A Traffic-Based Routing Algorithm by Using Mobile Agents

Wenyu Qu [1], Masaru Kitsuregawa [1], Hai Zhuge [2], Hong Shen [3], and Yinwei Jin [4]

[1] Institute of Industrial Science
The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan

[2] Institute of Computing Technology
Chinese Academy of Sciences
PO Box 2704-28, Beijing, 100080, China

[3] School of Computer Science
University of Adelaide
SA 5005, Australia

[4] Department of Computer Science and Engineering
Dalian University of Technology
No 2, Linggong Road, Dalian, 116024, China

Abstract

In this paper, we propose a traffic-based routing algorithm by using mobile agents. The traffic cost function for each link is defined based on known routing information. We theoretically analyze the probability distribution which is very useful for mobile agents to select a neighboring node to move to. The optimal probability distribution makes inference on the known information and approximates to a unbiased distribution. Simulation experiments are conducted to compare the performance of our algorithm with existing algorithms. The results show that our algorithm outperforms existing algorithms on the performance metrics considered.

Key words: Mobile agents, routing, cost function, unbiased distribution.

1 Introduction

The amazing advances in the computer industry have created many new application areas for network routing such as grid computing [35] and Internet computing [19]. People communicate with the outer world through wireless networks, LANs, and the Internet. The widespread popularity of the World Wide Web demands new paradigms for building computer systems [18]. The deployment of mobile agents, which are small decision-making programs capable of migrating autonomously from node to node in a computer network, is an important representative of these new paradigms and is an effective way to reduce network load and latency [17].

In [7], Milojicic described that mobile agents are autonomous, adaptive, reactive, mobile, cooperative, interactive, and delegated software entities. The key idea underlying mobile agents is to bring the computation to the data rather than the data to the computation [26]. The application of mobile agents in network routing has attracted significant attention [6, 23, 34]. Successful examples of such applications can be found in [15, 16]. The use of mobile agents in applications ranging from electronic commerce to distributed computation has also been studied extensively.

Routing is a key factor for network performance. It is the process of moving a packet of data from source to destination. Once request for sending a packet is received, the router should recommend the optimal path (or the shortest path) for sending this packet over the network. As searching for the optimal path in a stationary network is already a difficult problem, the searching for the optimal path in a dynamical network or mobile network will be much more difficult. Mobile agent-based routing algorithm is a promising option for use in these environments [5]. In a mobile agent-based routing algorithm, a group of mobile agents build paths between pair of nodes, exploring the network concurrently and exchanging data to update routing tables [9,12]. Once a request for sending a packet is received from a server, a number of mobile agents are generated and dispatched to the network. These agents roam around the network and gather relevant information. Once an agent accomplishes its task, the collected information is sent back to the server. When a certain number of those agents have come back, the server selects the optimal path by certain criterion and sends the packet to the destination along the new path. At the same time, the server updates its routing table by the information of the new path.

It can be seen that in a large communication network such as Internet, agents have to be generated frequently and dispatched to the network. Thus, they will certainly consume a certain amount of bandwidth of each link in the network. If there are too many agents migrating through one or several links at the same time, they will introduce too much transferring overhead to the links. Eventually, these links will be busy and indirectly block the network traffic. Therefore, there is a need of developing routing algorithms that consider about the traffic load. Since the state of different links may change dynamically over time, the agents have to dynamically adapt themselves to the environment, which increases the difficulty for both algorithm design and theoretical analysis.

In this paper, we propose a traffic-based routing algorithm by using mobile agents in which the traffic cost for each link is considered. The idea of our model is useful not only for network routing, but also for other applications such as data transferring, task scheduling, and storage arrangement, etc. Theoretical analysis shows that our derived probability distribution that an agent on an intermediate node may select a neighboring node and move to not only refers to the known traffic information but also mostly balances the traffic load to the neighboring links. We also conduct simulation experiments to compare the performance of our algorithm with existing algorithms. The results show that our algorithm outperforms existing algorithms on the performance metrics considered.

The remainder of this paper is structured as follows. Section 2 describes related work. Section 3 presents our algorithm. Section 4 proposes the mathematical modeling of the decision making process of mobile agents. Section 5 deduces the optimal probability distribution for agents' decision making. Section 6 describes simulation and performance evaluation, and Section 7 concludes this paper.

2 Related Work

Real ants are capable of finding the shortest path from a food source to the nest based only on local information [2,9,11,13]. Inspired from the research on ants, Caro and Dorigo [4,5] firstly proposed mobile agent-based routing algorithm in 1998. In a mobile agent-based routing algorithm, a group of mobile agents build paths between pair of nodes, exploring the network concurrently and exchanging data to update routing tables. In [4,5], the authors showed that this mobile agent-based routing algorithm is very encouraging with many experiments over real and artificial IP datagram networks by comparing it with both static and adaptive state-of the-art routing algorithms, such as RIP and OSPF. A number of experiments have been conducted to compare the performance between the mobile agent-based routing algorithm and other routing algorithms. The results showed both favorable performances and robustness to classic routing algorithms like RIP and OSPF.

In [33], an adaptive agent routing system was described in which a genetic algorithm is integrated with the agent system. In [1], the authors proposed a mechanism of handling routing table entries at the neighbors of crashed routers, which significantly improved the algorithm in [4,5]. In [20], the authors showed that it is difficult to fulfill the requirement in the previously proposed algorithms in real networks, i.e., providing to each destination an entry in the routing table. To overcome this shortcoming, they proposed in [21] a new algorithm that eliminates this need.

In [3], Brewington et al. formulated a method of mobile agent planning, which is analogous to the travelling salesman problem [10] to decide the sequence of nodes to be visited by minimizing the total execution time until the desired information is found. In [24,25], the routing model proposed in [33] was studied and both the number of mobile agents and the probability of success were estimated. In [31], a macroscopic model is proposed for grid computing that described the dynamics of agent-based load balancing with time delays.

In this paper, we propose a traffic-based routing algorithm by using mobile agents and consider about the traffic load in the network. To balance the traffic load balance on each link, we introduce the maximum entropy theory into our algorithm to find an optimal probability distribution that makes inference on the known traffic information and balances the traffic load.

3 The Mobile Agent-Based Routing Algorithm

Let $G = \{V, E\}$ be a graph corresponding to a fixed agent-driven network, where $V = \{\nu_1, \nu_2, \cdots\}$ is the set of vertices (hosts) and $E = \{e_{ij}\}$ is the set of edges. In this paper, we assume that the topology of a network is a connected graph in order to ensure that communication are able to be made between any two host machines. $NB(i)$ is the set of neighboring nodes of node ν_i and $|NB(i)|$ is the number of nodes in $NB(i)$. Suppose that agents can be generated from every node in the network, and each node in the network provides to mobile agents an execution environment. A node from which mobile agents are generated is called the server of these agents. The traffic based routing algorithm by using mobile agents is described as follows:

- **Agent Generation.** At any time, requests may be keyed in the network. Once a request for sending a packet to a destination (point-to-point) or to multiple destinations (multicast) is received from a server (in our model, each node in the network can be seen as a server), the server will generate a number of mobile agents. Each agent carries the addresses of its server, its destination, the previous node it jumped from, and some control information for routing such as life-span limitation and hop counter. All these data can be contained in several lines of Java code; thus, the size of a mobile agent is very small, resulting in great reduction on network load and latency.

- **Destination Detection.** After being generated, these agents move out from the server and roam in the network. When an agent reaches a node, it checks whether the host node is its destination. If the host node is not the destination, the agent selects a neighboring node according to the costs of the neighboring links and move to. A cost function is defined on the traffic situation of links such as traffic load, transmission speed, bandwidth, etc. A link with lower cost will be selected with priority.

- **Path Selection.** Once an agent has reached the destination, it will go back to the server along the path searched, update the routing tables on the nodes along the path, and submit its report about the searched path to the server. When a certain number of those agents have come back, the server selects the optimal path by certain criterion and sends the packet to the destination along the new path. At the same time, the server updates its routing table by the information of the new path.

- **Life-Span Limitation.** To eliminate unnecessary searching in the network, a life-span limitation is assigned to each agent. An agent will die if it cannot find its destination in its life-span limitation. Moreover, if an agent cannot return to its server in two times the life-span limitation (e.g., its return route is interrupted due to a link/node failure), the agent also will die.

4 Load Balance Modeling

During agents' search process for its destination, if an agent reaches a node ν_i which is not its destination, it will select a node from the set of neighboring nodes of ν_i, i.e., $NB(i)$, according to the traffic costs of the neighboring links and move to. Originally, each node has no information about its neighboring nodes and links, and no agent passes it on the return trip. Therefore, all links around ν_i have the same values of traffic costs and each node in $NB(i)$ has the same probability, i.e., $1/|NB(i)|$, to be selected. With time going, the value of traffic cost will change time by time. To measure the traffic cost of of an agent migrating from ν_i to ν_j at time t, a cost function, $f_{ji}(t)$, is defined on the traffic situation of the link e_{ij} such as traffic load, transmission speed, bandwidth, etc., where $\nu_j \in NB(i)$. A link with lower cost will be selected with priority. Without lose of generalization, we assume that function $f_{ji}(x)$ is differentiable for $i \in NB(j)$. When there are multiple links with lowest cost, the link will be selected which can balance the traffic load most in the network. That is, the link selection on each node should satisfy two constraints:

1. It makes inference on all the known information.
2. It is unbiased. That is, the selection should mostly balance the traffic cost on each link.

In this paper, the link selection problem is modeled as a probability distributed problem to each immediate link of the host node of an agent. Each link has a probability to be selected based on the traffic situation of this link. Mobile agents on the host node will select one immediate link based on this probability distribution. In this way, a link with a high cost still has a possibility to be selected, which can greatly reduce the possibility of traffic jam due comparing with that only the link with lowest traffic cost will be selected.

For the purposes that not only making inference on all the known information but also balancing the traffic load on each link, one feasible way is to minimize the highest cost among all immediate links of the host node. Thus, the problem can be modeled as a min-max problem as follows:

$$\min_{x\in R^n} f^{(j)}_{\max}(x), \tag{1}$$

where x is a random variable with n entries which denotes n items to be considered to the cost of a link. The objective function $f^{(j)}_{\max}(x)$ is the maximum traffic cost defined as follows:

$$f^{(j)}_{\max}(x) \equiv \max_{i\in NB(j)} \{f_{ji}(x)\}. \tag{2}$$

Here, $f_{ji}(x)$ is the traffic cost function from node ν_j to ν_i. Without losing generalization, $f_{ji}(x)$ are assumed differentiable for all $i \in NB(j)$. Obviously, function $f^{(j)}_{\max}(x)$ is an undifferentiable function. To solve the min-max problem (1), it is desirable to find a differentiable function that approximates to function $f^{(j)}_{\max}(x)$. Let's look at the following Lagrange function:

$$\ell_j(x, p_j) = \sum_{i\in NB(j)} p_{ji} f_{ji}(x) \qquad \forall x \in R^n, p_j \in \Delta_j, \tag{3}$$

where $p_j = (p_{j1}, p_{j2}, \cdots, p_{j,|NB(j)|})^T$ is the vector of Lagrange multiplier, Δ_j is a simplex set defined as follows:

$$\Delta_j \equiv \left\{ p_j \in R^{|NB(j)|} \,\middle|\, \sum_{i\in NB(j)} p_{ji} = 1, p_{ji} \geq 0 \right\}. \tag{4}$$

It is easy to see that no matter which value the multiplier vector p_j is chosen, the value of the Lagrange function $\ell_j(x, p_j)$ is less than or equal to the maximum value function $f^{(j)}_{\max}(x)$, i.e.,

$$\ell_j(x, p_j) \leq f^{(j)}_{\max}(x). \tag{5}$$

From the definition of Lagrange function $\ell_j(x, p_j)$, we have the following lemma:

Lemma 1 *The maximum value function $f^{(j)}_{\max}(x)$, defined in (2), satisfies the following relationship:*

$$f^{(j)}_{\max}(x) = \sup_{p_j\in\Delta_j} \ell_j(x, p_j) = \max_{p_j\in\Delta_j} \ell_j(x, p_j). \tag{6}$$

Proof See Appendix I. □

Lemma 1 shows that since the multiplier vector p_j is limited inside the simplex Δ_j, the Lagrange function $\ell_j(x, p_j)$ can be interpreted as a convex combination of traffic cost functions $f_{ji}(x)$ $(i \in NB(j))$ and multipliers p_{ji} can be interpreted as the combination coefficients. Therefore, (1) can be transferred into an equivalent problem of finding a set of value of $p_{ji}(i \in NB(j))$ such that the Lagrange function $\ell_j(x, p_j)$ approximates to the maximum value function $f^{(}_{max}j)(x)$, i.e., to find the optimal combination $\hat{p}_j$ from all combinations that satisfies (4) such that the inequality (5) can be transferred into the following equality:

$$\ell_j(x, \hat{p}_j) = \sum_{i \in NB(j)} \hat{p}_{ji} f_{ji}(x) = f^{(j)}_{\max}(x). \tag{7}$$

Endue a probability sense to the Lagrange multipliers p_{ji} $(i \in NB(j))$, also called the combination coefficients, i.e., describing p_{ji} as the corresponding probabilities such that the element function $f_{ji}(x)$ becomes the maximum value function $f^{(j)}_{\max}(x)$, then from the concept of probability, Problem (1) can be transferred into a maximized problem of finding the optimal probability distribution that satisfies:

$$\begin{cases} \max\limits_{p_j \in R^{|NB(j)|}} & \ell_j(x, p_j) \\ s.t. & \sum\limits_{i \in NB(j)} p_{ji} = 1; \\ & p_{ji} \geq 0, \quad i \in NB(j). \end{cases} \tag{8}$$

On the other hand, regarding to the unbiased requirement, let's consider about the following cases. If there is only one link to be selected, then the probability that this link will be selected is 100%. If there are two links to be selected without any information, then the probability that each link will be unbiased selected should be 50%. In this case, which link will be selected? Suppose that for a probability experiment, there are only two possible results (a_1, a_2) with probability distribution (p_1, p_2). For different probability distribution of these two possible results, the uncertainty of the possible outcomes are different. For example, when two probability distributions are assigned as $(p_1 = 0.5, p_2 = 0.5)$ and $(p_1 = 0.99, p_2 = 0.01)$, respectively, it is obvious that the experimental result of the first distribution has a higher uncertainty then that of the second. The reason is that for the first case, we cannot predict which result will be achieved in the experiment, while for the second case, we can "almost affirm" that the experimental result would be a_1. It is consistent with our intuition that uniform probability distribution is much uncertain then a non-uniform probability distribution. Thus, for the unbiased requirement, it is desirable that the resulted probability distribution not only satisfies all constraints of the probability experiments, but also has the maximum uncertainty, i.e., the maximum fairness to each possible outcome. In our model, it means the load balance on each link in the network. To find an unbiased probability distribution that can balance the network traffic load, i.e., to achieve the maximum uncertainty, we introduce the maximum entropy theory (see Appendix II) in our model. Based on the maximum entropy theory, the optimal probability distribution p_{ji} in (8) should also satisfy:

$$\begin{cases} \max\limits_{p_j \in R^{|NB(j)|}} & H(p_j) = -\sum\limits_{i \in NB(j)} p_{ji} \ln p_{ji} \\ s.t. & \sum\limits_{i \in NB(j)} p_{ji} = 1; \\ & p_{ji} \geq 0, \quad i \in NB(j). \end{cases} \tag{9}$$

Therefore, combine the two considerations above, the problem to be solved, i.e., (1), can be modeled as a multi-objective problem as follows:

$$\begin{cases} \max\limits_{p_j \in R^{|NB(j)|}} & \{\ell_j(x, p_j), H(p_j)\} \\ s.t. & \sum\limits_{i \in NB(j)} p_{ji} = 1; \\ & p_{ji} \geq 0, \quad i \in NB(j). \end{cases} \tag{10}$$

5 Probability Distribution

In this section, we will find the optimal probability distribution that not only makes inference on all known traffic information but also balances the traffic load in the network based on the analysis in the previous section.

By the weighting coefficient method, the multi-object problem (10) can be transferred into a single-object problem as follows:

$$\left\{\begin{array}{ll}\max\limits_{p_j} & L_\theta^{(j)}(x,p_j)=\sum\limits_{i\in NB(j)} p_{ji}f_{ji}(x) \\ & \qquad\qquad -\dfrac{1}{\theta}\sum\limits_{i\in NB(j)} p_{ji}\ln p_{ji}; \\ s.t. & \sum\limits_{i\in NB(j)} p_{ji}=1; \\ & p_{ji}\geq 0,\ \ i\in NB(j),\end{array}\right. \tag{11}$$

where $\theta \geq 0$ is a weighting coefficient. Obviously, when θ is small, the second item of the object function $L_\theta^{(j)}(x,p_j)$ is dominative. Then the gained probability distribution mainly reflects the requirement of unbiased distribution. With the increase of θ's value, the effect of the first item increases; thus, the object of maximizing the Lagrange function becomes dominative.

To solve Problem (11), we first consider the following problem:

$$\sup_{p_j\in\Delta_j}\left\{L_\theta^{(j)}(x,p_j):=\ell_j(x,p_j)-\theta^{-1}\sum_{i\in NB(j)} p_{ji}\ln p_{ji}\right\}. \tag{12}$$

Based on the knowledge of convex analysis and the property of entropy function, we can prove that the function defined by (12) has the following property:

Theorem 1 *The function $F_\theta^{(j)}(x)$ defined by (12) is differentiable and uniformly approximate to function $f_{\max}^{(j)}(x)$ on the whole space R^n.*

Proof See Appendix III. □

Theorem 1 shows that function $F_\theta^{(j)}(x)$ uniformly converges to the objective function $f_{\max}^{(j)}(x)$ defined in (2). Thus, solving the undifferentiable problem (1) is equivalent to solving the following differentiable problem:

$$\min_{x\in R^n} F_\theta^{(j)}(x). \tag{13}$$

To facilitate the solving of problem (13), we give several properties of function $F_\theta^{(j)}(x)$ as follows:

Theorem 2 *For $\forall x\in R^n$, function $F_\theta^{(j)}(x)$ has the following properties:*

1. $f_{\max}^{(j)}(x)\leq F_\theta^{(j)}(x)\leq f_{\max}^{(j)}(x)+(\ln m)/\theta$.
2. $\lim_{\theta\to\infty} F_\theta^{(j)}(x)=f_{max}^{(j)}(x)$.
3. *If all the functions f_{ji} $(i=1,2,\cdots,m)$ in the original problem (1) are convex, $F_\theta^{(j)}(x)$ is a convex function too.*
4. $\nabla_x F_\theta^{(j)}(x)=\sum_{i\in NB(j)}\hat{p}_{ji}(x)\nabla_x f_{ji}(x)$.
5. $-(\ln m)/\theta^2\leq \partial F_\theta^{(j)}(x)/\partial\theta\leq 0$.
6. $F_\theta^{(j)}(x)\leq F_\vartheta^{(j)}(x)$, $\forall\theta\leq\vartheta$.

Proof See Appendix IV. □

In Theorem 2, property 1 provides error bounds of function $F_\theta^{(j)}(x)$, property 2 shows that function $F_\theta^{(j)}(x)$ uniformly approximates to function $f_{\max}^{(j)}(x)$, property 3 shows the convex property of function $F_\theta^{(j)}(x)$, property 4 is for the continuity and the differentiability of function $F_\theta^{(j)}(x)$, property 5 provides both upper bound and lower bound of the derivation of function $F_\theta^{(j)}(x)$ on p, and property 6 shows that function $F_\theta^{(j)}(x)$ is a monotonously decrease function on θ.

Based on the properties in Theorem 2, it can be seen that function $F_\theta^{(j)}(x)$ is differentiable. Thus, the optimal solution, $\hat{p}_j(x)$, of Eq. (11) can be easily derived from applying the $K-T$ condition as follows:

$$\hat{p}_{ji}(x) = \frac{\exp\{\theta f_{ji}(x)\}}{\sum_{l\in NB(j)} \exp\{\theta f_{jl}(x)\}}, \quad i \in NB(j). \tag{14}$$

where p_{ji} is the probability that an agent on node ν_j selects node ν_i to migrate to, $\theta \geq 0$ is a weight coefficient defined according to the effect of the known traffic state of the network. Mobile agents in the network will make their selection based on this probability distribution.

Substitute $\hat{p}_j(x)$ in the objective function of (11), we have

$$\begin{aligned} F_\theta^{(j)}(x) &= L_\theta^{(j)}\left(x, \hat{p}_j(x)\right) \\ &= \frac{1}{\theta} \ln \left\{ \sum_{i\in NB(j)} \exp\left[\theta f_{ji}(x)\right] \right\}. \end{aligned} \tag{15}$$

which indicates the maximum cost of immediate links of node ν_j. Thus, the optimal probability distribution of immediate links that an agent may select to migrate to and the maximum traffic cost among those immediate links have been solved.

6 Simulation and Performance Evaluation

We implemented our algorithm (OA) together with OSPF [8] and BF [27] to evaluate and compare the efficiency of these three algorithms. The simulation experiments are performed on OMNet++ simulator [30] for comparing the results of our model with those of the existing models. The network in the simulation consists of numerous nodes. BF is a routing algorithm based on Bellman-Ford algorithm in which link costs are computed as weighted averages between short and long term real valued statistics reflecting the delay over fixed time intervals. It is used in RIP supplied with Unix. It provides reasonable performance on small- to medium-sized networks, but on larger networks the algorithm is slow at calculating updates to the network topology. OSPF is a routing protocol used in Internet. It is based on the shortest path routing by using the Dijkstra algorithm. It uses only the best path towards the destination to forward the data packets and this best path is computed with the costs of edges, normally the distances of links. As OSPF ignores the queuing delays, it cannot work well during heavy traffic loads.

We use the Japanese Internet Backbone (NTTNET) as the simulation framework (see Figure 1). It is a 57 node, 162 bidirectional links network. The link bandwidth is 6 Mbits/sec and the propagation delay is from 2 to 5 milliseconds. The size of data packet is 512 bytes and the size of an agent is 48 bytes. Traffic is defined in terms of open sessions. A session is defined between two nodes and it remains active until a certain amount of data are transferred at a given rate. Each session is characterized completely by session size, the generation rate of sessions (GRS), and the sending rate of the agents (SRA). In the simulation, session size is set to be 2 Mbits.

Two parameters are used in our comparison, throughput and search time. The throughput is defined as the number of search agents forwarded per second, which shows the ability of the algorithms to deliver search agents. The search time is defined as the time interval from the creation of a search agent to its arrival at the destination, which is only calculated for the search agents that arrived at the destination. It indicates the quality of paths chosen by the agents. These two parameters are normal metrics for evaluating the performance of routing

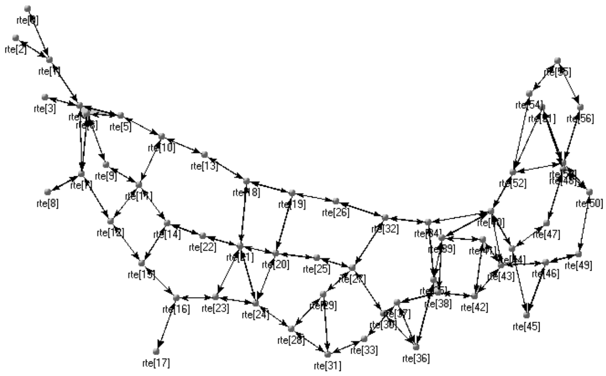

Figure 1: The Japanese Backbone (NTTNET)

algorithms [4, 32]. Originally, there is a normal load of GRS=2.7 seconds and SRA=0.3 second. The node 4 is used for a hot spot. From 500 seconds to 1000 seconds, all nodes sent agent to node 4 with SRA=0.05 seconds. Figure 2 and Figure 3 show the comparison results of the average throughput and the average search time between OSPF, BF, and OA.

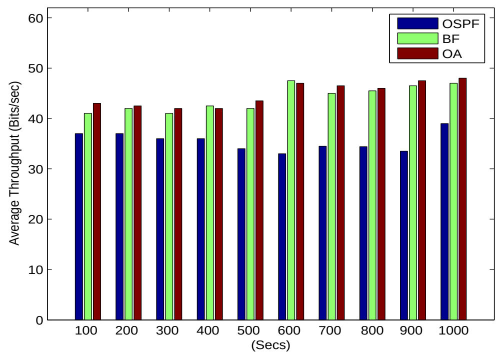

Figure 2: The average throughput when all nodes sent agent to node 4 from 500 seconds to 1000 seconds

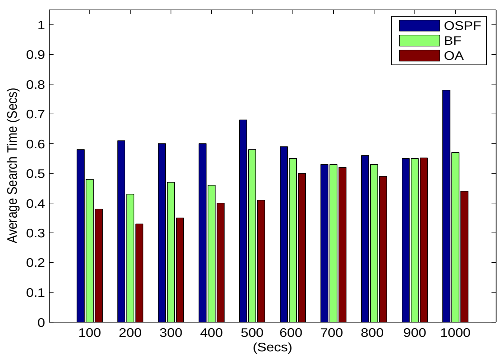

Figure 3: The average search time when all nodes sent agent to node 4 from 500 seconds to 1000 seconds

It can be seen that both OA and BF are able to cope with the transient overload. OSPF shows the poorest performance. It can also be seen that the average search time for OA is less than BF. Again, OSPF shows the poorest performance.

We also compared the success rate among these three algorithms which states the proportion of agents arrived at destinations correctly. Table 1 shows the results of the success rate which is defined as the quotient of the number of agents arrived at the destinations to the number of dispatched agents. From the simulation results,

Table 1: The Comparison of success rate

Parameters		Success Rate (%)		
GRS(Sec)	SRA(Sec)	OSPF	BF	OA
4.5	0.5	83.21	95.72	96.83
2.5	0.5	82.46	94.61	97.03
1.5	0.5	80.13	93.32	96.99
2.5	0.05	83.94	94.83	96.11

we can see that OA achieves a similar performance to BF, which is much better than OSPF.

Figure 4 and Figure 5 show the results of the comparison between OA and BF in which node 21 crashed at 300 seconds, node 40 crashed at 500 seconds, and both of them were repaired at 800 seconds. Here, GRS=4.7 seconds and SRA=0.05. The purpose of this experiment was to analyze the fault tolerant behavior of OA. The GRS and SRA is selected to ensure that no agents are terminated because of the congestion. Based on our experimental results, OA is able to deliver 97% of agents as compared to 95% by BF. From the figure we can see that OA has a superior throughput and lesser search time. But once node 40 crashes, the search time of OA increases because of higher load at node 43. From Figure 1 it is obvious that the only path to the upper part of the network is via node 43 once node 40 crashed. Since OA is able to deliver more agents the queue length at node 43 increased and this led to relatively longer search time. On the contrary, although node 21 is critical, but in case of its crash still multiple paths exist to the middle and upper part of the topology.

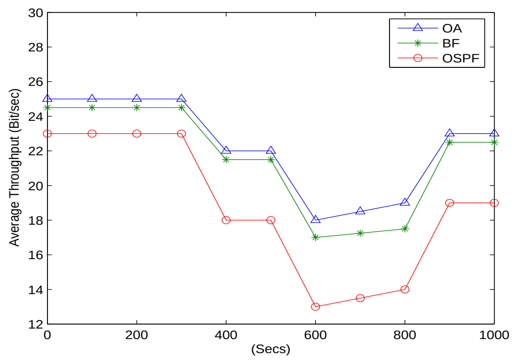

Figure 4: The average throughput node 21 is down at 300 and node 40 is down at 500 and both repaired at 800

OA does not need any global information such as the structure of the topology and cost of links among peers. It can not only balance the local traffic flow but also enhance fault tolerance.

7 Conclusion

Routing is a key factor for network routing, and mobile agent-based routing is newly proposed for use in large dynamic network. In this paper, we proposed a new mobile agent-based routing algorithm in which the balance of traffic cost is considered. On each node, there is a probability distribution for an agent to select one of the neighboring nodes and move to. We macroscopically define a traffic cost function for each link according to the

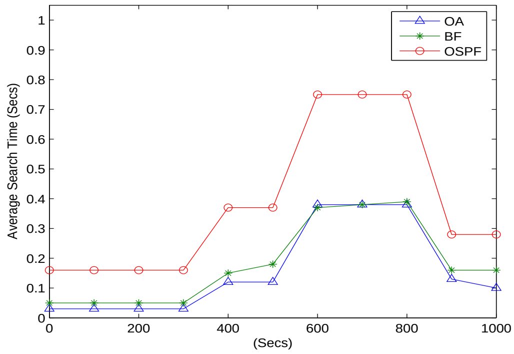

Figure 5: The average search time when node 21 is down at 300 and node 40 is down at 500 and both repaired at 800

known information and found a probability distribution that not only makes inference on the known information but also balances the traffic costs. The proposed approach can realize the advanced features of the Knowledge Grid [35]. Theoretical analysis showed that our probability distribution approximates to the balance of traffic costs. The main contributions of this paper can be summarized as follows:

- We proposed a novel mobile agent-based routing model. In our model, a routing agent on a node select a neighboring node to move according to a probability distribution which should both makes inference on all known routing information and balances the traffic load on each link, i.e., to be unbiased to each link.
- By using maximum entropy theory, we analyzed and achieved the optimal probability distribution for mobile agents to select a neighboring node to move to.
- We conducted simulation experiments to compare the performance of our algorithm with existing algorithms. The results showed that our algorithm outperforms existing algorithms.

In this paper, our model was established for traffic-based routing in large networks by using mobile agents. The analytical results can be easily extended to other application areas where bandwidth limitation is a main concern to programmer or there is a large amount of data transmission, such as peer to peer networks and wireless sensor networks.

Appendix I: Proof of Lemma 1

Proof For $\forall x \in R^n$ and $\forall p_j \in \Delta_j$, it is easy to see that

$$\sum_{i \in NB(j)} p_{ji} f_{ji}(x) \le f_{\max}^{(j)}(x). \tag{16}$$

Therefore,

$$\sup_{p_j \in \Delta_j} \ell_j(x, p_j) \le f_{\max}^{(j)}(x). \tag{17}$$

Let $I_{\max}^{(j)}(x)$ be the indicator set of element functions $f_{ji}(x)(i \in NB(j))$ that equal to the maximum value function $f_{\max}^{(j)}$ at point x, i.e.,

$$I_{\max}^{(j)}(x) := \{k | f_{jk}(x) = f_{\max}^{(j)}(x)\}. \tag{18}$$

If $k \in I_{\max}^{(j)}(x)$, then for arbitrary $x \in R^n$ and $p_j \in \Delta_j$, we have

$$\sup_{p_j \in \Delta_j} \ell_j(x, p_j) \ge \sum_{i \in NB(j)} \bar{p}_{ji} f_{ji}(x) = f_{\max}^{(j)}(x) b \tag{19}$$

where

$$\bar{p}_{ji} = \begin{cases} 1, & i = k; \\ 0, & i \neq k. \end{cases} \tag{20}$$

From (17) and (19), the first equality in (6) is hold. Consider that Δ_j is a tight set and $\ell_j(x, p_j)$ is a continuous function on p_j, the second equality in (6) is also hold. □

Appendix II: A Brief Look at Maximum Entropy Theory

In [28], Shannon first introduced the concept of entropy into informatics as a measurement of uncertainty. Suppose that there are a set of possible events whose probabilities of occurrence are $\lambda_1, \lambda_2, \cdots, \lambda_n$. These probabilities are known but that is all we know concerning which event will occur. Can we find a measure of how much "choice" is involved in the selection of the event or of how uncertain we are of the outcome? Shannon pointed that if there is such a measure, say $H(\lambda_1, \lambda_2, \cdots, \lambda_n)$, it should have the following properties:

1. H should be continuous on λ_i.
2. If all λ_i are equal, i.e., $\lambda_i = 1/n$, then H should be a monotonically increasing function of n.
3. If a choice is broken down into two successive choices, the original H should be the weighted sum of the individual values of H.

In [28], it is proved that the entropy function $H = -k \sum_{i=1}^{n} \lambda_i \ln \lambda_i$ is the only function that can satisfies all the requirements, where k is a positive constant decided by measurement units. Usually, k is set to be 1. The Shannon entropy has the following properties:

1. $H_n(\lambda_1, \lambda_2, \cdots, \lambda_n) \geq 0$;
2. If $\lambda_k = 1$ and $\lambda_i = 0$ $(i = 1, 2, \cdots, n; i \neq k)$, then $H_n(\lambda_1, \lambda_2, \cdots, \lambda_n) = 0$;
3. $H_{n+1}(\lambda_1, \lambda_2, \cdots, \lambda_n, \lambda_{n+1} = 0) = H_n(\lambda_1, \lambda_2, \cdots, \lambda_n)$;
4. $H_n(\lambda_1, \lambda_2, \cdots, \lambda_n) \leq H_n(1/n, 1/n, \cdots, 1/n) = \ln n$;
5. $H_n(\lambda_1, \lambda_2, \cdots, \lambda_n)$ is a symmetrical concave function on all variables.

where $H = -\sum_{i=1}^{n} \lambda_i \ln \lambda_i$.

E. T. Jaynes found that in many probabilistic executions, the resulting probability distribution cannot foreknown; thus, the entropy cannot be calculated. But he also claimed that the probability distribution could be induced by the accumulated test data such as the mean and the variance. In [14], E.T.Jaynes proposed the maximum entropy theory: "in making inference on the basis of partial information we must use that probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we do not have". Notice that "entropy" is a measurement of the degree of uncertainty and the great the entropy's value, the less known information, the maximum entropy theory can be mathematically expressed as follows:

$$\begin{cases} \max & H = -\sum_{i=1}^{N} \lambda_i \ln \lambda_i \\ s.t. & \sum_{i=1}^{N} \lambda_i = 1; \\ & \sum_{i=1}^{N} \lambda_i g_j(x_i) = E\left[g_j\right], j = 1, 2, \cdots, m; \\ & \lambda_i \geq 0, i = 1, 2, \cdots, N, \end{cases} \tag{21}$$

where $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n)$, $g_j (j = 1, 2, \cdots, m)$ is some predefined constrained function, and $E[\cdot]$ is the mean of these constrained function.

Templeman et al. [29] first applied maximum entropy theory to solve optimization problems in which the objective function is unanimously approximated by a smooth one. By solving the resulting problem, an approximate solution of the original problem can be obtained. The purpose of deploying maximum entropy theory in agents' searching process is to find a probability distribution that both satisfies the known routing information and mostly approximate to the unbiased (uniform) distribution.

Appendix III: Proof of Theorem 1

Proof From the definition of indicator function, Problem (12) can be reduced as

$$\sup_{p_j \in R^{|NB(j)|}} \left\{ \sum_{i \in NB(j)} p_{ji} f_{ji}(x) - \theta^{-1} \sum_{i \in NB(j)} p_{ji} \ln p_{ji} - \delta(p_j | \Delta_j) \right\}, \tag{22}$$

where $\delta(p_j|\Delta_j)$ is an indicator function on the closed convex set Δ_j. From the strictly convex property of entropy function $\sum_{i \in NB(j)} p_{ji} \ln p_{ji}$, it can be seen that for arbitrary fixed $x \in R^n$, the object function of the maximum problem (22) is a closed normal strictly concave function on variable p_j and the effective region is the tight set Δ_j. Therefore, from the Weiertrass theory, the maximum problem exists an unique solution $p_j^*(x, \theta)$ and reaches its finite optimal value on the unique solution. That is, Problem (12) defines a real value function on R^n as follows:

$$F_\theta^{(j)}(x) := \sum_{p_j \in \Delta_j} L_\theta^{(j)}(x, p_j) = L_\theta^{(j)}\left[x, p_j^*(x, \theta)\right]. \tag{23}$$

Consider that function $-\sum_{i \in NB(j)} p_{ji} \ln p_{ji}$ is unnegative on the bounded closed convex set Δ_j and has an upper bound $(\ln m)/\theta$, that is

$$1 \leq -\sum_{i \in NB(j)} p_{ji} \ln p_{ji} \leq \frac{\ln m}{\theta}, \forall p_j \in \Delta_j, \tag{24}$$

we have

$$\sup_{p_j \in \Delta_j} \ell_j(x, p_j) \leq \sup_{p_j \in \Delta_j} L_\theta^{(j)}(x, p_j) \leq \sup_{p_j \in \Delta_j} \ell_j(x, p_j) + \frac{\ln m}{\theta}. \tag{25}$$

Thus, from (6) and (23), we have

$$f_{\max}^{(j)}(x) \leq F_\theta^{(j)}(x) \leq f_{\max}^{(j)}(x) + \frac{\ln m}{\theta}. \tag{26}$$

This indicates that the function $F_\theta^{(j)}(x)$, defined by the maximum problem (12), is uniformly approximate to $f_{\max}^{(j)}(x)$ on the whole space R^n.

At the same time, if function $K(p_j)$ is defined as follows:

$$K(p_j) := \begin{cases} \sum_{i \in NB(j)} p_{ji} \ln p_{ji} & p_{ji} \geq 0; \\ +\infty & p_{ji} < 0, \end{cases} \tag{27}$$

then it is a closed normal strictly convex function on $R^{|NB(j)|}$ and $\mathrm{ri}(\mathrm{dom}K) \cap \mathrm{ri}(\Delta_j) \neq \phi$. Therefore, from (22) and the definition of convex conjugate function, we have

$$\begin{aligned} F_\theta^{(j)}(x) &= \theta^{-1} \cdot (K + \delta)^*(\theta F_j(x)) \\ &= \theta^{-1} \cdot (K^* \Box \delta^*)(\theta F_j(x))), \end{aligned} \tag{28}$$

where $F_j(x) := (f_{ji}(x))_{i \in NB(j)}^T$ is a vector function and K^* is the convex conjugate function of K. Since $K \in Leg(R_+^m)$, i.e., $K(p_j)$ is a Legendre convex function, $K^* \in Leg(\mathrm{int}(\mathrm{dom}K^*))$ and $(K^* \Box \delta^*)(\cdot)$ is essentially smooth. Thus, from (28) and the property that $F_\theta^{(j)}(x)$ is a real value function on R^n, we have $\mathrm{dom}(K^* \Box \delta^*) = R^{|NB(j)|}$. Due to the continuous differentiable property of $F_j(x)$, $F_\theta^{(j)}(x)$ is a smooth function.

Combine the above two aspects, the theorem is proven. □

Appendix IV: Proof of Theorem 2

Proof 1. Here, we provide a different proof from Theorem 1. From the expression of $F_\theta^{(j)}(x)$ in (15), we have

$$\begin{aligned} F_\theta^{(j)}(x) &= f_{\max}^{(j)}(x) \\ &+ \frac{1}{\theta} \ln \left\{ \sum_{i \in NB(j)} \exp \left[\theta \left(f_{ji}(x) - f_{\max}^{(j)}(x) \right) \right] \right\}. \end{aligned} \tag{29}$$

From the definition of maximum value function $f_{\max}^{(j)}(x)$ in (2), we have

$$1 \leq \sum_{i \in NB(j)} \exp \left[\theta \left(f_{ji}(x) - f_{\max}^{(j)}(x) \right) \right] \leq m. \tag{30}$$

Substitute this inequality into (29), we have

$$f_{\max}^{(j)}(x) + \frac{1}{\theta} \ln 1 \leq F_\theta^{(j)}(x) \leq f_{\max}^{(j)}(x) + \frac{1}{\theta} \ln m. \tag{31}$$

2. Take limitation on both side of (31), this property is proven.
3. For any $x, y \in R^n$ and $\alpha \in (0,1)$, since all the functions f_{ji} $(i \in NB(j))$ are convex, we have

$$\begin{aligned} & F_\theta^{(j)}(\alpha x + (1-\alpha) y) \\ &= \frac{1}{\theta} \ln \left\{ \sum_{i \in NB(j)} \exp \left[\theta f_{ji} \left(\alpha x + (1-\alpha) y \right) \right] \right\} \\ &\leq \frac{1}{\theta} \ln \left\{ \sum_{i \in NB(j)} \exp \left[\theta \left(\alpha f_{ji}(x) + (1-\alpha) f_{ji}(y) \right) \right] \right\} \\ &= \frac{1}{\theta} \ln \left\{ \sum_{i \in NB(j)} \left(\exp \left[\theta f_{ji}(x) \right] \right)^\alpha \left(\exp \left[\theta f_{ji}(y) \right] \right)^{1-\alpha} \right\}. \end{aligned}$$

Applying to the Hölder inequality, we have

$$\begin{aligned} & \sum_{i \in NB(j)} \left(\exp \left[\theta f_{ji}(x) \right] \right)^\alpha \left(\exp \left[\theta f_{ji}(y) \right] \right)^{1-\alpha} \\ & \quad \leq \left\{ \sum_{i \in NB(j)} \exp \left[\theta f_{ji}(x) \right] \right\}^\alpha \\ & \quad \quad \cdot \left\{ \sum_{i \in NB(j)} \exp \left[\theta f_{ji}(y) \right] \right\}^{1-\alpha}. \end{aligned}$$

Combined the above two relationships, we have

$$
\begin{aligned}
& F_\theta^{(j)}\left(\alpha x + (1-\alpha) y\right) \\
& \leq \frac{\alpha}{\theta} \ln\left\{ \sum_{i \in NB(j)} \exp\left[\theta f_{ji}(x)\right] \right\} \\
& \quad + \frac{1-\alpha}{\theta} \ln\left\{ \sum_{i \in NB(i)} \exp\left[\theta f_{ji}(y)\right] \right\} \\
& = \alpha F_\theta^{(j)}(x) + (1-\alpha) F_\theta^{(j)}(y).
\end{aligned} \tag{32}
$$

Hence, function $F_\theta^{(j)}(x)$ is a convex function.
4. Take derivation about x on both side of (29), we have

$$
\begin{aligned}
& \nabla_x F_\theta^{(j)}(x) = \nabla_x f_{\max}^{(j)}(x) \\
& + \frac{1}{\theta} \nabla_x \ln\left\{ \sum_{i \in NB(j)} \exp\left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right] \right\}.
\end{aligned} \tag{33}
$$

Since

$$
\begin{aligned}
& \nabla_x \ln\left\{ \sum_{i \in NB(j)} \exp\left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right] \right\} \\
& = \frac{\sum_{i \in NB(j)} \nabla_x \exp\left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right]}{\sum_{i \in NB(j)} \exp\left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right]} \\
& = \sum_{i \in NB(j)} \frac{\exp\left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right]}{\sum_{i \in NB(j)} \exp\left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right]} \\
& \quad \cdot \nabla_x \left[\theta\left(f_{ji}(x) - f_{\max}^{(j)}(x)\right)\right] \\
& = \theta \sum_{i \in NB(j)} \hat{p}_{ji}(x) \nabla_x f_{ji}(x) - \nabla_x f_{\max}^{(j)}(x),
\end{aligned} \tag{34}
$$

where $\hat{p}_{ji} x$ is defined as (14). Substitute this results into (33), this property is proven.
5. From the expression of function $F_\theta^{(j)}(x)$ in (15) and the definition of $\hat{p}_{ji}(x)$ in (14), we have

$$
\begin{aligned}
& \frac{\partial F_\theta^{(j)}(x)}{\partial \theta} \\
& = -\frac{F_\theta^{(j)}(x)}{\theta} + \frac{\sum_{i \in NB(j)} \exp\left(\theta f_{ji}(x)\right) f_{ji}(x)}{\theta \sum_{i \in NB(j)} \exp\left[\theta f_{ji}(x)\right]} \\
& = \theta^{-1} \left[\sum_{i \in NB(j)} \hat{p}_{ji}(x) f_{ji}(x) - F_\theta^{(j)}(x) \right].
\end{aligned} \tag{35}
$$

According to (11) and (15), we have

$$
\begin{aligned}
F_\theta^{(j)}(x) = & \sum_{i \in NB(j)} p_{ji}(x) f_{ji}(x) \\
& - \theta^{-1} \sum_{i \in NB(j)} p_{ji}(x) \ln\left[p_{ji}(x)\right].
\end{aligned} \tag{36}
$$

Therefore,

$$\partial F_\theta^{(j)}(x)/\partial\theta = \theta^{-2} \sum_{i \in NB(j)} p_{ji}(x) \ln\left[p_{ji}(x)\right]. \tag{37}$$

Thus, from the following inequality:

$$-\frac{\ln m}{\theta} \leq \theta^{-2} \sum_{i=1}^{m} p_{ji}(x) \ln\left[p_{ji}(x)\right] \leq 0, \quad \forall x \in R^n, \tag{38}$$

this property is proven.
6. From property 5, we can see that function $F_\theta^{(j)}(x)$ is a decreasing function on θ, thus, this property is straight forward for property 5. □

References

[1] B. Baran and R. Sosa, "A New Approach for Antnet Routing," *Proc. of the 9th Int. Conf. on Computer, Communications and Networks*, pp. 303-308, 2000.

[2] R. Beckers, J. L. Deneubourg, and S. Goss, "Trails and U-turns in the Selection of the Shortest Path by the Ant Lasius Niger," *Theoretical Biology*, No. 159, pp. 397-415, 1992.

[3] B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus, "Mobile Agents in Distributed Information Retrieval," *Intelligent Information Agents: Agents-Based Information Discovery and Management on the Internet*, M. Klusch, ed., Springer-Verlag, Chapter 15, pp. 355-395, 1999.

[4] G. D. Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, Vol. 9, pp. 317-365, 1998.

[5] G. D. Caro and M. Dorigo, "Mobile Agents for Adaptive Routing," *Proc. of the 31st Hawaii Int. Conf. on System Sciences*, pp. 74-83, 1998.

[6] J. Claessens, B. Preneel, and J. Vandewalle, "(How) Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of the Security Issues and the Current Solutions," *ACM Trans. on Internet Technology*, Vol. 3, No. 1, pp. 28-48, 2003.

[7] D. Milojicic, "Guest Editor's Introduction: Agent Systems and Applications," *IEEE Concurrency*, Vol.8, No. 2, pp. 22-23, 2000.

[8] E. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerical Mathematics*, pp. 269-271, 1959.

[9] M. Dorigo, V. Maniezzo, A. Colorni, "The Ant System: Optimization by a colony of c ooperating agents," *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, Vol. 26, No. 1, pp. 29-41, 1996.

[10] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the Theorey of NP-Completeness," *Freeman*, San Francisco, 1979.

[11] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels, "Self-Organized Shortcuts in the Argentine Ant," *Naturwissenschaften*, Vol. 76, pp. 579-581, 1989.

[12] P. S. Heck and S. Ghosh, "A Study of Synthetic Creativity through Behavior Modeling and Simulation of an Ant Colony," *IEEE Intelligent Systems*, Vol. 15, No. 6, pp. 58-66, 2000.

[13] B. Hölldobler and E. O. Wilson, "The Ants," *Springer-Verlag*, Berlin, 1990.

[14] E. T. Joynes, "Information theory and statistical mechanics," *The Physical Review.* No. 108, pp. 171-190, 1957.

[15] G. Karjoth, D. Lange, and M. Oshima, "A Security Model for Aglets," *IEEE Internet Computing*, Vol. 1, No. 4, pp. 68-77, 1997.

[16] D. Lange and M. Oshima, "Programming and Developing Java Mobile Agents with Aglets," *Addison Wesley*, 1998.

[17] D. Lange and M. Oshima, "Seven Good Reasons for Mobile Agents," *Communications of the ACM*, Vol. 42, pp. 88-89, 1999.

[18] K. Li and H. Shen, "Coordinated Enroute Multimedia Object Caching in Transcoding Proxies for Tree Networks," *ACM Transactions on Multimedia Computing, Communications and Applications (TOMCAPP)*, Vol. 1, No. 3, pp. 289-314, 2005.

[19] K. Li, H. Shen, F. Y. L. Chin, and S. Q. Zheng, "Optimal Methods for Coordinated Enroute Web Caching for Tree Networks," *ACM Trans. on Internet Technology (TOIT)*, Vol. 5, No. 3, pp. 480-507, 2005.

[20] S. Liang, A. Zincir-Heywood, and M. Heywood, "The Effect of Routing Under Local Information Using a Social Insect Metaphor," *Proc. of IEEE Int'l Congress on Evolutionary Computation*, pp 1438-1443, 2002.

[21] S. Liang, A. Zincir-Heywood, and M. Heywood, "Intelligent Packets for Dynamic Network Routing Using Distributed Genetic Algorithm," *Proc. of Genetic and Evolutionary Computation Conference*, pp. 88 - 96, 2002.

[22] D. Milojicic, "Trend Wars: Mobile Agent Applications," *IEEE Concurrency*, Vol. 7, No. 3, pp. 80-90, 1999.

[23] S. Pleisch and A. Schiper, "Fault-Tolerant Mobile Agent Execution," *IEEE Trans. on Computers*, Vol. 52, No. 2, pp. 209-222, 2003.

[24] W. Qu, H. Shen, and J. Sum, "New Analysis on Mobile Agents Based Network Routing," *Proc. of the 3rd Int'l Conf. on Hybrid Intelligent Systems (HIS'03)*, pp. 769-778, 2003.

[25] J. Sum, H. Shen, C. Leung, and G. Young, "Analysis on a Mobile Agent-Based Ant Algorithm for Network Routing and Management," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 14, No. 3, pp. 193-202, 2003.

[26] D. Schoder and T. Eymann, "The Real Challenges of Mobile Agents," *Communications of the ACM*, Vol. 43, No. 6, pp. 111-112, 2000.

[27] A.U. Shankar, C. Alaettinoglu, K. Dussa-Zieger, and I. Matta, "Performance Comparison of Routing Protocols under Dynamic and Static File Transfer Connections", *ACM Computer Communication Review*, Vol. 22, No. 5, pp. 39-52, 1992.

[28] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, No. 3, pp. 379-428, 1948.

[29] A. B. Templeman and X. Li, "A Maximum Entropy Approach to Constrained Non-Linear Programming," *Engineering Optimization*, Vol. 12, No. 3, pp. 191-205, 1987.

[30] A. Varga, "OMNeT++: Discrete Event Simulation System: User Manual", *http://www.omnetpp.org/* , July 2006.

[31] Y. Wang and J. Liu, " Modelling Agent-Based Load Balancing with Time Delays," *Proc. of the 2003 IEEE/WIC Int'l Conf. on Intelligent Agent Technology*, pp. 189-195, 2003.

[32] H.F. Wedde, M. Farooq, and Y. Zhang, "BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior", *Proc. of the 4th Int'l Wksp. on Ant Colony Optimization and Swarm Intelligence*, pp. 83-94, 2004.

[33] T. White, B. Pagurek, and F. Oppacher, "ASGA: Improving the Ant System by Integration with Genetic Algorithms," *Proc. of the 3rd Genetic Programming (GP/SGA'98)*, pp. 610-617, 1998.

[34] D. Xu, J. Yin, Y. Deng, and J. Ding, "A Formal Architectural Model for Logical Agent Mobility," *IEEE Trans. on Software Engineering*, Vol. 29, No. 1, pp. 31-45, 2003.

[35] H.Zhuge, X.Sun, J. Liu, E. Yao, and X. Chen, "A Scalable P2P Platform for the Knowledge Grid," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 17, No. 12, pp. 1721-1736, 2005.