# A Novel Quantum Evolutionary Algorithm Based on Dynamic Neighborhood Topology

Feng Qi, Qianqian Feng, Xiyu Liu, and Yinghong Ma

Shandong Normal University, Jinan 250014, China
qfsdnu@126.com

**Abstract.** A variant of quantum evolutionary algorithm based on dynamic neighborhood topology(DNTQEA) is proposed in this paper. In DNTQEA, the neighborhood of a quantum particle are not fixed but dynamically changed, and the learning mechanism of a quantum particle includes two parts, the global best experience of all quantum particles in population, and the best experiences of its all neighbors, which collectively guide the evolving direction. The experimental results demonstrate the better performance of the DNTQEA in solving combinatorial optimization problems when compared with other quantum evolutionary algorithms.

**Keywords:** Quantum Evolutionary Algorithm, Particle Swarm Optimization, Dynamic Neighborhood Topology, Algorithm.

## 1 Introduction

Quantum computing with its powerful computing power has become one of the most focused technology of current science. By considering the quantum information processing, researchers attend to introduce quantum computing mechanism into some traditional optimization algorithms. The combination of quantum computing and evolutionary computation was proposed in [1] named quantum genetic algorithm. In [2][3] quantum-inspired evolutionary algorithms(QEA) are first investigated for a class of combinatorial optimization problems in which quantum rotation gates act as update operators. Many works have tried to improve the performance of QEA. In [4] a new two phase scheme and a new He Gate is proposed for QEA. Reference [5] establishes that QEA is an original algorithm that belongs to the class of estimation of distribution algorithms (EDAs), while the common points and specifics of QEA compared to other EDAs are highlighted. Since proposed, QEA has been applied on several applications in science. Using QEA and Markov Model [6] presents a new method for static VAR which considers existing wing generator voltages and transformer taps as controller to regulate the voltage profile in a distribution system with wind farms. In order to solve the problem of highly non-linear economic load dispatch problem with value point loading [7] proposes an improved real quantum evolutionary algorithm which shows better performance than QEA.

Generally, the structure of the population in evolutionary algorithms is an important parameter. A graph based evolutionary algorithm is proposed in [8]

Y. Tan et al. (Eds.): ICSI 2014, Part I, LNCS 8794, pp. 267–274, 2014.

in which the individuals are located on the nodes of a graph structured population. The effect of variable population structure on Particle Swarm Optimization is investigated in [9]. Random graphs and their performance on several criteria are compared in their work. Similar with the above algorithms, Tayarani [10] proposed a sinusoid size ring structure QEA, experimental results show that the ring structure can be an efficient architecture for an effective Exploration/Exploitation tradeoff and improves the performance of QEA. [11] proposes a dynamic structured interaction among members of population in QEA and the study shows that cellular structure is the best. The structure of the population in above algorithms is fixed and the relationship among individuals never change in evolving process, this characteristics may cause the optimization process trap in local optimums and make the algorithm unstable.

As the similar particle study strategy, many hybrid algorithms by combing QEA and particle swarm optimization(PSO) are proposed . [12] proposed a binary Quantum-behaved PSO algorithm with cooperative approach, the updating method of particle's previous best position and swarm's global best position are performed in each dimension of solution vector to avoid loss some components. The experimental results show that this technique can increase diversity of population and converge more rapidly than other binary algorithms. In [13], a hybrid real-coded QEA is proposed by combing PSO, crossover and mutation. Simulation results show that it performs better in terms of ability to discover the global optimum and convergence speed. [14] uses quantum PSO principles to resolve the satisfiability problem. In [15], a quantum inspired PSO is applied to optimize of simultaneous recurrent neural networks and shows better performance than traditional methods.

From what has been discussed above, this paper proposed a variant of quantum evolutionary algorithm based on dynamic neighborhood topology(DNTQEA), In DNTQEA, the neighborhood of a quantum particle are not fixed but dynamically changed, and the learning mechanism of a quantum particle includes two partsthe global best experience of all quantum particles in populationand the best experiences of its all neighbors, which guide the final evolving direction of the quantum particle.

This paper is organized as follows. Section 2 describes the original QEA. In Section 3 the dynamic structure for QEA and the hybrid algorithm are given. In Section 4 the proposed algorithm is evaluated on some benchmark functions and finally the proposed algorithm is concluded in section 5.

## 2 Brief Description of Quantum Evolutionary Algorithm

Quantum evolutionary algorithm(QEA) combines quantum mechanism and basic evolutionary algorithm is a kind of probability search algorithm. Its essential characteristics are making full use of the superposition and coherence of quantum state. QEA adopts a new coding party quantum bit code; the concrete form can be described as:

$$\begin{bmatrix} \alpha_1 | & \alpha_2 | & \cdots | & \alpha_m \\ \beta_1 | & \beta_2 | & \cdots | & \beta_m \end{bmatrix} \tag{1}$$

In equation (1), $(\alpha_i, \beta i)^T (i = 1, 2, \cdots, m)$, represents a quantum bit and satisfies the following expression $|\alpha^2 + \beta^2| = 1$, m denotes the number of quantum bit; $q$ is called a quantum chromosome used to describe the problem in QEA.

The above representation method has the advantage that it is able to represent any superposition of states, so evolutionary computing with the Q-bit representation has a better characteristic of diversity than classical approaches, since it can represent superposition of states. As $\alpha_i^2$ or $\beta_i^2$ approaches to 1 or 0, the Q-bit chromosome converges to a single state and the property of diversity disappears gradually, and the algorithm converges.

The structure of QEA is described in Table 1 [3]:

**Table 1.** The details description of QEA

| **procedure QEA** | |
|---|---|
| **begin** | |
| 1. | $t \leftarrow 0$; |
| 2. | initialize $Q(t)$; |
| 3. | make $P(t)$ by observing $Q(t)$ states; |
| 4. | evaluate $P(t)$; |
| 5. | store the best solution among $P(t)$; |
| | **while**(not termination-condition) **do** |
| | **begin** |
| 6. | $t \leftarrow$ t+1; |
| 7. | make $P(t)$ by observing $Q(t-1)$ states; |
| 8. | evaluate $P(t)$; |
| 9. | update $Q(t)$ using quantum gates $U(t)$; |
| 10. | store the best solution among $P(t)$; |
| | **end** |
| **end** | |

Here, $Q(t)$ represents the population in $t$ generation; $P(t)$ denotes the set of binary solution of $t$ generation. When initializing the population, generally all the quantum bits in quantum chromosomes are initialized to $\sqrt{2}$ which means that all possible superposition states appear in the same probability. In step (3), $Q(t)$ generates $P(t)$ through the observation operation. This process can described as follows: randomly generating a random number between [0, 1], if it is greater than $\alpha_i^2$, the Q-bit value of corresponding binary solution is 1, otherwise its value is 0. In the step(9), QEA adopts the quantum revolving door $u(t)$ to update $Q(t)$, and mathematical formula can be defined as:

$$\begin{bmatrix} \alpha_i^{'} \\ \beta_i^{'} \end{bmatrix} = \begin{bmatrix} cos(\Delta\theta) & -sin(\Delta\theta) \\ sin(\Delta\theta) & cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \tag{2}$$

Here, $\Delta\theta$ is the rotation angle and controls the speed of convergence and determined by look table given in [3], which shows that these values for $\Delta\theta$ have better performance.

## 3 Dynamic Neighborhood Topology Based QEA

### 3.1 Dynamic Neighborhood Topology Structure and Updating Rules

Usually, the structure of the population in improved QEAs [9][10][11] is fixed and the neighbors of each individual are never changed in evolving process, this will cause each individual has fewer learning samples and greatly reduce the diversity of the population. So in order to keep each individual has opportunity to learn from more individuals, avoid trapping in local optimums and existing precocious phenomenon, this paper proposes a strategy based on fitness of each individual to form its neighbors dynamically. This makes the learning samples become diversity and promotes the individual converge to the global optimal position.

In DNTQEA, the neighbor selection rule of each individual is Euclidean distance among individuals. Based on this rule, the neighbors of current individual $i$ can be calculated as follows:

$$\begin{cases} D_i(t) = \{d_{ij}(t) | d_{ij}(t) = ||x_i(t) - x_j(t)||\}, j \neq i, j \in P_s; \\ neighbor_i(t) = arg(min(sort(D_i(t)), n)). \end{cases} \tag{3}$$

Where, $D_i(t)$ denotes the Euclidean distance set between the current individual $i$ and other individuals in population at $t$ generation; $P_s$ denotes the population size; $neighbor_i$ means the neighbors set of the current individual $i$ at $t$ generation; $n$ denotes the number of neighbors belong to the current individual $i$, its value usually is set to $1/4 \sim 1/3$ of the population size and in DNTQEA, $n = P_s/3$; $sort(A)$ means sorting elements in A from smallest to largest. When executing DNTQEA, if the individual fitness holds the same in continuous $T$ generation, its neighbors need to be reselected. After repeated experiments, when $T = 10$, the algorithm gains the best result, so in DNTQEA, parameter $T$ is set to 10.

The updating method of each individual in [2][3][9][10][11] can be summarized as that each individual only learning from the global best individual or the best one in its neighbors. This method may cause the searching process trapped in local optimums, while in particle swarm optimization(PSO)[15] algorithm, the evolving direction of each particle is decided by its history experience and the global experience. So in DNTQEA, the global best individual and the best individual of neighbors are used to guide the evolving direction of each individual. This updating rule can be described as follows:

$$\begin{cases} q_i^1 \longleftarrow Learn(q_i, g_{best}); \\ q_i^2 \longleftarrow Learn(q_i, n_{best}); \\ q_i^{'} \longleftarrow Random(q_i^1, q_i^2, p); \end{cases} \tag{4}$$

Here, $q_i$ and $q_i^{'}$ denote the $i$th original individual and the new individual after learning from the $g_{best}$ and $n_{best}$, respectively; $g_{best}$ means the global best individual in current population; $n_{best}$ denotes the best individual in neighbors of $q_i$;

$Learn(I_1, I_2)$ means $I_1$ is updated by predefined quantum gate with reference to $I_2$; $q_i^1$ and $q_i^2$ denote the learning results from $g_{best}$ and $n_{best}$; $Random(I_1, I_2, p)$ means to choose $I_1$ with probability $p$ as the final evolving individual.

### 3.2 Procedure of the Proposed DNTQEA

By combing the dynamic neighborhood topology and updating rule introduced in above subsection, the procedure of the proposed DNTQEA is described in Table 2.

**Table 2.** The details description of DNTQEA

```
Procedure DNTQEA
begin
    t = 0;
1.  initialize quantum population Q(t) with the size of P_s;
2.  make X(t) by observing the states of Q(t);
3.  evaluate X(t);
4.  for all binary solutions x_i^t do
      begin
5.      find neighborhood set N_i in X(t) by definition (3)
6.      find binary solution x with best fitness in N_i
7.      save x in B_i
      end
8.  save solution y with best fitness of X_t in g_best;
9.    while(not termination-condition) do
      begin
        t = t + 1;
10.     make X(t) by observing the states of Q(t − 1);
11.     evaluate X(t);
12.     update Q(t) based on B_i and g_best using Q-gates by rule (4);
13.     for all binary solutions x_i^t do
          begin
14.         find neighborhood set N_i in X(t) by definition (3)
15.         select binary solution x with best fitness in N_i
16.         if x is fitter than B_i save x in B_i
          end
17.     select solution y with best fitness of X(t)
18.     if y is fitter than g_best save y in g_best;
      end
end
```

The pseudo code of DNTQEA is described as below:

1. In initialization step, $[\alpha_{i,k}^0 \ \beta_{i,k}^0]^T$ in $q_i^0$ are initialized with $1/\sqrt{2}$, where $i = 1, 2, \cdots, P_s$ is the index of the individuals in the population, $k = 1, 2, \cdots, m$, and $m$ is the number of Q-bits in a individual. This initialization means that each Q-bit individual $q_i[0]$ represents the linear superposition of all possible states with equal probability.

2. This step makes a set of binary solutions $X(0) = \{x_i^0 | i = 1, 2, \cdots, P_s\}$ at generation $t = 0$ by observing $Q(0) = \{q_i^0 | i = 1, 2, \cdots, P_s\}$ states, where $X(t)$ at generation $t$ is a random solution of Q-bit population and $P_s$ is population size. Each binary solution, $x_i^0$ with length $m$, is formed by selecting each bit using the probability of Q-bit, either $|\alpha_{i,k}^0|^2$ or $|\beta_{i,k}^0|^2$ of $q_i^0$. The binary bit $x_{i,k}^t$ can be obtained from Q-bit $[\alpha_{i,k}^0 \ \ \beta_{i,k}^0]^T$ in following way:

$$x_{i,k}^t = \begin{cases} 0 \; if & random(0,1) < |\alpha_{i,k}^t|^2 \\ 1 & otherwise \end{cases} \tag{5}$$

Where $random(0,1)$ is a uniform random number generator.

3. Each binary solution $x_i^0 \in X(0)$ is evaluated to give some measure of its fitness.

4,5,6,7,8. In these steps the neighborhood set $N_i$ of all binary solutions $x_i^0$ in $X(0)$ is selected by rule 3, meanwhile, the best solution among $N_i$ is stored in $B_i$ and the best solution among $X(0)$ is saved in $g_{best}$.

9. The **while** loop is terminated when the maximum number of iterations is reached.

10. Observing the binary solutions $X(t)$ from $Q(t-1)$.

11. Evaluating the binary solutions $X(t)$.

12. The quantum individuals are updated using Q-gate based on $B_i$ and $g_{best}$ with updating rule 4.

13. The **for** loop is for all binary solutions $x_i^t (i = 1, 2, \cdots, P_s)$ in population.

14. Finding the neighbors of the binary solution $x_i^t$.

15. Selecting the best possible solution in $N_i$ and save it to $x$.

16. If $x$ is fitter than $B_i$ and then replace $B_i$ with $x$.

17. Finding the best possible solution in $X(t)$ and save it to $y$.

18. If $y$ is fitter than $g_{best}$ and then replace $g_{best}$ with $y$.

## 4 Simulations

The proposed DNTQEA is compared with the original version of QEA and FSQEA [11] which used Cellular structure and a functional population size for QEA. The experimental results are performed for several dimensions (m=50, 100, 250) of Knapsack Problem and 14 numerical benchmark functions. Similar settings with reference [11], the population size of all algorithms is set to 25; termination condition is set for a maximum of 1000 generations. Due to statistical nature of the optimization algorithms, all results are averaged over 30 runs. The parameter of QEA is set to $\Delta\theta = 0.01\pi$ and the parameters of FSQEA are set to the same value with reference [11].

Table 3 summarizes the experimental results on DNTQEA, FSQEA and QEA for Knapsack Problem and 14 benchmark functions (The results for some dimensions are not summarized in Table 3 because of small space of the paper). As it seen in Table 3, DNTQEA has the best results.

## 5 Conclusions

This paper proposes a variant of quantum evolutionary algorithm based on dynamic neighborhood topology(DNTQEA). In DNTQEA, the neighbors of a quantum particle are dynamically changed by Euclidean distance set between the current individual and other individuals in population. The learning mechanism of a quantum particle contains the global best experience of all quantum

**Table 3.** Experimental results on Knapsack Problem and 14 numerical benchmark functions for m=100 and m=250

| | m=100 | | | | | | m=250 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DNTQEA | | FSQEA | | QEA | | DNTQEA | | FSQEA | | QEA | |
| | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| KP1 | 590.28 | 0.73 | 562.13 | 0.92 | 546.82 | 11.29 | 1355.8 | 15.23 | 1252.6 | 17.40 | 1173 | 30.19 |
| KP2 | 438.54 | 1.14 | 418.15 | 1.94 | 406.91 | 4.39 | 1029.55 | 10.11 | 994.46 | 10.31 | 942.99 | 20.90 |
| f1 | 47126 | 1289.3 | 45889 | 1558.8 | 34437 | 3984.1 | 80081 | 2719 | 76292 | 2939 | 55844 | 5845.3 |
| f2 | -1097.3 | 66.49 | -1287.8 | 97.97 | -2096.3 | 199.45 | -5015 | 287.18 | -5118.9 | 304.75 | -6511 | 360.74 |
| f3 | -13.76 | 0.08 | -16.89 | 0.14 | -17.19 | 0.09 | -16.89 | 0.08 | -17.39 | 0.11 | -17.62 | 0.11 |
| f4 | -27.28 | 4.19 | -30.08 | 5.24 | -39.60 | 8.48 | -123.89 | 10.19 | -134.53 | 10.16 | -154.42 | 14.04 |
| f5 | -1.27e5 | 12459 | -1.43e5 | 14756 | -1.67e5 | 16807 | -5.41e5 | 20115 | -5.60e5 | 20743 | -6.05e5 | 43232 |
| f6 | -21459 | 2412.8 | -23017 | 2742.1 | -36949 | 4918.7 | -0.97e5 | 4873.2 | -1.09e5 | 5039 | -1.44e5 | 8269.3 |
| f7 | 38.22 | 2.27 | 31.17 | 2.53 | 22.04 | 2.46 | 55.81 | 3.28 | 52.24 | 4.39 | 39.43 | 3.02 |
| f8 | 54.11 | 1.21 | 50.57 | 1.75 | 38.19 | 3.24 | 97.89 | 4.89 | 93.51 | 5.31 | 73.50 | 5.02 |
| f9 | -2.198e5 | 21054 | -2.46e5 | 23972 | -4.55e5 | 68039 | -1.15e6 | 88756 | -1.26e6 | 93119 | -1.64e6 | 1.34e5 |
| f10 | -3.561 | 0.22 | -4.2161 | 0.29 | -5.21 | 0.68 | -5.45 | 0.28 | -5.95 | 0.30 | -6.52 | 0.46 |
| f11 | -167.01 | 5.45 | -169.84 | 7.76 | -176.72 | 3.78 | -178.24 | 1.56 | -189.69 | 2.35 | -192.03 | 1.36 |
| f12 | -1.19e7 | 2.21e6 | -1.33e7 | 2.53e6 | -2.56e7 | 9.20e6 | -2.19e8 | 2.01e7 | -2.44e8 | 2.45e7 | -3.09e8 | 3.59e7 |
| f13 | -46873 | 5104 | -49229 | 5885 | -1.08e5 | 35412 | -2.84e5 | 33298 | -3.1e5 | 35376 | -4.81e5 | 72311 |
| f14 | -0.074 | 0.057 | -0.098 | 0.065 | -1.29 | 1.26 | -7.29 | 1.74 | -7.53 | 1.94 | -20.38 | 6.19 |

particles in population and the best experiences of its all neighbors, which together guide the evolving direction. The performance of the proposed algorithm is tested on Knapsack Problem and 14 benchmark functions, and simulation results show that DNTQEA is better than other improved QEA and more suitable for solving combinatorial optimization problems problems.

The objective functions which are used here are f1: Schwefel 2.26 [16], f2: Rastrigin [16], f3: Ackley [16], f4: Griewank [16], f5: Penalized 1 [16], f6: Penalized 2 [16], f7: Michalewicz [17], f8: Goldberg [18], f9: Sphere Model [16], f10: Schwefel 2.22 [16], f11: Schwefel 2.21 [16], f12: Dejong [17], f13: Rosenbrock [18], and f14: Kennedy [18].

**Acknowledgements.** This Research is supported by Natural Science Foundation of China(No.60873058), Natural Science Foundation of Shandong Province (No.ZR2011FM001), Shandong Young Scientists Award Fund(NO.BS2013DX037), Shandong Soft Science Major Project(No.2010RKMA2005) and Natural Science Foundation of China(No.71071090).

## References

1. Narayanan, A., Moore, M.: Quantum-inspired Genetic Algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, pp. 61–66 (1996)
2. Han, K.H., Kim, J.H.: Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem. In: Proceedings of the 2000 Congress on Evolutionary Computation, Piscataway, vol. 2, pp. 1354–1360 (2000)
3. Han, K.H., Kim, J.H.: Quantum-inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Trans. Evolutionary Computation. 6(6), 580–593 (2000)

4. Han, K., Kim, J.: Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, He Gate, and Two-Phase Scheme. IEEE Transactions on Evolutionary Computation 8(2), 156–169 (2004)
5. Platel, M.D., Schliebs, S., Kasabov, N.: Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. IEEE Transactions on Evolutionary Computation 13(6), 1218–1232 (2009)
6. Hong, Y., Pen, K.: Optimal VAR Planning Considering Intermittent Wind Power Using Markov Model and Quantum Evolutionary Algorithm. IEEE Transactions on Power Delivery 25(4), 2987–2996 (2010)
7. Sinha, N., Hazarika, K.M., Paul, S., Shekhar, H., Karmakar, A.A.: Improved Real Quantum Evolutionary Algorithm for Optimum Economic Load Dispatch with Non-convex Loads. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) SEMCCO 2010. LNCS, vol. 6466, pp. 689–700. Springer, Heidelberg (2010)
8. Bryden, K.M., Daniel, A.A., Steven, C., Stephen, J.W.: Graph-Based Evolutionary Algorithms. IEEE Trans. Evol. Comput. 10(5), 550–567 (2006)
9. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. IEEE Proceedings of the Congress on Evolutionary Computation 2, 1671–1676 (2002)
10. Tayarani-N, M.H., Akbarzadeh-T, M.R.: A Sinusoid Size Ring Structure Quantum Evolutionary Algorithm. Proceeding of Cybernetics and Intelligent Systems, 1165–1169 (2008)
11. Najaran, T., Akbarzadeh, T., Mohammad, R.: Improvement of Quantum Evolutionary Algorithm with Functional Sized Population. Applications of Soft Computing 58, 389–398 (2009)
12. Zhao, J., Sun, J., Xu, W.B.: A binary quantum-behaved particle swarm optimization algorithm with cooperative approach. International Journal of Computer Science 10(1), 112–118 (2013)
13. Hossain, A.M., Hossain, K.M., Hashem, M.: A generalized hybrid real-coded quantum evolutionary algorithm based on particle swarm theory with arithmetic crossover. International Journal of Computer Science & Information Technology 2(4), 172–187 (2010)
14. Layeb, A.: A quantum inspried particle swarm algorithm for solving the maximum statisfiability problem. International Journal of Combinatorial Optimization Problems and Informatics 1(1), 13–23 (2010)
15. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
16. Zhong, W., Liu, J., Xue, M., Jiao, L.: A Multi-agent Genetic Algorithm for Global Numerical Optimization. IEEE Trans. Sys, Man and Cyber. 34, 1128–1141 (2004)
17. Khorsand, A.-R., Akbarzadeh-T, M.-R.: Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3055–3062 (2005)
18. Koumousis, V.K., Katsaras, C.P.: A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance. IEEE Trans. Evol. Comput. 10, 19–28 (2006)