

# Quantum-inspired Multiobjective Evolutionary Algorithm for Multiobjective 0/1 Knapsack Problems

Yehoon Kim, Jong-Hwan Kim, and Kuk-Hyun Han

**Abstract**— This paper proposes a multiobjective evolutionary algorithm (MOEA) inspired by quantum computing, which is named quantum-inspired multiobjective evolutionary algorithm (QMEA). In the previous papers, quantum-inspired evolutionary algorithm (QEA) was proved to be better than conventional genetic algorithms for single-objective optimization problems. To improve the quality of the nondominated set as well as the diversity of population in multiobjective problems, QMEA is proposed by employing the concept and principles of quantum computing such as uncertainty, superposition, and interference. Experimental results pertaining to the multiobjective 0/1 knapsack problem show that QMEA finds solutions close to the Pareto-optimal front while maintaining a better spread of nondominated set.

## I. INTRODUCTION

Evolutionary algorithms (EAs) inspired from the processes of evolution in nature are stochastic search mechanisms. A lot of current research in EAs is focused simultaneous optimization problems of several objectives. The growing interest in highly complex search space has spurred the growth of multiobjective evolutionary algorithms (MOEAs) [1]-[6]. The strength Pareto evolutionary algorithm (SPEA) [1] was proposed based on elitism by maintaining an external population. Its improved version, SPEA2 [2], employing a refined fitness assignment, coupled with an enhanced archive truncation technique, was followed. The nondominated sorting genetic algorithm (NSGA) appeared earlier [3] and a better performing NSGA-II was presented [4]. They have tried to remedy many drawbacks of NSGA. Their approach uses an elite conservation strategy and diversity preservation mechanism. It shows good performance in solving challenging problems.

To explore effective the search space of multiobjective problems (MOPs), the concepts of quantum computing are adopted in the proposed approach. Quantum mechanical computers were proposed in the early 1980s [7], [8]. It was then formalized in the late 1980s [9], [10]. Quantum-inspired evolutionary computing [11]-[17] for digital computer has been one of the issues and research on merging quantum

computing into evolutionary computation has started since the late 1990s. Recently, quantum-inspired evolutionary algorithms (QEAs) was proposed [15], [16]. QEA can explore and exploit search space for a global optimal solution.

This paper proposes quantum-inspired multiobjective evolutionary algorithm (QMEA) to improve proximity to the Pareto-optimal front, preserving diversity intact by employing advantages of QEA. The improving proximity means to find the better solutions which are evaluated as good individuals by fitness function. The investigation is within the NSGA-II framework. NSGA-II is a strong elitist method with mechanisms to maintain diversity efficiently using nondominated sorting and crowding distance assignment. It is even more powerful if the elitism is further strengthened and the solutions are spread out by quantum mechanism. Multiple observations of Q-bit individuals allow a local search in the vicinity of the nondominated solutions. Also, maintaining best Q-bit individuals in every generation can avoid the possibility of losing high quality individuals. Furthermore to deal with quantum computing concepts in MOEAs, the comparison mechanism is presented between the best group and the others. Convergence and preservation of diversity being the key issues under scrutiny, the proposed approach is expected to help improve the performance of any MOEA.

This paper is organized as follows. Section II presents an overview of QEA and Section III describes MOEAs, in particular NSGA-II. Section IV defines the procedure of proposed QMEA. The experimental results in Section V show that QMEA is capable of approaching a proximate Pareto-optimal front and with good diversity. Finally concluding remarks follow in Section VI.

## II. QUANTUM-INSPIRED EVOLUTIONARY ALGORITHM

QEA utilizes a new representation, called a Q-bit, for the probabilistic representation that is based on the concept of qubits [15]. A qubit may be in the “1” state, in the “0” state, or in any superposition of the two [22]. The state of a qubit can be represented as

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where  $\alpha$  and  $\beta$  are complex numbers that specify the probability amplitudes of the corresponding states. Normalization of the state to unity always guarantees:

$$|\alpha|^2 + |\beta|^2 = 1. \quad (2)$$

A Q-bits is defined as the smallest unit of information in

Yehoon Kim and Jong-Hwan Kim are with EECS Department, KAIST, Guseong-dong, Yuseong-gu, Daejeon-shi, 305-701, Republic of Korea (e-mails: {yhkim, johkim}@rit.kaist.ac.kr).

Kuk-Hyun Han is with the Digital Media R&D Center, Samsung Electronics Co., Ltd., 416, Maetan-3dong, Youngtong-gu, Suwon, Gyeonggi, 443-742, Republic of Korea (e-mail: khhan@khhan.com)

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

QEA, which is defined with a pair of number,  $(\alpha, \beta)$ , as

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (3)$$

where  $|\alpha|^2 + |\beta|^2 = 1$ .

A Q-bit individual is defined as a string of Q-bits.  $Q(t) = \{\mathbf{q}_1^t, \mathbf{q}_2^t, \dots, \mathbf{q}_n^t\}$  at generation  $t$ , where  $n$  is the population size, and  $\mathbf{q}_j^t$ ,  $j = 1, 2, \dots, n$ , is each Q-bit individual. Since the Q-bit representation is able to express as a linear superposition of states probabilistically, it is profitable for generating diversity in the evolutionary process. A Q-bit individual is defined as:

$$\mathbf{q}_j^t = \begin{bmatrix} \alpha'_{j1} & \alpha'_{j2} & \dots & \alpha'_{jm} \\ \beta'_{j1} & \beta'_{j2} & \dots & \beta'_{jm} \end{bmatrix} \quad (4)$$

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

The procedure of QEA and the overall structure are described in the following:

---

### Procedure QEA

#### Begin

$t \leftarrow 0$

- i) initialize  $Q(t)$
- ii) make  $P(t)$  by observing the states of  $Q(t)$
- iii) evaluate  $P(t)$
- iv) store the best solutions among  $P(t)$  into  $B(t)$
- 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** (migration condition)
- then** migrate  $\mathbf{b}$  or  $\mathbf{b}_j^t$  to  $B(t)$  globally or locally
- end**

#### end

---

i)  $\mathbf{q}_j^0 = \mathbf{q}_j^t|_{t=0}$ ,  $j = 1, 2, \dots, n$ , are initialized with  $1/\sqrt{2}$ . It means that one Q-bit individual,  $\mathbf{q}_j^0$ , represents the linear superposition of all possible states with the same probability.

ii) This step makes binary solutions in  $P(0)$  by observing the states of  $Q(0)$ , where  $P(0) = \{\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_n^0\}$  at generation  $t = 0$ . One binary solution,  $\mathbf{x}_j^0$ ,  $j = 1, 2, \dots, n$ , is formed by selecting either 0 or 1 for each bit using the probability, either  $|\alpha_i^0|^2$  or  $|\beta_i^0|^2$ ,  $i = 1, 2, \dots, m$ , of  $\mathbf{q}_j^0$ . QEA is working on a digital computer and collapsing into a single state does not occur in QEA.

iii) Each binary solution,  $\mathbf{x}_j^0$ , is evaluated to give a level of its fitness.

iv) The initial best solutions among the binary solutions are stored into  $B(0)$ , where  $B(0) = \{\mathbf{b}_1^0, \mathbf{b}_2^0, \dots, \mathbf{b}_n^0\}$ , and  $\mathbf{b}_j^0 (= \mathbf{b}_j^t|_{t=0})$  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.

vi), vii) In the while loop, binary solutions in  $P(t)$  are formed by multiple observing the states of  $Q(t-1)$  as in step ii), and each binary solution is evaluated for the fitness value.  $\mathbf{x}_j^t$  should be replaced by  $\mathbf{x}_{jl}^t$ , where  $l$  is an observation index.

viii) Q-bit individuals in  $Q(t)$  are updated by applying rotation gate defined below

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \quad (5)$$

where  $\Delta\theta$  is a rotation angle of each Q-bit.  $\Delta\theta$  should be designed in compliance with the application problem.

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

xi) If the migration condition is satisfied, the best solution  $\mathbf{b}$  is migrated to  $B(t)$ . The global or local migration operation is helpful to treat the balance between exploration and exploitation in QEA for single-objective optimization problems (SOPs). However migration operation can have a negative influence in MOPs because migration of solutions may disturb the endeavor for preserving diversity. If the local best individual is substituted by the global best solution, solutions are then crowded in search space. To prevent this problem, the proposed algorithm does not utilize migration operation.

### III. MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

MOEAs have two goals: first, solutions have to be close to the Pareto-optimal front and second, diversity of population should be preserved well in order to find as many solutions as possible. In this section, the main schemes of the state of the art MOEA is reviewed such as fast nondominated sorting and crowding distance calculation for both issues [4].

#### A. Fast nondominated sorting

Elitism, which prevents losing the best individuals, is a good strategy [18]. Therefore, the elitism becomes a general scheme in MOEAs [20]. For the elitism, population must be sorted into different levels. The sort procedure [4] is as follows: nondominated front is founded and temporarily saved to search next nondominated front. This procedure is repeated until all individuals are ranked. Fast nondominated sorting algorithm reduces computation time from  $O(MN^3)$  to  $O(MN^2)$ .

### B. Crowding distance calculation

In order to satisfy the second issue, efficient diversity preservation method, where the density of each individual is estimated, was proposed [4]. Normalized crowding distance calculation is useful to obtain an estimate of the density of solutions. The crowding distance of a solution refers to the average side length of the cuboid that has the vertices of the nearest neighbors.  $O(MN \log N)$  computations take to get all crowding distance values.

## IV. QUANTUM-INSPIRED MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM

This section describes the proposed QMEA for enhancing proximity and diversity of nondominated solutions.

### A. Main procedure

A bridge for fitting QEA into the MOEA framework is required. The framework (that was devised in [4]) behind the fitting procedure is employed. The whole procedure of QMEA is as follows.

---

#### procedure QMEA

**begin**

$t \leftarrow 0$

- i) initialize  $Q(t)$
- ii) make  $P(t)$  by observing the states of  $Q(t)$
- iii) evaluate  $P(t)$
- iv) **while** (not termination condition) **do**
- begin**
- $t \leftarrow t + 1$
- v) make  $P(t)$  by observing the states of  $Q(t-1)$
- vi) evaluate  $P(t)$
- vii) run the fast nondominated sort algorithm for  $P(t) \cup P(t-1)$
- viii) calculate crowding distance and sort
- ix)  $P(t)$  is formed by the first  $N$  elements in the sorted population  $2N$ .
- x)  $Q(t)$  is classified into several groups
- xi) update  $Q(t)$  using Q-gates refer to best group
- end**

**end**

---

i) ~ vi) These steps are same as QEA procedure. In this paper, the termination criterion used is maximum number of generations.

vii) The individuals in  $2N$  population ( $P(t) \cup P(t-1)$ ) are rearranged by the fast nondominated sort algorithm, which is introduced in [4].

viii) Also,  $2N$  population is sorted by crowding distance calculation.

ix) The survival of the superior  $N$  individuals in a generation follows in the same way as in [4]. The survived individuals form  $P(t)$ . The Q-bit individuals corresponded to  $P(t)$  is also copied to  $Q(t)$ .

x) Group classification rule is utilized in this step.

xi) Instead of crossover and mutation, the update operation

(rotation gate) perturbs the Q-bit.

### B. Update Operation

Q-gates in QEA play a role of perturbation operation in genetic algorithm. A rotation gate  $U(\Delta\theta)$  [15] is employed to update a Q-bit individual as a variation operator in QMEA.

### C. Group Classification

When Q-bit individuals are updated by a rotation gate, the update operation refers to bits of the best solution. Population ( $N$ ) is divided into several groups ( $G_1, G_2, \dots, G_n$ ) from the top front in the sorted population  $P(t)$ . Since better (higher ranked and less crowded) solutions have been already sorted,  $G_1$  is the best group, which is utilized to update Q-bit individuals of other groups. The individual in  $G_1$  is the best solution  $\mathbf{b}$ . Q-bit individuals in lower ranked groups ( $G_2, G_3, \dots, G_n$ ) are updated according to best group  $G_1$  (Fig. 1). For the elitism, Q-bits in  $G_1$  are retained. Comparison between  $\mathbf{x}$  and  $\mathbf{b}$  follows the rules:

All individuals in  $G_i$  compare with  $i^{\text{th}}$  solution in  $G_1$ ,

$$S_i = \left\lfloor \frac{N}{n} \right\rfloor, \quad S_i \geq n \quad (6)$$

$$S_n = N - \sum_{i=1}^{n-1} S_i \quad (7)$$

where  $S_i$  is the number of individuals in a  $i^{\text{th}}$  group,  $i = 1, 2, \dots, n-1$ ,  $N$  is the population size,  $n$  is the total number of groups. Since  $S_i$  should be an integer value, the value of  $S_i$  can be different.

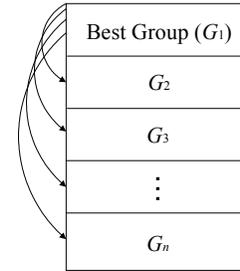


Fig. 1. Comparison between the groups

## V. EXPERIMENTAL RESULTS

In this section, the multiobjective 0/1 knapsack problem is briefly reviewed. It also describes the performance measure method, and investigates the performance of QMEA.

### A. Multiobjective 0/1 Knapsack Problem

The multiobjective 0/1 Knapsack Problems have been a good benchmark as a test problem to evaluate the performance of MOEAs. The problem is to find items, maximizing the total profit such that the total weight does not exceed the given capacity. The problem is to find  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$  such that

$$\sum_{j=1}^n w_{i,j} \cdot x_j \leq c_i \quad (1 \leq i \leq k) \quad (8)$$

where  $p_{ij}$  is the profit of item  $j$  in the knapsack  $i$ ,  $w_{ij}$  is the weight of item  $j$  in the knapsack  $i$ , and  $c_i$  is the capacity of the knapsack  $i$ .  $f(x) = (f_1(x), \dots, f_n(x))$  is maximized, where

$$f_i(x) = \sum_{j=1}^n p_{i,j} \cdot x_j \quad (9)$$

A greedy repair method is used to produce the best outcomes for constraint handling.

### B. Performance Measure

In order to evaluate the quality of nondominated solutions, following two scaling-independent metrics [20] are chosen: Size of the dominated space ( $S$ ) and coverage of two sets ( $C$ ). A diversity metric is also employed [21] that efficiently evaluates the spread of nondominated solutions. The diversity metric is given as follows:

$$D = \frac{\sum_{k=1}^n (f_k^{(\max)} - f_k^{(\min)})}{\sqrt{\frac{1}{N_0} \sum_{i=1}^{|N_0|} (d_i - \bar{d})^2}} \quad (10)$$

where  $N_0$  is a set of nondominated solutions,  $d_i$  is the minimal distance between the  $i^{\text{th}}$  solution and the nearest neighbor,  $\bar{d}$  is the mean value of all  $d_i$ .  $f_k^{(\max)}$  ( $f_k^{(\min)}$ ) represents the maximum (minimum) fitness of the  $k^{\text{th}}$  objective. A larger value means a higher diversity of the nondominated solutions.

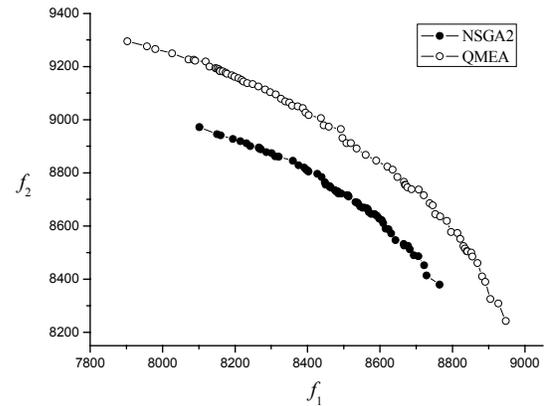
### C. Experimental Results

We chose two-knapsacks with 250 items, 500 items, and 750 items for test purposes. The NSGA-II was chosen as a reference. A pair-wise tournament selection, binary-coded GA with 1-point crossover and bitwise mutation were in force for NSGA-II. Parameters used in this experiment are given in Table I.  $l$  is the length of binary string.

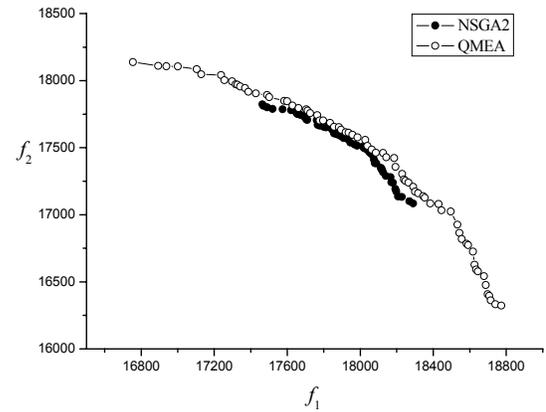
TABLE I PARAMETER SETTING	
Parameters	Values
Population size ( $N$ )	100
No. of generations	100
Crossover Prob. ( $p_c$ )	0.9
Mutation Prob. ( $p_m$ )	1 / $l$
No. of observations	10
No. of groups ( $n$ )	10
$\Delta\theta$	$0.01\pi$

Since similar proximity of two methods (QMEA and NSGA-II) is possible in a large enough population, The population size was fixed at 100. The number of generations

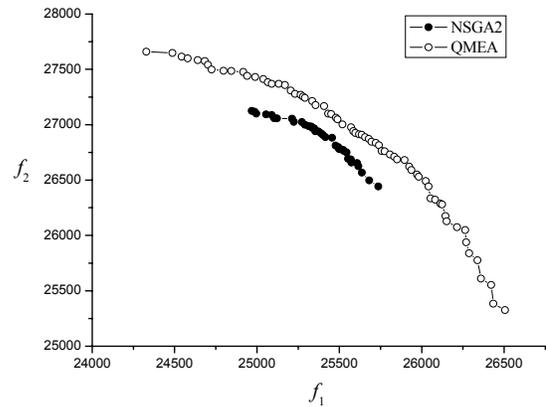
was also fixed at 100 for the same reason. The parameter values were obtained from experimental evaluations. The comparison results were averaged over 10 tests.



(a) 250 items



(b) 500 items



(c) 750 items

Fig. 2. Comparison results for 2-knapsack problem

Fig. 2 compares the results found by the QMEA with that of NSGA-II. Only non-dominated solutions are plotted in the graph. The results show that QMEA can find higher quality solutions than NSGA-II. It is due to the characteristics of QEA. Binary strings of high quality can be obtained by

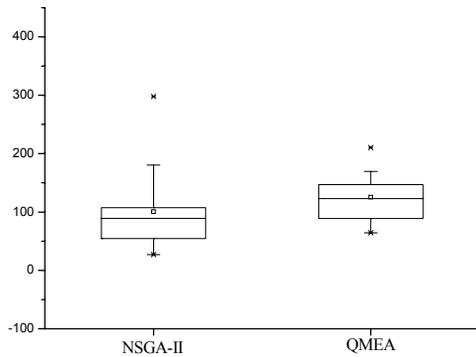
multiple observations in each Q-bit individuals. If multiple observations of Q-bit individuals are applied for creating the offspring, there is a high probability that the offspring visits some region that is unexplored from the vicinity of the chosen candidates. It helps that a set of solutions is close to Pareto-optimal front and spread out.

Table II shows the scaling-independent metrics, where the size of the dominated space of QMEA is larger than that of NSGA-II in each items. These results show that QMEA dominates more search space than NSGA-II. Coverage also shows that QMEA dominates NSGA-II.

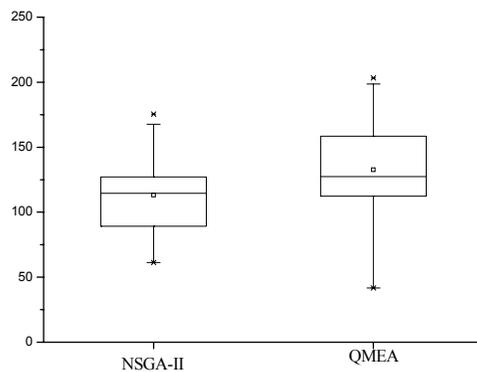
TABLE II  
SIZE OF THE DOMINATED SPACE AND COVERAGE

metric	250 items	500 items	750 items
$S(QMEA)$	$8.2784708 \cdot 10^7$	$3.39353211 \cdot 10^8$	$7.31504155 \cdot 10^8$
$S(NSGA2)$	$7.8479896 \cdot 10^7$	$3.25536637 \cdot 10^8$	$6.97593468 \cdot 10^8$
$C(QMEA, NSGA2)$	1.0	1.0	1.0

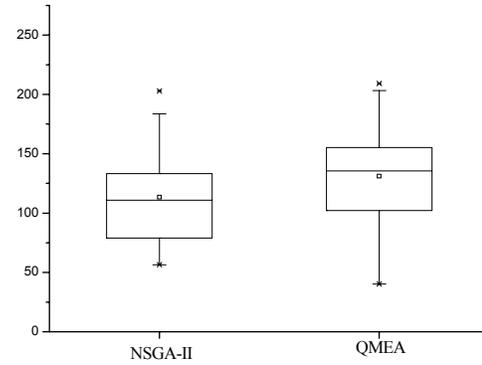
The diversity preservation performance is compared in Fig. 3. It shows an increased tendency toward diversity preservation of QMEA. The reason is that it searches broad regions of solution space. In conclusion, QMEA performs better than NSGA-II on both counts such as proximity and distribution of solutions.



(a) 250 items



(b) 500 items



(c) 750 items

Fig. 3. Diversity measure for 2-knapsack problem

## VI. CONCLUSIONS

In this paper, quantum-inspired multiobjective evolutionary algorithm (QMEA) was proposed based on the quantum computing concept. QMEA is the extended version of QEA for multiobjective problems. The experimental results for multiobjective 0/1 knapsack problem supported the claim that proposed approach exhibits better proximity performance as well as diversity maintenance. Schemes that utilize a nondominated set can benefit from the proposed approach because QEA is applicable to any MOEA framework.

## REFERENCES

- [1] E. Zitzler, and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.
- [2] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm," *Technical Report 103*, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Zurich (2001).
- [3] N. Srinivas and K. Deb, "Multiobjective Function Optimization using Nondominated Sorting Genetic Algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221-248, Fall 1995.
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [5] E. Zitzler, K. Deb and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evol. Comput.*, vol. 8, no. 2, pp. 173-195, Summer 2000
- [6] K. Deb and T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence," E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, D. Corne (eds.) in *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Berlin, pp. 67-81, 2001.
- [7] P. Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," *Journal of Statistical Physics*, vol. 22, pp. 563-591, 1980.
- [8] R. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467-488, 1982.
- [9] D. Deutsch, "Quantum Theory, the Church-Turing principle and the universal quantum computer," in *Proceedings of the Royal Society of London A*, vol. 400, pp. 97-117, 1985.
- [10] D. Deutsch, "Quantum computational networks," in *Proceedings of the Royal Society of London A*, vol. 425, pp. 73-90, 1989.

- [11] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Proc. Congress on Evolutionary Computation*, vol. 2, pp. 1354-1360, La Jolla, CA, July 2000.
- [12] K.-H. Han, K.-H. Park, C.-H. Lee, and J.-H. Kim, "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem," in *Proc. Congress on Evolutionary Computation*, vol.2, pp. 1422-1429, Seoul, Korea, May 2001.
- [13] 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.
- [14] K.-H. Han and J.-H. Kim, "On setting the parameters of quantum-inspired evolutionary algorithm for practical applications," in *Proc. Congress on Evolutionary Computation*, Canberra, Australia, pp. 178-184, Dec. 2003.
- [15] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evolutionary Computation*, vol. 6, pp. 580-593, 2002
- [16] K.-H. Han and J.-H. Kim "Quantum-inspired evolutionary algorithms with a new termination criterion, He Gate, and two phase scheme," accepted for publication in the *IEEE Trans. Evolutionary Computation*, vol. 8, No. 2, 2004.
- [17] 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, Proc. Genetic Evolutionary Computation Conf. (2724), pp. 2147-2156, July 2003
- [18] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 1975, (University Microfilms No. 76-9381).
- [19] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, "Evolutionary algorithms for solving multi-objective problems," ser. Genetic algorithms and evolutionary computation: 5. New York: Kluwer Academic, 2002.
- [20] E.Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," *Ph. D. Thesis, Swiss Federal Institute of Technology*, Zurich, Switzerland, 1999.
- [21] H. Li, Q. Zhang, E. Tsang, and J. A. Ford, "Hybrid Estimation of Distribution Algorithm for Multiobjective Knapsack Problem," *EvoCOP 2004, LNCS 3004*, pp. 145-154, 2004.
- [22] T. Hey, "Quantum computing: An introduction," in *Computing & Control Engineering Journal*. Piscataway, NJ: IEEE Press, June 1999, vol. 10, no. 3, pp. 105-112