Malaysian Journal of Computer Science, Vol. 15 No. 2, December 2002, pp. 70-77

A HEURISTIC GENETIC ALGORITHM FOR DISTRIBUTED MULTICAST ROUTING

Youwei Yuan and Lamei Yan
Department of Computer Science and Technology
Zhuzhou Engineering Institute
Hunan, China 412008
email: yuanyouwei@163.net

Mustafa Mat Deris
Department of Computer Science,
College University of Science & Technology
Malaysia
email: mustafa@kustem.edu.my

ABSTRACT

Multicast (MC) routing algorithms capable of satisfying the quality of services(QoS) requirements of real-time applications will be essential for future high-speed networks. Genetic Algorithms (GA) are stochastic search optimisation methods used in combinatorial optimisation and parameter tuning applications. In this paper, a shared-tree routing protocol based on distributed Genetic Algorithms(Gas) is presented, including building and dynamic maintenance of multicast routing tree in package exchange network. The algorithm has the following characteristics: (1) the preprocessing mechanism, (2) the tree structure coding method, (3) the heuristic crossover technique, and (4) the instructional mutation process. Simulation results over random networks show that the genetic algorithms are capable of successfully constructing MC trees which satisfy the QoS requiremnets of real time traffic. GA heuristic constucts low cost trees to solve the minimal multicast tree with delay constraint. As a result, the algorithm is efficient and effective.

Keywords: ***Delay constrained, QoS, Genetic algorithm, Multicast routing***

1.0 INTRODUCTION

In multicast communication, messages are concurrently sent to multiple destinations, all members of the same multicast group. Multicast constrained QoS routing optimisation is an important problem in the current communication network research. Most distributed real-time applications involve more than two users and hence the need for efficient multicast routing. Real-time applications, e.g., multimedia and distributed real-time control applications, will be popular applications of high-speed networks in the near future. Real-time traffic is usually bandwidth extensive and requires quality of service (QoS) guarantees from the underlying network. Hence we need efficient multicast routing algorithms which define cost as a function of the utilised link bandwidth, and are capable of constructing low-cost multicast trees that satisfy the constraints imposed by the QoS requirements. A number of delay-constrained multicast routing algorithms have been proposed in the past few years, but they are all designed for static multicast groups. Though E. Biersack and J. Nonnenmacher [8] have presented a dynamic algorithm (WAVE) for the multi-constrained QoS routing problem, there are drawbacks that WAVE does not consider explicitly for a given delay constraint. References [3-4] have proposed some dynamic multicast routing heuristics. However, the underlying multicast routing algorithms have been designed only for better-effort delivery. The existing delay-constrained minimum Steiner tree heuristics can be used to construct delay-constrained broadcast trees, but most of them are complicated [9]. The comparison given in [2] shows that BSMA [1] is the most efficient delay-constrained minimum Steiner tree heuristics.

In this paper, a shared-tree routing protocol based on distributed genetic algorithms (Gas) is presented, the heuristic genetic algorithm is adopted to solve the minimal multicast tree with delay-constrained least-cost multicast routing problem. With this algorithm, a dynamic constructed multicast routing tree which has a near optimal network cost under the delay bound constraint can be constructed.

The article is structured as follows: Section 2 presents the delay-constrained least-cost multicast routing problem. Section 3 exposes the Distributed routing algorithm based on Genetic Algorithm. Section 4 specifies the simulation results.

2.0 PROBLEM FORMULATION

A network is modeled as a weighted graph G (V, E), with node set V and edge set E. The edges in E correspond to communication links connecting the network nodes. The nodes in set V can be of three types. Source node denotes the node connecting to the source that sends out the data stream. Destination node denotes a node connecting a destination that receives the data stream. Relay node denotes an intermediate hop in the path from the source to a destination, E is the set of edges representing physical or logical connectivity between nodes. Each link is bi-directional. Let $s \in V$ be the multicast source, $M \subseteq V-\{s\}$ the multicast destinations, and R_+ the set of positive real numbers. We define two additive functions to each link $e \in E$: the delay function delay(e):$E \rightarrow R_+$ and the cost function cost(e):$E \rightarrow R_+$. Let t be any destination node of M, $p(s, t)$ the path from s to t. For a given source node $s \in V$ and a destination node set M, there exist the following relationships for the multicast tree constructed by s and M.

A spanning tree $T(s) \subseteq E$ is rooted at a source node $s \in V$ and contains a path from s to any node $v \in (V - \{s\})$. The total cost of a tree *T(s)* is simply:

$$Cost\ (T(s)) = \sum_{t \in T(s)} c(t) \tag{1}$$

A path $P(T(s), v) \subseteq T(s)$ is the set of tree links connecting s to $v \in V$. The cost of the path *P(T(s),v)* is:

$$Cost\ (P(T(s), v) = \sum_{t \in P(T(s))} c(t) \tag{2}$$

and the end-to-end delay along that path is:

$$Delay\ \ (P(T(s), v) = \sum_{t \in P(T(s), v)} d(t) \tag{3}$$

Thus the maximum end-to-end delay of a spanning tree is:

$$Max\_Delay\ (T(s)) = \max_{v \in V} (Delay\ (P(T(s), v))) \tag{4}$$

The Delay-Constrained minimum spanning tree (DCMST) problem in directed networks constructs the spanning tree *T(s)* rooted at *s* that has minimum total cost among all possible spanning trees rooted at *s* which have a maximum end-to-end delay less than or equal to a given delay constraint.

Definition 1. Delay-constrained least-cost multicast routing problem: Given a network N (V, E), a source node $s \in V$, a destination node set $M \subseteq V-\{s\}$, the delay-constrained least-cost multicast routing problem is to find a multicast tree that satisfies:

$$\min\{\cos t(T(s, M)), T(s, M) \in T_f(s, M) \tag{5}$$

Where $T(s, M)$ is the set of all delay-constrained multicast trees constituted by s and M. It has been demonstrated that the delay-constrained least-cost multicast routing problem is NP-complete.

3.0 DISTRIBUTED ROUTING ALGORITHM BASED ON GENETIC ALGORITHM

3.1 A Computing Model of Distributed Genetic Algorithm

A computing model of distributed genetic algorithm is shown in Fig. 1. Genetic Algorithms(GA) are stochastic search optimisation methods used in combinatorial optimisation and parameter tuning applications. GA seek to mimic the biological processes of reproduction and natural selection. Natural selection determines which members of a population survive to reproduce, and reproduction ensures that the species will continue.

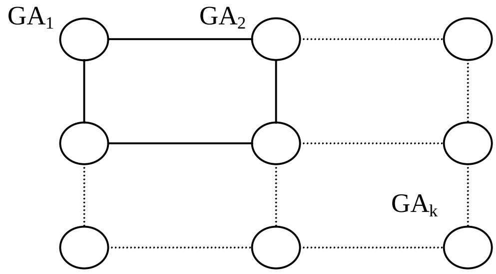


Fig. 1: A computing model of distributed genetic algorithms

Definition 2. The network is described as G = (V, E, C, D), one of the multicast task $\Pi = (S,T,c,d,w)$, where $S, T \subset V, |S| \geq 2, |T| \geq 2, c, d, w \in R^+$. In order to find a multicast tree $\Phi^* + (V_\Phi, E_\Phi, C, D)$, it must satisfy the following conditions:

(1) $C(\Phi^*) \geq c$.

(2) $D(\Phi^*) \leq d$.

(3) $W(\Phi^*) \leq w$

(4) $\forall \Phi \in \Psi(\prod) : Cost(\Phi) \geq Cost(\Phi^*)$

3.2 The Solving QoS Multicast Routing Problem Based on GA

Genetic Algorithms (GA) are stochastic search optimisation methods used in combinatorial optimisation and parameter tuning applications. GA seek to mimic the biological processes of reproduction and natural selection. Natural selection determines which members of a population survive to reproduce, and reproduction ensures that the species will continue. The solving QoS multicast routing problem based on GA is shown in Fig. 2. Algorithm is started with a set of solutions (represented by chromosomes) called population. A solution consists of a set of variables. Each variable can have a value taken from a domain. In our problem setting, the solution is a set of real numbers drawn from [0, P_{max}]. The value of the variable, t_i, corresponds to the transmission power of the source station of connection i. A collection of solutions is referred to as a population. Since GA tries to mimic nature, at any discrete time t, the current population is referred to as the t^{th} generation. By moving from one generation to another over time, the quality of the population is improved. To move from generation t to generation (t+1), we use three genetic operators that have been widely research and used. The first operator is the crossover which exploits the solutions in the current population. Two mutation operators were used: creep and Random Store to diversify and explore other regions of the search space [6]. Solutions from one population are taken and used to form a new population. This is motivated by a hope that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce. The process is terminated if the number of iterations exceeds a specific value or the population converges.

3.2.1 Fitness Proportionate Reproduction

The fitness function is typically one of the objective functions of the problem. Fitness proportionate reproduction is a simple rule whereby the probability of reproduction during a given generation is proportional to the fitness of the individual. We evaluate the Weight Based Genetic Algorithm (WBGA) which uses the weighted sum of each objective function as fitness function, i.e.,

$$Fitness = \sum_{i=1}^{n} w_i \cdot f_i, \tag{6}$$

where n denotes the number of objective functions and w_i denotes the fixed weight for objective function f_i (with $\sum_{i=1}^{n} w_i = 1$).

3.2.2 Crossover Operation

Crossover takes a portion of each parent and combines the two portions to create offsprings. After selection, the strings are copied into a mating pool and crossover occurs on the copies. The main objective of the crossover operation is to produce offsprings that have large fitness values (and satisfy the problem's constraints). This is achieved by swapping parts of the chromosomes of the fittest individuals in the current generations. The crossover operation typically proceeds as follows. First, the individuals in the generation t-1 are sorted in decreasing order of their fitness values. A mating pool is formed from the first $P_P(t)$ individuals in the ordering. Parts of the chromosomes of the individuals in the mating pool are then exchanged (swapped) with a fixed crossover probability, which we fix at the typical value 0.9. We put the M value of the first individual in place of the M value of the second individual, and vice versa. With the complementary crossover probability (0.1), the chromosomes of the two individuals remain unchanged. The two individuals then become members of the reproduction group in the next generation. We then move on to the third and fourth individuals in the ordering, and swap their M values with probability 0.9, move them to reproduction group in the next generation, and so on.

3.2.3 Mutation Operation

Mutation, the random alteration of a string position, performs a second reproduction process. The mutation operation modifies individuals by changing small parts in their chromosomes with a given mutation probability (which we fix at the typical value 0.05). If an individual is selected for the mutation operation, then one bit in the individual's chromosomes is flipped. The location of the bit is drawn randomly from a uniform distribution over the length of the chromosome. In our network optimisation, we implement the mutation operation by randomly drawing an M value from a uniform distribution over [1, F]. We chose to randomly mutate M interval [1, F], as this operation does not result in constraint violations, yet keeps the population sufficiently diverse. After the mutation operation, we evaluate the average throughout and mean delay achieved by the individual in the new generation and start the next evolution cycle, as illustrated in Fig. 2.

3.3 Multicast Routing Algorithm Based on GA

```
Algorithm_2 : the routing based on GA QoSMR_GA(G,s,D)
 /*Input: C set of connections*/
/*Output :Number of established connections
 /* create initial population by random depth first
   search algorithm * /
        for (i=1; i<=N_p; i++) {
              A(i)=RandomDFS(G,s,D);
        }
        for(j=1; j<=N_R; j++){
        / *N_g is the generation times * /
        /* heuristic crossover * /
                  for(k=1; k<=N_p —N_best; k++){
             T_m=MSTSelect(A);
             T_f=MSTSelect(A);
                /* select parents tree for crossover * /
             Select the same link from T_m and T_f;
                 If(rand()<P_m)
                 }/* for k * /
               }/ * for j * /
             select the best individual and output it;
             }/* QoSMR_GA */
```

4.0 SIMULATIONS

GA has been implemented in C++. We used simulation for experimental investigations to avoid the limiting assumptions of analytical modeling. We use the random weight genetic algorithm (RWGA) with elitism with the parameter settings found in the preceding sections , i.e., a population (represented by chromosomes) size of P = 200, G = 40 generations, crossover probability 0.9, and mutation probability 0.05. The number of the nodes in the network is set to N = [20, 200].

To compare the relative advantages or disadvantages of the GA algorithm with the other algorithms, a simulation study was conducted. The following experiments were conducted : (1) operation time testing, (2) routing request success ratio, (3) multicast tree cost inefficiency.

4.1 Multicast Tree Cost Inefficiency

The first experiment compares the different algorithms when each of them is applied to create an MC tree for a given source node generating video traffic with an equivalent bandwidth of 0.5Mb/s and a given MC group. For each run of the experiment we generated a random set of links to interconnect the fixed nodes, we generated random background traffic for each link, we selected a random source node and an MC group of randomly chosen destination nodes. The equivalent bandwidth of each link's background traffic was a random variable uniformly distributed between B_{min} and B_{max}. As the range of the link loads, i.e., the difference between B_{max} and B_{min}, increased, the asymmetry of the link loads also increased. The experiment was repeated with different MC group sizes. We measured the total cost of the MC trees, the maximum end –to-end delay, and the failure rate of the algorithm. The experiment was run repeatedly until confidence intervals of less than 5%, using 95% confidence level, were achieved for all measured quantities. On the average, 320 different networks were simulated in each experiment in order to reach such confidence levels. At least 260 networks were simulated in each case.

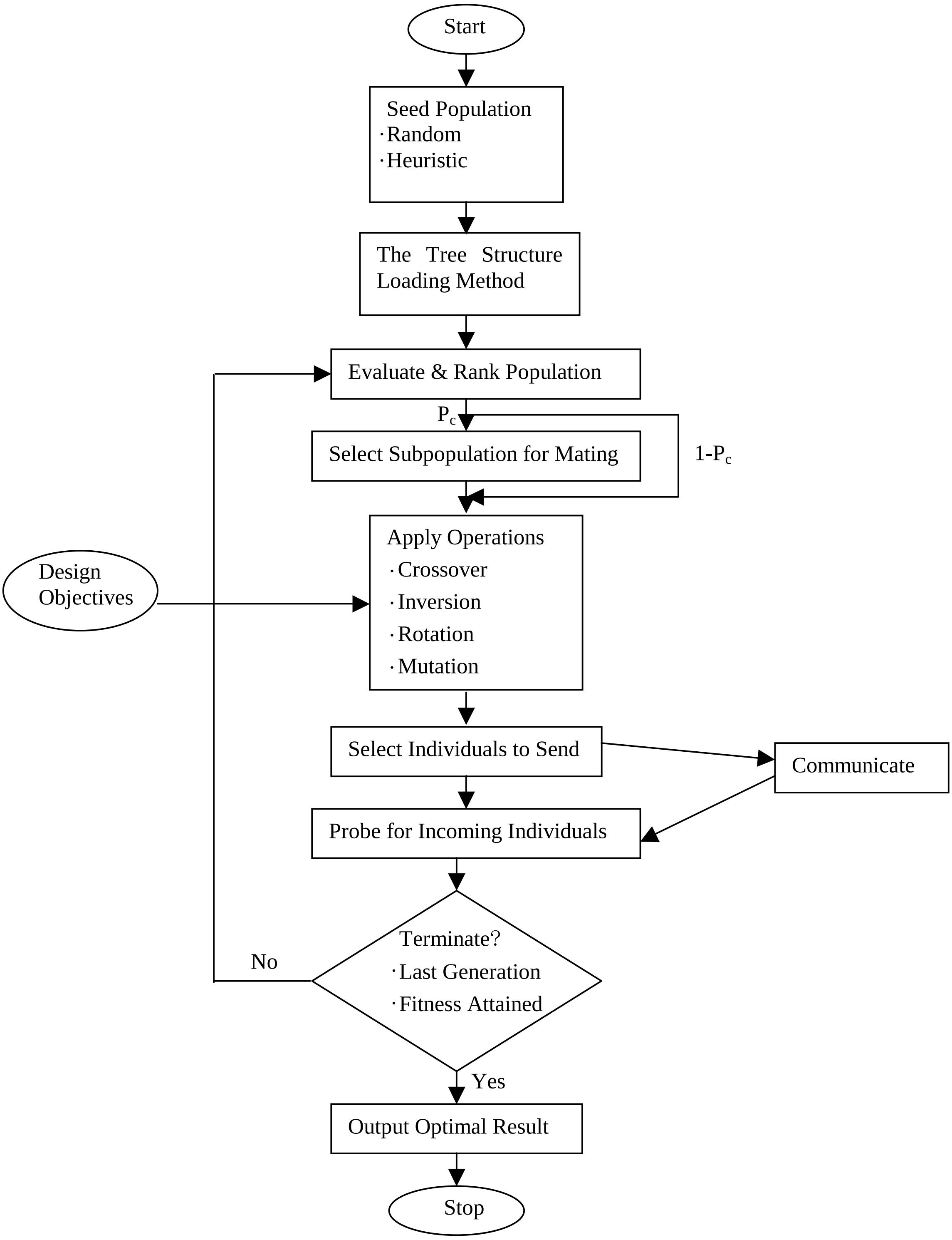
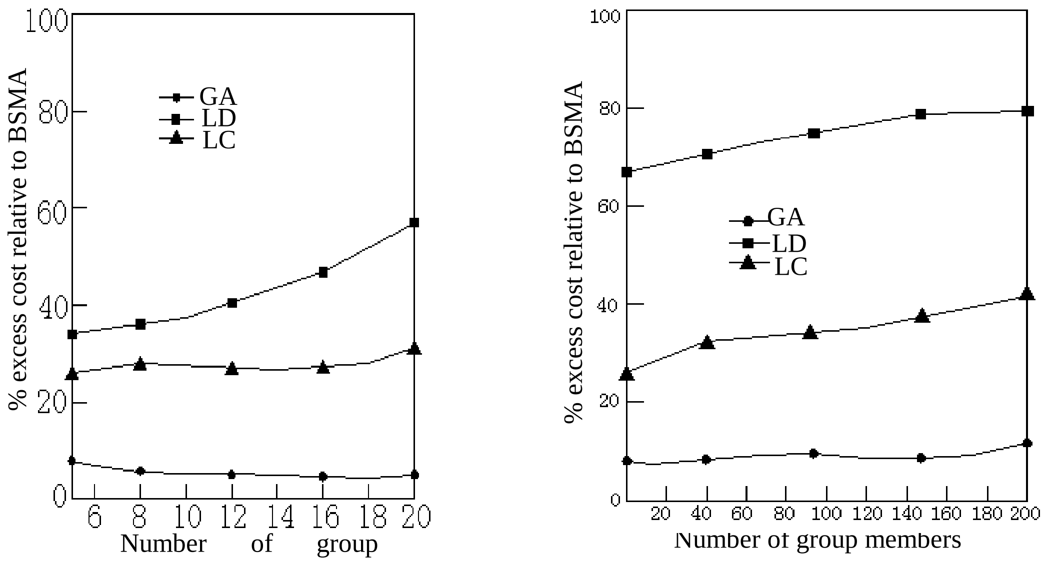


Fig. 2: The flow diagram of solving multicast routing based on GA

Fig. 3 shows the percentage increase in total cost of constrained MC tree relative to BSMA. GA heuristic yields very low tree cost. GA's costs are approximately 10% worse than BSMA. LC's costs are up to 30% worse than BSMA. LD yields the most expensive trees, and the cost of the least delay trees is independent of the range of the link loads. We repeated the same experiment using larger networks. Fig. 3 indicates that the cost performance of the GA relative to each other remain approximately unchanged as the network size increases. GA yields the best performance because it has the ability to locate the lowest cost links in the network and include them in the MC tree.



(a) (b)

Fig. 3: Total cost of an MC tree relative to optimal constrained, 20 nodes, average degree four:
(a) N = 20 nodes, (b) N = 200 nodes

4.2 Routing Request Success Ratio

In the second experiment, we started with a completely unloaded network and kept adding MC sessions and constructing the corresponding MC trees until the cumulative tree failure rate exceeded 15%. Our objective here was to determine how efficiently these constrained algorithms manage the network in the absence of a delay bound. The experiment was repeated until the confidence interval for the number of successfully established MC sessions was <5% using the 95% confidence level. Similar to the first experiment, in this experiment a random network topology was generated before each run.

Fig. 4 shows LD and LC can also manage the network resources efficiently, although not as efficiently as GA, and GA is the best.

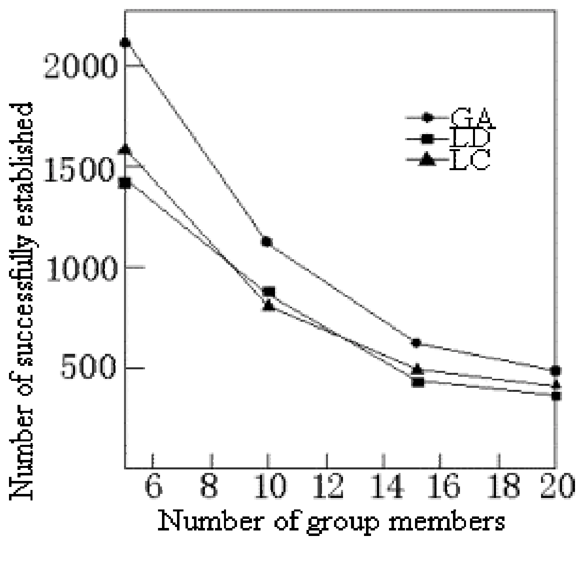


Fig. 4: Number of successful session, unconstrained algorithms, 20 nodes, average degree four, no delay constraint

4.3 Operation Time Testing

With the number of nodes varying in [20, 200], the operation time of GA algorithm is not added much as shown in Table 1. This proves the results of the operation time testing of GA are satisfied.

Table 1: The operation time testing of GA

Nodes	Routers	CPU (time)	Variations
20	32	0.28	0
40	95	0.45	1
80	180	1.20	0
100	188	2.35	1
120	230	3.86	0
140	288	5.96	2
160	328	7.23	1
180	370	9.95	3
00	429	9.56	1

5.0 CONCLUSION AND FUTURE WORK

Distributed real-time applications with QoS constraints will extensively utilise the resources of high-speed networks in the near future. In this paper, we proposed a heuristic genetic algorithm for distributed multicast routing. We studied the bandwidth, delay, delay jitter, and packet loss-constrained least-cost routing problem which is known to be NP-complete. With this algorithm, a dynamic constructed multicast routing tree which has a near optimal network cost under the delay bound constraint can be constructed real time. Simulations have verified that this algorithm is a simple, efficient, distributed multicast routing algorithm that scales well to large network sizes.

We plan to study the performance of the algorithms for a wider range of network architectures. The specification and analysis of algorithms that minimise the tree cost subject to delay and delay variation constraints should be explored in future research.

REFERENCES

[1] V. P. Kompella, J. C. Pasquale, G. C Polyzos, “Multicast Routing for Multimedia Communication”. *IEEE/ACM Transactions on Networking*, 1 (3) : 86-293, 1993.

[2] Hyo-Slik Yang, Martin Maier, Martin Reisslein, and W. Mattehew Carlyle, “A Genetic Algorithm Based Methodology for Optimizing Multi-Service Convergence in a Metro WDM Network”. *Tech. Rep. Telecomm. Research Centre*, Arizona State University, (4), 2002, 1-16.

[3] Hussein F. Salama, Douglas S. Reeves, and Yannis Viniotis, “Evaluation of Multicast Routing Algorithms for Real-Time Communication on High Speed Networks”. *IEEE Journal on Selected Areas in Communications*. Vol. 15, No. 3 (4), 1997, pp. 332-344.

[4] Z. Wang, J. Crowcroft, “Quality of Service for Supporting Multimedia Applications”. *IEEE Journal on Selected Areas in Communications*, 14 (7) : 1228-1234, 1996.

[5] Wang Zheng-ying, Shi Bing-xin, Liu Wei, “A Distributed Dynamic-Constrained Least-Cost Multicast Routing Heuristic”. *Journal of Software*, China, 12 (1) 1-10, 2001.

[6] Chen Ming, Li Zhi-jie, "A Real-Time Multicast Routing Algorithm Based on Genetic Algorithms". *Journal of Software*, China, 12 (5) 729-734, 2001.

[7] Wang Zheng-ying, Shi Bing-xin, "Solving QoS Multicast Routing Problem Based on Heuristic Genetic Algorithm", *Chinese J. Computer* (55-61) 2001.

[8] Z. Wang, J. Crowcroft, "Quality of Service for Supporting Multimedia Applications". *IEEE Journal on Selected Areas in Communications*, 14 (7) : 1228-1234, 1996.

[9] B. M. Waxman, "Routing of Multipoint Connections". *IEEE SAC*, 6 (9) : 1617-1622, 1988.

[10] M. Doar, I. Leslie, "How Bad is Native Multicast Routing". *IEEE INFOCOM 93*, San Francisio, *http://www.ieee-inform.org/1993.*

[11] M. Parsa, Q. ZHU, J. J Garcia-Luna-Aceves, "An Iterative Algorithm for Delay-Constrained Minimum-Cost Multicasting". *IEEE/ACM Transactions on Networking*, 6 (4) : 461-474 1998.

The research is supported by the Natural Science Foundation of China under the grant No. 50274080.

BIOGRAPHY

Youwei Yuan was born in 1966. He is an Associate Professor in China. He is working at the Department of Computer Science and Technology, Zhuzhou Institute Engineering, Hunan, China. His current research interests are high-speed computer network, protocol engineering, and intelligent computer network. He has published more than 20 papers in journals.

Mustafa Mat Deris received his Ph.D in Computer Science from Universiti Putra Malaysia. He is an Associate Professor of Computer Science in the Department of Computer Science, KUSTEM. His current research focuses on distributed database and data replication, computer network performance and data mining. He has published more than 30 papers in journals and conferences. He is a member of IEEE Computer Society.

Lamei Yan is an Associate Professor in China. She is now working at the Department of Computer Science and Technology, Zhuzhou Institute Engineering, Hunan, China. Her research interest are in image processing and pattern recognition. She has published more than 30 papers in journals and proceedings.