# A Multi-Start Quantum-Inspired Evolutionary Algorithm for Solving Combinatorial Optimization Problems


Parvaz Mahdabi
Tarbiat Modares University
Computer Engineering Dept.
Tehran, Iran
parvaz.mahdabi@modares.ac.ir

Saeed Jalili
Tarbiat Modares University
Computer Engineering Dept.
Tehran, Iran
sjalili@modares.ac.ir

Mahdi Abadi
Tarbiat Modares University
Computer Engineering Dept.
Tehran, Iran
abadi@modares.ac.ir



## ABSTRACT

Quantum-inspired evolutionary algorithms (QIEAs), as a subset of evolutionary computation, are based on the principles of quantum computing such as quantum bits and quantum superposition. In this paper, we propose a multi-start quantum-inspired evolutionary algorithm, called MSQIEA. To improve the performance of the algorithm, a multi-measurement operator and a new strategy for updating the rotation angle is proposed. When Q-bit individuals start to converge to their final states, the best solution is stored and all Q-bits in each Q-bit individual are reinitialized. We compare the effectiveness of MSQIEA with a popular quantum-inspired evolutionary algorithm, called QEA, for solving 0-1 knapsack problem. The experimental results show that MSQIEA outperforms QEA and finds a solution with higher profit.



## Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

## General Terms

Algorithms, Experimentation, Theory

## Keywords

Evolutionary Algorithms, Quantum Computing, Combinatorial Optimization, Knapsack Problem


## 1. INTRODUCTION

Quantum-inspired evolutionary algorithms (QIEAs), a novel subsection in the evolutionary computation field, are built by merging the concepts of quantum computing with evolutionary algorithms for solving the combinatorial optimization problems. QIEAs consider the concepts of quantum computing such as quantum bits, superposition of the states, and the quantum gates as the basis [1]. Quantum bit representation and quantum gate operators are used for denoting the population individuals and simulating the evolution process, respectively. Performance of QIEAs is superior to that of conventional genetic algorithms as a result of their powerful global search ability, rapid convergence, short computation time, and small population size [1-3].



*GECCO'08*, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.


## 2. MSQIEA

The structure of MSQIEA is summarized as follows:

**procedure** MSQIEA
  $t \leftarrow 0$
1. Initialize $Q(t)$, $C(t)$, and $B(t)$
  **while** $t < t_{\max}$ **do**
    $t \leftarrow t+1$
2. Update each current measurement $c_i(t)$ and personal best measurement $b_i(t)$ using a multi-measurement operator
3. Calculate the global best measurement $\hat{b}(t)$
4. Update each Q-bit individual $q_i(t)$ using a Q-gate operator
5. **if** the population $Q(t)$ has converged **then**
6. $\hat{b}(t+1) \leftarrow \hat{b}(t)$
7. Reinitialize $Q(t)$ and $B(t)$
    **end if**
    $Q(t+1) \leftarrow Q(t)$ , $B(t+1) \leftarrow B(t)$
  **end while**
**end procedure**

1. In the initialization step, $Q(0)$, $C(0)$, and $B(0)$ are initialized as follows: To initialize $Q(0)$, the probability amplitudes of all Q-bits are set to the same value.

$$\alpha_{ij}(0) = \beta_{ij}(0) = \frac{1}{\sqrt{2}} \tag{1}$$

where $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$. $n$ is the size of the population and $m$ is the length of each Q-bit individual. To initialize $C(0)$ and $B(0)$, first, each Q-bit individual $q_i(0)$ is measured to form a binary solution $x_i(0)$ of length $m$ by selecting either 0 or 1 for each bit using the probability, either $|\alpha_{ij}(0)|^2$ or $|\beta_{ij}(0)|^2$ of $j$th Q-bit, respectively. The binary solution $x_i(0)$ is evaluated to give the level of its fitness. Then, the binary solution $x_i(0)$ is stored into both the current measurement $c_i(0)$ and the personal best measurement $b_i(0)$.

2. In this step, $c_i(t)$ and $b_i(t)$ are updated using a multi-measurement operator. In this operator, each $q_i(t)$ is measured

to form a binary solution $x_i(t)$. Then $c_i(t)$ is updated as

$$c_i(t) = \begin{cases} c_i(t) & \text{if } f(c_i(t)) > f(x_i(t)) \\ x_i(t) & \text{if } f(c_i(t)) \le f(x_i(t)) \end{cases} \tag{2}$$

The above process is repeated $l_0$ times, where $l_0$ is the number of measurements. Then $b_i(t)$ is updated as

$$b_i(t) = \begin{cases} b_i(t) & \text{if } f(b_i(t)) > f(c_i(t)) \\ c_i(t) & \text{if } f(b_i(t)) \le f(c_i(t)) \end{cases} \tag{3}$$

3. The global best measurement $\hat{b}(t)$ is calculated as the best among the personal best measurements $b_i(t), i = 1, 2, ..., n$.
4. In this step, each $q_i(t)$ is updated by applying the Q-gate operator $U$ on each Q-bit $q_{ij}(t)$.

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

$\Delta\theta_{ij}(t)$ is the rotation angle of $q_{ij}(t)$ towards "0" or "1".

$$\Delta\theta_{ij}(t) = sign(\xi_{ij}(t)) \cdot \vartheta_{ij}(t) \tag{5}$$

where $\xi_{ij}(t)$ is the Q-bit phase of $q_{ij}(t)$ and its sign indicates which quadrant $q_{ij}(t)$ lies in.

$$\xi_{ij}(t) = \arctan(\beta_{ij}(t) / \alpha_{ij}(t)) \tag{6}$$

$\vartheta_{ij}(t)$ is the rotation step for $q_{ij}(t)$.

$$\vartheta_{ij}(t) = \theta_0 \cdot (r_{1j}(t) \cdot (\hat{b}_j(t) - c_{ij}(t)) + r_{2j}(t) \cdot (b_{ij}(t) - c_{ij}(t))) \tag{7}$$

where $r_{1j}(t)$ and $r_{2j}(t)$ are randomly selected from one of the values 0 or 1. $\theta_0$ is a user defined parameter.

5. As the probability of each Q-bit approaches either 1 or 0 by the Q-gate operator, the Q-bit individuals start to converge to their final states and the diversity property of the population disappears gradually. Hence, in the step 5, the convergence criterion $C_{av} > \gamma$ in [2] is used to detect this situation.

If the convergence criterion is met, the following steps will be taken:

6. The global best measurement $\hat{b}(t)$ is stored in order to retain the knowledge of previous best solutions.
7. Diversity of the population is increased by reinitializing $Q(t)$ and $B(t)$. To reinitialize $Q(t)$, the probability amplitudes of all Q-bits are set to predefined values.

$$(\alpha_{ij}(t), \beta_{ij}(t)) = \begin{cases} (\varepsilon_0, \sqrt{1-\varepsilon_0^{\,2}}) & \text{if } \hat{b}_j(t) = 0 \\ (\sqrt{1-\varepsilon_0^{\,2}}, \varepsilon_0) & \text{if } \hat{b}_j(t) = 1 \end{cases} \tag{8}$$

where $(\alpha_{ij}(t), \beta_{ij}(t))$ are the probability amplitudes of the Q-bit $q_{ij}(t)$ and $\hat{b}_j(t)$ is the $j$th bit of $\hat{b}(t)$. $\varepsilon_0 \ge 1/\sqrt{2}$ is the predefined reinitialization value.

$B(t)$ is reinitialized as described in the step 1.

# 3. EXPERIMENTAL RESULTS

In this section, we compare the performance of MSQIEA for solving 0-1 knapsack problem with QEA [1], as a popular quantum-inspired evolutionary algorithm

To make the results comparable with those of QEA, we used the similar dataset and the same repair method as described in [1]. The comparison of QEA and MSQIEA on 0-1 knapsack problem is presented in Table 1. The table reports the best, mean and worst solutions found by QEA and MSQIEA over 30 runs on three instances of the problem with 100, 250, and 500 items. In the experiments of MSQIEA, the parameters were set to $\varepsilon_0 = 1/\sqrt{2}$, $l_0 = 5$, $\gamma = 0.99$, and $t_{\max} = 1000$. $\theta_0$ was set to $0.01\pi$, $0.03\pi$, and $0.04\pi$ for the problem instances with 100, 250, and 500 items, respectively.

**Table 1. Experimental results of the 0-1 knapsack problem**

| Method (pop-size) | Profit | Items | | |
|---|---|---|---|---|
| | | 100 | 250 | 500 |
| QEA [1] (10) | best | 612.70 | 1525.20 | 3025.80 |
| | mean | 609.50 | 1518.70 | 3008.00 |
| | worst | 607.60 | 1515.20 | 2996.10 |
| MSQIEA (10) | best | 612.73 | 1530.28 | 3036.35 |
| | mean | 612.06 | 1525.94 | 3027.48 |
| | worst | 607.73 | 1515.28 | 3016.35 |

As shown in Table 1, MSQIEA performs significantly better than QEA and finds a solution with higher profit. The value of $\theta_0$ should be designed in compliance with the application problem, however, the values from $0.01\pi$ to $0.05\pi$ are recommended.


# 4. ACKNOWLEDGMENTS

This work was supported in part by the ITRC.


# 5. REFERENCES


[1] Han, K.-H. and Kim, J.-H. 2002. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE T. Evolut. Comput. 6, 6 (Dec. 2002), 580-593.

[2] Han, K.-H. and Kim, J.-H. 2004. Quantum-inspired evolutionary algorithms with a new termination criterion, $H_\varepsilon$ gate, and two-phase scheme. IEEE T. Evolut. Comput. 8, 2 (April 2004), 156-169.

[3] Zhang, G. and Rong, H. 2007. Real-observation quantum-inspired evolutionary algorithm for a class of numerical optimization Problems. In Proceedings of the International Conference on Computational Science (Beijing, China, May 27 - 30, 2007). ICCS '07, LNCS 4490, 989-996.