Quantum-Inspired Evolutionary Algorithms With a New Termination Criterion, H_ϵ Gate, and Two-Phase Scheme

Kuk-Hyun Han, *Associate Member, IEEE,* and Jong-Hwan Kim, *Senior Member, IEEE*

***Abstract*—From recent research on combinatorial optimization of the knapsack problem, quantum-inspired evolutionary algorithm (QEA) was proved to be better than conventional genetic algorithms. To improve the performance of the QEA, this paper proposes research issues on QEA such as a termination criterion, a Q-gate, and a two-phase scheme, for a class of numerical and combinatorial optimization problems. A new termination criterion is proposed which gives a clearer meaning on the convergence of Q-bit individuals. A novel variation operator H_ϵ gate, which is a modified version of the rotation gate, is proposed along with a two-phase QEA scheme based on the analysis of the effect of changing the initial conditions of Q-bits of the Q-bit individual in the first phase. To demonstrate the effectiveness and applicability of the updated QEA, several experiments are carried out on a class of numerical and combinatorial optimization problems. The results show that the updated QEA makes QEA more powerful than the previous QEA in terms of convergence speed, fitness, and robustness.**

***Index Terms*—Initial condition, numerical and combinatorial optimization, Q-bit representation, Q-gate, quantum-inspired evolutionary algorithm (QEA), termination criterion.**

I. Introduction

Evolutionary algorithms (EAs) are principally a stochastic search and optimization method based on the principles of natural biological evolution. Compared with traditional optimization methods, such as calculus-based methods and enumerative strategies, EAs are robust, global in operation, 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].

EAs are characterized by the representation of the individual, the evaluation function representing the fitness level of the individuals, and the population dynamics such as population size, variation operators, parent selection, reproduction and inheritance, survival competition method, etc. To have a good balance between exploration and exploitation, these components should be designed properly.

Manuscript received May 17, 2003; revised October 27, 2003. This work was supported in part by the Brain Korea 21 Project, School of Information Technology, Korea Advanced Institute of Science and Technology (KAIST), and in part by the ITRC Intelligent Robot Research Center (IRRC) at KAIST supported by the Korea Ministry of Information and Communication in 2003.

K.-H. Han was with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea. He is now with the Digital Media R&D Center, Samsung Electronics Company, Ltd., Suwon, Gyeonggi 443-742, Korea (e-mail: khhan@ khhan.com).

J.-H. Kim is with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: johkim@rit.kaist.ac.kr).

Digital Object Identifier 10.1109/TEVC.2004.823467

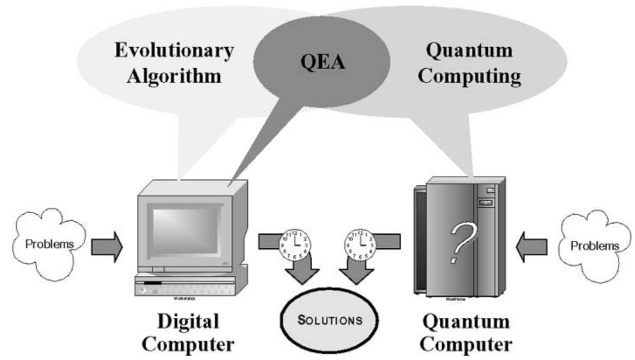


Fig. 1. Quantum-inspired evolutionary algorithm (QEA).

The quantum-inspired evolutionary algorithm (QEA) recently proposed in [3] can treat the balance between exploration and exploitation more easily when compared with conventional genetic algorithms (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, such as the quantum bit and the superposition of states. However, QEA is not a quantum algorithm, but a novel EA [4] as shown in Fig. 1. Like any other EAs, QEA is also characterized by the representation of the individual, the evaluation function, and the population dynamics.

Quantum computing is a research area that includes concepts like quantum mechanical computers and quantum algorithms. Quantum mechanical computers were proposed in the early 1980s [5], [6] and their description was formalized in the late 1980s [7], [8]. Many efforts on quantum computers have progressed actively since the early 1990s because these computers were shown to be more powerful than digital computers for solving various specialized problems. There are well-known quantum algorithms such as Deutsch-Jozsa algorithm [9], Simon's algorithm [10], Shor's quantum factoring algorithm [11], [12], and Grover's database search algorithm [13], [14]. In particular, since the difficulty of the factoring problem is crucial for the security of the RSA cryptosystem [15] which is in widespread use today, interest in quantum computing is increasing [16].

Research on merging evolutionary computation and quantum computing began in the late 1990s. It can be classified into two groups. One group concentrates on generating new quantum algorithms using automatic programming techniques such as genetic programming [17]–[19]. The other concentrates on quantum-inspired evolutionary computing for a digital computer, and is a branch of study on evolutionary computation that is characterized by certain principles of quantum mechanics such as uncertainty, superposition, interference, etc., [3], [20]–[25].

Unlike other research areas, there has been relatively little work done in applying quantum computing to EAs. Quantum-inspired computing was introduced in [26]. In [20], a modified crossover operator which includes the concept of interference was introduced. In [21], a probabilistic representation and a novel population dynamics inspired by quantum computing were proposed. In [22], 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 [23], a QEA-based disk allocation method (QDM) was proposed. According to the results, the average query response times of QDM were equal to or less than those of disk allocation methods using GA (DAGA), and the convergence speed of QDM was 3.2–11.3 times faster than that of DAGA. In [24], QEA was applied to a decision boundary optimization for face verification. The proposed face verification system was tested by face and nonface images extracted from AR face database [27]. Compared with the conventional principal components analysis (PCA) method, improved results were achieved both in terms of the face verification rate and false alarm rate. In [25], some guidelines for setting the parameters of QEA were presented.

This paper proposes research issues on QEA such as a termination criterion, a variation operator H_ϵ gate which is a modified version of the rotation gate, and a two-phase scheme, to improve its performance. A new termination criterion is proposed to give a clearer meaning to how much Q-bit individuals converge to their final states on an average. The proposed H_ϵ gate provides a chance to escape effectively from local optima. As an extended version of QEA, a two-phase QEA (TPQEA) scheme is also proposed by analyzing the effect of changing the initial values of Q-bits. In the first phase, some promising initial values of Q-bits are searched, which will be used in the second phase. By employing the second phase, the performance of QEA can be increased for a class of optimization problems. To demonstrate the effectiveness and applicability of the updated QEA, several experiments are carried out on a class of numerical and combinatorial optimization problems. The results show that the updated QEA becomes more efficient and powerful in terms of convergence speed, fitness, and robustness.

This paper is organized as follows. Section II reviews the previous work on QEA. Section III describes the updated QEA. The new termination criterion, the H_ϵ gate, and TPQEA designed by the analysis of the effect of changing the initial values of Q-bits are presented, respectively. Section IV summarizes the experimental results on a class of numerical and combinatorial optimization problems to demonstrate the performance of the updated QEA. Finally, concluding remarks follow in Section V.

II. Preliminaries

In this section, QEA proposed in [3] is described.

A. 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

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

where $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, 2, \ldots, m$.

Q-bit representation has the advantage that it is able to represent a linear superposition of states probabilistically. The Q-bit representation has a better characteristic of generating diversity in population than any other representations.

B. QEA

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

$$\mathbf{q}_j^t = \left[\begin{array}{c|c|c|c} \alpha_{j1}^t & \alpha_{j2}^t & \cdots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \cdots & \beta_{jm}^t \end{array}\right] \tag{2}$$

where m is the number of Q-bits, i.e., the string length of the Q-bit individual, and $j = 1, 2, \ldots, n$.

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

i) In the step of "initialize $Q(t)$," α_i^0 and β_i^0, $i = 1, 2, \ldots, m$, of all $\mathbf{q}_j^0$, $j = 1, 2, \ldots, n$, are initialized with $1/\sqrt{2}$. It means that one Q-bit individual, $\mathbf{q}_j^0$ represents the linear superposition of all the possible states with the same probability

$$|\psi_{\mathbf{q}_j^0}\rangle = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |X_k\rangle \tag{3}$$

where X_k is the kth state represented by the binary string $(x_1 x_2 \cdots x_m)$, where x_i, $i = 1, 2, \ldots, m$, is either 0 or 1 according to the probability of either $|\alpha_i^0|^2$ or $|\beta_i^0|^2$, respectively. However, it should be noted that the performance of QEA can be influenced by the initial value. The effect of the initial value is discussed in Section III-C.

ii) This step makes binary solutions in $P(0)$ by observing the states of $Q(0)$, where $P(0) = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \ldots, \mathbf{x}_n^0\}$ at generation $t = 0$. One binary solution $\mathbf{x}_j^0$, $j = 1, 2, \ldots, n$, 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 $|\beta_i^0|^2$, $i = 1, 2, \ldots, m$, of $\mathbf{q}_j^0$, respectively. In a quantum computer, in the act of observing a

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 $B(t)$

v) **while** (**not** termination condition) **do**

begin

$t \leftarrow t+1$

vi) make $P(t)$ by observing the states of $Q(t-1)$

vii) evaluate $P(t)$

viii) update $Q(t)$ using Q-gates

ix) store the best solutions among $B(t-1)$ and $P(t)$ into $B(t)$

x) store the best solution $\mathbf{b}$ among $B(t)$

xi) **if** (global migration condition)

then migrate $\mathbf{b}$ to $B(t)$ globally

xii) **else if** (local migration condition)

then migrate $\mathbf{b}_j^t$ in $B(t)$ to $B(t)$ locally

end

end

Fig. 2. Procedure QEA.

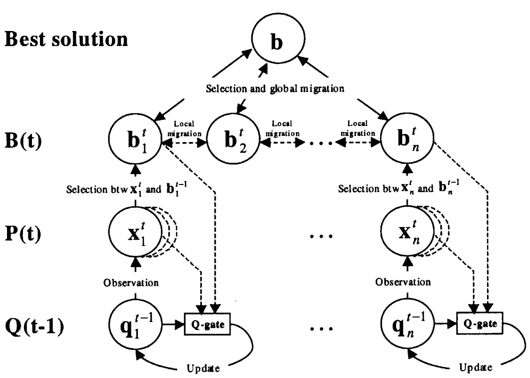


Fig. 3. Overall structure of QEA.

quantum state, it collapses to a single state. However, collapsing into a single state does not occur in QEA, since QEA is working on a digital computer, not on a quantum computer.

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 $B(0) = \{\mathbf{b}_1^0, \mathbf{b}_2^0, \ldots, \mathbf{b}_n^0\}$, and $\mathbf{b}_j^0$ is the same as $\mathbf{x}_j^0$ at the initial generation.

v) Until the termination condition is satisfied, QEA is running in the **while** loop. In particular, termination criteria are described in Section III-A.

vi), vii) 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 $Q(t-1)$. In this case, $\mathbf{x}_j^t$ should be replaced by $\mathbf{x}_{jl}^t$, where l is an observation index.

viii) 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

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \tag{4}$$

where $\Delta\theta_i$, $i = 1, 2, \ldots, m$, 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. $\Delta\theta_i$ can be obtained as a function of the ith bit of the best solution $\mathbf{b}_j^t$, the ith bit of the binary solution $\mathbf{x}_j^t$, and some meaningful conditions. It should be noted that NOT gate, controlled NOT gate, or Hadamard gate can be used as a Q-gate. NOT gate changes the probability of the 1 (or 0) state to that of the 0 (or 1) state. It can be used to escape a local optimum. In controlled NOT gate, one of the two bits should be a control bit. If the control bit is 1, the NOT operation is applied to the other bit. It can be used for the problems which have a large dependency of two bits. Hadamard gate is suitable for the algorithms which use the phase information of Q-bit, as well as the amplitude information, and it should be noted that H_ϵ gate which is a novel Q-gate as a variation operator is designed in Section III-B.

ix), x) 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.

xi), xii) If the global migration condition is satisfied, the best solution $\mathbf{b}$ is migrated to $B(t)$ globally. 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. The migration process can induce a variation of the probabilities of a Q-bit individual.

Definition 1: A *local group* in QEA is defined as the subpopulation affected mutually by a local migration, and its size is the number of individuals in the local group.

III. QEA Issues

The structure of QEA is studied to improve its performance. The following issues are addressed in this section: 1) a new termination criterion which is more efficient; 2) a novel variation operator H_ϵ gate to provide a chance to escape effectively from local optima; 3) the effects of changing the initial values of Q-bits since the initial values can influence the performance

TABLE I
EXPERIMENTAL RESULTS OF THE KNAPSACK PROBLEM WITH 500 ITEMS TO SHOW THE EFFECTS OF CHANGING γ_0 FOR THE TERMINATION CONDITION (6). THE POPULATION SIZE WAS 12, THE GLOBAL MIGRATION PERIOD 100, THE LOCAL GROUP SIZE 3, AND THE NUMBER OF RUNS 30. σ AND t REPRESENT THE STANDARD DEVIATION AND THE AVERAGE GENERATION NUMBER, RESPECTIVELY

γ_0		0.001	0.01	0.1	0.8	0.9
500	best	3031.2	3036.2	3031.3	3031.3	3036.3
	mean	3008.0	3014.3	3019.0	3018.0	3020.3
	worst	2980.7	2991.1	3006.3	3006.3	3001.3
	σ	12.571	9.798	6.548	6.993	7.895
	t	905	1045	1240	1896	3071

TABLE II
RESULTS ON CHANGING γ FOR THE TERMINATION CRITERION (7). THE PARAMETER SETTINGS WERE THE SAME AS IN TABLE I. σ AND t REPRESENT THE STANDARD DEVIATION AND THE AVERAGE GENERATION NUMBER, RESPECTIVELY

γ		0.9	0.95	0.99
500	best	2979.5	3016.2	3031.3
	mean	2949.5	2993.9	3020.1
	worst	2905.9	2960.7	3001.3
	σ	20.372	14.295	7.681
	t	484	722	1164

of QEA (note that in the standard QEA, the initial Q-bit is set to $\left(1/\sqrt{2}, 1/\sqrt{2}\right)$ for the uniform distribution of states 0 and 1); and 4) a two-phase scheme from the analysis of 3).

A. Termination Criteria

To decide the appropriate termination of QEA, a proper termination condition is necessary. Although the maximum number of generations is a generally used termination criterion in EAs, in QEA the probability of the best solution can be employed as a termination criterion because of the probability representation. The termination condition in [3] is designed by using the average probability of the best solution $\mathbf{b}$ as follows:

$$\mathrm{Prob}(\mathbf{b}) = \frac{1}{n}\sum_{j=1}^{n}\left(\prod_{i=1}^{m} p_{ji}\right) \tag{5}$$

with

$$p_{ji} = \begin{cases} |\alpha_{ji}|^2, & \text{if } b_i = 0 \\ |\beta_{ji}|^2, & \text{if } b_i = 1 \end{cases}$$

where b_i is the ith bit of the best solution $\mathbf{b}$ and $(\alpha_{ji}, \beta_{ji})$ the ith Q-bit of the jth Q-bit individual. The termination condition is defined as

$$\mathrm{Prob}(\mathbf{b}) > \gamma_0 \tag{6}$$

where $0 < \gamma_0 < 1$. The probability $\mathrm{Prob}(\mathbf{b})$ represents the average convergence of all Q-bit individuals to the best solution. It must be a substantial termination criterion of QEA. However, since the probability is sensitive to each Q-bit's probability, it is not easy to set the value γ_0. The slight difference of γ_0 can increase the processing time for a particular problem. Table I shows the effects of changing the value γ_0. From the table, if $\gamma_0 \geq 0.1$, all the results were almost the same. However, the generation number at $\gamma_0 = 0.9$ was about 2.5 times of that at $\gamma_0 = 0.1$.

To design a new termination criterion regardless of the sensitivity, a measure of the convergence of Q-bit individual is defined.

Definition 2: Q-bit convergence C_b is defined to be the convergence measure of a Q-bit individual in QEA as

$$C_b(\mathbf{q}) = \frac{1}{m}\sum_{i=1}^{m}\left|1 - 2|\alpha_i|^2\right|$$

or

$$C_b(\mathbf{q}) = \frac{1}{m}\sum_{i=1}^{m}\left|1 - 2|\beta_i|^2\right|$$

where $\mathbf{q}$ is a Q-bit individual, and (α_i, β_i) is its ith Q-bit.

Using the Q-bit convergence, the following termination criterion can be designed

$$C_{\mathrm{av}} = \left(\frac{1}{n}\sum_{j=1}^{n} C_b(\mathbf{q}_j)\right) > \gamma \tag{7}$$

where $C_b(\mathbf{q}_j)$ is the Q-bit convergence of the jth Q-bit individual. The termination criterion of (7) is shown to be regardless of the best solution $\mathbf{b}$. However, the average convergence of all Q-bit individuals can represent the processing status of QEA properly, and it gives a clearer meaning on how much each Q-bit converges to 0 or 1 on an average. Consequently, users can more systematically set the termination condition. For example, if C_{av} is 0.99, it means that 99% of the Q-bits converge to the true value (0 or 1) on an average. Table II shows the results on changing the value γ.

Fig. 4(a) and (b) shows the difference between the two measures of (6) and (7) for termination criteria. In Fig. 4(b), it should be noted that the Q-bit convergence provides an easier understanding of the Q-bit individuals' convergence.

It is worthwhile to mention that if a faster termination is needed, the following maximum Q-bit convergence $C_{\max}$ can be used

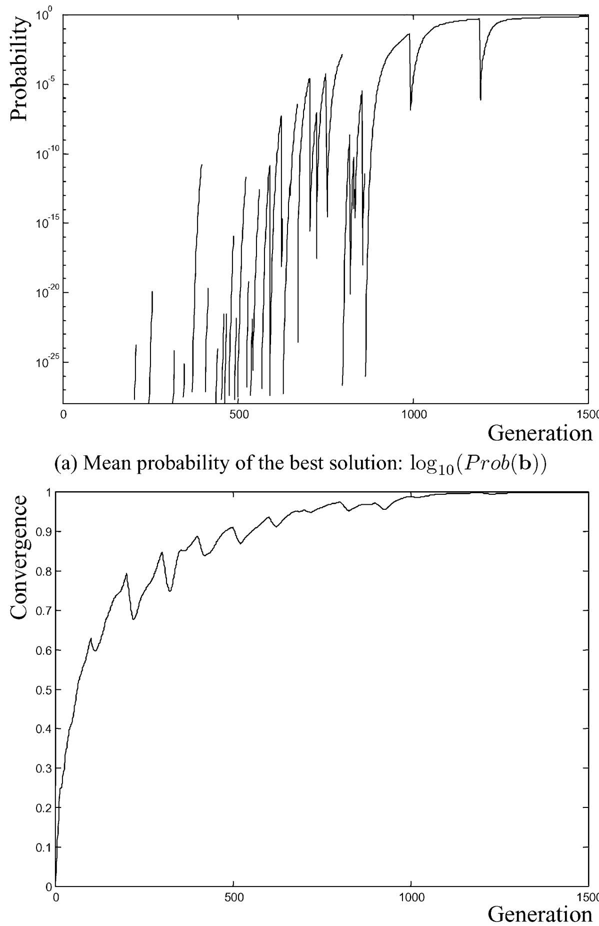
$$C_{\max} = \left(\max_{j=1}^{n} C_b(\mathbf{q}_j)\right) > \gamma. \tag{8}$$

B. H_ϵ Gate

The rotation gate used as a Q-gate induces the convergence of each Q-bit to either 0 or 1. However, a Q-bit converged to either 0 or 1 cannot escape the state by itself, although it can be changed passively by a global or local migration. If the value of $|\beta|^2$ is 0 (or 1), the observing state of the Q-bit is always 0 (or 1). To prevent the premature convergence of Q-bit, H_ϵ gate is defined as a Q-gate.

Definition 3: An *H_ϵ gate* is defined as a Q-gate extended from the rotation gate

$$[\alpha_i' \; \beta_i']^T = H_\epsilon(\alpha_i,\, \beta_i,\, \Delta\theta_i) \tag{9}$$



(a) Mean probability of the best solution: $\log_{10}(Prob(\mathbf{b}))$

(b) Mean Q-bit convergence: C_{av}

Fig. 4. Difference between the two measures of (6) and (7) for termination criteria. A logarithmic (base 10) scale is used for the vertical axis of (a).

where for $[\alpha_i'' \; \beta_i'']^T = R(\Delta\theta_i)[\alpha_i \; \beta_i]^T$:

i) if $|\alpha_i''|^2 \leq \epsilon$ and $|\beta_i''|^2 \geq 1 - \epsilon$

$$[\alpha_i' \; \beta_i']^T = [\sqrt{\epsilon} \; \sqrt{1-\epsilon}]^T;$$

ii) if $|\alpha_i''|^2 \geq 1 - \epsilon$ and $|\beta_i''|^2 \leq \epsilon$

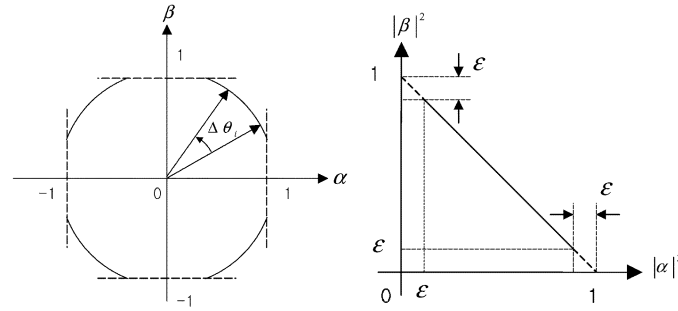
$$[\alpha_i' \; \beta_i']^T = [\sqrt{1-\epsilon} \; \sqrt{\epsilon}]^T;$$

iii) otherwise

$$[\alpha_i' \; \beta_i']^T = [\alpha_i'' \; \beta_i'']^T$$

where $0 < \epsilon \ll 1$, $R(\Delta\theta_i)$ is the rotation gate, and $\Delta\theta_i$, $i = 1, 2, \ldots, m$, is the rotation angle of each Q-bit toward either 0 or 1 state, depending on its sign.

Fig. 5 shows the H_ϵ gate, where $\lim_{\epsilon \to 0} H_\epsilon(\cdot)$ is the same as the rotation gate. While the rotation gate makes the probability of $|\alpha|^2$ or $|\beta|^2$ converge to either 0 or 1, H_ϵ gate makes it converge to ϵ or $(1-\epsilon)$. It should be noted that if ϵ is too big, the convergence tendency of a Q-bit individual may disappear.



(a) H_ϵ gate

(b) ϵ constraints

Fig. 5. H_ϵ gate based on rotation gate.

Definition 4: A *Hamming distance* H of the two binary strings, $\mathbf{x}_1$ and $\mathbf{x}_2$, is defined as the number of their bit-wise-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.

Theorem 1: The entropy of the probability distribution for the search space represented by Q-bit individual with H_ϵ gate converges to

$$-\sum_{h=0}^{m} \left(\frac{m!}{h!(m-h)!} \epsilon^{m-h} (1-\epsilon)^h \left(\log_2 \left(\epsilon^{m-h} (1-\epsilon)^h \right) \right) \right) \quad (10)$$

where m is the length of Q-bit individual.

Proof: Let $\mathbf{x}$ be the converged binary solution and $\mathbf{x}^h$ the binary solution with Hamming distance h from $\mathbf{x}$. Each Q-bit converges to either $(\sqrt{\epsilon}, \sqrt{1-\epsilon})$ or $(\sqrt{1-\epsilon}, \sqrt{\epsilon})$ in any case. For clarification purpose, let us consider one simple case, where all Q-bits corresponding to $\mathbf{x}$ converge to $(\alpha, \beta) = (\sqrt{\epsilon}, \sqrt{1-\epsilon})$. Then, the probability of $\mathbf{x}^h$ is described as

$$p(\mathbf{x}^h) = \epsilon^{m-h}(1-\epsilon)^h$$

and the number of all possible $\mathbf{x}^h$ is

$$n(\mathbf{x}^h) = \frac{m!}{h!(m-h)!}.$$

Therefore, the entropy [28] of the probability distribution for the search space represented by the Q-bit individual with H_ϵ gate is obtained as

$$\begin{aligned} & I(p(\mathbf{x}) | \mathbf{x} \in \mathbb{X}) \\ &= -\sum_{\mathbf{x} \in \mathbb{X}} p(\mathbf{x}) \log_2 p(\mathbf{x}) \\ &= -\sum_{h=0}^{m} \left(n(\mathbf{x}^h) p(\mathbf{x}^h) \log_2 p(\mathbf{x}^h) \right) \\ &= -\sum_{h=0}^{m} \left(\frac{m!}{h!(m-h)!} \epsilon^{m-h} (1-\epsilon)^h \left(\log_2 \left(\epsilon^{m-h} (1-\epsilon)^h \right) \right) \right). \end{aligned}$$

ONEMAX Problem: Maximize

$$\text{ONEMAX}(\mathbf{x}) = \sum_{i=1}^{m} x_i \quad (11)$$

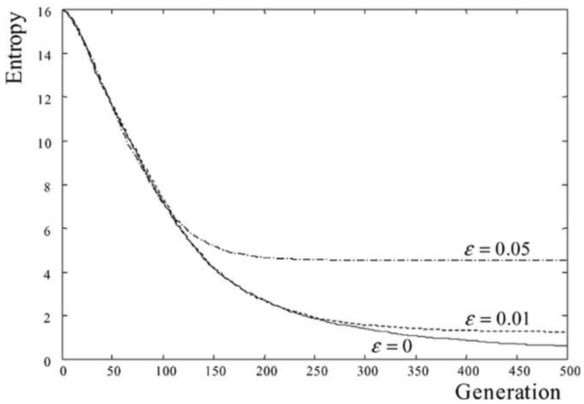


Fig. 6. Comparison of the entropy of the probability distribution for the search space among QEA1s with H_ϵ gates for ϵ =0, 0.01 and 0.05, respectively. All the results were averaged over 30 runs for the OneMax problem for length m, where $m = 16$.

where x_i is the ith bit of $\mathbf{x}$, m is the length of $\mathbf{x}$, and the global maximum value is m at $\mathbf{x} = 111\cdots1$.

Fig. 6 shows the differences of the entropy of the probability distribution for the search space among QEA1s with H_ϵ gates for $\epsilon = 0$, 0.01, and 0.05, respectively. From the results for the OneMax problem of (11), it should be noted that the entropy converges to the larger value as ϵ is bigger. It means that if ϵ is too big, the mechanism for exploitation may not work.

If H_ϵ gate is used as a Q-gate, the termination conditions of (7) and (8) should be modified as

$$C_{\text{av}} = \left(\frac{1}{n}\sum_{j=1}^{n} C_b(\mathbf{q}_j)\right) > (1-2\epsilon)\gamma \tag{12}$$

and

$$C_{\max} = \left(\max_{j=1}^{n} C_b(\mathbf{q}_j)\right) > (1-2\epsilon)\gamma \tag{13}$$

respectively. To increase the period for fine tuning caused by the ϵ boundary, the mixed termination condition can be also used as follows:

$$\text{MAXGEN} = \tau t_\gamma \tag{14}$$

where t_γ is the number of generations when the termination condition with γ of (12) or (13) is satisfied and $\tau > 1$.

To investigate the performance of H_ϵ gate, Schwefel function (see Appendix B) is considered. Table III shows the effects of changing ϵ for the H_ϵ gate. As shown in the table, the results for $\epsilon = 0.01$ were the best for the Schwefel function, although the average generation was larger than other results. It should be noted that if ϵ is too big, the performance would be worse than that of QEA with the rotation gate ($\epsilon = 0$). While a large ϵ ($= 0.03$) induces a fast premature convergence, a properly selected-small value of ϵ ($= 0.01$) provides better solutions. In particular, H_ϵ gate is recommended for a class of numerical optimization problems which have many local optima.

C. *Effects of Changing the Initial Values of* Q*-bits*

In the "initialize $Q(t)$" step of Fig. 2, all Q-bits are initialized with $(1/\sqrt{2}, 1/\sqrt{2})$ to represent a linear superposition of all possible solutions with the same probability [3]. It means that we have no information about the search space. Here, let us assume that we have a little bit of information about the search space to be explored. Then, we can see that the prior knowledge can be easily put into the initial values of Q-bits.

TABLE III
Experimental Results of the Schwefel Function to Show the Effects of Changing ϵ for H_ϵ Gate. The Population Size Was 15, the Global Migration Period 100, the Local Group Size 3, the Number of Observations 3, and the Number of Runs 30. γ of (12) Was Set to 0.9999. $b.$, $m.$, and $w.$ Mean *Best*, *Mean*, and *Worst*, Respectively. σ and t Rrepresent the Standard Deviation and the Average Generation Number, Respectively

ϵ		0	0.005	0.01	0.015	0.02	0.03
$f_{Schwefel}$	b.	1677.9	3.8×10^{-4}	3.8×10^{-4}	2.6×10^{-3}	0.2841	500.90
	m.	2467.5	31.583	4.2×10^{-4}	7.9145	67.980	1046.6
	w.	3624.3	236.87	7.5×10^{-4}	118.45	475.05	1763.1
	σ	503.83	60.649	7.2×10^{-5}	29.541	104.58	343.73
	t	4567.5	7183.9	8817.2	7364.0	6207.0	2903.2

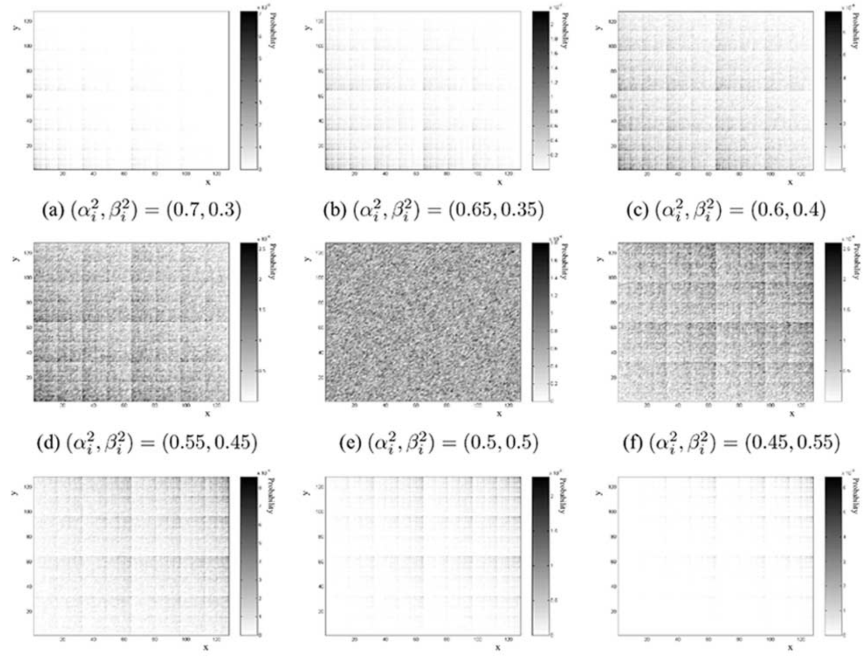
TABLE IV
Experimental Results of the Knapsack Problem With 500 Items to Show the Effects of Changing the Initial Values of Q-Bits. The Population Size Was 15, the Global Migration Period 100, the Local Group Size 3, and the Number of Runs 30. The Termination Condition of (7) Was Used With $\gamma = 0.99$. $b.$, $m.$, and $w.$ Mean *Best*, *Mean*, and *Worst*, Respectively. σ and t Represent the Standard Deviation and the Average Generation Number, Respectively

(α_i^2, β_i^2)		(0.99, 0.01)	(0.5, 0.5)	(0.01, 0.99)
500	b.	94.993	94.856	94.892
	m.	86.973	81.978	80.969
	w.	79.942	69.983	69.996
	σ	3.774	5.846	5.662
	t	697	4682	8231

For instance, let us consider the knapsack problem (see Appendix A) with 500 items, which does not have an average knapsack capacity as a constraint, but a restrictive knapsack capacity of $C = 2v$, where $v = 10$. In this case, the optimal solution contains very few items. An infeasible search space, where the constraint is not satisfied, occupies almost the whole search space. For this type of knapsack problem, we already have some prior knowledge like "the optimal solution contains very few items." From this prior knowledge, we can set the initial value of each Q-bit to $\left(\sqrt{1-\beta_i^2}, \beta_i\right)$, where β_i is a small value close to zero.

Table IV shows the experimental results of the knapsack problem with 500 items using the knapsack capacity to be restricted to $C = 20$. The table shows that the results are highly dependent on the initial values of Q-bits. The results of (0.99,0.01) are the best and also its average generation number is the smallest one. More specifically, the convergence speed of (0.99,0.01) is 6.7 and 11.8 times faster than those of (0.5, 0.5) and (0.01, 0.99), respectively. In addition to that, the average standard deviation of (0.99, 0.01) over 30 runs is the best one.

The results of Table IV agree with our prediction through the prior knowledge. It can be explained by the relation between the initial search space and the optimal solution. If the initial



(a) $(\alpha_i^2, \beta_i^2) = (0.7, 0.3)$ (b) $(\alpha_i^2, \beta_i^2) = (0.65, 0.35)$ (c) $(\alpha_i^2, \beta_i^2) = (0.6, 0.4)$

(d) $(\alpha_i^2, \beta_i^2) = (0.55, 0.45)$ (e) $(\alpha_i^2, \beta_i^2) = (0.5, 0.5)$ (f) $(\alpha_i^2, \beta_i^2) = (0.45, 0.55)$

(g) $(\alpha_i^2, \beta_i^2) = (0.4, 0.6)$ (h) $(\alpha_i^2, \beta_i^2) = (0.35, 0.65)$ (i) $(\alpha_i^2, \beta_i^2) = (0.3, 0.7)$

Fig. 7. Differences in the initial search spaces with respect to the initial values of Q-bits. The size of the search space is 2^{14} (7 bits for each x and y). On the (x, y) plane, the darker points have a higher probability to be present in the initial search space. The probabilities were obtained by observing each Q-bit individual 10^5 times.

search space is formed near the optimal solution, the solution can be searched in a short span of time. To measure the distance between the initial search space and the optimal solution, ones-number distance, defined below, can be considered.

Definition 5: A *ones-number distance*, D_n, of the two binary strings, $\mathbf{x}_1$ and $\mathbf{x}_2$, is defined as the difference between their numbers of ones, which is defined as

$$D_n(\mathbf{x}_1, \mathbf{x}_2) = |\text{ones}(\mathbf{x}_1) - \text{ones}(\mathbf{x}_2)|$$

where the function $\text{ones}(\mathbf{x})$ returns the number of ones in the binary string $\mathbf{x}$.

The distribution of the ones-number distance between the initial search space and the optimal solution can be obtained, since the initial search space determined by each value of Q-bit has a distribution with respect to ones number.

Fig. 7 shows the differences of the initial search spaces with respect to the initial values of Q-bits. In the case of Fig. 7(e), the distribution of the initial search space is nearly a random noise. It means that the initial search starts randomly. In the cases of Fig. 7(a)–(d), the points which include less ones have higher probabilities. On the other hand, in the cases of Fig. 7(f)–(i), the points which include more ones have higher probabilities. It means that the initial search space can be formed effectively by changing the initial values of Q-bits. It is worthwhile to mention that the initial search space is distributed globally, although the distribution spreads to the space including more (or less) ones depending on Q-bit values. For instance, let us consider 7-bit strings with 1 for the number of ones. There are seven strings of which number of ones is 1. Their integer numbers are 1, 2, 4, 8, 16, 32, and 64, respectively. It means that the solutions which have the same number of ones spread widely in the search space.

Procedure TPQEA
begin
    First-phase QEA
    Second-phase QEA
end

Fig. 8. Procedure TPQEA.

The reason why the initial search space represented by the initial values of Q-bits is distributed globally can be explained by this characteristics of binary coding.

D. Two-Phase Scheme

We have already verified that changing the initial values of Q-bits can provide better performance of QEA. The initial values of Q-bits are directly connected to the initial search space as shown in Fig. 7. If the initial values of Q-bits can be found to represent the initial search space with small distance to the best solution, the Q-bit individuals can converge to the best solution effectively. To put this idea to the algorithm, TPQEA is proposed in the following.

TPQEA has two procedures as shown in Fig. 8. In the first phase, as shown in Fig. 9, a promising initial value is searched and stored into $[\alpha^b\ \beta^b]^T$. The first-phase is similar to the standard procedure of QEA except the following.

1) The "initialize $Q(t)$" step is different. In this step, each local group has a different value of Q-bit from other local

Procedure first-phase QEA (phase I)
begin
    $t \leftarrow 0$
    initialize $Q(t)$
    make $P(t)$ by observing the states of $Q(t)$
    evaluate $P(t)$
    store the best solutions among $P(t)$ into $B(t)$
    while (**not** termination condition) **do**
    begin
        $t \leftarrow t+1$
        make $P(t)$ by observing the states of $Q(t-1)$
        evaluate $P(t)$
        update $Q(t)$ using Q-gates
        store the best solutions among $B(t-1)$ and $P(t)$ into $B(t)$
        if (local migration condition)
        then migrate b_j^t to $B(t)$ locally
    end
    store the initial value of Q-bit inducing the best result into $[\alpha^b\ \beta^b]^T$
end

Fig. 9. First-phase QEA.

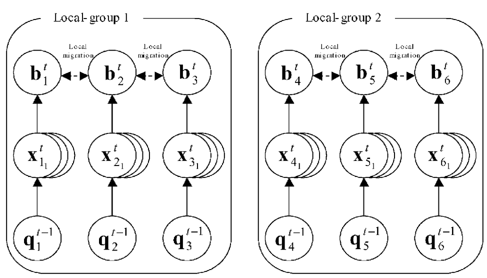


Fig. 10. Relations between variables for the first-phase QEA, when the population size, the local group size, and the number of observations are 6, 3, and 3, respectively.

groups to explore a different search space each. In the gth local group, the initial value of Q-bit can be assigned as

$$\begin{bmatrix} \alpha_g \\ \beta_g \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{(1-2\delta)}{N_g-1}g+\delta} \\ \sqrt{1-\frac{(1-2\delta)}{N_g-1}g+\delta} \end{bmatrix} \tag{15}$$

where N_g is the number of local groups, and δ, $0<\delta\ll 1$, is the minimum probability of the state 1 (or 0). Equation (15) assigns a probability of each group dividing an interval $[\delta, 1-\delta]$ into N_g equal parts.

2) To guarantee the homogeneity of each group, the best solution $\mathbf{b}$ is not used and the global migration process is removed from the standard structure of QEA. For example, the relations between variables are shown in Fig. 10 when the population size, the local group size, and the number of observations are 6, 3, and 3, respectively.
3) The condition of (7) [or (12)] can be used as a termination condition for the first phase. However, if faster transition from the first phase to the second phase is required, the termination condition of (8) [or (13)] can be used for the first phase.
4) At the end of the first phase, a process is added to store the initial value of Q-bit inducing the best result into $[\alpha^b\ \beta^b]^T$.

Procedure second-phase QEA (phase II)
begin
    initialize $Q(t)$ using $[\alpha^b\ \beta^b]^T$
    make $P(t)$ by observing the states of $Q(t)$
    evaluate $P(t)$
    store the best solutions among $P(t)$ into $B(t)$
    while (**not** termination-condition) **do**
    begin
        $t \leftarrow t+1$
        make $P(t)$ by observing the states of $Q(t-1)$
        evaluate $P(t)$
        update $Q(t)$ using Q-gates
        store the best solutions among $B(t-1)$ and $P(t)$ into $B(t)$
        store the best solution b among $B(t)$
        if (global migration condition)
        then migrate b to $B(t)$ globally
        else if (local migration condition)
        then migrate b_j^t in $B(t)$ to $B(t)$ locally
    end
end

Fig. 11. Second-phase QEA.

Fig. 11 depicts the second phase of TPQEA. It is almost the same as the procedure QEA, except that time initialization "$t \leftarrow 0$" is removed and the "initialize $Q(t)$" step shall use the initial value of Q-bit, i.e., $[\alpha^b\ \beta^b]^T$, obtained from the first phase.

Let us consider the concatenated 5-bit traps as

$$f_{\mathrm{trap}}(\mathbf{x}) = \sum_{i=0}^{N_{\mathrm{trap}}-1} \mathrm{trap}\,(x_{5i+1}, x_{5i+2}, x_{5i+3}, x_{5i+4}, x_{5i+5})$$

where N_{trap} is the number of traps and

$$\mathrm{trap}(\mathbf{x}) = \begin{cases} 4-\ \mathrm{ones}(\mathbf{x}), & \text{if ones}(\mathbf{x}) \le 4 \\ 5, & \text{if ones}(\mathbf{x}) = 5 \end{cases}.$$

To maximize f_{trap}, the individuals should be able to escape from the 5-bit traps, at (0, 0, 0, 0, 0), as shown in Fig. 12. Table V shows the results of QEA and TPQEA for the concatenated 5-bit traps with $N_{\mathrm{trap}} = 20$. The global maximum value of f_{trap} is 100 when all the 100 bits are ones. While QEA fell into the trap point (00000) in 15 traps on an average, TPQEA did not fall into the traps at all. In particular, the average generation number of TPQEA was shorter than QEAs, although TPQEA has two phases. However, it should be noted that TPQEA may need a larger number of generations to find the best solution for a particular problem as compared with QEAs, if the best solution is included in the search space with ones-number distance

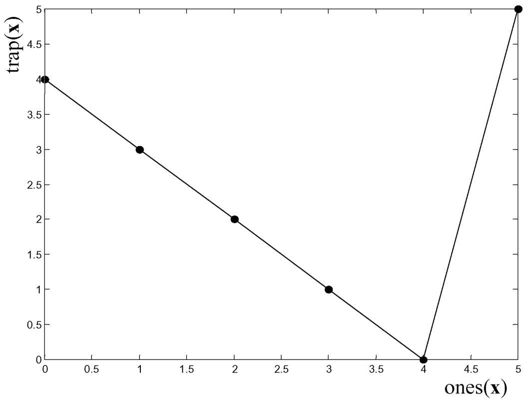


Fig. 12. Five-bit trap.

TABLE V
EXPERIMENTAL RESULTS OF THE CONCATENATED 5-BIT TRAPS WITH $N_{\mathrm{trap}} = 20$. THE POPULATION SIZE WAS 15, THE GLOBAL MIGRATION PERIOD 100, THE LOCAL GROUP SIZE 3, AND THE NUMBER OF RUNS 30. THE TERMINATION CONDITION OF (12) WAS USED WITH $\gamma = 0.99$. γ AND δ FOR THE FIRST PHASE OF TPQEA WERE 0.9 AND 0.05, RESPECTIVELY. ϵ OF THE H_ϵ GATE WAS 0.01. σ AND t REPRESENT THE STANDARD DEVIATION AND THE AVERAGE GENERATION NUMBER, RESPECTIVELY

		QEA	TPQEA
f_{trap}	best	90	100
	mean	85.033	100
	worst	81	100
	σ	1.722	0
	t	218	84

0 from the initial space defined with the initial value of Q-bit, $[1/\sqrt{2}\ 1/\sqrt{2}]^T$.

IV. EXPERIMENTAL RESULTS

In this section, six numerical optimization problems and one combinatorial optimization problem are discussed to demonstrate the effectiveness of H_ϵ gate and the applicability of TPQEA, respectively.

A. Numerical Optimization

QEA was proposed for a class of combinatorial optimization in [3], and no work for a class of numerical optimization has been done till date. The reason may be that the representation of real number is more suitable for numerical optimization than that of binary string. Here, the six numerical optimization functions of Sphere, Ackley, Griewank, Rastrigin, Schwefel, and Rosenbrock (see Appendix B) were considered to demonstrate the effectiveness of QEA with the H_ϵ gate to a class of numerical optimization.

To minimize the six functions, QEAs were tested using the parameter settings as given in Table VI. In particular, to verify the effectiveness of H_ϵ gate, the global migration was not used. The effectiveness of the new termination criterion was already discussed in Sections III-A–III-B. Considering the resolutions of variables, the numbers of the Q-bits for the six functions were set to 18, 18, 21, 17, 22, and 18 bits (per variable), respectively. Gray coding was used to convert from binary string to real value.

TABLE VI
PARAMETER SETTINGS OF QEAS FOR THE EXPERIMENTS ON THE NUMERICAL OPTIMIZATION FUNCTIONS (16)–(21)

	QEA with H_ϵ gate	QEA with Rotation gate
Population size	100	100
Number of observations	1	1
Local group size	100	100
ϵ for H_ϵ gate	0.01	(0)
Termination condition	MAXGEN	MAXGEN

TABLE VII
EXPERIMENTAL RESULTS OF THE SIX NUMERICAL OPTIMIZATION FUNCTIONS (16)–(21). THE RESULTS OF EP WERE REFERRED FROM [29]. THE NUMBER OF RUNS WAS 50. THE PARENTHESIZED VALUES ARE THE POPULATION SIZES. $m.$, σ, AND t REPRESENT THE MEAN BEST, THE STANDARD DEVIATION, AND THE MAXIMUM NUMBER OF GENERATIONS, RESPECTIVELY

		QEAs		EP	
		QEA w/ H_ϵ (100)	QEA w/ R (100)	FEP (100)	CEP (100)
f_{Sphere}	m.	1.8×10^{-4}	4.3×10^{-6}	5.7×10^{-4}	2.2×10^{-4}
$t = 1500$	σ	1.3×10^{-4}	0	1.3×10^{-4}	5.9×10^{-4}
f_{Ackley}	m.	2.5×10^{-3}	4.8×10^{-4}	1.8×10^{-2}	9.2
$t = 1500$	σ	8.1×10^{-4}	0	2.1×10^{-3}	2.8
$f_{Griewank}$	m.	3.6×10^{-2}	5.8×10^{-2}	1.6×10^{-2}	8.6×10^{-2}
$t = 2000$	σ	3.2×10^{-2}	7.5×10^{-2}	2.2×10^{-2}	0.12
$f_{Rastrigin}$	m.	3.9×10^{-2}	18.7	4.6×10^{-2}	89.0
$t = 5000$	σ	1.9×10^{-1}	7.4	1.2×10^{-2}	23.1
$f_{Schwefel}$	m.	3.8×10^{-4}	216.04	14.987	4652.3
$t = 9000$	σ	3.0×10^{-9}	163.8	52.6	634.5
$f_{Rosenbrock}$	m.	11.73	7.18	5.06	6.17
$t = 20000$	σ	18.36	6.77	5.87	13.61

For the comparison purpose, the termination condition with the maximum number of generations was used, since the experiments referred from [29] were tested with a fixed number of generations. The rotation angles of Θ were set to $[0\,0\,p\,0\,n\,0\,0\,0]^T$ as in [3], where p and $|n|$ (absolute value of n) were set to 0.06π, 0.06π, 0.06π, 0.04π, 0.04π, and 0.04π for the six functions, respectively.

The results taken from the reference [29] were tested using classical evolutionary programming (CEP) and fast EP (FEP). The population size was selected to be 100 for each experiment. According to the reference, FEP provided better solutions than CEP.

Table VII shows the results of the numerical functions (16)–(21) for QEA with the H_ϵ gate, QEA with the rotation gate [3], FEP, and CEP. In the cases of f_{Sphere} and f_{Ackley} which are relatively simple functions compared with the other functions, QEAs and EP had almost the same results, although the results of QEA with the rotation gate were slightly better than the others'. However, in the cases of f_{Griewank}, $f_{\mathrm{Rastrigin}}$, and f_{Schwefel} which have many local optima, QEA with the H_ϵ gate and FEP had better results than the others. In particular,

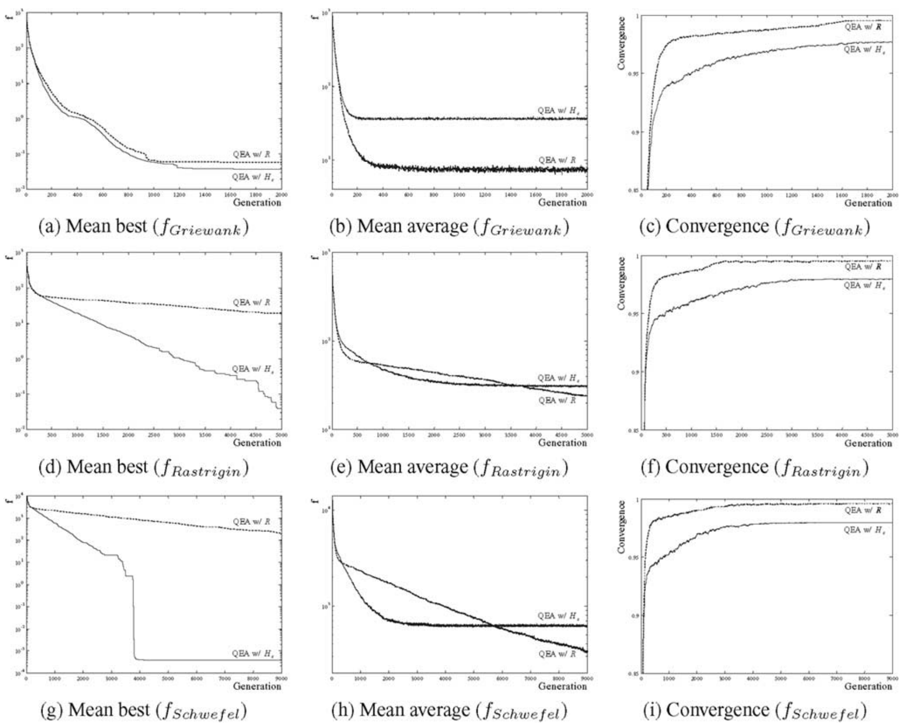


Fig. 13. Comparison between QEA with the H_ϵ gate and QEA with the rotation gate on Griewank, Rastrigin, and Schwefel functions. The parameter values were set to the same as shown in Table VI. The vertical axis for (a), (b), (d), (e), (g), and (h) shows the function value, the vertical axis for (c), (f), and (i) shows the value of $C_{\rm av}$, and the horizontal axis represents the number of generations. (a), (d), and (g) show the best results, (b), (e), and (h) the average results, and (c), (f), and (i) the average Q-bit convergence. All results were averaged over 50 runs.

in the case of $f_{\rm Schwefel}$, only QEA with the H_ϵ gate found the global solution. In the case of $f_{\rm Rosenbrock}$, no algorithm found the global solution on an average. The reason why QEAs could not find the global solution of $f_{\rm Rosenbrock}$ is described in Appendix C.

Fig. 13 shows the comparison between QEA with the H_ϵ gate and QEA with the rotation gate on Griewank, Rastrigin, and Schwefel functions. It should be noted that the best results of QEA with the H_ϵ gate were significantly better compared with those of QEA with the rotation gate, although the average results of QEA with the H_ϵ gate were somewhat worse compared with those of QEA with the rotation gate. The results of Griewank show, in particular, that the best results of QEA with the H_ϵ gate were better, although the average results of QEA with the H_ϵ gate were worse. The reason behind this type of result is the H_ϵ gate, that prevents each Q-bit converging to the final state (0 or 1). It is worthwhile to mention that the results of the average Q-bit convergence show that the final values for QEAs with the H_ϵ and rotation gate converge to $(1-2\epsilon)$ and 1, respectively. Since the converged entropy of QEA with the H_ϵ gate is a nonzero value of (10), the converged Q-bit individual still includes various binary solutions. It means that the solutions selected as a population are obtained from a little wider search space area. For this reason, the average results of QEA with H_ϵ gate did not converge below a certain value.

B. Combinatorial Optimization

The knapsack problem with the restrictive knapsack capacity (see Appendix A) was considered for a class of combinatorial optimization problem to demonstrate the applicability of TPQEA. Comparison was made with the experimental results of QEA. The random repair method used in [3] was considered for handling the constraint of the knapsack capacity to compare their performance, although the greedy repair method guaranteed better solutions for the restrictive knapsack problem. Table VIII shows the parameter settings of TPQEA and QEA. The parameters of TPQEA were set to the same set of values as those of QEA except the additional parameters for the first phase of TPQEA. The rotation angle of p (or $|n|$) for Q-gate was set to 0.01π.

TABLE VIII
PARAMETER SETTINGS OF QEA AND TPQEA FOR THE EXPERIMENTS ON THE RESTRICTIVE KNAPSACK PROBLEM. "-" MEANS THAT THE PARAMETER IS NOT NEEDED

	QEA	TPQEA	
		Phase I	Phase II
Population size	15	15	
Number of observations	1	1	
Local group size	3	3	
Global migration period	100	-	100
Termination condition: γ	0.99	0.99	0.99
Equation of the termination condition	(7)	(8)	(7)
Minimum probability of (15): δ	-	0.01	-

Table IX shows the experimental results of the knapsack problems with 100, 250, and 500 items. As the table shows, TPQEA yielded much better results compared with QEA. Moreover, the average elapsed generation number of TPQEA is smaller than that of QEA. More specifically, the convergence

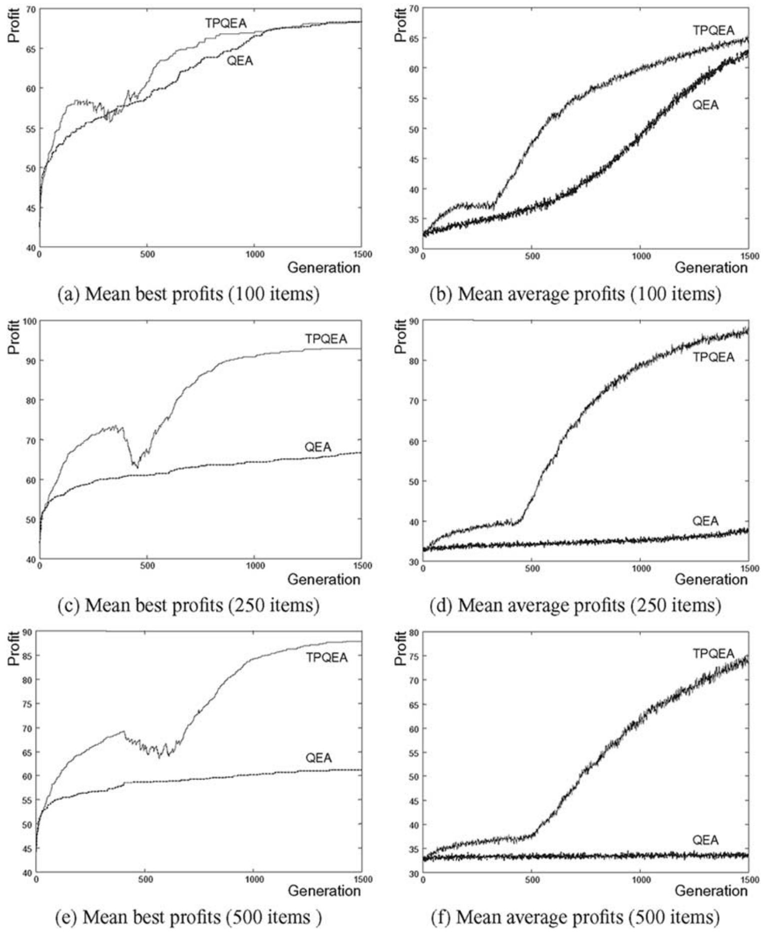


Fig. 14. Comparison of TPQEA and QEA on the knapsack problem. The parameter values were set to the same as shown in Table VIII. However, regardless of the termination criteria, the results from 1 to 1500 generations were plotted. The vertical axis shows the profit value of knapsack and the horizontal axis represents the number of generations. The best profit values and the average profit values were averaged over 30 runs.

TABLE IX
EXPERIMENTAL RESULTS OF THE KNAPSACK PROBLEM WITH THE RESTRICTIVE KNAPSACK CAPACITY, $C = 20$, FOR 100, 250, AND 500 ITEMS, RESPECTIVELY. THE PARAMETER VALUES WERE SET TO THE SAME AS SHOWN IN TABLE VIII, THE NUMBER OF RUNS WAS SELECTED TO BE 30. σ AND t REPRESENT THE STANDARD DEVIATION AND THE AVERAGE GENERATION NUMBER FOR TERMINATION, RESPECTIVELY

		QEA	TPQEA
100	best	69.998	69.999
	mean	67.819	68.467
	worst	59.995	64.969
	σ	3.774	2.279
	t	1463	804
250	best	94.998	94.997
	mean	87.484	90.122
	worst	74.991	84.674
	σ	4.604	3.551
	t	2680	929
500	best	89.998	94.968
	mean	81.788	85.309
	worst	69.983	74.983
	σ	5.082	4.998
	t	4624	1165

speed of TPQEA is 1.8, 2.9, and 4.0 times faster than that of QEA for 100, 250, and 500 items, respectively. The results of the standard deviations show that TPQEA is more robust than QEA for finding solutions.

Fig. 14 shows clearly that TPQEA performs significantly better than QEA in terms of convergence speed and the amount of profit. The transition point from the first phase to the second phase can be found out easily. After the transition, the rising slope of TPQEA is steeper than that of QEA. In particular, the tendency of convergence rate is shown clearly in the results of the mean of average profits of population. After the transition, all the population converges to better solutions at a faster rate.

V. CONCLUSION

This paper proposed research issues on QEA such as a termination criterion, a Q-gate, and a two-phase scheme, for a class of numerical and combinatorial optimization problems.

A new termination criterion was proposed by using the definition of Q-bit convergence. It gives a clearer meaning to how much each Q-bit individual converges to its final state on an average. According to applications, users can choose one between the average and maximum Q-bit convergence as a measure of termination condition.

H_ϵ gate was proposed as a variation operator to escape effectively from local optima. In particular, the experimental results

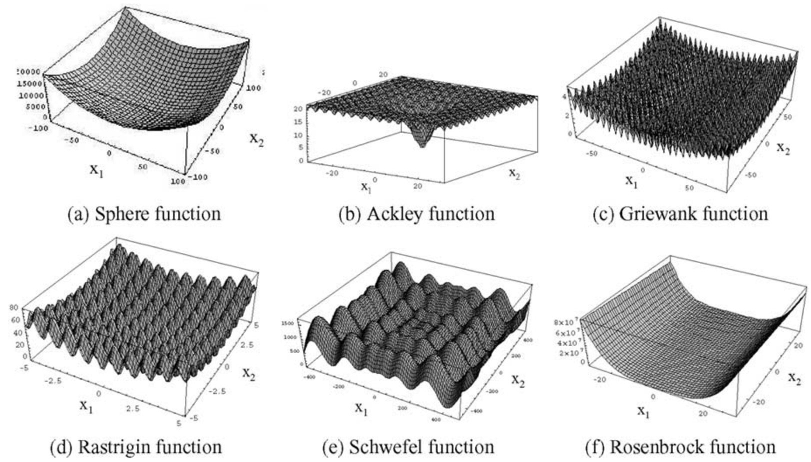


Fig. 15. Numerical optimization functions (16)–(21). Each of them has 30 variables. In these figures, however, only two variables were considered to plot their shapes approximately.

on the six numerical functions showed that QEA with the H_ϵ gate performed much better than QEA with the rotation gate for a class of numerical optimization problems which have many local optima.

TPQEA was designed by using the analysis of changing the initial values of Q-bits. A promising initial value is searched and stored in the first-phase, and the global solution is searched by using the initial value in the second phase. The experimental results on the problem of the concatenated 5-bit traps, which is a well-known trap for EAs, showed that TPQEA performed better than QEA when the global optimum was included in the search space with larger ones-number distance from randomly generated search space. In particular, the experimental results on the restrictive knapsack problem showed that TPQEA performed much better than QEA in terms of convergence speed, the amount of profit, and robustness.

The study of research issues on QEA will be useful in designing QEA to improve its performance for a class of numerical and combinatorial optimization problems.

Appendix A
Knapsack Problem

Knapsack problem which is a well-known combinatorial optimization problem is included in a class of NP-hard problems [30]. The knapsack problem can be described as selecting a subset of items from among various items so that it is 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 the condition

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

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 ith item is selected for the knapsack.

In this paper, strongly correlated sets of data were considered as

$$w_i = \text{uniformly random}\,[1, v]$$
$$p_i = w_i + r$$

where $v = 10$ and $r = 5$.

There are two types of knapsack capacity considered in this paper [31].

1) Average knapsack capacity is

$$C = \frac{1}{2}\sum_{i=1}^{m} w_i.$$

2) Restrictive knapsack capacity is

$$C = 2v.$$

Appendix B
Numerical Optimization Problems

The following numerical optimization functions were considered in this paper. Fig. 15 shows their shapes approximately.

Sphere Function

Minimize

$$f(\mathbf{x}) = \sum_{i=1}^{N} x_i^2 \tag{16}$$

where $-100.0 \le x_i \le 100.0$ and $N = 30$. The global minimum value is 0.0 at $\mathbf{x} = (0, 0, \ldots, 0)$.

Ackley Function

Minimize

$$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e \tag{17}$$

where $-32.0 \le x_i \le 32.0$ and $N = 30$. The global minimum value is 0.0 at $\mathbf{x} = (0, 0, \ldots, 0)$.

Griewank Function

Minimize

$$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (18)$$

where $-600.0 \leq x_i \leq 600.0$ and $N = 30$. The global minimum value is 0.0 at $\mathbf{x} = (0, 0, \ldots, 0)$.

Rastrigin Function

Minimize

$$f(\mathbf{x}) = 10N + \sum_{i=1}^{N}\left(x_i^2 - 10\cos(2\pi x_i)\right) \quad (19)$$

where $-5.12 \leq x_i \leq 5.12$ and $N = 30$. The global minimum value is 0.0 at $\mathbf{x} = (0, 0, \ldots, 0)$.

Schwefel Function

Minimize

$$f(\mathbf{x}) = 418.9829N - \sum_{i=1}^{N} x_i \sin(\sqrt{|x_i|}) \quad (20)$$

where $-500.0 \leq x_i \leq 500.0$ and $N = 30$. The global minimum value is 0.0 at $\mathbf{x} = (-420.9687, -420.9687, \ldots, -420.9687)$.

Rosenbrock Function

Minimize

$$f(\mathbf{x}) = \sum_{i=1}^{N-1}\left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right) \quad (21)$$

where $-30.0 \leq x_i \leq 30.0$ and $N = 30$. The global minimum value is 0.0 at $\mathbf{x} = (1, 1, \ldots, 1)$.

Appendix C
Rosenbrock Function

As shown in Table VII, no algorithm could find out the global minimum of Rosenbrock function, although the maximum number of generations was selected to be 20,000. In the experiments, we could find that x_{30} did not converge to the optimal value 1 in most of the experiments. Let us see the partial derivative of $f_{\mathrm{Rosenbrock}}$ with respect to each x_i

$$\begin{aligned}
\frac{\partial f}{\partial x_1} &= 400x_1^3 - 400x_2x_1 + 2x_1 - 2 \in \Theta(x_1^3)\\
\frac{\partial f}{\partial x_2} &= 400x_2^3 - 400x_3x_2 + 202x_2 - 200x_1^2 - 2 \in \Theta(x_2^3)\\
&\vdots = \vdots\\
\frac{\partial f}{\partial x_{29}} &= 400x_{29}^3 - 400x_{30}x_{29} + 202x_{29} - 200x_{28}^2 - 2 \in \Theta(x_{29}^3)\\
\frac{\partial f}{\partial x_{30}} &= 200x_{30} - 200x_{29}^2 \in \Theta(x_{30}).
\end{aligned}$$

The partial derivative of $f_{\mathrm{Rosenbrock}}$ with respect to each x_i is of order of x_i^3. However, the partial derivative with respect to x_{30} is of order of x_{30}. The sensitivity of Rosenbrock function with respect to x_{30} is much smaller than that with respect to the others. It indicates the difficulty to make x_{30} converge to the optimal value by using only the function value as a fitness function. In particular, the error of x_{30} induced the error of x_{29} and the error of x_{i+1} also induced the error of x_i. Let us see the Rosenbrock function with respect to each pair of (x_i, x_{i+1})

$$\begin{aligned}
f(x_i, x_{i+1}) &= 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\\
&= f_e(x_i, x_{i+1}) + f_s(x_i).
\end{aligned} \quad (22)$$

The function of $f(x_i, x_{i+1})$ can be divided into two functions, $f_e(x_i, x_{i+1})$ and $f_s(x_i)$ as shown in (22). The optimal value of x_i is determined by $f_s(x_i)$. However, since the coefficient of $(x_{i+1} - x_i^2)^2$ in $f_e(x_i, x_{i+1})$ is too big, $f_s(x_i)$ is negligible. From this reason, each x_i converges not to 1, but to $\sqrt{x_{i+1}}$. Therefore, other techniques which can make x_i converge effectively to the optimal value are needed to obtain much better solutions.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful and constructive comments on this paper.

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 Univ. Press, 1996.

[3] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 580–593, Dec. 2002.

[4] G. W. Greenwood, "Finding solutions to NP problems: Philosophical differences between quantum and evolutionary search algorithms," in *Proc. 2001 Congress Evolutionary Computation*, Seoul, Korea, May 2001, pp. 815–822.

[5] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines," *J. Stat. Phys.*, vol. 22, pp. 563–591, 1980.

[6] R. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, vol. 21, no. 6, pp. 467–488, 1982.

[7] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum computer," in *Proc. Royal Society London A*, vol. 400, 1985, pp. 97–117.

[8] ——, "Qautnum computational networks," in *Proc. Royal Society London A*, vol. 425, 1989, pp. 73–90.

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

[10] D. R. Simon, "On the power of quantum computation," in *Proc. 35th Annu. Symp. Foundations Computer Science*, Sante Fe, NM, Nov. 1994, pp. 116–123.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Foundations Computer Science*, Sante Fe, NM, Nov. 1994, pp. 124–134.

[12] ——, "Quantum computing," *Documenta Mathematica*, vol. Extra Volume ICM, pp. 467–486, 1998.

[13] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th ACM Symp. Theory Computing*, 1996, pp. 212–219.

[14] ——, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett., Amer. Phys. Soc.*, vol. 79, no. 2, pp. 325–328, July 1997.

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

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

[17] L. Spector, H. Barnum, H. J. Bernstein, and N. Swamy, "Finding a better-than-classical quantum AND/OR algorithm using genetic programming," in *Proc. 1999 Congr. Evolutionary Computation*, vol. 3, Washington, D.C., July 1999, pp. 2239–2246.

[18] B. I. P. Rubinstein, "Evolving quantum circuits using genetic programming," in *Proc. 2001 Congr. Evolutionary Computation*, vol. 1, Seoul, Korea, May 2001, pp. 144–151.

[19] M. Lukac and M. Perkowski, "Evolving quantum circuits using genetic algorithm," in *Proc. 2002 NASA/DOD Conf. Evolvable Hardware*, July 2002, pp. 177–185.

[20] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proc. IEEE Int. Conf. Evolutionary Computation*, Nagoya, Japan, May 1996, pp. 61–66.

[21] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Proc. 2000 Congr. Evolutionary Computation*, vol. 2, La Jolla, CA, July 2000, pp. 1354–1360.

[22] K.-H. Han, K.-H. Park, C.-H. Lee, and J.-H. Kim, "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem," in *Proc. 2001 Congr. Evolutionary Computation*, vol. 2, Seoul, Korea, May 2001, pp. 1422–1429.

[23] 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 Trans. Inform. Syst.*, vol. E86-D, pp. 645–649, Mar. 2003.

[24] J.-S. Jang, K.-H. Han, and J.-H. Kim, "Quantum-inspired evolutionary algorithm-based face verification," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, July 2003, Proc. Genetic Evolutionary Computation Conf. (2724), pp. 2147–2156.

[25] K.-H. Han and J.-H. Kim, "On setting the parameters of quantum-inspired evolutionary algorithm for practical applications," in *Proc. 2003 Congr. Evolutionary Computation*, Canberra, Australia, Dec. 2003, pp. 178–184.

[26] M. Moore and A. Narayanan, "Quantum-Inspired Computing," Dept. Comput. Sci., Univ. Exeter, Exeter, U.K., 1995.

[27] AR Face Database [Online]. Available: http://rvl1.ecn.purdue.edu/ ãleix/aleix_face_DB.html

[28] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[29] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 82–102, July 1999.

[30] M. R. Garey and D. S. Johnson, *Computers and Intractability—A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[31] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd, revised and extended ed. Berlin, Germany: Springer-Verlag, 1999.

Kuk-Hyun Han (S'00–A'03) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1997, 1999, and 2003, respectively.

He is currently a Senior Engineer in the Digital Media R&D Center, Samsung Electronics Company, Ltd., Suwon, Gyeonggi, Korea. His research interests are in the areas of quantum-inspired evolutionary computation, Internet-based personal robot systems, intelligent control, real-time operating system, and ubiquitous computing.

Dr. Han was the recipient of the Presidential Award at the 38th Nationwide Science Exhibition, Korea, in 1992, the runner-up awards at the 1997 International Micro-Robot Soccer Tournament and the 2001 FIRA Cup China, and the Bronze and Gold Prizes at the 7th Samsung HumanTech Thesis Prize Award and the 9th Samsung HumanTech Thesis Prize Awards in 2001 and 2003, respectively.

Jong-Hwan Kim (S'85–M'88–SM'03) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981, 1983 and 1987, respectively.

Since 1988, he has been with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, where he is currently a Professor. He was a Visiting Scholar at Purdue University, West Lafayette, IN, and the University of California, Davis, in 1992 and 2000, respectively. He currently serves as an Associate Editor of the *International Journal of Intelligent and Fuzzy Systems*. He was Guest Editor of the special issues on *MiroSot* in 1996, for the *Journal of Robotics and Autonomous Systems*, and on *Soccer Robotics* in 2000, for the *Journal of Intelligent Automation and Soft Computing*. He was one of the cofounders of the Asia-Pacific Conference on *Simulated Evolution and Learning*. He was General Chair of the *2001 Congress on Evolutionary Computation* held in Seoul, Korea. His current research interests are in the area of evolutionary multiagent robotic systems.

Dr. Kim currently serves as an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. His name is included in the *Barons 500 Leaders for the New Century* as Founder of the Federation of International Robot-Soccer Association (FIRA) and the Robot Olympiad. He is currently serving the FIRA and the International Robot Olympiad Committee (IROC) as President. He was the recipient of the Choongang Young Investigator Award in 1988 from Choongang Memorial Foundation, the LG YonAm Foundation Research Fellowship in 1992, the Korean Presidential Award in 1997, and the SeoAm Foundation Research Fellowship in 1999.