2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

On the Analysis of the Quantum-inspired Evolutionary Algorithm with a Single Individual

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

***Abstract*— This paper discusses the reason why QEA works and verifies how QEA works. The theoretical analysis of the simplified model of the segment process of QEA shows that QEA with a single individual for ONEMAX problem guarantees the global solution in terms of expected running number of generations. The analysis for exploration shows clearly that QEA starts with a global search scheme and changes automatically into a local search scheme as generation advances because of its inherent probabilistic mechanism, which leads to a good balance between exploration and exploitation. For comparison purpose, simulated annealing is considered with three test functions. The results support the conclusions derived from the theoretical analysis of QEA with a single individual.**

I. INTRODUCTION

Evolutionary algorithms (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.

Quantum-inspired evolutionary algorithm (QEA) recently proposed in [1] can treat the balance between exploration and exploitation more easily 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, such as the quantum bit and the superposition of states. However, QEA is not a quantum algorithm, but a novel evolutionary algorithm [2]. Like any other evolutionary algorithms, QEA is also characterized by the representation of the individual, the evaluation function, and the population dynamics [3].

In [4], a probabilistic representation and a novel population dynamics inspired by quantum computing were first proposed. In [1], the basic structure of QEA and its characteristics were formulated and analyzed, respectively. According to [1], the results (tested on the knapsack problem) of QEA were proved to be better than those of CGA. In [5], some guidelines for setting the parameters of QEA were presented. In [6], QEA was extended to numerical optimization problems, and several research issues on QEA, such as a termination criterion, a variation operator, and a two-phase scheme, were discussed to improve its performance. In [7], QEA was applied to a decision boundary optimization for face verification. 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.

With no connection to quantum computing, a number of evolutionary algorithms that guide the exploration of the search space by building probabilistic models of promising solutions found have been introduced since the late 1990s [8]. These algorithms have shown to perform well on a variety of problems. In the population-based incremental learning (PBIL) which is a method of combining the mechanisms of a generational genetic algorithm with simple competitive learning [9], the solutions are represented by binary strings and the population of solutions is replaced with a probability vector. The compact genetic algorithm (cGA) [10] replaces the population with a single probability vector as in PBIL, however its modification method of the probability vector is different from PBIL. The univariate marginal distribution algorithm (UMDA) [11] also assumes that the probabilities of bits are independent of each other. It should be worthwhile to compare QEA with these estimation of distribution algorithms (EDAs), though it is beyond the scope of this paper.

This paper discusses the reason why QEA works and verifies how QEA works. The QEA algorithm with a single individual is considered with a simple test function, ONEMAX problem. A simplified model of the segment process of QEA is defined to analyze its convergence for exploitation, and Shannon entropy is introduced to investigate the strategy of exploration for QEA. From the analysis of the simplified model of the segment process of QEA, QEA with a single individual for the ONEMAX problem guarantees the global solution in terms of expected running number of generations. The analysis for exploration shows clearly that QEA starts with a global search scheme and changes automatically into a local search scheme as generation advances because of its inherent probabilistic mechanism, which leads to a good balance between exploration and exploitation. For comparison purpose, simulated annealing which seems to be similar to QEA with a single individual at a glance (but, QEA is quite different from simulated annealing) is considered with three test functions. The results support the conclusions derived from the theoretical analysis of QEA with a single individual.

Kuk-Hyun Han is with the Digital Media R&D Center, Samsung Electronics Company, Ltd., Maetan-3dong, Yeongtong-gu, Suwon, Gyeonggi-do, 443-742, Republic of Korea (email: khhan@khhan.com).

Jong-Hwan Kim is with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Guseong-dong, Yuseong-gu, Daejeon, 305-701, Republic of Korea (email: johkim@rit.kaist.ac.kr).

0-7803-9487-9/06/$20.00/©2006 IEEE

This paper is organized as follows. Section II describes the QEA algorithm. Section III describes the verification of the QEA algorithm with a single individual from the two viewpoints of exploration and exploitation. The experimental results on three test functions were also summarized for the purpose of comparison between QEA with a single individual and simulated annealing. Finally, concluding remarks follow in Section IV.

II. PRELIMINARIES

A. Representation

QEA uses a Q-bit[1] 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[2], (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.

B. 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 $\mathbf{q}_j^t$ is a Q-bit individual defined as

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

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

Figure 1 shows the procedure of QEA that can be explained in the following:

```
Procedure QEA
begin
          t ← 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 ← 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 b among B(t)
xi)            if (global migration condition)
               then migrate b to B(t) globally
xii)           else if (local migration condition)
               then migrate b_j^t in B(t) to B(t) locally
          end
end
```

Fig. 1. Procedure of QEA.

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. However, it should be noted that the performance of QEA can be influenced by the initial value. The effect of the initial value was discussed in [6].

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) Until the termination condition is satisfied, QEA is running in the **while** loop. Note that termination criteria were described in [6].

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 $\mathbf{x}_{j_l}^t$, 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:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix}, \qquad (3)$$

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. $\Delta\theta_i$

[1]Q-bit is defined in [1], and means quantum-inspired bit which is different from qubit.

[2]QEA uses real numbers for α and β of Q-bit in this paper. However, QEA can be extended to use complex numbers for those of Q-bit to include more information for mutual dependencies of Q-bits.

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 H_ϵ gate which is a novel Q-gate as a variation operator was designed in [6].

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. 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. Verification of the QEA algorithm

There have been some works done based on the theoretical analysis of EAs for certain simple functions [12], [13], [14], [15], [16]. However, the theories behind these analyses cannot be applied to the analysis of QEA, since the structure of QEA is quite different from any other EAs. In this section, the reason why and how QEA works is investigated by using a simple function with two viewpoints like its exploitation and exploration.

A. Exploitation

A theoretical model for the whole process of QEA is hard to find, since each state of QEA is dependent on the past history. However, if a simplified model for a segment of the QEA process (as shown in Figure 2) is considered, the abstract model can be regarded as a Markov chain.

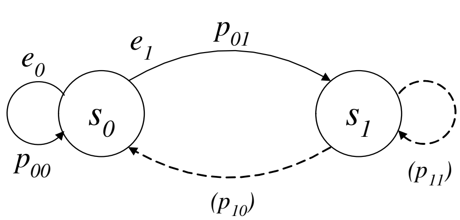


Fig. 2. Simplified process model for a segment of the QEA process.

The simplified model of the segment process of QEA represents the process which is defined during the state holding period t_h between the t_sth generation when the current best solution visits and the t_eth (or $(t_s + t_h)$th) generation when the current best solution jumps to another better solution. In Figure 2, s_0 is the state which indicates the state when the current best solution is maintained, and s_1 is the state which indicates the state when the current best solution is changed to another better solution. e_0 is the event that states that the observed solution is worse than the current best solution, and e_1 is the event that states that the observed solution is better than the current best solution. And p_{ij}, $i, j = 0, 1$, is the transition probability from state i to state j. It should be noted that p_{10} and p_{11} are not needed, since the process corresponding to this model is terminated if the state is changed from s_0 to s_1. The whole process of QEA can be regarded as a sequence of segment processes.

The segment process of QEA (SPQEA) is described by using Markov process [17] as follows:

$$\begin{aligned} &SPQEA = (\mathbb{E}, \mathbb{S}, \Gamma, p, p_0) \\ &\mathbb{E} = \{e_0, e_1\}, \quad \mathbb{S} = \{s_0, s_1\}, \\ &\Gamma(s_0) = \{e_0, e_1\}, \quad \Gamma(s_1) = \{\}, \\ &p(s_0; s_0, e_0) = p_{00}, \quad p(s_1; s_0, e_1) = p_{01}, \\ &p_0(s_0) = 1, \quad p_0(s_1) = 0, \end{aligned} \tag{4}$$

where $\mathbb{E}$ is an event set, $\mathbb{S}$ a state space, $\Gamma(s)$ a set of feasible events defined for all $s \in \mathbb{S}$ with $\Gamma(s) \subseteq \mathbb{E}$, $p(s'; s, e')$ a state transition probability defined for all $s, s' \in \mathbb{S}$, $e' \in \mathbb{E}$, and such that $p(s'; s, e') = 0$ for all $e' \notin \Gamma(s)$, and $p_0(s)$ the probability mass function $P[S_0 = s]$, $s \in \mathbb{S}$, of the initial state S_0 which is a discrete random variable.

Let us consider the OneMax problem as follows:
OneMax problem: Maximize

$$\text{OneMax}(\mathbf{x}) = \sum_{i=1}^{m} x_i, \tag{5}$$

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 $\mathbf{x} = 111 \cdots 1$.
Let us suppose that all the rotation angles of the rotation gate in QEA are zeros. Then the QEA process is the same as the process of random search. In this case, each solution in the search space has the same probability and its probability is invariant all the time. It means that this process can be modelled by using only one SPQEA with $p_{00} = \frac{2^m - 1}{2^m}$ and $p_{01} = \frac{1}{2^m}$. The expected running number of generations for this model is described in the following.

Theorem 1: The expected running number of generations t_h of the random search is

$$t_h = -\frac{\log 2}{\log(1 - p_{01})}, \tag{6}$$

where p_{01} is the transition probability from state s_0 to state s_1.
Proof. Let $V(s)$ be the number of generations spent at state s when it is visited.

$$\begin{aligned} P[V(s_0) = 1] &= p_{01} \\ P[V(s_0) = 2] &= p_{00}p_{01} = (1 - p_{01})p_{01} \\ P[V(s_0) = 3] &= p_{00}^2 p_{01} = (1 - p_{01})^2 p_{01} \\ \vdots &= \vdots \\ P[V(s_0) = t] &= p_{00}^{t-1} p_{01} = (1 - p_{01})^{t-1} p_{01} \end{aligned}$$

To give the expected running number of generations t_h, the summation of the probability $P[V(s_0) = k]$ from $k = 1$ to

$k = t_h$ should be $\frac{1}{2}$.

$$\sum_{t=1}^{t_h} P[V(s_0) = t] = 1 - (1 - p_{01})^{t_h} = \frac{1}{2}$$

$$\therefore t_h = -\frac{\log 2}{\log(1 - p_{01})}. \blacksquare$$

Theorem 2: The expected running number of generations t_h of the random search for the ONEMAX problem for length m is

$$t_h = -\frac{\log 2}{\log(1 - \frac{1}{2^m})}. \quad (7)$$

Proof. Each solution in the search space for the random search has the same probability $\frac{1}{2^m}$ and its probability is invariant all the time. Let s_0 be the state when the current best solution is one of all the possible solutions except the global maximum. Then the transition probabilities p_{00} and p_{01} are $\frac{2^m-1}{2^m}$ and $\frac{1}{2^m}$, respectively. By Theorem 1,

$$t_h = -\frac{\log 2}{\log(1 - p_{01})} = -\frac{\log 2}{\log(1 - \frac{1}{2^m})}. \blacksquare$$

However, if the rotation angles are not zeros, the QEA process should be considered as a sequence of SPQEA models. Also, one SPQEA model should not be considered as a homogeneous Markov chain, since the transition probability p_{ij} is dependent on generation t. Let us consider only one segment of the QEA process, SPQEA. The transition probability at the generation t is supposed to be $p_{01}(t) = \xi(t)p_{01}(t-1)$, where $\xi(t)$ is the increasing rate of the transition probability $p_{01}(t)$, $0 < p_{01}(t) \leq 1$, $\xi(1) = 1$, and $1 < \xi(t) \ll \frac{1}{p_{01}(t)}$ for $t > 1$, the expected running number of generations of SPQEA can be obtained as follows.

Theorem 3: The expected running number of generations t_h of SPQEA with time-varying transition probability can be approximated as

$$t_h \approx \frac{\log\left(1 - \frac{\xi}{2} + \frac{\xi-1}{2p_{01}(0)}\right)}{\log(\xi - \xi p_{01}(0))}, \quad (8)$$

where $p_{01}(0)$ is the initial transition probability from state s_0 to state s_1 and ξ is a constant satisfying $\sum_{k=1}^{t_h} p_{01}(k) = \sum_{k=1}^{t_h} \xi^{k-1} p_{01}(0)$.

Proof. Let $p_{01}(t)$ be the transition probability from s_0 to s_1 and $\xi(t)$ the increasing rate of the transition probability at the generation t, where $\xi(t) = \frac{p_{01}(t)}{p_{01}(t-1)}$ for $t > 1$ and $\xi(1)$ = 1. The probabilities for the state holding period of s_0 are

$$\begin{aligned}
P[V(s_0) = 1] &= p_{01}(1) = \xi(1)p_{01}(0) = p_{01}(0) \\
P[V(s_0) = 2] &= (1 - p_{01}(1))p_{01}(2) \\
&= (1 - p_{01}(0))\xi(2)p_{01}(0) \\
P[V(s_0) = 3] &= (1 - p_{01}(0))(1 - \xi(2)p_{01}(0)) \times \\
&\quad \xi(3)\xi(2)p_{01}(0) \\
P[V(s_0) = 4] &= (1 - p_{01}(0))(1 - \xi(2)p_{01}(0)) \times \\
&\quad (1 - \xi(3)\xi(2)p_{01}(0))\xi(4)\xi(3)\xi(2)p_{01}(0) \\
\vdots &= \vdots .
\end{aligned}$$

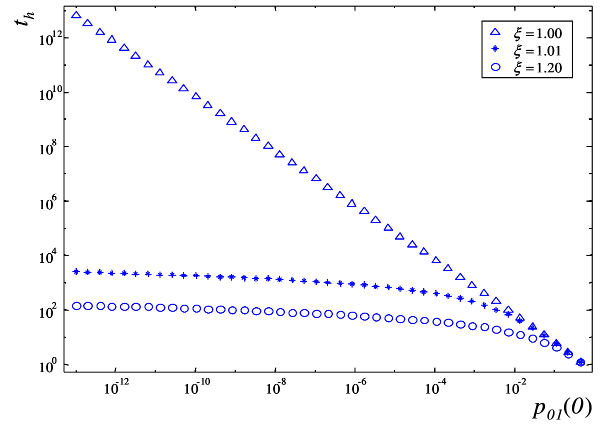


Fig. 3. Comparison of the expected running number of generations (t_h) with respect to the initial transition probability ($p_{01}(0)$) between QEA (ξ = 1.01 and 1.2) and random search ($\xi = 1.0$). ξ is the increasing rate of (8). A logarithmic (base 10) scale is used for the horizontal and vertical axes.

Let $\xi(t)$ be a constant ξ satisfying $\sum_{k=1}^{t} p_{01}(k) = \sum_{k=1}^{t} \xi^{k-1} p_{01}(0)$, the above can be rewritten as

$$\begin{aligned}
P[V(s_0) = 1] &= p_{01}(0) \\
P[V(s_0) = 2] &= (1 - p_{01}(0))\xi p_{01}(0) \\
P[V(s_0) = 3] &= (1 - p_{01}(0))(1 - \xi p_{01}(0))\xi^2 p_{01}(0) \\
P[V(s_0) = 4] &= (1 - p_{01}(0))(1 - \xi p_{01}(0)) \times \\
&\quad (1 - \xi^2 p_{01}(0))\xi^3 p_{01}(0) \\
\vdots &= \vdots \\
P[V(s_0) = t] &= \prod_{k=0}^{t-2} (1 - \xi^k p_{01}(0))\xi^{t-1} p_{01}(0).
\end{aligned}$$

Since ξ can be considered as $\left(1 + \frac{\delta}{p_{01}(0)}\right)$, where $0 < \delta \ll p_{01}(0)$, $P[V(s_0) = t]$ can be approximated as

$$P[V(s_0) = t] \approx (1 - p_{01}(0))^{t-1} \xi^{t-1} p_{01}(0).$$

To give the expected running number of generations t_h, the summation of the probability $P[V(s_0) = k]$ from $k = 1$ to $k = t_h$ should be $\frac{1}{2}$. Therefore, the expected running number of generations of SPQEA is obtained as

$$\sum_{t=1}^{t_h} P[V(s_0) = t] \approx p_{01}(0) \frac{1 - (1 - p_{01})^{t_h} \xi^{t_h}}{1 - (1 - p_{01})\xi} = \frac{1}{2}$$

$$\therefore t_h \approx \frac{\log\left(1 - \frac{\xi}{2} + \frac{\xi-1}{2p_{01}(0)}\right)}{\log(\xi - \xi p_{01}(0))}. \blacksquare$$

It should be noted that if ξ is 1, t_h of (8) remains the same as that of (6) for random search. Figure 3 shows the expected running number of generations t_h with respect to the initial transition probability $p_{01}(0)$. In the case of random search ($\xi = 1.0$), if $p_{01}(0)$ is small, t_h is very large, e.g. $t_h|_{p_{01}(0)=1.0\times10^{-13}} = 6.9 \times 10^{12}$ at $\xi = 1.0$. However, for the cases of QEA ($\xi = 1.01$ and 1.2), the expected running number of generations is much smaller than that of random search, e.g. $t_h|_{p_{01}(0)=1.0\times10^{-13}} = 2,475$ at $\xi = 1.01$ and 151 at $\xi = 1.2$.

TABLE I

SIMULATION RESULTS FOR THE VERIFICATION OF THE INCREASING RATE ξ BY A SIMPLE CALCULATION FOR THE ONEMAX PROBLEM FOR LENGTH m, WHERE $m = 4$. t IS THE TIME STEP (OR GENERATION), $\mathbf{x}_t$ THE OBSERVED SOLUTION AT t, $p_{01}(t)$ THE TRANSITION PROBABILITY FROM s_0 TO s_1, AND $\xi(t)$ THE INCREASING RATE $\frac{p_{01}(t)}{p_{01}(t-1)} \cdot p_{01}(t)$ WAS OBTAINED BY THE SUM OF $P[X_t = 1111]$, $P[X_t = 1110]$, $P[X_t = 1101]$, $P[X_t = 1011]$, AND $P[X_t = 0111]$.

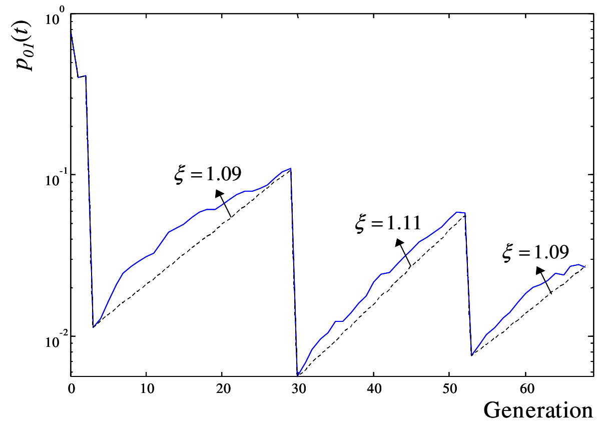
$t = 1$			$t = 2$			$t = 3$		
$\mathbf{x}_0$	$p_{01}(1)$	$\xi(1)$	$\mathbf{x}_1$	$p_{01}(2)$	$\xi(2)$	$\mathbf{x}_2$	$p_{01}(3)$	$\xi(3)$
1100	0.3125	1.0	0000	0.3849	1.2318	0000	0.4590	1.1923
						0001	0.4168	1.0828
						0010	0.4168	1.0828
						0100	0.4209	1.0935
						1000	0.4209	1.0935
			0001	0.3458	1.1067	0000	0.4168	1.2052
						0001	0.3761	1.0876
						0010	0.3743	1.0824
						0100	0.3801	1.0991
						1000	0.3801	1.0991
			0010	0.3458	1.1067	0000	0.4168	1.2052
						0001	0.3743	1.0824
						0010	0.3761	1.0876
						0100	0.3801	1.0991
						1000	0.3801	1.0991
			0100	0.3476	1.1124	0000	0.4209	1.2109
						0001	0.3801	1.0934
						0010	0.3801	1.0934
						0100	0.3815	1.0974
						1000	0.3849	1.1073
			1000	0.3476	1.1124	0000	0.4209	1.2109
						0001	0.3801	1.0934
						0010	0.3801	1.0934
						0100	0.3849	1.1073
						1000	0.3815	1.0974

Let us consider the ONEMAX problem for length m, where $m = 4$. If the initial state s_0 has the current best solution of 1100, the transition probability is $p_{01}(0) = \frac{5}{2^m} = 0.3125$, since all the solutions have the same probability $\frac{1}{2^m}$ at $t = 0$ and there are five solutions, such as 1111, 1110, 1101, 1011, and 0111, better than 1100. Table I shows the simulation results of all the possible situations from $t = 1$ to $t = 3$ to verify the value of the increasing rate $\xi(t)$. The rotation angle of p (or $|n|$) for the rotation gate was set to 0.03π in this simple calculation. The table shows that the values of $\xi(t)$ are greater than 1 in all the possible situations. It means that the probability at which the better solution is to be found increases each generation and the better solution can be found in a shorter span of time as shown in (8).

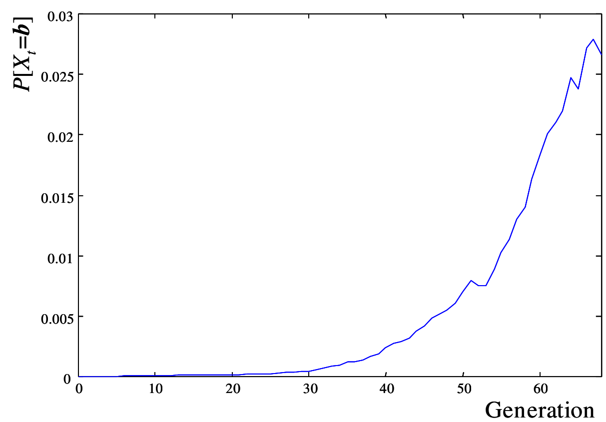
Figure 4 shows the experimental results of QEA for the ONEMAX problem for length m, where $m = 16$. In Figure 4 (a), the dotted line gives a reference for finding a proper ξ which can provide an upper bound of the expected running number of generations for each segment process. If the segment processes of QEA are modelled by SPQEA, the expected running numbers of generations of (8) with values of $\xi = 1.09$, 1.1, and 1.09 can provide the upper bound for those of the 2nd, 3rd, and 4th segment processes of QEA, respectively.

It should be noted that the increasing rate $\xi(t)$ of the transition probability was greater than 1 in the results of Table I and Figure 4. Also, the statement that $\xi(t)$ is always greater than 1 for the ONEMAX problem for length m can be verified by a simple calculation.

Theorem 4: The expected number of Q-bits toward the state 1 for the ONEMAX problem is a positive value in SPQEA.



(a) Transition probability ($p_{01}(t)$)



(b) Probability of the best solution

Fig. 4. Experimental results of QEA1 for the ONEMAX problem for length m, where $m = 16$. The dotted line gives a reference for finding a proper ξ which can provide an upper bound of the expected running number of generations for each segment process. A logarithmic (base 10) scale is used for the vertical axis of (a).

Proof. Let m be the binary string length and n_1 the number of ones for the current best solution. If the number of ones for the observed binary solution is k, where $k < n_1$, the number of Q-bits toward the state 1 is $(n_1 - k)$ and the number of binary solutions which have k ones is $\frac{m!}{k!(m-k)!}$. Since the number of all the possible binary solutions in SPQEA is $\sum_{k=0}^{n_1-1} \frac{m!}{k!(m-k)!}$, the expected number of Q-bits toward the state 1 for the ONEMAX problem for length m is a positive value:

$$\frac{\sum_{k=0}^{n_1-1} \left(\frac{m!}{k!(m-k)!} (n_1 - k) \right)}{\sum_{k=0}^{n_1-1} \frac{m!}{k!(m-k)!}} > 0. \quad \blacksquare \tag{9}$$

In other words, QEA for the ONEMAX problem has the tendency of converging to better solutions in a short span of time. The reason can be explained by the concept of building block which is a small, tightly clustered group of genes. In the case of the ONEMAX problem, the group of ones for the current best solution can be regarded as a building block and the probability of this building block is increased by the rotation gate. As a consequence, the probabilities of the better solutions increase.

It is worthwhile to mention that a sequence of SPQEA for the ONEMAX problem guarantees the global solution in terms of expected running number of generations, since the number of better solutions always decreases after one sequence of SPQEA and it eventually becomes 1 to be considered as the only global solution.

B. Exploration

To increase the performance of EAs for various optimization problems, exploration as well as exploitation discussed earlier should be considered. The global optimum for a unimodal function which has no local optimum can be exploited without exploration. However, if an EA has no scheme for exploration, the global optimum for a multimodal function which has many local optima is not guaranteed to be found out.

To verify the strategy of exploration for QEA, Shannon entropy [18] can be considered as a measure of the amount of information included in a Q-bit individual. The entropy of $p(\mathbf{x})$, $\mathbf{x} \in \mathbb{X}$, is described as

$$I(p(\mathbf{x})) = -p(\mathbf{x}) \log_2 p(\mathbf{x}),$$

where $\mathbb{X}$ is a search space, $I(\cdot)$ the entropy (or information) of the probability, and $p(\mathbf{x})$ the probability of $\mathbf{x}$, i.e. $P[X = \mathbf{x}]$. The entropy of the probability distribution for the search space represented by a Q-bit individual is

$$I(p(\mathbf{x})|\mathbf{x} \in \mathbb{X}) = -\sum_{\mathbf{x}\in\mathbb{X}} p(\mathbf{x}) \log_2 p(\mathbf{x}), \qquad (10)$$

where

$$p(\mathbf{x}) = \prod_{i=1}^{m} p_i$$

with

$$p_i = \begin{cases} |\alpha_i|^2, & \text{if } x_i = 0 \\ |\beta_i|^2, & \text{if } x_i = 1 \end{cases},$$

where x_i is the ith bit of $\mathbf{x}$ and (α_i, β_i) is the ith Q-bit. It should be noted that the entropy initially has the maximum value of m and it decreases gradually, since each probability of $p(\mathbf{x})$, $\mathbf{x} \in \mathbb{X}$, is shifted with a small amount by the rotation gate as generation advances.

For comparison purpose, let us consider $(1+1)$ GA with mutation rate $\frac{1}{m}$, where m is the length of binary solution.

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.

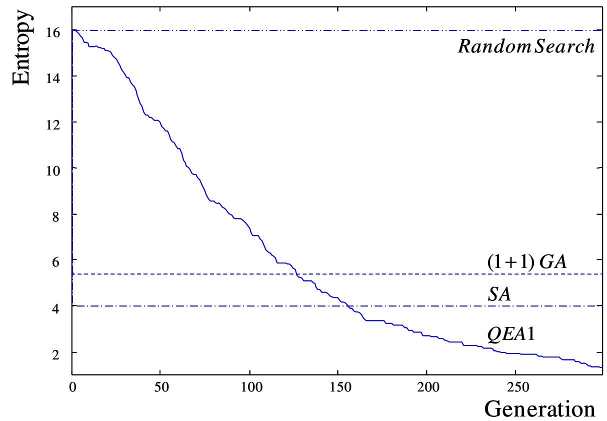


Fig. 5. Comparison of the entropy of the probability distribution for the search space with respect to the time step (or generation) among QEA1, $(1+1)$ GA, SA, and the random search. The results were obtained from the ONEMAX problem for length m, where $m = 16$.

Theorem 5: The entropy of the probability distribution for the search space represented by $(1+1)$ GA is a constant regardless of the generation t for $t > 0$.

Proof. Let $\mathbf{x}$ be the current binary solution, $\mathbf{x}'$ the next binary solution, and h the Hamming distance between $\mathbf{x}$ and $\mathbf{x}'$. If $\mathbf{x}'$ with Hamming distance h from $\mathbf{x}$ is $\mathbf{x}^h$, the probability of $\mathbf{x}^h$ can be described as

$$p(\mathbf{x}^h) = \left(\frac{m-1}{m}\right)^{m-h} \left(\frac{1}{m}\right)^{h},$$

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

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

The entropy of the probability distribution for the search space is obtained as

$$\begin{aligned} I(p(\mathbf{x})|\mathbf{x} \in \mathbb{X}) &= -\sum_{\mathbf{x}\in\mathbb{X}} p(\mathbf{x}) \log_2 p(\mathbf{x}) \\ &= -\sum_{h=0}^{m} \left(n(\mathbf{x}^h) p(\mathbf{x}^h) \log_2 p(\mathbf{x}^h)\right). \qquad (11) \end{aligned}$$

Therefore, the entropy of the probability distribution for the search space represented by $(1+1)$ GA is a constant regardless of the generation t as shown in (11). ∎

Let us also consider a simulated annealing (SA) method which is a specific version for binary representation (see Appendix A).

Theorem 6: The entropy of the probability distribution for the search space represented by SA with binary representation is a constant regardless of the time step t for $t > 0$.

Proof. Let $\mathbf{x}$ be the current binary solution, $\mathbf{x}'$ the next solution, and h the Hamming distance between $\mathbf{x}$ and $\mathbf{x}'$. Then the distance h is always 1 for SA with binary representation. If the length of binary string is m, the number of all the possible $\mathbf{x}^1$ is m and the probability of $\mathbf{x}^1$ is $\frac{1}{m}$. Since $p(\mathbf{x}^h)$ is 0 for all h excluding $h = 1$, the entropy of the probability distribution for the search space is obtained as

$$I(p(\mathbf{x})|\mathbf{x} \in \mathbb{X}) = -\sum_{\mathbf{x}\in\mathbb{X}} p(\mathbf{x}) \log_2 p(\mathbf{x}) = -\log_2 \frac{1}{m}. \quad (12)$$

Therefore, the entropy of the probability distribution for the search space represented by SA with binary representation is a constant regardless of the time step t as shown in (12). ∎

Figure 5 shows the differences of the entropy of the probability distribution for the search space among QEA1, $(1+1)$ GA, SA, and the random search. While the entropy values for $(1+1)$ GA, SA, and the random search are constant values of (11), (12), and m, respectively, that of QEA1 is not a constant. The entropy value of QEA1 is initially the same as that of the random search, and it decreases gradually as generation advances. This result shows clearly that the strategy of QEA for exploration differs from those of $(1+1)$ GA and SA. It is hard to say that which strategy is the superior one compared to others, since the performance of the strategy may depend on the specific problems. However, it is clear that QEA starts with a global search scheme and changes automatically into a local search scheme as generation advances because of its inherent probabilistic mechanism, which leads to a good balance between exploration and exploitation as already mentioned in [1].

C. Simple Experiments

The theoretical analysis of QEA with a single individual for the ONEMAX problem had been discussed in Section III-A, B. Here, three test functions (see Appendix B) were considered to verify the performance of QEA with a single individual. For comparison purpose, SA (see Appendix A) was considered. As a performance measure of the algorithms, we picked up the best search cost for the first hitting time over 50 runs. The number of times the fitness function was called was regarded as the search cost, since the evaluation of fitness function generally consumes most of time compared to any other functions. The number of bits for the three test functions was set to 25 bits (per variable). The value of ϵ for the H_ϵ gate was set to 0.01π.

Table II shows the experimental results for the three test functions (13)-(15). In the results of $f_{DeJong1}$ and $f_{DeJong3}$, QEA with a single individual yielded better results compared to SA. In the results of $f_{DeJong2}$, which is relatively simple function compared to the other functions, SA performed better compared to QEA with a single individual. In particular, it should be noted that the results of $f_{DeJong3}$ showed that SA had several failure cases of which search cost was greater than 10^6. The reason is that the function (3) has many discontinuous valleys as shown in Figure 7 (c) and SA may fall into one such valley. From these results, it is worthwhile to mention that QEA with a single individual performs well although the search space is distorted or it has many discontinuous valleys.

TABLE II
EXPERIMENTAL RESULTS OF THE THREE TEST FUNCTIONS (13)-(15). EACH PARENTHESIZED VALUE OF QEA IS THE ROTATION ANGLE p (OR $|n|$) AND THAT OF SA IS THE VALUE OF COOLING PARAMETER k FOR ITS TEMPERATURE SCHEDULER. THE NUMBER OF RUNS WAS 50. $m.$, σ, AND $r.$ REPRESENT THE MEAN BEST OF SEARCH COST, THE STANDARD DEVIATION OF SEARCH COST, AND THE SUCCESS RATE, RESPECTIVELY. THE VALUES MARKED WITH * WERE OBTAINED EXCLUDING THE FAILURE CASES FOR WHICH SEARCH COST WAS GREATER THAN 10^6.

		$f_{DeJong1}$	$f_{DeJong2}$	$f_{DeJong3}$
QEA (0.0001π)	m.	31544.9	9196.0	3802.1
	σ	15944.0	1764.2	797.8
	$r.$	50 / 50	50 / 50	50 / 50
QEA (0.0005π)	m.	122712.1	4117.2	1229.9
	σ	94058.7	3962.7	344.5
	$r.$	50 / 50	50 / 50	50 / 50
QEA (0.001π)	m.	141143.2	7306.8	894.7
	σ	90850.7	16085.5	248.4
	$r.$	50 / 50	50 / 50	50 / 50
SA (0.01)	m.	299097.1	1705.0	1279.4
	σ	145643.4	826.3	782.0
	$r.$	50 / 50	50 / 50	50 / 50
SA (0.1)	m.	185057.8	1446.9	1207.5*
	σ	71646.6	682.3	916.2*
	$r.$	50 / 50	50 / 50	46 / 50
SA (1.0)	m.	193786.4	1446.8	706.2*
	σ	64214.7	637.0	409.7*
	$r.$	50 / 50	50 / 50	21 / 50

IV. CONCLUSIONS

This paper discussed the reason why QEA works and verified how QEA works. The theoretical analysis of the simplified model of the segment process of QEA showed that QEA with a single individual for the ONEMAX problem guarantees the global solution in terms of expected running number of generations. The analysis for exploration showed clearly that QEA starts with a global search scheme and changes automatically into a local search scheme as generation advances because of its inherent probabilistic mechanism, which leads to a good balance between exploration and exploitation. For comparison purpose, simulated annealing was considered with three test functions. The results support the conclusions derived from the theoretical analysis of QEA with a single individual.

It is worthwhile to mention that the proposed approach with two viewpoints of exploitation and exploration will be useful in understanding and verifying QEA, although QEA with a single individual is verified only for ONEMAX problem.

APPENDICES

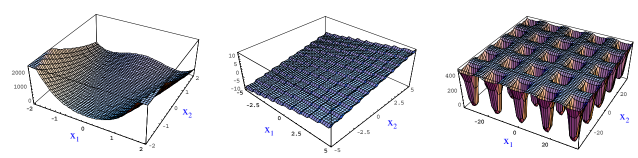
A. Simulated annealing

SA is quite similar to the hill climbing method [18]. Instead of picking the best move, it picks a random move. If the move actually improves the situation, it is always executed. Otherwise, the algorithm makes the move with some probability less than 1. The probability decreases exponentially as time advances.

Figure 6 shows the procedure SA which is a specific version for binary representation. In this figure, $\mathbf{x}_c$ is a current binary string, $\mathbf{x}_n$ a new binary string, T the current temperature, t the time step, $s(T,t)$ the scheduler for the

```
Procedure SA
begin
    t ← 0
    initialize temperature T
    select a current string x_c at random
    while (T > 0) do
    begin
        t ← t + 1;  T ← s(T, t)
        select a new string x_n
            in the neighborhood of x_c by flipping a single bit of x_c
        ΔE ← f(x_n) − f(x_c)
        if (ΔE > 0) then x_c ← x_n
        else if (exp^{ΔE/T} > random[0, 1]) then x_c ← x_n
    end
end
```

Fig. 6. Simulated annealing.



(a) $f_{DeJong1}$ (b) $f_{DeJong2}$ (c) $f_{DeJong3}$

Fig. 7. Test functions of (13)-(15).

temperature T, $f(\cdot)$ the fitness function of the problem, and $random[0,1]$ a random number from the range $[0,1)$.

There are several techniques for implementing the temperature scheduler $s(T,t)$. In this paper, the following technique was used for implementing the scheduler $s(T,t) = \frac{1}{k(t+1)}$, where k is the parameter for cooling temperature.

B. Test functions

The following numerical optimization functions were considered in this paper.

De Jong function (1): Minimize

$$f(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \quad (13)$$

where $-2.048 \leq x_i \leq 2.048$. The global minimum value is 0.0 at $(x_1, x_2) = (1, 1)$.

De Jong function (2): Minimize

$$f(\mathbf{x}) = \sum_{i=1}^{5} \text{integer}(x_i), \quad (14)$$

where $-5.12 \leq x_i \leq 5.12$. The global minimum value is -30 for all $-5.12 \leq x_i < -5.0$.

De Jong function (3): Minimize

$$f(\mathbf{x}) = \frac{1}{\frac{1}{K} + \sum_{j=1}^{25} g_j^{-1}(x_1, x_2)}, \quad (15)$$

where $g_j(x_1, x_2) = c_j + \sum_{i=1}^{2} (x_i - a_{ij})^6$, $-65.536 \leq x_i \leq 65.536$, $K = 500$, $c_j = j$, and $[a_{ij}]$ is

$$\begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \cdots & 32 & 32 & 32 \end{bmatrix}.$$

The global minimum is 0.998 at $(x_1, x_2) = (-32, -32)$.

Figure 7 shows their shapes approximately.

ACKNOWLEDGMENT

This work was supported by the Ministry of Information and Communication, Korea, under the Information Technology Research Center (ITRC) Support Program.

REFERENCES

[1] 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.

[2] G. W. Greenwood, "Finding Solutions to NP Problems: Philosophical Differences Between Quantum and Evolutionary Search Algorithms," in *Proceedings of the 2001 Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Press, pp. 815-822, May 2001.

[3] K.-H. Han, *Quantum-inspired Evolutionary Algorithm*. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea, 2003.

[4] 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, Jul. 2000.

[5] K.-H. Han and J.-H. Kim, "On Setting the Parameters of Quantum-inspired Evolutionary Algorithm for Practical Applications," in *Proceedings of the 2003 Congress on Evolutionary Computation*, IEEE Press, pp. 178-184, Dec. 2003.

[6] K.-H. Han and J.-H. Kim, "Quantum-inspired Evolutionary Algorithms with a New Termination Criterion, H_ϵ Gate, and Two-Phase Scheme," *IEEE Transactions on Evolutionary Computation*, Piscataway, NJ: IEEE Press, vol. 8, no. 2, pp. 156-169, Apr. 2004.

[7] J.-S. Jang, K.-H. Han, and J.-H. Kim, "Evolutionary Algorithm-based Face Verification," *Pattern Recognition Letters*, Elsevier B. V., Vol. 25, No. 16, pp. 1857-1865, Dec. 2004.

[8] M. Pelikan, D. E. Goldberg, and F. Lobo, *A Survey of Optimization by Building and Using Probabilistic Models*. Technical report: IlliGAL Report No. 99018, Department of General Engineering, University of Illinois at Urbana-Champaign, IL, Sep. 1999.

[9] S. Baluja, *Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*. Technical report: CMU-CS-94-163, School of Computer Science, Carnegie Mellon University, PA, Jun. 1994.

[10] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," in *Proceedings of the International Conference on Evolutionary Computation*, Piscataway, NJ: IEEE Press, pp. 523-528, May 1998.

[11] H. Mühlenbein and G. Paaβ, "From recombination of genes to the estimation of distributions I. Binary parameters," pp. 178-187, 1996.

[12] G. Rudolph, "Convergence Rates of Evolutionary Algorithms for a Class of Convex Objective Functions," *Control and Cybernetics*, Poland: Systems Research Institute, vol. 26, no. 3, pp. 375-390, 1997.

[13] J. Garnier, L. Kallel, and M. Schoenauer, "Rigorous Hitting Times for Binary Mutations," *Evlutionary Computation*, MIT Press, vol. 7, no. 2, pp. 173-203, 1999.

[14] J. He and X. Yao, "Drift Analysis and Average Time Complexity of Evolutionary Algorithms," *Artificial Intelligence*, Elsevier Science, vol. 127, no. 1, pp. 57-85, Mar. 2001.

[15] S. Droste, T. Jansen, and I. Wegener,, "On the Analysis of the (1+1) Evolutionary Algorithm," *Theoretical Computer Science*, Elserier Science, vol. 276, pp. 51-81, Apr. 2002.

[16] J. He and X. Yao, "From an Individual to a Population: An Analysis of the First Hitting Time of Population-Based Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, Piscataway, NJ: IEEE Press, vol. 6, no. 5, pp. 495-511, Oct. 2002.

[17] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*. Boston: IRWIN, 1993.

[18] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, New Jersey: Prentice Hall, 1995.