

Route Reinforcement for Efficient QoS Routing Based on Ant Algorithm

Jae Seuk Oh, Sung-il Bae, Jin-ho Ahn, Sungho Kang

Dept. of Electrical and Electronic Engineering, Yonsei University
132 Shinchon-Dong, Seodaemoon-Gu, 120-749, Seoul, Korea
{jay3204, sominaby}@soc.yonsei.ac.kr,
{anet0727, shkang}@yonsei.ac.kr

Abstract. In this paper, we present a new method to calculate reinforcement value in QoS routing algorithm for real-time multimedia based on Ant algorithm to efficiently and effectively reinforce ant-like mobile agents to find the best route toward destination in a network. Simulation results show that the proposed method realizes QoS routing more efficiently and more adaptively than those of the existing method thereby providing better solutions for the best route selection.

1 Introduction

As Internet expands, the demand for real time and quality of services (QoS) in a network increases. The quality of services are sensitive to the network's characteristics such as bandwidth, delay, delay jitter, packet loss and cost depending on the type of applications. Furthermore, the use of multiple metrics is needed to better characterize a network and to support a wide range of QoS requirements [1].

Ant algorithm is a routing algorithm, which is inspired by the trail following behavior of real ant colonies, and this algorithm realizes an adaptive and social behavior of ants of finding the best route to the food source from the nest by indirect communications between ants using a chemical substance called pheromone [2,3]. This algorithm is proved to be very effective in terms of solving Traveling Salesman Problem [4], Quadratic Assignment Problem [5], and etc... However, thus far, there haven't been many researches that are done to realize QoS routing based on Ant algorithm, leaving large room for improvements and explorations.

The main purpose of this paper is to propose a new method to calculate reinforcement value reflecting all the necessary QoS metrics to better realize an adaptive behavior of Ant algorithm for real-time applications.

This paper is organized as follows. In section 2, QoS routing algorithm based on Ant algorithm is introduced. In section 3, detailed description of the proposed method of reinforcement calculation under the QoS routing algorithm described in

section 2 is introduced. In section 4, experimental results are presented; and at last, conclusion from this research is drawn out in section 5.

2 Ant Algorithm Based QoS Routing

In an event of collecting foods, ants travel from the nest (source node) to a location where food source is placed (destination node), and come back. During their trips, each ant indirectly communicates with each other by depositing a chemical substance called, *pheromone*, on the trail to attract other ants to the path it took to get to the destination. Initially, ants take various paths to their destination, the food source, because none or an insignificant amount of pheromone to attract ants is deposited on any of the possible paths as shown in Fig. 1.

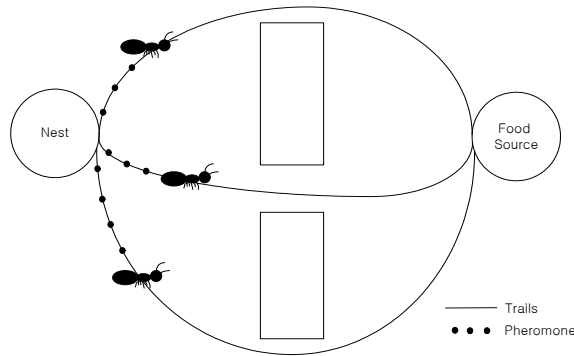


Fig. 1. Initial search of ants for the shortest path to food source (destination)

As more and more ants travel to the same location, a path with the shortest distance between the nest and the food source gets more pheromone deposits. This is due to what is called pheromone decay. Pheromone decays with time; therefore, longer the distance means more pheromone gets decayed in that path. Thus, a path with the shortest trip time or the shortest distance eventually left with the most pheromone, which in turn attracts more and more ants as shown in Fig. 2. Now ants have established or found the shortest path to their destination.

In Ant algorithm, there are two types of mobile agents, forward and backward ants that mimic the aforementioned behaviors of ants. The forward ant is first sent out from a source node to its randomly selected destination, and it gathers the network information of the path it took to the destination, which will be used to update routing table later on.

The backward ant is created at the destination node once the forward ant safely arrives. The backward ant literally goes back to the source node by taking the same path that the forward ant took. On its way back, it pops the stored network information to know which node to travel next and also to update routing table of the node it is currently standing. The backward ant repeats this until it arrives back at the source

node, and dies. This is very brief and basic explanation of Ant algorithm and with this in mind, QoS routing based on Ant algorithm in [1] is as follows.

There are T sets of ants where every set is consisting of M types of ants belonging to M different call requirements, where each ant type must find the best path to its destination that satisfies all the requirements. Furthermore, the properties of pheromone deposits of each ant type are different from each other, so that an ant selects a route relying only on the pheromone deposited by the same type of ant on the path.

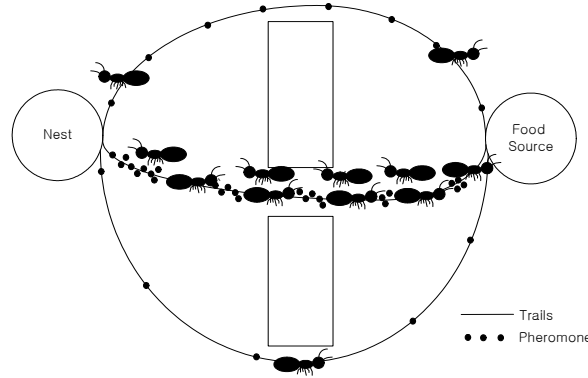


Fig. 2. After certain number of trips ants found the best path to food source (destination)

To speed up the process, this algorithm makes some adjustments: considering the delay jitter constraint outside the ant algorithm, and filter the topology of the network by canceling the edges that do not satisfy the bandwidth constraint [1]. These modifications are done since this algorithm is mainly targeting real-time applications, which put large emphasis on delay jitter and bandwidth. The steps needed to take in the algorithm are described below.

At first, if actual end-to-end delay jitter is greater than the delay jitter constraint, then routing fails; if not, then next step is to eliminate the link that does not satisfy the bandwidth constraint. These first two steps are the modifications to speed up the algorithm.

Next, the amount of pheromone deposited on every edge in the network topology is initialized for every type of ants. Then a set of ants of every type is sent out one at a time at constant interval toward corresponding destinations to collect network information, while choosing its path by repeatedly applying the state transition rule.

After the set of ants has chosen the paths between corresponding source nodes and destination nodes successfully, the amount of every type of pheromone on every path is adjusted by using the local updating rule [1]. The previous two steps get repeated for another set of ants, until all sets of ants finish the step.

Next, choose the globally best ant of each type, and then use the global updating rule to adjust amount of pheromone on each path in the route that is selected by the best ant of each type.

Finally, the steps after the initialization step are repeatedly performed, until the accuracy requirement is satisfied [1].

In order to follow the above-mentioned steps of Ant algorithm based QoS routing algorithm, the state transition rule and the pheromone-updating rule are proposed. Under the state transition rule proposed in [1], a d type of an ant at node r selects a next node s to travel according to following rule

If $q \leq q_0$, then

$$\rho_d(r, s) = \begin{cases} 1 & \max(\text{phero}(d, r, s)), s \in J_d(r) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

otherwise,

$$\rho_d(r, s) = \begin{cases} \frac{\text{phero}(d, r, s)}{\sum_{u \in J_d(r)} \text{phero}(d, r, u)} & s \in J_d(r) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where, q_0 represents a constant value that lies between 1 and 0, which is use to compare with q , a randomly chosen number between 1 and 0, to determine how often either (1) or (2) is used to determine the probability of choosing the next node from the current node out of all neighbor nodes that lead to the destination.

The pheromone-updating rule is further divided into two sub-rules: local updating rule and global updating rule, the concept was first proposed in [3]. Under the local updating rule, suppose a d type ant at node r chooses a neighbor node s as the next node to travel, the amount of pheromone $\text{phero}(d, r, s)$ is adjusted in accordance with equation (3), otherwise no pheromone amount gets adjusted

$$\text{phero}(d, r, s) = (1 - a_0) \bullet \text{phero}(d, r, s) + a_0 \bullet \text{cons} \quad (3)$$

where, a_0 is a value between 0 and 1, and cons is a constant. In this way, ants will make a better use of pheromone information; without the local updating, all ants would search in a narrow neighborhood of the best previous path.

The global updating rule is used when the globally best path to the destination is found. Once the globally best path is determined, pheromone amounts of edges between all nodes in the globally best path are adjusted in accordance with equation (4), and pheromone amounts of all the other edges get adjusted by equation (5)

$$\text{phero}(d, r, s) = (1 - a_1) \bullet \text{phero}(d, r, s) + a_1 \bullet F \quad (4)$$

$$\text{phero}(d, r, s) = (1 - a_1) \bullet \text{phero}(d, r, s) \quad (5)$$

where, a_1 is a value between 0 and 1, and F is the cost function, and it plays the same role as the reinforcement value in AntNet^[2] for adjusting pheromone amount deposited in the path in accordance with the network conditions experienced by an ant that found the globally best path toward the destination. Furthermore, the value of F is calculated by following equations:

$$F = -F_1 + F_2 \quad (6)$$

where,

$$F_1 = \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N LC_{ij} \bullet P_{ij}^d . \quad (7)$$

$$F_2 = A \bullet \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N H(Z_{ij}) + B \bullet H(Z_2) + C \bullet H(Z_3) . \quad (8)$$

where, N is the node number, A , B and C are positive real coefficients to indicate importance of each term in reinforcement calculation according to the QoS constraints. Here, F_1 represents the total cost of the route selected by an ant; and F_2 represents the QoS constraints.

$H(Z) = 0$, if $Z < 0$, otherwise $H(Z) = Z$.

$$Z_{ij} = P_{ij}^d \bullet LB_{ij} - B_w . \quad (9)$$

$$Z_2 = D_w - (\sum LD_{ij} \bullet P_{ij}^d + \sum N_i^d \bullet ND_i) . \quad (10)$$

$$Z_3 = \prod_{i=1}^N (1 - N_i^d \bullet NL_i) - (1 - L_w) . \quad (11)$$

Here, $P_{ij}^d = 1$ if an edge between node i and node j is an edge in the d type ant selected route, otherwise $P_{ij}^d = 0$. $N_i^d = 1$ if node i is the node in the d type ant selected route, otherwise $N_i^d = 0$. Symbols LB_{ij} , LC_{ij} and LD_{ij} are the available bandwidth, cost and delay of an edge between node i and node j respectively, and L_w , B_w and D_w represent link, bandwidth and delay constraints respectively

3 Proposed Method to Reinforce Route

As described in the previous sections, [1] calculates the differences between actual QoS measurements and QoS constraints in the process of calculating reinforcement value. However, the equations described in the previous section do not realize the adaptive behavior well. Unlike [1], the proposed reinforcement calculation uses ratios between QoS measurements and QoS constraints. The proposed method to calculate the reinforcement value is as follows.

If ($F_2 \geq 1$),

$$F = F_2 - k \left(\frac{F_1}{F_2} \right) . \quad (12)$$

else

$$F = F_2 - k \bullet F_1 . \quad (13)$$

$$F_1 = \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N LC_{ij} \bullet P_{ij}^d . \quad (14)$$

$$F_2 = A \bullet \min(Band_{ij}) + B \bullet Dly + C \bullet PLR + D \bullet DJ \quad (15)$$

where $i \neq j$, and $i \in P$ and $j \in P$
 $P = \text{All nodes in the chosen path}$

where, F is the cost function or the reinforcement; F_1 is the total cost of the route; F_2 is the QoS constraints; and k is weight constant for cost to indicate its importance compare to other QoS metrics.

$$Band = \begin{cases} \frac{B_{mea}}{B_w}, & \text{where } B_{mea} \geq B_w \\ \left(\frac{B_{mea}}{B_w} \right)^2, & \text{where } \text{toleration rate} < \frac{B_{mea}}{B_w} < 1 \\ 0, & \text{where } \frac{B_{mea}}{B_w} < \text{toleration rate} \end{cases} \quad (16)$$

$$Dly = \begin{cases} \frac{D_w}{\sum D_{mea}}, & \text{where } \sum D_{mea} \leq D_w \\ \left(\frac{D_w}{\sum D_{mea}} \right)^2, & \text{where } \text{toleration rate} < \frac{D_w}{\sum D_{mea}} < 1 \\ 0, & \text{where } \frac{D_w}{\sum D_{mea}} < \text{toleration rate} \end{cases} \quad (17)$$

$$PLR = \begin{cases} \frac{1 - (1 - L_w)}{1 - \prod (1 - L_{mea})}, & \text{where } \prod (1 - L_{mea}) \geq (1 - L_w) \\ \left(\frac{1 - (1 - L_w)}{1 - \prod (1 - L_{mea})} \right)^2, & \text{where } \text{toleration rate} < \frac{1 - (1 - L_w)}{1 - \prod (1 - L_{mea})} < 1 \\ 0, & \text{where } \frac{1 - (1 - L_w)}{1 - \prod (1 - L_{mea})} < \text{toleration rate} \end{cases} \quad (18)$$

$$DJ = \begin{cases} \frac{J_w}{\sum DJ_{mea}}, & \text{where } \sum J_{mea} \leq J_w \\ \left(\frac{J_w}{\sum J_{mea}} \right)^2, & \text{where } \text{toleration rate} < \frac{J_w}{\sum J_{mea}} < 1 \\ 0, & \text{where } \frac{J_w}{\sum J_{mea}} < \text{toleration rate} \end{cases} \quad (19)$$

Among equations (12) through (19), equation (15) calculates the amount of positive influence the QoS measurements have on the reinforcement calculation by considering the goodness of each QoS measurement compare to its constraint, and their rate of importance in the calculation [5]. The goodness of each QoS measurement is calculated using equations (16) through (19).

Equations (16) through (19) also have a term “*toleration rate*.” Toleration rates are set individually for each QoS metric with values between 0 and 1. Each toleration rate represents percentage of negative discrepancy that the QoS metric can tolerate. However, in the case of having no paths satisfying the QoS constraints, *toleration*

rate can be used to find a path that provides a decent level of quality of service but with some service degradation.

Equations (12) and (13) are the top level calculation of reinforcement. Equation (12) subtracts the total amount of goodness of all QoS measurements by the fractional value of the ratio between the total cost of the chosen path and the total amount of goodness of all QoS measurements. Equation (13) is for when F_2 less than 1.

In addition to the reinforcement calculation, the global update rule is also proposed to get the best out of the proposed reinforcement calculation.

$$phero(d, r, s) = (1 - a_1) \bullet phero(d, r, s) + c \bullet F \quad (20)$$

where, c is coefficient of value between 0 and 1.

Equation (20) is used to adjust pheromone amount on the globally best route, and equation (21) shown below is a new method to adjust pheromone amount of less qualified paths that were chosen in the process.

$$phero(d, r, s) = (1 - a_1) \bullet phero(d, r, s) - d \bullet (F - F_{others}) \quad (21)$$

Here d is weight coefficient and F_{others} is the reinforcement value of less qualified path. Since the reinforcement value under the proposed method is relative to the delay size, if there exist some routes that have reinforcement values almost as good as the globally best route, then the pheromone amounts of the nearly good routes reduce at a rate close to $(1 - a_1) \bullet phero(d, r, s)$.

4 Simulation

Fig. 3 depicts the topology of a network system, which consists of eight nodes and twelve edges connecting the nodes. This topology is adopted from the simulation environment used in [1] to ensure that the simulation results of the existing method in this experiment agree with the results presented in [1].

The values in the parenthesis near each node in Fig. 3 represent node delay, packet loss rate and node delay jitter respectively; whereas, the values in the parenthesis near each edge represent link cost, bandwidth and link delay respectively.

The QoS requirements are set to $B_w = 70$, $D_w = 8$, $L_w = 10^{-5}$ and $J_w = 3$. For the simulation of [1], the simulation parameters are set to T (sets of ants) = 8, M (call requests) = 3, $a_0 = 0.069$, $a_1 = 0.079$, $cons = 0.32$, $q_0 = 0.20$, $A = 0$, $B = 10$ and $C = 15$. For the simulation of the proposed method, most of the simulation parameters are set exactly same as in the existing method; however, since the calculation method is different from the existing method, some new parameters are introduced. And those parameters are set to $C = 0.01$, $D = 0$, $k = 1$, $c = 0.079$, $c_l = 0.02$, $\alpha = 1$ and $\beta = 1$; and these are set relative to [1].

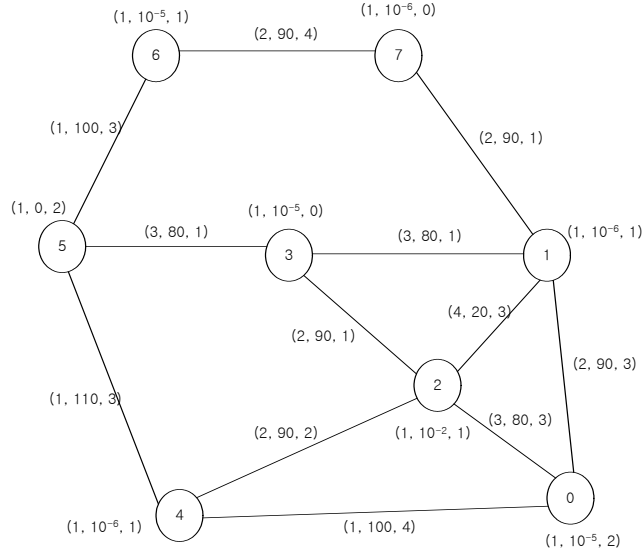


Fig. 3. Network Topology model and its parameters used for simulation^[1]

At first, several unicast routing requests: node 0 to node 5, node 1 to node 5, node 1 to node 6, node 2 to node 6, node 2 to node 7, node 5 to node 7, and node 4 to node 7 are assumed. By simulation, the globally best routes are found for each method as shown in Table 1.

Table 1. Average iterations to find the globally best route

Routing Requests (s, d)	Existing method of [1]		Proposed method w/ TR=0.99		Proposed method w/ TR=1	
	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.
(0, 5)	0 → 1 → 3 → 5 0 → 2 → 3 → 5	62.2	0 → 1 → 3 → 5	31.4	0 → 1 → 3 → 5 0 → 2 → 3 → 5	36.9
(1, 5)	1 → 3 → 5	27.2	1 → 3 → 5	25.2	1 → 3 → 5	25.5
(1, 6)	1 → 7 → 6	36.8	1 → 7 → 6	31.4	1 → 7 → 6	31.5
(2, 6)	2 → 3 → 5 → 6	48.7	2 → 3 → 5 → 6	29.7	2 → 3 → 5 → 6	29.2
(2, 7)	2 → 3 → 1 → 7	30.6	2 → 3 → 1 → 7	28.5	2 → 3 → 1 → 7	28.3
(5, 7)	5 → 3 → 1 → 7	32.6	5 → 3 → 1 → 7	27.7	5 → 3 → 1 → 7	27.1
(4, 7)	N/A	N/A	4 → 5 → 3 → 1 → 7	27.5	4 → 5 → 3 → 1 → 7	27.8

If we take a good look at Fig. 3, for the routing request (0,5), there exists a route 0 → 4 → 5 that satisfies all the QoS requirements and has the cost of 2; however, the

reason why the route $0 \rightarrow 1 \rightarrow 3 \rightarrow 5$ or $0 \rightarrow 2 \rightarrow 3 \rightarrow 5$ is selected as the best route is because we set the delay term to be more sensitive than the cost.

Looking at Table 1, we can see that the proposed method finds the best routes faster than the existing method. Faster convergence to the best route does not prove the effectiveness of the proposed method, since the faster convergence can be simply achieved through making the reinforcement value to be large. However, under the proposed method, faster convergences are achieved for all the routing requests, where the reinforcement values of some the routing requests are considerably less than those of the existing method. This is due to the adaptive behavior that is realized in the proposed method of the global update using equations (20) and (21).

Next, an experiment is performed to see how the proposed method dynamically adjusts the reinforcement value, by comparing the simulation results of one with small delay constraint and the other with large delay constraint, thereby identifying the advantages of the proposed method.

To simulate one with small delay constraint, the QoS information on edges between node 5 and 6, and node 6 and 7 are changed to (1, 100, 0) and (2, 90, 1) respectively; and to simulate one with large delay constraint, the QoS information on edges between node 0 and 1, 1 and 3, 2 and 3, 1 and 7, 5 and 6, and 6 and 7 are changed to (2, 90, 93), (3, 80, 91), (2, 90, 91), (2, 90, 90), (1, 100, 1) and (2, 90, 1) respectively. Simulation results of both figures under the existing method and the proposed method are shown in Table 2.

Table 2. Simulation result of routing request (1,5) under two different constraints

	Existing method of [1]		Proposed method	
	Selected Route	Average Iteration	Selected Route	Average Iteration
Under Small Delay Constraint ($D_w = 8$)	$1 \rightarrow 3 \rightarrow 5$	27.63	$1 \rightarrow 3 \rightarrow 5$	25.22
Under Large Delay Constraint ($D_w = 98$)	$1 \rightarrow 3 \rightarrow 5$	27.47	$1 \rightarrow 3 \rightarrow 5$	34.53

Assuming the routing request of (1,5), the globally best route should be $1 \rightarrow 3 \rightarrow 5$ since it satisfies the constraints and has the smallest delay measurement. However, the route $1 \rightarrow 7 \rightarrow 6 \rightarrow 5$ also satisfies the constraints and has the end-to-end delay of 4 and 94, just one unit of delay more than $1 \rightarrow 3 \rightarrow 5$ in both cases. If we look at Table 2, the average iterations of two different cases under the existing method are nearly identical to each other. However, the average iterations of two different cases under the proposed method show some discrepancy.

The reason for such discrepancy is due to how the reinforcement values are calculated in the methods. Under the existing method, amount of impact a unit of delay exerts on the reinforcement value is constant regardless of delay size. However, in the proposed method, such fact is realized, and that is why it takes more iterations to converge when the delay size and the constraint are large.

One last experiment is done using network topology with edges between node 0 and 1, 1 and 3, 2 and 3, 1 and 7, 5 and 6, and 6 and 7 having QoS information of (2,90,93), (3,80,91), (2,90,99), (2,90,90), (1,100,1) and (2,90,10) respectively. As-

suming the routing request (1,5), the simulation result is compared with that of the previous experiment under large delay constraint.

Table 3. Simulation result of the last experiment

Routing Requests (s, d)	Under large delay constraint			
	Existing method in [1]		Proposed method	
	Selected Route	Average Iteration	Selected Route	Average Iteration
(1, 5)	1 → 3 → 5	27.47 (27.47)	1 → 3 → 5	28.53 (34.53)

If we compare the results in Table 2 with Table 3, the average iteration of [1] is the same, while the average iteration of the proposed method is reduced. The reason for such reduction in number of iteration under the proposed method is due to equation (21), which reflects more of an adaptive behavior. Since the reinforcement value under the proposed method is relative to the delay size, if there exists some routes that have reinforcement values almost as good as the globally best route, then the pheromone amounts of the nearly good routes reduce at a rate close to $(1 - a_1) \bullet phero(d, r, s)$, which was the case for the previous experiment.

5 Conclusions

This paper has presented with an adaptive method to reinforce route in QoS routing algorithm based on Ant algorithm. Under the proposed method, the reinforcement value is calculated using the fractional ratios between the measurements and the constraints, which in turn provides with the reinforcement value relative to the measurement's size and the constraint's range. And with further changes in the global update equations, the proposed method realizes more adaptive behaviors than the existing method, providing faster convergence time.

References

1. Zhang, S., Liu, Z.: "A QoS Routing Algorithm Based on Ant Algorithm," 25th Annual IEEE Conference on Local Computer Networks, pp. 574-578, (2000)
2. Dorigo, M., Di Caro, G.: "AntNet: Distributed Stigmergetic Control for Communications Networks," Journal of Artificial Intelligence Research, Number 9, pp. 317-365, (1999)
3. Quadros, G., Monteiro, E., Boavida, F.: "A QoS Metric for Packet Networks," Proc. of SPIE International Symp. on Voice, Video, and Data Communications Conference, (1998)
4. Dorigo, M., Gambardella, L. M.: "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Trans. on Evolutionary Computation, pp. 53-66, (1997)
5. Stutzle, T., Dorigo, M.: "ACO algorithms for the quadratic assignment problem," New Ideas in Optimization, pp. 33-50, McGraw Hill, (1999)

6. Chu, C., Gu, J., Hou, X., Gu, Q.: "A Heuristic Ant Algorithm for Solving QoS Multicast Routing Problem," Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, Vol. 2, pp. 12-17, (2002)