$$

where $\xi(\Delta\theta_i) = s(\alpha_i, \beta_i) * \Delta\theta_i$; $s(\alpha_i, \beta_i)$ and $\Delta\theta_i$ represent the rotation direction and angle, respectively. The lookup table is presented in Table 1. Where delta is the step size and should be designed in compliance with the application problem. However, it has not had the theoretical basis till now, even though it usually is set as small value. In the comparison experiments, we set delta $= 0.01\pi$. Our proposed algorithm, QSE, which was proposed in this work, is based on another improved quantum rotation gate strategy.

2.2. Particle swarm optimization

PSO is a population-based optimization strategy introduced by Kennedy and Eberhart [12]. And it has demonstrated good performance in many function optimization problems and parameter optimization problems in recent years, such as solving TSP [18], combining forecasting [3] and optimizing neural networks systems [10,19,20]. It is initialized with a group of random particles and then updates their velocities and positions with following formulae:

$$v(t+1) = v(t) + c_1 * \mathrm{rand}() * (p\mathrm{best}(t) - \mathrm{present}(t)) + c_2 * \mathrm{rand}() * (g\mathrm{best}(t) - \mathrm{present}(t)),$$

$$\mathrm{present}(t+1) = \mathrm{present}(t) + v(t+1),$$

Table 1
A modified rotation gate lookup table

x_i	b_i	$f(X)>f(B)$	$\Delta\theta_i$	$s(\alpha_i,\beta_i)$			
				$\alpha_i\beta_i>0$	$\alpha_i\beta_i<0$	$\alpha_i=0$	$\beta_i=0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	Delta	+1	−1	0	±1
0	1	True	Delta	−1	+1	±1	0
1	0	False	Delta	−1	+1	±1	0
1	0	True	Delta	+1	−1	0	±1
1	1	False	0	0	0	0	0
1	1	True	0	0	0	0	0

where $v(t)$ is the particle velocity, present(t) is the current particle. $p$best(t) and $g$best(t) are defined as individual best and global best. Rand( ) is a random number between [0, 1]. c_1, c_2 are learning factors. Usually $c_1 = c_2 = 2$. To accelerate searching velocity and to avoid oscillation, an improvement of v that satisfies the convergence condition of the particles is utilized in Section 3.2 and the following experiments.

3. Main results

3.1. Quantum angle

In order to adopt PSO to update the Q-bit automatically, we first give a definition on the quantum angle.

Definition 1. A quantum angle is defined as an arbitrary angle θ and a Q-bit is presented as $[\theta]$.

Then $[\theta]$ is equivalent to the original Q-bit as $\begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix}$. It satisfies that $|\sin(\theta)|^2 + |\cos(\theta)|^2 = 1$ spontaneously. Then an m-Q-bits $\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \ldots & \alpha_m \\ \beta_1 & \beta_2 & \ldots & \beta_m \end{array}\right]$ could be replaced by $[\theta_1|\theta_2|\ldots|\theta_m]$. The common rotation gate

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = \begin{vmatrix} \cos(\xi(\Delta\theta_i)) & -\sin(\xi(\Delta\theta_i)) \\ \sin(\xi(\Delta\theta_i)) & \cos(\xi(\Delta\theta_i)) \end{vmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$$

is replaced by $[\theta_i'] = [\theta_i + \xi(\Delta\theta_i)]$.

Table 1 shows that θ_i' is only a simple function of $\Delta\theta_i$, $\cos\theta_i$, $\sin\theta_i$, $\mathbf{B}$, $\mathbf{X}$, $f(\mathbf{B})$ and $f(\mathbf{X})$. And as Delta has not had the theoretical basis till now, the efficiency of quantum gate $[\theta_i'] = [\theta_i + \xi(\Delta\theta_i)]$ is to be limited. Therefore, a novel QSE is proposed in the following section.

3.2. Quantum swarm evolutionary algorithm

In this section, we use the concept of swarm intelligence of PSO and regard all m-Q-bits in the population as an intelligence group, which is named quantum swarm. First, we find the local best quantum angle and the global best value from the local ones. Then according to these values, we update quantum angles by Q-gate. However, each individual still has large random adjustment space at that time.

The proposed procedure, which is called QSE, based on the procedure of QEA is summarized as follows:

1. Use quantum angle to encode Q-bit,

$$\mathbf{Q}(t) = \{q_1^t, q_2^t, \ldots, q_n^t\}, \quad q_j^t = [\theta_{j1}^t|\theta_{j2}^t|\ldots|\theta_{jm}^t].$$

2. **Make** each $x_{ji}^t = 0$ or 1 of $\mathbf{P}(t)$ by observing the state of $\mathbf{Q}(t)$ through $|\cos(\theta_{ji})|^2$ or $|\sin(\theta_{ji})|^2$ as follows:

Begin

$j = 0$;

While ($j<n$) do
  $j=j+1$; $i=0$;
  While ($i<m$) do
    $i=i+1$;
    If random $[0,1]>|\cos(\theta_{ji})|^2$
    Then $x_{ji}^t=1$
    Else $x_{ji}^t=0$
  End
End
End

3. Modify **update** procedure to update $\mathbf{Q}(t)$ with the following improved PSO formulae instead of using traditional Q-gate $U(t)$:

$$v_{ji}^{t+1}=\chi*(\omega*v_{ji}^t+C1*\mathrm{rand}()*(\theta_{ji}^t(p\mathrm{best})-\theta_{ji}^t)$$
$$+C2*\mathrm{rand}()*(\theta_i^t(g\mathrm{best})-\theta_{ji}^t)),$$
$$\theta_{ji}^{t+1}=\theta_{ji}^t+v_{ji}^{t+1},$$

where v_{ji}^t, θ_{ji}^t, $\theta_{ji}^t(p\mathrm{best})$ and $\theta_i^t(g\mathrm{best})$ are the velocity, current position, individual best and global best of the ith Q-bit of the jth m-Qbits, respectively. Set $\chi=0.99$, $W=0.7298$, $C_1=1.42$ and $C_1=1.57$, which satisfy the convergence condition of the particles: $W>(C_1+C_2)/2-1$. Since $C_2>C_1$, the particles will converge faster to the global optimal position of the swarm than the local optimal position of each particle, i.e., the algorithm has global searching property [10,17].

4. Experimental results

4.1. Solving 0–1 knapsack problem

The 0–1 knapsack problem is described as: given a set of items and a knapsack, select a subset of the items so as to maximize the profit $f(\mathbf{X})=\sum_{i=1}^m p_i x_i$ subject to $\sum_{i=1}^m \omega_i x_i \leqslant C$, where $\mathbf{X}=\{x_1\dots x_m\}$, $x_i=0$ or 1, ω_i is the weight of the ith item, p_i is the profit of the ith item, C is the capacity of the knapsack, respectively. $x_i=1$ if the ith item is selected, otherwise $x_i=0$. In the experiments, we used the similar data sets as in Ref. [6]. Set random $\omega_i\in[1,10]$, $p_i=\omega_i+l_i$, where the random figure $l_i\in[0,5]$, knapsack capacity is set as $C=(1/2)\sum_{i=1}^m\omega_i$, and three knapsack problems with 100, 250 and 500 items are considered. At the same time, we employed the same profit evaluation procedure and added the similar repair strategy mentioned in Ref. [6], which was based on the structure of QSE proposed

Procedure repair P(t)
Begin
  Knapsack-overfilled = false
  If $\sum_{i=1}^m \omega_i x_i^t > C$ then
  Knapsack-overfilled = true
  While (knapsack-overfilled = true) do
    Select the jth item from the knapsack randomly
    $x_j{}^t=0$
    If $\sum_{i=1}^m \omega_i x_i^t \leqslant C$ then
    knapsack-overfilled = false
  End
  While (knapsack-overfilled = false) do
    Select the kth item from the knapsack randomly
    $x_k{}^t=1$
    If $\sum_{i=1}^m \omega_i x_i^t > C$ then
    Knapsack-overfilled = true
  End
  $x_k{}^t=0$
End

And the evaluate $f(\mathbf{X}_j^t)$ is the profit $f(\mathbf{X}_j^t)=\sum_{i=1}^m p_i x_{ji}^t$.

The comparison of QSE and QEA on the knapsack problem with different items and the same population size is shown in Fig. 1. (The results of Fig. 1 are the average of 10 tests, which has the population size as 20, delta = 0.01π, and iteration times as 1000.) It shows that QSE is better than QEA in both speeds and profits. Table 2 and Fig. 2 show that QSE with larger population size obtains faster convergent speed and dominates larger item number, but spends more running time. (The results of Fig. 2 are the average of 10 tests, which has population

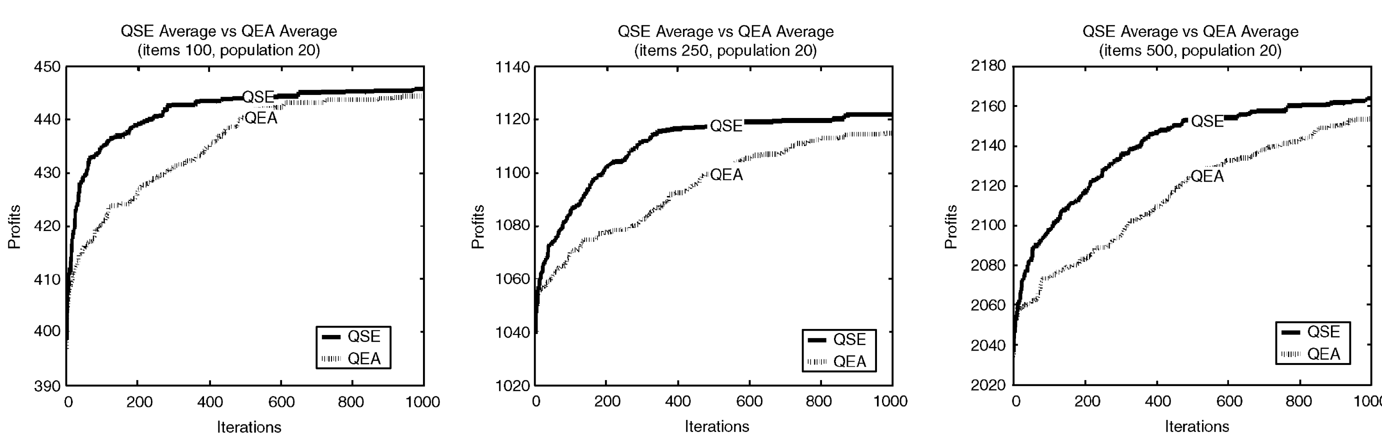

Fig. 1. Comparison of QSE and QEA on the knapsack problems with 100, 250 and 500 items.

Table 2
Results of different population size and iteration times of QSE in knapsack problem

Item number		100		250		500	
Size	Iteration times	Best profit	Time (s)	Best profit	Time (s)	Best profit	Time (s)
10	100	413.96	1	1079.0	2	2083.5	4
	500	449.11	3	1119.5	8	2161.0	16
	1000	453.18	5	1130.3	14	2174.4	34
20	100	443.12	2	1096.8	3	2122.9	7
	500	452.66	6	1132.4	14	2182.6	33
	1000	455.96	12	1137.2	28	2190.0	67
30	100	445.64	2	1104.5	5	2128.6	10
	500	453.89	8	1132.7	19	2186.9	46
	1000	457.03	16	1141.9	38	2190.0	98

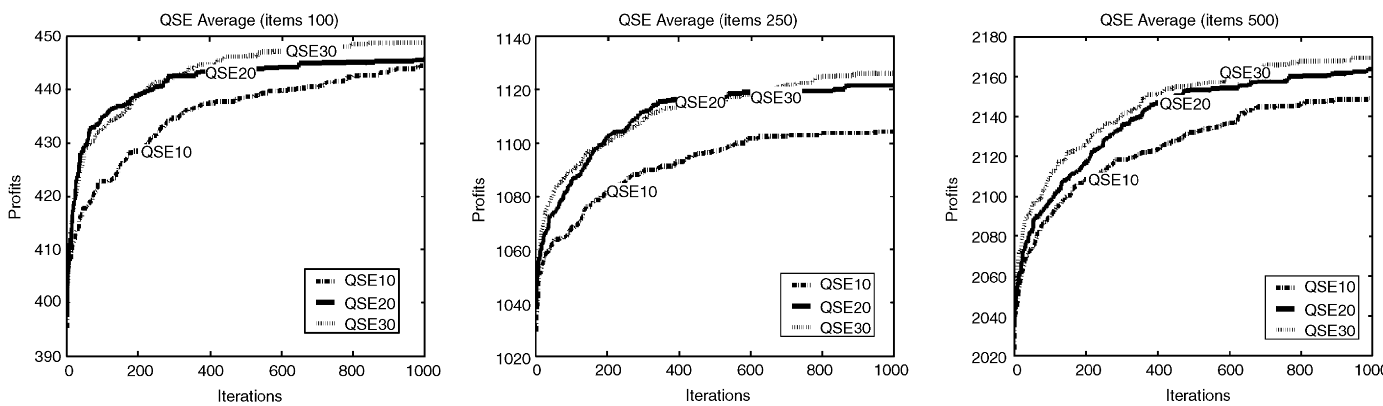

Fig. 2. Comparison of different population size of QSE on the knapsack problems with 100, 250 and 500 items.

size as 10, 20, 30, respectively, and iteration times as 1000. In Table 2, the running time is the second round of the average time of 10 tests.) Fig. 3 and Table 3 show that QSE is better than many traditional heuristic algorithms. (In Fig. 3, each method tests 10 times and each test iterates 1000 times. In Table 3, all parameters are set as same as Fig. 3 and the running time is the second round of the average time of 10 tests.) The comparison experiment is performed by heuristic algorithm tool kit [9]. It includes several heuristic functions in solving 0–1 knapsack problems, such as the climb hill algorithm (hillks), simulation anneal algorithm (saks) and taboo search algorithm (tabuks). The test environment is P4 2.6G, 512M, Windows XP, Matlab6.5.

4.2. Solving traveling salesman problem

As a well-known NP-hard combinatorial optimization problem, TSP is a general and simple form of many complicated problems in many different fields. It is easy to be described but hard to be resolved [1,16].

Give a weighted graph $G = (S,D)$, where $S = (S_1, S_2, \ldots, S_m)$ is the set of cities and $D = \{(S_i, S_j) : S_i, S_j \in S\}$ is the set of edges. Let $d(S_i,S_j)$ be a cost measure associated with edge $(S_i, S_j) \in D$. In the following experiment, city $S_i \in S$ is given by its coordinates (x_i,y_i) and $d(S_i,S_j)$ is the Euclidean distance between S_i and S_j. The object of TSP is to find a roundtrip of minimal total length visiting each city exactly once.

According to the characteristic of the TSP and the proposed QSE method, we encoded each Q-bit of $\mathbf{Q} = \{q_1, q_2, \ldots, q_n\}$ as $q = [\theta_{11}|\cdots|\theta_{1c}|\theta_{21}|\cdots|\theta_{2c}|\cdots|\theta_{m1}|\cdots|\theta_{mc}]$ by the quantum angle, where m is the number of cities and c is a constant, which satisfies $2^c \geqslant m$. Then we treated each binary string $\{x_{i1} \ldots x_{ic}\}$ of the observing value $\mathbf{X} = \{x_{11} \ldots x_{1c}, x_{21} \ldots x_{2c}, \ldots, x_{m1} \ldots x_{mc}\}$ as the visited sequence of the ith city, where $i = 1, 2, \ldots, m$. Before evaluating $f(\mathbf{X})$, we first sorted all the binary strings $\{x_{i1} \ldots x_{ic}\}$ to obtain visited sequence $S_1 \to S_2 \to \ldots \to S_m$. And then $f(\mathbf{X}) = -\sum_{i=1}^{m-1} d(S_i, S_{i+1}) + d(S_m, S_1)$ is calculated, where $d(S_i,S_j)$ denotes the distance between cities S_i and S_j. Therefore, the following repair and evaluate procedure are employed.

Procedure repair P(t)
Begin
Sort all the binary strings $\{x_{i1}^t \ldots x_{ic}^t\}$.
If some $\{x_{k_1 1}^t \ldots x_{k_1 c}^t\}, \ldots, \{x_{k_j 1}^t \ldots x_{k_j c}^t\}$ have the same values, which $k_j = 1, 2, \ldots, m$, then sort them randomly.
End

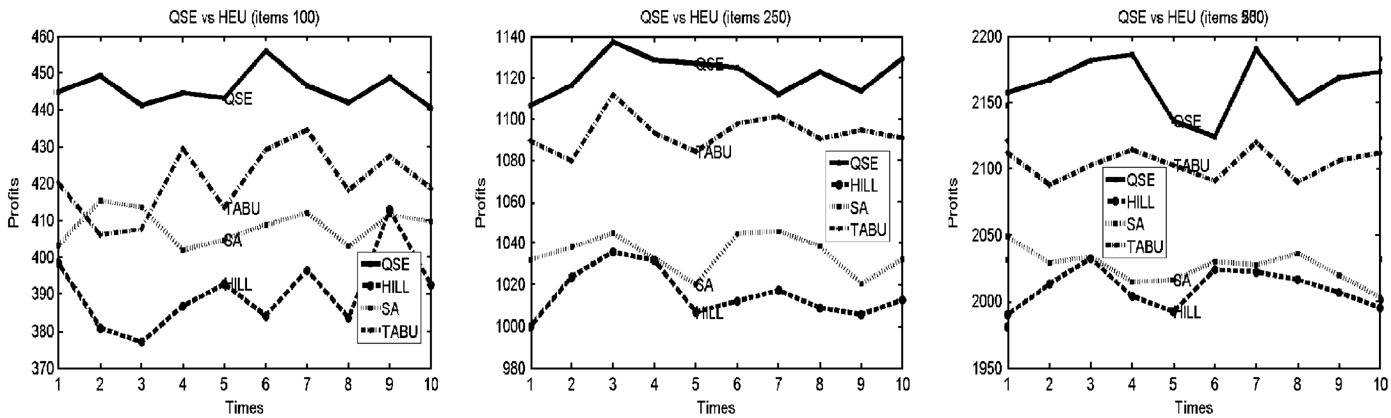

Fig. 3. Comparison of QSE (population size 20) and heuristic algorithms on knapsack problems, includes HILLKS, SAKS (anneal coefficient 0.99, initial temperature 100) and TABUKS (taboo table is 20).

Table 3
Results of QSE vs. traditional heuristic algorithms in knapsack problem

Item number	100		250		500	
Method	Best profit	Time (s)	Best profit	Time (s)	Best profit	Time (s)
QSE	455.96	12	1137.2	28	2190	67
HILLKS	412.74	2	1035.9	4	2032.3	7
SAKS	415.27	9	1045.8	27	2048.8	29
TABUKS	429.47	10	1111.8	29	2120.1	77

And the evaluate $f(\mathbf{X}_j^t)$ is the total length

$$f(\mathbf{X}_j^t) = -\sum_{i=1}^{m-1} d(S_{ji}^t, S_{j(i+1)}^t) + d(S_{jm}^t, S_{j1}^t).$$

The performance of the proposed algorithm using QSE for TSP is examined by the benchmark problem BURMA14 with 14 nodes from [15]. The data for the symmetric TSP of 14 cities are shown in Table 4.

The related parameters of the approach are set as follows: $n = 20$, $c = 4$ and $m = 14$.

After 500 iterations, the obtained optimal solution is

$$1 \to 10 \to 9 \to 11 \to 8 \to 13 \to 7 \to 12 \to 6 \to 5 \to 4 \to 3 \to 14 \to 2.$$

Its cost is 30.8785, which is equal to the data that was obtained by Wang and Huang [18]. The initial random solution with cost 50.7981 and the best one obtained by using QSE are shown in Fig. 4, respectively. The scale of the searched space is the product of the number of the individual, the Q-bit and the running iterations, which is equal to $20 \times 4 \times 14 \times 1500 = 1\,680\,000$. Therefore, it is easy to conclude that the searched space is only 0.054% of the solution space. The experiment is implemented on a PC (P4 2.6G, 512 M, Windows XP, Matlab 6.5).

Table 4
The data for the symmetric traveling salesman problem of 14 cities

City position	X	Y
1	16.47	96.10
2	16.47	94.44
3	20.09	92.54
4	22.39	93.37
5	25.23	97.24
6	22.00	96.05
7	20.47	97.02
8	17.20	96.29
9	16.30	97.38
10	14.05	98.12
11	16.53	97.38
12	21.52	95.59
13	19.41	97.13
14	20.09	94.55

5. Conclusions

In this paper, a novel QSE is presented, which is based on the QEA. A novel quantum bit expression mechanism called quantum angle is employed and the improved PSO is adopted to update the Q-bit automatically. The simulated results in solving 0–1 knapsack problem show that QSE is superior to traditional QEA. The comparison experiments

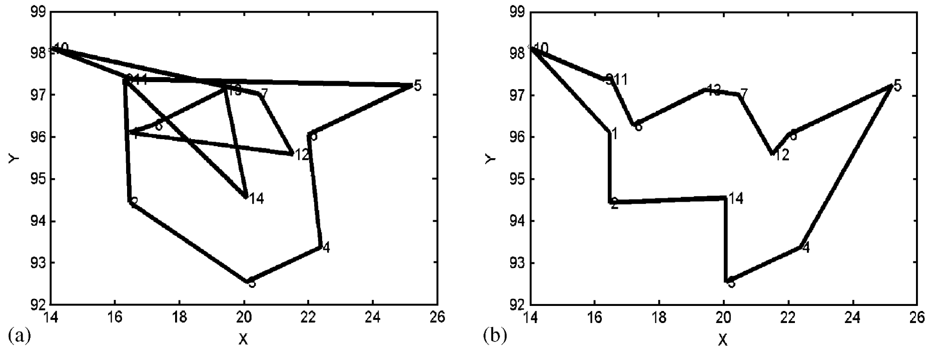

Fig. 4. (a) Initial solution (cost 50.7981), (b) best solution (cost 30.8785).

also show that QSE is better than many conventional heuristic algorithms, such as climb hill algorithm, simulation anneal algorithm and taboo search algorithm. In addition, the examination of solving TSP indicates that the proposed approach of QSE obtained the best result by searching a small-size proportion of the solution space. It has also shown that a worse performance of the behavior was observed when the number of the cities increased. This can be ascribed to the binary string coding which we used to represent the visit orders of the cities. The study on the limitation of the binary coding is in progress. Future research works will include how to find a more effective method for choosing the parameters according to the information of different problems.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 60433020, the Science-technology Development Project of Jilin Province of China under Grant No. 20050705-2, and the 985 Project of Jilin University.

References

[1] L.G. Bendall, Domination analysis beyond the traveling salesman problem, Ph.D. Thesis, Department of Mathematics, University of Kentucky, 2004.

[2] P. Benioff, The computer as a physical system: a microscopic quantum mechanical hamiltonian model of computers as represented by Turing machines, J. Stat. Phys. 22 (1980) 563–591.

[3] X.Y. Feng, Y.C. Liang, Y.F. Sun, H.P. Lee, C.G. Zhou, Y. Wang, A hybrid algorithm for combining forecasting based on AFTER–PSO, in: Proceedings of PRICAI 2004, Lecture Notes in Artificial Intelligence, vol. 3157, Springer, Berlin, 2004, pp. 942–943.

[4] R. Feynman, Simulating physics with computers, Int. J. Theoret. Phys. 21 (6) (1982) 467–488.

[5] L.K. Grover, Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 1994, pp. 124–134.

[6] K.H. Han, J.H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, IEEE Trans. Evolut. Comput., 6(6) (2002) 580–593.

[7] K.H. Han, J.H. Kim, Quantum-inspired evolutionary algorithms with a new termination criterion, HεGate, and two-phase scheme, IEEE Trans. Evol. Comput. 8 (2) (2004) 156–169.

[8] K.H. Han, K.H. Park, C.H. Lee, J.H. Kim, Parallel quantum-inspired genetic algorithm for combinatorial optimization problem, in: Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, Korea, 2001, pp. 1422–1429.

[9] Heuristic Algorithm Tool Kit, Copyright 2002, Lars Aurdal/ Rikshospitalet. Online available: 〈http://www.idi.ntnu.no/~lau/Forelesninger/.

[10] Y.X. Huang, Y. Wang, W.G. Zhou, Z.Z. Yu, C.G. Zhou, A fuzzy neural network system based on generalized class cover and particle swarm optimization, in: Proceedings of ICIC 2005, Hefei, China, Lecture Notes in Computer Science, vol. 3645, Springer, Berlin, 2005, pp. 119–128.

[11] J.S. Jang, K.H. Han, J.H. Kim, Quantum-inspired evolutionary algorithm-based face verification, in: Proceedings of Genetic Evolutionary Computation Conference Lecture Notes in Computer Science, vol. 2724, Springer, Berlin, 2003, pp. 2147–2156.

[12] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceeding of IEEE International Conference on Neural Networks, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[13] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, third, revised and extended ed., Springer, Berlin, 1999.

[14] P.W. Shor, Quantum computing. Documenta Mathematica, in: Extra Volume, Proceedings of the International Congress of Mathematicians, Berlin, Germany, 1998, pp. 467–486.

[15] Standard TSPLIB, 1997. Online available: http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html.

[16] X.J. Tang, Genetic Algorithms with application to engineering optimization, Ph.D. Thesis, University of Memphis, 2004.

[17] D. Van, F. Bergh, An Analysis of Particle Swarm Optimizers, Ph.D. Thesis, Natural and Agricultural Science Department, University of Pretoria, 2001.

[18] K.P. Wang, L. Huang, C.G. Zhou, W. Pang, Particle swarm optimization for traveling salesman problem, in: Proceeding of the 2nd ICMLC, Xi'an, China, Vol. 3, IEEE Press, 2003, pp. 1583–1585.

[20] Y. Wang, C.G. Zhou, Y.X. Huang, X.Y. Feng, Training minimal uncertainty neural networks by Bayesian theorem and particle swarm optimization, in: Proceedings of ICONIP 2004, Calcutta, India, Lecture Notes in Computer Science, vol. 3316, Springer, Berlin, 2004, pp. 579–584.

[19] Y. Wang, C.G. Zhou, Y.X. Huang, X.Y. Feng, Identification of taste signals of tea based on minimal uncertainty neural networks, Comput. Res. Dev. (in Chinese) 42 (1) (2005) 66–71.

[21] J.A. Yang, B. Li, Z.Q. Zhuang, Z.F. Zhong, Quantum genetic algorithm and its application research in blind source separation, Mini-Micro Systems (in Chinese) 24 (8) (2003) 1518–1523.

Yan Wang, born in 1978, is currently a Ph.D. student in the College of Computer Science and Technology, Jilin University, China. His research interests are computational intelligence, pattern recognition, quantum-inspired evolutionary computation and bioinformatics. He has published over 20 papers.

Xiaoyue Feng, born in 1977, is currently a Ph.D. student in the College of Computer Science and Technology, Jilin University, China. Her research interests are neural networks, combining forecasting, and quantum-inspired evolutionary computation.

Yanxin Huang, born in 1967, is a senior lecturer in the College of Computer Science and Technology, Jilin University, China. He is interested in the theory and applications of computational intelligence, pattern recognition.

Dongbing Pu, born in 1970, is a lecturer in the College of Computer, Northeast Normal University, China. His research interests are computational intelligence and embedded system.

Wengang Zhou, born in 1981, is a postgraduate in the College of Computer Science and Technology, Jilin University, China. His research interests include computational intelligence and bioinformatics.

Yanchun Liang, born in 1953, is a professor in the College of Computer Science and Technology, Jilin University, China. He is interested in computational intelligence and bioinformatics.

Chunguang Zhou, born in 1947, is currently a professor and the Dean in the College of Computer Science and Technology, Jilin University, China. He is interested in the theory and applications of computational intelligence, pattern recognition and image processing.