On Setting the Parameters of Quantum-inspired Evolutionary Algorithm for Practical Applications

Kuk-Hyun Han **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
{khhan, johkim}@rit.kaist.ac.kr

Abstract- In this paper, some guidelines for setting the parameters of quantum-inspired evolutionary algorithm (QEA) are presented. QEA is based on the concept and principles of quantum computing, such as a quantum bit and superposition of states. However, QEA is not a quantum algorithm, but a novel evolutionary algorithm. Like other evolutionary algorithms, QEA is also characterized by the representation of the individual, the evaluation function, and the population dynamics. From recent research on the knapsack problem, the results of QEA are better than those of CGA (conventional GA). Although the performance of QEA is excellent, there is relatively little or no research on the effects of different settings for its parameters. This paper describes some guidelines for setting these parameters. The guidelines are drawn up based on extensive experiments carried out for a class of combinatorial and numerical optimization problems. Through the guidelines, the performance of QEA can be maximized.

1 Introduction

Evolutionary algorithms (EAs) are principally a stochastic search and optimization method based on the principles of natural biological evolution. Compared to traditional optimization methods, such as calculus-based and enumerative strategies, EAs are robust, global, and may be applied generally without recourse to domain-specific heuristics, although their performance may be affected by these heuristics. Overviews of current state of the art in the field of evolutionary computation are given by Fogel [1] and Bäck [2].

Quantum-inspired evolutionary algorithm (QEA) recently proposed in [3] can treat the balance between exploration and exploitation more easily when compared to conventional GAs (CGAs). Also, QEA can explore the search space with a smaller number of individuals and exploit the search space for a global solution within a short span of time. QEA is based on the concept and principles of quantum computing[1], such as the quantum bit and the superposition of states. However, QEA is not a quantum algorithm, but a novel evolutionary algorithm as shown in Figure 1. Like any other EAs, QEA is also characterized by the representation of the individual, the evaluation function, and the population dynamics.

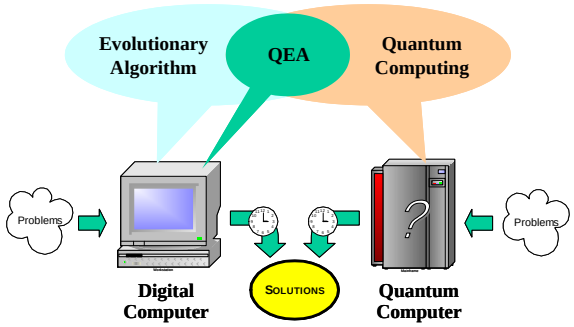


Figure 1: Quantum-inspired evolutionary algorithm (QEA)

Unlike other research areas, there has been relatively little work done in applying quantum computing to evolutionary algorithms. Quantum-inspired computing was introduced in [13]. In [14], a modified crossover operator which includes the concept of interference was introduced. In [15], a probabilistic representation and a novel population dynamics inspired by quantum computing were proposed. In [16], the applicability of QEA to a parallel scheme, particularly, PC clustering, was verified successfully. In [3], the basic structure of QEA and its characteristics were formulated and analyzed, respectively. According to [3], the results (tested on the knapsack problem) of QEA were proved to be better than those of CGA. In [17], a QEA-based disk allocation method (QDM) was introduced. According to the results, the average query response times of QDM were equal to or less than those of DAGA (disk allocation methods using GA), and the convergence speed of QDM was 3.2-11.3 times faster than that of DAGA. In [18], QEA was applied to a decision boundary optimization for face verification. The proposed face verification system was tested by face and non-face images extracted from AR face database. Compared to the conventional PCA (principal components analysis) method, improved results were achieved both in terms of the face verification rate and false alarm rate.

[1]Quantum computing is a research area which includes quantum mechanical computers and quantum algorithms. Quantum mechanical computers were proposed in the early 1980s [4], [5] and their description was formalized in the late 1980s [6]. Many efforts on quantum computers have progressed actively since the early 1990s because these computers were shown to be more powerful than digital computers on various specialized problems. There are well-known quantum algorithms such as Deutsch-Jozsa algorithm [7], Simon's algorithm [8], Shor's quantum factoring algorithm [9], and Grover's database search algorithm [10]. In particular, since the difficulty of the factoring problem is crucial for the security of the RSA cryptosystem [11] which is in widespread use today, interest in quantum computing is increasing [12].

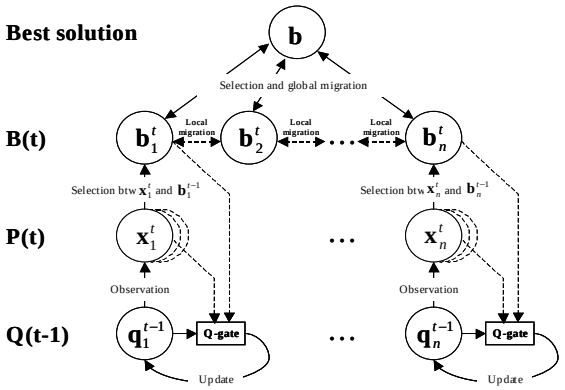


Figure 2: Overall structure of QEA

This paper proposes some guidelines for setting the parameters of QEA. These guidelines are drawn up based on empirical results. The effects of different settings for the parameters are examined from several experiments on optimization problems.

This paper is organized as follows. Section 2 describes the QEA. Section 3 verifies the rotation angle selection for Q-gate. Section 4 examines the effects of different parametric settings of QEA. Concluding remarks follow in Section 5.

2 QEA

2.1 Representation

QEA uses a novel Q-bit representation which is a kind of probabilistic representation. A Q-bit is defined as the smallest unit of information in QEA, which is defined as a pair of numbers, (α, β), 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 states.

A Q-bit individual as a string of *m* Q-bits is defined as

$$\left[\begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array}\right], \qquad (1)$$

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 probabilistically. If there is, for instance, a three-Q-bit system with three pairs of amplitudes, the system can contain the information of eight states. The Q-bit representation has a better characteristic of population diversity than other representations.

2.2 Basic structure of QEA

QEA is a probabilistic algorithm similar to 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 $\mathbf{q}_j^t$, $j = 1, 2, \cdots, n$, is a Q-bit individual defined as (1).

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 $B(t)$
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 $B(t-1)$ and $P(t)$ into $B(t)$
ix) store the best solution $\mathbf{b}$ among $B(t)$
x) **if** (global migration condition) **then** migrate $\mathbf{b}$ to $B(t)$ globally
xi) **else if** (local migration condition) **then** migrate $\mathbf{b}_j^t$ in $B(t)$ to $B(t)$ locally
end
end

Figure 3: Procedure QEA.

Figure 2 shows the overall structure of QEA and Figure 3 shows the procedure QEA that can be explained in the following manner.

i) In the step of 'initialize $Q(t)$,' α_i^0 and β_i^0, $i = 1, 2, \cdots, m$, of all $\mathbf{q}_j^0$, $j = 1, 2, \cdots, n$, are initialized with $\frac{1}{\sqrt{2}}$. It means that one Q-bit individual $\mathbf{q}_j^0$ represents the linear superposition of all the possible states with the same probability.

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 measure of its fitness.

iv) The initial best solutions are then selected among the binary solutions $P(0)$, and stored into $B(0)$, where $B(0) = \{\mathbf{b}_1^0, \mathbf{b}_2^0, \cdots, \mathbf{b}_n^0\}$, and $\mathbf{b}_j^0$ is the same as $\mathbf{x}_j^0$ at the initial generation.

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. It should be noted that $\mathbf{x}_j^t$ in $P(t)$ can be formed by multiple observations of $\mathbf{q}_j^{t-1}$ in $Q(t-1)$.

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 basic Q-gate in QEA, such as

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \quad (2)$$

where $\Delta\theta_i, i = 1, 2, \cdots, m$, is a rotation angle of each Q-bit toward either 0 or 1 state depending on its sign. $\Delta\theta_i$ should be designed in compliance with the application problem.

viii, ix) The best solutions among $B(t-1)$ and $P(t)$ are selected and stored into $B(t)$, and if the best solution stored in $B(t)$ is better fitted than the stored best solution $\mathbf{b}$, the stored solution $\mathbf{b}$ is replaced by the new one.

x) If the global migration condition is satisfied, the best solution $\mathbf{b}$ is migrated to $B(t)$ globally.

xi) If the local migration condition is satisfied, the best one in a local group in $B(t)$ is migrated to others in the same local group.

Until the termination condition is satisfied, QEA is running in the **while** loop.

3 Rotation angles for Q-gate

In [3], Table 1 was suggested to guide the selection of the angle parameters for the rotation gate given in (2). It was suggested and verified to set a positive number p for θ_3, a negative number n for θ_5, and 0 for the rest of the angle parameters in Θ of Table 1 for the knapsack problem. In particular, the empirical results showed that θ_2, θ_4, θ_6, and θ_8 could be set to any one among 0, p, and n. However, it is questionable whether it is possible to use Θ of Table 1 for other problems.

x_i	b_i	$f(\mathbf{x}) < f(\mathbf{b})$	$\Delta\theta_i$
0	0	$true$	$\theta_1 = 0$
0	0	$false$	$\theta_2 = *$
0	1	$true$	$\theta_3 = p$
0	1	$false$	$\theta_4 = *$
1	0	$true$	$\theta_5 = n$
1	0	$false$	$\theta_6 = *$
1	1	$true$	$\theta_7 = 0$
1	1	$false$	$\theta_8 = *$

Table 1: Lookup table of $\Delta\theta_i$ suggested from [3], where $f(\cdot)$ is the fitness, and b_i and x_i are the ith bits of the best solution $\mathbf{b}$ and the binary solution $\mathbf{x}$, respectively. $*$ is one of (0, p, and n), where p is a positive number and n is a negative number.

Three numerical problems are considered to show that the results on Θ can be applied to other optimization problems. To deal with numerical problems, real values of the variables should be encoded as binary strings since QEA uses a Q-bit representation to generate a binary bit. The three numerical problems are as follows:

Problem 1: Maximize $f_1(\mathbf{x}) = 100 - (100(x_1^2 - x_2)^2 + (1 - x_1)^2)$, where $-2.048 \le x_i \le 2.048$. The global maximum value is 100 at $(x_1, x_2) = (1, 1)$. This function is a modified version of De Jong function F2.

Problem 2: Maximize $f_2(\mathbf{x}) = -\sum_{i=1}^{5} integer(x_i)$, where $-5.12 \le x_i \le 5.12$. The global maximum value is 30 for all $-5.12 \le x_i < -5.0$. This function is a modified version of De Jong function F3.

Problem 3: Maximize $f_3(\mathbf{x}) = -\frac{1}{\frac{1}{K} + \sum_{j=1}^{25} g_j^{-1}(x_1, x_2)} + 100.98$, where $g_j(x_1, x_2) = c_j + \sum_{i=1}^{2}(x_i - a_{ij})^6$, where $-65.536 \le x_i \le 65.536$, $K = 500$, $c_j = j$, and $[a_{ij}] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \cdots & 32 & 32 & 32 \end{bmatrix}$.

The global maximum value is 100 at $(x_1, x_2) = (-32, -32)$. This function is a modified version of De Jong function F5.

Each variable was encoded as a 25-bit string. The population size was 1. The maximum number of generations was $1,000$. The values of 0, 0.005π, and -0.005π were used for each of the eight angle parameters. The experiments of Problems 1, 2, and 3, respectively, carried out step by step to find proper signs $(0, p, n)$ of the angle parameters. The results on θ_2, θ_4, θ_6, and θ_8, that is, the cases which $f(\mathbf{x}) < f(\mathbf{b})$ is false, were worthwhile to mention that the values of θ_2, θ_4, θ_6, and θ_8 had little effect on the performance. These are the same results of the knapsack problem as shown in [3]. In particular, the set of Θ for finding the maximum value of each problem was obtained from the experimental results as follows:

- f_1: $[0 * p * 0 * n *]^T$, $[0 * p * 0 * 0 *]^T$, $[0 * p * n * 0 *]^T$, $[p * p * n * 0 *]^T$, and $[p * p * n * n *]^T$;
- f_2: $[0 * p * n * 0 *]^T$, $[0 * p * n * n *]^T$, $[0 * n * n * 0 *]^T$, $[0 * n * n * n *]^T$, $[0 * 0 * n * 0 *]^T$, $[0 * 0 * n * n *]^T$, $[n * n * n * 0 *]^T$, and $[n * n * n * n *]^T$;
- f_3: $[0 * p * n * 0 *]^T$, $[0 * p * n * n *]^T$, $[p * p * n * 0 *]^T$, and $[p * p * n * n *]^T$.

Table 2 shows the average frequencies of 0, p, and n for each θ_i in Θ from the above results. From the table, $[0 * p * n * 0 *]^T$ has a higher frequency and is included in each set of Θ for Problems 1, 2, and 3. It means that Θ can be assigned as $[0 * p * n * 0 *]^T$ for other problems.

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
$f_a(0)$	**0.62**	$*$	0.08	$*$	0.13	$*$	**0.53**	$*$
$f_a(p)$	0.3	$*$	**0.75**	$*$	0	$*$	0	$*$
$f_a(n)$	0.08	$*$	0.17	$*$	**0.87**	$*$	0.47	$*$

Table 2: Average frequencies of 0, p, and n for each θ_i in Θ from Problems 1, 2, and 3. $f_a(\cdot)$ is the average frequency to find the global maximum and its value is scaled between 0 and 1.

From the empirical results, Table 1 for the rotation gate can be simplified as Table 3. The magnitude of $\Delta\theta_i$ has an effect on the speed of convergence, but if it is too big, the solutions may diverge or converge prematurely to a local optimum. The values from 0.001π to 0.1π are recommended for the magnitude of $\Delta\theta_i$, although they depend on the problems. The sign of $\Delta\theta_i$ determines the direction of convergence. It should be noted that θ_1 and θ_7 can be assigned nonzero values in compliance with the application problems.

	x_i	b_i	$\Delta\theta_i$	$rec.$
$f(\mathbf{x}) < f(\mathbf{b})$ $(true)$	0	0	θ_1	0
	0	1	θ_3	p
	1	0	θ_5	n
	1	1	θ_7	0

Table 3: Simplified lookup table of $\Delta\theta_i$, where b_i and x_i are the ith bits of the best solution $\mathbf{b}$ and the binary solution $\mathbf{x}$, respectively. $rec.$ means the recommended value of $\Delta\theta_i$. p is a positive number, and n is a negative number.

4 Effects of different parametric settings

In this section, the effects of changing parameters (such as the population size, the global and local migration periods, and the rotation angles) of QEA are investigated.

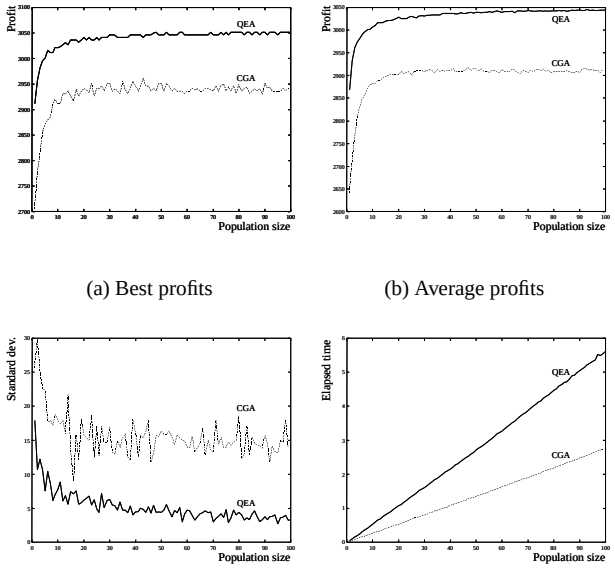
4.1 Population size

To investigate the effects of changing the population size of QEA, the knapsack problem with 500 items considered in [3] was used. The population size was tested from 1 to 100. The rotation gate was used for Q-gate. The values of 0.01π, -0.01π, and 0 were used for θ_3, θ_5, and the rest of Θ, respectively. The global migration period in generation was 100, and the local migration period was 1. The local group size n_g was set as

$$n_g = \max\left(integer\left(\frac{n}{5}\right), 1\right), \tag{3}$$

where n is the population size. For the comparison purpose, the conventional GA (CGA) which outperformed all other CGAs which were considered in [3] was tested. The values of 0.001 and 0.7 for the mutation and crossover probabilities, respectively, were selected for CGA ($Rep2$). The maximum number of generations was 1,000.

Figure 4 shows the results on the effects of changing the population sizes of QEA and CGA. In Figure 4 (a) and (b), the profits increased fast until the population size was 10-20, however the increasing rate was nearly constant after the population size reached 30. The tendency of the results on QEA was similar to that of CGA. However, it should be noted that the best and average profits of QEA with population size 2 were better than those of CGA with population size 100 (according to (d), the convergence speed of QEA with population size 2 was 29 times faster than that of CGA with population size 100). In Figure 4 (c), it is also worthwhile to mention that the standard deviation of the best profits of QEA over 30 runs decreased as population size increased. It means that the larger population size could provide better robustness for QEA. However, this relation between population size and robustness did not appear in the result of CGA after the population size reached 20. The processing time of QEA was about twice of the CGA's for the same population size. This is because QEA uses Q-bit individuals as a population. Q-bit individuals need floating point calculations to represent the corresponding probabilities. However, it should be noted that the processing time of QEA is linearly proportional to the population size n, which is the same as CGA's.



(a) Best profits (b) Average profits

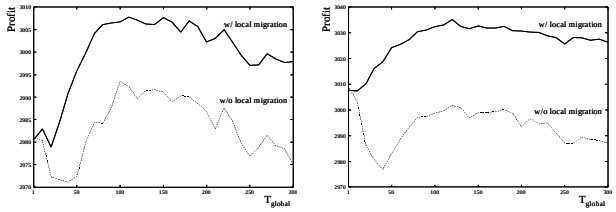
(c) Standard deviation (d) Processing time (sec .)

Figure 4: Effects of changing the population sizes of QEA and CGA for the knapsack problem with 500 items. The global migration period and the local migration period were 100 and 1, respectively. The local group size was set as (3). The results were averaged over 30 runs.

4.2 Global and local migrations

The performance of QEA with a few individuals was already verified in the previous results. It means that QEA can be easily extended to a parallel scheme as proposed in the structure of QEA. Since the parallel scheme can increase the population diversity, it helps QEA to explore the search space effectively.

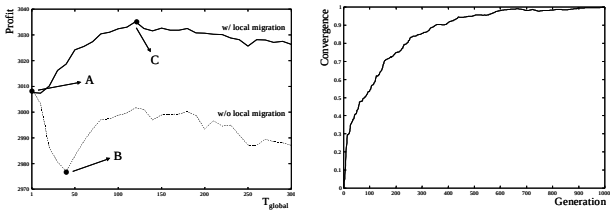
To show the effects of changing the global migration period, the same knapsack problem with 500 items was considered. The population sizes of 10, 30, and 50 were tested. The maximum number of generations was 1,000. To investigate the effects of using the local migration, QEAs with local migration and without local migration were considered for each population size. Figure 5 shows the effects of changing the global migration period in QEAs with and without local migration. The local group size of QEA with local migration was set to be the same as (3). In the results of QEA without local migration, an undershooting point at near 40 was found, since the increasing diversity from the global migration disturbed the convergence of homogeneous individuals. In the results of QEA with local migration, the undershooting point disappeared. This is because the local migration with period 1 guaranteed the convergence of homogeneous individuals in the same local group. From these results, we can say that it is desirable that the local migration period be set to 1 to guarantee the convergence of homogeneous individuals. It is also worthwhile to mention that the best results were found at the global migration



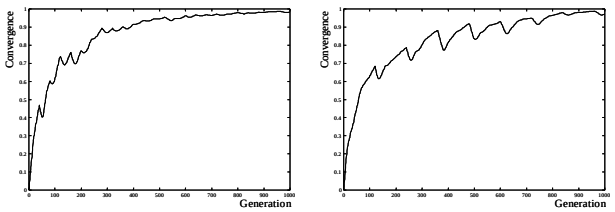
(a) Population size 10 (b) Population size 30

(c) Population size 50

Figure 5: Effects of changing the global migration period in QEAs with and without local migration for the knapsack problem with 500 items. The global migration period T_{global} was set to the values ranging from 1 to 300. For the QEA with local migration, the period was 1 and the local group size was set as (3). The profits were averaged over 30 runs.



(a) Effects of migrations (b) Q-bit convergence (A)



(c) Q-bit convergence (B) (d) Q-bit convergence (C)

Figure 6: Relations between migration and Q-bit convergence for the knapsack problem with 500 items. The population size was 30. The global migration periods of (b), (c), and (d) were 1, 40, and 120, respectively. The local group sizes of (b), (c), and (d) were 1, 1, and 6, respectively.

period between 100 and 150, although the migration period could be affected by other parameters. Consequently, the global migration period should be set properly considering the convergence period of the local groups.

Definition 1 A *Hamming distance* H of the two binary strings, $\mathbf{x}_1$ and $\mathbf{x}_2$, is defined as the number of their bitwise-different bits, which is defined as

$$H(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{m} |x_{1i} - x_{2i}|$$

where m is the binary string length.

Figure 6 shows the relations between migration and Q-bit convergence. While the profits of the point A (Figure 6 (b)) increased continuously without perturbation, those of the point B (Figure 6 (c)) and the point C (Figure 6 (d)) increased with perturbation. The perturbation was caused by the global migration. The difference of the perturbation level between the points B and C can be explained by using the concept of Hamming distance. When a new best solution comes from the neighbor local group through the global migration:

i) if the new best solution has a large Hamming distance from the current best solution, the Q-bit individual varies largely to adapt the new one;

ii) if the new best solution has a small Hamming distance from the current best solution, the Q-bit individual changes a little. In this case, the Q-bit individual has a chance to have a premature convergence to a local optimum.

4.3 Rotation angles

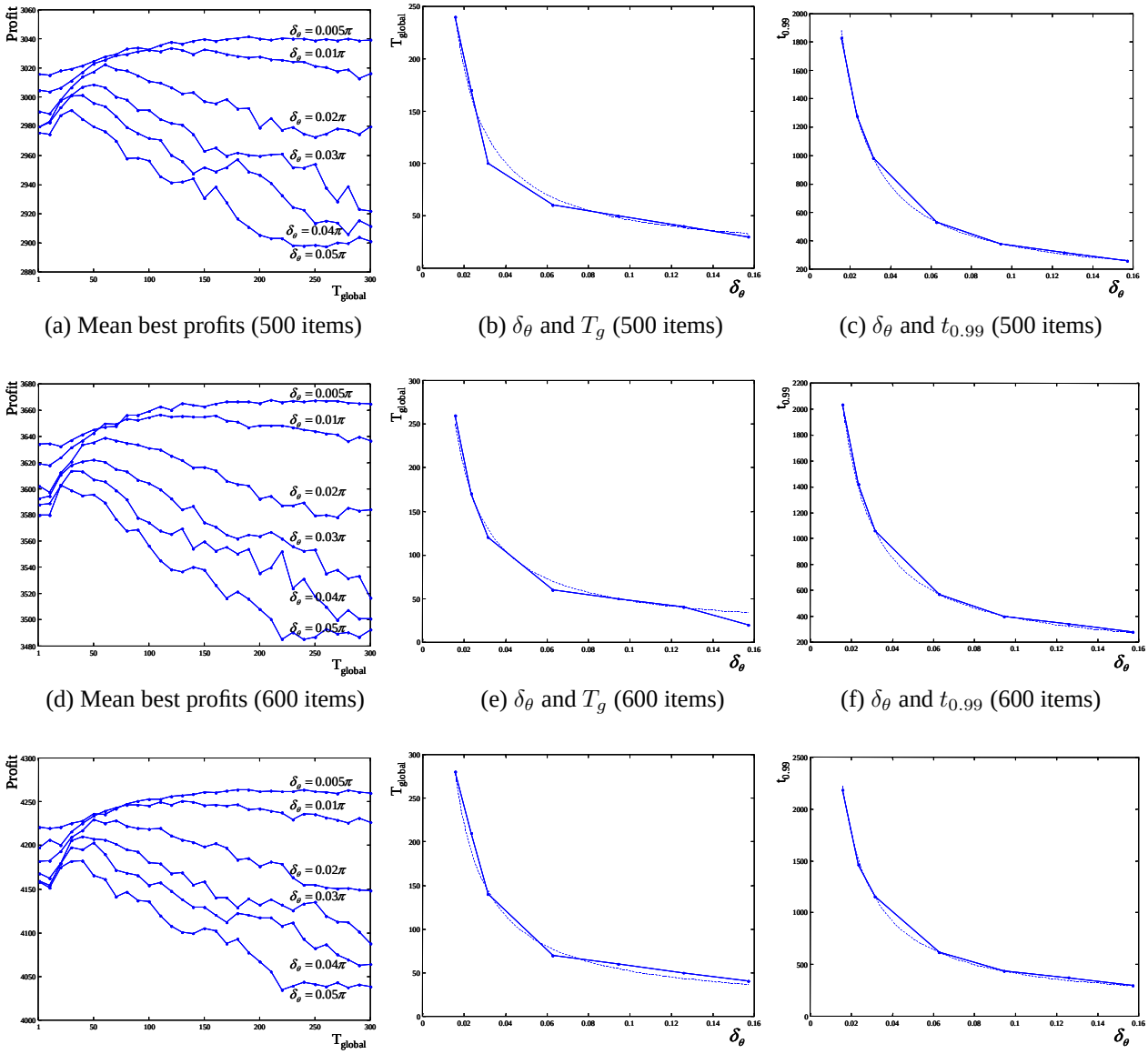
In the previous empirical results, the best results on the knapsack problem with 500 items were found at the global migration period between 100 and 150. And the rotation angle of p (or $|n|$) was set to 0.01π. However, if the rotation angle is changed, the global migration period for inducing the best result may be changed. If the value of rotation angle is smaller, the global migration period must be larger, since the convergence speed is changed to be slower.

Here, to investigate the effects of changing the rotation angles, the knapsack problems with 500, 600, and 700 items were considered. The population size and the local group size were set to 30 and 6, respectively. The local migration period was set to 1. The termination condition of Q-bit convergence [19] was used instead of MAXGEN and the value of γ was set to 0.99.

Figure 7 (a) shows the effects of changing the value of rotation angle ranging from 0.005π to 0.05π. As shown in this figure, there is a peak value of the mean best profits for the same rotation angle δ_θ. The value of global migration period for the peak is larger as the value of rotation angle is smaller. Figure 7 (b) shows the relation between the rotation angle and the global migration period for each peak. It shows that the rotation angle is inversely proportional to the global migration period. The result is approximately the same as

$$T_g = \frac{1.15\pi}{\delta_\theta} + 10. \quad (4)$$

Also, in Figure 7 (c), the running number of generations where the algorithm is terminated by the termination condi-



(a) Mean best profits (500 items) (b) δ_θ and T_g (500 items) (c) δ_θ and $t_{0.99}$ (500 items)

(d) Mean best profits (600 items) (e) δ_θ and T_g (600 items) (f) δ_θ and $t_{0.99}$ (600 items)

(g) Mean best profits (700 items) (h) δ_θ and T_g (700 items) (i) δ_θ and $t_{0.99}$ (700 items)

Figure 7: Effects of changing the rotation angles for the knapsack problems with 500, 600, and 700 items. The global migration period was set to the values ranging from 1 to 300. The population size and the local group size were set to 30 and 6, respectively. The termination condition of Q-bit convergence [19] was used and the value of γ was set to 0.99. All the results were averaged over 30 runs. δ_θ is the rotation angle of p (or $|n|$), T_g the global migration period, and $t_{0.99}$ the number of generations where the algorithm is terminated by the termination condition for $\gamma = 0.99$. The dotted lines of (b), (e), and (h) are $T_g = \frac{1.15\pi}{\delta_\theta} + 10$, $T_g = \frac{1.2\pi}{\delta_\theta} + 10$, and $T_g = \frac{1.34\pi}{\delta_\theta} + 10$, respectively, and those of (c), (f), and (i) are $t_{0.99} = \frac{9.0\pi}{\delta_\theta} + 80$, $t_{0.99} = \frac{9.8\pi}{\delta_\theta} + 80$, and $t_{0.99} = \frac{10.7\pi}{\delta_\theta} + 80$, respectively.

tion for $\gamma = 0.99$ is approximately the same as

$$t_{0.99} = \frac{9.0\pi}{\delta_\theta} + 80. \tag{5}$$

Figure 7 (d)-(i) show the similar results in the relations among the rotation angle, the global migration period, and the running number of generations to those of Figure 7 (a)-(c).

Consequently, the relation between the rotation angle and the global migration can be approximated as

$$T_g = \frac{\lambda_g}{\delta_\theta} + k_g, \tag{6}$$

where $\lambda_g > 0$ and $k_g > 0$. The relation between the rotation angle and the running number of generations can be also approximated as

$$t_\gamma = \frac{\lambda_\gamma}{\delta_\theta} + k_\gamma, \tag{7}$$

where $\lambda_\gamma > 0$ and $k_\gamma > 0$.

It is worthwhile to mention that k_g of (6) and k_γ of (7) are nonzero values, since each Q-bit is not updated when the current best solution is changed to the current observed solution in the update procedure. As the result of the knapsack problem with 500 items for checking how many times

the current best solution changes during the running number of generations, the best solution $\mathbf{b}$ was changed about 80 times and the best solution $\mathbf{b}_j$ for each individual was changed about 10 times, where the rotation angle and the global migration period were set to 0.01π and 100, respectively.

5 Conclusions

This paper examined the effects of changing the parameters of QEA to provide some useful guidelines for application users. The values of the rotation angles for Q-gate were recommended reasonably. Moreover, the relation between the population size and the performance showed that the larger population size could provide better robustness for QEA. In particular, the empirical results for different migration periods showed the importance of the global and local migrations, and could provide specific values for the migration periods. Also, the relation among the rotation angle, the global migration period, and the running number of generations was presented. These guidelines can help researchers and engineers who want to use QEA for their application problems.

References

[1] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd ed. Piscataway, NJ: IEEE Press, 2000.

[2] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. New York: Oxford University Press, 1996.

[3] K.-H. Han and J.-H. Kim, "Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization," *IEEE Transactions on Evolutionary Computation*, Piscataway, NJ: IEEE Press, vol. 6, no. 6, pp. 580-593, Dec. 2002.

[4] 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*, vol. 22, pp. 563-591, 1980.

[5] R. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467-488, 1982.

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

[7] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," in *Proceedings of the Royal Society of London A*, vol. 439, pp. 553-558, 1992.

[8] D. R. Simon, "On the Power of Quantum Computation," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, Piscataway, NJ: IEEE Press, pp. 116-123, Nov. 1994.

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

[10] 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.

[11] R. L. Rivest, A. Shamir, and L. Adleman, "A method of obtaining digital signatures and public-keycryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[12] C. P. Williams and S. H. Clearwater, *Explorations in Quantum Computing*. Berlin: Springer-Verlag, 1998.

[13] M. Moore and A. Narayanan, *Quantum-inspired Computing*. Technical report, Department of Computer Science, University of Exeter, UK, 1995.

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

[15] 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*, Piscataway, NJ: IEEE Press, vol. 2, pp. 1354-1360, July 2000.

[16] 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*, Piscataway, NJ: IEEE Press, vol. 2, pp. 1422-1429, May 2001.

[17] K.-H. Kim, J.-Y. Hwang, K.-H. Han, J.-H. Kim, and K.-H. Park, "A Quantum-inspired Evolutionary Computing Algorithm for Disk Allocation Method," *IEICE Transactions on Information and Systems*, IEICE Press, vol. E86-D, no. 3, pp. 645-649, Mar. 2003.

[18] J.-S. Jang, K.-H. Han, and J.-H. Kim, "Quantum-inspired Evolutionary Algorithm-based Face Verification," in *Lecture Notes in Computer Science (GECCO 2003)*, eds. E. Cantu-Paz et al., Berlin Heidelberg: Springer-Verlag, pp. 2147-2156, July 2003.

[19] K.-H. Han and J.-H. Kim, "Quantum-inspired Evolutionary Algorithms with a New Termination Criterion, H_ϵ Gate, and Two-Phase Scheme," submitted to the *IEEE Transactions on Evolutionary Computation*, Piscataway, NJ: IEEE Press, May 2003.