ELSEVIER

Computer Communications 22 (1999) 1392–1399

computer communications

# QoS routing based on genetic algorithm ☆

F. Xiang*, L. Junzhou, W. Jieyi, G. Guanqun

*Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China*

Received 19 October 1998; received in revised form 13 May 1999; accepted 14 May 1999

**Abstract**

In this paper, the requirements of routing due to the multimedia applications are briefly discussed. In order to solve the QoS constrained routing effectively and efficiently, the scheme of routing based on a genetic algorithm (GA) is proposed after the analysis of related works. Then the QoS routing algorithms for unicast and multicast based on improved GA are described. Finally, the results of the simulations and the comparison of these results are given. 

*Keywords:* QoS routing; Unicast; Multicast; Genetic algorithm

## 1. Introduction

As a result of the emergence of many kinds of high-speed communication systems, such as ATM, and increasing demands of distributed multimedia applications, such as video on demand, multimedia conference, efficient and effective support of quality of service (QoS) has become more and more essential. Five significant issues which have to be considered to design a network system that can guarantee QoS for multimedia applications are described in Ref. [1]. These are flow specification, routing, resource reservation, packet scheduling and admission control. Several approaches have been proposed to support guaranteed performance at the communication system level, and they all adopt a connection-oriented and reservation-oriented approach [2]. The reservation protocol support QoS guarantees for each connection such as maximum delay and loss rare. These parameters are negotiated during the establishment phase by performing the following steps [2]:

1. Send the reserve request to reserve the required resources that the new connection requires, starting from the first node of the selected path between the source and the destination.
2. When an intermediate node receives the reserve request, it checks whether it has sufficient resources to support the requested QoS. If the answer is yes, then the reserve request is forwarded to the next node, otherwise a reject primitive is sent back towards the source node and the previous nodes must free the resources that they have reserved for this request.
3. When the destination node receives the reserve request, it must verify that end to end QoS is satisfied by evaluating the levels of QoS supported by the different nodes of the selected path.
4. If the end to end QoS is not satisfied, then a reject message is sent back towards the source node, each previous node deallocates the reserved resources. Otherwise, a confirmation message with the remaining excess QoS (provided QoS-requested QoS) is sent back to the source node, so that a new connection is successfully established.

The main drawback of those approaches lies in the assuption that the path between two entities is initially known. While this may be reasonable when there is only one path between a source and a destination, it will increase the block probability when multiple paths exist: several retries (steps 1–4) are necessary before establishment. Hence, routing must be combined with QoS constraints so as to minimize the number of retries, maximize the network utilization and reduce the congestion on each entity.

QoS constrained routing algorithm must be able to find a path which (1) is capable of providing resources as requested by the application/user, (2) use existing resources very effectively and efficiently to maximize the probability that new request in future can be fulfilled, and (3) can find the rout in real time. Further, most interactive multimedia applications are multi-party applications and they require

☆ The project is supported by National and Jiangsu Provincial Natural Science Foundation, and key Science and Technology Project of National Education Ministry.

* Corresponding author.

*E-mail address:* xfei@seu.edu.cn (F. Xiang)

0140-3664/99/$ - see front matter 
PII: S0140-3664(99)00113-9

group communication or multicast support from the network.

According to what has been mentioned above, if we could select a path which could satisfy the QoS requirements of applications at the minimal costs in an efficient way, and then use the connection establishment procedure and resources reservation protocol described above, the block probability will be decreased, and the network utilization could be maximized significantly.

In this paper, the problem formulation of QoS routing is first established. Then the QoS routing algorithms based on genetic algorithms (GA) for unicast and multicast are proposed respectively, after the analysis of related works. Finally, the results of the simulations and the comparison of these results are given.

## 2. Problem formulation

We model the network system as a directed graph $G(N,L)$ where the network nodes are presented by vertexes and links that are represented by edges. $N$ is the set of network nodes and $L$ is the set of links in the graph. $W \subset N_i \times N$ is the set of $(s,d)$ pairs (single destination) in network, and $M \subset \{(s,D)|(D \subset N - \{s\})\}$ is the set of multicast group (multiple destination) in the network. We define the QoS routing as the optimization of a cost function while satisfying the QoS requirements of unicast and multicast. Here, we assume that the QoS is guaranteed in the session because of the resource reservation protocol.

*Unicast*: a unicast request, denoted by $(w = (s,d) \in W)$ consists of a source vertex $s$, a destination vertex $d$ and QoS requirements. Where QoS requirements consist of bandwidth requirement $B_w$, delay $D_w$, loss rate $L_w$, and packet delay variation $J_w$ (jitter), which are non-negative real numbers provided by the application/user.

Let AB($l$) denote the available bandwidth of link L. A routing request $w$ will be accepted if the routing algorithm can find a path with minimum cost for $w$, so that the following four conditions are satisfied:

Bandwidth Availability Constraint: AB($l$) $\geq B_w$, for each link $l$ on the rout for $w$;
Delay End to End Constraint: $\sum_{n\in N_1} \mathrm{DN}(n) + \sum_{l\in L1} \mathrm{DL}(l) \leq D_w$, in the rout for $w$,
Loss Rate End to End constraint: $\prod_{n\in N_1} (1 - \mathrm{LR}(n)) \geq (1 - L_w)$, in the rout for $w$, here we consider only losses due to node's buffer overflow.
Jitter End to End constraint: $NDV(\mathrm{d}) \leq \mathrm{J_w}$, in the destination.

Where DN is the node process delay, DN : $N \rightarrow R^+$, DL the link delay, DL : $L \rightarrow R^+$, $N1$ the set of nodes in the rout for $w$, $L1$ the set of links in this rout for $w$, LR the node loss rate, LR : $N \rightarrow R^+ \cup \{0\}$, NDV the node delay variation at the destination $d$, NDV : $N \rightarrow R^+ \cup \{0\}$, $R^+$ the set of positive real number.

*Multicast*: here we consider the case denoted by $m = \{(s,D)|(D \subset N - \{s\})\}$, which consists of a source vertex $s \in N$, multiple destination $d_i \in D$, and the QoS requirements which comprise bandwidth requirement $B_m$, delay $D_m$, loss rate $Lm$, and packet delay variation $J_m$. All of these parameters are non-negative real numbers provided by the application/user. Then a multicast routing request m will be accepted if the routing algorithm can find a routed directed tree for m with the minimum cost, while the following four conditions are realized:

Bandwidth Availability Constraint: AB(1) $\geq B_m$, for each link $l$ on the tree for $m$;
Delay End to End Constraint: $\sum_{n\in N_{1i}} \mathrm{DN}(n) + \sum_{l\in L1i} \mathrm{DL}(l) \leq D_m$, in the $i$th rout for $m$,
Loss Rate End to End constraint: $\prod_{n\in N_{1i}} (1 - \mathrm{LR}(n)) \geq (1 - Lm)$, in the $i$th rout for $m$, here we consider only losses due to node's buffer overflow.
Jitter End to End constraint: NDV($d_i$) $< J_m$, in the $i$th destination.

As for multiple-source-multiple-destination case, it equals to multiple single-source-multiple-destination cases.

It has been demonstrated that if the metrics of routing consists of, at least, two additive metrics (delay, cost, etc.) or the combination of additive and multiplicative metrics, this kind of routing is a NP-complete problem [3,4]. Hence, the QoS constrained routing mentioned above is definitely NP-complete. Traditional routing algorithms cannot solve the QoS routing satisfactorily. Some ways to this problem have been found, such as heuristic algorithms, which could find near optimal solution within a polynomial time. However, they are not so efficient in case of large networks. A proposal using the neural network to solve such problem has been made [5]. However, just as what has been mentioned in Ref. [5], there are many coefficients in energy function that are coupled with one another, hence the selection of the coefficient values is one of the hardest task dealing with Hopfield neural networks, which sometimes leads to wrong solution. Another important fact is that the Hopfield method uses differential equations of neuron voltages as the network dynamics to get the solution, which requires the neuron voltages representing the solutions of the problem to be continuous, whereas the solutions to QoS routing are discrete, so that the solutions to QoS routing must be assumed to be continuous which makes the problem more complex. GA is a kind of parallel optimization algorithm that simulates the evolution process of a creature, and is suitable to find optimal solution in a large and complicated search space. It can automatically obtain and accumulate the knowledge about the research space, so as to control the search process adaptively, until the optimal solution is finally found. GA does not need the solutions of the problem to be continuous so that it is just fit for the QoS routing problem. Further, it is quite simple and robust, and very easy to be implemented via parallel distributed process, so that using GA to solve QoS routing problems could be

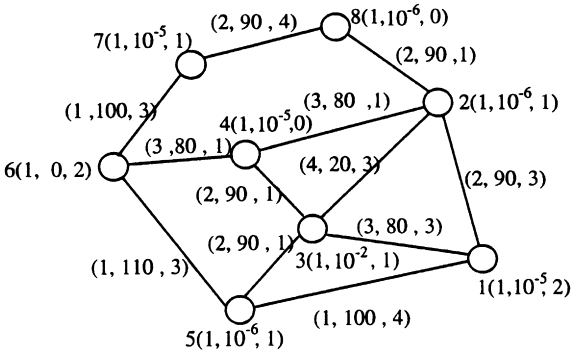


Fig. 1. The topology and its parameters of network system.

quite effective. A method of computing near-optimal solutions to the Steiner problem in a graph using a GA is proposed [6]. In Ref. [6], the algorithm is based on a bitstring encoding of selected Steiner vertices and the corresponding Steiner tree is computed using a deterministic SPG heuristic: KMB heuristic. However, only "cost" metric is taken into account in Ref. [6], while QoS routing requires not only considering the cost to be minimized, but satisfying the QoS constraints. Hence, in this paper, a new GA based procedure is proposed to solve the QoS routing problem.

Fig. 1 shows the topology of a network system which consists of eight nodes and several edges connecting the nodes, each of the nodes is denoted by tuple ⟨ndl, lr, dv⟩, the elements of the tuple are node delay, node loss rate and node delay variation, respectively. Each of the edges is denoted by tuple ⟨cost, bw, ldl⟩, the elements of which are cost, bandwidth and link delay, respectively.

In accordance with the end to end delay variation constraint, if the jitter of destination is larger than the required jitter, we can immediately make the conclusion that the QoS could not be satisfied, in another word, the routing fails. Therefore, here we consider the case in which the jitter in destination is smaller than required.

In this paper, we adopt the scheme of source routing in which each node maintains an image of the global state of the network, and based on which, a routing path is computed at the source. The global state of the network could be obtained by the information exchanges among the nodes.

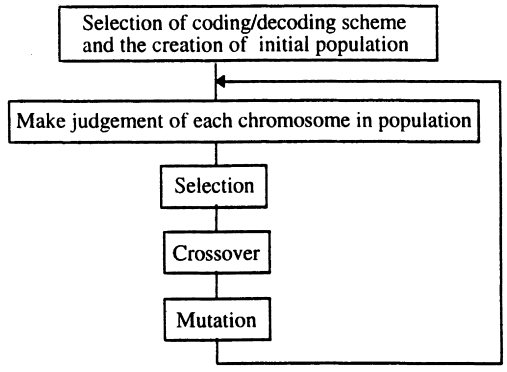


Fig. 2. The general flow of GA.

It should be pointed out that when a node, s, wants to send data with the required QoS parameters to another node, *t* (or *T* which presents a group of destinations), it issues a QoS connection request. The system first identifies a path from *s* to *t* (or *T*) which can satisfy the QoS requirement, and then checks and reserves resources along the path to establish the connection. In this way, the QoS requirements could be guaranteed during the whole session.

According to the general flow of GA [7], as shown in Fig. 2, we first construct general routing algorithm based on GA, then, on account of the inherent speciality of the problem, the simplification and the improvement of the algorithm are proposed.

## 3. Unicast QoS routing algorithm

### 3.1. Coding

We choose $N_n \times N_n$ one-dimensional binary code as the coding scheme of GA, where $N_n$ represents the number of the node in network. Consider Fig. 1 as an example, $N_n = 8$, so the length of the code is 64. The chromosome is represented by $1 \times 2 \times 64$,

$$x_k = \begin{cases} 1 & \text{if link } l_{ij} \text{ from node } i \text{ to node } j \text{ is the link of the selected rout.} \\ 0 & \text{otherwise} \end{cases}$$

where $i = k/N_n$, $j = k\%N_n$, i.e. $i$ is the integral part of $k$ divided by $N_n$, $j$ equals to the arithmetical compliment of $k$ divided by $N_n$. Obviously, this coding scheme satisfies completeness, soundness and non-redundancy.

### 3.2. Crossover and mutation

We, here, discuss the crossover and mutation strategy in SGA: one-point crossover and bit mutation.

### 3.3. Decoding

According to the coding scheme mentioned above and the fitness function to be discussed later, the decoding of the chromosome aims at the translation of $N_n \times N_n$ one-dimensional code into a two-dimensional array

$$O_{ij}^d = \begin{cases} 1 & \text{if the link from node i to node j is the link of the selected rout;} \\ 0 & \text{otherwise.} \end{cases}$$

At the same time, construct $N_n$ one-dimensional array

$$N_m^d = \begin{cases} 1 & \text{if node } m \text{ is the node in the selected rout,} \\ 0 & \text{otherwise} \end{cases}.$$

### 3.4. Fitness function

The design of fitness function should reflect the object of the problem, and is the basis for selection operation.

Before the establishment of fitness function, we introduce several symbol definition:

$$P_{ij} = \begin{cases} 1 & \text{if link from node } i \text{ to node } j \text{ exists,} \\ 0 & \text{otherwise} \end{cases}.$$

$LC_{ij}$: the cost of the link from node $i$ to node $j$;
$LD_{ij}$: the delay of the link from node $i$ to node $j$;
$ND_i$: the process delay of node $i$;
$LB_{ij}$: the bandwidth of the link from node $i$ to node $j$;
$NL_i$: the loss rate of node $i$.

Hence, the establishment of fitness function must satisfy the following conditions:

1. The total cost of the selected route should be minimal.
2. There is only one rout from source to destination.
3. Prevent non-existing links in the network to be selected.
4. Satisfy QoS constraints.

According to the above mentioned conditions, we design the fitness function as:

$$F = A\cdot F_1 + B\cdot F_2 + C\cdot F_3 + D\cdot F_4,$$

where $A$, $B$, $C$, $D$ are positive real coefficients ( $*$ )

$$F_1 = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i \\ (i,j)\neq(d,s)}}^{n} LC_{ij}\cdot O_{ij}^{d},$$

$$F_2 = 1 - O_{ds}^{d} + \sum_{i=1}^{n} \left\{ \sum_{\substack{j=1 \\ j\neq i}}^{n} O_{ij}^{d} - \sum_{\substack{j=1 \\ j\neq i}}^{n} O_{ji}^{d} \right\}^2,$$

$$F_3 = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i \\ (i,j)\neq(d,s)}}^{n} P_{ij}\cdot O_{ij}^{d},$$

$$F_4 = L\cdot \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i \\ (i,j)\neq(d,s)}}^{n} H(z_{ij}) + M\cdot H(z_2) + N\cdot H(z_3),$$

where $L$, $M$, $N$ are positive real coefficients, According to the parameters in Fig. 1, $L$ is set to 1, $M$ and $N$ are set to 10.

$$H(z_i) = \begin{cases} 0 & \text{if } z < 0; \\ z_i & \text{otherwise} \end{cases},$$

$$z_{ij} = O_{ij}^{d}\cdot LB_{ij} - BW,$$

$$z_2 = \left( \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i \\ (i,j)\neq(d,s)}}^{n} LD_{ij}\cdot O_{ij}^{d} + \sum_{i=1}^{n} N_i^{d}\cdot ND_i \right) - D_w,$$

$$z_3 = (1 - L_w)\prod_{i=1}^{n} (1 - N_i^{d}\cdot Nl_i).$$

Here $F_1$, $F_3$, $F_4$ represent the conditions 1, 3 and 4, respectively. In $F_2$, we introduce virtual link from destination to source, as shown in the first term of $F_2$. The second term represents that the number of incoming links equals the number of outgoing links, which means the selected links constitute one rout from source to destination except the virtual link. Hence, $F_2$ represents the second condition.

### 3.5. Selection

In this GA, elitist model is adopted as the selection operator, which means at first execute the selection operation according to the Monte Carlo method, then copy the chromosome with highest fitness to the next generation directly. According to Theorem 2.7 in Ref. [7], this GA could finally converge to the global optimal solution.

### 3.6. Simplification of general routing algorithm

When using the *general routing algorithm based on GA* mentioned above, we could get the optimal solution by simulation. However, we can further simplify the problem scale by making use of the inherent speciality of the problem itself. With the given topology of the network system, we could do the following simplifications. (1) The size of the chromosome is directly proportional to the network scope (e.g. 100 nodes = 10 000 bit long chromosomes), which will lead the processing time to exceed the total transmission time. As we are aware, the interconnection devices (such as router, switch, etc) in the real network system are not in full interconnection, so the values of some of the genes in the process of coding are determinate, i.e. if there is no link between node $i$ and node $j$, then $x_k$ must be 0, where $i = k/N_n$, $j = k\%N_n$. Thus, it can be excluded in the chromosome, and in the process of decoding, corresponding $O_{ij}^{d}$ will be set as 0. (2) As known from $F_2$, virtual link $l_{ds}$ is used to obtain a complete rout, and so it can also be excluded in the chromosome and in the process of decoding, set $O_{ds}^{d}$ as 1. (3) As known from the bandwidth constraint, it requires that the bandwidth of each link is not smaller than the required bandwidth, and we could filter the topology of the network to a new topology, where the links whose available bandwidth is smaller than the required bandwidth are filtered. The new topology definitely satisfies the bandwidth requests.

### 3.7. Modified unicast QoS routing algorithm

With these simplifications, a modified unicast QoS routing algorithm can be described as follows:

1. If NDV($d$) $> J_w$, then exit, routing failed, else go to (2);
2. Filter the topology of the network to a new topology by cancelling the links with bandwidth smaller than the required bandwidth;
3. Produce code $x_1x_2$ in terms of the new topology, which excludes the genes representing the links that do not exist and the virtual link $l_{ds}$;
4. Set the number of generation at 100 according to the experience [7], then, create initial generation randomly;
5. Establish and calculate the fitness function $F$, here the modified $F$ is: (compared with ( $*$ ))

   $$F = A{\cdot}F_1 + B{\cdot}F_2 + D{\cdot}F_4,$$

   where

   $$F_2 = \sum_{i=1}^{n} \left\{ \sum_{\substack{j=1 \\ j\neq i}}^{n} O_{ij}^d - \sum_{\substack{j=1 \\ j\neq i}}^{n} O_{ji}^d \right\}^2,$$

   $$F_4 = M{\cdot}H(z_2) + N{\cdot}H(z_3),$$

   $$z_2 = \left( \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i \\ (i,j)\neq(d,s)}}^{n} LD_{ij}{\cdot}O_{ij}^d + \sum_{i=1}^{n} N_i^d{\cdot}ND_i \right) - D_w,$$

   $$z_3 = (1 - L_w)\prod_{i=1}^{n}(1 - N_i^d{\cdot}Nl_i),$$

6. Make selection operation *with the elitist method*, make crossover operation and mutation operation according to the probability of crossover $P_{\text{crossover}}$ and the probability of mutation $P_{\text{mutation}}$, respectively.
7. If the iteration number exceeds the set iteration number, go to (9). Then, make judgement of each chromosome in the new generation: if there exists a chromosome so that $F_2 + F_4 = 0$, put the value of $F$ and its code to a set *Solution*, then enter (8), else go to (6).
8. Increase the value of $P_{\text{mutation}}$ so as to create more new chromosomes, then go to (5);
9. Choose the chromosome with the largest value $F$, and regard the decoding of this chromosome, $O_{ij}^d$, as the solution of QoS routing, since some applications need the paths to be selected in real-time, which is the trade-off between time and result performance.

Here, the elements in the set *Solution* are the near-optimal solutions, which means the QoS constraints can be satisfied, while the cost is not the minimum. This is due to GA, which may also lead to the local optimal solution. The aim of step (8) is to jump out of the local optimal point, so as to enter a new (local) optimal point. If the scale of the problem is very large, the optimal solution could not be known, and hence we choose the solution with largest value of $F$ in *Solution* as the optimized solution.

## 4. Multicast QoS routing algorithm

We still take Fig. 1 for example, because there exist many similarities between unicast routing and multicast routing, such as Mutation, Selection, part of Coding/Decoding scheme and part of Fitness Function, we will focus on the different parts of the multicast QoS routing compared with unicast routing.

### 4.1. Coding

If we adopt a coding scheme of the general routing algorithm, the length of the chromosome will be $N_n \times N_n \times N_d$, where $N_d$ represents the number of destinations. In Case 2, as described later in Section 5, $N_d = 4$, the length of the code is 256, causing the problem space to be very large, and the low efficiency in optimization, so that the simplified coding scheme must be adopted, in which the length of the code is reduced only to 87, about one-third of the general coding scheme, and the size of the chromosome is shortened significantly.

### 4.2. Crossover strategy

In this problem, we adopt a new method which is based on `"and" logic operation and "or" logic operation plus one-point crossover operation`. The main idea is: select two parents P1 and P2 according to the elitist model, then let C1 = P1&P2 bit by bit, and C1 = P1|P2 bit by bit, where "&" denotes "and" operation and "|" denotes "or " operation, and finally put one-point crossover operation on C1 and C2. Obviously, this method makes the children inherit the same genes of parents: "and" operation is the method where 0 is dominant, while "or" is where 1 is dominant. From the results of simulation, we could see that this method would converge to optimal solution more quickly than one-point crossover method.

### 4.3. Fitness function

Let $D$ denote a one-dimensional array with the same length as $N_d$ $D_k$ represents the $k$th destination. In comparison with the symbols in unicast routing, $O_{ij}^d$should be changed into $O_{ij}^{D_k}$, and $N_m^d$ should be $N_m^{D_k}$, and $F$ should be:

$$F = \sum_{k=1}^{N_d} F^k,$$

$$F^k = A{\cdot}F_1^k + B{\cdot}F_2^k + D{\cdot}F_4^k.$$

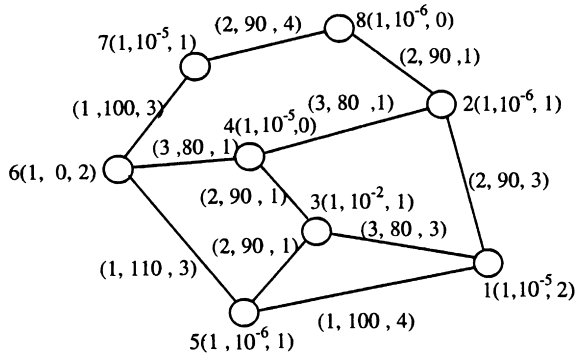


Fig. 3. The filtered topology and its parameters.

Compared with the energy function in Ref. [5] which includes six coefficients, the fitness function here includes only three coefficients, so that the adjustment of these coefficients becomes very easy, where

$$F_1^k = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i \\ (i,j)\neq(d,s)}}^{n} LC_{ij} \cdot S_{ij}^{D_k}(O) \cdot O_{ij}^{D_k},$$

$$S_{ij}^{D_k}(O) = 1/1 + \sum_{\substack{n \neq 1 \\ n \neq D_k \\ n \in D}}^{N_n} O_{ij}^{n},$$

which means, if a link in one selected rout $(s, D_k)$ is also selected by other routs, then the cost is reduced in proportion to the number of routs using this link. Finally, the total cost of the link will be considered only once even though it is used by different routs.

$$F_2^k = \sum_{i=1}^{n} \left\{ \sum_{\substack{j=1 \\ j\neq i}}^{n} O_{ij}^{D_k} - \sum_{\substack{j=1 \\ j\neq i}}^{n} O_{ji}^{D_k} \right\}^2$$

$$F_4^k = M \cdot H(Z_2^k) + N \cdot H(Z_3^k)$$

Table 1
Optimal rout in unicast case (result of routing)

The destination is 6
$V[0][0] = 0V[0][1] = 0V[0][2] = 1V[0][3] = 0V[0][4] = 0V[0][5] = 0V[0][6] = 0V[0][7] = 0$
$V[1][0] = 0V[1][1] = 0V[1][2] = 0V[1][3] = 0V[1][4] = 0V[1][5] = 0V[1][6] = 0V[1][7] = 0$
$V[2][0] = 0V[2][1] = 0V[2][2] = 0V[2][3] = 1V[2][4] = 0V[2][5] = 0V[2][6] = 0V[2][7] = 0$
$V[3][0] = 0V[3][1] = 0V[3][2] = 0V[3][3] = 0V[3][4] = 0V[3][5] = 1V[3][6] = 0V[3][7] = 0$
$V[4][0] = 0V[4][1] = 0V[4][2] = 0V[4][3] = 0V[4][4] = 0V[4][5] = 0V[4][6] = 0V[4][7] = 0$
$V[5][0] = 1V[5][1] = 0V[5][2] = 0V[5][3] = 0V[5][4] = 0V[5][5] = 0V[5][6] = 0V[5][7] = 0$
$V[6][0] = 0V[6][1] = 0V[6][2] = 0V[6][3] = 0V[6][4] = 0V[6][5] = 0V[6][6] = 0V[6][7] = 0$
$V[7][0] = 0V[7][1] = 0V[7][2] = 0V[7][3] = 0V[7][4] = 0V[7][5] = 0V[7][6] = 0V[7][7] = 0$

$$z_2^k = \left( \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i \\ (i,j)\neq(d,s)}}^{n} LD_{ij} \cdot O_{ij}^{D_k} + \sum_{i=1}^{n} N_i^{D_k} \cdot ND_i \right) - D_w,$$

$$z_3^k = (1 - L_w) \prod_{i=1}^{n} (1 - N_i^{D_k} \cdot N_{1i}),$$

The framework of multicast QoS routing algorithm is the same as unicast case, which is not presented here for brevity.

## 5. Simulation

**Case 1.** *Unicast QoS Routing*. Fig. 1 shows the topology of network system. Assume node 1 has routing request and the destination is node 6, QoS requirements are: $B_w = 70$, $D_w = 8$, $L_w = 10^{-3}$, $J_m = 3$, then the filtered topology is shown in Fig. 3.

We first use general unicast QoS routing algorithm based

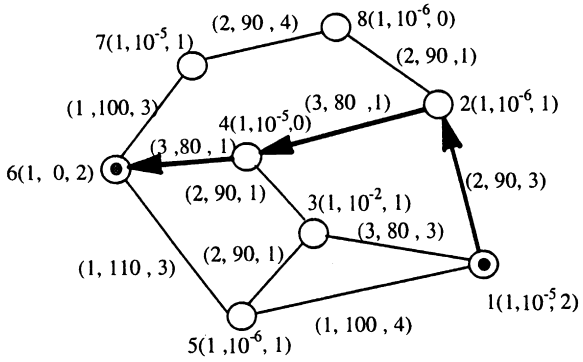


Fig. 4. An optimal rout in unicast case.

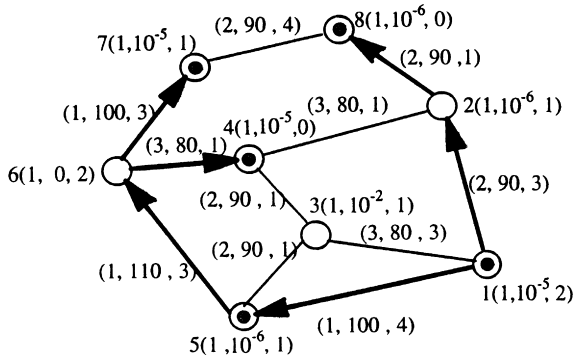


Fig. 5. An optimal rout in multicast case.

Table 2
Optimal rout in multicast case (result of routing)

Generation times: 146
The destination is 3
$V[0][0] = 0V[0][1] = 0V[0][2] = 0V[0][3] = 0V[0][4] = 1V[0][5] = 0V[0][6] = 0V[0][7] = 0$
$V[1][0] = 0V[1][1] = 0V[1][2] = 0V[1][3] = 0V[1][4] = 0V[1][5] = 0V[1][6] = 0V[1][7] = 0$
$V[2][0] = 0V[2][1] = 0V[2][2] = 0V[2][3] = 0V[2][4] = 0V[2][5] = 0V[2][6] = 0V[2][7] = 0$
$V[3][0] = 1V[3][1] = 0V[3][2] = 0V[3][3] = 0V[3][4] = 0V[3][5] = 0V[3][6] = 0V[3][7] = 0$
$V[4][0] = 0V[4][1] = 0V[4][2] = 0V[4][3] = 0V[4][4] = 0V[4][5] = 1V[4][6] = 0V[4][7] = 0$
$V[5][0] = 0V[5][1] = 0V[5][2] = 0V[5][3] = 1V[5][4] = 0V[5][5] = 0V[5][6] = 0V[5][7] = 0$
$V[6][0] = 0V[6][1] = 0V[6][2] = 0V[6][3] = 0V[6][4] = 0V[6][5] = 0V[6][6] = 0V[6][7] = 0$
$V[7][0] = 0V[7][1] = 0V[7][2] = 0V[7][3] = 0V[7][4] = 0V[7][5] = 0V[7][6] = 0V[7][7] = 0$
The destination is 4
$V[0][0] = 0V[0][1] = 0V[0][2] = 0V[0][3] = 0V[0][4] = 1V[0][5] = 0V[0][6] = 0V[0][7] = 0$
$V[1][0] = 0V[1][1] = 0V[1][2] = 0V[1][3] = 0V[1][4] = 0V[1][5] = 0V[1][6] = 0V[1][7] = 0$
$V[2][0] = 0V[2][1] = 0V[2][2] = 0V[2][3] = 0V[2][4] = 0V[2][5] = 0V[2][6] = 0V[2][7] = 0$
$V[3][0] = 0V[3][1] = 0V[3][2] = 0V[3][3] = 0V[3][4] = 0V[3][5] = 0V[3][6] = 0V[3][7] = 0$
$V[4][0] = 1V[4][1] = 0V[4][2] = 0V[4][3] = 0V[4][4] = 0V[4][5] = 0V[4][6] = 0V[4][7] = 0$
$V[5][0] = 0V[5][1] = 0V[5][2] = 0V[5][3] = 0V[5][4] = 0V[5][5] = 0V[5][6] = 0V[5][7] = 0$
$V[6][0] = 0V[6][1] = 0V[6][2] = 0V[6][3] = 0V[6][4] = 0V[6][5] = 0V[6][6] = 0V[6][7] = 0$
$V[7][0] = 0V[7][1] = 0V[7][2] = 0V[7][3] = 0V[7][4] = 0V[7][5] = 0V[7][6] = 0V[7][7] = 0$
The destination is 6
$V[0][0] = 0V[0][1] = 0V[0][2] = 0V[0][3] = 0V[0][4] = 1V[0][5] = 0V[0][6] = 0V[0][7] = 0$
$V[1][0] = 0V[1][1] = 0V[1][2] = 0V[1][3] = 0V[1][4] = 0V[1][5] = 0V[1][6] = 0V[1][7] = 0$
$V[2][0] = 0V[2][1] = 0V[2][2] = 0V[2][3] = 0V[2][4] = 0V[2][5] = 0V[2][6] = 0V[2][7] = 0$
$V[3][0] = 0V[3][1] = 0V[3][2] = 0V[3][3] = 0V[3][4] = 0V[3][5] = 0V[3][6] = 0V[3][7] = 0$
$V[4][0] = 0V[4][1] = 0V[4][2] = 0V[4][3] = 0V[4][4] = 0V[4][5] = 1V[4][6] = 0V[4][7] = 0$
$V[5][0] = 0V[5][1] = 0V[5][2] = 0V[5][3] = 0V[5][4] = 0V[5][5] = 0V[5][6] = 1V[5][7] = 0$
$V[6][0] = 1V[6][1] = 0V[6][2] = 0V[6][3] = 0V[6][4] = 0V[6][5] = 0V[6][6] = 0V[6][7] = 0$
$V[7][0] = 0V[7][1] = 0V[7][2] = 0V[7][3] = 0V[7][4] = 0V[7][5] = 0V[7][6] = 0V[7][7] = 0$
The destination is 7
$V[0][0] = 0V[0][1] = 1V[0][2] = 0V[0][3] = 0V[0][4] = 0V[0][5] = 0V[0][6] = 0V[0][7] = 0$
$V[1][0] = 0V[1][1] = 0V[1][2] = 0V[1][3] = 0V[1][4] = 0V[1][5] = 0V[1][6] = 0V[1][7] = 1$
$V[2][0] = 0V[2][1] = 0V[2][2] = 0V[2][3] = 0V[2][4] = 0V[2][5] = 0V[2][6] = 0V[2][7] = 0$
$V[3][0] = 0V[3][1] = 0V[3][2] = 0V[3][3] = 0V[3][4] = 0V[3][5] = 0V[3][6] = 0V[3][7] = 0$
$V[4][0] = 0V[4][1] = 0V[4][2] = 0V[4][3] = 0V[4][4] = 0V[4][5] = 0V[4][6] = 0V[4][7] = 0$

Table 2 (*continued*)

Generation times: 146
The destination is 3
$V[5][0] = 0V[5][1] = 0V[5][2] = 0V[5][3] = 0V[5][4] = 0V[5][5] = 0V[5][6] = 0V[5][7] = 0$
$V[6][0] = 0V[6][1] = 0V[6][2] = 0V[6][3] = 0V[6][4] = 0V[6][5] = 0V[6][6] = 0V[6][7] = 0$
$V[7][0] = 1V[7][1] = 0V[7][2] = 0V[7][3] = 0V[7][4] = 0V[7][5] = 0V[7][6] = 0V[7][7] = 0$

on GA. The length of coding is 64, and the population size is 100, $P_{\text{crossover}} = 0.95$, $P_{\text{mutation}} = 0.03$, coefficient $A = 10$, $B = 150$, $C = 500$, $D = 400$. These coefficients are chosen to meet the unity requirement, and to guarantee that the selected path is rational and the QoS requirements are satisfied by increasing the corresponding coefficient number. By simulation, we could obtain the global optimal solution as shown in Fig. 4 and Table 1. The average iteration number is 152.

When the simplified coding scheme is adopted, the length of code is reduced to 22, about one-third of that using the general coding scheme, with a significant reduce in the length of the chromosome. Other parameters are the same as general algorithm. By simulation, we could get the global optimal solution much quickly, for the average iteration number is only 15, one-tenth of that using the general algorithm.

We also compare the QoS routing based on GA with that based on the Hopfield network, the iteration number of unicast QoS routing using the Hopfield network is up to 10 941, much more than that using GA.

**Case 2.** *Multicast QoS Routing*. Also consider Fig. 1 as an example, the source is node 1, $D = (4, 5, 7, 8)$, $N_d = 4$ and QoS requirements: $B_m = 70$, $D_m = 15$, $L_m = 10^{-3}$, $J_m = 3$.

We, first use the simplified coding scheme and the one-point crossover method, all the parameters are the same as mentioned above. By simulation, we could only get to near-optimal solutions within the set iteration number, the average iteration number to get the first near-optimal solution is 1035.

In the application of the simplified coding scheme and the "and"/"or" logic operation in addition to the one-point crossover operation with all the other parameters being the same, the average iteration number of the first near-optimal solution could be reduced to 330, one-third of that using the one-point crossover method. At the same time, we could obtain the global optimal solution, and the average iteration number is 992, where 40% of the samples have the iteration number below 200, and 20% between 200 and 500, 20% between 500 and 1000, and 20% more than 1000. When the Hopfield network is used, the iteration number of multicast QoS routing is up to 14 979, which is much

more than using GA. The global optimal solution is shown in Fig. 5 and Table 2.

## 6. Conclusion

In this paper, GA is adopted to solve the problem of routing with QoS constraints required by multimedia applications. Unicast QoS routing algorithm based on GA and multicast QoS routing algorithm based on GA are proposed. Especially, by the simplification and the improvement of the general algorithm, the effect of the modified algorithms proved to be much better. Further, due to the parallelism of GA, the executing time of the algorithms using parallel computing will be further reduced, so the real time requirement of routing in high-speed network system could be satisfied. With the increasing size of the network system and the application of the multilevel hierarchical routing mode, the method of using the algorithm proposed in this paper on such hierarchy needs to be further studied, to shorten the size of chromosomes in addition to the use of simplified coding scheme mentioned above. At the same time, how to implement such an algorithm efficiently by using parallel processing is also very important for real applications.

## References

[1] L. Zhang, S. Deering, D. Estrin, et al., RSVP: a new resource reservation protocol, IEEE Network Magazine September (1993).
[2] M. Nour, A. Hafid, J.W. Atwood, Routing with quality of service constraints, Pacific Workshop on Distributed Multimedia, Vancouver, 1997.
[3] E. Crawley et al., A Framework for QoS Routing in the Internet. Internet-draft, 1996.
[4] Z. Wang, J. Crowcroft, Quality of service routing for supporting multimedia applications, JSAC 14 (7) (1996) 1228–1234.
[5] P. Chotipat, C. Goutam, S. Norio, Neural network approach to multicast routing in real-time communication networks, in: International Conference on Network Protocols 1995, IEEE, pp. 332–339.
[6] H. Esbensen, Computing near-optimal solutions to the Steiner problem in a graph using a genetic algorithm, Networks 26 (1995) 173–185.
[7] C. Guoliang, W. Xufa, Z. Zhenquan, et al., Genetic algorithm and its application, People's Posts and Telecommunications Press, 1996.

**Fei Xiang** received his B. Engng and M. Engng degrees in the Department of Control Engineering from Southeast University, Nanjing, P. R. China in 1992 and 1995, respectively. He is currently working towards the Ph D degree in the Department of Computer Science and Engineering at Southeast University. His research interests include high-speed computer network, protocol engineering and knowledge based network resource management and scheduling.

**Gu Quanqun** is the President of Southeast University, a member of the Chinese Academy of Engineering, and a Member of CIMS Expert Group of National "863" Program. His current research interests include high-performance network, protocol engineering, intelligent computer network, and computer integrated manufacturing (CIM).

**Wu Jieyi** is the Vice President of the Southeast University, a professor of the Department of Computer Science and Engineering, Southeast University, and a Member of CIMS Expert Group of Jiangsu Province. His current research interests include network and information integration in CIMS.

**Luo Junzhou** is a professor of the Department of Computer Science and Engineering, Southeast University, the secretary-general of Petri net committee of China Computer Federation, an active member of New York Academy of Science. His current research interests include Petri net based protocol engineering, computer network, and concurrent engineering.