2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Quantum-Inspired Evolutionary Algorithm with Linkage Learning

Bo Wang*, Hua Xu*, Yuan Yuan*
*State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
Email: wbthu@outlook.com, xuhua@tsinghua.edu.cn, yyxhdy@gmail.com

***Abstract*—The quantum-inspired evolutionary algorithm (QEA) uses several quantum computing principles to optimize problems on a classical computer. QEA possesses a number of quantum individuals, which are all probability vectors. They work well for linear problems but fail on problems with strong interactions among variables. Moreover, many optimization problems have multiple global optima. And because of the genetic drift, these problems are difficult for evolutionary algorithms to find all global optima. Local and global migration that QEA uses to synchronize different individuals prevent QEA from finding multiple optima. To overcome these difficulties, we proposed a quantum-inspired evolutionary algorithm with linkage learning (QEALL). QEALL uses a modified concept-guide operator based on low order statistics to learn linkage. We also replaced the migration procedure by a niching technology to prevent genetic drift, accordingly to find all global optima and to expedite convergence speed. The performance of QEALL was tested on a number of benchmarks including both unimodal and multimodal problems. Empirical evaluation suggests that the proposed algorithm is effective and efficient.**

I. INTRODUCTION

The quantum-inspired evolutionary algorithm (QEA) was firstly proposed by Han and Kim [1]. This algorithm was inspired by quantum computing and adopted some principles and concepts of quantum computing to optimize problems on a classical computer. Over the past decade, there has been considerable research in QEA [2]. But there are still some pending issues to be studied. Firstly, QEA uses a simple probabilistic model which can not reflect the high order relationship between variables. QEA employs quantum-bit individual (Q-individual) instead of other representations to constitute population. Each Q-individual is in fact equivalent to probability vector as used in some estimation of distribution algorithms (EDA) like PBIL [3] and cGA [4]. Researchers have demonstrated that QEA belongs to univariate EDAs [5], [6]. Secondly, QEA is unable to deal with problems with multimodal landscape, especially symmetric problems. QEA uses a migration method to exchange and share information between individuals, and this makes all individuals converge to the same solution.

For the first issue, we incorporated the concept-guided combination (cg-combination) operator into QEA framework. The cg-combination operator firstly proposed by Emmendorfer and Pozo [7] was originally used on probability vector. We firstly demonstrated that a Q-individual was in fact a probability vector. Then the cg-combination was applied to Q-individual under the assumption that each Q-individual stood for a cluster. By using the cg-combination, the proposed algorithm is able to learn linkage.

For the second issue, we removed the migration method that promotes genetic drift in QEA. Genetic drift, which is the inherent quality of the evolutionary algorithms, is the change in frequency of gene in population due to selection operation. The population will finally evolve into one of the basins randomly. This tendency leads evolutionary algorithms which have no particular technology for multimodal problems to converge to only one optimal solution.

Additionally, the existence of several global optima does not only weaken the effectiveness but also reduce the efficiency of evolutionary algorithms. Most evolutionary algorithms guide the exploration of the search space by assembling different parts of solutions. But in multimodal problems, especially symmetric problems, combining solutions from different basins without proper guidance often results in poor solutions. We addressed this issue with the introduction of a nearest replacement strategy that is similar with the idea of crowding method which is a classical niching technology.

The contribution of this paper is as follows: firstly, by incorporating QEA framework and cg-combination operator, we enable QEA to learn linkage quickly and efficiently. The experiment result of unimodal problems indicates that our proposed QEALL outperforms BOA. Secondly, we removed the migration method and employed niching technology, that is a nearest replacement strategy into QEA to prevent genetic drift, so as to find all the global optima and to accelerate the convergence speed. This is the first algorithm based on QEA framework that can solve multimodal problems to the best of our knowledge.

In the remainder of this paper, we briefly review previous studies relevant to this work in section II. In section III, the proposed algorithm is described completely. In section IV and section V, the test problems and experimental results of unimodal and multimodal are reported respectively. Finally, conclusions are in section VI.

II. PRELIMINARIES

This section is a concise introduction to the basic technology used in our proposed algorithm. The fundamental principles of QEA are introduced. The relationship of QEA and EDA is discussed and the concept-guided combination (cg-combination) operator is described. Niching methods are also described briefly.

A. Quantum-Inspired Evolutionary Algorithm

Quantum computation is the study of handling information using a quantum mechanical system. Narayanan and Moore [8] attempted to employ concepts and principles of quantum mechanics like interference crossover in the genetic algorithms for the first time. That work indicated the potential of incorporating the concepts and principles of quantum computing into evolutionary computation. Han and Kim [9] proposed the genetic quantum algorithm and the concept of quantum-bit individual (Q-individual), rotation gates was presented. Afterwards, the quantum-inspired evolutionary algorithm (QEA) was proposed [1]. QEA uses a migration method to exchange the result found by different Q-individual. Han and Kim [1], [9] used QEA to solve 0-1 knapsack problem compared with classical genetic algorithm and the results demonstrated the effectiveness and capabilities of the algorithm.

The major difference between classical evolutionary algorithm and QEA is the basic unit stored information adopted in them. Binary digit (bit) used in classical evolutionary algorithms can be one of the two states, '0' or '1'. But quantum-bit (qubit) of QEA can be a superposition of two states.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are complex numbers in quantum mechanic. The moduli of α and β, $|\alpha|$ and $|\beta|$ are real numbers and denote the probability of getting the corresponding state with the normalizing condition $|\alpha|^2 + |\beta|^2 = 1$. The state of a qubit can be considered a point or a vector on the unit circle in the first quadrant. To modify the qubit, quantum gate (Q-gate) was proposed. It is actually a rotate operator on a qubit and defined as

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \tag{2}$$

Q-gates change the α and β under normalizing condition as follows

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix} \tag{3}$$

The modified rotation gates named H_ϵ gate, were proposed by Han and Kim [10]. H_ϵ gates prevent Q-individual from converging to 0 or 1 from which case the Q-individual cannot escape the state by itself. Specifically, to prevent the qubit evolves to state $|0\rangle$ or $|1\rangle$, that is, $\alpha = 0, \beta = 1$ or $\alpha = 1, \beta = 0$, the α and β are confined in the interval $[\epsilon, \sqrt{1-\epsilon^2}]$ as shown in Fig.1 . Experiments on numerical problems and combinatorial problems demonstrated that the QEA using H_ϵ gate could get better results with respect to the best results found [10].

Q-individual consists of qubits, so each quantum individual can be written in the form as an array of qubits:

$$Q_j(t) = \left[\begin{array}{c|c|c|c} \alpha_1^t & \alpha_2^t & \dots & \alpha_d^t \\ \beta_1^t & \beta_2^t & \dots & \beta_d^t \end{array} \right] \tag{4}$$

Because the relation of α and β, a quantum individual is equivalent to a probability vector with $p_i = \beta_i^2$.

$$PV(t) = \begin{bmatrix} |\beta_1^t|^2 & |\beta_2^t|^2 & \dots & |\beta_d^t|^2 \end{bmatrix} \tag{5}$$

As we can see, each quantum individual is a low order probabilistic model. In quantum mechanic, when observe or measure the qubit, the qubit will "collapsed" to one of its eigenstate, that is 0 or 1 with probability α^2 and β^2 respectively. To get the binary code of a solution to the problem, each Q-individual should be observed. In QEA, the sampled individual from Q-individual is called collapsed state. The best result is stored as an attractor to guide the search.

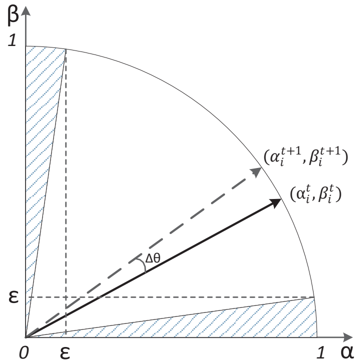


Fig. 1. The Q-gate in QEA Framework

B. QEA is a Multimodel Univariate EDA

In the evolutionary computation community, researchers have investigated a kind of algorithm called probabilistic model-building genetic algorithms (PMBGAs), or estimation of distribution algorithms (EDAs) [11]. EDAs evolve a probabilistic model and use it to guide the search instead of using a population as genetic algorithm does. Crossover operator and mutation operator are removed because they cannot preserve building blocks. According to the different types of probabilistic models adopted, the EDAs are divided into three categories: univariate EDA, bivariate EDA and multivariate EDA. Representative algorithms are population-based incremental learning (PBIL) [3], compact genetic algorithm (cGA) [4], extended compact genetic algorithm (ECGA) [12] and the Bayesian optimization algorithm (BOA) [13], to name a few.

Zhou and Sun [5] proposed a Quantum-inspired genetic algorithm with only one chromosome (SCQGA) and cast this algorithm into the framework of the EDAs. Platel et al. [6] integrated QEA into the class of EDAs in a more systematical way and regarded QEA as a multimodel EDA. As the probabilistic model used by QEA is Q-individual which is essentially a probability vector just like PBIL and cGA employed, the QEA belongs to univariate EDA.

Univariate EDA is simple and efficient, but an oversimple probabilistic model can not handle problems like trap-k where variables have complex relations effectively. The development of EDA accompanied by the increase of complexity of the probabilistic model. The merit and demerit of employing complicated probabilistic models are plain to see. The more complex probabilistic model can express the relationship between variable more precisely. However, complicated probabilistic model requires a significant amount of computing resources. To learn linkage with less computing resources, Emmendorfer and Pozo [7], [14]–[16] proposed an operator called concept-guided combination using ideas from information theory. This operator is computationally efficient while powerful in learning linkage because it uses a low order statistics. The main idea of the concept-guided combination operator is that we should choose the most informative parent for each gene when combining two PVs into a temp PV. This operator enable PVs which are simple probabilistic models to detect building blocks.

C. Niching Method

Multimodal problems are difficult for evolutionary algorithms. Niching method [17] is proposed for evolutionary computation to find multimodal optima. The sharing [18], clearing [19], crowding [20] are representative niching methods. Cluster analysis or clustering is also used to partition population into different niches [17]. There are works having integrated the EDAs and niching method in order to solve multimodal problems and improve effectiveness. Peña et al. [21] proposed the unsupervised estimation of Bayesian network algorithm (UEBNA). Emmendorfer and Pozo [7], [14]–[16] proposed φ-PBIL combining PBIL and clustering technique. The islands model is another method for multimodal problems. It is inspired by organic evolution. In islands model, the population is divided into multiple group which evolve independently for some generations and then exchange individuals among different groups. QEA framework is similar to islands model, but the global migration strategy promotes genetic drift and weakens the capability of finding multiple optima.

In addition to this, niching method could promote population diversity and at the same time increase the efficiency [22]. Chuang and Hsu [22] put forward a multivariate multimodel approach equip a heuristic mechanism to choose the number of models. Comparing to ECGA, the multivariate multimodel approach could obtain more global optima and reduce the number of generations to converge.

III. Quantum-Inspired Evolutionary Algorithm with Linkage Learning

In this section, our proposed algorithm, quantum-inspired evolutionary algorithm with linkage learning is described in detail.

A. Framework of QEALL

The outline of the algorithm is depicted as Fig.2. The quantum population is made up of units having three parts: a Q-individual $Q_i(t)$ and its corresponding collapsed state $C_i(t)$ and the attractor $A_i(t)$. There are two ways to generate a collapsed state, one is to directly sample from Q-individual, the other way is to cross-breed using cg-combination operator. The attractors store the best collapsed states ever found. The Q-gate modifies Q-individual towards the attractor based on the condition of the attractor and the collapsed state.

Algorithm 1 gives pseudocode of the QEALL and the details of the procedure are described as follows:

Lines 1-5 are initialization. Firstly, α and β of the Q-bits in Q-individual are initialized with $1/\sqrt{2}$ for all. Secondly, it would observe the state of Q to get the collapsed state C, then evaluate C. Attractors are copies of the corresponding collapsed state for initialization. Finally, the matrix of information measure W is initialized to a zero matrix, because that all Q-bits are same and that means they have the same amount of information.

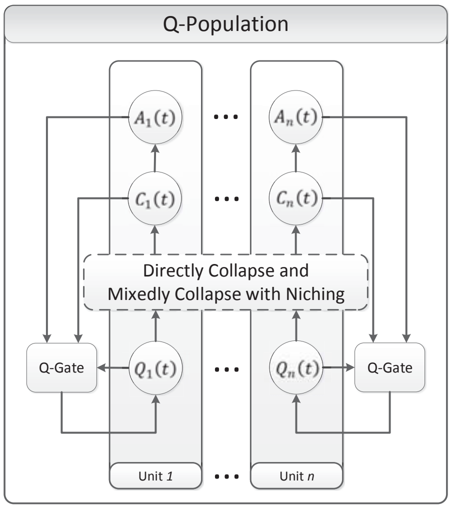


Fig. 2. Outline of Quantum-Inspired Evolutionary Algorithm with Linkage Learning.

Lines 6-16 are the main loop. The termination condition can be set to be the maximum iterations, the number of convergent Q-individuals or other criterion defined by user.

Algorithm 1 QEA with Linkage Learning

1: initialize $Q \equiv (Q_1, Q_2, \cdots, Q_N)$
2: make $C \equiv (C_1, C_2, \cdots, C_N)$ by observing the state of Q
3: evaluate C
4: $A \leftarrow C$
5: compute $W = (w_{i,j}) = (0)$
6: **while** not termination-condition **do**
7:     generate(Q, C), see details from Algorithm 2
8:     **for** $i = 1, 2, \ldots, N$ **do**
9:         **if** $f(A_i)$ better than or equal to $f(C_i)$ **then**
10:             updateModel (Q_i, C_i, A_i)
11:         **else**
12:             $A_i \leftarrow C_i$
13:         **end if**
14:     **end for**
15:     compute $W = (w_{i,j})$ using Equation (12)
16: **end while**

17: **procedure** UPDATEMODEL(Q_i, C_i, A_i)
18:     **for** $j = 1, 2, \ldots, d$ **do**
19:         **if** $A_i^j \neq C_i^j$ **then**
20:             **if** $A_i^j = 1$ **then**
21:                 $Q_i^j \leftarrow$ rotate $\Delta\theta$ using Equation (3)
22:             **else**
23:                 $Q_i^j \leftarrow$ rotate $-\Delta\theta$ using Equation (3)
24:             **end if**
25:         **end if**
26:     **end for**
27: **end procedure**

Line 7 is the heart of the algorithm as shown in Algorithm 2. The generation of C in the light of Q includes two parts: $N \times p_c$ collapsed states were created by cg-combination operator and the rest $N \times (1 - p_c)$ collapsed states were directly created from observing the corresponding Q-individual. When a collapsed state was generated by observing the temporary Q-individual assembled by cg-combination operator from two Q-individuals selected randomly, it would replace the more similar one of two collapsed state corresponding . By using the cg-combination operator, the QEALL is able to learn linkage and identify building blocks. The nearest replacement method also belongs to this procedure. We will explain the details of the generation procedure of collapsed states C in section III-B.

Lines 8-14 are the loop of the update process. For each unit, either the probabilistic model, that is the Q-individual Q_i or the attractor A_i will be updated based on the fitness of A_i and C_i. The fitness of A_i and C_i are denoted by $f(A_i)$ and $f(C_i)$. If $f(A_i)$ is better than or equal to $f(C_i)$, that means the attractor is better than the collapsed state, we should let the attractor remain unchanged and use it to guide the Q-individual by means of Q-gate. Otherwise, we should replace the attractor with the collapsed state. The model updating procedure is shown in the subprocedure indicated by lines 17-27, it gives the concrete steps about how to update the Q-individual. The A_i^j and C_i^j are the j-th bit of the attractor A_i and collapsed state C_i, respectively.

Line 15 expresses the recalculation of information measure matrix W. The element of W, w_{ij}, is calculated based on Equation (12) and will be described in the next subsection.

B. Generative Process of Collapsed States

The generative process of collapsed states is a key step of the QEALL. The shortcoming of QEA is that the Q-individual is too simple for complex problem and cannot reveal the relation of variables. The cg-combination operator is designed for probability vector and it gives PV the capacity of learning linkage. This operator recombines two PVs into a new PV by selecting the most informative parts. We introduce the cg-combination operator into the QEA framework to give QEA the ability of learning linkage. The pseudocode of the generative process of a collapsed state C is shown as Algorithm 2 and the detail of the procedure is described as follows.

Algorithm 2 Generating Procedure

```
1:  procedure GENERATE(Q, C)
2:      for j = 1, 2, ..., N × p_c do
3:          Q_r, Q_s ← select two Q-individuals randomly
4:          Q_temp ← create using concept-guided combination
5:          C_temp ← make by observing the state of Q_temp
6:          if Dist_H(C_r, C_temp) < Dist_H(C_s, C_temp) then
7:              C_r ← C_temp
8:          else
9:              C_s ← C_temp
10:         end if
11:     end for
12:     make the rest of C by observing corresponding Q
13: end procedure
```

1) Details of the Generative Process: In QEALL, there are two ways to generate the collapsed states: one is to use the concept-guided combination operator , the other way is by observing the corresponding Q-individual directly. Parameter p_c in line 2 determines the ratio between mixedly and directly collapsed states in population. $N \times p_c$ is the number of collapsed states that should be created using cg-combination operator. Because a quantum individual is equivalent to a probability vector, the concept-guided combination operator can be easily modified for the use of quantum individuals. Define

$$h_j = - \sum_{q \in \{0,1\}} p_{j,q} \log(p_{j,q}) \tag{6}$$

be the entropy h_j of the distribution of gene j, where

$$p_{j,0} = \frac{1}{N} \sum_k \alpha_{k,j}^2 \tag{7}$$

$$p_{j,1} = \frac{1}{N} \sum_k \beta_{k,j}^2 \tag{8}$$

In original cg-combination operator, the $p_{j,q}$ is the proportion of individuals possessing the value q for gene j in the whole population. The $[\alpha_{k,j}, \beta_{k,j}]$ is the j-th qubit of individual k. Here, we assume that the each Q-individual is a cluster processing only one virtual solution. Under such condition, the number of virtual population is N. Let

$$h'_{i,j} = - \sum_{q \in \{0,1\}} p'_{i,j,q} \log(p'_{i,j,q}) \tag{9}$$

be the entropy of the distribution of gene j without the i-th Q-individual, where

$$p'_{i,j,0} = \frac{\sum_{k \in \{1,2,\ldots,N\} \setminus \{i\}} \alpha_{k,j}^2}{N-1} \tag{10}$$

$$p'_{i,j,1} = \frac{\sum_{k \in \{1,2,\ldots,N\} \setminus \{i\}} \beta_{k,j}^2}{N-1} \tag{11}$$

The difference of h_j and $h'_{i,j}$

$$w_{i,j} = h_j - h'_{i,j} \tag{12}$$

indicates the increase of entropy of gene j after the i-th quantum individual was taken into account than that without the i-th quantum individual. When recombining two Q-individuals (Q_r and Q_s for example) into a temporary Q-individual, each qubit q_j of the new Q-individuals is defined as

$$q_j = \begin{cases} q_{r,j} & \text{if } w_{r,j} > w_{s,j} \\ q_{s,j} & \text{otherwise} \end{cases} \tag{13}$$

where $q_{r,j}$ and $q_{s,j}$ are the j-th qubit of the corresponding Q_r and Q_s. At each time, we select two Q-individual untapped randomly and recombine them into a temporary Q-individual based on the informative measure matrix, refer to Equation (13). Then observe the temporary Q-individual and we will get a temporary collapsed state. The new collapsed state is associated with two Q-individuals. Here we have two replacement strategy, one is random replacement and the other is nearest replacement. As the idea lay behind crowding method as mentioned in section II-C, we would let the new collapsed state replace the more similar one. So the similarity should

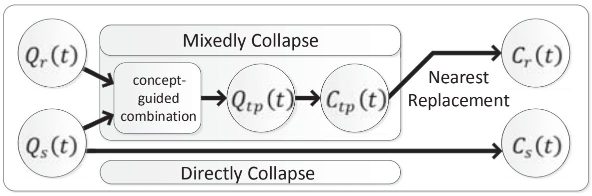


Fig. 3. A simple example with only two Q-individuals. Assuming $p_c = 0.5$ and the number of collapsed state generated by concept-guided combination is $2 \times 0.5 = 1$. Q_{tp} is recombination of Q_r and Q_s using concept-guided combination. C_{tp} is the result of observing Q_{tp}. By calculating similarity of C_{tp} to C_r and C_s, replace the more similar one (assuming C_{tp} and C_r are more similar) with C_{tp}. The C_s remind unchanged and should be updated by observing the corresponding Q_s.

be computed. We use Hamming distance when dealing with binary coding.

$$Dist_H(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{m} |x_{1i} - x_{2i}| \tag{14}$$

By removing the migration method in QEA, our proposed algorithm was able to find more than one optima. By introducing the nearest replacement procedure besides the cg-combination operator, algorithm achieves a speedup. Repeat this for $N \times p_c$ times. But there are still $N \times (1 - p_c)$ units in which the collapsed states had not been updated in this iteration. These unchanged collapsed states will be updated by observing their corresponding Q-individuals directly as original QEA. A simple example of the process of generating C in the two approaches is illustrated in the Fig.3.

2) Effect of the Generative Process: The generative process of collapsed states in QEALL has three main effects. The first effect is that the introduction of cg-operator gives the Q-individual the ability to learn linkage. The second effect is that this process acts as intermediary of information interchange. In QEALL, the migration procedure was removed and Q-individuals can still exchange information by the generative process and the replacement procedure. The last effect is that QEALL achieves a speed-up due to nearest replacement. This is in fact a niching technology.

IV. Performance Experiments on Unimodal Problems

This section presents experimental results of QEALL comparing with the BOA that is the well-known multivariate EDA on unimodal problems. The effectiveness of niching method was also tested by a comparative trial employing a QEALL using random strategy of replacing. This section includes two parts: Firstly, the test problems, concatenated trap-5 and overlapping concatenated trap-5, are introduced briefly. Secondly, it describes the evaluation setup for QEALL and BOA. Then the performance of the QEALL and BOA on these optimization problems are reported and compared.

A. Problems

The problems we selected for testing the performance of QEALL are concatenated trap-5 [13] and overlapping trap-5 problems [23]. It is hard for evolutionary algorithms like classical genetic algorithm to solve these problems because the variables in these problems are dependent on each other. EDAs can solve this kind of problems effectively. In overlapping trap-5, building blocks are overlapping. This structure is even awkward for BOA which uses a complex Bayesian network.

1) Concatenated Trap-5: The concatenated trap-5 is an additively decomposable function. It is an aggregation of some trap-5 functions and the objective function is defined as:

$$f_{trap\_5}(\mathbf{z}) = \sum_{i=1}^{\frac{n}{5}} trap_5(z_{5i-4}, z_{5i-3}, ..., z_{5i}) \tag{15}$$

where

$$trap_5(\mathbf{z}) = \begin{cases} 5 & \text{if } u = 5 \\ 4 - u & \text{otherwise} \end{cases} \tag{16}$$

The objective is maximization and the global optima is $\mathbf{z} = (1, 1, \ldots, 1)$ with fitness n. Instances of size 30, 60 and 90 were considered and denoted as P_{trap30}, P_{trap60} and P_{trap90} respectively.

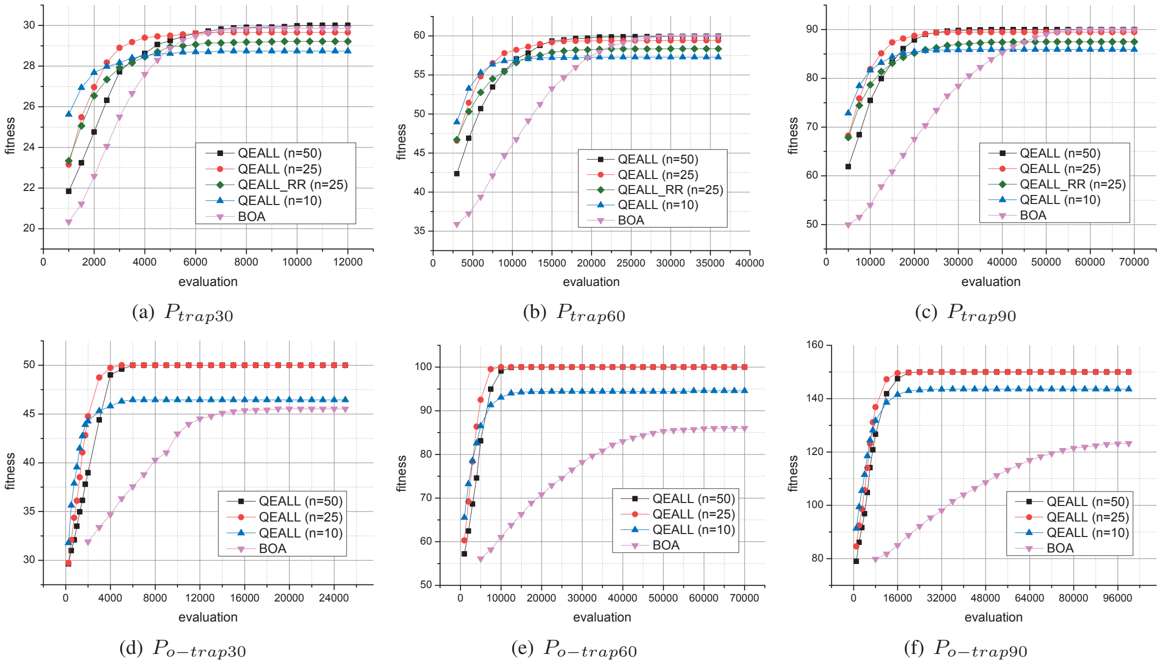
2) Overlapping Trap-5: The overlapping Trap-5 is an overlapping additive decomposable function. Building blocks shares their variables with other building blocks bordering on. Each building block is a trap-5 function. Two variables in the left an right are shared with building blocks on both sides. The objective is maximization and the global optima is $\mathbf{z} = (1, 1, \ldots, 1)$ with fitness $5n/3$. Instances of size 30, 60 and 90 were considered, denoted as $P_{o-trap30}$, $P_{o-trap60}$ and $P_{o-trap90}$ respectively.

B. Evaluation Setup and Results

QEALL has four main parameters, they are population size N, lower boundary of Q-gate ϵ, rotation angle $\Delta\theta$ and the ratio of two generating approach p_c. In all the experiments in this section, ϵ, $\Delta\theta$ and p_c are set to be 0.1, 0.03π and 0.5 respectively. The population size N is set to be 10, 25 and 50 to test the influence of population size. To measure the performance of QEALL and compare it with BOA, all the best solution found were averaged over 50 independent runs. BOA needs a relatively larger population size to get its best performance. If the population size is too small, BOA tends to converge to local optima. Otherwise too large population size result in overlarge evaluations. In order to conduct a fair comparison, the termination condition is set to be the max number of evaluations instead of max number of generations.

On P_{trap30}, P_{trap60} and P_{trap90}, the population size of BOA are set to be 1000, 3000 and 5000 respectively. Fig.4 (a) (b) (c) show the experiment results. A successful run means the algorithm had found the global optima. For three concatenated Trap-5 instances, BOA can find global optima in most of the runs, the success rates are $43/50$, $49/50$ and $48/50$. Performance of QEALL with population size 50 slightly exceeded BOA, the success rates are $50/50$, $49/50$ and $49/50$. Furthermore, QEALL converged faster than BOA. As the size of the problem gets larger, QEALL's advantage of efficiency increases. QEALL with population size 10 and 25 can not find global optima in every run. But they also have an advantage of faster convergence speed.

To verify the effect of employing the nearest replacement, we removed the comparison step in lines 6-10 of Algorithm



(a) P_{trap30} (b) P_{trap60} (c) P_{trap90}

(d) $P_{o-trap30}$ (e) $P_{o-trap60}$ (f) $P_{o-trap90}$

Fig. 4. Experimental Results on Concatenated Trap-5 and Overlapping trap-5 Problems

2 and replaced it with a random replacement. The random replacement means when a new collapsed state was generated by the cg-combination operator, it would replace one of two related collapsed states randomly. We tested three modified QEALL (denoted as QEALL_RR) with population size 25 on P_{trap30}, P_{trap60} and P_{trap90}. Fig.4 (a) (b) (c) show that the introduction of niching technology brought improvement in quality and efficiency under the same conditions.

On $P_{o-trap30}$, $P_{o-trap60}$ and $P_{o-trap90}$, we set the population size of BOA to be 2000, 5000 and 8000. The parameters of QEALL were same as above. Experimental results show that QEALL can handle overlapping problems. The BOA, although employing multivariate probabilistic model, cannot deal with this kind of problem effectively. Fig.4 (d) (e) (f) show that the BOA can not converge to global optima in all runs. For the three overlapping Trap-5 problems, the success rates of BOA are $21/50$, $2/50$ and $0/50$ respectively. It is beneficial for BOA if we use a larger population. Meanwhile, the number of evaluations would increase rapidly. QEALL of 25 and 50 population can find the global optima for all the tests. Although not all QEALL of which the population was 10 can find global optima, it outperforms BOA on average in speed and quality.

V. Performance Experiments on Multimodal Problems

This section presents the dynamic of QEALL and QEA on twomax problems and also reports the experimental results of QEALL comparing with φ-PBIL and UEBNA for some graph bisection problems.

A. Problems

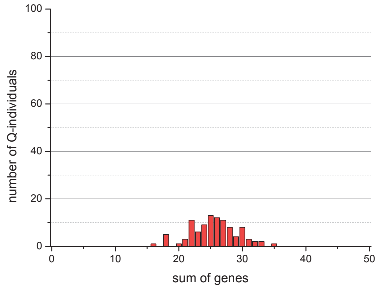
1) Twomax: The twomax problem is multimodal optimization problem whose two global optima are symmetrical. The search space is $\{0,1\}^n$, where n is the dimension of the problem. The objective function is as follows:

$$f_{twomax}(\mathbf{z}) = \left| \frac{n}{2} - \sum_{i=1}^{n} z_i \right| \tag{17}$$

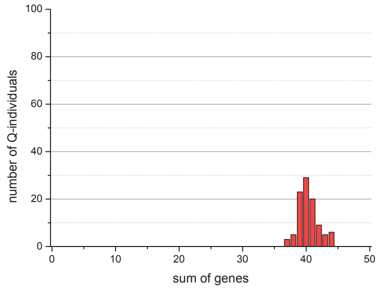
The objective is maximization and there exists two global optima. One is $\mathbf{z}_0 = (0, 0, \ldots, 0)$ and the other one is $\mathbf{z}_1 = (1, 1, \ldots, 1)$ with fitness equal to $n/2$. An instance of twomax problem of $n = 50$ was used to show the dynamic of QEA and QEALL. It was denoted by $P_{twomax50}$.

2) Graph bisection problems: Graph bisection problem, also named as bipartitioning problem, aims to split a given graph into two equally sized subsets each of which is called a partition. The objective is to find a graph partitioning in which the number of edges whose adjacent vertices are located in different partitions is minimal [24]. These edges are named cut edges. In order to transform the objective be maximization. The fitness of a given partitioning is set to be the difference of the number of nodes and the cut edges. A solution of the problem is encoded as a vector whose length equals to n. The i-th gene of the vector represents the label of the i-th node. The node numbers in both partition should be equal, so a lot of solutions obtained are unfeasible. Hence a randomized repair operator that inverts the gene in the majority until a solution is obtained was used.

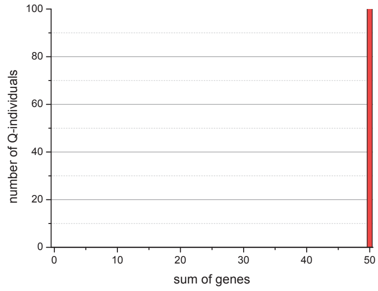
Three categories of graph bisection problem in [21] were used. The first is grid-like graphs. Three instances with $n = 16, 36, 64$ were involved and denoted by P_{grid16}, P_{grid36}, P_{grid64}. The rest instances are graphs composed of seven node subgraphs and named caterpillar graphs. One part contains three instances each of which composed of 4, 6 and 8 subgraphs connected in line. These three instances were denoted



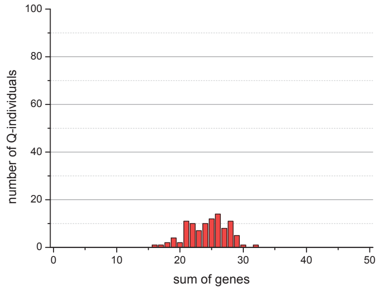
(a) QEA (generation 0)



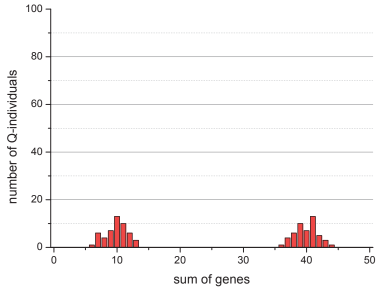
(b) QEA (generation 80)



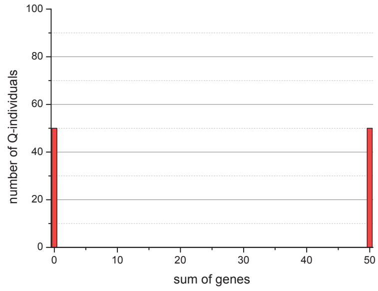
(c) QEA (generation 300)



(d) QEALL (generation 0)



(e) QEALL (generation 80)



(f) QEALL (generation 300)

Fig. 5. Dynamic of QEA and QEALL on $P_{twomax50}$.

by P_{cat28}, P_{cat42}, P_{cat56}. All the six instances mentioned above possess 2 global optima. The last category has two instances in which 4 and 6 subgraphs connected like a ring. These instances are denoted by P_{ring28} and P_{ring42} with the number of global optima 4 and 6.

B. Evaluation Setup and Results

The QEA uses a global migration would set all Q-individual to be the same value and this strategy aggravates genetic drift. In QEALL, the migration process was removed. Q-individuals exchange messages with each others by the niching method. Solutions from different basins will be preserved. Parameter of QEA an QEALL are same for N, ϵ and $\Delta\theta$ and they were 100, 0.01 and 0.015π. In QEA, the local immigration period was 1 and the global immigration period was 100. For QEALL, the probability of mixedly collapse p_c was 0.5.

For grid-like graphs and caterpillar graphs whose number of global optima is 2. The ϵ, $\Delta\theta$ and p_c are set to be 0.1, 0.015π and 0.5 respectively. The termination condition is set to be maximum generations or the attractors are remained unchanged for a certain number of generations. QEALL will stop if its attractors remained unchanged for 200 generations or the max generations, that is 2000, is reached. Analogously, For two ring-like instances, the $\Delta\theta$ is 0.02π and QEALL will stop when attractors remained unchanged for 50 generations or max generations reach 2000. Each instance was tested on QEALL for 10 independent runs. This is conformance to specifications of the experiments settings of UEBNA [21] and φ-PBIL [16] both of which had been tested on graph bisection problems for 10 times.

The dynamic of QEA and QEALL on $P_{twomax50}$ is shown on Fig.5. QEALL can preserve solutions in different basins but the QEA can not. Results of graph bisection problems are shown in Table I where the data for UEBNA and φ-PBIL were extracted from [16], [21]. Population size is indicated in the table. In all instances, the QEALL could find all global optima. Except for P_{ring42}, it was shown that QEALL outperforms UEBNA and φ-PBIL on all other problems. The number of evaluations is less than UEBNA and φ-PBIL. Experimental results show QEALL can solve problems having multimodal landscape effectively.

VI. Conclusion

QEA is a simple and effective algorithm and has been investigated widely. But algorithms based on QEA framework are limited by its oversimple model and cannot effectively address problem with complex links. Besides, though QEA uses more than one Q-individual and adopts the migration procedure which is similar to island modal in niching method, the global migration promotes genetic drift which deprives its ability to find multiple optima on problems that have a multimodal landscape. To solve both of these problems, we proposed QEALL equipped cg-combination operator and niching technology.

QEALL is in the framework of QEA and uses Q-individuals. In QEALL, we introduced the cg-combination operator of φ-PBIL into our algorithms. The cg-combination operator can select the most informative parts from two parent Q-individuals to form a new Q-individual. Building blocks will be detected and reserved so that QEALL can learn linkage. The migration procedure was removed to make QEA have the ability to conserve multiple optima simultaneously. Additionally, we added a nearest replacement method for acceleration.

The QEALL was tested on both unimodal problems and multimodal problems. We used concatenated trap-5 and over-

TABLE I. QEALL ON GRAPH BISECTION PROBLEMS

Problem	Algorithm	Optima ± sd	Eval ± sd
P_{grid16} (2 optima)	UEBNA $K = 4$	2.0 ± 0.0	51400±2366
	φ-PBIL $K = 5$	2.0 ± 0.0	10126±606
	QEALL $N = 10$	2.0 ± 0.0	**4578±934**
P_{grid36} (2 optima)	UEBNA $K = 2$	2.0 ± 0.0	85600±8462
	φ-PBIL $K = 5$	2.0 ± 0.0	28963±10754
	QEALL $N = 16$	2.0 ± 0.0	**16048±3998**
P_{grid64} (2 optima)	UEBNA $K = 4$	2.0 ± 0.0	124900±3479
	φ-PBIL $K = 10$	2.0 ± 0.0	64245±10999
	QEALL $N = 24$	2.0 ± 0.0	**42655±5714**
P_{cat28} (2 optima)	UEBNA $K = 2$	2.0 ± 0.0	57100±2846
	φ-PBIL $K = 5$	2.0 ± 0.0	14311±1299
	QEALL $N = 10$	2.0 ± 0.0	**6370±1683**
P_{cat42} (2 optima)	UEBNA $K = 2$	2.0 ± 0.0	73900±1449
	φ-PBIL $K = 10$	2.0 ± 0.0	29714±3644
	QEALL $N = 16$	2.0 ± 0.0	**14856±5482**
P_{cat56} (2 optima)	UEBNA $K = 4$	2.0 ± 0.0	96400±2366
	φ-PBIL $K = 10$	2.0 ± 0.0	46151±4362
	QEALL $N = 24$	2.0 ± 0.0	**33425±12564**
P_{ring28} (4 optima)	UEBNA $K = 2$	4.0 ± 0.0	54700±949
	φ-PBIL $K = 5$	4.0 ± 0.0	12694±853
	QEALL $N = 24$	4.0 ± 0.0	**9804±2486**
P_{ring42} (6 optima)	UEBNA $K = 6$	5.9 ± 0.3	75700±3302
	φ-PBIL $K = 15$	6.0 ± 0.0	**32361±1513**
	QEALL $N = 50$	6.0 ± 0.0	36925±5107

Note: Data for UEBNA extracted from [21], data for φ-PBIL extracted from [16].

lapping trap-5 problems. These kind of problems are hard for evolutionary algorithms. Experimental results suggested that QEALL precede BOA. This implied that QEALL can learn linkage quickly and effectively. A contrast experiment was also conducted to verify effects of introduction of the nearest replacement strategy. The nearest replacement would improve quality and speed. QEA can also handle overlapping trap-5 problems if it has enough Q-individuals while BOA can not. The QEA and QEALL were conducted on the twomax problem to show their dynamic when confronts multimodal problems. QEALL were test graph bisection problems. The results show that the QEALL has the ability to solve multimodal problems efficiently.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (Grant No: 61175110), National Basic Research Program of China (973 Program) (Grant No: 2012CB316305), National S&T Major Projects of China (Grant No: 2011ZX02101-004) and National Banking Information Technology Risk Management Projects of China.

REFERENCES

[1] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 6, pp. 580–593, December 2002.

[2] G. Zhang, "Quantum-inspired evolutionary algorithms: a survey and empirical study," *Journal of Heuristics*, vol. 17, no. 3, pp. 303–351, June 2011.

[3] S. Baluja, "Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning," DTIC Document, Tech. Rep., 1994.

[4] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 287–297, 1999.

[5] S. Zhou and Z. Sun, "A new approach belonging to EDAs: quantum-inspired genetic algorithm with only one chromosome," in *Advances in Natural Computation*. Springer, 2005, pp. 141–150.

[6] M. D. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: A multimodel EDA," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 6, pp. 1218–1232, December 2009.

[7] L. R. Emmendorfer and A. T. R. Pozo, "An empirical evaluation of linkage learning strategies for multimodal optimization," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 326–333.

[8] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Evolutionary Computation, 1996. Proceedings of IEEE International Conference on*. IEEE, 1996, pp. 61–66.

[9] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 2. IEEE, 2000, pp. 1354–1360.

[10] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithms with a new termination criterion, H_ϵ gate, and two-phase scheme," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 2, pp. 156–169, 2004.

[11] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions i. binary parameters," in *Parallel Problem Solving from Nature ł PPSN IV*, ser. Lecture Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 1996, vol. 1141, pp. 178–187.

[12] G. Harik, "Linkage learning via probabilistic modeling in the ECGA," *Urbana*, vol. 51, no. 61, p. 801, 1999.

[13] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, "Linkage problem, distribution estimation, and Bayesian networks," *Evolutionary computation*, vol. 8, no. 3, pp. 311–340, 2000.

[14] L. R. Emmendorfer and A. T. R. Pozo, "An incremental approach for niching and building block detection via clustering," in *Intelligent Systems Design and Applications, 2007. ISDA 2007. Seventh International Conference on*. IEEE, 2007, pp. 303–308.

[15] L. Emmendorfer and A. Pozo, "A clustering-based approach for linkage learning applied to multimodal optimization," in *Linkage in Evolutionary Computation*. Springer, 2008, pp. 225–248.

[16] L. R. Emmendorfer and A. T. R. Pozo, "Effective linkage learning using low-order statistics and clustering," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 6, pp. 1233–1246, 2009.

[17] O. M. Shir, "Niching in evolutionary algorithms," in *Handbook of Natural Computing*. Springer, 2012, pp. 1035–1069.

[18] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings of the Second International Conference on Genetic Algorithms and their application*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987, pp. 41–49.

[19] A. Petrowski, "A clearing procedure as a niching method for genetic algorithms," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 1996, pp. 798–803.

[20] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, Ann Arbor, MI, USA, 1975, aAI7609381.

[21] J. M. Peña, J. A. Lozano, and P. Larrañaga, "Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of Bayesian networks," *Evolutionary Computation*, vol. 13, no. 1, pp. 43–66, 2005.

[22] C.-Y. Chuang and W.-L. Hsu, "Multivariate multi-model approach for globally multimodal problems," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 311–318.

[23] T.-L. Yu, K. Sastry, and D. E. Goldberg, "Linkage learning, overlapping building blocks, and systematic strategy for scalable recombination," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, ser. GECCO '05. New York, NY, USA: ACM, 2005, pp. 1217–1224.

[24] J. Kim, I. Hwang, Y.-H. Kim, and B.-R. Moon, "Genetic approaches for graph partitioning: a survey," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 473–480.