

## Multiple Constraints for Ant Based Multicast Routing in Mobile Ad Hoc Networks

<sup>1</sup>A. Sabari and <sup>2</sup>K. Duraiswamy

<sup>1</sup>Department of Information Technology, KS Rangasamy College of Technology, Tiruchengode, India

<sup>2</sup>Department of Computer Science and Engineering, KS Rangasamy College of Technology,  
Tiruchengode, India

---

**Abstract: Problem statement:** A Mobile Ad hoc Network (MANET) is one of the challenging environments for multicast. Since the associated overhead is more, the existing studies illustrate that tree-based and mesh-based on-demand protocols are not the best choice. The costs of the tree under multiple constraints are reduced by the several algorithms which are based on the Ant Colony Optimization (ACO) approach. The traffic-engineering multicast problem is treated as a single-purpose problem with several constraints with the help of these algorithms. The main disadvantage of this approach is the need of a predefined upper bound that can isolate good trees from the final solution. **Approach:** In order to solve the traffic engineering multicast problem which optimizes many objectives simultaneously this study offers a design on Ant Based Multicast Routing (AMR) algorithm for multicast routing in mobile ad hoc networks. **Results:** Apart from the existing constraints such as distance, delay and bandwidth, the algorithm calculates one more additional constraint in the cost metric which is the product of average-delay and the maximum depth of the multicast tree. Moreover it also attempts to reduce the combined cost metric. **Conclusion:** By simulation results, it is clear that our proposed algorithm surpasses all the previous algorithms by developing multicast trees with different sizes.

**Key words:** MANETs multicast routing, ant colony optimization

---

### INTRODUCTION

A Mobile Ad Hoc Network (MANET) is a kind of wireless ad hoc network and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links-the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Mobile ad hoc networks became a popular subject for research as laptops and 802.11/Wi-Fi wireless networking became widespread in the mid to late 1990s.

A Mobile Ad hoc Network (MANET) is a set of mobile nodes which communicate over radio and do not need any infrastructure<sup>[1]</sup>. This kind of networks are very flexible and suitable for several situations and applications, thus they allow the establishing of temporary communication without pre-installed infrastructure. The interfaces exhibit limited transmission range to facilitate communication between

two nodes. Many intermediate nodes have been involved to relay communication traffic. Therefore, this kind of networks is also called mobile multi-hop ad-hoc networks.

In order to transmit data to a subset of destination nodes in a computer network multicast consists of simultaneous data transmission from a source node. Multicast routing algorithms are used in radio and TV transmission, on demand video and teleconference. End-to-end delay, minimum bandwidth resources and cost of the tree are the main QoS parameters which are included in the multicasting. Thus the traffic engineering multicast problem should be treated as a multi-objective problem.

In this study, an Ant Based Multicast Routing (AMR) algorithm for multicast routing in mobile ad hoc networks has been proposed to solve the Traffic Engineering Multicast problem that optimizes several objectives simultaneously. This algorithm calculates one more additional constraint in the costs metric which is the product of average-delay and the maximum depth of the multicast tree and tries to minimize this combined cost metric.

---

**Corresponding Author:** A. Sabari, Department of Information Technology, KS Rangasamy College of Technology, Tiruchengode, India

**Related work:** Some algorithms which have elements in common with our algorithm, such as multipath routing, data load spreading and proactive path maintenance have been analyzed. The basic idea behind ACO algorithms for routing is the use of mobile agents, called ants.

Gunes *et al.*<sup>[1]</sup> proposed a new on-demand routing algorithm for mobile, multi-hop ad hoc networks. The algorithm is based on ant algorithms which are a class of swarm intelligence. Ant algorithms try to map the solution capability of ant colonies to mathematical and engineering problems. The Ant-Colony-Based Routing Algorithm (ARA) is highly adaptive, efficient and scalable. The main goal in the design of the algorithm was to reduce the overhead for routing.

Pitakaso *et al.*<sup>[5]</sup> an ant-based algorithm for solving unconstrained multi-level lot-sizing problems called ant system for multi-level lot-sizing algorithm (ASMLLS). A hybrid approach which uses ant colony optimization in order to find a good lot-sizing sequence and a simple single stage lot-sizing rule is applied with modified setup costs. They have modified the setup costs depends on the position of the item in the lot-sizing sequence, on the items which have been lot-sized before and on two further parameters, which are tried to be improved by a systematic search.

Baras *et al.*<sup>[6]</sup> proposed a novel approach to the routing problem in MANETs by using swarm intelligence inspired algorithms. The proposed algorithm uses Ant-like agents to discover and maintain paths in a MANET with dynamic topology.

Kazuyuki Fujita<sup>[7]</sup> proposed an Ants-Routing with routing History (ARH) and Ants-Routing with routing history and no return rule (ARHnr), that can perform a robust routing by selecting stochastically the good route and learn quickly the route by using routing history. ARH and ARHne adapt reinforcement learning to the routing algorithm.

Matsuo *et al.*<sup>[8]</sup> accelerated Ants-Routing which increase convergence speed and obtain good routing path is discussed. Experiment on dynamic network showed that accelerated Ants-Routing learns the optimum routing in terms of convergence speed and average packet latency.

Marwaha *et al.*<sup>[9]</sup> overcome these shortcomings of ant-based routing and AODV by combining them to develop a hybrid routing scheme. The Ant-AODV hybrid routing protocol is able to reduce the end-to-end delay and route discovery latency by providing high connectivity as compared to AODV and ant-based routing schemes.

Heissen Büttel *et al.*<sup>[10]</sup> addressed the problem of routing in large-scale Mobile Ad-Hoc Networks (MANETs), both in terms of number of nodes and coverage area. Our approach aims at abstracting from the dynamic, irregular topology of a MANET to obtain a topology with “logical routers” and “logical links”, where logical router and logical links are just a collection of nodes and (multihop) paths between them, respectively. To “build” these logical routers, nodes geographically close to each other are grouped together. Logical links are established between selected logical routers.

Sinha *et al.*<sup>[11]</sup> proposed the MCEDAR multicast routing algorithm for ad hoc networks. MCEDAR is an extension to the CEDAR architecture and provides the robustness of mesh based routing protocols and approximates the efficiency of tree based forwarding protocols. It decouples the control infrastructure from the actual data-forwarding infrastructure. The decoupling allows for a very minimalistic and low overhead control infrastructure while still enabling very efficient data forwarding.

Devarapalli *et al.*<sup>[12]</sup> proposed a new multicast protocol for mobile ad hoc networks, called the Multicast routing protocol based on Zone Routing (MZR). MZR is a source-initiated on demand protocol, in which a multicast delivery tree is created using a concept called the zone routing mechanism. The protocol’s reaction to topological changes can be restricted to a node’s neighborhood instead of propagating it throughout the network.

Vaishampayan *et al.*<sup>[13]</sup> proposed the Protocol for Unified Multicasting through Announcements (PUMA) in ad hoc networks, which establishes and maintains a shared mesh for each multicast group, without requiring a unicast routing protocol or the pre assignment of cores to groups. PUMA achieves a high data delivery ratio with very limited control overhead, which is almost constant for a wide range of network conditions.

Jetcheva *et al.*<sup>[14]</sup> proposed the design and initial evaluation of the Adaptive Demand Driven Multicast Routing protocol (ADMR), a new on demand ad hoc network multicast routing protocol that attempts to reduce as much as possible any non on demand components within the protocol.

Castro *et al.*<sup>[15]</sup> proposed a scalable application-level multicast infrastructure. Scribe supports large numbers of groups, with a potentially large number of members per group. Scribe is built on top of Pastry, a generic peer-to-peer object location and routing substrate overlaid on the Internet and leverages Pastry’s reliability, self-organization and locality properties. Pastry is used to create and manage groups

and to build efficient multicast trees for the dissemination of messages to each group. Scribe provides best-effort reliability guarantees and we outline how an application can extend Scribe to provide stronger reliability.

Zhang *et al.*<sup>[16]</sup> proposed a hybrid multicast scheme in p2p networks. Borg is motivated by the asymmetry in routing in structured p2p networks. The overlay path taken in routing a message from node A to node B is likely to be distinct and therefore has a different routing delay from the path taken in routing a message from node B to node A. Borg exploits this asymmetry by building the upper part of a multicast tree using a hybrid of forward-path forwarding and reverse-path forwarding and leverages the reverse-path multicast scheme for its low link stress by building the lower part of the multicast tree using reverse-path forwarding. The boundary nodes of the upper and lower levels are defined by the nodes' distance from the root in terms of the number of overlay hops.

Nasipuri *et al.*<sup>[17]</sup> proposed a particular on-demand protocol, called Dynamic Source Routing and show how intelligent use of multipath techniques can reduce the frequency of query floods. They develop an analytic modeling framework to determine the relative frequency of query floods for various techniques.

## MATERIALS AND METHODS

**Ant Colony Optimization (ACO):** Ant colony optimization is a probabilistic technique for solving computational problems which can be reduced to find the good paths through graphs. Ants are used as the agents and the routing is on basis of the food searching behavior of the real ants. These agents are divided into forward and backward ants. The sender to the neighbor nodes broadcasts the forward ants. The backward ants utilize the useful information like end-to-end delay, number of hops gathered by the forward ants on their trip from source to the destination.

Ant Colony Optimization (ACO) is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems<sup>[18]</sup>. The first algorithm which can be classified within this framework was presented in 1991. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

The characteristic of ACO algorithms is their explicit use of elements of previous solutions. In fact, they drive a constructive low-level solution, as GRASP

does, but including it in a population framework and randomizing the construction in a Monte Carlo way. A Monte Carlo combination of different solution elements is also suggested by Genetic Algorithms, but in the case of ACO the probability distribution is explicitly defined by previously obtained solution components.

An ACO algorithm includes two more mechanisms: Trail evaporation and, optionally, daemon actions. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective.

It has been experimentally observed that ants in a colony can converge on moving over the shortest among different paths connecting their nest to a source of food. The main catalyst of this colony-level shortest path behavior is the use of a volatile chemical substance called pheromone: Ants moving between the nest and a food source deposit pheromone and preferentially move in the direction of areas of higher pheromone intensity. Shorter paths can be completed quicker and more frequently by the ants and will therefore be marked with higher pheromone intensity. These paths will therefore attract more ants, which will in turn increase the pheromone level, until there is convergence of the majority of the ants onto the shortest path. The local intensity of the pheromone field, which is the overall result of the repeated and concurrent path sampling experiences of the ants, encodes a spatially distributed measure of goodness associated with each possible move.

### Ant-system-based QOS multicasting algorithm: Multicast algorithm:

**Step1: Backup-paths-set:** For each destination node  $m_i \in M$ , Dijkstra K shortest path algorithm is used to compute the least-cost paths from  $s$  to  $m$  to construct backup-paths set. Let  $P_i$  be paths set for destination node  $i$ :

$$P_i = \{P_i^1, \dots, P_i^j, \dots, P_i^k\} \quad (1)$$

where,  $P_i^j$  is the  $j$ th path for destination node  $i$ .

If the delay constraint is violated by some of the trees, then the cost is to be increased, so that it is likely to be rejected.

**Step 2: Tree formation:** In this algorithm, a multicast tree  $T$  is represented as an array of  $m$  elements:

$$T = \{P_1, P_2, \dots, P_m\} \quad (2)$$

Where:

$P_i = P(s, m_i) =$  The path set selected from (1)  
 $S =$  The source and  $m_i$  is the destination

**Step 3: Path selection:** When an ant moves from the node  $i$  to the next node  $j$ , the probability function of the ant choosing node  $j$  as the next node as follows:

$$f_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in N_h(i)} [\tau_{iu}]^\alpha [\eta_{iu}]^\beta} & \text{if } j \in N_h(i) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$\alpha$  and  $\beta$  are the relative importance of pheromone strength and the distance between nodes that affect an ant's judgment when choosing the next node to select.

**Step 4: Pheromone update:** The pheromone trail associated to every edge is evaporated by reducing all pheromones by a constant factor:

$$\tau_{ij} \leftarrow (1 - P) \tau_{ij} \quad (4)$$

where,  $p \in (0,1)$  is the evaporation rate. Next, each ant retracts the path it has followed and deposits an amount of pheromone  $\Delta\tau_{ij}^h$  on each traversed connection:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^h, \quad \alpha_{ij} \in Sh \quad (5)$$

The pheromone on a connective path  $(i,j)$  left by the  $m$ th ant is the inverse of the total length traveled by the ant in a particular cycle. The formula is as follows:

$$\tau_{ij}^h = Q / Lm$$

In the above formula:

$Q =$  A constant  
 $Lm = (C_j - C_i)$

Where:

$C_i =$  cost of sub multicast tree node  $i$   
 $C_j =$  cost of sub multicast tree node  $j$

To avoid the situation of  $C_i = C_j$  compute:

$$Lm = (C_j - C_i)^2 + 1 \quad (6)$$

**Step 5: Stopping criterion:** The stopping criterion of the algorithm could be specified by a maximum number of iterations or a specified CPU time limit.

**Problem formulation:** The multicast tree will be determined on a particular set of nodes in which the delay can be measured between all nodes. A graph representation is considered as  $G = (V,E)$ .  $V$  is the set of all vertices (end systems in the network) and  $E$  is the set of weighted, undirected edges between all nodes. Let us consider only networks in which all nodes are subscribers of the multicast group or one in which non-subscribers can be ignored. Edges are assigned weights corresponding to the bandwidth and delay between the nodes they connect:

- Cost of the tree (C)
- Average end-to-end delay (d)
- Maximum Depth (D)

We will compute the normalized product:

$$P = C * d * D$$

A multicast routing problem tries to find the multicast tree  $T$  that minimizes  $P$ . Bounding the maximum depth of the tree and therefore bounding the maximum hops, is a meaningful metric for networks in which Time To Live (TTL) is a parameter on messages. Reducing the number of hops between the root and the leaves, reduces the number of failure points along any given root to leaf path.

**Ant based multicast routing with multiple constraints:** Our new algorithm uses a heuristic to form degree-bounded spanning trees by ensuring that all hosts connect to the source by connecting through a host that is closer to the root. This helps to reduce delay introduced by deviations from the optimal path.

Our algorithm calculates the shortest path tree from the root using the ACO technique. The shortest path tree is then modified using the heuristic whereby it is stated that each node must be connected to a parent that is closer to the root than itself, that is if  $a$ 's parent is  $b$ , then  $dM(b, \text{root}) \leq dM(a, \text{root})$ . Visually, this is represented by concentric circles are drawn for each radius of a node from the root and all nodes are connected to a parent in a circle of equal to, or small radius than the ring they are a member of. Initially, all nodes in the shortest path tree will either be connected

directly to the root, or have a path to the root that connects only through nodes closer to the root.

**Algorithm for constraint based tree construction:**  
 Form the cluster  $V_i$  with cluster member  $v_i$  and maximum out-degree  $N$  by optimizing the shortest path tree and such that each node has a parent closer to the root than itself.

- L                      Tree level  
 $C_v$                     Current vertex  
 CL ( $c_1, c_2, \dots$ ) Set of closest  $N$  children of  $C_v$   
 RL ( $r_1, r_2, \dots$ ) Set of non-closest children of  $C_v$   
 $D(C_v)$                 Degree bound of  $C_v$   
 $R_v$                     Current vertex of the set RL.
1. Find the shortestPathTree ( $V_i, v_i$ )
  2. for each L, do
  3.     For each  $C_v$ , do
  4.         Find CL ( $c_1, c_2, \dots, c_N$ )
  5.         Add CL to set A1.
  6.         Find RL ( $r_1, r_2, \dots$ )
  7.         Add RL to set A2.
  8.         If  $d(C_v) < N$  then
  9.             Add  $C_v$  to A1
  10.     end for
  11. for each  $R_v$  do
  12.     If  $R_v$  is moveable, then
  13.         Connect  $R_v$  to the closest vertex in set A1
  14.         Closer to the root.
  15.     else
  16.         Swap  $R_v$  with a sibling that is farther from the root than  $R_v$

The relationship of each node always connecting to the root through nodes closer to the root is kept in effect by correcting the tree one level at a time. For example, starting at the root (level 1), the closest  $B$  children at each vertex are kept as children, while the remaining children up to  $n-B-1$  are made children of the closed  $B$  children. The process is then repeated until all nodes in the tree have  $\leq B$  children at all levels.

Consider a scenario in Fig. 1. Here node  $Y$  would normally be the third child of node  $X$  and node  $Z$  would be made a grandchild of  $X$ . However, this would violate the organizational rule stating that each vertex be connected to a vertex that is closer to the root than itself. This is solved by swapping the tree position of  $Y$  and  $Z$ . Node  $Z$  is made a child of  $X$  and  $Y$  is made a grandchild of  $X$  and a child of  $Z$ .

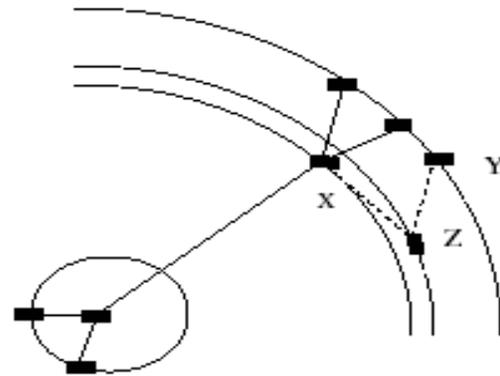


Fig. 1: Scenario when a node's  $B$  closest children are not the 3 closest nodes to the root (with a maximum out-degree of 3)

## RESULTS AND DISCUSSION

**Simulation model and parameters:** The NS2 is used to simulate the proposed algorithm. In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol is used. It has the functionality to notify the network layer about link breakage.

In the simulation, mobile nodes move in a  $600 \times 600$  m rectangular region for 50 sec simulation time. Initial locations and movements of the nodes are obtained using the Random Waypoint (RWP) model of NS2. I assume each node moves independently with the same average speed. All nodes have the same transmission range of 250 m.

In this mobility model, a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between the minimal speed and maximal speed. After it reaches its destination, the node stays there for a pause time and then moves again.

In the simulation, the maximal speed is  $10 \text{ m sec}^{-1}$  and pause time is 5 sec. The various no. of nodes are 25, 50, 75 and 100 is to investigate the performance influence of different topologies. The simulated traffic is Constant Bit Rate (CBR). For each scenario, ten runs with different random seeds were conducted and the results were averaged.

The AMR protocol is compared with MAODV. The evaluation is mainly based on performance according to the following metrics:

**Control overhead:** The control overhead is defined as the total number of routing control packets received.

Table 1: Simulation results

Nodes	Overhead		Load		Delay		PD fraction	
	MAODV	AMR	MAODV	AMR	MAODV	AMR	MAODV	AMR
25	1313	0	0.52078	0.0000	0.2238	0.1458	24.9578	38.5447
50	1863	850	1.42977	0.5501	0.5651	0.1674	23.9750	28.4740
75	5809	861	2.19373	0.7284	0.8084	0.2620	13.2185	21.7840
100	5865	1274	2.27524	0.8581	0.9247	0.2723	11.1377	19.2310

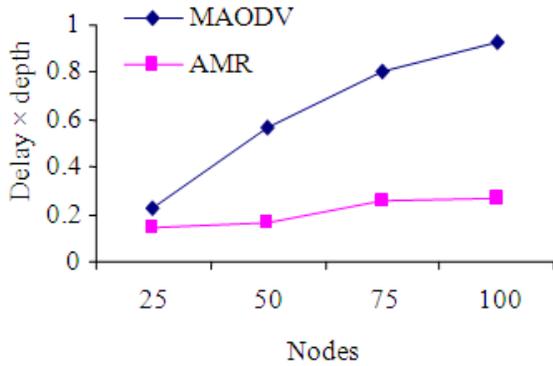


Fig. 2: Delay x depth

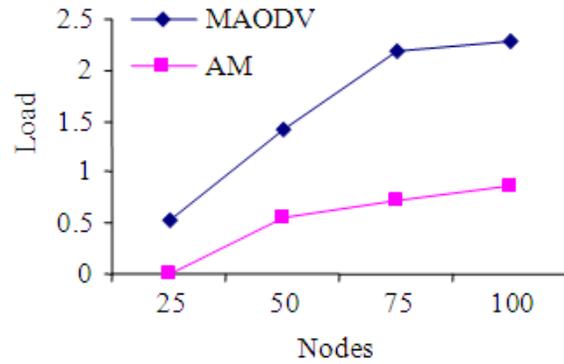


Fig. 4: Normalized routing load

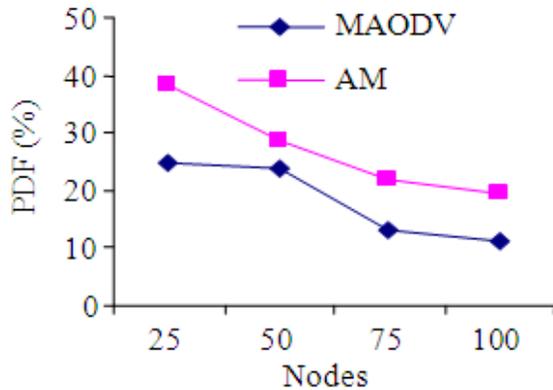


Fig. 3: Packet delivery fraction

**Routing load:** The normalized routing load is the ratio of no. of routing packets and the total no. of packets received.

**Packet delivery fraction:** It is the ratio of the fraction of packets received successfully and the total no. of packets sent.

**Delay X depth:** It is the normalized product of end-to-end-delay and average tree depth.

**Simulation results:** Table 1 shows the simulation results of AMR and MAODV<sup>[19]</sup> for the above metrics.

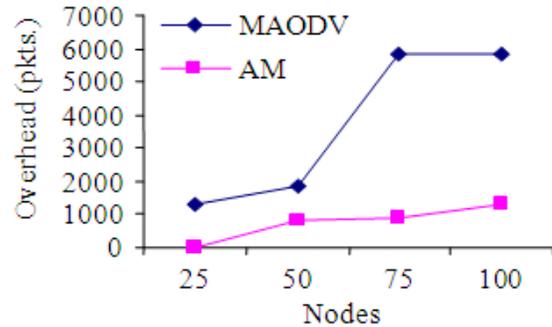


Fig. 5: Routing overhead

From the Table 1, it can be seen that AMR is better than MAODV in all the metrics.

Figure 2 shows that the normalized product of average delay and depth of the tree is less when compared with MAODV.

Figure 3 shows the average packet delivery fraction is more when compared to MAODV.

Figure 4 and 5 shows that, the routing load and overhead are significantly less compared to the MAODV routing protocol.

### CONCLUSION

The costs of the tree under multiple constraints are reduced by the several algorithms which are based on the Ant Colony Optimization (ACO) approach. The

Traffic-Engineering Multicast problem is treated as a single-purpose problem with several constraints with the help of these algorithms. The main disadvantage of this approach is the need of a predefined upper bound that can isolate good trees from the final solution. In order to solve the traffic engineering multicast problem which optimizes many objectives simultaneously this study offers a design on Ant Based Multicast Routing (AMR) algorithm for multicast routing in mobile ad hoc networks. The algorithm calculates one more additional constraint in the cost metric which is the product of average-delay and the maximum depth of the multicast tree. Moreover it also attempts to reduce the combined cost metric. By simulation results, it is clear that our proposed algorithm surpasses all the previous algorithms by developing multicast trees with different sizes.

### REFERENCES

1. Mesut, G. and O. Spaniol, 2003. Routing algorithms for mobile multi-hop ad-hoc networks. <http://portal.acm.org/citation.cfm?id=963972>
2. Dorigo, M. *et al.*, 1999. Ant algorithms for distributed discrete optimization. *Artif. Life*, 5: 137-172. DOI: 10.1162/106454699568728
3. Goss, S., S. Aron, J.L. Deneubourg and J.M. Pasteels, 1999. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76: 579-581. DOI: 10.1007/BF00462870, Pages579-581
4. Theraulaz, G. and E. Bonabeau, 1999. A brief history of stigmergy. *Artif. Life*, 5: 97-116. DOI: 10.1162/106454699568700
5. Pitakaso, R., C. Almeder, K.F. Doerner and R.F. Hartl, 2007. A max-min ant system for unconstrained multi-level lot sizing problems. *Comput. Operat. Res.*, 34: 2533-2552. <http://portal.acm.org/citation.cfm?id=1225132>
6. Baras, J.S. and H. Mehta, 2003. A probabilistic emergent routing algorithm for mobile ad hoc networks. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.7561>
7. Fujita, K. A. Saito, T. Matsui and H. Matsuo, 2002. An adaptive ant-based routing algorithm used routing history in dynamic networks. <http://ml.matlab.nitech.ac.jp/~matsuo/SEAL02-2.pdf>
8. Matsuo, H. and K. Mori, 2001. Accelerated ants routing in dynamic networks. Proceeding of the 2nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, (SEAINPDE'01), pp: 1-7. <http://www.matlab.nitech.ac.jp/~matsuo/SNPD01-2.pdf>
9. Marwaha, S. C. Khong and T.D. Srinivasan, 2002. Mobile agents based routing protocol for mobile ad hoc networks. Proceeding of the IEEE Globecom, Nov. 17-21, IEEE Xplore Press, USA., pp: 163-167. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1188062](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1188062)
10. Heissenb, M. and T. Braun, 2003. Ants-based routing in large scale mobile ad-hoc networks. Proceedings of the Research on Mobile Information and Communication Systems, (NCCR-MICS' 03), National Competence Center, pp: 1-9. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.2058&rep=rep1&type=pdf>
11. Sinha, P., R. Sivakumar and V. Bharghavan, 1999. MCDAR: Multicast core extraction distributed ad hoc routing. Proceedings of the IEEE Wireless Communications and Networking Conference, Sept. 21-24, IEEE Xplore Press, New Orleans, LA., USA., pp: 1313-1317. DOI: 10.1109/WCNC.1999.796950
12. Devarapalli, V. and D. Sidhu, 2001. MZR: A multicast protocol for mobile ad hoc networks Proceeding of the IEEE International Conference on Communications, June 11-14, IEEE Xplore Press, Helsinki, Finland, pp: 886-891. DOI: 10.1109/ICC.2001.937365
13. Vaishampayan, R. and J.J. Garcia-Luna-Aceves, 2004. Efficient and robust multicast routing in mobile ad hoc networks. Proceeding of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems, Oct. 25-27, IEEE Xplore Press, USA., pp: 304-313. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1392169](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1392169)
14. Jorjeta G.J. and D.B. Johnson, 2001. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. Proceeding of the 2nd ACM International Symposium on Mobile and Ad-hoc Networking and Computing, Oct. 4-5, ACM Press, Long Beach, CA., USA., pp: 33-44. <http://portal.acm.org/citation.cfm?id=501423>
15. Castro, M. P. Druschel, A.M. Kermarrec and A.I.T. Rowstron, 2002. Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE J. Select. Areas Commun.*, 20: 1489-1499. DOI: 10.1109/JSAC.2002.803069
16. Zhang, R. and Y.C. Hu, 2003. Borg: A hybrid protocol for scalable application-level multicast in peer-to-peer networks. Proceedings of the 13th international workshop on Network and Operating Systems Support for Digital Audio and Video, June 1-3, ACM Press, Monterey, CA., USA., pp: 172-179. <http://portal.acm.org/citation.cfm?id=776349&dl=GUIDE&coll=GUIDE&CFID=64403391&CFTOKEN=61380964>

17. Nasipuri, A. and S.R. Das, 1999. On-demand multipath routing for mobile ad hoc networks. Proceeding of the 8th IEEE International Conference Computer Communication and Networks, Oct. 11-13, IEEE Xplore Press, Boston, MA., USA., pp: 64-70. DOI: 10.1109/ICCCN.1999.805497
18. Vittorio Maniezzo, Luca Maria Gambardella and Fabio de Luigi, 2004. Ant Colony Optimization. Springer-Verlag, pp: 101-117. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.7560>
19. Royer, E.M. and C.E. Perkins, 1999. Multicast operation of the ad-hoc on-demand distance vector routing protocol. Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking, Aug. 15-19, ACM Press, Seattle, Washington, United States, pp: 207-218. <http://portal.acm.org/citation.cfm?id=313538>