A Fault Tolerant Distributed Routing Algorithm Based on Combinatorial Ant Systems

Jose Aguilar[1] and Miguel Labrador[2]

[1] CEMISID. Dpto. de Computación, Facultad de Ingeniería. Universidad de los Andes
Av. Tulio Febres. Mérida, 5010, Venezuela
aguilar@ula.edu

[2] Department of Computer Science and Engineering University of South Florida. Tampa, FL 33620 USA
labrador@cse.usf.edu

Abstract. In this paper, a general Combinatorial Ant System-based fault tolerant distributed routing algorithm modeled like a dynamic combinatorial optimization problem is presented. In the proposed algorithm, the solution space of the dynamic combinatorial optimization problem is mapped into the space where the ants will walk, and the transition probability and the pheromone update formula of the Ant System is defined according to the objective function of the communication problem.

1 Introduction

The problem to be solved by any routing system is to direct traffic from sources to destinations while maximizing some network performance metric of interest. Depending on the type of network, common performance metrics are call rejection rate, throughput, delay, distance, and energy, among the most important ones. Routing in communication networks is necessary because in real systems not all nodes are directly connected. Currently, routing algorithms face important challenges due to the increased complexity found in modern networks. The routing function is particularly challenging in modern networks because traffic conditions, the structure of the network, and the network resources are limited and constantly changing. The lack of adaptability of routing algorithms to frequent topological changes, node capacities, traffic patterns, load changes, energy availability, and others, reduces the throughput of the network. This problem can be defined as a distributed time-variant dynamic combinatorial optimization problem [2, 11].

Artificial Ant Systems provide a promising alternative to develop routing algorithms for modern communication networks. Inherent properties of ant systems include massive system scalability, emergent behavior and intelligence from low complexity local interactions, autonomy, and stigmergy or communication through the environment, which are very desirable features for many types of networks. In general, real ants are capable of finding the shortest path from a food source to their nest by exploiting pheromone information [1, 3, 4, 5, 6]. While walking, ants deposit pheromone trails on the ground and follow pheromone previously deposited by other

D.S. Huang, X.-P. Zhang, G.-B. Huang (Eds.): ICIC 2005, Part I, LNCS 3644, pp. 514 – 523, 2005.

ants. The above behavior of real ants has inspired the Ants System (AS), an algorithm in which a set of artificial ants cooperate to the solution of a problem by exchanging information via pheromone deposited on a graph.

Ants systems have been used in the past to solve other combinatorial optimization problems such as the traveling salesman problem and the quadratic assignment problem, among others [3, 4, 5, 6, 7]. We have proposed a distributed algorithm based on AS concepts, called the Combinatorial Ant System (CAS), to solve static discrete-state and dynamic combinatorial optimization problems [1,2]. The main novel idea introduced by our model is the definition of a general procedure to solve combinatorial optimization problems using AS. In our approach, the graph that describes the solution space of the combinatorial optimization problem is mapped on the AS graph, and the transition function and the pheromone update formula of the AS are built according to the objective function of the combinatorial optimization problem. In this paper, we present a routing algorithm based on CAS. Our scheme provides a model for distributed network data flow organization, which can be used to solve difficult problems in today's communication networks. The remaining of the paper is organized as follows. Section 2 presents the Combinatorial Ant System and the Routing Problem. Section 3 presents the general distributed routing algorithm based on the CAS. Then, Section 4 presents and evaluates the utilization of this algorithm on communication networks. Finally, conclusions are presented.

2 Theoretical Aspects

2.1 The Combinatorial Ant System (CAS)

Swarm intelligence appears in biological swarms of certain insect species. It gives rise to complex and often intelligent behavior through complex interaction of thousands of autonomous swarm members. Interaction is based on primitive instincts with no supervision. The end result is the accomplishment of very complex forms of social behavior or optimization tasks [1,3,4,5,7]. The main principle behind these interactions is the autocatalytic reaction like in the case of Ant Systems where the ants attracted by the pheromone will lay more of the same on the same trail, causing even more ants to be attracted.

The Ant System (AS) is the progenitor of all research efforts with ant algorithms, and it was first applied to the Traveling Salesman Problem (TSP) [5, 6]. Algorithms inspired by AS have manifested as heuristic methods to solve combinatorial optimization problems. These algorithms mainly rely on their versatility, robustness and operations based on populations. The procedure is based on the search of agents called "ants", i.e. agents with very simple capabilities that try to simulate the behavior of the ants.

AS utilizes a graph representation (*AS graph*) where each edge *(r,s)* has a desirability measure γ_{rs}, called *pheromone*, which is updated at run time by artificial ants. Informally, the following procedure illustrates how the AS works. Each ant generates a complete tour by choosing the nodes according to a probabilistic state transition rule; ants prefer to move to nodes that are connected by short edges, which have a high pheromone presence. Once all ants have completed their tours, a global pheromone updating rule is applied. First, a fraction of the pheromone evaporates on all edges, and then each ant deposits an amount of pheromone on the edges that belong to its tour in proportion to how short this tour was. At his point, we continue with a new iteration of the process.

There are two reasons for using AS on the TSP. First, the TSP graph can be directly mapped on the AS graph. Second, the transition function has similar goals to the TSP. This is not the case for other combinatorial optimization problems. In [1, 2], we proposed a distributed algorithm based on AS concepts, called the CAS, to solve any type of combinatorial optimization problems. In this approach, each ant builds a solution walking through the AS graph using a transition rule and a pheromone update formula defined according to the objective function of the combinatorial optimization problem. This approach involves the following steps:

1. *Definition of the graph that describes the solution space of the combinatorial optimization problem (COP graph).* The solution space is defined by a graph where the nodes represent partial possible solutions to the problem, and the edges the relationship between the partial solutions.
2. *Building the AS graph.* The COP graph is used to define the *AS graph,* the graph where the ants will finally walk through.
3. *Definition of the transition function and the pheromone update formula of the CAS.* These are built according to the objective function of the combinatorial optimization problem.
4. Executing the AS procedure described before.

2.1.1 Building the AS Graph

The first step is to build the COP graph, then we define the AS graph with the same structure of the COP graph. The AS graph has two weight matrices. The first matrix is defined according to the COP graph and registers the relationship between the elements of the solution space (COP matrix). The second one registers the pheromone trail accumulated on each edge (pheromone matrix). This weight matrix is calculated/updated according to the pheromone update formula. When the incoming edge weights of the pheromone matrix for a given node become high, this node has a high probability to be visited. On the other hand, if an edge between two nodes of the COP matrix is low, then it means that, ideally, if one of these nodes belongs to the final solution then the other one must belong too. If the edge is equal to infinite then it means that the nodes are incompatible, and therefore, they don't belong to at the same final solution.

We define a data structure to store the solution that every ant k is building. This data structure is a vector (A^k) with a length equal to the length of the solution, which is given by n, the number of nodes that an ant must visit. For a given ant, the vector keeps each node of the AS graph that it visits.

2.1.2 Defining the Transition Function and the Pheromone Update Formula

The state transition rule and the pheromone update formula are built using the objective function of the combinatorial optimization problem. The transition function between nodes is given by:

$$Tf\left(\gamma_{rs}(t), Cf^{k}_{r->s}(z)\right) = \frac{\gamma_{rs}(t)^{\alpha}}{Cf^{k}_{r->s}(z)^{\beta}}$$

where $\gamma_{rs}(t)$ is the pheromone at iteration t, $Cf^{k}_{r->s}(z)$ is the cost of the partial solution that is being built by ant k when it crosses the edge (r, s) if it is in the position r, $z-1$ is the current length of the partial solution (current length of A^k), and, α and β are two adjustable parameters that control the relative weight of trail intensity $(\gamma_{rs}(t))$ and the cost function.

In the CAS, the transition probability is as follows: an ant positioned at node r chooses node s to move to according to a probability $P^{k}_{rs}(t)$, which is calculated according to Equation 1:

$$P^{k}_{rs}(t) = \begin{cases} \dfrac{Tf(\gamma_{rs}(t), Cf^{k}_{r->s}(z))}{\sum_{u \in J^{k}_{r}} Tf\left(\gamma_{ru}(t), Cf^{k}_{r->u}(z)\right)} & \textit{If } s \in J^{k}_{r} \\ 0 & \textit{Otherwise} \end{cases} \tag{1}$$

where J^{k}_{r} is the set of nodes connected to r that remain to be visited by ant k positioned at node r. When $\beta=0$ we exploit previous solutions (only trail intensity is used), and when $\alpha=0$ we explore the solution space (a stochastic greedy algorithm is obtained). A tradeoff between quality of partial solutions and trail intensity is necessary. Once all ants have built their tours, pheromone, i.e. the trail intensity in the pheromone matrix, is updated on all edges according to Equation 2 [1, 2, 3, 4, 5, 6]:

$$\gamma_{rs}(t) = (1-\rho)\gamma_{rs}(t-1) + \sum_{k=1}^{m} \Delta\gamma^{k}_{rs}(t) \tag{2}$$

where ρ is a coefficient such that $(1 - \rho)$ represents the trail evaporation in one iteration (tour), m is the number of ants, and $\Delta\gamma_{rs}^{k}(t)$ is the quantity per unit of length of trail substance laid on edge (r, s) by the k^{th} ant in that iteration

$$\Delta\gamma^{k}_{rs}(t) = \begin{cases} \dfrac{1}{C^{k}_{f}(t)} & \text{If } \textit{edge}(r,s) \textit{ has been crossed by ant } k \\ 0 & \textit{Otherwise} \end{cases} \tag{3}$$

where $C^{k}_{f}(t)$ is the value of the cost function (objective function) of the solution proposed by ant k at iteration t. The general procedure of our approach is summarized as follows:

1. Generation of the AS graph.
2. Definition of the state transition rule and the pheromone update formula, according to the combinatorial optimization problem.
3. Repeat until system reaches a stable solution
   3.1. Place m ants on different nodes of the AS graph.
   3.2. For i=1, n

3.2.1. For j=1, m
    3.2.1.1. Choose node *s* to move to, according to the transition probability (Equation 1).
    3.2.1.2. Move ant *m* to the node *s*.
3.3. Update the pheromone using the pheromone update formula (Equations 2 and 3).

2.2 The Routing Problem

Routing is the function that allows information to be transmitted over a network from a source to a destination through a sequence of intermediate switching/buffering stations or nodes. Routing is necessary because in real systems not all nodes are directly connected. Routing algorithms can be classified as static or dynamic, and centralized or distributed [10]. Centralized algorithms usually have scalability problems, and single point of failure problems, or the inability of the network to recover in case of a failure in the central controlling station. Static routing assumes that network conditions are time-invariant, which is an unrealistic assumption in most of the cases. Adaptive routing schemes also have problems, including inconsistencies arising from node failures and potential oscillations that lead to circular paths and instability. Routing algorithms can also be classified as minimal or non-minimal [10]. Minimal routing allows packets to follow only minimal cost paths, while non-minimal routing allows more flexibility in choosing the path by utilizing other heuristics. Another class of routing algorithms is one where the routing scheme guarantees specified QoS requirements pertaining to delay and bandwidth [10].

Commonly, modern networks utilize dynamic routing schemes in order to cope with constant changes in the traffic conditions and the structure or topology of the network. This is particularly the case of wireless ad hoc networks where node mobility and failures produce frequent unpredictable node/link failures that result in topology changes. A vast literature of special routing algorithms for these types of networks exist [13, 14], all of them with the main goal of making the network layer more reliable and the network fault tolerant and efficient. However, maximizing throughput for time-variant load conditions and network topology is a NP-complete problem. A routing algorithm for communication networks with these characteristics can be defined as a dynamic combinatorial optimization problem, that is, like a distributed time-variant problem. In this paper, we are going to use our model to propose a routing algorithm for these networks, which support multiple node and link failures.

3 The General CAS-Based Distributed Routing Algorithm

There are a number of proposed ant-based routing algorithms [3, 9, 12, 13, 14]. The most celebrated one is AntNet, an adaptive agent-based routing algorithm that has outperformed the best-known routing algorithms on several packet-switched communication networks. Ant systems have also been applied to telephone networks. The Ant-Based Control (ABC) scheme is an example of a successful application. We are going to propose a new routing algorithm based on our CAS that can be used in different networking scenarios, such as networks with static topologies, networks with constant topology changes, and network with energy constraints.

We can use our approach for point to point or point to multipoint requests. In the case of point to point, one ant is launched to look for the best path to the destination. For a multipoint request with *m* destinations, *m* ants are launched. The route where intermediate nodes have large pheromone quantities is selected. For this, we use the local routing tables of each node like a transition function to its neighbours. Thus, according to the destination of the message, the node with highest probability to be visited corresponds to the entry in the table with the larger amount of pheromone. Then, the local routing table is updated according the route selected. Our algorithm can work in combinatorial stable networks (networks where the changes are sufficiently slow for the routing updates to be propagate to all the nodes) or not, because our approach works with local routing tables and the changes only must be propagated to the neighbours.

3.1 Building the AS Graph

We use the pheromone matrix of our AS graph like the routing table of each node of the network. Remember that this matrix is where the pheromone trail is deposited. Particularly, each node *i* has k_i neighbors, is characterized by a capacity C_i, a spare capacity S_i, and by a routing table $R_i=[r^i_{n,d}(t)]_{ki,N-1}$. Each row of the routing table corresponds to a neighbor node and each column to a destination node. The information at each row of node *i* is stored in the respective place of the pheromone matrix (e.g., in position *i*, *j* if k_i neighbor = *j*). The value $r^i_{n,d}(t)$ is used as a probability. That is, the probability that a given ant, where the destination is node *d*, be routed from node *i* to neighbor node *n*. We use the COP matrix of our AS graph to describe the network structure. If there are link or node failures, then the COP graph is modified to show that. In addition, in each arc of the COP graph, the estimation of the trip time from the current node *i* to its neighbor node *j*, denoted $\Gamma_i=\{\mu_{i->j}, \sigma^2_{i->j}\}$ is stored, where $\mu_{i->j}$ is the average estimated trip time from node *i* to node *j*, and $\sigma^2_{i->j}$ is its associated variance. Γ_i provides a local idea of the global network's status at node *i*. Finally, we define a cost function for every node, called $C_{ij}(t)$, that is the cost associated with this link. It is a dynamic variable that depends on the link's load, and is calculated at time *t* using Γ_i.

3.2 Defining the Transition Function and the Pheromone Update Formula

We have explained that in our decentralized model each node maintains a routing table indicating where the message must go in order to reach the final destination. Artificial ants adjust the table entries continually affecting the current network state. Thus, routing tables are represented like a pheromone table having the likelihood of each path to be followed by artificial ants. Pheromone tables contain the address of the destination based on the probabilities for each destination from a source. In our network, each ant launched influences the pheromone table by increasing or reducing the entry for the proper destination.

In our model, each node of the network is represented as a class structure containing various parameters (identification of the node, adjacent nodes, spare capacity, number of links), and Equation 3 has the following meaning: $C_f^k(t)$ is the cost of k^{th} ant's route, $\Delta\gamma_{is}^{k}(t)$ is the amount of pheromone deposited by ant k if edge (i, s) belongs to the k^{th} ant's route (it is used to update the routing table R_i in each node), and $P_{ij}^k(t)$ is the probability that ant k chooses to hop from node i to node j (it is calculated from the routing table R_i). In this way, ants walk according to the probabilities given in the pheromone tables and they visit one node every time. Ant k updates its route cost each time it traverses a link $C_f^k(t) = C_f^k(t) + C_{ij}(t)$ if $i,j \in path$ followed by ant k. In this way, an ant collects the experienced queues and traffic load that allows it to define information about the state of the network. Once it has reached the destination node d, ant k goes all the way back to its source node through all the nodes visited during the forward path, and updates the routing tables (pheromone concentration) and the set of estimations of trip times of the nodes that belong to its path (COP graph), as follows:

- The times elapsed of the path $i\text{->}d$ $(T_{i\text{->}d})$ in the current k^{th} ant's route is used to update the mean and variance values of Γ_i of the nodes that belong to the route. $T_{i\text{->}d}$ gives an idea about the goodness of the followed route because it is proportional to its length from a traffic or congestion point of view.
- The routing table R_i is changed by incrementing the probability $r^i_{j,d}(t)$ associated with the neighbor node j that belongs to the k^{th} ant's route and the destination node d, and decreasing the probabilities $r^i_{n,d}(t)$ associated with other neighbor nodes n, where $n \neq j$ for the same destination (like a pheromone trail).

The values stored in Γ_i are used to score the trip times so that they can be transformed in a reinforcement signal $r=1/\mu_{i>j}$, $r\in[0,1]$. r is used by the current node i as a positive reinforcement for the node j:

$$r^i_{i-1,d}(t+1) = r^i_{i-1,d}(t)\,(1-r)+r$$

and the probabilities $r^i_{n,d}(t)$ for destination d of other neighboring nodes n receive a negative reinforcement

$$r^i_{n,d}(t+1) = r^i_{n,d}(t)\,(1-r) \qquad\qquad for\ n \neq j$$

In this way, artificial ants are able to follow paths and avoid congestion while balancing the network load. Finally, $C_{ij}(t)$ is updated using Γ_i and considering the congestion problem (we must avoid congested nodes):

$$C_{ij}(t+1) = Ce^{-ds_j(t)}\frac{\mu_{i\to j}}{\sigma^2_{i\to j}} \tag{4}$$

where C and d are constants, and $s_j(t)$ is the spare capacity of the node j at time t. The incorporation of delay ($Ce^{-ds_j(t)}$) reduces the ant flow rate to congested nodes, permitting other pheromone table entries to be updated and increased rapidly (negative backpropagation). In the case of link failures, the algorithm avoid those nodes according to the following formula:

$$C_{ij}(t+1) = \infty \qquad (node\ j\ with\ failures) \tag{5}$$

4 Performance Evaluation of the CAS Algorithm

In this section we test our approach considering three cases, networks with static topologies (no failures), networks with constant topology changes due to node and link failures, and networks with energy constraints with and without failures.

In this experiment, we evaluate our algorithm in networks with constant topology changes introducing link and node failures. Here, if a link failure occurs and the node has more than one linkage, then the node can be reached via other path. If the node has no other link to any node in the network then a node failure occurs. We assume that link failures follow a uniform distribution and do not exceed 10% of the total number of links in the network. In the presence of a link failure, the cost of a call from source node i to destination node j will be defined as infinite (see Eqn. 5), and the probability in the proper column and row in the pheromone table is set to zero.

As in [8], we also consider the incorporation of additive noise in order to handle the so-called *shortcut* and *blocking* problems. The shortcut problem occurs when a shorter route becomes suddenly available while the blocking problem occurs when an older route becomes unavailable. In both situations, artificial ants have difficulties finding new routes, as they work guided by the pheromone tables and don't have an adequate dynamic reaction. With the inclusion of the noise factor f, ants select a purely random path with probability f and a path guided by the pheromone table with probability $(1-f)$. As shown in [8, 9, 12], the noise factor must not exceed 5%, because a noise factor greater that 5% makes the system unstable, reducing the network throughput and the performance of the routing method.

We performed simulations and compared our algorithm with the approach presented in [8] using the same partially meshed Synchronous Digital Hierarchy (SDH) network. The network has 25 nodes partially connected and all links have a capacity of 40 calls. We make random selection of call probabilities, link failure random generations, and collect data to evaluate the performance of the schemes in terms of throughput and mean delay per node. Figures 1 and 2 show these results.

In Figure 1, we show that our approach provides better performance than [8] in the presence of link failures. The mean delay per node is considerably better because we consider the congested node problem. Similarly, Figure 2 shows that the throughput response of the proposed system is better, as it handles the incoming call variations and simultaneous link failures better than [8]. Link failures essentially form a constantly changing network topology to which our agent-based algorithm seems to adapt particularly well. This actually means that the proposed routing algorithm is a good candidate for networks with constant topology changes such as mobile wireless ad hoc networks, where node mobility causes constant link failures.

We also compared our model with the traditional *Link State* routing scheme described in [10] and the Ant-Based approach proposed in [9] using the same network. In Figure 3, it is shown that our CAS scheme provides substantially better throughput performance in the presence of multiple link/node failures.

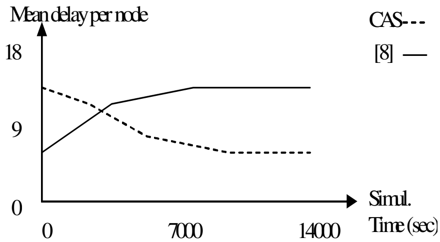

Fig. 1. Mean delay per node

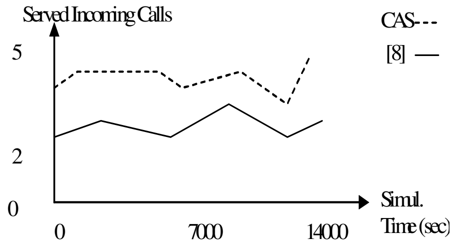

Fig. 2. Throughput response

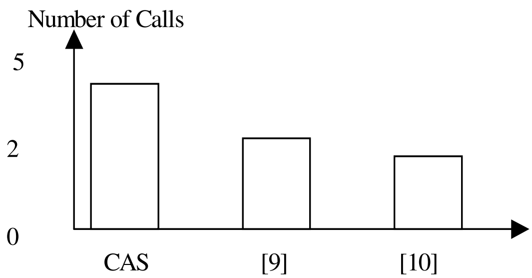

Fig. 3. Throughput response

5 Conclusions

In this work we propose a General Combinatorial Ant System-based Distributed Routing Algorithm for wired, wireless ad hoc and wireless sensor networks based on the Combinatorial Ant System. This work shows the versatility of our routing algorithm exemplified by the possibility of using the same model to solve different telecommunication problems, like dynamic combinatorial optimization problems of various sizes. Our approach can be applied to any routing problem by defining an appropriate graph representation of the solution space of the problem considered, the dynamic procedure to update that representation, and an objective function that guides our heuristic to build feasible solutions. In our approach, the dynamic environment of the combinatorial optimization problem is defined through the Combinatorial Optimization Problem matrix that forms part of the space through which the ants will walk (AS graph). Ants walk through this space according to a set of probabilities updated by a state transition and a pheromone update rule defined according to the objective function of the combinatorial optimization problem considered. Messages between nodes are replaced by ants simultaneously biasing the network parameters by laying pheromone on route from source to destination. The results show that our approach obtains good performance in the presence of multiple failures (links, nodes), contributes to congestion avoidance (load balancing), and keeps the network throughput stable.

Reference

1. Aguilar, J., Velásquez, L.: Pool M. The Combinatorial Ant System. Applied Artificial Intelligence, Vol. 18, (2004) 427-446.
2. Aguilar, J.: The Combinatorial Ant System for Dynamic Combinatorial Optimization Problems. Revista de Matematica: Teoria y Aplicaciones (to appear), Universidad de Costa Rica, San José, Costa Rica, (2005).
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: from Natural to Artificial Swarm Systems. Oxford University Press, 1999.
4. Corne, D., Dorigo, M., Glover, F.: New Ideas in Optimization. McGraw Hill, 1999.
5. Dorigo, M., Maniezzo, V., Coloni, A.: The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Trans. of Systems, Man, Cybernetics, Vol. 26, (1996) 29-41.
6. Dorigo, M., Gambardella, L.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans. on Evol. Computation, Vol. 1, (1997) 53-66.
7. Hidrobo, F., Aguilar, J.: Toward a Parallel Genetic Algorithm Approach Based on Collective Intelligence for Combinatorial Optimization Problems. Proc. of IEEE International Conference on Evolutionary Computation (1998) 715-720.
8. Mavromoustakis, C., Karatza, H.: Agent-Based Throughput Response in Presence of Node and/or Link Failure (on demand) for Circuit Switched Telecommunication Networks. Computer Communications, Vol. 27 (2004) 230-238.
9. Schoonderwoerd, R., Holland, O., Bruten, J., Rothkrantz, L.: Ant-based Load Balancing in Telecommunications Networks, Adaptive Behavior, Vol. 5 (1997) 169-207.
10. Bertsekas, D., Gallager, R.: Data Networks. Prentice Hall, 1987.
11. Aguilar, J., Ramirez, W.: Some Results of CAS for Ad Hoc Networks", *Technical Report*, CEMISID, Facultad de Ingenieria, Universidad de los Andes, 2004.
12. Appleby, S., Stewart, S.: Mobile Software Agents for Control in Telecommunication Networks. BT Technology Journals, Vol. 12, (1994) 104-113.
13. Caro, G.D., Dorigo, M.: AntNet: Distributed Stigmergetic Control for Communication Networks. Journal of Artificial Intelligence Research, Vol. 9 (1998) 317-365.
14. Gabber, E., Smith, M.: Trail Blazer: A Routing Algorithm Inspired by Ants. Proc. of the 12th IEEE International Conference on Network Protocols (2004).