Introduction of Quantum-inspired Evolutionary Algorithm

Kuk-Hyun Han and Jong-Hwan Kim
Department of Electrical Engineering and Computer Science,
Korea Advanced Institute of Science and Technology (KAIST),
373-1, Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea
(Tel: +82-42-869-8048; Fax: +82-42-869-8877; Email:{khhan, johkim}@vivaldi.kaist.ac.kr)

Abstract: This paper introduces quantum-inspired evolutionary algorithm (QEA), which is based on the concept and principles of quantum computing such as a quantum bit and superposition of states. Like other evolutionary algorithms, QEA is also characterized by the representation of the individual, the evaluation function and the population dynamics. However, instead of binary, numeric, or symbolic representation, QEA uses a Q-bit, defined as the smallest unit of information, for the probabilistic representation and a Q-bit individual as a string of Q-bits.

Keywords: evolutionary algorithm, quantum computing, probabilistic representation

1. Introduction

Quantum mechanical computers were proposed in the early 1980's [1] and the description of quantum mechanical computers was formalized in the late 1980's [2]. Many efforts on quantum computers have progressed actively since the early 1990's because these computers were shown to be more powerful than classical computers on various specialized problems. There are well-known quantum algorithms such as Shor's quantum factoring algorithm [3], [4] and Grover's database search algorithm [5], [6]. Research on merging evolutionary computing and quantum computing has been started by some researchers since late 1990's. They can be classified into two fields. One concentrates on generating new quantum algorithms using automatic programming techniques such as genetic programming [7]. The other concentrates on quantum-inspired evolutionary computing for a classical computer, a branch of study on evolutionary computing that is characterized by certain principles of quantum mechanics such as standing waves, interference, coherence, etc. In [8] and [9], the concept of interference was included in a crossover operator.

This paper presents quantum-inspired evolutionary algorithm (QEA), which is based on the concept and principles of quantum computing such as a quantum bit and superposition of states. In [10], [11], and [12], the basic concept of QEA was introduced, its applicability to the parallel algorithm was verified, and its characteristics were analyzed, respectively. Like other evolutionary algorithms, QEA is also characterized by the representation of the individual, the evaluation function and the population dynamics. However, instead of binary, numeric, or symbolic representation, QEA uses a Q-bit, defined as the smallest unit of information, for the probabilistic representation and a Q-bit individual as a string of Q-bits. It should be noted that although QEA is based on the concept of quantum computing, QEA is not a quantum algorithm, but a novel evolutionary algorithm for a classical computer.

This paper is organized as follows. Section 2 introduces briefly the basics of quantum computing. Section 3 describes QEA. Section 4 analyzes the characteristics of QEA. Concluding remarks follow in Section 5.

2. Basics of Quantum Computing

The smallest unit of information stored in a two-state quantum computer is called a quantum bit or qubit [13]. A qubit may be in the '1' state, in the '0' state, or in any superposition of the two. The state of a qubit can be represented as

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where α and β are complex numbers that specify the probability amplitudes of the corresponding states. $|\alpha|^2$ gives the probability that the qubit will be found in the '0' state and $|\beta|^2$ gives the probability that the qubit will be found in the '1' state. Normalization of the state to unity guarantees

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2}$$

The state of a qubit can be changed by the operation with a quantum gate. A quantum gate is a reversible gate and can be represented as a unitary operator, U acting on the qubit basis states satisfying $U^\dagger U = UU^\dagger$, where $U^\dagger$ is the hermitian adjoint of U. There are several quantum gates, such as NOT gate, Controlled NOT gate, Rotation gate, Hadamard gate, etc.[14]. NOT operation is shown as follows:

$$|0\rangle \longrightarrow |1\rangle$$
$$|1\rangle \longrightarrow |0\rangle$$

In Controlled NOT gate, the NOT operation is only operative when the state of the controlled qubit is '1' state. Hadamard operation is shown in the following.

$$|0\rangle \longrightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|1\rangle \longrightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

If there is a system of m-qubits, the system can represent 2^m states at the same time. However, in the act of observing a quantum state, it collapses to a single state.

3. Quantum-inspired Evolutionary Algorithm

Inspired by the concept of quantum computing, QEA was designed with a novel Q-bit representation, a Q-gate as a variation operator, and an observation process.

3.1. Representation

A number of different representations can be used to encode the solutions onto individuals in evolutionary computation. The representations can be classified broadly as: binary, numeric, and symbolic ones[15]. QEA uses a new representation, called a Q-bit, for the probabilistic representation that is based on the concept of qubits, and a Q-bit individual as a string of Q-bits. A Q-bit is defined as the smallest unit of information in QEA, which is defined with a pair of numbers, (α, β), as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

where $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ gives the probability that the Q-bit will be found in the '0' state and $|\beta|^2$ gives the probability that the Q-bit will be found in the '1' state. A Q-bit may be in the '1' state, in the '0' state, or in a linear superposition of the two. A Q-bit individual as a string of m Q-bits is defined as

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array} \right], \tag{3}$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, 2, \cdots, m$.
Q-bit representation has the advantage that it is able to represent a linear superposition of states. If there is, for instance, a three-Q-bit system with three pairs of amplitudes such as

$$\left[\begin{array}{c|c|c} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & \frac{\sqrt{3}}{2} \end{array} \right], \tag{4}$$

the states of the system can be represented as

$$\begin{aligned} &\frac{1}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle - \frac{1}{4}|010\rangle - \frac{\sqrt{3}}{4}|011\rangle \\ &+\frac{1}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{1}{4}|110\rangle - \frac{\sqrt{3}}{4}|111\rangle. \end{aligned} \tag{5}$$

The above result means that the probabilities to represent the states $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, and $|111\rangle$ are system of (4) contains information of eight states. Evolutionary computing with Q-bit representation has a better characteristic of population diversity than other representations, since it can represent linear superposition of states probabilistically. Only one Q-bit individual such as (4) is enough to represent eight states, but in binary representation at least eight strings, (000), (001), (010), (011), (100), (101), (110), and (111) are needed.

3.2. QEA

The structure of QEA is described in the following.

procedure QEA
begin
    $t \leftarrow 0$
i)   initialize $Q(t)$
ii)   make $P(t)$ by observing the states of $Q(t)$
iii)   evaluate $P(t)$
iv)   store the best solutions among $P(t)$ into **b**
    while (**not** termination-condition) **do**
    begin
        $t \leftarrow t + 1$
v)   make $P(t)$ by observing the states of $Q(t-1)$
vi)   evaluate $P(t)$
vii)   update $Q(t)$ using Q-gates
viii)   store the best solutions among $P(t)$ into **b**
    end
end

QEA is a probabilistic algorithm like other evolutionary algorithms. QEA, however, maintains a population of Q-bit individuals, $Q(t) = \{\mathbf{q}_1^t, \mathbf{q}_2^t, \cdots, \mathbf{q}_n^t\}$ at generation t, where n is the size of population, and $\mathbf{q}_j^t$ is a Q-bit individual defined as

$$\mathbf{q}_j^t = \left[\begin{array}{c|c|c|c} \alpha_{j1}^t & \alpha_{j2}^t & \cdots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \cdots & \beta_{jm}^t \end{array} \right], \tag{6}$$

where m is the number of Q-bits, i.e., the string length of the Q-bit individual, and $j = 1, 2, \cdots, n$. The procedure of QEA is described in the following.
i) In the step of 'initialize $Q(t)$,' α_i^0 and β_i^0, $i = 1, 2, \cdots, m$, of all $\mathbf{q}_j^0 = \mathbf{q}_j^t|_{t=0}$, $j = 1, 2, \cdots, n$, are initialized with $\frac{1}{\sqrt{2}}$. It means that one Q-bit individual, $\mathbf{q}_j^0$ represents the linear superposition of all possible states with the same probability:

$$|\Psi_{\mathbf{q}_j^0}\rangle = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |X_k\rangle, \tag{7}$$

where X_k is the k-th state represented by the binary string $(x_1 x_2 \cdots x_m)$, where x_i, $i = 1, 2, \cdots, m$, is either 0 or 1 according to the probability of either $|\alpha_i^0|^2$ or $|\beta_i^0|^2$, respectively.
ii) This step makes binary solutions in $P(0)$ by observing the states of $Q(0)$, where $P(0) = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \cdots, \mathbf{x}_n^0\}$ at generation $t = 0$. One binary solution, $\mathbf{x}_j^0$, $j = 1, 2, \cdots, n$, is a binary string of length m, which is formed by selecting either 0 or 1 for each bit using the probability, either $|\alpha_i^0|^2$ or $|\beta_i^0|^2$, $i = 1, 2, \cdots, m$, of $\mathbf{q}_j^0$, respectively.
iii) Each binary solution $\mathbf{x}_j^0$ is evaluated to give a level of its fitness.
iv) The initial best solutions are then selected among the binary solutions, $P(0)$, and stored into **b**.
v, vi) In the **while** loop, binary solutions in $P(t)$ are formed by observing the states of $Q(t-1)$ as in step ii), and each binary solution is evaluated for the fitness value.
vii) In this step, Q-bit individuals in $Q(t)$ are updated by applying Q-gates defined as a variation operator of QEA, by which operation the updated Q-bit should satisfy the normalization condition, $|\alpha'|^2 + |\beta'|^2 = 1$, where α' and β' are the values of the updated Q-bit. The following rotation gate is used as a Q-gate in QEA, such as

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \tag{8}$$

where $\Delta\theta_i$, $i = 1, 2, \cdots, m$, is a rotation angle of each Q-bit towards either 0 or 1 state depending on its sign. Figure 1 shows the polar plot of the rotation gate. $\Delta\theta_i$ should be designed in compliance with the application problem. Table 1 can be used as an angle parameters for the rotation gate.

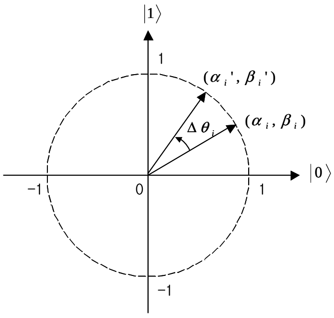


Fig. 1. Polar plot of the rotation gate

x_i	b_i	$f(\mathbf{x}) \geq f(\mathbf{b})$	$\Delta\theta_i$
0	0	$false$	θ_1
0	0	$true$	θ_2
0	1	$false$	θ_3
0	1	$true$	θ_4
1	0	$false$	θ_5
1	0	$true$	θ_6
1	1	$false$	θ_7
1	1	$true$	θ_8

Table 1. Lookup table of $\Delta\theta_i$, where $f(\cdot)$ is the fitness function, and b_i and x_i are the i-th bits of the best solution $\mathbf{b}$ and the binary solution $\mathbf{x}$, respectively.

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 have a premature convergence to a local optimum. The sign of $\Delta\theta_i$ determines the direction of convergence.

viii) If the best solution among $P(t)$ is fitter than the stored best solution $\mathbf{b}$, the stored solution $\mathbf{b}$ is replaced by the new one.

The binary solutions in $P(t)$ are discarded at the end of the loop because $P(t+1)$ will be produced by observing the updated $Q(t)$ in step vii). Until the termination condition is satisfied, QEA is running in the **while** loop.

4. Characteristics of QEA

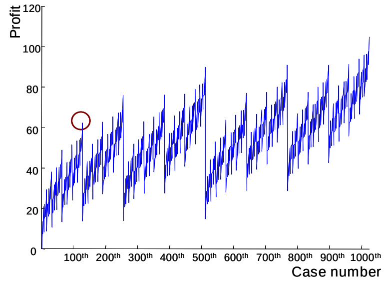
To investigate the characteristics of QEA, a simple knapsack problem with only 10 items is considered. The knapsack problem can be described as selecting, from among various items, those items which are most profitable, given that the knapsack has limited capacity. The 0-1 knapsack problem is described as follows: given a set of m items and a knapsack, select a subset of the items to maximize the profit $f(\mathbf{x})$:

$$f(\mathbf{x}) = \sum_{i=1}^{m} p_i x_i,$$

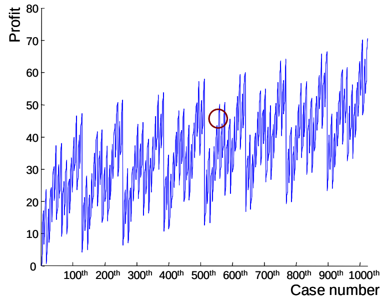
subject to

$$\sum_{i=1}^{m} w_i x_i \leq C,$$

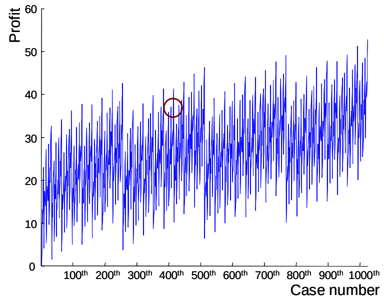
where $\mathbf{x} = (x_1 \cdots x_m)$, x_i is 0 or 1, p_i is the profit of item i, w_i is the weight of item i, and C is the capacity of the knapsack. If $x_i = 1$, the i-th item is selected for the knapsack.



(a) Strongly correlated data



(b) Weakly correlated data

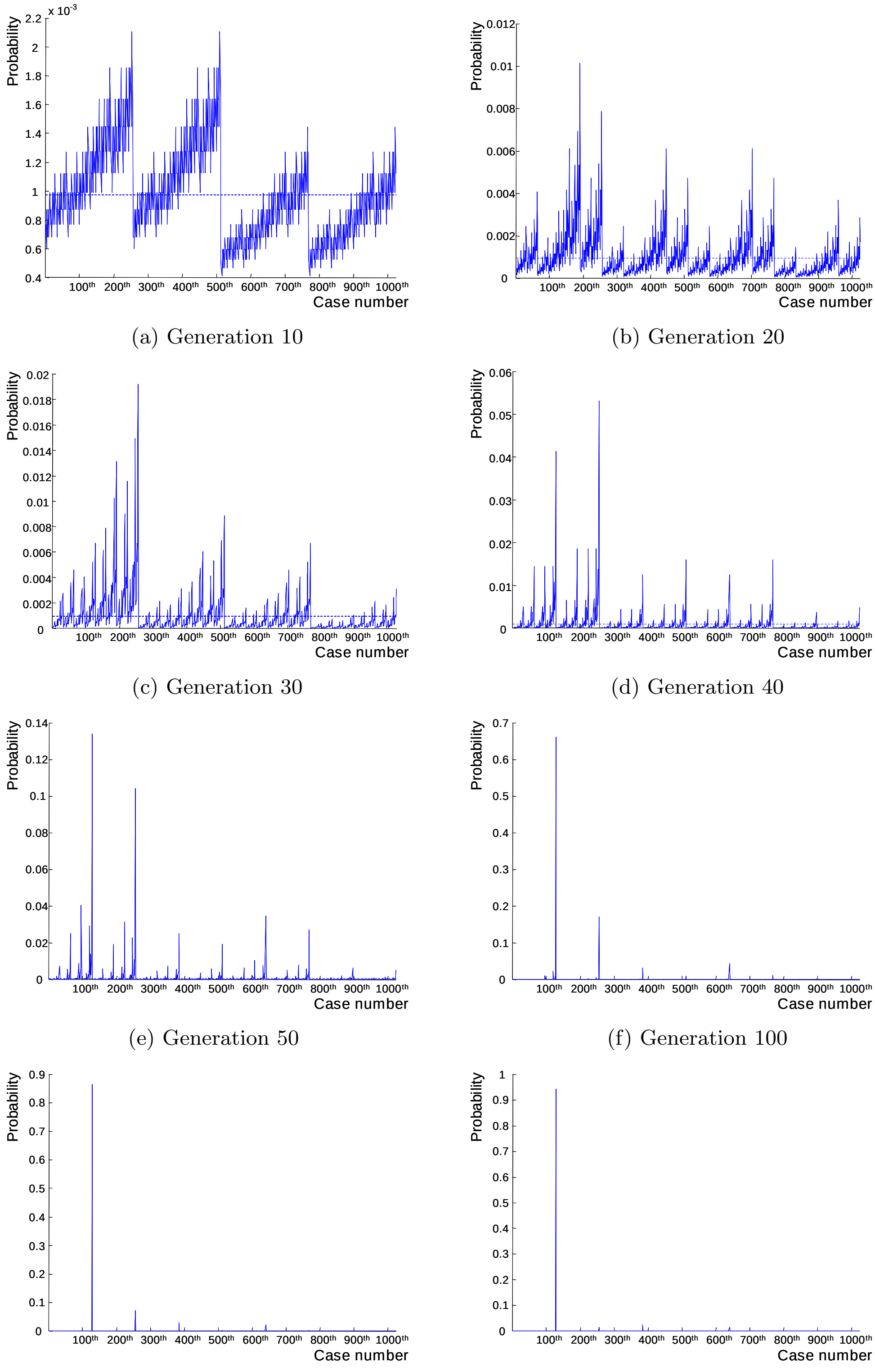


(c) Uncorrelated data

Fig. 2. Profit values of 1024 cases in the knapsack problem with 10 items obtained by a simple calculation. The vertical axis is the profit values of the knapsack, and the horizontal axis is the number of 1024 cases selected as a subset from 10 items. The best profit satisfying the capacity constraint is marked with O.

In our experiments, strongly correlated, weakly correlated, and uncorrelated sets of data were considered, and the average knapsack capacity was used. While selecting a subset from 10 items, there exist 2^{10} cases. By a simple calculation, we could obtain the profit values of 1024 cases in the knapsack problem as shown in Figure 2. The best profit satisfying the capacity constraint was (a) 62.192938 at the 127th case, (b) 46.263497 at the 557th case, and (c) 37.134556 at the 415th case.

Now, to investigate the characteristics of QEA, a single Q-bit individual was used. A rotation gate was used for the Q-gate, and the parameter setting of Table 1 was $(0,\ 0,\ 0.01\pi,\ 0,\ -0.01\pi,\ 0,\ 0,\ 0)$. Figure 3 shows the probabilities of 1024 solutions for the strongly correlated data using the Q-bit individual at generations 10, 20, 30, 40, 50, 100, 200 and 300. Since all the possible solutions of the Q-bit individual are initialized with the same probability as described in (7), we have a probability of 0.001 ($\frac{1}{\sqrt{2^{10}}}^{2} = \frac{1}{2^{10}}$) for each solution which is shown in Figure 3 (a), (b), and (c)



(a) Generation 10 (b) Generation 20

(c) Generation 30 (d) Generation 40

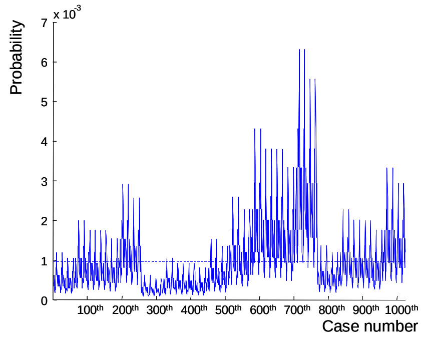
(e) Generation 50 (f) Generation 100

(g) Generation 200 (h) Generation 300

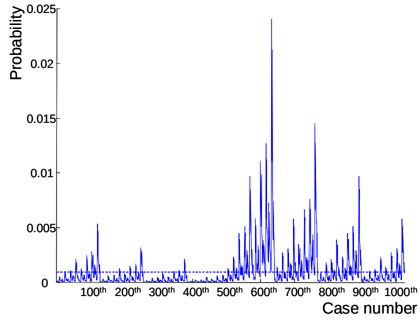
Fig. 3. Probabilities of all solutions for the strongly correlated data using a Q-bit individual. The vertical axis is the probability of the solution, and the horizontal axis is the number of 1024 cases selected as a subset from 10 items.

as a horizontal line. It means that QEA initially starts with a random search.

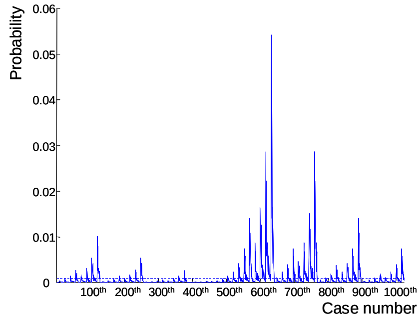
The result at generation 10 is worth mentioning as the probabilities of 1024 solutions had a pattern similar to the profit distribution of Figure 2 (a). This suggests that it may be possible to use only one Q-bit individual to represent 1024 cases. At generation 20, solutions with larger probability appeared. From generations 30 to 50, the probabilities of the solutions with larger profits increased on a large scale. At generation 100, however, all the peak values decreased except for those with better solutions. The same feature was obtained at generation 200. At generation 300, the probability of the best solution was over 0.9, and those of the



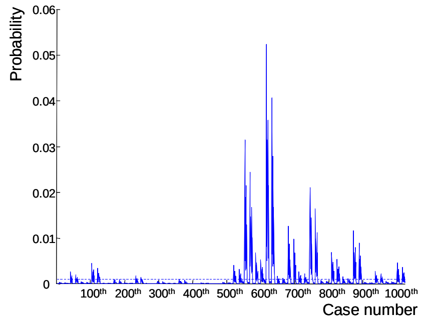
(a) Generation 10



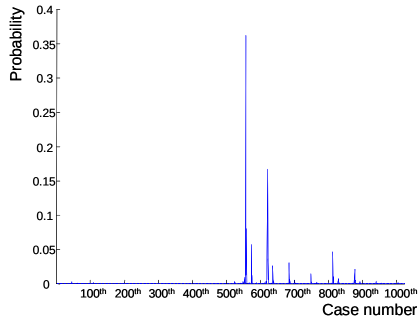
(b) Generation 20



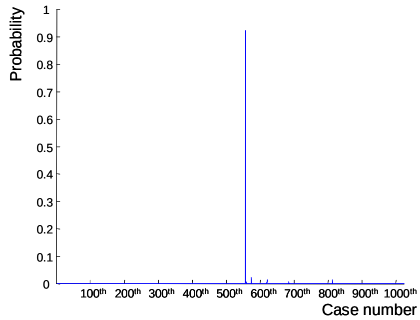
(c) Generation 30



(d) Generation 50

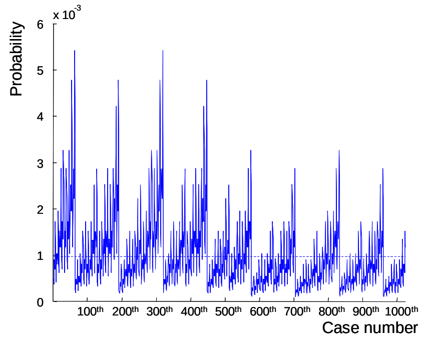


(e) Generation 100

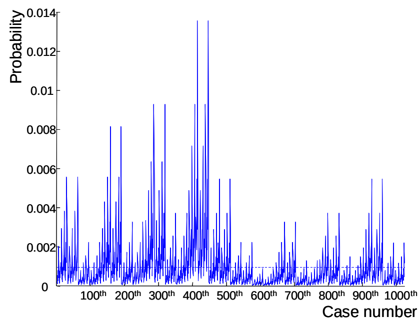


(f) Generation 300

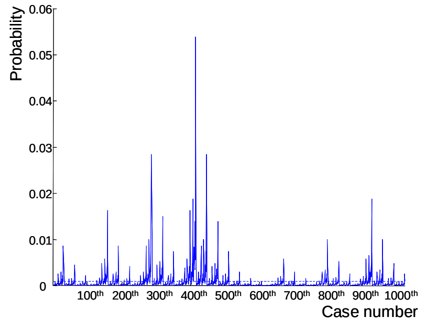
Fig. 4. Probabilities of all solutions for the weakly correlated data using a Q-bit individual. The vertical axis is the probability of the solution, and the horizontal axis is the number of 1024 cases selected as a subset from 10 items.



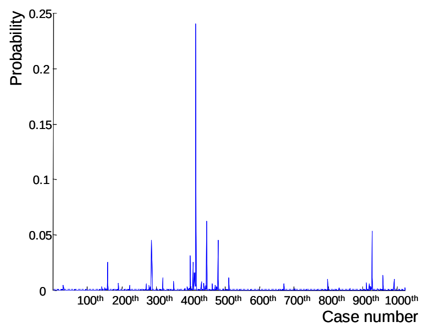
(a) Generation 10



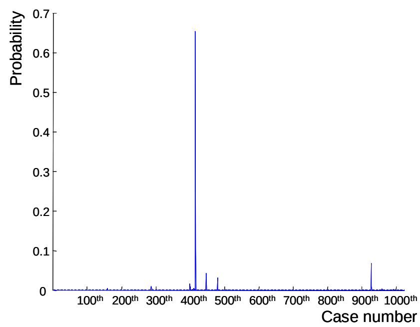
(b) Generation 20



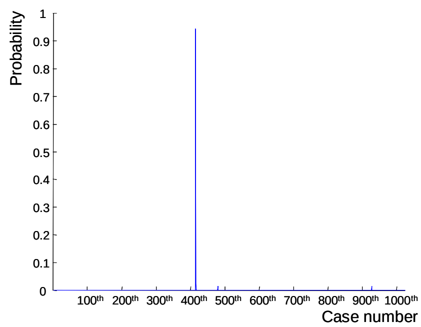
(c) Generation 30



(d) Generation 50



(e) Generation 100



(f) Generation 300

Fig. 5. Probabilities of all solutions for the uncorrelated data using a Q-bit individual. The vertical axis is the probability of the solution, and the horizontal axis is the number of 1024 cases selected as a subset from 10 items.

other solutions were around 0. This means that the Q-bit individual had almost converged to the best solution.

The results above can be summarized in the following. Initially, QEA starts with a random search. At generation 10, the distribution of the probabilities of all the solutions becomes similar to the profit distribution shown in Figure 2 (a). As the probabilities of the solutions with larger profits increase, QEA starts a local search. Finally, the probability of the best Q-bit individual converges to 1. This means that

QEA starts with a global search and switches automatically to a local search because of its inherent probabilistic mechanism, which leads to a good balance between exploration and exploitation.

As shown in Figure 4 and Figure 5, the results for the weakly correlated and uncorrelated data also demonstrated the effectiveness and the applicability of QEA as the results for the strongly correlated data did.

5. Conclusions

This paper has introduced a quantum-inspired evolutionary algorithm (QEA), which is based on the concept and principles of quantum computing such as a quantum bit and the superposition of states. The key ideas of QEA lie in the concept of a Q-bit individual, defined as a string of Q-bits for the probabilistic representation, and the operation of a Q-gate, designed to introduce variation to the Q-bit individual. To investigate the characteristics of QEA, the knapsack problem has been studied. The experimental results demonstrate the effectiveness and applicability of QEA for this class of combinatorial optimization problems.

References

[1] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *Journal of Statistical Physics*, 22, pp. 563-591, 1980.

[2] D. Deutsch, "Quantum Theory, the Church-Turing principle and the universal quantum computer," in *Proceedings of the Royal Society of London*, A 400, pp. 97-117, 1985.

[3] P. W. Shor, "Quantum Computing," *Documenta Mathematica*, vol. Extra Volume ICM, pp. 467-486, 1998, http://east.camel.math.ca/ EMIS/ journals/ DMJDMV/ xvol-icm/ 00/ Shor.MAN.html.

[4] P. W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, IEEE Press, pp. 124-134, 1994.

[5] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th ACM Symposium on Theory of Computing*, pp. 212-219, 1996.

[6] L. K. Grover, "Quantum Mechanical Searching," in *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 2255-2261, Jul 1999.

[7] L. Spector, H. Barnum, H. J. Bernstein and N. Swamy, "Finding a Better-than-Classical Quantum AND/OR Algorithm using Genetic Programming," in *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 2239-2246, Jul 1999.

[8] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proceedings of IEEE International Conference on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 61-66, 1996.

[9] A. Narayanan, "Quantum computing for beginners," in *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 2231-2238, Jul 1999.

[10] K.-H. Han and J.-H. Kim, "Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem," in *Proceedings of the 2000 Congress on Evolutionary Computation*, IEEE Press, pp. 1354-1360, July 2000.

[11] K.-H. Han, K.-H. Park, C.-H. Lee and J.-H. Kim, "Parallel Quantum-inspired Genetic Algorithm for Combinatorial Optimization Problem," in *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE Press, May 2001.

[12] K.-H. Han and J.-H. Kim, "Analysis of Quantum-Inspired Evolutionary Algorithm," in *Proceedings of the 2001 International Conference on Artificial Intelligence*, CSREA Press, Vol. 2, pp. 727-730, June 2001.

[13] T. Hey, "Quantum computing: an introduction," *Computing & Control Engineering Journal*, pp. 105-112, Jun 1999.

[14] A. O. Pittenger, *An Introduction to Quantum Computing Algorithms*, Birkhäuser, 2000.

[15] R. Hinterding, "Representation, Constraint Satisfaction and the Knapsack Problem," in *Proceedings of the 1999 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, pp. 1286-1292, Jul 1999.