Master’s Thesis

Title

Proposal and evaluation of a predictive mechanism for controlled self-organization based routing

Supervisor

Professor Masayuki Murata

Author

Naomi Kuze

February 12th, 2013

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University

Master's Thesis

Proposal and evaluation of a predictive mechanism for controlled self-organization based routing

Naomi Kuze

Abstract

To tackle problems emerging with rapid growth of information networks in scale and complexity, bio-inspired self-organization is considered one of promising design principles of a new generation network which is scalable, robust, adaptive, and sustainable. However, self-organizing systems would fall into a local optimum or never converge under some environmental conditions. Controlled or guided self-organization is a novel concept attracting many researchers in these years, where loose and moderate control is imposed on a self-organizing system to push it toward a desired state. Furthermore, in order to adapt to dynamically changing condition of information networks, each component needs to predict the future state of its neighbors from their past behaviors and to adapt its movement to conform to the predicted states. There are several investigations into self-organization with prediction in the field of biology, but its application to information network systems and technologies needs more discussion. In this thesis, we take AntNet, an ant-based routing protocol, as an example and consider a mechanism to accelerate convergence with prediction. The proposed mechanism is compared with AntNet from viewpoints of the recovery time, path length, and control overhead. Simulation results show that our predictive mechanism can accelerate convergence after environmental changes.

Keywords

Controlled Self-organization

Prediction

Routing

Ant Colony Optimization (ACO)

Contents

List of Figures

1 Introduction

Due to rapid growth of information networks in scale and complexity, conventional information network systems and technologies, which are based on central control or distributed control with global information, are to face limitations. An information network system adopting conventional control technologies suffers from the considerable overhead in managing up-to-date information to grasp dynamically changing conditions as the scale and mobility increase. Considering the problems that would emerge in future networking, there have been research activities such as FIND [1], GENI [2] in the USA, Euro-NGI/FGI [3] and FIRE [4] in Europe, and the AKARI Project [5] in Japan to establish a novel network architecture and relevant technologies. Taking into account requirements for new generation networks, i.e. scalability, adaptability, robustness, and sustainability higher than ever before, the paradigm shift is needed to organize and control the whole network system in a fully distributed and self-organizing manner. Moreover, in order to realize information network systems and technologies which can adapt to dynamically changing conditions in a timely manner, it is necessary that systems should be controlled considering the future state of systems which is predicted by observing behaviors of systems.

Self-organization is a natural phenomenon of distributed systems, where components behave individually and autonomously. In a self-organizing system, they behave in accordance with simple rules and information locally available to a component. Through direct or indirect interaction among components, a global behavior or pattern emerges on a macroscopic level without central control. In a self-organizing system, the cost of information management can be considerably reduced since none needs up-to-date information of the entire system or many other components. Moreover, local failures and small environmental changes are handled locally and immediately by neighbor components without involving the entire system. Therefore, self-organizing system can recover from failures and adapt environmental changes automatically.

However, it is pointed out that self-organizing control has some disadvantages [6]. First, in a large-scale system, it may take long time for a global pattern to emerge because it appears as a consequence of interaction between autonomous components. Second, self-organization, which uses only local information, would fall into a local optimum while a conventional system using global information can reach an optimal solution in most cases. Furthermore, a self-organizing system is not controllable in general, whereas unnecessity of control is one of the significant aspects of self-organization. For example, Ant Colony Optimization (ACO), which is a heuristic in the traveling salesman problem, is a mathematical model of foraging behavior of ants [7]. Because of the similarity, it has been adopted as a routing mechanism by many researchers [8, 9, 10]. Previous research shows that AntNet is superior to conventional mechanisms in robustness against failure, control overhead, and communication performance [11]. However, the time required for path establishment to converge depends on the length of the path, i.e. the distance between a source node and a destination node [12]. Moreover, a considerable amount of control messages generated in path establishment deplete network bandwidth and hinder data message transmission. These disadvantages and complaints about them from engineers brought an idea of controlled or guided self-organization where a self-organizing system is moderately controlled through a feedback mechanism or adaptation of control parameters [13, 14, 15, 16].

In [17], a predictive mechanism was proposed for faster consensus in flocking birds. In self-organized flocking with a predictive mechanism, each component predicts the future state of its neighbors from their past behaviors and adapts its movement to conform to the predicted states. When applied to a self-organizing system besides flocking, a predictive mechanism is considered to contribute to faster self-adaptation to environmental changes. There are several investigations into self-organization with prediction in the field of biology [18, 19], but its application to information network systems and technologies needs more discussion.

In this thesis, we take AntNet [9], which is an ant-based routing, as an example of self-organization based control and propose a predictive mechanism for ant-based routing. First, we propose a controlled self-organization based routing mechanism. The poor convergence performance of AntNet comes from the fact that ants randomly explore the whole network for a destination node and once found preferential pheromone reinforcement is performed on paths established across the whole. Then we consider to reduce convergence time by limiting the area of exploration of ants. In our mechanism, the whole network is autonomously divided into subareas by ants and path finding and establishment are performed within a subarea. The path from a source node to a destination node can be formed by concatenating sub-paths. Though simulation experiments, we show that path establishment is accelerated with moderate controls. Second, we propose a predictive mechanism for AntNet. In an ant-based routing mechanism, a shorter path collects more pheromones than longer paths. Then the preferentially accumulated pheromones attract more ants which further deposit pheromones on the path. Therefore, the increase rate of pheromone implicitly indicates the goodness of a path. In our mechanism, each node predicts a path which will obtain the largest amount of pheromone from historical information about pheromone accumulation. Then, it boosts pheromone accumulation on the predicted path to have faster convergence. We show that prediction helps adaptation to environmental changes through simulation experiments.

The reminder of this thesis is organized as follows. First, we describe related work in section 2. In section 3, we describe AntNet. In section 4, we propose and explain our controlled ant-based routing mechanism with autonomous zoning. Then we propose and explain our predictive mechanism for ant-based routing in section 5 and give simulation results and discussion of our mechanism in section 6. Finally, in section 7, we provide conclusion and future work.

2 Related Work

In this section, we introduce research on controlled self-organization, prediction, and ant-based routing.

2.1 Controlled Self-organization and Prediction

Controlled or guided self-organization is a novel scheme where a self-organizing system is moderately controlled to achieve better performance or faster convergence than uncontrolled self-organization [13, 14]. For example, a control mechanism is adopted for self-organized Virtual Network Topology (VNT) control in WDM networks [15] and potential-based routing in a large scale wireless sensor network [16]. Both introduce the concept of an observer/controller architecture [20]. An observer observes a system to measure, quantify, and predict the emergent behavior using general measures. Then, the controller controls the system by for example regulating parameters that a self-organization mechanism uses, so that the desired behavior emerges. Whereas the observer/controller architecture requires a third-party component which observes and controls the behavior of a self-organizing system, it retains the certain freedom of control leaving the autonomy and independency to self-organizing system components. Self-organized VNT control methods have highly adaptability against environmental changes but it is insufficient for multiple VNTs environments to use only the self-organized VNT control methods because self-organized VNT control method is not careful about other VNTs. In [15], a controlled self-organization method of VNT controls to achieve adaptive and efficient VNT controls for multiple VNTs environments is proposed. A observer does not observe the details of the network but collects the activities that represent the conditions of VNTs. Using the collected information, a controller sends a feedback to each VNTs.

Self-organized systems based on only the current available information of the network are shown to have high scalability, robustness, and adaptability. However, biologists have shown that each individual has a ability to predict the future state, which influences its behavior. In [17], a predictive mechanism was proposed for faster consensus in flocking birds. Flocking is the behavior exhibited when a group of birds, which is called a flock, flight in parallel. With a predictive mechanism, each bird predicts the future state, i.e. location of neighbor birds, only from past behaviors of them and adapts its movement to conform to the predicted states. It implies that ‘*his-*

torical local information is equivalent to current global information' [17]. Mathematical analysis and numerical simulations prove that a predict mechanism contributes to both the exploration of emergent behaviors and the design of autonomous and reliable consensus networks. The consensus performance is significantly enhanced while the communication energy or cost is effectively reduced with a predictive mechanism. Other studies of a predictive mechanism are conducted for the foraging behavior of honey bees [18], and the forthcoming perception in the frontal cortex [19].

2.2 Ant-based Routing

Biological systems are inherently self-organizing and the biology is one of mines of self-organization models that can be applied to information networking [21, 22]. For example, foraging behavior of ants is well-known biological self-organization [7]. An ant coming out from a nest randomly wanders around looking for food. When it finds food, it returns to the nest while leaving chemical substances called *pheromone* on the ground. Pheromones laid on the ground attract other ants and guide them to the food. Since ants are not deterministic, they do not necessarily follows an existing pheromone trail and often make multiple trails. However, pheromone evaporates over time, thus a shorter trail has more pheromone than longer trails because a shorter trail gets marched over more frequently. As such, more ants traverse a shorter trail and those ants further add pheromone on the trail. This feedforward-based reinforcement mechanism makes ants concentrated on the shortest path. Moreover, since pheromone stimulation is not deterministic, there still exist ants that reach the food by taking other routes after convergence. Therefore, even if one route becomes not available, ants can find food soon by traversing other route which has pheromones. Ant System, which is the first ACO algorithm, and its variant were proposed in [23, 24] as a heuristic algorithm for stochastic combinatorial optimization problems such as a traveling salesman problem, a quadratic assignment problem, and a job-shop scheduling problem.

In information networks, information generated at a host is divided into packets and sent to a receiving host traversing a path established between the sender and receiver. Because of the similarity, many routing mechanisms are proposed based on the foraging behavior of ants or its mathematical model (ACO) [25, 26]. The first ant-based routing mechanism employed Ant-Based Control (ABC) [8]. In ABC algorithm, a number of agents called ants are continuously exploring the network from random sources to random destinations. Arriving at a node, they update the pheromones to their *source node* for all its neighbor node, which corresponds to the probabilities for ants to select each neighbor node. AntNet is another pioneering mechanism [9], which we use as a benchmark in this thesis. The latest variant is AntHocNet and it is designed for mobile ad hoc network [10]. Simulation results prove that AntNet distributes the load on the different available paths, which realizes high throughput and low delays of packets.

3 AntNet

We use AntNet as a basis of our investigation of self-organization with prediction. In this section, we give a summary of a mechanism of AntNet.

3.1 Overview

AntNet [9] is an ACO algorithm for adaptive best-effort routing in packet-switched wired networks. AntNet introduces two types of ants, i.e., *forward ants* and *backward ants*. A source node proactively launches mobile agents called forward ants at regular interval. A forward ant stochastically selects a neighbor node to visit in accordance with the amount of pheromones. On a way to a destination node, a forward ant records its path and the time of arrival at each node in order to evaluate the quality of the travelled path.

When a forward ant arrives at the destination node, it changes to a backward ant. A backward ant returns to the source node on the reverse path of the forward ant, updating pheromone values along the way. When the path has better quality, i.e. smaller delay, a backward ant increases a pheromone value for the neighbor node it came more.

Each data packet selects a neighbor node as a next hop node according to the pheromone values that backward ants updated. Since a neighbor node with a larger pheromone value is more likely to be selected, a data packet reaches a destination node following a shorter path.

3.2 Self-Organization based Path Establishment and Maintenance

In AntNet, Each node has a pheromone table $\mathcal{T}^k$ as routing information. $\mathcal{T}^k = \{\mathcal{T}_d^k\}$ where $\mathcal{T}_d^k$ is a list of pheromone values τ_{nd}^k for all neighbor node $n \in N_k$ regarding destination node d, i.e. $\mathcal{T}_d^k = \{\tau_{nd}^k\}$. N_k is a set of neighbor nodes of node k. Source node s establishes and maintains a path to destination node d by sending *forward ants* at regular intervals. A forward ant stochastically selects a next hop node to visit. The probability p_{nd} that neighbor node $n \in N_k$, where N_k is a set of neighbor nodes of node k, is selected as a next hop node of node k for destination node d is given as follows.

If there is no pheromone information for destination node d at node k, a next hop node is randomly chosen.

$$p_{nd} = \begin{cases} 1, & \text{if } |N_k| = 1 \\ \frac{1}{|N_k|-1}, & \text{if } |N_k| > 1 \wedge n \neq v_{i-1} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Otherwise, selection is performed based on the pheromone value τ_{nd}.

$$p_{nd} = \begin{cases} 1, & \text{if } |N_k| = 1 \\ \frac{1}{|N_k|-1}, & \text{if } |N_k| > 1 \wedge \forall n \in V_{s\to k} \wedge n \neq v_{i-1} \\ \frac{\tau^k_{nd}+\alpha l_n}{1+\alpha(|N_k|-1)}, & \text{if } |N_k| > 1 \wedge n \notin V_{s\to k} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$V_{s\to k} = \{s, v_1, v_2, \cdots, v_{i-1}\}$ is a list of nodes that the forward ant has visited before arriving at node k at the i-th step and v_{i-1} is an identifier of the $(i-1)$-th node on the path. l_n is a variable indicating the degree of congestion for neighbor node n at node k, which is given by $1 - \frac{q_n}{\sum_{j\in N_k} q_j}$ and q_n is the number of messages waiting in a sending buffer for neighbor node n. $\alpha \in [0, 1]$ is a coefficient. A larger α allows forward ants to select a next hop node in accordance with local traffic condition. As a consequence, path convergence becomes hard to accomplish. On the contrary, with α close to 0, a path traversing congested links would be established. A forward ant whose travelled hop count reaches the predetermined TTL is discarded at a node.

A forward ant changes to a *backward ant* when it reaches the destination node d and returns to the source node s following the path that the forward ant traversed while updating pheromone values at visited nodes. The pheromone value τ^k_{nd} for neighbor node $n \in N_k$ at node k is updated by Eq. (3).

$$\tau^k_{nd} \leftarrow \begin{cases} \tau^k_{nd} + r(1-\tau^k_{nd}), & \text{if } n = f \\ \tau^k_{nd} - r\tau^k_{nd}, & \text{otherwise} \end{cases} \tag{3}$$

f corresponds to the previous node that the backward ant visited just before arriving at node k, i.e. the first node of the path from the node to the destination node. r reflects the goodness of the path, on the transmission delay from node k to the destination node d. The shorter the path is, the larger r is. Consequently, the shortest path among paths that forward ants found has the largest amount of pheromones and attracts most of forward ants.

The parameter r, which determines the increasing amount of pheromones, is evaluated from the trip time $T_{k\to d}$ and the local statistical model $\mathcal{M}^k = \{\mathcal{M}_d^k\}$, where $\mathcal{M}_d^k = \{W_k^d, \mu_d^k, \sigma_d^k\}$.

$$r = c_1\left(\frac{W_k^d}{T_{k\to d}}\right) + c_2\left(\frac{I_{sup} - I_{inf}}{(I_{sup} - I_{inf}) + (T_{k\to d} - I_{inf})}\right) \tag{4}$$

$T_{k\to d}$ is the ant's trip time from node k to destination node d. W_k^d is the best traveling time of ants from node k to destination node d over the last observation window of size w samples, and (μ_d^k, σ_d^k) are the average and dispersion of the traveling time of ants over the last observation window. I_{sup} and I_{inf} are estimates of the limit of an approximate confidence interval for μ, which are given Eqs. (5), (6).

$$I_{inf} = W_k^d \tag{5}$$

$$I_{sup} = \mu_d^k + z(\sigma_d^k/\sqrt{w}),\ \text{with}\ z = 1/\sqrt{1-\gamma} \tag{6}$$

c_1, c_2, and γ are coefficients, and (c_1, c_2, γ) is set at $(0.7, 0.3, 1.7)$ in [9].

3.3 Transmission of Data Messages

A data message selects a next hop node based on pheromone values, where the selection probability R_{nd}^k that neighbor node n is chosen as a next hop node for destination node d is given as $\frac{(\tau_{nd}^k)^\epsilon}{\sum_{j\in N_k}(\tau_{jd}^k)^\epsilon}\ (\epsilon \geq 0)$. Therefore, data messages follow the shortest path established by forward and backward ants.

4 Controlled Ant-based Routing with Autonomous Zoning

In this section, we propose our controlled ant-based routing mechanism. As control placed on self-organizing routing, we consider dividing a region to explorer into zones. By limiting the range of search for a path to a destination node, we can accelerate path establishment in a large scale network.

4.1 Overview

As stated in section 1, it is a well-known problem of ant-based routing that it takes long time to find a path and converge to a good path in a large scale network. Therefore, it is effective to limit of search for a path to a destination for faster path establishment, while letting ants behave autonomously and self-organize themselves. When we assume a wireless network, it is difficult to place a central unit, i.e. observer and controller, and observe the behavior of the entire network for limited network resources. Therefore, we consider an autonomous zoning mechanism.

Each node has a pheromone table $\mathcal{T}^k$ as routing information. $\mathcal{T}^k = \{\mathcal{T}_d^k\}$ where $\mathcal{T}_d^k$ is a list of pheromone values τ_{nd}^k for all neighbor node $n \in N_k$ regarding destination node d, i.e. $\mathcal{T}_d^k = \{\tau_{nd}^k\}$. N_k is a set of neighbor nodes of node k. The pheromone value is updated by backward ants, but we set limits on the amount of pheromones that a node can deposit. The pheromone value is used for next-hop selection by ants and data messages.

In our mechanism, there are two types of forward ants, called *exploration ants* and *maintenance ants*. Figure 1 illustrates behavior of three types of ants. To establish a path to a destination node, a source node sends exploration ants at regular intervals. An exploration ant wanders looking for a destination node like a forward ant of AntNet, but it has a new role to determine a border of zones. When the number of hops it takes reaches the predetermined limit, an exploration ant sets the halfway node a *border node*. We call an area surrounded by border nodes *zone*.

An exploration ant setting a border node changes to a backward ant and returns to the source node while leaving pheromones on visited nodes in similar to a backward ant of AntNet. Then, on receiving the backward ant the source node begins sending maintenance ants in addition to exploration ants to maintain and improve the path to the border node. Each maintenance ant goes to a border node by selecting next hop nodes in a stochastic manner like a forward ant and returns to the source node as a backward ant to update pheromone values. Simultaneously an established

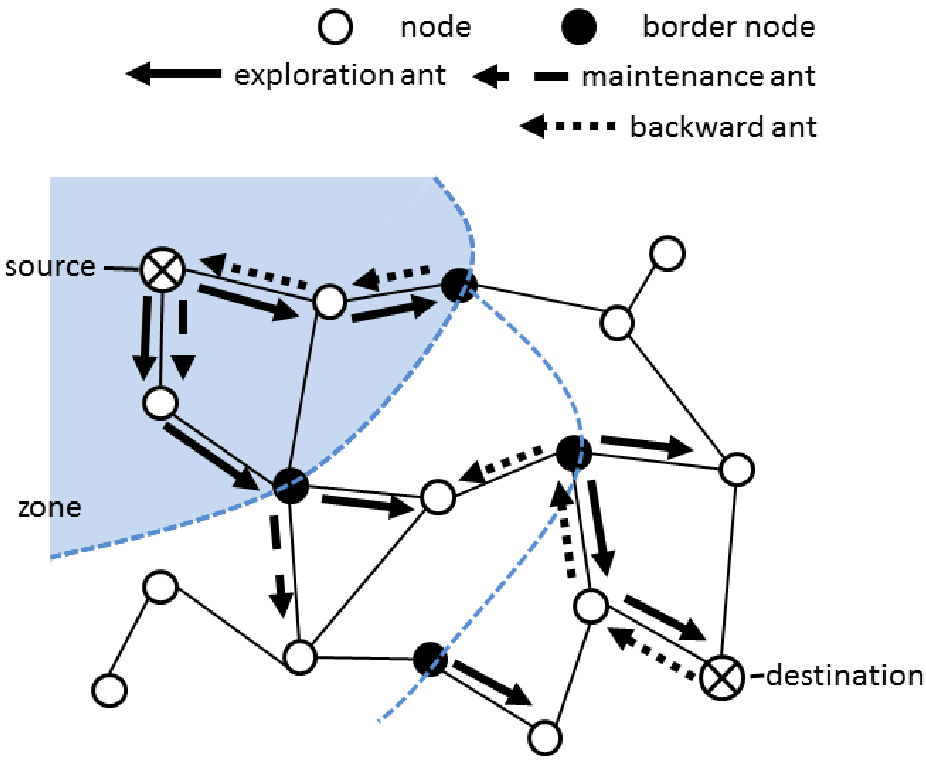


Figure 1: Snapshot of controlled ant-based routing with autonomous zoning

border node begins sending exploration ants to look for a destination node. They also set border nodes and go back to the border node. On their return, the border node sends maintenance ants to maintain paths to new border nodes. By repeating the exploration, border nodes are scattered over a network among which the shortest path is established by maintenance ants.

When any exploration ant finds a destination node, a backward ant returns to its originating border node. It raises the ceiling of the amount of pheromone at nodes on the path so that those nodes can deposit more pheromones than nodes on other paths. The backward ant eventually returns a border node from which the corresponding forward ant departed. Then backward ants which depart from the border node also begin raising the limit at nodes on the path to the previous border node. Consequently, a chain of paths from the source node to the destination node have stronger pheromones than other paths leading to nodes other than the destination node. Finally, the source node starts emission of data messages.

4.2 Path Exploration and Autonomous Zoning by Exploration Ant

When a source node intends to start a new session to a destination node for which it does not have routing information, it first generates $|N_k|$ exploration ants and sends one exploration ant for each neighbor node at the same time. Then, it keeps sending an exploration ant per interval of Δt. An exploration ant looks for a destination until the number of travelled hops reaches the given limit of $2h$ hops. If there is no pheromone for destination d on the arrived node, an exploration ant chooses a next hop node at random by Eq. (1). Otherwise, it chooses a next hop node based on the probability p_{nd} given not by Eq. (2) but by Eq. (8), which will be explained later. In the course of the search, an exploration ant records all visited nodes in a visiting order.

If an exploration ant cannot find a destination node or does not arrive at a border node within $2h$ hops, it returns to the node visited at h-th hop from the source node and appoints the node as a border node. A border node remembers h as its distance from the source node. To avoid placing border nodes next to each other, an exploration ant does not set a border node if there exists any of source, border, or destination node in the immediate vicinity. In this case, an exploration ant immediately dies. Otherwise, an exploration ant which set or arrived at a border node changes to a backward ant and returns to the source node. Each node that receives a backward ant from neighbor node f updates pheromone values for destination node d by

$$\tau_{nd}^{k} \leftarrow \begin{cases} \min(\tau_{nd}^{k} + r(1-\tau_{nd}^{k}), \theta), & \text{if } n = f \\ \max(\tau_{nd}^{k} - r\tau_{nd}^{k},\ \ \tau_{nd}^{k} - r\tau_{nd}^{k} + \frac{\tau_{nd}^{k}+r(1-\tau_{nd}^{k})-\theta}{|N_k|-1}), & \text{otherwise} \end{cases} \tag{7}$$

where θ is an upper limit of the amount of pheromone and it is set as $\theta = \theta_l$ $(0 < \theta_l < 1)$. An exploration ant reaching a destination node also becomes a backward ant, but its behavior is different from a backward ant of AntNet and will be described in section 4.4.

Similarly to a source node, a new border node generates $|N_k|$ exploration ants and sends one exploratory ant for each neighbor node at the same time. Then, it keeps sending one exploration ant per interval of Δt. An exploration ant sent from a border node looks for a destination node, sets a new border node, and changes to a backward ant as an exploration ant from a source node does. In this way, border nodes are distributed in a network. An exploration ant remembers the distance of the originating border node from the source node. A new border node created by the exploration ant considers its distance as the sum of the distance of the previous border node that the exploration ant remembers and the number of hops that the exploration ant made from the previous border node. Therefore, the distance that a border node remembers is always a multiple of h. The distance information is used to avoid making a path going back to the source node. When an exploration ant arrives at a border node which is not the originating border node, it first compares its distance from the source node with that of its originating node. If the arrived border node is closer to the source node, it immediately dies. Otherwise it returns to the originating border node as a backward ant.

In order to scatter border nodes, i.e. bases of exploration, over a network an exploration ant selects a next hop node by avoiding pheromones that show a path to other border node. For this purpose, we use the following equation proposed in [27] to give the probability p_{nd} that an exploration ant whose previous node is f chooses neighbor node $n \in N_k - \{f\}$ as a next hop node for destination node d at node k.

$$p_{nd} = \frac{\frac{1}{\tau^k_{nd}}}{\sum_{j \in N_k} \frac{1}{\tau^k_{jd}} - \frac{1}{\tau^k_{fd}}} \tag{8}$$

If node f is the only neighbor node of node k, an exploration ant moves to node f.

4.3 Path Maintenance by Maintenance Ant

On reception of a backward ant, a source node and border nodes start sending maintenance ants to all neighbor nodes whose pheromone value τ^k_{nd} is larger than the initial pheromone value, i.e. $\frac{1}{|N_k|}$, at regular intervals Δt. A maintenance ant chooses a next hop in accordance with the amount of pheromones to maintain an existing path to a border node. A maintenance ant whose travelled hop count reaches the given TTL, i.e. $2h$ hops, is discarded at the node.

The probability p_{nd} that a maintenance ant which arrives at node k from node f chooses neighbor node $n \in N_k - \{f\}$ as a next hop for destination node d is given by Eq. (9). τ^k_{nd} shows the pheromone value for neighbor node $n \in N_k$ at node k for destination node d.

$$p_{nd} = \frac{\tau^k_{nd}}{\sum_{j \in N_k} \tau^k_{jd} - \tau^k_{fd}} \tag{9}$$

If node f is the only neighbor node of node k, a maintenance ant moves to node f.

Similarly to an exploration ant, a maintenance ant reaching a destination node becomes a backward ant to raise the ceiling of pheromone values and leave pheromones as will be described in section 4.4. A maintenance ant becomes a backward ant to reinforce its path when it arrives at a border node except for the case that the arrived border node is closer to the source node than the originating border node.

4.4 Construction of Paths between Source and Destination

In our mechanism, paths are constructed and maintained on a per zone basis. A path from a source node to a destination node is established as a chain of those sub-paths of zones which lie between them. So that maintenance ants and data messages are concentrated on the path leading to the destination node, we need to differentiate it from other paths on each node. For this purpose, we put another upper limit θ_h $(0 < \theta_l < \theta_h = 1)$ on the amount of pheromone deposited at a node.

An upper limit of θ_l is applied to all τ_{nd}^k when node k makes entry T_d^k for a new session to destination node d. Eventually node k receives a backward ant departing from a destination node or a next border node to a destination node if its zone is located between the source node and the destination node. Then, it raises the ceiling to θ_h and stops sending exploration ants. At the same time, the border node sets a timer at ten times as long as Δt, an interval of ant emission. The timer is restarted when it receives either of an exploration ant, a maintenance ant, or a backward ant with a role of threshold raising. When the timer expires, the threshold returns to θ_l.

In addition to the favoring mechanism, we have a pruning mechanism to reduce redundant border nodes after a path from a source node to a destination node is found. A border node using θ_l starts a pruning timer at 100 times as long as Δt when it receives an exploration ant or a maintenance ant departing from a border node using θ_h. It implies that the border node is located at the border of zone having a path to the destination node but not on the path itself. The pruning timer is cancelled and discarded when the border node can raise the threshold to θ_h. When the pruning timer expires on the other hand, it stops generating both of exploration ants and maintenance ants. Then, it moves to a standby state. In the standby state, as explained in section 4.1, a node set another timer, waits for its expiration, and discards the pheromone list T_d^k on timer expiration. In addition a border node which receives a backward ant departing from a border node in the standby state also starts a pruning timer. It eventually moves to the standby mode unless it raises the threshold until timer expiration and finally discards T_d^k. In this way, zones next to those zones constituting a path to the destination node disappear first and then pruning proceeds to the edge of a network.

4.5 Evaluation of Controlled Ant-based Routing with Autonomous Zoning

We evaluate and compare our mechanism with AntNet and HOPNET [28] from viewpoint of the convergence time, path length, and control overhead in order to show that path establishment is accelerated with moderate controls. HOPNET is a hybrid mechanism adopting ant-based routing in proactive and reactive manners. In HOPNET, zones are constructed per node in accordance with the proximity of nodes. The size of zones is determined by a parameter $radius$ and a zone of node k consists of a set of nodes which can be reached in $radius$ hops from node k. Each node proactively maintains paths for nodes within its zone and reactively finds a path for a destination node if a destination node is outside its zone.

4.5.1 Simulation Settings

We change the size of network while keeping the node density. We experimentally distribute 150 nodes at random locations in the area of 200 m$\times$200 m. We call this setting $scale = 2$. Then, we consider different sizes from $scale = 1$ (100 m$\times$100 m) to 10 (1,000 m$\times$1,000 m). Independently of the size of network we set the communication range at 30 m and the one-hop transmission delay at 2 msec. We appoint a node at the top-left corner as a source node and one at the bottom-right corner as a destination node. The interval Δt of control message emissions is set at 10 msec in AntNet and our mechanism. The parameter h is set at 3 in our mechanism. Other parameters of AntNet and HOPNET are set in accordance with their default settings [9, 28]. Data messages are not generated in simulation experiments.

Regarding performance measures, the convergence time is defined as the time from the beginning of a simulation run where no routing information exists in a network till when the same path is selected for 10 consecutive times or a cyclic selection of the fixed set of paths in the same order, e.g. path A, path B, path C, path A, path B, and path C, for 100 consecutive times. Convergence check is done everytime a backward ant reaches a source node. In the case of HOPNET, reactive path establishment is performed for a destination node outsize a zone. Therefore in our simulation we first allow nodes to proactively establish paths within zones for 300 msec. Then we start construction of a path from the source node and the destination node. Taking into account this the convergence time of HOPNET is defined as the time from 300 msec to convergence. The path length is defined as the number of hops of a created path. The control overhead is defined as the

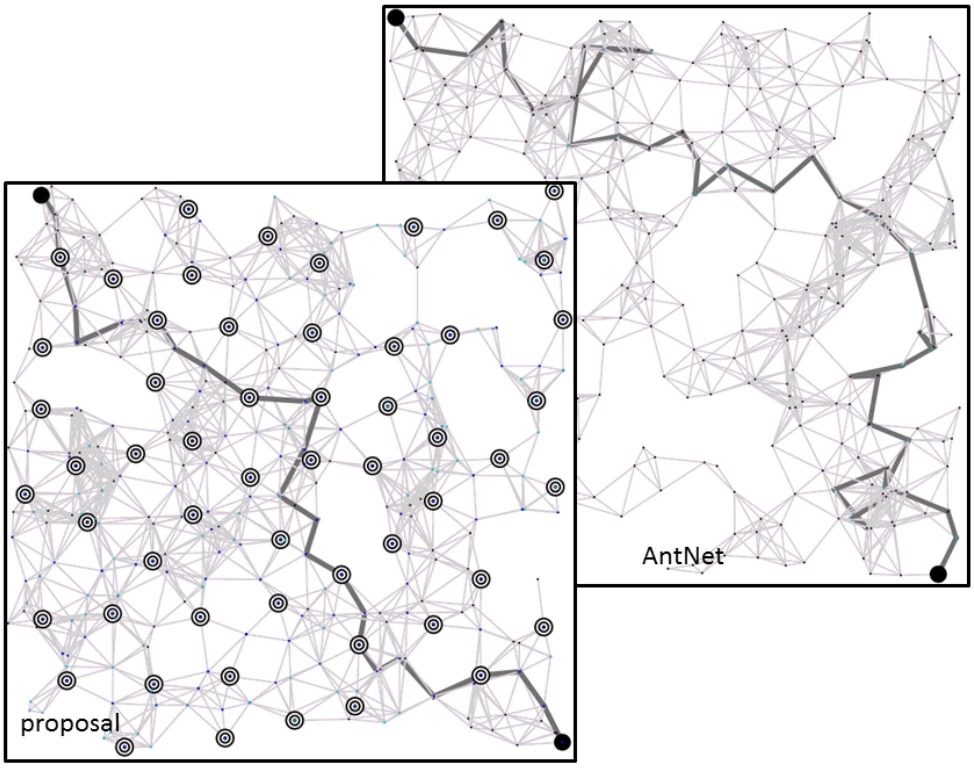


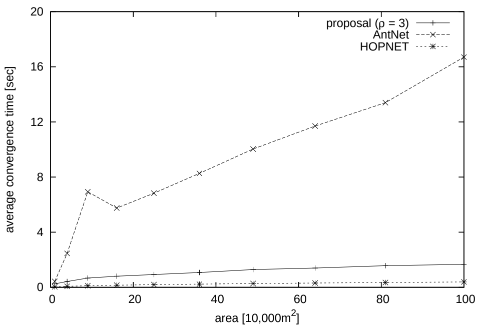
Figure 2: Examples of established path

total number of travelled hops of control messages until convergence.

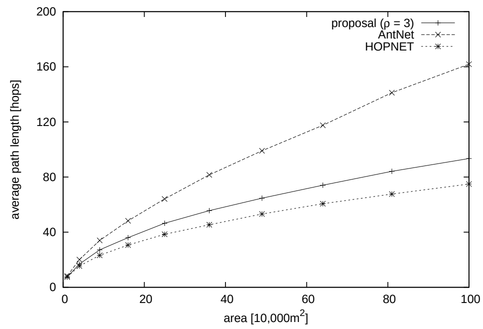
Figure 2 shows an example of paths created by AntNet and our mechanism in networks of $scale = 3$. Each small dot corresponds to a node and double circles are border nodes. A filled circle at a top-left corner is a source node and one at a bottom-right corner is a destination node. Each thin line means that nodes of its ends can communicate with each other, that is, they are neighbors. Thick lines show a path established by ants. In the figure, the path length of AntNet is 49 hops and that of our mechanism is 25 hops. In the following figures, we show results averaged over 100 simulation runs for each $scale$ except for cases that convergence cannot be achieved by the end of a simulation run.

4.5.2 Results and Discussion

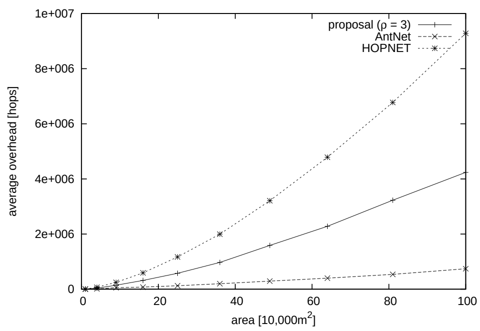
Figure 3(a) shows the average convergence time against different $scale$ setting. As shown in the figure, the convergence time of our mechanism is much smaller than that of AntNet because path



(a) Convergence time



(b) Path length



(c) Cumulative overhead

Figure 3: Performance of controlled ant-based routing mechanism with autonomous zoning

convergence is accelerated by zoning. In ant-based routing, a path has to accumulate pheromones as fast as possible and distinguish an appropriate neighbor node from other neighbors in next-hop selection for faster convergence. However, in AntNet, nodes on a path have to wait long for a backward ant to come from a distant destination node in order to increase the amount of pheromone. While it is waiting, forward ants visiting the node randomly choose a next hop node. Then, there appear multiple paths from a source node to a destination node, which disturbs fast convergence. In addition, a forward ant explores the whole area with AntNet. It delays the discovery of a path. On the contrary, because of zoning, exchanges of forward and backward ants are limited within a zone with our mechanism. The length of path connecting a pair of a source node and a border node, two border nodes, or a border node and a destination node, is considerably shorter than a path between a source node and a destination node. As a result, the speed of reinforcement of a path is much faster with our mechanism than AntNet. In conclusion our mechanism is more scalable than AntNet.

HOPNET converges the fastest among the three mechanisms. The first reason is that paths within a zone have already been established before a session starts in our simulation. In addition, with HOPNET a source node sends forward ants to all nodes at a border of its zone if it does not have routing information for a destination node. The greedy exploration is repeated by border nodes and as a result the fast path discovery is accomplished. Finally, HOPNET uses a path which is found first and does not search for a better path. Therefore, when a path is found, HOPNET converges.

We also evaluate the path length constructed by each mechanism in Fig. 3(b). Since HOPNET conducts greedy and exhaustive search, the length of a path becomes close to the optimal shortest path. Compared with AntNet, our mechanism constructs shorter paths. Whereas exploration ants with AntNet search for a destination node making a random walk across the whole network and often make an indirect path as shown in Fig. 2, our mechanism gradually expands the scope of exploration by pushing the front line consisting of border nodes from which exploration ants perform local search. Furthermore, our mechanism does not allow exploration ants to go back to a source node. This mechanism also contributes in making a shorter path than AntNet. However, a chain of border nodes does not always form the shortest path from a source node to a destination node. As a result, the path length with our mechanism is slightly longer than that of HOPNET. Figures 3(a) and 3(b) show that HOPNET is the best among three, but it is at the sacrifice of considerable overhead as will be shown in the next.

Finally, Fig. 3(c) shows the average overhead, i.e. the total number of hops of control messages before convergence. A reason why HOPNET incurs significant overhead is that it conducts a greedy and exhaustive search by multicasting control messages. It easily causes heavy congestion and both of path establishment and data communication would fail. In conclusion HOPNET is not scalable to a large network with many nodes and many sessions. On the contrary, the instantaneous overhead is small with our mechanism. A reason that the overhead per unit time is quite small with AntNet is that it takes long to converge.

In a self-organizing system, the global pattern emerges as a consequence of mutual interaction among individuals. In a case of ant-based routing, a path is constructed through interaction among ants mediated by pheromones. In this section, as an example of controlled self-organization, we propose and evaluate a mechanism to accelerate path convergence of AntNet by limiting the search space. Simulation results show that our proposal can facilitate path establishment and make ant-based routing more scalable to the size of network.

5 Predictive Mechanism for Ant-based Routing

In this section, we propose a predictive mechanism for AntNet. We consider prediction only from pheromone changes and control by updating pheromones independently of internal control in AntNet.

5.1 Overview

As shown in section 4, convergence is accelerated in self-organizing systems with moderate control. However, it is difficult for components to adapt faster to dynamically changing conditions of networks only with local current information even with moderate control. Therefore, we take AntNet as example of self-organization based control and consider a predictive mechanism in which components observe their past behaviors, predict the future state of the system, and then control their behaviors in accordance with the predicted future information.

In AntNet, the pheromone value τ^k_{nd} for neighbor node $n \in N_k$ at node k is updated by Eq. (3). In order to observe pheromone changes, we give time evolution of pheromone values by

$$\tau^k_{nd}(t) = \begin{cases} \tau^k_{nd}(t-1) + r(1 - \tau^k_{nd}(t-1)), & \text{if } n = f \\ \tau^k_{nd}(t-1) - r\tau^k_{nd}(t-1), & \text{otherwise} \end{cases} \tag{10}$$

where $\tau^k_{nd}(t)$ corresponds to t-th pheromone value of node k for neighbor node $n \in N_k$ regarding destination node d.

Each node has a pheromone table $\mathcal{T}^k$ as routing information. $\mathcal{T}^k = \{\mathcal{T}^k_d\}$ where $\mathcal{T}^k_d$ is a list of pheromone values τ^k_{nd} for all neighbor node $n \in N_k$ regarding destination node d, i.e. $\mathcal{T}^k_d = \{\tau^k_{nd}\}$. N_k is a set of neighbor nodes of node k. At the beginning, τ^k_{nd} is initialized to $\frac{1}{|N_k|}$. The pheromone value is updated by backward ants, but we set limits on the amount of pheromones that a node can deposit. The pheromone value is used for next-hop selection by ants and data messages.

Each node also has a increase rate table $\mathcal{E}^k$ for prediction. $\mathcal{E}^k = \{\mathcal{E}^k_d\}$ where $\mathcal{E}^k_d$ is a list of increase rates of the pheromone value e^k_{nd} for all neighbor node $n \in N_k$ regarding destination node d. At the beginning, $\mathcal{E}^k_d$, e^k_{nd} is initialized to 0. The increase rate $\mathcal{E}^k_d$ is updated by backward ants, and is used for predictive control.

In this thesis, we consider two approaches for prediction. First, each node performs prediction based on locally available information. In this approach, each node uses only its pheromone table for prediction. It observes pheromone changes for its neighbor nodes, calculates increase rates of the pheromones, and predicts a neighbor node that will collects more pheromones. Second, each node performs prediction using information of neighbor nodes, more specifically, increase rate table $\mathcal{E}^k$. It regularly sends agents to all its neighbor nodes and obtains their pheromone information.

5.2 Prediction based on Locally Available Information

In this approach, prediction is based on locally available information. It observes pheromone changes for its neighbor nodes, calculates increase rates of the pheromones, and predicts a neighbor node that will collects more pheromones.

First, each node observes changes of pheromone values in order to predict a path that will collect more pheromones. Node k that receives a backward ant updates the increase rate $e^k_{nd} \in [0, 1]$ of all its neighbor nodes $n \in N_k$ regarding destination node d by

$$e^k_{nd} \leftarrow \begin{cases} (1-\beta)e^k_{nd} + \beta, & \text{if } \Delta\tau^k_{nd}(t) > 0 \\ (1-\beta)e^k_{nd}, & \text{otherwise} \end{cases} \tag{11}$$

where $\beta \in [0, 1]$ is a parameter which determines the weight of one time increment of pheromones and $\Delta\tau^k_{nd}(t)$ corresponds to $\tau^k_{nd}(t) - \tau^k_{nd}(t-1)$. When the increase rate of pheromones for neighbor node n becomes large, node k considers that the neighbor node n will collect more pheromones. Specifically, when the increase rate of pheromones for neighbor node n is larger than those of other neighbor nodes and its value exceeds 0.5, we identifies the neighbor node as a predicted neighbor node n_p. Node k boosts pheromone accumulation on the predicted neighbor node n_p for faster convergence.

Second, each node predicts the increasing amount of pheromones. From Eq. (10), if node k receives a backward ant from neighbor node n, the $(t+1)$-th pheromone value of neighbor node n is given by Eq. (12) as a function of $\tau^k_{nd}(t)$ and $\tau^k_{nd}(t-1)$.

$$\tau^k_{nd}(t+1) = \tau^k_{nd}(t) + \frac{\Delta\tau^k_{nd}(t)}{1-\tau^k_{nd}(t-1)}(1-\tau^k_{nd}(t)) \tag{12}$$

Then the increasing amount of pheromone is predicted to be $\frac{\Delta\tau^k_{nd}(t)}{1-\tau^k_{nd}(t-1)}(1-\tau^k_{nd}(t))$. Consequently, node k updates its pheromone table by Eq. (13) at regular intervals.

$$\tau^k_{nd}(t+1) = \begin{cases} \tau^k_{nd}(t) + \frac{\Delta\tau^k_{n_pd}(t)}{1-\tau^k_{n_pd}(t-1)}(1-\tau^k_{nd}(t)), & \text{if } n = n_p \\ \tau^k_{nd}(t) - \frac{\Delta\tau^k_{n_pd}(t)}{1-\tau^k_{n_pd}(t-1)}\tau^k_{nd}(t), & \text{otherwise} \end{cases} \tag{13}$$

5.3 Prediction based on Information of from Neighbor Nodes

In this approach, each node performs prediction based on information of neighbor nodes. Each node regularly sends agents to all its neighbor nodes and obtain their pheromone information. With this approach, each node boosts pheromone accumulation on its neighbor node, where its pheromone table is converging.

Each node k sends *predictive ants* to all its neighbor nodes at regular intervals. A predictive ant that arrives at a neighbor node remembers the node's increase rate table and returns to node k. Node k that receives the node $f \in N_k$'s increase rate table $\mathcal{E}^f$ updates pheromone table regarding destination node d by

$$\tau^k_{nd}(t+1) = \begin{cases} \tau^k_{nd}(t) + b(\pi - \tau^k_{nd}(t)), & \text{if } n = f \\ \frac{\pi\tau^k_{nd}(t)}{\sum_{x\in N_k-\{f\}}\tau^k_{xd}(t)}, & \text{otherwise} \end{cases} \tag{14}$$

if the max value of the increase table of node f regarding destination node d, i.e. $\mathcal{E}^f_d \in \mathcal{E}^f$, exceeds 0.5. The parameter π is the upper limit of pheromone value which is increased by predictive control and the parameter b determines increasing amount of pheromones. The increase rate of pheromones are calculated at each node in the same way as mentioned in section 5.2.

6 Performance Evaluation

In order to evaluate robustness and adaptability of our prediction mechanism, we evaluate the time to recover from environmental changes such as node failures and traffic changes.

6.1 Simulation Settings

We distribute 100 nodes on a 10×10 grid with separation of 30 m. We appoint a node at the top-left corner as a source node and one at the bottom-right corner as a destination node. The communication range is set at 30 m. Therefore, each node can communicate with four neighbors. The interval of prediction or predictive ant emissions is set at 100 msec and we change the interval of forward ant emissions from 100 msec to 1 sec. The coefficient α in Eq. (2) is set at 0.004. Other parameters of AntNet are set in accordance with their default settings [9].

In order to establish the path considering the traffic, l_n, which is a variable indicating the degree of congestion for neighbor node n at node k, is given by

$$l_n = 1 - \frac{\lambda_{kn} T_s}{\sum_{j \in N_k} \lambda_{kj} T_s} \tag{15}$$

where λ_{uv} corresponds to the average arrival rate of data packets to the queue for sending to node v at node u and T_s corresponds to the average processing time per one data packet. The one-hop transmission delay at link (u, v) is given by

$$cost(u, v) = \frac{|(u, v)|}{15} + \frac{\rho_{uv}}{1 - \rho_{uv}} T_s \text{ [msec]} \tag{16}$$

where ρ_{uv} is the average utilization rate of link (u, v), which is given by $(\lambda_{uv} + \lambda_{vu})T_s$, and $|(u, v)|$ corresponds to the Euclidean distance between node u and node v. At the beginning of the simulation, λ_{uv} of links between 6×6 nodes in the center of network is set at $10 + R$ packet/sec, λ_{uv} of links between 4×4 nodes in the center of network is set at $5 + R$ packet/sec, and λ_{uv} of other links is set at $20 + R$ packet/sec (R is a random number in $[-0.5, 0.5]$) as shown in Fig. 4. The average processing time T_s is set at 6.5 msec. In the following figures, we show results averaged over 300 simulation runs for each interval of forward ants except for cases that convergence cannot be achieved by the end of a simulation run.

In this evaluation, we first have the network converge to a state where ants repeatedly select the same path using AntNet. Convergence of the network is defined as the state where the same

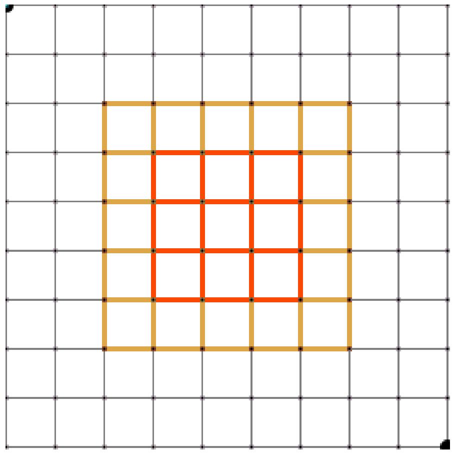


Figure 4: Network in simulation

path is selected for 10 consecutive times. We call the path selected by ants at the first convergence the initial path. When the network converges, we cause node failures or traffic changes. Then, we evaluate the recovery time, path length, and control overhead of AntNet with and without prediction. .

Regarding performance measures, the recovery time is defined as the time from the occurrence of environmental change till convergence. Convergence check is done everytime a backward ant reaches a source node. The path delay corresponds to total delay of a created path from a source node to a destination node. The control overhead is defined as the total number of traveled hops of control messages until convergence.

6.2 Results and Discussion

We first evaluate our predictive mechanism based locally available information and then evaluate our predictive mechanism with neighbor nodes' information.

6.2.1 Prediction based on Locally Available Information

In this section, we evaluate our predictive mechanism based on locally available information under condition of node failures or traffic changes. The parameter β, which determines the weight of one time increment of pheromones in the increase rate of pheromones (Eq. (11)), is set at 0.1, 0.2, and 0.4.

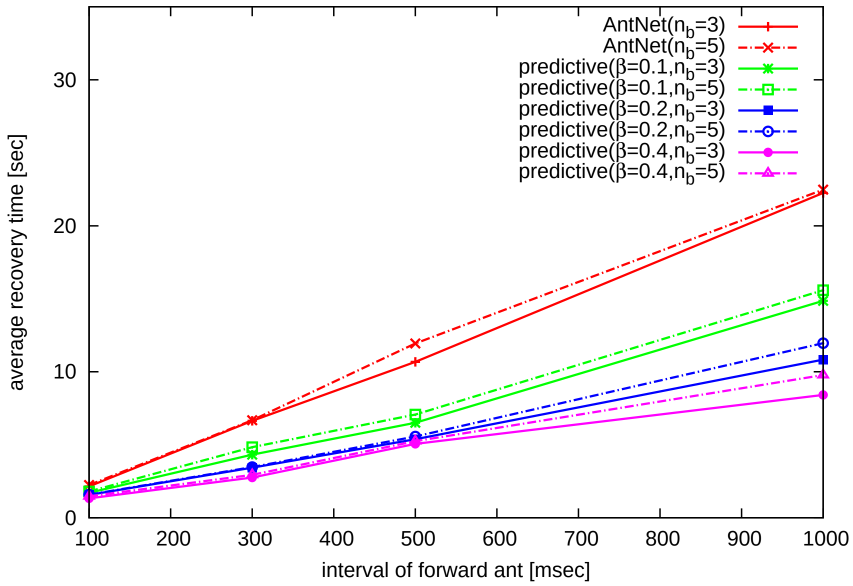


Figure 5: Recovery time after node failures

First, we evaluate recovery from node failures. Once the path converges, n_b nodes randomly fail on the initial path where n_b is 3 or 5.

Figures 5 through 7 shows the recovery time, path delay, and control overhead after node failures. As shown in Fig. 5, our proposal shows a shorter recovery time than AntNet because path establishment is accelerated with a predictive mechanism. In AntNet, nodes on a path have to wait long for a backward ant to come back from a distant destination node in order to increase the amount of pheromone. Therefore, it take long time for pheromone values to converge since a short path is found. On the contrary, in our predictive mechanism, each node predicts the neighbor node which will collect more pheromones by observing pheromone changes and boosts pheromone accumulation on the predicted neighbor node. Therefore, pheromone values can converge faster without waiting for come back of backward ants. As a result, the speed of reinforcement of a path is much faster with our predictive mechanism than AntNet. In addition, as shown in Fig. 5, the recover time is shorter with larger β. With larger β, the weight of one time pheromone update when each node receives a backward ant is larger. Therefore, each node begins to boost pheromone accumulation faster with a few past information of pheromone changes when β is larger. As a result, the path convergence is faster with larger β in our predictive mechanism.

As shown in Fig. 6, our predictive mechanism constructs longer path compared with AntNet.

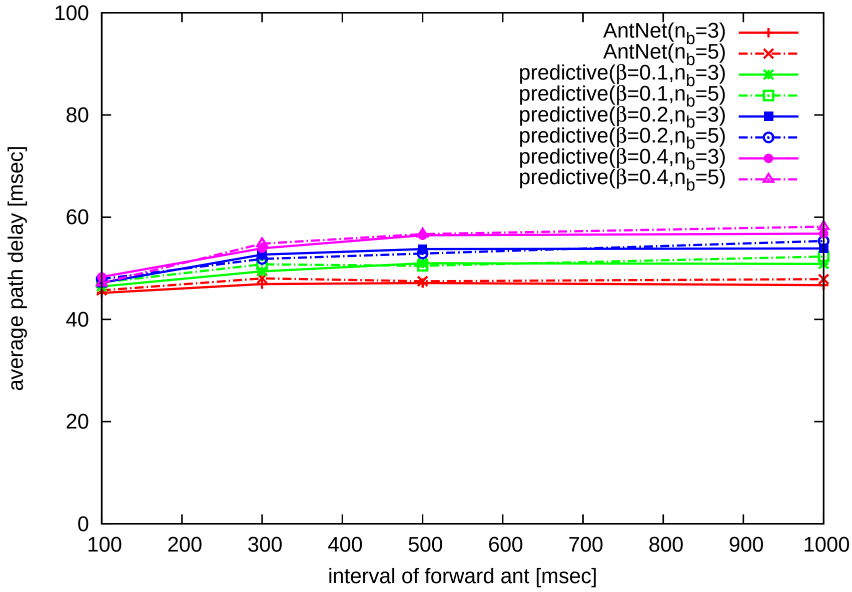


Figure 6: Path delay after node failures

In AntNet, nodes that receive a backward ant leave pheromones for the path that the backward ant travelled even when the path length is not short enough. In our predictive mechanism, each node boosts pheromone accumulation on the path that gets pheromones more frequently than other paths regardless of the goodness of the path. As a result, a longer path sometimes collects more pheromones than other shorter paths in our predictive mechanism. With larger β, each node begins to boost pheromone accumulation faster, but, at the same time, pheromone accumulation on a longer path also likely to be accelerated. As a result, our predictive mechanism constructs a longer path with larger β.

Figure 7 shows that cumulative overhead in our predictive mechanism is lower or almost equal to that of AntNet. However, as shown in Fig. 8, overhead per time in our predictive mechanism is higher compared with AntNet. In AntNet, a loop avoidance mechanism is introduced [29]. A forward ant records its path and avoids making loops. On the contrary, in our predictive mechanism, pheromone accumulation on a path including loops can be boosted because prediction is performed using only its own pheromone table independently of the existence of a loop.

Second, we evaluate recovery from traffic changes. Once the path converges, λ_{uv} of links between 6×6 nodes in the center of network is increased to $40 + R$ packet/sec and λ_{uv} of links between 4×4 nodes in the center of network is increased to $60 + R$ packet/sec.

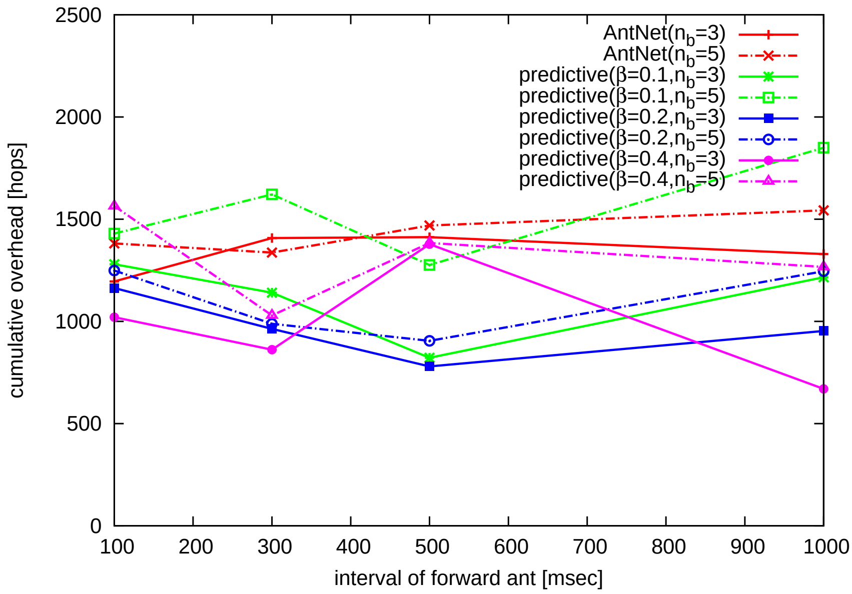


Figure 7: Cumulative overhead after node failures

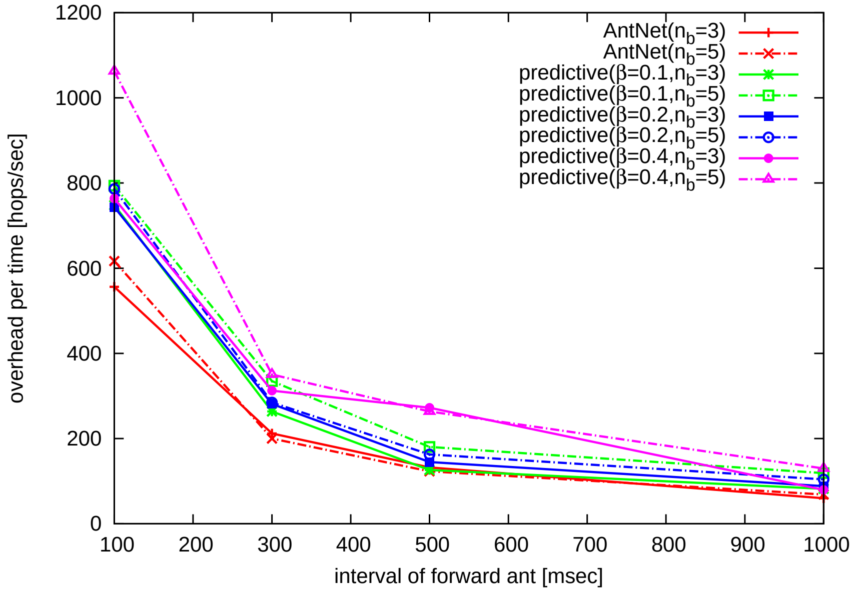


Figure 8: Control overhead per time after node failures

Figures 9 through 11 shows the recovery time, path delay, and control overhead after traffic changes. As shown in Fig. 9, the recovery time of our predictive mechanism is shorter compared with AntNet. It is because path establishment is accelerated with a predictive mechanism like after node failures.

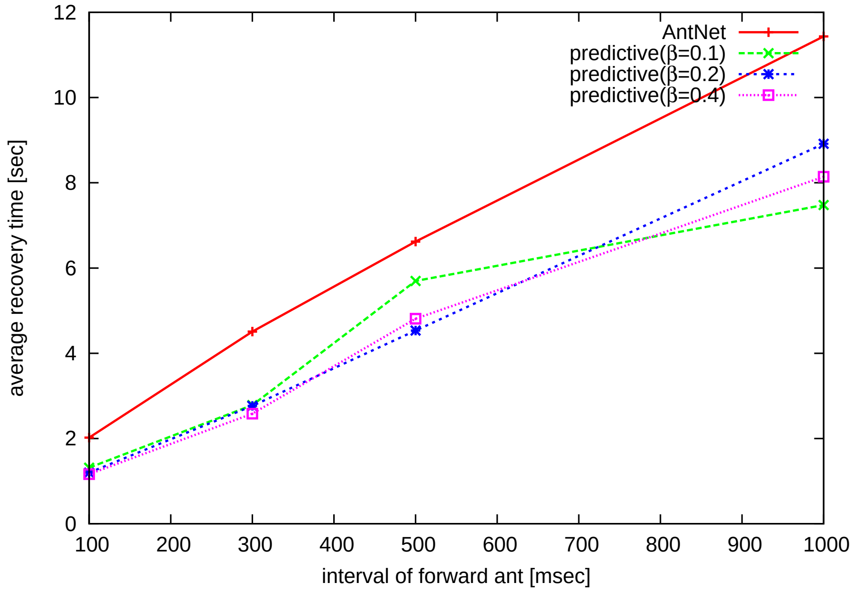


Figure 9: Recover time after traffic changes

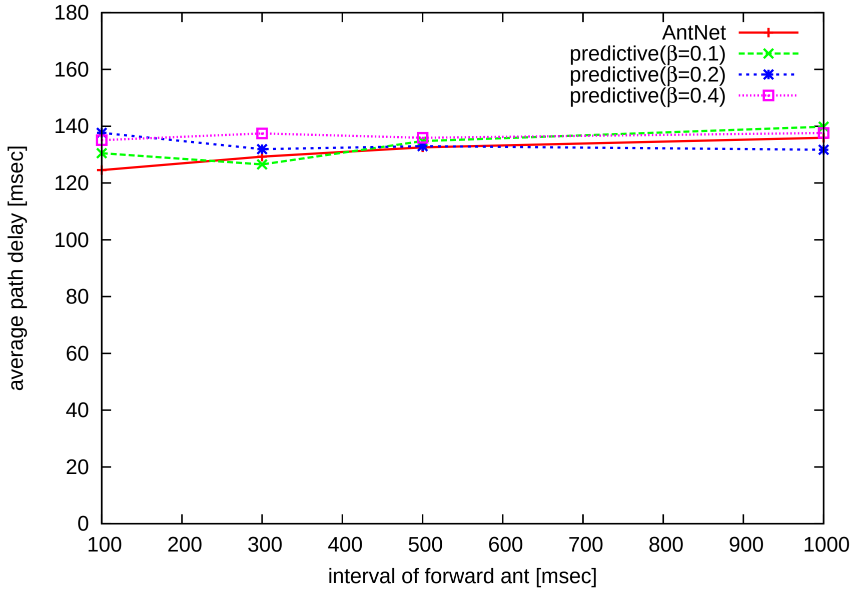


Figure 10: Path delay after traffic changes

As shown in Fig. 10, path delay of our predictive mechanism is almost equal to that of AntNet. Introduction of a predictive mechanism to AntNet has mainly two goal. One is to let paths converge faster and another is to let ants traverse a shorter path faster. The first goal is achieved as shown in Fig. 9. However, the second goal is not achieved as shown in Fig. 10 and this remains

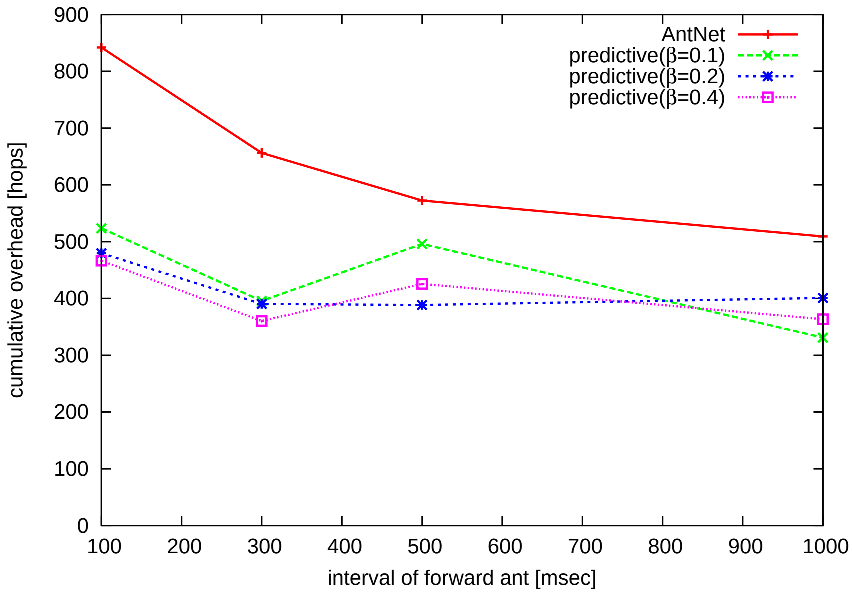


Figure 11: Cumulative overhead after traffic changes

future work.

As shown in Fig. 11, the cumulative overhead of our predictive mechanism is lower compared with AntNet. In our predictive mechanism, the recovery time is shorter compared with AntNet. Therefore, the number of ants needed for paths to converge is reduced. As a result, overhead is reduced compared with AntNet.

In conclusion the path convergence is accelerated with a predictive mechanism. However, it still be a challenging problem to let ants traverse a shorter path faster.

6.2.2 Prediction based on Information of Neighbor Nodes

In this section, we evaluate our predictive mechanism based on information of neighbor nodes. The parameter β, which determines the weight of one time increment of pheromones in the increase rate of pheromones (Eq. (11)), is set at 0.1, 0.2, and 0.4. The parameter π and b in Eq. (14) are set at 0.5 and 0.1, respectively. Once the path converges, λ_{uv} of links between 6×6 nodes in the center of network is increased to $40 + R$ packet/sec and λ_{uv} of links between 4×4 nodes in the center of network is increased to $60 + R$ packet/sec. In this simulation, we appoint a node at the first row and forth column and one at the forth row and first column as source nodes in addition to one at the top-left corner. Destinations of three sessions are the same, i.e. one at the bottom-right

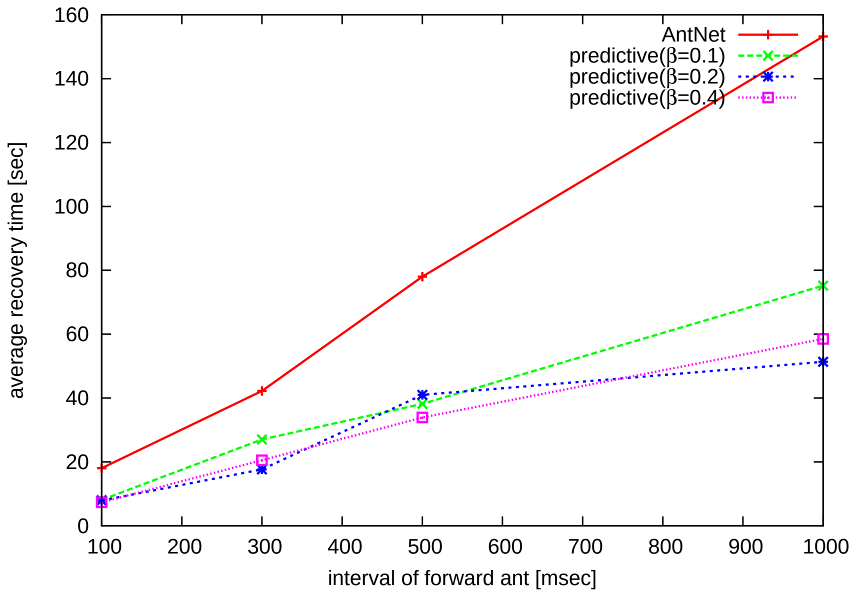


Figure 12: Recovery time with neighbor nodes' information

corner.

Figures 12 through 14 shows the recovery time, path delay, and control overhead after traffic changes. As shown in Fig. 12, the recovery time of our predictive mechanism is shorter than that of AntNet. In our predictive mechanism, each node boosts pheromone accumulation on the neighbor node where pheromones are near to convergence. As a result, the path convergence is accelerated.

As shown in Fig. 13, our predictive mechanism constructs a shorter path than AntNet. In our predictive mechanism, each node boosts pheromone accumulation on the neighbor node where pheromones are near to convergence in our predictive mechanism. Therefore, when the shorter path is being established at a node, its neighbor node can utilize the shorter path. As a result, the path delay of our predictive mechanism is shorter than that of AntNet. However, it is sensitive to condition of network such as locations of source nodes and destination nodes.

As shown in Fig. 14, control overhead of our predictive mechanism is much larger than that of AntNet. It is because each node regularly sends predictive ants to all its neighbor nodes in order to obtain neighbor nodes' information in our predictive mechanism. However, overhead of forward and backward ants is reduced because the recovery time is shortened with prediction. Therefore, overhead of predictive ants becomes trivial as the number of sessions becomes larger than about 30.

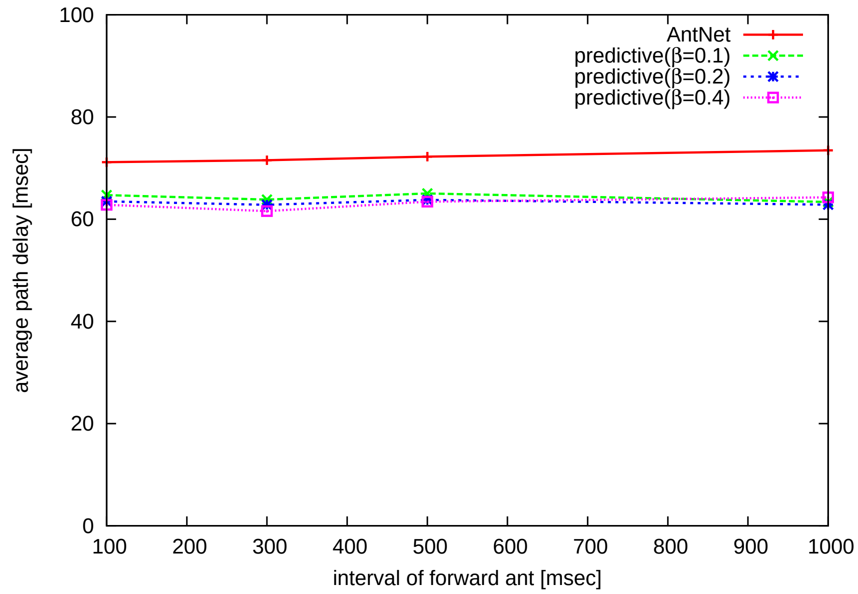


Figure 13: Path delay with neighbor nodes' information

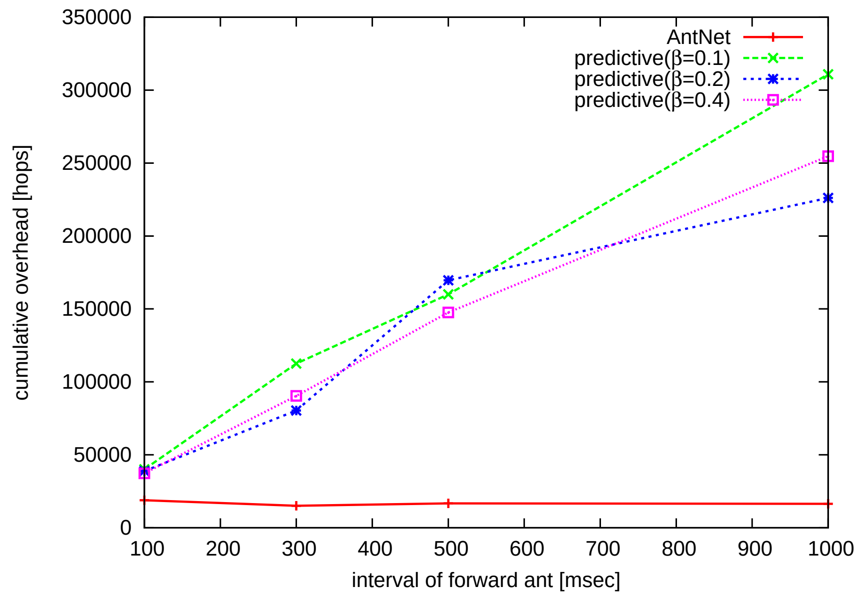


Figure 14: Cumulative overhead with neighbor nodes' information

In conclusion, the path convergence is accelerated and the path delay is shortened with prediction using information of neighbor nodes in this simulation settings. However, we need more discussion because the simulation setting is mere one case of network conditions.

7 Conclusion and Future Work

In a self-organizing system, each component behaves in accordance with only local current information. In this thesis, as an example of a predictive mechanism for controlled self-organization system, we propose and evaluate a predictive mechanism for AntNet. Simulation results show that our predictive mechanism can facilitate path reestablishment when the environment of the network changes.

As future work, we consider the improvement of prediction precision, i.e. the acceleration of path convergence on a shorter path after environmental changes. In addition, we will evaluate our predictive mechanism in more real network environment, such as random topology.

Acknowledgement

The thesis would not have been possible without lot of grate supports of several people. First, I would like to express my sincere gratitude to my supervisor, Professor Masayuki Murata of Osaka University, for his invaluable advice, continuous support and encouragement. I sincerely appreciate Professor Naoki Wakamiya of Osaka University. He always guides my research and gives me valuable advice.

I would like to express my deepest appreciation to Professors Koso Murakami, Hirotaka Nakano and Teruo Higashino of Osaka University for their appropriate guidance. I am also deeply grateful to Associate Professor Shin'ichi Arakawa and Assistant Professor Yuichi Ohsita of Osaka University, who gave me helpful comments and feedback. I would like to express my gratitude to Mr. Hiroshi Yamamoto, Mr. Daichi Kominami, and Mr.Takuya Iwai, who gave me a lot of advice.

Finally, I truly thank my friends and colleagues in the Department of Information Networking, Graduate School of Information Science and Technology of Osaka University for their support.

References

[1] National Science Foundation, http://www.nets-find.net/, *Future internet design (FIND)*.

[2] National Science Foundation, http://www.geni.net, *Global environment for network innovations (GENI)*.

[3] European Commission, http://eurongi.enst.fr, *Network of excellence EuroNGI and EuroFGI design and engineering of the next generation internet*.

[4] European Commission, http://cordis.europa.eu/fp7/ict/fire/, *Future Internet Research & Experimentation*.

[5] AKARI project, *New generation network architecture AKARI conceptual design*, October 2007.

[6] F. Dressler, *Self-organization in sensor and actor networks*. Wiley, January 2008.

[7] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, USA, October 1999.

[8] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adaptive behavior*, vol. 5, pp. 169–207, January 1997.

[9] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, December 1998.

[10] G. Di Caro, F. Ducatelle, and L. Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications*, vol. 16, pp. 443–455, September 2005.

[11] S. Dhillon and P. Van Mieghem, "Performance analysis of the AntNet algorithm," *Computer Networks*, vol. 51, pp. 2104–2125, June 2007.

[12] L. Carvelli and G. Sebastiani, *Some Issues of ACO Algorithm Convergence*, ch. 4. InTech, February 2011.

[13] C. Müller-Schloer, H. Schmeck, and T. Ungerer, *Organic Computing-A Paradigm Shift for Complex Systems*. Birkhauser Verlag AG, May 2011.

[14] M. Prokopenko, “Guided self-organization,” *HFSP Journal*, vol. 3, pp. 287–289, October 2009.

[15] S. Arakawa, Y. Minami, Y. Koizumi, T. Miyamura, K. Shiomoto, and M. Murata, “A managed self-organization of virtual network topology controls in WDM-based optical networks,” *Journal of Optical Communications*, vol. 32, pp. 233–242, December 2011.

[16] D. Kominami, M. Sugano, M. Murata, and T. Hatauchi, “Controlled potential-based routing for large-scale wireless sensor networks,” in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems 2011*, pp. 187–196, December 2011.

[17] H. Zhang, M. Chen, G. Stan, T. Zhou, and J. Maciejowski, “Collective behavior coordination with predictive mechanisms,” *IEEE Circuits and Systems Magazine*, vol. 8, pp. 67–85, August 2008.

[18] P. Montague, P. Dayan, C. Person, T. Sejnowski, *et al.*, “Bee foraging in uncertain environments using predictive hebbian learning,” *Nature*, vol. 377, pp. 725–728, October 1995.

[19] C. Summerfield, T. Egner, M. Greene, E. Koechlin, J. Mangels, and J. Hirsch, “Predictive codes for forthcoming perception in the frontal cortex,” *Science*, vol. 314, pp. 1311–1314, November 2006.

[20] U. Richter, M. Mnif, J. Branke, C. Müller-Schloer, and H. Schmeck, “Towards a generic observer/controller architecture for organic computing,” *Informatik*, pp. 112–119, October 2006.

[21] F. Dressler and O. Akan, “A survey on bio-inspired networking,” *Computer Networks*, vol. 54, pp. 881–900, April 2010.

[22] M. Meisel, V. Pappas, and L. Zhang, “A taxonomy of biologically inspired research in computer networking,” *Computer Networks*, vol. 54, pp. 901–916, April 2010.

[23] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant System: optimization by a colony of cooperating agents,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, pp. 29–41, February 1996.

[24] T. Stützle and H. Hoos, "MAX–MIN ant system," *Future generation computer systems*, vol. 16, pp. 889–914, June 2000.

[25] K. Sim and W. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, pp. 560–572, September 2003.

[26] M. Saleem, G. Di Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions," *Information Sciences*, vol. 181, pp. 4597–4624, October 2011.

[27] Y. Ohtaki, N. Wakamiya, M. Murata, and M. Imase, "Scalable and efficient ant-based routing algorithm for ad-hoc networks," *IEICE Transactions on Communications*, vol. 89, no. 4, pp. 1231–1238, 2006.

[28] J. Wang, E. Osagie, P. Thulasiraman, and R. Thulasiram, "HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network," *Ad Hoc Networks*, vol. 7, pp. 690–705, June 2009.

[29] D. Caro, *Ant Colony Optimization and its application to adaptive routing in telecommunication networks*. PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, September 2004.