ARA – The Ant-Colony Based Routing Algorithm for MANETs [*]

Mesut Güneş, Udo Sorges, Imed Bouazizi

{mesut, udo, imed }@i4.informatik.rwth-aachen.de
Department of Computer Science, Informatik 4
Aachen University of Technology
Aachen, Germany

Abstract

A mobile ad-hoc network (MANET) is a collection of mobile nodes which communicate over radio. These kind of networks are very flexible, thus they do not require any existing infrastructure or central administration. Therefore, mobile ad-hoc networks are suitable for temporary communication links. The biggest challenge in this kind of networks is to find a path between the communication end points, what is aggravated through the node mobility.

In this paper we present a new on-demand routing algorithm for mobile, multi-hop ad-hoc networks. The protocol is based on swarm intelligence and especially on the ant colony based meta heuristic. These approaches try to map the solution capability of swarms to mathematical and engineering problems. The introduced routing protocol is highly adaptive, efficient and scalable. The main goal in the design of the protocol was to reduce the overhead for routing. We refer to the protocol as the ***A****nt-Colony-Based* ***R****outing* ***A****lgorithm (ARA).*

Keywords: *Ad-hoc network, MANET, Routing*

1. Introduction

A mobile ad-hoc network (MANET) is a set of mobile nodes which communicate over radio and do not need any infrastructure. 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. Due to the limited transmission range of wireless interfaces, the communication traffic has to be relayed over several intermediate nodes to enable the communication between two nodes. Therefore, this kind of networks are also called mobile multi-hop ad-hoc networks. Nodes not only have to fulfill the functionality of hosts, but also each node has also to be a router, forwarding packets for other nodes.

One interesting application for mobile ad-hoc networks beside the classical ones, disaster and military applications, is the deployment of mobile ad-hoc networks for multimedia applications. However, the performance of such networks have to be improved before this can be realized. With upcoming radio technologies, e.g., IEEE 802.11a and Bluetooth, the realization of multimedia applications over mobile ad-hoc networks comes closer.

The main problem in mobile ad-hoc networks is still, the finding of a route between the communication end-points, which is aggravated through the node mobility. In the literature one can find many different approaches which try to handle this problem [4, 10], but there is no routing algorithm which fits in all cases.

In this paper we present a new approach for an on-demand ad-hoc routing algorithm, which is based on *swarm intelligence*. Ant colony algorithms are a subset of swarm intelligence and consider the ability of simple ants to solve complex problems by cooperation. The interesting point is, that the ants do not need any direct communication for the solution process, instead they communicate by *stigmergy*. The notion of stigmergy means the indirect communication of individuals through modifying their environment. Several algorithms which are based on ant colony problems were introduced in recent years to solve different problems, e.g. optimization problems. To show that the approach has the potential to become an appropriate algorithm for mobile multi-hop ad-hoc networks we present some results, which are based on simulations made with the current implementation in ns-2 [5].

The remainder of this paper is organized as follows. In section 2 we present the basics and the background of ant colony optimization meta heuristic. In section 3 we present the routing algorithm ARA in detail and discuss its advantages and problems. Subsequently, in section 4 we present

[*] International Workshop on Ad Hoc Networking (IWAHN 2002), Vancouver, British Columbia, Canada, August 18-21, 2002

some simulation results to show the ability of the approach. Finally, a conclusion is given in section 5.

2. Basics and Background

The ant colony optimization meta-heuristic is a particular class of ant algorithms. Ant algorithms are multi-agent systems, which consist of agents with the behavior of individual ants, see [3, 1] for more information.

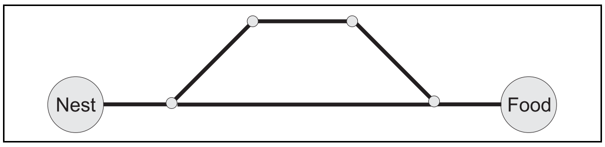

Figure 1. All ants take the shortest path after an initial searching time.

2.1. Basic ant algorithm

The basic idea of the ant colony optimization meta heuristic is taken from the food searching behavior of real ants. When ants are on they way to search for food, they start from their nest and walk toward the food. When an ant reaches an intersection, it has to decide which branch to take next. While walking, ants deposit *pheromone*[1], which marks the route taken. The concentration of pheromone on a certain path is an indication of its usage. With time the concentration of pheromone decreases due to diffusion effects. This property is important because it is integrating dynamic into the path searching process.

Figure 1 shows a scenario with two routes from the nest to the food place. At the intersection, the first ants randomly select the next branch. Since the below route is shorter than the upper one, the ants which take this path will reach the food place first. On their way back to the nest, the ants again have to select a path. After a short time the pheromone concentration on the shorter path will be higher than on the longer path, because the ants using the shorter path will increase the pheromone concentration faster. The shortest path will thus be identified and eventually all ants will only use this one.

This behavior of the ants can be used to find the shortest path in networks. Especially, the dynamic component of this method allows a high adaptation to changes in mobile ad-hoc network topology, since in these networks the existence of links are not guaranteed and link changes occur very often.

2.2. Simple ant colony optimization meta-heuristic algorithm

Let $G = (V, E)$ be a connected graph with $n = |V|$ nodes. The simple ant colony optimization meta-heuristic can be used to find the shortest path between a source node v_s and a destination node v_d on the graph G. The path length is given by the number of nodes on the path. Each edge $e(i, j) \in E$ of the graph connecting the nodes v_i and v_j has a variable $\varphi_{i,j}$ (artificial pheromone), which is modified by the ants when they visit the node. The pheromone concentration, $\varphi_{i,j}$ is an indication of the usage of this edge.

An ant located in node v_i uses pheromone $\varphi_{i,j}$ of node $v_j \in N_i$ to compute the probability of node v_j as next hop. N_i is the set of one-step neighbors of node v_i.

$$p_{i,j} = \begin{cases} \frac{\varphi_{i,j}}{\sum_{j \in N_i} \varphi_{i,j}} & \text{if } j \in N_i \\ 0 & \text{if } j \notin N_i \end{cases}$$

The transition probabilities $p_{i,j}$ of a node v_i fulfill the constraint:

$$\sum_{j \in N_i} p_{i,j} = 1, \qquad i \in [1, N]$$

During the route finding process, ants deposit pheromone on the edges. In the simple ant colony optimization meta-heuristic algorithm, the ants deposit a constant amount $\Delta\varphi$ of pheromone. An ant changes the amount of pheromone of the edge $e(v_i, v_j)$ when moving from node v_i to node v_j as follows:

$$\varphi_{i,j} := \varphi_{i,j} + \Delta\varphi \tag{1}$$

Like real pheromone the artificial pheromone concentration decreases with time to inhibit a fast convergence of pheromone on the edges. In the simple ant colony optimization meta-heuristic, this happens exponentially:

$$\varphi_{i,j} := (1 - q) \cdot \varphi_{i,j}, \qquad q \in (0, 1] \tag{2}$$

2.3. Why ant colony optimization meta-heuristic suits to ad-hoc networks

The simple ant colony optimization meta-heuristic shown in the previous section illustrates different reasons why this kind of algorithms could perform well in mobile multi-hop ad-hoc networks. We will discuss various reasons by considering important properties of mobile ad-hoc networks.

- **Dynamic topology:** This property is responsible for the bad performance of several routing algorithms in mobile multi-hop ad-hoc networks. The ant colony optimization meta-heuristic is based on agent systems and works with individual ants. This allows a high adaptation to the current topology of the network.

[1] Pheromone is a kind of cumarin, which ants are able to smell.

- **Local work:** In contrast to other routing approaches, the ant colony optimization meta-heuristic is based only on local information, i.e., no routing tables or other information blocks have to be transmitted to neighbors or to all nodes of the network.
- **Link quality:** It is possible to integrate the connection/link quality into the computation of the pheromone concentration, especially into the evaporation process. This will improve the decision process with respect to the link quality. It is here important to notice, that the approach has to be modified so that nodes can also manipulate the pheromone concentration independent of the ants, i.e. data packets, for this a node has to monitor the link quality.
- **Support for multi-path:** Each node has a routing table with entries for all its neighbors, which contains also the pheromone concentration. The decision rule, to select the next node, is based on the pheromone concentration on the current node, which is provided for each possible link. Thus, the approach supports multi-path routing.

3. The Routing Algorithm

In this section we discuss the adaptation of the ant colony optimization meta-heuristic for mobile ad-hoc networks and describe the **A**nt colony based **R**outing **A**lgorithm (ARA). The routing algorithm is very similar constructed as many other routing approaches and consists of three phases.

3.1. Route Discovery Phase

In the route discovery phase new routes are created. The creation of new routes requires the use of a *forward ant* (FANT) and a *backward ant* (BANT). A FANT is an agent which establishes the pheromone track to the source node. In contrast, a BANT establishes the pheromone track to the destination node. The FANT is a small packet with a unique sequence number. Nodes are able to distinguish duplicate packets on the basis of the sequence number and the source address of the FANT.

A forward ant is broadcasted by the sender and will be relayed by the neighbors of the sender (see figure 2). A node receiving a FANT for the first time, creates a record in its routing table. A record in the routing table is a triple and consists of `(destination address, next hop, pheromone value)`. The node interprets the source address of the FANT as destination address, the address of the previous node as the next hop, and computes the pheromone value depending on the number of hops the FANT needed to reach the node. Then the node relays the FANT to its neighbors. Duplicate FANTs are identified through the unique sequence number and destroyed by the nodes. When the FANT reaches the destination node, it is processed in a special way. The destination node extracts the information of the FANT and destroys it. Subsequently, it creates a BANT and sends it to the source node (see figure 3). The BANT has the same task as the FANT, i.e. establishing a track to this node. When the sender receives the BANT from the destination node, the path is established and data packets can be sent.

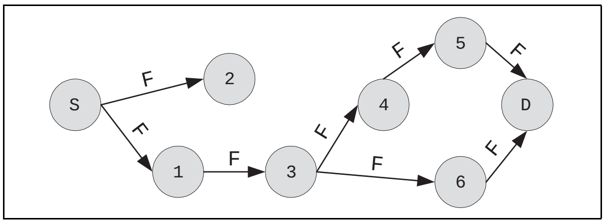

Figure 2. Route discovery phase. A forward ant (F) is send from the sender (S) toward the destination node (D). The forward ant is relayed by other nodes, which initialize their routing table and the pheromone values.

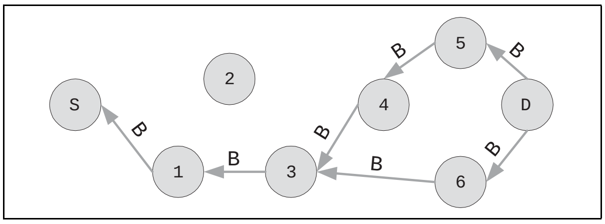

Figure 3. Route discovery phase. The backward ant (B) has the same task as the forward ant. It is send by the destination node toward the source node.

Figures 2 and 3 schematically depict the route discovery phase. In the depicted case, node 3 has two ways for the path, via node 4 and over node 6. In our case, the forward ant creates only one pheromone track toward the source node, but the backward ant creates two pheromone tracks toward the destination node. So multi-path routing is also supported by ARA.

3.2. Route Maintenance

The second phase of the routing algorithm is called route maintenance, which is responsible for the improvement of the routes during the communication. ARA does not need any special packets for route maintenance. Once the FANT and BANT have established the pheromone tracks for the source and destination nodes, subsequent data packets are used to maintain the path. Similar to the nature, established paths do not keep their initial pheromone values forever. When a node v_i relays a data packet toward the destination v_D to a neighbor node v_j, it increases the pheromone value of the entry (v_D, v_j, φ) by $\Delta\varphi$, i.e., the path to the destination is strengthened by the data packets. In contrast, the next hop v_j increases the pheromone value of the entry (v_S, v_i, φ) by $\Delta\varphi$, i.e. the path to the source node is also strengthened. The evaporation process of the real pheromone is simulated by regular decreasing of the pheromone values, which is performed according to equation 2.

The above method for route maintenance could lead to undesired loops. ARA prevents loops by a very simple method, which is also used during the route discovery phase. Nodes can recognize duplicate receptions of data packets, based on the source address and the sequence number. If a node receives a duplicate packet, it sets the `DUPLICATE_ERROR` flag and sends the packet back to the previous node. The previous node deactivates the link to this node, so that data packets cannot be send to this direction any more.

3.3. Route Failure Handling

The third and last phase of ARA handles routing failures, which are caused especially through node mobility and thus very common in mobile ad-hoc networks. ARA recognizes a route failure through a missing acknowledgement. If a node gets a `ROUTE_ERROR` message for a certain link, it first deactivates this link by setting the pheromone value to 0.

Then the node searches for an alternative link in its routing table. If there exists a second link it sends the packet via this path. Otherwise the node informs its neighbors, hoping that they can relay the packet. Either the packet can be transported to the destination node or the backtracking continues to the source node. If the packet does not reach the destination, the source has to initiate a new route discovery phase.

3.4. Properties of ARA

According to [7] a routing algorithm for mobile ad-hoc networks should fulfill the following requirements:

- **Distributed operation:** In ARA, each node owns a set of pheromone counter[2] $\varphi_{i,j}$ in its routing table for a link between node v_i and v_j. Each node controls the pheromone counter independently, when ants visit the node on route searches.
- **Loop-free:** The nodes register the unique sequence number of route finding packets, FANT and BANT, so they do not generate loops.
- **Demand-based operation:** Routes are established by manipulating the pheromone counter $\varphi_{i,j}$ in the nodes. Over time, the amount of pheromone decreases to zero when ants do not visit this node. A route finding process is only run, when a sender demands.
- **Sleep period operation:** Nodes are able to sleep when their amount of pheromone reaches a threshold. Other nodes will then not consider this node.

Additionally, ARA has the following properties:

- **Locality:** The routing table and the statistic information block of a node are local and they are not transmitted to any other node.
- **Multi-path:** Each node maintains several paths to a certain destination. The choice of a certain route depends on the environment, e.g., link quality to the relay node.
- **Sleep mode:** In the sleep mode a node snoops, only packets which are destined to it are processed, thus saving power.

3.5. Overhead of ARA

The expected overhead of ARA is very small, because there are no routing tables which are interchanged between the nodes. Unlike other routing algorithms, the FANT and BANT packets do not transmit much routing information. Only a unique sequence number is transmitted in the routing packets. Most route maintenance is performed through data packets, thus they do not have to transmit additional routing information. ARA only needs the information in the IP header of the data packets.

4. Simulation Results

4.1. Simulation Environment

We have implemented ARA in ns-2 [5]. The results shown in this section were obtained using the current version of the implementation. For our results we employed 50 mobile nodes according to IEEE 802.11, which moved on a simulation area of 1500m×300m. The nodes moved with a maximal velocity of 10 m/s and according to the random waypoint mobility model [2]. The simulation time is 900 seconds. The node mobility is expressed by the pause time. We performed simulations with 7 different pause times 0, 30, 60, 120, 300, 600 and 900 seconds. In the cases of

[2] We call the pheromone concentration here as a counter, because of its regularly decreasing by the node.

0 seconds pause time, the nodes move constantly. In contrast, in the cases of 900 seconds pause time the nodes do not move.

As mentioned in the introduction our main goal was to reduce the routing overhead. So we will mainly discuss this aspect of ARA.

4.2. Comparison with existing routing algorithms

A new routing algorithm should show its performance in comparison with existing and known algorithms. Therefore, to get a feeling for the performance of ARA we will discuss the results in comparison with AODV [9, 4], DSDV [8, 4], and DSR [6, 4].

We present here results made with simulations with 10 parallel connections with constant bit rate (CBR) traffic. The parameter are similar to those in [2].

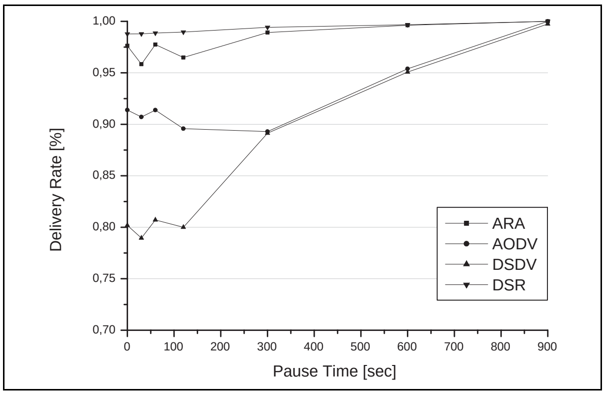

Figure 4. Comparison of four protocols by the fraction of successful delivered packets as a function of pause time. Simulations with 10 CBR connections.

We will first discuss the robustness of the routing protocols. Figure 4 shows the delivery rate, i.e., the part of packets a certain routing protocol was able to deliver properly. This value is important, since it describes the performance which transport protocols will see, i.e., the throughput is restricted by this value.

In the case with low pause time, i.e., high topology changes, only DSR and ARA are able to deliver more than 95% of the sent packets. In situations with very high dynamics DSR shows the best performance followed by ARA. With less dynamic, up to 300 seconds of pause time, ARA is very close to DSR. AODV and DSDV perform poorly in situations with high mobility. They are only able to deliver 95% of the packets in situations with more than 600 seconds pause time.

Figure 5 shows the delivery rate of ARA within the confidence interval of 95%. The presented confidence interval is computed out of 10 simulation runs for each pause time. For the most simulation scenarios the results are very close. Only in simulations with 600 seconds pause time the variation of the results is higher, this is maybe due to mobility impacts. All results are above 85% and most results are within the range 90% and 100%.

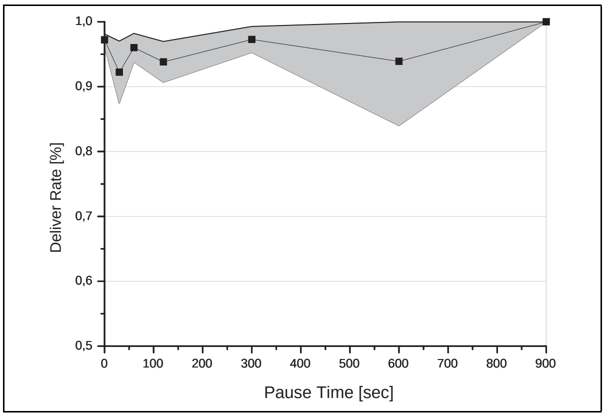

Figure 5. Delivery rate of ARA. Confidence interval with $\alpha = 0.05$

Figure 6 depicts the overhead of all routing protocols for the scenario discussed in figure 4. The graph shows the fraction of routing packets needed to deliver a data packet. We counted bits used for routing, because the different protocols generate the overhead in very different ways. Here ARA shows its advantage. In the case of very high dynamics 0 - 100 seconds of pause time, it generates the least overhead. With less dynamic the generated overhead stabilizes and is very low for the whole simulation time. In cases with pause time of 300 - 600 seconds AODV and ARA are very close in generating overhead. With less dynamics, AODV generates less overhead than ARA, but the difference is very small. ARA and AODV are followed by DSR through all simulation scenarios. Only DSDV shows here a very high difference in comparison to the other three algorithms.

Figure 7 shows the overhead of ARA within the confidence interval of 95%. All results are very close through all simulation scenarios. This shows that ARA creates much less routing overhead for all considered mobility scenarios.

We will now consider the overhead of the routing protocols by the number of packets which create the overhead. Figure 8 depicts the needed number of packets to perform the routing job for all four routing algorithms. In the cases with high mobility it is obvious that DSR and ARA create the least overhead. Especially DSR shows here a better performance than ARA. This is due to the needed flooding of the approach in the route finding phase. With high node mobility route failure occur more often, thus requires the performing of the route failure handling part of the algo-

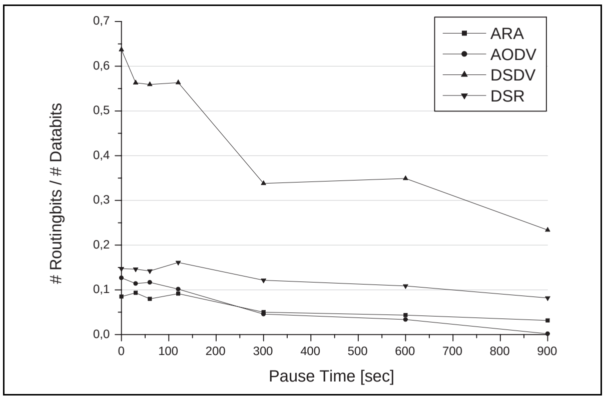

Figure 6. Comparison of four protocols by the fraction of successfully send bits and the needed bits.

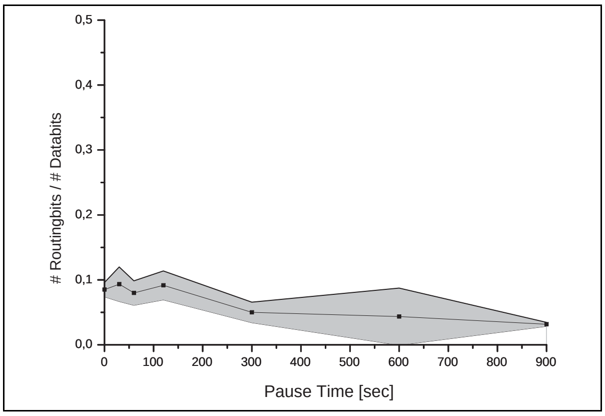

Figure 7. Overhead of ARA. Confidence interval with $\alpha = 0.05$. Simulations with 10 CBR connections and confidence interval of 10 simulation runs.

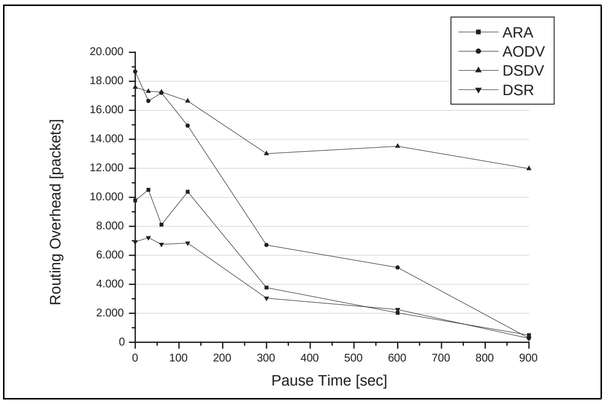

Figure 8. Comparison of four protocols by the number of needed routing packets. Simulations with 10 CBR connections.

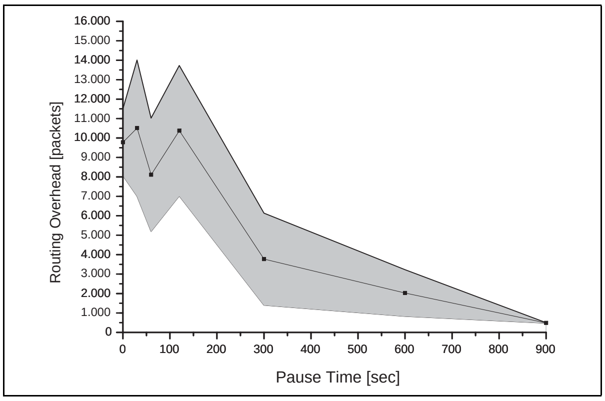

Figure 9. Overhead of ARA in packets. Confidence interval with $\alpha = 0.05$. Simulations with 10 CBR connections and confidence interval of 10 simulation runs.

rithm, which in worst case has to backtrack the path until the sender. With less mobility up to 300 seconds of pause time, the overhead of ARA and DSR are very close. AODV and DSDV show here again their poor performance by creating large numbers of routing packets.

Figure 9 shows the generated overhead of ARA for the considered scenario of figure 8 within the confidence interval of 95%. It is clear that in the cases with high mobility the results show a higher variation, which decreases with less mobility.

5. Conclusions and Future Work

Mobile multi-hop ad-hoc networks are flexible networks, which do not require pre-installed infrastructure. With upcoming wireless transmission technologies and highly sophisticated devices their application will increase. However the main challenge in mobile multi-hop ad-hoc networks is still the routing problem, which is aggravated by the node mobility. Various approaches were introduced in the recent years which try to handle the problems in this kind of networks, but no one fits best for all applications.

In this paper we presented a new on-demand routing approach for mobile multi-hop ad-hoc networks. The approach is based on swarm intelligence and especially on the ant colony optimization meta-heuristic. These fascinating family of algorithms try to apply the ability of swarms to mathematical problems and were applied successfully to several optimization problems.

We have discussed the adaptation of the method to mobile multi-hop ad-hoc networks and showed through simulations its ability to perform well in such kind of networks.

The performance for the considered simulation scenarios is very close to the performance of DSR, but generates less overhead.

We are working further on the implementation to improve the algorithm. Our further investigations include experiments with high network load and multimedia data. Additionally, analysis of the maintenance of the pheromone concentration is needed. There are different ways to manipulate the pheromone concentration on the edges, which influence the performance of the routing algorithm.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence: from natural to artificial intelligence*. Oxford University Press, 1999. ISBN 0-19-513158-4.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multihop wireless ad hoc network routing protocols. *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 85–97, 1998.

[3] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill, London, 1999.

[4] C. E. P. (Editor). *Ad Hoc Networking*. Addison-Wesley, 2001. ISBN 0-201-30976-9.

[5] K. Fall and K. Varadhan. *The ns Manual*, Nov 2000.

[6] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. IETF Internet draft, draft-ietf-manet-dsr-04.txt, November 2000.

[7] J. P. Macker and M. S. Corson. Mobile ad hoc networking and the IETF. *Mobile Computing and Communications Review*, 2(1):9–14, 1998.

[8] C. E. Perkins and P. Bhagvat. Hihgly dynamic destination-sequenced distnace-vector routing (dsdv) for mobile computers. *Computer Communications Rev.*, pages 234–244, October 1994.

[9] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on-demand distance vector (aodv) routing. IETF Internet draft, draft-ietf-manet-aodv-07.txt, November 2000.

[10] C.-K. Toh. *Ad hoc mobile wireless networks: protocols and systems*. Prentice Hall, 2002. ISBN: 0-13-007817-4.