

A Memory-Storable Quantum-Inspired Evolutionary Algorithm for Network Coding Resource Minimization

Yuefeng Ji and Huanlai Xing
Beijing University of Posts & Telecommunications,
China

1. Introduction

Network coding technology is a new communication paradigm that is superior to traditional routing in many aspects, especially in terms of the ability of increasing multicast throughput (Ahlsvede et al., 2000; Li et al., 2003). Traditional routing adopts *store-and-forward* data forwarding scheme at every intermediate node that simply replicates and forwards the incoming data to downstream nodes. However, the maximum throughput of a multicast scenario could not be often achieved under such a scheme (Ahlsvede et al., 2000; Li et al., 2003). With *code-and-forward* data forwarding scheme at network layer, network coding allows arbitrary intermediate node to combine (or code) the data received from different incoming links and output the coded information if necessary, being able to obtain a multicast throughput that is maximized according to the MAX-FLOW MIN-CUT theorem (Li et al., 2003).

Fig. 1 shows why network coding performs better than traditional routing in terms of the maximum multicast throughput they achieve. Fig.1(a) shows a network with source s and sinks y, z . Each direct link has a capacity of 1 bit per time unit. Source s expects to send two bits, a and b , to y and z . According to the MAX-FLOW MIN-CUT theorem, the min cut C_{min} between s and $\{y, z\}$ is 2 bits per time unit, which means the maximum multicast throughput from s to y and z should be 2 bits per time unit. However, if traditional routing is adopted, the multicast throughput is 1.5 bits information per time unit since link $w \rightarrow x$ could only forward 1 bit, a or b , to x , and thus y and z can not simultaneously receive two bits, a and b , as indicated in Fig.1(b). In Fig.1(c), if the intermediate node w is allowed to combine the 2 bits, a and b , it receives from t and u respectively to 1 bit $a \oplus b$ (here, symbol \oplus is Exclusive-OR operation) and output $a \oplus b$ to x , y and z are both able to obtain $\{a, a \oplus b\}$ and $\{b, a \oplus b\}$, which means two bits information is available at both y and z . Meanwhile, y and z can use $\{a, a \oplus b\}$ and $\{b, a \oplus b\}$ to decode b and a by calculate $a \oplus (a \oplus b)$ and $b \oplus (a \oplus b)$ respectively.

To the best of our knowledge, most of the network-coding-related research works suppose that coding operation should be implemented at all coding-possible intermediate nodes. However, to achieve a desired throughput, coding operation may only be necessarily performed at a subset of those nodes (Kim et al., 2006; 2007a; 2007b). In Fig.2, there are two network coding schemes that could both achieve the maximum multicast throughput. Network coding scheme A adopts all coding-possible nodes, m and n , as shown in Fig.2(a).

Nevertheless, the same throughput is also obtained by network coding scheme B where one of the two coding-possible nodes, m , is required to perform coding operation (see Fig.2(b)). Since coding operation consumes computational time and increases data processing complexity, it is of vital importance to minimize the amount of coding operations required. Unfortunately, such problem is NP-Hard (Kim et al., 2006; 2007a).

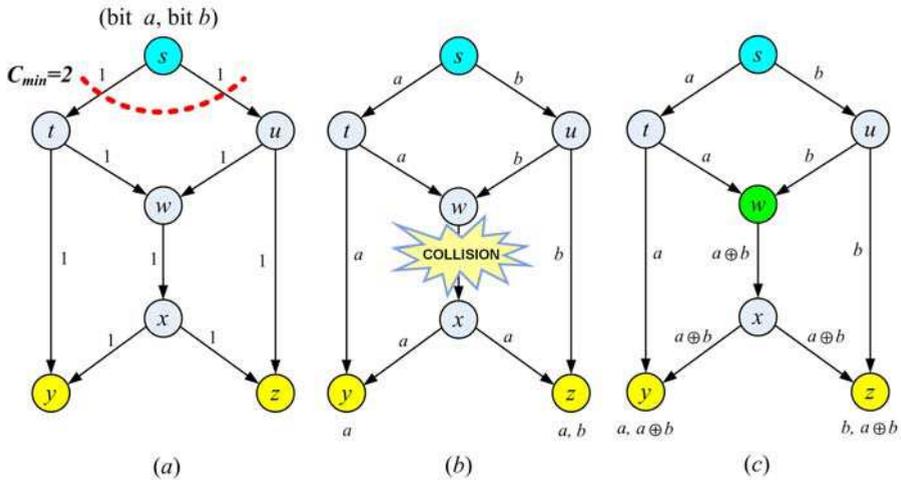


Fig. 1. Traditional routing vs. network coding. (a) A network topology with maximum multicast throughput of 2 bits per time unit. (b) Traditional routing scheme. (c) Network coding scheme.

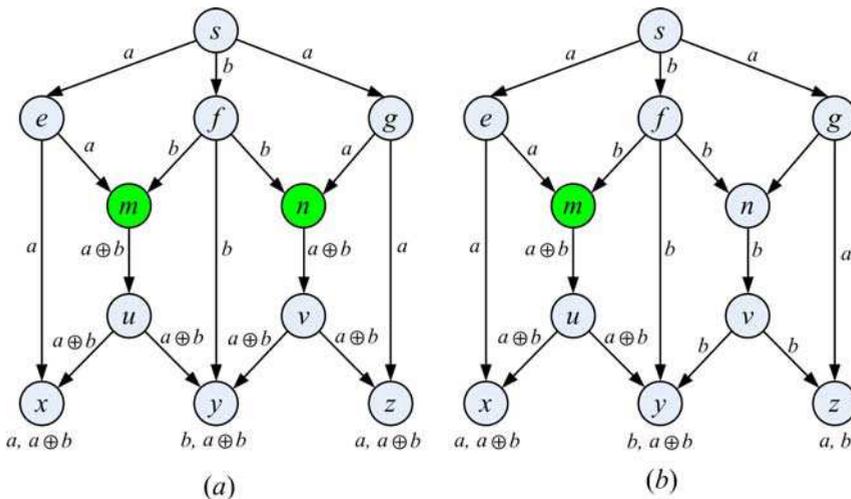


Fig. 2. Two different network coding schemes. (a) Network coding scheme A with two coding nodes. (b) Network coding scheme B with only one coding node.

In order to solve this problem, several algorithms have been proposed, which are mainly based on greedy algorithms or evolutionary approaches. In (Fragouli & Soljanin, 2006), a minimal subtree graph for a network coding multicast was created. The amount of coding operations totally depends on the link traversal order in a corresponding labeled line graph. Different link traversal order may result in different utilization of coding resource. Langberg et al. (2006) first transformed the given network to a network in which the degree of each node is up to 3. Then they checked on links one by one, and remove those which do not make contribution to the achievable rate. However, both of the above algorithms assume that the nodes with multiple incoming links must carry out network coding. Besides, their optimization performance depends on the link traversal order in the corresponding labeled line graph. In (Bhattad et al., 2005), linear programming formulations were proposed to optimize various network coding resources. Nevertheless, the number of variables and constraints grows with the number of sinks. Thus, this method limits itself to the case where the number of sinks is not large. Some genetic algorithms (GAs) with both centralized and distributed versions have been proposed to minimize the network coding operations required (Kim et al., 2006; 2007a; 2007b). The GA based algorithms seem to perform much better than the aforementioned minimal algorithms. However, due to the inherent shortcomings of GA such as pre-maturity, slow convergence speed, weak global searching capability, poor optimal performance is usually achieved.

Quantum-inspired evolutionary algorithm (QIEA), a combination of quantum computation and genetic algorithm, was formally introduced by Han and Kim (Han & Kim, 2002). Unlike other evolutionary algorithms, QIEA maintains a population of Q-bit representation based individuals, each of which represents a linear superposition of all states in search space probabilistically, and adopts various quantum gates, e.g. quantum rotation gate (Han & Kim, 2002) and quantum NOT gate (Xing et al., 2009a; 2009b), to drive the individuals toward the global optima. As one of estimation of distribution algorithms (EDAs), QIEA maintains and incrementally modifies multiple probabilistic models (Platel et al., 2009). QIEA is characterized by maintaining a diversified population due to the Q-bit representation, being able to explore the search space with a smaller number of individuals and exploiting the search space for a global solution within a short computational time (Han & Kim, 2004). However, in QIEA, global exploration and local exploitation can be provided simultaneously, only if proper evolution parameter values are set. Having a great effect on optimization performance of QIEA, how to set proper evolutionary parameters values for algorithms must be paid enough attention. However, in most of the existing QIEAs, the determination of evolutionary parameters does not take the differences among individuals into consideration. In (Han et al., 2001; Han & Kim, 2002; Li & Wang, 2007), fixed rotation angle step (FRAS) schemes were put forward. At arbitrary evolutionary generation, the algorithms use the same rotation angle step (RAS) strategy to evolve its population. If two individuals are under the same case according to the corresponding lookup table, they use the same RAS value to update. QIEA with FRAS scheme often results in slow convergence since the RAS values of lookup table never change. Later, the dynamic rotation angle step (called DRAS below) schemes were proposed (Zhang et al., 2003; Lv & Liu, 2007), where new RAS schemes were given at each generation. By using DRAS schemes, the searching grid of QIEA varies from large to small, and it is of some help to accelerate the convergence and achieve better solutions. However, at each generation, all the individuals under DRAS schemes only refer to one lookup table to update, which means DRAS schemes are also designed for a population but may be not suitable for every individual. Mutation is an

effective operation to prevent premature convergence and raise global search capability. Nevertheless, it was not adopted as a basic evolutionary operation in conventional QIEA (Han & Kim, 2002). Although, quantum mutation operations was introduced in (Yang et al., 2003), the differences among individuals were not considered so that the algorithm may sometimes be trapped in local search.

In order to provide an efficient network coding multicast scheme with less coding operations occupied and overcome the problems caused by the existing RAS schemes and conventional quantum mutation operation, this chapter presents a novel evolutionary algorithm called Memory-Storable Quantum-Inspired Evolutionary Algorithm (MS-QIEA). MS-QIEA is able to assign its individuals more suitable evolutionary parameter values according to their previous searching situations (such as, their fitness and evolutionary parameter values of the former generation). Each individual could easily evolve itself to a better searching position with respect to its former situation. Individuals, whose current searching situations are better than their previous searching situations, are allowed to have relatively large rotation angle step (RAS) values to accelerate the exploration speed and relatively small quantum mutation probability (QMP) values to survive, while those whose current searching situations are worse than their former searching situations are allocated with relatively small RAS values to avoid invalid evolution and relatively large QMP values to wait for a turn-to-excellent-individual opportunity. We evaluated the performance of MS-QIEA over a number of multicast scenarios. Simulation results showed that our algorithm performs better than some traditional evolutionary algorithms in terms of robustness, success ratio, convergence and global search capability.

2. Problem formulation

A communication network can be modeled as a directed graph $G = (V, E)$ where V is the set of nodes and E is the set of links (Li et al., 2003). Assume that each link $e \in E$ has a unit capacity. A single-source multicast scenario is considered as a 4-tuple (G, s, T, R) that includes a graph $G(V, E)$, a source node $s \in V$, a set of sinks $T = \{t_1, t_2, \dots, t_d\} \subset V$, and a data rate R at which s wishes to transmit to all the sinks T . Rate R is said to be achievable only if there exists a transmission scheme that enables all $|T|$ sinks to receive all of the information sent at the same rate as R (Kim et al., 2007a). Since linear network coding is sufficient for multicast (Li et al., 2003), this chapter only considers linear coding, where the coded information is a linear combination of the information from its incoming links. Based on linear coding, a network coding based multicast subgraph (called NCM subgraph, denoted by $G_{NCM}(s, T)$) could be constructed if the subgraph guarantees to have R link-disjoint paths from s to each sink t_i , $i = 1, 2, \dots, d$, respectively. A node is required to perform coding operation if there are at least two paths, $Path(s, t_m)$ and $Path(s, t_n)$, that are input to the node and have to flow out on the same outgoing link, where t_m and t_n are two sinks. Fig.3 shows how NCM subgraph is constructed and how it works. Fig.3(a) is a target network with source s and sinks y and z . All links has a capacity of one bit per time unit. Here, R is set to 2 bits per time unit. To generate a NCM subgraph, we need to find two link-disjoint paths between each source-sink pair, s - y and s - z . As indicated in Fig.3(b), there are two link-disjoint paths, a red path and a blue path, between s and y (or z). Meanwhile, node w needs to perform coding operation as there are two paths join together at w and both have link $w \rightarrow x$. Fig.3(c) shows the network coding scheme based on the constructed NCM subgraph. It is clear that w performs coding operation.

For a multicast scenario (G, s, T, R) , it is expected to determine a minimal amount of coding operations to make the rate R achievable. The number of coding links is a precise estimator of the total amount of coding operations required (Langberg et al., 2006). Hence, our optimization objective hereafter is to minimize the number of coding links while achieving a desired throughput. It is easy to understand that different NCM subgraphs, with different amount of coding links but the same multicast throughput, may probably be constructed for a given multicast scenario (refer to Fig.2). Let $n_c(G_{NCM}(s,T))$ be the number of coding links of the created NCM tree. Our goal is shown as follows:

$$\text{Min}\{n_c(G_{NCM}(s,T))\} \tag{1}$$

Note that no coding is necessary at a node with single incoming link, for such node has nothing to combine. According to (Kim et al., 2006; 2007a; 2007b), we refer to a node with multiple incoming links as a *merging node*. To determine whether coding is necessary on an outgoing link of a merging node, it is excessively imperative to verify whether the output depends on a single input without destroying the achievability of the given data rate. The necessity of coding at a link depends on which other links code and thus the problem of deciding where to perform network coding in general involves a selection out of exponentially many possible choices (Kim et al., 2006; 2007a).

Consider a merging node with $m \geq 2$ incoming links and $n \geq 1$ outgoing links. For each $i \in \{1, \dots, m\}$ and each $j \in \{1, \dots, n\}$, if the information from incoming link i contributes to the linearly coded output on outgoing link j , set $a_{ij}=1$, otherwise set $a_{ij}=0$. We refer to the '1' and '0' states as *active* and *inactive* states, respectively. Network coding is required over link j only if two or more link states are active at the same time. Thus, it is meaningful to consider $a_j = \{a_{ij} \mid i = 1, \dots, m, j = 1, \dots, n\}$ as a data block of length m (Kim et al., 2006; 2007a) (see Fig.4 for an example).

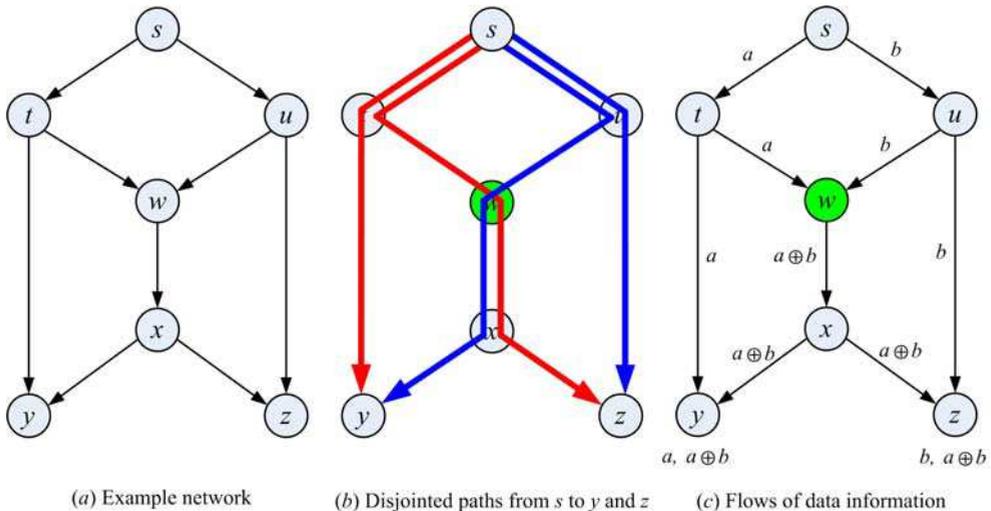


Fig. 3. An example of constructing a NCM subgraph and its data transmission scheme.

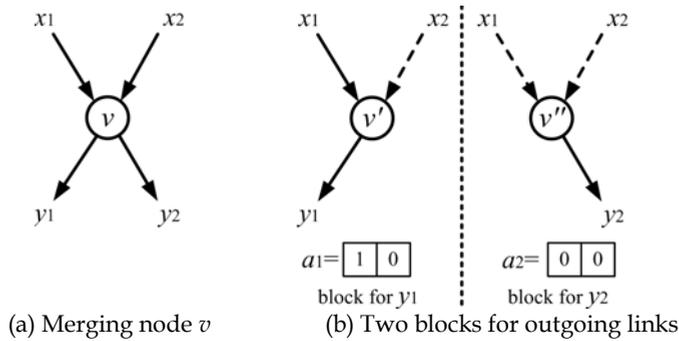


Fig. 4. A possible input states of node v described by vectors $a_1 = (a_{11}, a_{21})$ and $a_2 = (a_{12}, a_{22})$.

3. An overview of QIEA

QIEA is a probabilistic algorithm which exploits the power of quantum computation in order to accelerate genetic procedures (Han & Kim, 2002). The basic information unit is called quantum bit (Q-bit). A Q-bit is a two-level quantum system which may be in the $|0\rangle$ state, in the $|1\rangle$ state, or in any superposition of the two. The state of a Q-bit can be represented as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2}$$

where $|\alpha|^2 + |\beta|^2 = 1$, and α and β are complex numbers that specify the probability amplitudes of the corresponding states.

A Q-bit representation of m Q-bits individual is defined as:

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{pmatrix}, \tag{3}$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1, i = 1, 2, \dots, m$. In this way, the m -Q-bit individual can simultaneously represent 2^m states. Thanks to the Q-bit representation, QIEA has a better characteristic of diversity than classical evolutionary approaches, since it can represent a linear superposition of many states. For example, whereas the 2-bit binary expression $\langle 0, 1 \rangle$ represents one state, a 2-bit Q-bit expression, e.g.

$$\begin{pmatrix} 1/\sqrt{2} & 1/2 \\ 1/\sqrt{2} & \sqrt{3}/2 \end{pmatrix}$$

represents four states: $\frac{1}{8}\langle 0,0 \rangle, \frac{3}{8}\langle 0,1 \rangle, \frac{1}{8}\langle 1,0 \rangle,$ and $\frac{3}{8}\langle 1,1 \rangle,$ where $\frac{1}{8}, \frac{3}{8}, \frac{1}{8},$ and $\frac{3}{8}$ are probabilities with which the corresponding states emerge.

For update, suitable quantum gate $U(\theta)$ is usually adopted in compliance with practical optimization problems. For the problem in this chapter, quantum rotation gate, such as

$$U(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}, \tag{4}$$

where θ is rotation angle, is used as a basic gate of QIEA. The procedure of QIEA is shown in Fig.5. More details are referred to (Han & Kim, 2002).

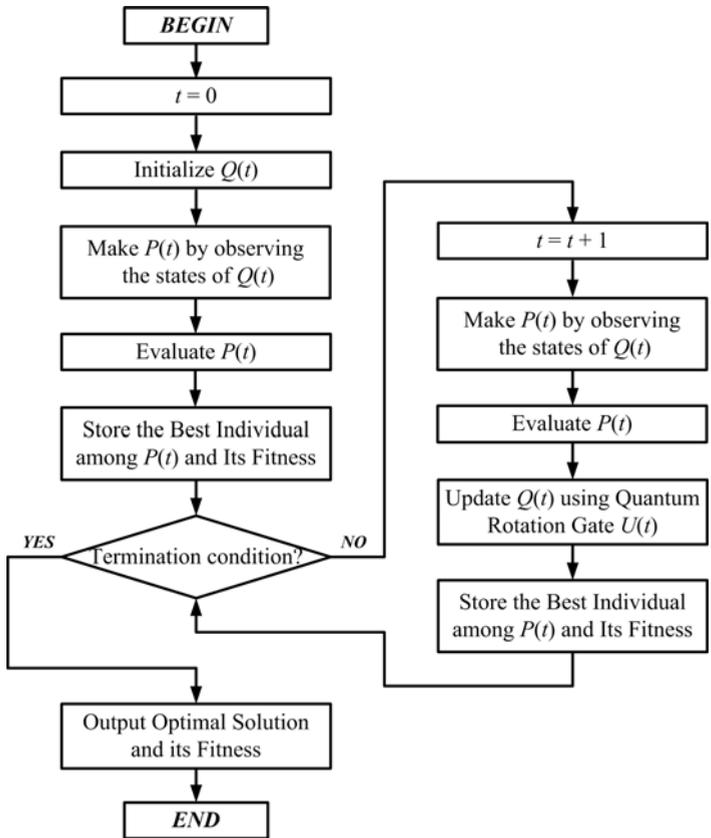


Fig. 5. Flow chart of QIEA

In Fig.5, $Q(t)=\{q_1^t, q_2^t, \dots, q_N^t\}$ is a population of N Q-bit individuals at generation t , and q_j^t is the j -th ($j = 1, 2, \dots, N$) individual defined as

$$q_j^t = \begin{pmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \dots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \dots & \beta_{jm}^t \end{pmatrix} \tag{5}$$

and $P(t) = \{X_1^t, X_2^t, \dots, X_N^t\}$ is a set of binary solutions of observation states of $Q(t)$, where X_j^t is the binary solution obtained by observing q_j^t ($j = 1, 2, \dots, N$). In 'initialize $Q(t)$ ', each pair of Q-bit probability amplitudes, α_{ji}^t and β_{ji}^t , $i = 1, 2, \dots, m$, are initialized with $1/\sqrt{2}$, $\forall q_j^t \in Q(t)$. The step 'Make $P(t)$ by observing $Q(t)$ ' generates a set of binary solutions $P(t) = \{X_1^t, X_2^t, \dots, X_N^t\}$, where each bit of X_j^t , $j = 1, 2, \dots, N$, is formed by determining each explicit state of the corresponding Q-bit of q_j^t , $|0\rangle$ state or $|1\rangle$ state, according to either $|\alpha_{ji}^t|^2$ or $|\beta_{ji}^t|^2$ of q_j^t , $i = 1, 2, \dots, m$. For example, to form a explicit state of the i -th bit of X_j^t ($i = 1, 2, \dots, m$), a

uniformly distributed variable $rand$ between 0 and 1 is generated randomly. If $rand < |\beta_{ji}^t|^2$, the i -th bit of X_j^t is set to 1, otherwise, it is set to 0. Each solution $X_j^t \in P(t)$, $j = 1, 2, \dots, N$, is a binary string of length m , and is evaluated according to the fitness function. The initial best solution X_{best}^t is then selected from the binary solutions $P(t)$ and stored. In the while loop, the quantum gate $U(\theta)$ is used to update $Q(t-1)$ so that fitter states of the Q-bit individuals are generated. The i -th Q-bit value $(\alpha_{ji}^t, \beta_{ji}^t)$ of q_j^t is updated as:

$$\begin{pmatrix} \alpha_{ji}^{t+1} \\ \beta_{ji}^{t+1} \end{pmatrix} = U(\theta_{ji}^t) \begin{pmatrix} \alpha_{ji}^t \\ \beta_{ji}^t \end{pmatrix} = \begin{pmatrix} \cos(\theta_{ji}^t) & -\sin(\theta_{ji}^t) \\ \sin(\theta_{ji}^t) & \cos(\theta_{ji}^t) \end{pmatrix} \begin{pmatrix} \alpha_{ji}^t \\ \beta_{ji}^t \end{pmatrix} \quad (6)$$

Here, θ_{ji}^t is the rotation angle to update the i -th Q-bit of q_j^t . The best solution among $P(t)$ is selected after 'Evaluate $P(t)$ '. If the best solution of the current generation is fitter than the stored best solution X_{best}^t , the stored best solution X_{best}^t is replaced by this new solution.

4. The proposed MS-QIEA

This section provides an efficient coding resource optimization algorithm which is based on quantum-inspired evolutionary algorithm (QIEA) with Memory-Storable RAS (MS-RAS) scheme and Memory-Storable QMP (MS-QMP) scheme. At first, MS-RAS and MS-QMP schemes are introduced. Then, the individual representation and the fitness function are described. Eventually, we show the structure of the proposed algorithm MS-QIEA.

4.1 Memory-Storable RAS scheme

As RAS has a great influence on the convergence of QIEA, it is imperative to choose suitable RAS (Han & Kim, 2002). Individuals may fail to reach optimum solutions in few generations if RAS is excessively small, or they may miss optimum solutions if RAS is excessively large. However, most of the existing QIEA algorithms adopt either FRAS or DRAS as their RAS determination strategy, ignoring whether the RAS scheme is fit for all individuals. They, thus, may eventually result in bad optimization performance.

In order to improve the optimization efficiency, this chapter proposes an MS-RAS scheme, which adaptively allocates each individual a suitable RAS value at each generation. Under this scheme, every individual is treated separately according to its own situation. For arbitrary individual, if its fitness value at generation t is larger than that at generation $t-1$, the current RAS value is set to be smaller than its last RAS value; If its fitness value is smaller than that at last generation, the current RAS value is set to be larger than its last RAS value; Otherwise, the current RAS value is set to be equal to its last RAS value. Note that the problem concerned is a minimum problem where individuals with large fitness values are regarded to be inferior to those with small fitness values (refer to fitness definition in subsection 4.4).

A Q-bit individual q_j^t can be updated by rotation gate $U(\theta_{ji}^t)$, where θ_{ji}^t is the rotation angle for updating the i -th Q-bit of q_j^t , $i = 1, 2, \dots, m$, (refer to section 3). Here, θ_{ji}^t is given as $S(\alpha_{ji}^t, \beta_{ji}^t) \cdot \Delta\theta_{ji}^t$, where $S(\alpha_{ji}^t, \beta_{ji}^t)$ and $\Delta\theta_{ji}^t$ are the sign and the RAS of θ_{ji}^t , respectively. The general updating scheme for the j -th individual at generation t is proposed, as shown in Table 1. The symbol $f(X)$ is the fitness value of observation state X , and b_i and x_i are the i -th bit of the stored best solution X_{best}^t and a current solution X , respectively.

x_i	b_i	$f(X) \leq f(X_{best}^t)$	$\Delta \theta_j^t$	$S(\alpha_{ji}^t, \beta_{ji}^t)$			
				$\alpha_{ji}^t \beta_{ji}^t > 0$	$\alpha_{ji}^t \beta_{ji}^t < 0$	$\alpha_{ji}^t = 0$	$\beta_{ji}^t = 0$
0	0	false	0	0	0	0	0
0	0	true	0	0	0	0	0
0	1	false	0	0	0	0	0
0	1	true	δ_j^t	-1	1	± 1	0
1	0	false	δ_j^t	-1	1	± 1	0
1	0	true	δ_j^t	1	-1	0	± 1
1	1	false	δ_j^t	1	-1	0	± 1
1	1	true	δ_j^t	1	-1	0	± 1

Table 1. MS-RAS scheme for the i -th individual

We denote by δ_j^t the RAS value of the j -th individual. The principle difference among FRAS, DRAS and MS-RAS schemes is shown in Fig.6. At each generation, all individuals under either FRAS scheme or DRAS scheme are assumed to use only one RAS value, while each individual under MS-RAS scheme is able to use a special RAS value to get updated. The expression of δ_j^t is defined as follows:

$$\delta_j^t = \begin{cases} \sigma_0, & t = 0 \\ \delta_j^{t-1} + \Delta(f_j^{t-1}, f_j^t), & t > 0 \ \& \ \delta_j^{t-1} + \Delta(f_j^{t-1}, f_j^t) > 0 \\ \Delta\sigma, & t > 0 \ \& \ \delta_j^{t-1} + \Delta(f_j^{t-1}, f_j^t) \leq 0 \end{cases} \quad (7)$$

where, we have

$$\Delta(f_j^{t-1}, f_j^t) = \begin{cases} 0, & f(X_j^t) = f(X_j^{t-1}) \\ -\Delta\sigma, & f(X_j^t) > f(X_j^{t-1}) \\ \Delta\sigma, & f(X_j^t) < f(X_j^{t-1}) \end{cases} \quad (8)$$

Here, σ_0 and $\Delta\sigma$ are two constants initialized at the beginning of the algorithm which affect the convergent speed. Besides, $\Delta\sigma$ is smaller than σ_0 . We set $\Delta\sigma = 0.1\sigma_0$ in this chapter. X_j^t and X_j^{t-1} are the observation states of the j -th individual at generation t and $t-1$, respectively. Indicating whether an individual evolves to a better or worse searching situation, the value of $\Delta(f_j^{t-1}, f_j^t)$ is selected from $\{-\Delta\sigma, 0, \Delta\sigma\}$ according to whether $f(X_j^t)$ is larger than $f(X_j^{t-1})$. If $f(X_j^t) > f(X_j^{t-1})$, $\Delta(f_j^{t-1}, f_j^t)$ is set to $-\Delta\sigma$, which means searching ability of the j -th individual is becoming worse and δ_j^t is then set to be smaller than δ_j^{t-1} to slow down its turn-to-bad performance. If $f(X_j^t) < f(X_j^{t-1})$, $\Delta(f_j^{t-1}, f_j^t)$ is set to $\Delta\sigma$, which means the j -th individual has reached a better searching position than its previous situation and δ_j^t is then larger than δ_j^{t-1} to accelerate its searching speed. If $f(X_j^t) = f(X_j^{t-1})$, it is not clear that the j -th individual performs better or worse, so we just reserve its previous RAS value.

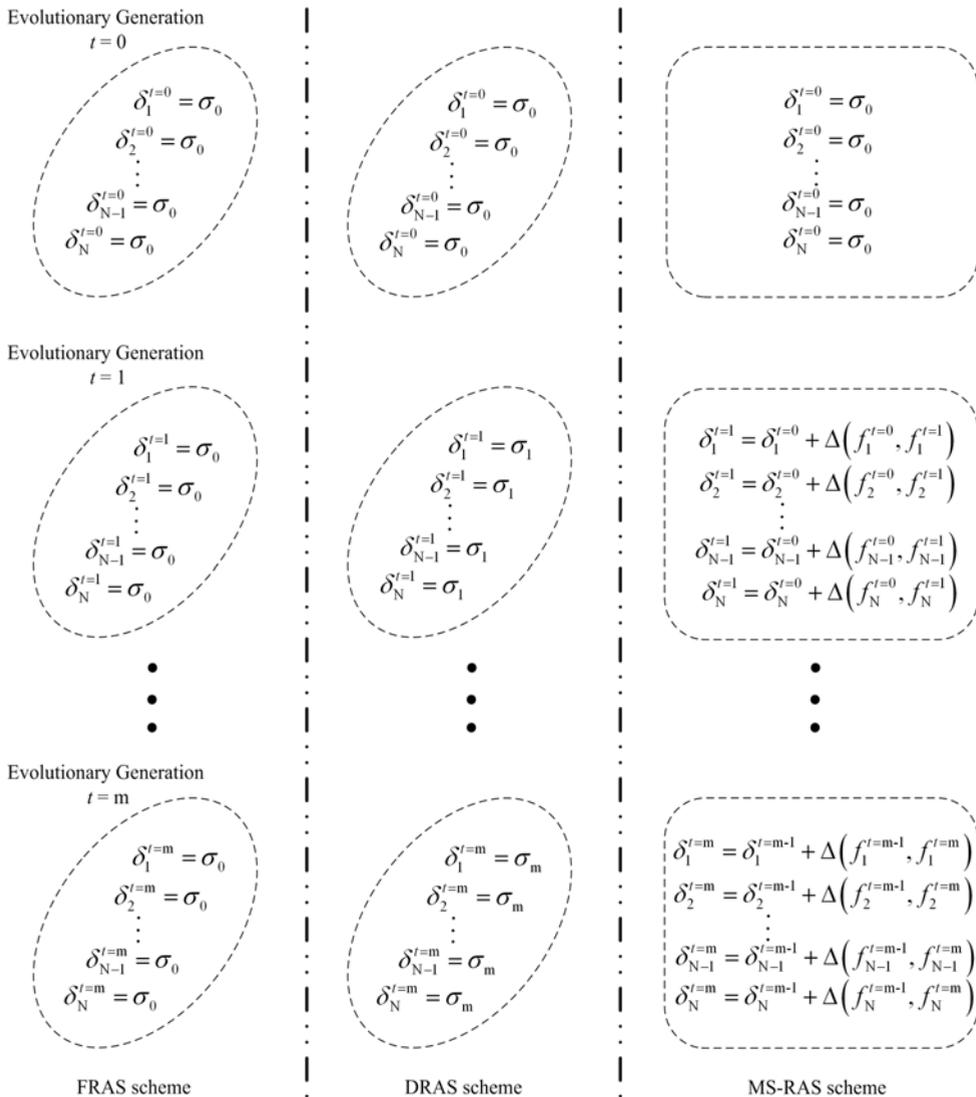


Fig. 6. Principle comparison of FRAS, DRAS and MS-RAS schemes

Based on the MS-RAS scheme, all individuals use the same RAS value σ_0 for update at the first generation. With the evolution continuing, the RAS value δ_j^t of the j -th individual at generation t is determined by its previous RAS value δ_j^{t-1} and its previous-and-current fitness values. When better than their previous searching performance, individuals are allowed to have relatively large rotation angle step (RAS) values to accelerate the exploration speed. When worse than their previous searching performance, individuals are allocated with relatively small RAS values to avoid invalid evolution. Thus, fast convergence and high optimization efficiency will be characterized.

4.2 Memory-Storable QMP scheme

Inspired by (Yang et al., 2003), a quantum mutation operation based on quantum NOT gate, called MS-QMP scheme, is proposed. Instead of employing only one mutation probability for the entire population, MS-QMP scheme assigns each individual a suitable QMP value according to the individual's own searching situation. For arbitrary individual at generation t , if its fitness value $f(X_j^t)$ is larger than $f(X_j^{t-1})$ at last generation, the current QMP value is set to be larger than its last RAS value; If its fitness value $f(X_j^t)$ is smaller than that of its last generation $f(X_j^{t-1})$, the current RAS value is set to be smaller than its last RAS value; Otherwise, the current RAS value is set to be equal to its last RAS value. The quantum mutation probability p_j^t of the j -th individual is defined as:

$$p_j^t = \begin{cases} p_0 & , t = 0 \\ p_j^{t-1} + \varphi(f_j^{t-1}, f_j^t) & , t > 0 \end{cases} \tag{9}$$

where we have,

$$\varphi(f_j^{t-1}, f_j^t) = \begin{cases} 0 & , f(X_j^t) = f(X_j^{t-1}) \\ -\Delta p & , f(X_j^t) < f(X_j^{t-1}) \\ \Delta p & , f(X_j^t) > f(X_j^{t-1}) \end{cases} \tag{10}$$

where p_0 and Δp are two constants initialized at the beginning of the algorithm. We set $\Delta p = 0.1p_0$ in this chapter. Similar to $\Delta(f_j^{t-1}, f_j^t)$, $\varphi(f_j^{t-1}, f_j^t)$ is to indicate whether $f(X_j^t) > f(X_j^{t-1})$. If $f(X_j^t) > f(X_j^{t-1})$, $\varphi(f_j^{t-1}, f_j^t)$ is set to a positive constant Δp which makes p_j^t larger than p_j^{t-1} and the j -th individual is thus more likely to mutate. If $f(X_j^t) < f(X_j^{t-1})$, $\varphi(f_j^{t-1}, f_j^t)$ is set to a negative constant $-\Delta p$ which makes p_j^t smaller than p_j^{t-1} and the j -th individual thus has a higher probability to survive. If $f(X_j^t) = f(X_j^{t-1})$, we just set $\varphi(f_j^{t-1}, f_j^t) = 0$.

The better the individual, the smaller the assigned mutation probability. Based on MS-QMP scheme, excellent individuals have more chance to survive while those bad ones are more likely to mutate, hence pre-maturity is avoided and global searching capability is enhanced.

4.3 Individual representation

Number all merging nodes sequentially from 1 to L , where L is the number of merging nodes. Assume the i -th merging node has $m(i)$ incoming links and $n(i)$ outgoing links. There are precisely $2^{m(i)n(i)}$ different ways that an information flow passes by the node. Besides, there are $2^{m(i)}$ possible ways that the information flow from $m(i)$ incoming links may pass by the j -th outgoing link of the i -th merging node. Obviously, there are $2^{\sum_{i=1}^L m(i)n(i)}$ possible ways that a multicast passes through all merging nodes.

Since each Q-bit represents a superposition of states '1' and '0' and it collapses to an explicit state after each measurement, i.e. either '1' or '0', it is suitable to represent incoming link states *active* or *inactive*, of a merging node (refer to section 2). This chapter adopts a $m(i) \cdot n(i)$ Q-bit individual representation $G = \{g_1, g_2, \dots, g_L\}$, where g_i , $i = 1, 2, \dots, L$, is the block of incoming link states for the i -th merging node. The segment g_i includes $n(i)$ sub-segments, each of which represents the block information of the k -th outgoing link of the i -th merging node, where $k = 1, 2, \dots, n(i)$. After observing g_i , a certain block of incoming link states for the i -th merging node is achieved (see Fig.7 as an example).

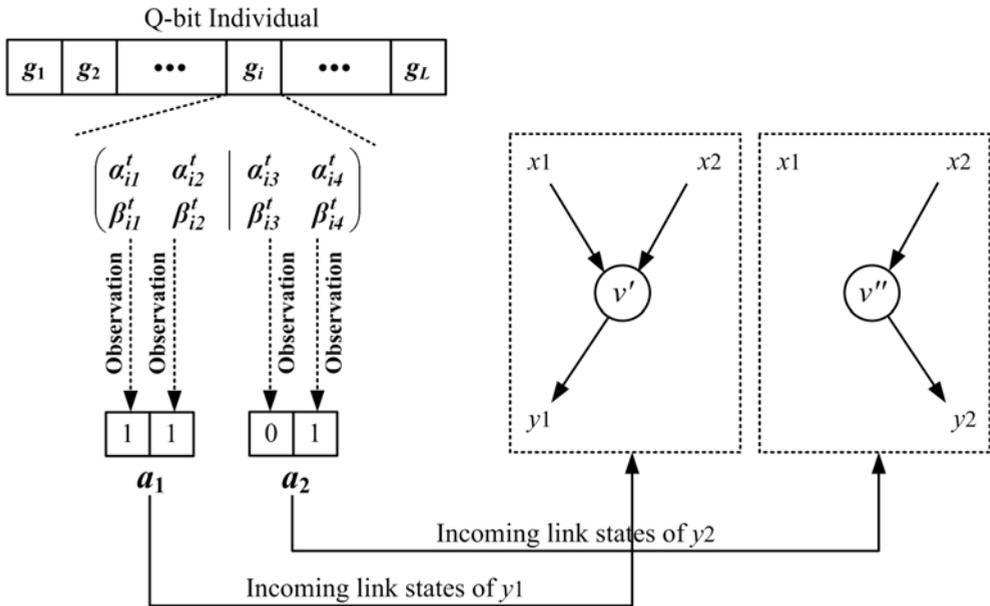


Fig. 7. Individual representation

4.4 Fitness evaluation

This section provides a penalty-function-based fitness function (PF-FF) definition. Based on PF-FF, the fitness value $f_{PF-FF}(X)$ of observation state X of individual q is defined as

$$f_{PF-FF}(X) = \begin{cases} n_{cl} & , \text{Flag}(X) = 0 \\ \sum_{i=1}^d \Theta(s, t_i) & , \text{Flag}(X) = 1 \end{cases} \quad (11)$$

Where we have,

$$\Theta(s, t_i) = \begin{cases} 0, & \text{Flag}(s, t_i) = 0 \\ \Gamma, & \text{Flag}(s, t_i) = 1 \end{cases} \quad (12)$$

where, $Flag(X)$ is to show whether the observation state X of individual q is feasible (According to (Kim et al., 2007a; 2007b), if the achieved max-flow is no less than the desired max-flow, we regard X feasible, otherwise, we regard X infeasible). If $Flag(X) = 0$, it means X is feasible and $f_{PF-FF}(X)$ is set to n_{cl} , where n_{cl} is the number of actual coding links. If $Flag(X) = 1$, X is supposed infeasible and must be punished. $Flag(s, t_i)$ is to indicate whether a desired max-flow is achieved from s to t_i , where s is the source node and $t_i, i = 1, 2, \dots, d$, is the i -th sink node. If $Flag(s, t_i) = 0$, the max-flow from s to t_i is achievable and the penalty factor $\Theta(s, t_i)$ is invalid. However, $Flag(s, t_i) = 1$ indicates that the max-flow from s to t_i can not be achieved and punishment should be added. Here, Γ is a constant that indicates the explicit amount of punishment.

To verify the feasibility of an observation state X , *Graph Decomposition Method*, by which every merging node in a given network is decomposed into a group of nodes, is adopted to calculate the max-flow from the source to the sinks (Kim et al., 2007a).

4.5 The structure of MS-QIEA

As each part of the algorithm being already introduced in details, the basic steps of the algorithm can be described in Fig.8.

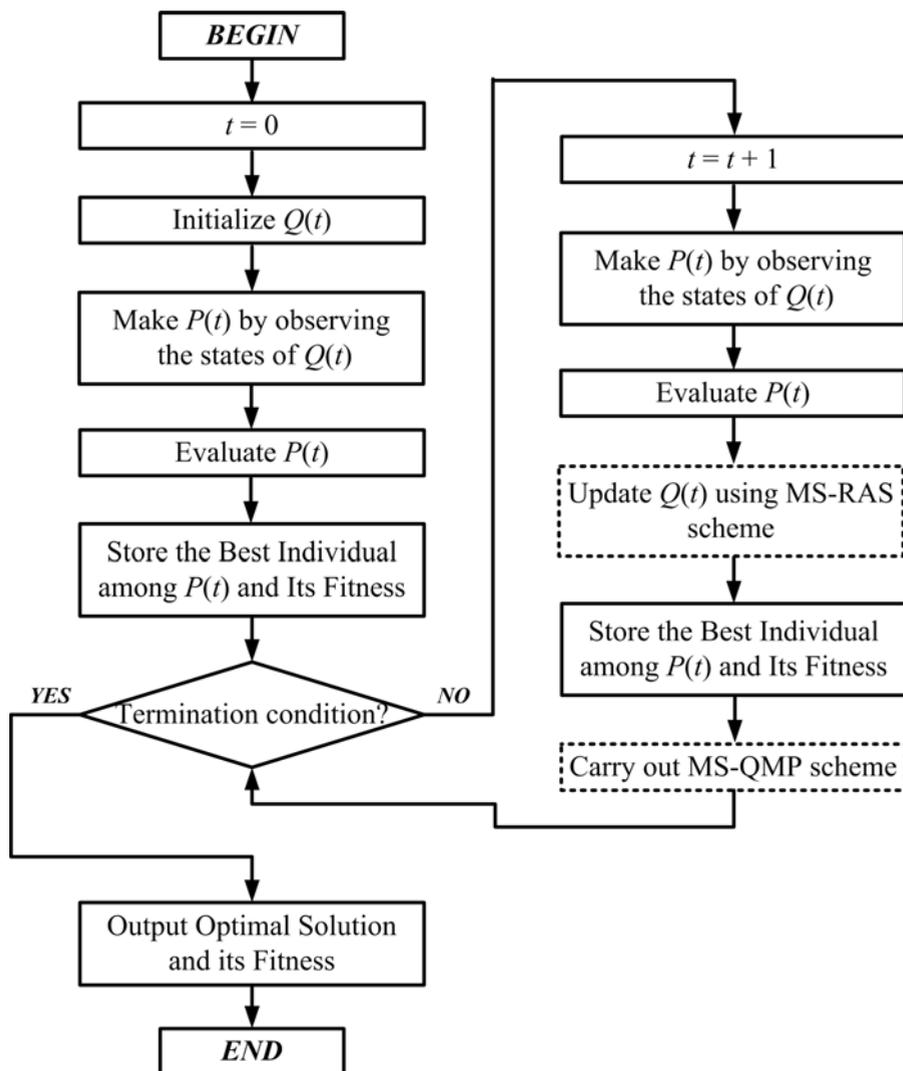


Fig. 8. Flow chart of MS-QIEA

The procedure of MS-QIEA is similar to that of QIEA except for two differences: (1) instead of FRAS and DRAS schemes, MS-RAS scheme is adopted to accelerate the convergence of the algorithm, and (2) MS-QMP scheme is introduced so as to enhance the global searching capability.

5. Experimental results and discussions

In order to evaluate the performance of the proposed algorithm, comparisons of GA (Kim et al., 2007b), QIEA with DRAS scheme (called D-QIEA below), and MS-QIEA have been carried out over three network topologies with the following parameters: Network-I (21nodes, 30links, 4sinks, rate=2), Network-II (20nodes, 37links, 5sinks, rate=3) and Network-III (40nodes, 85links, 10sinks, rate=4). Links in Network-I, Network-II and Network-III are supposed to have unit capacity. Note that all the above algorithms are based on binary link state (BLS) encoding approach (Kim et al., 2007b). However, different from (Kim et al., 2007b), no all one vector is inserted to the initial population so that the algorithms in this chapter may fail to find feasible NCM subgraphs. The population size n_{PS} and terminal iteration n_{TI} of the three algorithms under the cases of Network-I, Network-II, and Network-III are set to: $n_{PS}(20, 20, 40)$ and $n_{TI}(100, 300, 400)$ respectively. In GA, the crossover and the mutation rates are set to 0.8 and 0.1 respectively. In D-QIEA, the initialized RAS value is set to 0.05π . In MS-QIEA, set $\sigma_0 = 0.04\pi$, $\Delta\sigma = 0.004\pi$, $p_0 = 0.1$, $\Delta p = 0.01$ and $\Gamma = 20$. Performance comparisons and simulation results obtained in 500 random trials for each multicast scenario are shown in table 2 and Fig.9-Fig.11.

Multicast Scenarios		Algorithms		
		GA	D-QIEA	MS-QIEA
Network-I	MSR	100.0%	100.0%	100.0%
	BMNCL	2	2	2
	MMNCL	2.00	2.00	2.00
Network-II	MSR	98.7%	99.5%	100.0%
	BMNCL	0	0	0
	MMNCL	0.03	0.01	0.00
Network-III	MSR	99.1%	100.0%	100.0%
	BMNCL	0	0	0
	MMNCL	0.12	0.00	0.00

Table 2. Performance comparisons in three multicast scenarios

By running an algorithm (GA, QIEA, or MS-QIEA) once, one best NCM subgraph $G_{NCM}(s,T)$ can be obtained. So, by running an algorithm 500 times, 500 best NCM subgraphs were achieved. Among these subgraphs, some of them cannot achieve the desirable multicast rate while others (suppose there are W such subgraphs) are feasible ones. In table 2, MSR

represents Multicast Success Ratio, where $MSR = W/500$. BMNCL is the Best Minimum Number of Coded Links achieved among 500 trials, and MMNCL denotes the Mean Minimum Number of Coded Links over 500 trials. Table 2 shows that MS-QIEA outperforms D-QIEA and GA in each case since MS-QIEA always gets the highest MSR and the lowest BMNCL and MMNCL. The second best algorithm is D-QIEA, and the worst one is GA. High MSR indicates that MS-QIEA has higher probability to successfully construct a NCM subgraph with less coding links each time, which reflects the robustness of MS-QIEA. Furthermore, lower BMNCL and MMNCL demonstrate that MS-QIEA has better optimization performance on global searching than QIEA and GA.

Fig.9, Fig.10, and Fig.11 show the convergent speed of the three algorithms in three multicast scenarios. In each figure, note that MS-QIEA distinguishes itself by the fastest convergence. This is because MS-QIEA fully considers individual difference and is able to allocate suitable evolutionary parameter values to each individual according to the individual's previous searching situation, which makes it more suitable to speed up the search and to avoid local optima. As better characteristic of population diversity is achieved by the two QIEA algorithms, broader solution space is explored and exploited simultaneously so that MS-QIEA and D-QIEA perform better than GA.

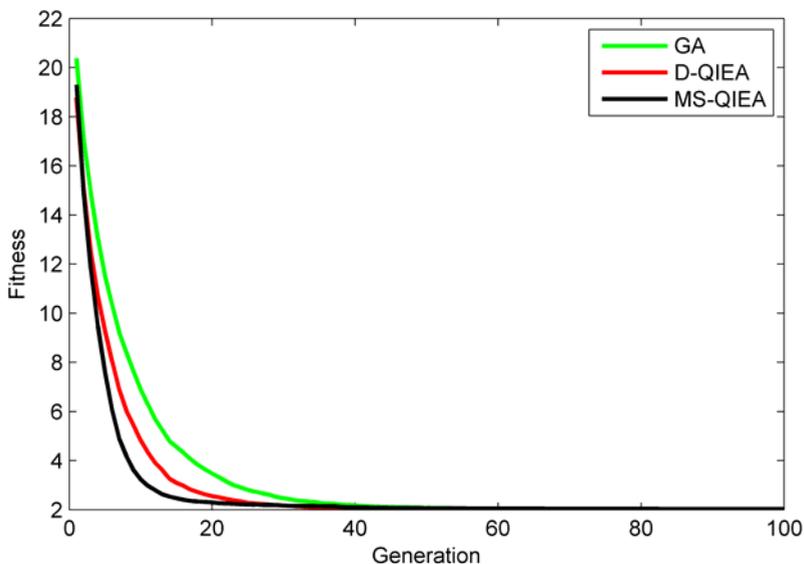
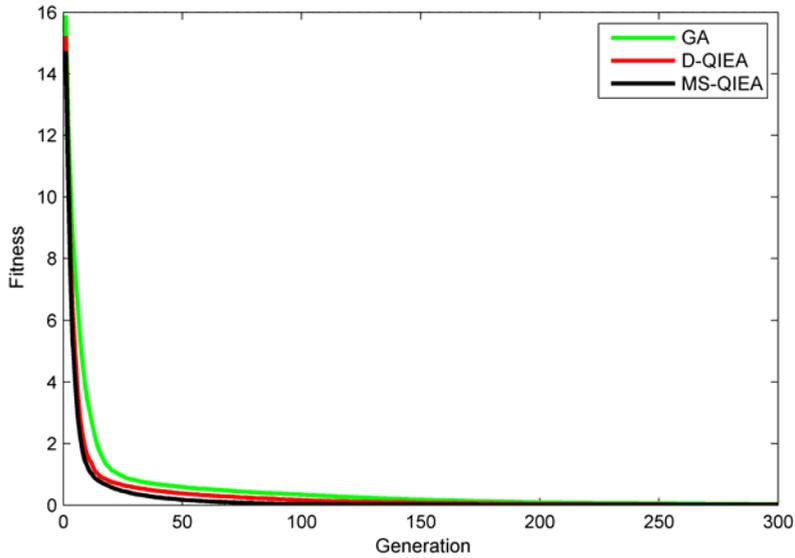
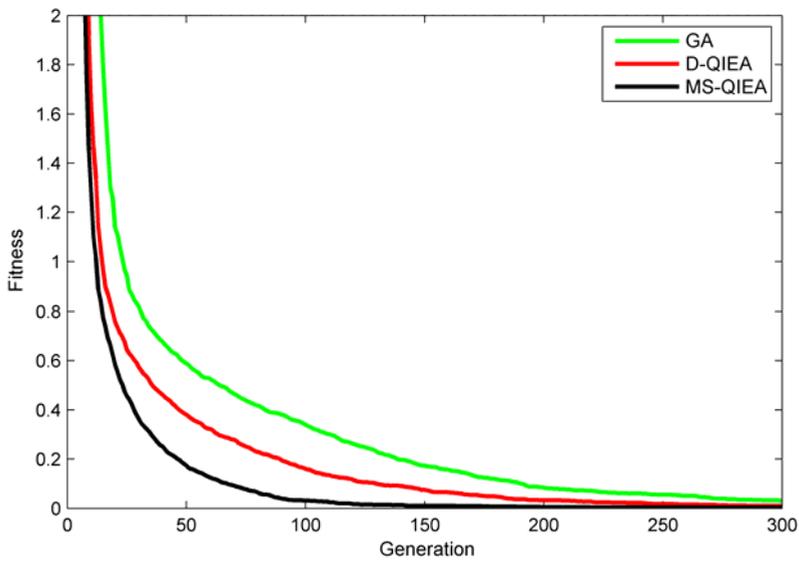


Fig. 9. Comparisons of convergence performance in Network-I



(a)



(b)

Fig. 10. Comparisons of convergence performance in Network-II. (a) Absolute curves without obviously identified trends. (b) Curves to indicate evidence in details that MS-QIEA is superior to D-QIEA and GA in terms of convergence and global searching capability.

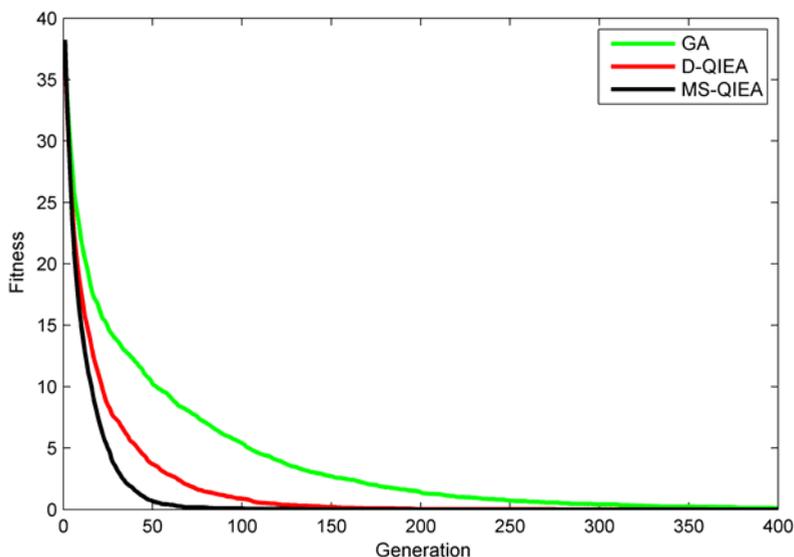


Fig. 11. Comparisons of convergence performance in Network-III

6. Conclusions

An improved QIEA (MS-QIEA) has been proposed for coding resource optimization problem. MS-QIEA updates its individuals with respect to their previous evolution situation. Excellent individuals (the ones achieve increasingly better solutions) are allowed to have relatively large RAS values to accelerate the exploration speed and relatively small QMP values to survive. Inferior individuals (the ones perform worse and worse) are allocated with relatively small RAS values to avoid invalid evolution and relatively large QMP values to expect a chance to become excellent individuals. The simulation results clearly demonstrate the superiority of this algorithm over GA and QIEA in terms of robustness, success ratio, convergence and global exploration.

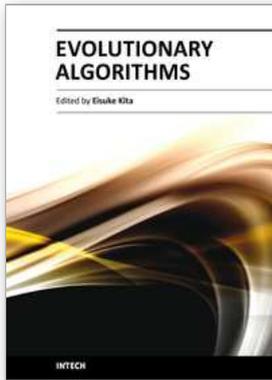
7. Acknowledgement

This research was supported in part by NSFC (No. 60932004), National 973 Program (No. 2007CB310705), National 863 Program (No. 2009AA01A345), P. R. China.

8. References

- Ahlsvede, R.; Cai, N.; Li, S. R. & Yeung, R. W. (2000). Network information flow, *IEEE Transactions on Information Theory*, Vol.46, No.4, July 2000, pp. 1204-1216.
- Bhattad, K.; Ratnakar, N.; Koetter, R. & Narayanan, R. (2005). Minimal network coding for multicast, *Proceedings on International Symposium on Information Theory (ISIT2005)*, pp. 1730-1734, Adelaide, Australia, September 2005.

- Fragouli, C. & Soljanin, E. (2006). Information flow decomposition for network coding, *IEEE Transactions on Information Theory*, Vol.52, No.3, March 2006, pp. 829-848.
- Han, K. H.; Park, K. H.; Lee, C. H. & Kim, J. H. (2001). Parallel quantum-inspired genetic algorithm for combinatorial optimization problem, *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, pp. 1422-1429, Seoul, Korea, May 2001.
- Han, K. H. & Kim, J. H. (2002). Quantum inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Transactions on Evolutionary Computation*, Vol.6, No.6, December 2002, pp. 580-593.
- Han, K. H. & Kim, J. H. (2004). Quantum-inspired evolutionary algorithms with a new termination criterion, H gate, and two-phase scheme, *IEEE Transactions on Evolutionary Computation*, Vol.8, No.2, April 2004, pp. 156-169.
- Kim, M.; Ahn, C. W.; Medard, M. & Effros, M. (2006). On minimizing network coding resources: an evolutionary approach, *Proceedings of Second Workshop on Network Coding, Theory, and Applications (NetCod2006)*, April 2006.
- Kim, M.; Medard, M.; Aggarwal, V.; O'Reilly, U.-M.; Kim, W.; Ahn, C. W. & Effros, M. (2007a). Evolutionary approaches to minimizing network coding resources, *Proceedings of 26th IEEE International Conference on Computer Communications (INFOCOM2007)*, pp.1991-1999, Anchorage, Alaska, May 2007.
- Kim, M.; Aggarwal, V.; O'Reilly, U.-M.; Medard, M. & Kim, W. (2007b). Genetic representations for evolutionary optimization of network coding, *Proceedings of EvoComnet*, 2007.
- Langberg, M.; Sprintson, A. & Bruck, J. (2006). The encoding complexity of network coding, *IEEE Transactions on Information Theory*, Vol.52, No.6, June 2006, pp. 2386-2397.
- Li, B. & Wang, L. (2007). A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling, *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*, Vol.37, No.3, June 2007, pp. 576-591.
- Li, S.R.; Yeung, R. W. & Cai, N. (2003). Linear network coding, *IEEE Transactions on Information Theory*, Vol.49, No.2, February 2003, pp.371-381.
- Lv, Y. & Liu, N. (2007). Application of quantum genetic algorithm on finding minimal reduct, *Proceedings of 2007 IEEE International Conference on Granular Computing*, pp. 728-733, Silicon Valley, California, November 2007.
- Platel, M. D.; Schliebs, S. & Kasabov, N. (2009). Quantum-inspired evolutionary algorithm: a multimodel EDA. *IEEE Transaction on Evolutionary Computation*, Vol.13, No.6, December 2009, pp. 1218-1232.
- Xing, H.; Liu, X.; Jin, X.; Bai, L. & Ji, Y. (2009a). A multi-granularity evolution based quantum genetic algorithm for QoS multicast routing problem in WDM networks. *Computer Communications*, Vol.32, No.2, February 2009, pp. 386-393.
- Xing, H.; Ji, Y.; Bai, L.; Liu, X.; Qu, Z. & Wang, X. (2009b). An adaptive-evolution-based quantum-inspired evolutionary algorithm for QoS multicasting in IP/DWDM networks. *Computer Communications*, Vol.32, No.6, April 2009, pp. 1086-1094.
- Yang, J.; Li, B. & Zhuang, Z. (2003). Multi-universe parallel quantum genetic algorithm and its application to blind source separation, *Proceedings of 2003 IEEE International Conference on Neural Networks and Signal Processing*, pp. 393-398, Nanjing, China, December 2003.
- Zhang, G.; Gu, Y.; Hu, L. & Jin, W. (2003). A novel genetic algorithm and its application to digital filter design, *Proceedings of 2003 IEEE Intelligent Transportation Systems*, Vol.2, pp.1600-1605, Shanghai, China, October 2003.



Evolutionary Algorithms

Edited by Prof. Eisuke Kita

ISBN 978-953-307-171-8

Hard cover, 584 pages

Publisher InTech

Published online 26, April, 2011

Published in print edition April, 2011

Evolutionary algorithms are successively applied to wide optimization problems in the engineering, marketing, operations research, and social science, such as include scheduling, genetics, material selection, structural design and so on. Apart from mathematical optimization problems, evolutionary algorithms have also been used as an experimental framework within biological evolution and natural selection in the field of artificial life.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yuefeng Ji and Huanlai Xing (2011). A Memory-Storable Quantum-Inspired Evolutionary Algorithm for Network Coding Resource Minimization, Evolutionary Algorithms, Prof. Eisuke Kita (Ed.), ISBN: 978-953-307-171-8, InTech, Available from: <http://www.intechopen.com/books/evolutionary-algorithms/a-memory-storable-quantum-inspired-evolutionary-algorithm-for-network-coding-resource-minimization>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.