J Supercomput (2012) 60:360–388
DOI 10.1007/s11227-008-0220-2

Statistical behaviors of mobile agents in network routing

Wenyu Qu · Keqiu Li · Masaru Kitsuregawa · Weilian Xue

Published online: 25 July 2008

Abstract Mobile agent-based network routing is a new technique for routing in large-scale networks. An analysis of the searching activity and population growth of mobile agents is important for improving performance in agent-driven networks. In this paper, we describe a general execution model of mobile agents for network routing and classify it into two cases. For each case, we analyze the population distribution of mobile agents (the distribution of mobile agents running in the network) and the probability of success (the probability that an agent can find its destination). We also perform extensive experiments for various network topologies to validate our analytical results. Both theoretical and experimental results show that the population distribution and the probability of success of mobile agents can be controlled by locally adjusting relevant parameters, such as the number of agents generated per request, the number of jumps each mobile agent can move, etc. Our results reveal new theoretical insights into the statistical behaviors of mobile agents and provide useful tools for effectively managing mobile agents in large networks.

Keywords Mobile agents · Routing · Population distribution · Probability of success

W. Qu
School of Computer Science and Technology, Dalian Maritime University, 1 Linghai Road, Ganjinzi District, Dalian, 116026, China
e-mail: quwenyo@tkl.iis.u-tokyo.ac.jp

K. Li (✉)
Department of Computer Science and Engineering, Dalian University of Technology, 1 Linggong Road, Ganjinzi District, Dalian, 116024, China
e-mail: keqiu@dlut.edu.cn

W. Qu · M. Kitsuregawa
Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan

W. Xue
School of Management, Liaoning Normal University, 850 Huanghe Road, Dalian, 116029, China

1 Introduction

Many network routing techniques have been developed to deal with the enormous amount of data available on the Internet. These techniques have created exciting opportunities for both business and science [7, 26, 27, 34]. 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 techniques and is an effective way to reduce network load and latency [25]. In [29], Milojicic described that mobile agents are autonomous, adaptive, reactive, mobile, cooperative, interactive, and delegated software entities. The application of mobile agents in network routing has attracted significant attention [9, 30, 40]. Successful examples of mobile agent applications can be found in [23, 24]. The use of mobile agents in applications ranging from electronic commerce to distributed computation has also been studied extensively [5, 6, 14, 17, 22, 28, 36].

Routing is a key factor for network performance. It is the process of moving a packet of data from source to destination. As searching for the optimal path in a stationary network is already a difficult problem, the searching for the optimal path in a dynamic network or mobile network will be much more difficult. Mobile agent-based routing algorithm is a promising option for use in these environments [11]. In a mobile agent-based routing algorithm [13, 18], a number of mobile agents are generated and dispatched to the network once the server receives a request. 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. Since little communication is needed between the agents and the server during the process of searching, the burden on the network generated by mobile agents is very light.

In order to conserve network resources and achieve a good probability of success, it is desirable to dispatch a small number of mobile agents. In a mobile agent-based routing algorithm for large-scale networks, mobile agents will be generated frequently. If there are too many agents running in the network, they will consume too many network resources, thus affecting the network performance and ultimately blocking the entire network; on the other hand, if there are too few agents running in the network, there is no guarantee that the destination will be found quickly. Therefore, for network management, it is necessary to analyze both the population growth of mobile agents and the probability of success. Furthermore, the probability of success directly affects the searching process, but existing studies pay little attention to this probability.

In this paper, considering the fact that the neighborhood information of a host node is readily available (e.g., it is built in the routing table in TCP/IP), we propose a general agent-based routing model, in which a mobile agent knows information not only about its host node, but also about the neighboring nodes of its host node. Since the connectivity of different nodes may change dynamically over time, the agents have to dynamically adapt themselves to the environment, which increases the difficulty for theoretical analysis. We further classify the model into two cases, namely settleable and nonsettleable (to the host node), and analyze the population distribution of mobile agents and the probability of success for each case. Both our theoretical and experimental results show that we can control the number of mobile agents and the probability of success by tuning the number of agents generated per request and the number of jumps each mobile agent can make. The main contributions of this paper are summarized as follows:

- A general and more practical agent-based routing model is described and is further classified into settleable and nonsettleable cases.
- An analysis on population growth of mobile agents is presented, providing a useful tool to reduce computational resource consumption by adjusting the number of agents to be generated at individual nodes.
- The probability of success is analyzed, which serves as an important measure for monitoring network performance.
- Extensive experiments validating our analysis have been performed and compared with previous work.

The rest of this paper is organized as follows: Sect. 2 provides a general view on related work. Section 3 presents a general agent-based routing model, introduces the notations used in this paper, and gives some basic assumptions in our model. Section 4 and Sect. 5 present mathematical analysis on the population distribution and the probability of success for both cases. Section 6 describes our simulation studies, and Sect. 7 concludes our paper.

2 Related work

With the exploitation of the Internet, mobile agent technology has attracted an increasing attention. In [40], the authors used a system net, agent nets, and a connector net to model the environment, agents, and the connector, respectively. They also presented a case study of modeling and analyzing an information retrieval system with mobile agents. In [9], an overview of the security issues of mobile agents was given and a state-of-the-art survey of mobile agent-based secure electronic transactions was presented. In [30], the authors identified two important properties (i.e., nonblocking and exactly-once) for fault-tolerant mobile agent execution and proposed that fault-tolerant mobile agent execution could be modeled as a sequence of agreement problems.

Routing is a key feature of the Internet because it enables messages to pass from one computer to another and eventually reach the target machine[38]. Existing routing algorithms can be broadly classified into two classes: static or dynamic. In static routing, packets are sent to the destination following a predetermined path, without considering the current network state. Static routing is appropriate for small networks and some dedicated links. In a large distributed network with irregular topology, routing decisions can only be made on the basis of local and approximate information about the current and the future network states. Dynamic routing can discover the changes of network states, automatically adjust its routing tables, and inform other routers of the changes.

Routing Information Protocol (RIP) is a popular dynamic routing protocol for small private networks [33]. With RIP, routers periodically exchange entire routing tables. Open Shortest Path First (OSPF) is now the most important protocol on large networks that provide Internet service [33]. With OSPF, routers send routing information to all nodes by calculating the shortest path to each node based on a topology of the Internet constructed by each node. Each router sends that portion of the routing table that describes the state of its own links. Both RIP and OSPF choose the path with minimum cost (generally the shortest path) between the pair of nodes. In [12, 32], the authors studied the behavior of routers and announced that specific routers are the cause of bottlenecks in the Internet. This is because the path with minimum cost could congest, in spite of other paths, possibly expensive, but less congested. In [20], the authors provided necessary and sufficient conditions for deadlock-free unicast and multicast routing with the path-based routing model in interconnection networks.

Real ants are capable of finding the shortest path from a food source to the nest based only on local information [4, 13, 16, 19]. Inspired from the research on ants, Caro and Dorigo [10, 11] 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 [10, 11], the authors conducted many experiments to compare the performance of the mobile agent-based routing algorithm with both static and adaptive state-of the-art routing algorithms, such as RIP and OSPF. The experimental results showed that the mobile agent-based routing algorithm is very encouraging over real and artificial IP datagram networks.

Although the efficiency of applying the mobile agent technique has been demonstrated and reported in the literature, the mathematical modeling and analysis of mobile agents' behaviors are still in its infancy. In [7], Brewington et al. formulated a method of mobile agent planning, which is analogous to the traveling salesman problem [15] to decide the sequence of nodes to be visited by minimizing the total execution time until the desired information is found. In [21], several statistical analysis of mobile agents were proposed, including dwell time on nodes, average life-span, and the reports arrival process. Theoretical analysis on the execution time and the number of agents can also be found in [1–3, 35].

In [35], an ant-like routing model was studied and the number of mobile agents was estimated. In [31], a smaller upper bound on the number of mobile agents was provided based on the same model, and for the first time the probability of success was considered. Mobile agents in the model were assumed to be blind; they could only know information about the node they were in and know nothing about the surrounding nodes. Under this assumption, the action of mobile agents for each step static, which reduced the analytical difficulty. However, it also resulted in a low probability of success. To improve the search efficiency, in the model proposed in this paper, mobile agents know information not only about their host nodes, but also about the neighboring nodes of their host nodes. We analyze both the population distribution and the probability of success of mobile agents. Comparison between the proposed model and the one in [31] is provided in Sect. 6.

3 A general mobile agent-based routing model

In our model, mobile agents are the medium for carrying routing information. The behavior of agents corresponds to the transport of routing information among nodes. For a large network, 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. Routing table on each node only details its neighboring nodes. A node which generates mobile agents is called the server of these agents. At any time, requests may be keyed in the network. Once a request for sending a packet is received from 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 routings such as life-span limit 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. These agents will then move out from the server and roam in the network. Once an agent reaches a node, it will search for the destination in the set of neighboring nodes of the host node. If the agent finds its 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. Otherwise, it will select a neighboring node and move on. To eliminate unnecessary searching in the network, a life-span limit is assigned to each agent. An agent will die if it cannot find its destination in its life-span limit. Moreover, if an agent cannot return to its server in two times the life-span limit (e.g., its return route is interrupted due to a link/node failure), the agent also will die. 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. In this paper, we classify the model into the following two cases:

- *Settleable* case: If the host node is not the agent's destination, the agent can either stay in the host node or move to a neighboring node.
- *Nonsettleable* case: An agent will not die unless it finds its destination or it is out of its life-span limit. If an agent cannot find its destination on the host node, it must jump out from the host node and go on searching.

As we know, the topology of a network can be decided uniquely by its connectivity matrix. Assume that there are n nodes in the network, matrix $C = (c_1, c_2, \dots, c_n) = (c_{ij})_{n\times n}$ is the connectivity matrix which describes the connectivity of the network, i.e., if there is a direct link between node i and node j ($i \neq j$), then $c_{ij} = c_{ji} = 1$; otherwise, $c_{ij} = c_{ji} = 0$. Let $d_j = \|c_j\|_1 = \sum_{i=1}^{n} |c_{ij}|$, $\sigma_1 = \max_{1\le j\le n} d_j$, $\sigma_n = \min_{1\le j\le n} d_j$. Clearly, $D = \mathrm{diag}(d_1, d_2, \dots, d_n)$ is a diagonal matrix. It is easy to see that d_j is the number of neighboring nodes of the j-th node, i.e., the degree of the j-th node, and $\|C\|_1 = \max_{1\le j\le n} \|c_j\|_1 = \sigma_1$. For the small network shown in Fig. 1, we have $n = 5$, $\sigma_1 = 3$, and $\sigma_5 = 2$. Matrices C and D are as follows:

$$C = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

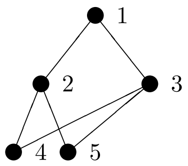

Fig. 1 An example of a small network

There are also some assumptions convenient for our analysis:

1. There are n nodes in the network, and each node has the same probability of $1/n$ to be the destination. The assumption of uniform probability of $1/n$ has been widely applied in the open literature [10, 11, 39].
2. The expected number of requests received from a server once is m. Once a request arrives, k agents are created and sent out into the network.
3. The life-span limit of agents is set to be d.

4 Population distribution

As we mentioned in Sects. 1 and 2, mobile agents are frequently generated and dispatched to the network. If the number of agents generated per request is small, there is no guarantee that the destination will be found quickly. On the other hand, if there are too many agents running in the network, they will introduce too much computational overhead; the host nodes will eventually become very busy and indirectly block the network traffic. Therefore, analysis of the number of mobile agents running in the network and on each node is another important task. Since mobile agents will search for their destinations node by node in the network, the agents in one node can be divided into two parts, as shown in Fig. 2.

We claim that the population distribution of mobile agents running in the network at time t can be expressed as follows:

$$\begin{aligned}\vec{p}(t) &= k\vec{r}(t-1) + kA\vec{r}(t-2)\\ &\quad + B\vec{p}(t-1) - B^d k\vec{r}(t-d-1). \end{aligned} \tag{1}$$

where

$$\vec{p}(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ \cdots \\ p_n(t) \end{bmatrix}, \quad \vec{r}(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \\ \cdots \\ r_n(t) \end{bmatrix},$$

$p_j(t)$ is the number of mobile agents running on the j-th node, while $r_j(t)$ indicates the number of requests received by the j-th node at time t and $r_j(t) = 0$ when $t < 0$. k is the number of mobile agents generated per request, d is the life-span limit of mobile agents. $A = (a_{ji})$ and $B = (b_{ji})$ are $n \times n$ coefficient matrices with elements

$$a_{ji} = \begin{cases} \Pr^{(1)}\{i \to j\} & \\ -\Pr^{(2)}\{i \to j\} & \text{if } j \in NB(i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

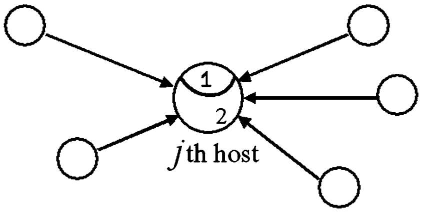

Fig. 2 Mobile agents in node j can be divided into two parts: *Part 1*—those generated by node j, *Part 2*—those migrated from j's neighboring nodes

and

$$b_{ji} = \begin{cases} \Pr^{(2)}\{i \to j\} & \text{if } j \in NB(i) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$NB(i)$ is the set of node i's neighboring nodes. $\Pr^{(l)}\{i \to j\}$ $(l = 1, 2)$ is the probability that a mobile agent in the l-th part at node i will select the j-th node to move to when $j \in NB(i)$.

From (1), we can see that the population distribution of mobile agents can be modeled as a stochastic process. In the following, we explain the validity of (1) in detail. It is easy to see that the number of mobile agents newly generated at time t (part 1 in Fig. 2) equals to $k \cdot r_j(t-1)$. Denote the number of agents in node j at time t that come from the i-th node by $p_{ji}(t)$, then the number of mobile agents in part 2 equals to $\sum_{i \in NB(j)} p_{ji}(t)$. Therefore, the total number of mobile agents running on the j-th node at time t, denoted by $p_j(t)$, satisfies

$$p_j(t) = k \cdot r_j(t-1) + \sum_{i \in NB(j)} p_{ji}(t). \tag{4}$$

The analysis of $p_{ji}(t)$ is much more complex. To simplify the analysis, we first consider the situation that mobile agents have an infinite life-span. Regarding to the population distribution of mobile agents with infinite life-span in the network, we have the following lemma.

Lemma 1 *The number of mobile agents with infinite life-span running on the j-th node with infinite life-span satisfies*

$$\overrightarrow{p}(t) = k\overrightarrow{r}(t-1) + kA\overrightarrow{r}(t-2) + B\overrightarrow{p}(t-1). \tag{5}$$

Proof It is no surprise that each neighboring node of the i-th node has the same possibility to be selected by mobile agents in the i-th node since each direction is equally likely to link with the destination. Thus, the average number of mobile agents that traverse from the i-th node to the j-th node at time t, denoted by $p_{ji}(t)$, equals to $\sum_{l=1}^{2} p_i^{(l)}(t-1)\Pr^{(l)}\{i \to j\}$, where $p_i^{(l)}(t-1)$ $(l = 1, 2)$ is the number of mobile agents belong to the l-th part on the i-th node at time $t-1$ (see Fig. 2). Thus, the number of mobile agents on the j-th node with infinite life-span satisfies

$$p_j(t) = kr_j(t-1) + \sum_{i \in NB(j)} \sum_{l=1}^{2} p_i^{(l)}(t-1)\mathrm{Pr}^{(l)}\{i \to j\}.$$

Since $p_i^{(1)}(t-1)$ equals to $kr_i(t-2)$, and $p_i^{(2)}(t-1)$ equals to $p_i(t-1) - kr_i(t-2)$, we have

$$\begin{aligned} p_j(t) &= kr_j(t-1) + \sum_{i \in NB(j)} \Big[kr_i(t-2)\mathrm{Pr}^{(1)}\{i \to j\} \\ &\quad + (p_i(t-1) - kr_i(t-2))\,\mathrm{Pr}^{(2)}\{i \to j\} \Big] \\ &= kr_j(t-1) + \sum_{i \in NB(j)} \Big[\mathrm{Pr}^{(2)}\{i \to j\} p_i(t-1) \\ &\quad + \Big(\mathrm{Pr}^{(1)}\{i \to j\} - \mathrm{Pr}^{(2)}\{i \to j\} \Big) kr_i(t-2) \Big]. \end{aligned}$$

Let $[\cdot]_j$ denotes the j-th entry of a vector. From (2) and (3), we have

$$\begin{aligned} &\sum_{i \in NB(j)} \mathrm{Pr}^2\{i \to j\} p_i(t-1) \\ &\quad = \sum_i \mathrm{Pr}^2\{i \to j\} p_i(t-1) \\ &\quad = \sum_i b_{ji} \cdot p_i(t-1) = [B\overrightarrow{p}(t-1)]_j, \end{aligned}$$

and

$$\begin{aligned} &\sum_{i \in NB(j)} \Big(\mathrm{Pr}^{(1)}\{i \to j\} - \mathrm{Pr}^{(2)}\{i \to j\} \Big) kr_i(t-2) \\ &\quad = \sum_i \Big(\mathrm{Pr}^{(1)}\{i \to j\} - \mathrm{Pr}^{(2)}\{i \to j\} \Big) kr_i(t-2) \\ &\quad = \sum_i a_{ji} \cdot r_i(t-2) = [A\overrightarrow{r}(t-2)]_j. \end{aligned}$$

Therefore, we have

$$\overrightarrow{p}(t) = k\overrightarrow{r}(t-1) + kA\overrightarrow{r}(t-2) + B\overrightarrow{p}(t-1).$$

□

Lemma 2 *Suppose that the distribution of mobile agents generated at $t = 0$ in the network is $\overrightarrow{p}(0)$, then the population distribution of these agents at time t is $B^t\overrightarrow{p}(0)$.*

Proof Since we only consider the distribution of mobile agents that generated at $t = 0$ and do not pay attention to other agents, $\overrightarrow{r}(t)$ in (5) equals to 0. Therefore,

$$\overrightarrow{p}(t) = B \cdot \overrightarrow{p}(t-1) = B^t \overrightarrow{p}(0).$$

□

From Lemma 2, it is straightforward that at time d, the distribution of mobile agents that are generated at time $t = 0$ is $B^d \overrightarrow{p}(0)$. Since these agents will die at time d, they should be removed from the distribution in (5). Thus, based on Lemmas 2 and 3, the population distribution of mobile agents with life-span limit at time t can be expressed as (1).

In particular, take expectation on both side of (1) and denote the average number of requests received from a node at any time by r, then when $t \leq d$, the population distribution of mobile agents satisfies

$$\begin{aligned}\overrightarrow{p}(t) &= kr(I+A)\overrightarrow{e} + B\overrightarrow{p}(t-1) \\ &= \sum_{i=0}^{t-1} B^i kr(I+A)\overrightarrow{e} + B^t \overrightarrow{p}(0),\end{aligned} \tag{6}$$

where $\overrightarrow{e} = (1, \ldots, 1)^T$. Here, we still use the notation $\overrightarrow{p}(t)$ as $E[\overrightarrow{p}(t)]$. When $t > d$, the population distribution of mobile agents satisfies

$$\begin{aligned}\overrightarrow{p}(t) &= (I+A)kr\overrightarrow{e} + B\overrightarrow{p}(t-1) - B^d kr\overrightarrow{e} \\ &= \sum_{i=0}^{t-d-1} B^i kr(I+A)\overrightarrow{e} - \sum_{j=0}^{t-d-1} B^j kr B^d \overrightarrow{e} + B^{t-d}\overrightarrow{p}(d) \\ &= \sum_{i=0}^{t-d-1} B^i kr(I+A)\overrightarrow{e} - \sum_{j=0}^{t-d-1} B^j kr B^d \overrightarrow{e} \\ &\quad + B^{t-d}\left[\sum_{l=0}^{d-1} B^l kr(I+A)\overrightarrow{e} + B^d \overrightarrow{p}(0)\right] \\ &= \sum_{i=0}^{t-1} B^i kr(I+A)\overrightarrow{e} + B^t \overrightarrow{p}(0) - \sum_{j=d}^{t-1} B^j kr\overrightarrow{e}.\end{aligned}$$

Thus, the population distribution of mobile agents in (1) can be described as follows

$$\overrightarrow{p}(t) = \begin{cases} \sum_{i=0}^{t-1} B^i kr(I+A)\overrightarrow{e} \\ \quad + B^t \overrightarrow{p}(0), & 0 \leq t \leq d; \\ \sum_{i=0}^{t-1} B^i kr(I+A)\overrightarrow{e} \\ \quad + B^t \overrightarrow{p}(0) - \sum_{j=d}^{t-1} B^j kr\overrightarrow{e}, & t > d. \end{cases} \tag{7}$$

4.1 Settleable case

As shown in Fig. 2, mobile agents in the i-th node can be subdivided into two parts: those newly generated in the i-th node and those migrated from its neighboring nodes. If an agent is newly generated, it can find its destination at birth with probability $(d_i + 1)/n$. Otherwise, the agent will select a neighboring node with probability $1/(d_i + 1)$. Therefore, we have

$$\Pr^{(1)}\{i \to j\} = \frac{1}{d_i + 1}\left(1 - \frac{d_i + 1}{n}\right).$$

For those agents came from neighboring nodes, since the i-th node and the node they came from have been checked, their chance of finding their destination has the probability $(d_i - 1)/n$. If they cannot find their destination, they will select a neighboring node with probability $1/d_i$. Therefore, we have

$$\Pr^{(2)}\{i \to j\} = \frac{1}{d_i}\left(1 - \frac{d_i - 1}{n}\right).$$

Substituting these two parameters into (1), the corresponding matrix A and B can be expressed as $A = C(D + I)^{-1} - (1 + \frac{1}{n})CD^{-1}$ and $B = (1 + \frac{1}{n})CD^{-1} + C/n$, respectively.

Lemma 3 *If there are* $\|\overrightarrow{p}(0)\|_1 = \delta$ *agents initially in the network* (*at time* 0), *then the total number of the survival agents running in the network at time* t, *is less than* $\delta(1 - \frac{\sigma_n - 1}{n})^t$.

Proof From the definition of matrix B, it is easy to see that

$$\|B\|_1 = \max\left|1 - \frac{d_i - 1}{n}\right| = 1 - \frac{\sigma_n - 1}{n}.$$

Therefore, from Lemma 2, we have

$$\sum_{1 \le j \le n} p_j(t) = \|\overrightarrow{p}(t)\|_1 = \|B^t \overrightarrow{p}(0)\|_1$$
$$\le (\|B\|_1)^t \, \|\overrightarrow{p}(0)\|_1.$$

□

Theorem 1 *If* $\overrightarrow{p}(0) = 0$, *the total number of agents running in the network at any time is no more than* $\alpha(\sigma_1, \sigma_n, d)nkr$ *where* $\alpha(\sigma_1, \sigma_n, d) = \frac{n\sigma_1 - \sigma_1 - 1}{(\sigma_1 + 1)(\sigma_n - 1)}[1 - (1 - \frac{\sigma_n - 1}{n})^d]$.

Proof According to (7), when $0 \le t \le d$, the total number of mobile agents in the network can be estimated as follows

$$\sum_{j=1}^{n} p_j(t) = \|\overrightarrow{p}(t)\|_1 = \left\|\sum_{i=0}^{t-1} B^i kr(I + A)\overrightarrow{e}\right\|_1$$

$$\leq nkr \cdot \|I + A\|_1 \cdot \sum_{i=0}^{t-1} (\|B\|_1)^i .$$

From the definitions of matrices A and B, it is easy to see that $\|I + A\|_1 = 1 - \frac{1}{n} - \frac{1}{\sigma_1+1}$ and $\|B\|_1 = 1 - \frac{\sigma_n - 1}{n}$. Substitute these two values into the above expression, we have

$$\sum_{j=1}^{n} p_j(t) \leq \frac{(n\sigma_1 - \sigma_1 - 1)nkr}{(\sigma_1 + 1)(\sigma_n - 1)} \left[1 - \left(1 - \frac{\sigma_n - 1}{n} \right)^t \right]$$
$$= \alpha(\sigma_1, \sigma_n, t)nkr,$$

where $\alpha(\sigma_1, \sigma_n, t) = \frac{n\sigma_1 - \sigma_1 - 1}{(\sigma_1+1)(\sigma_n-1)}[1 - (1 - \frac{\sigma_n-1}{n})^t]$.[1] When $t \geq d$, from (7), we have

$$\vec{p}(t) = \sum_{i=0}^{d-1} B^i kr(I + A)\vec{e} + \sum_{i=d}^{t-1} B^i kr A \vec{e} .$$

Due to the fact that

$$a_{ji} = \frac{1}{d_i + 1} \left(1 - \frac{d_i + 1}{n} \right) - \frac{1}{d_i} \left(1 - \frac{d_i - 1}{n} \right) < 0$$

when $j \in NB(i)$ and $a_{ji} = 0$ otherwise, we have

$$\|\vec{p}(t)\|_1 \leq \left\| \sum_{i=0}^{d-1} B^i kr(I + A)\vec{e} \right\|_1$$
$$\leq nkr \cdot \|I + A\|_1 \cdot \sum_{i=0}^{d-1} (\|B\|_1)^i$$
$$\leq \alpha(\sigma_1, \sigma_n, d)nkr.$$

□

For the number of agents running on each node, we have the following theorem.

Theorem 2 *The number of agents running on the j-th node, denoted by $p_j(t)$, is no more than* $\frac{nd_j kr}{(\sigma_n - 1)^2 + 1}$.

Proof See Appendix. □

We now estimate the number of agents in each link by the following theorem.

[1] $\alpha(\sigma_1, \sigma_n, t)$is analyzed in Sect. 6.

Theorem 3 *The number of agents moving out from the j-th node to each of its neighboring nodes at time t, denoted by $f_j(t)$, satisfies*:

$$f_j(t) = \left(1 - \frac{d_j - 1}{n}\right)\frac{p_j(t)}{d_j} \leq \frac{n - \sigma_n}{n - \sigma_n + \sigma_n^2} kr. \tag{8}$$

Proof This theorem is proved by induction as follows. Based on (7), we have

$$\begin{aligned} p_j(0) &= 0, \\ p_j(1) &= kr, \\ &\vdots \\ p_j(t) &= kr + \sum_{i \in NB(j)} \left(1 - \frac{d_i - 1}{n}\right)\frac{p_i(t-1)}{d_i}. \end{aligned}$$

Then by the definition of $f_j(t)$, we have:

$$\begin{aligned} f_j(0) &= 0, \\ f_j(1) &= \frac{1}{d_j}\left(1 - \frac{d_j - 1}{n}\right) kr \leq \frac{n - \sigma_n + 1}{\sigma_n(\sigma_n - 1)} kr. \end{aligned}$$

If the theorem holds for time $t - 1$, then at time t, we have

$$\begin{aligned} f_j(t) &\leq \frac{1}{d_j}\left(1 - \frac{d_j - 1}{n}\right)\left[kr + \sum_{i \in NB(j)} f_i(t-1)\right] \\ &\leq \frac{1}{d_j}\left(1 - \frac{d_j - 1}{n}\right)\left[kr + \sum_{i \in NB(j)} \frac{n - \sigma_n + 1}{\sigma_n(\sigma_n - 1)} kr\right] \\ &\leq \frac{1}{\sigma_n}\left(1 - \frac{\sigma_n - 1}{n}\right) kr + \left(1 - \frac{\sigma_n - 1}{n}\right)\frac{n - \sigma_n + 1}{\sigma_n(\sigma_n - 1)} kr \\ &= \frac{kr}{\sigma_n}\left[\left(1 - \frac{\sigma_n - 1}{n}\right)\left(1 + \frac{n - \sigma_n + 1}{\sigma_n - 1}\right)\right] kr \\ &= \frac{kr}{\sigma_n}\left[\frac{n - \sigma_n + 1}{n} \cdot \frac{n}{\sigma_n - 1}\right] \\ &= \frac{n - \sigma_n + 1}{\sigma_n(\sigma_n - 1)} kr. \end{aligned}$$

□

4.2 Nonsettleable case

In this case, the corresponding probabilities are

$$\Pr^{(1)}\{i \to j\} = \frac{1}{d_i}\left(1 - \frac{d_i + 1}{n}\right), \tag{9}$$

$$\Pr^{(2)}\{i \to j\} = \frac{1}{d_i - 1}\left(1 - \frac{d_i - 1}{n}\right), \tag{10}$$

and the corresponding coefficient matrix in (1) are $A = C[(1 - \frac{1}{n})D^{-1} - (D - I)^{-1}]$ (where $a_{ji} \leq 0$) and $B = C[(1 + \frac{1}{n})(D - I)^{-1} - \frac{I}{n}]$. Furthermore, based on the definition of matrix 1-norm, we have $\|A\|_1 = \frac{1}{n} + \frac{1}{\sigma_n - 1}$, $\|A + I\|_1 = 1 - \frac{1}{n} - \frac{1}{\sigma_1 - 1}$, and $\|B\|_1 = 1 + \frac{1}{n} - \frac{\sigma_n}{n}$. Similar to that in settleable case, we have the following theorem.

Theorem 4 *If* $\overrightarrow{p}(0) = 0$, *the total number of mobile agents running in the network is no more than* $\beta(\sigma_1, \sigma_n, d)nkr$ *where* $\beta(\sigma_1, \sigma_n, d) = \frac{n(\sigma_1 - 2) - (\sigma_1 - 1)}{(\sigma_1 - 1)(\sigma_n - 1)}[1 - (1 + \frac{1}{n} - \frac{\sigma_n}{n})^d]$.[2]

Regarding to the number of agents running at one node, we have the following theorem.

Theorem 5 *The number of agents running on the* j*-th node is no more than* $kr + \frac{nd_j kr}{(\sigma_n - 1)^2}$.

Proof See Appendix. □

Regarding to the number of agents moving out from the j-th node at time t, $f_j(t)$, we have the following theorem.

Theorem 6 *The number of agents moving out from the* j*-th node at time* t, *denoted by* $f_j(t)$, *satisfies*:

$$f_j(t) \leq \frac{n - \sigma_n + 1}{(\sigma_n - 1)^2} kr. \tag{11}$$

Proof By the definition of $f_j(t)$, we have

$$\begin{aligned} f_j(t) &= (1 - \frac{d_j - 1}{n})\frac{p_j(t-1)}{d_j - 1} \\ &= \left(1 - \frac{d_j - 1}{n}\right)\frac{1}{d_j - 1}\left[kr + \sum_{i \in NB(j)} f_i(t-1)\right] \end{aligned}$$

[2] $\beta(\sigma_1, \sigma_n, t)$is analyzed in Sect. 6.

and $f_j(0) = 0$ for $j = 1, \ldots, n$. Then by induction, the theorem can be easily proven. □

It can be seen that the number of agents in the settleable case is smaller than that in the nonsettleable case, but with a lower probability of success. Therefore, each case has its own strength and weakness. Thus, we can apply appropriate strategies for these different cases to control the number of mobile agents running in the network and improve network performance for differing application requirements and network characteristics.

5 Probability of success

Analysis of the probability of success is important because the probability of success directly affects the searching process, and influences the network performance as a result. In this section, the probabilities of success for both single agent and multiple agents are analyzed. Our results show that the probability of success is affected by the connectivity of the network, the life-span limits, and the number of mobile agents.

In a mobile agent-based network routing model, a mobile agent will visit a sequence of nodes. The sequence of nodes between the server and the destination is called the itinerary of the mobile agent. A static itinerary is entirely defined at the server and does not change during the agent's travels, whereas a dynamic itinerary is subject to modification by the agent itself. In this context, the mobile agents traverse the network with a dynamic itinerary. Suppose that the w-th node in the network is the destination. The sequence of nodes in the itinerary of an agent is denoted by $J^{(0)}, J^{(1)}, \ldots, w$ or $J^{(d)}$, where $J^{(0)}$ denotes the server node and $J^{(i)}$ is the i-th node visited by the agent.

Lemma 4 *Let $i \to j$ indicates the event that an agent jumps from the i-th node to the j-th node* (*in one jump*), *the probability of success that an agent can find its destination at the t-th jump*, *denoted by $p(t)$*, *satisfies*

$$p(t) = \sum_{l \in NB(J^{(t-1)})} \Pr\left\{J^{(t-1)} \to l\right\} \frac{d_l - 1}{n} \left[1 - \sum_{k=0}^{t-1} p(k)\right], \tag{12}$$

where $NB(j)$ is the set of node j's neighboring nodes and node j itself, $p(0) = (d_{J^{(0)}} + 1)/n$, and $0 < t \leq d$.

Proof After being generated by the server, $J^{(0)}$, mobile agents move out from the server and search for their destination in the network. The probability that an agent can find its destination at birth, $p(0)$, i.e., the probability that the destination is in $NB(J^{(0)})$, equals to $(d_{J^{(0)}} + 1)/n$. If the agent cannot find its destination in $NB(J^{(0)})$, it will jump to one of the server's neighboring node, $J^{(1)}$, and continue searching. Since both $J^{(0)}$ and $J^{(1)}$ are not the destination, the probability that the agent can find its destination at the first jump equals to $p(1) = \sum_{i \in NB(J^{(0)})} \frac{d_i - 1}{n} \Pr\{J^{(0)} \to i\}[1 - p(0)]$. If the agent cannot find its destination in

$NB(J^{(1)})$, it will jump to a neighboring node, $J^{(2)}$, and continue searching. The probability that an agent can find its destination at the second jump is $p(2) = \sum_{j \in NB(J^{(1)})} \frac{d_j - 1}{n} \Pr\{J^{(1)} \to j\}[1 - p(0) - p(1)]$. By recursion, it is easy to prove that the probability, $p(t)$, that an agent can find its destination at the t-th jump for $t \geq 1$ satisfies:

$$p(t) = \sum_{l \in NB(J^{(t-1)})} \frac{d_l - 1}{n} \Pr\{J^{(t-1)} \to l\} \left[1 - \sum_{k=0}^{t-1} p(k)\right]. \qquad \square$$

It can be seen that the probability $p(t)$ is relevant to the connectivity of the network and the number of jumps. From Lemma 1, it is easy to estimate the probability of success that an agent can find its destination in d jumps, denoted by $P(d)$, by the relationship

$$\begin{aligned} P(d) &= p(0) + \sum_{t=1}^{d} p(t) \\ &= \frac{d_{J^{(0)}} + 1}{n} + \sum_{t=1}^{d} \sum_{l \in NB(J^{(t-1)})} \\ &\quad \times \Pr\{J^{(t-1)} \to l\} \cdot \frac{d_l - 1}{n} \left[1 - \sum_{k=0}^{t-1} p(k)\right]. \end{aligned} \tag{13}$$

From Lemma 4, it can be seen that $p(t)$ is a decreasing function on t. Therefore, based on (13), the increase of $P(d)$ is slower when d becomes larger. Obviously, if the increase of $P(d)$ is too small, it is not necessary to go on searching. Thus, the life-span limit of mobile agents, d, can be defined by a given threshold $\epsilon \geq 0$ such that $P(d) - P(d-1) \leq \epsilon$. Based on (12) and (13), we have the following theorem:

Theorem 7 *The probability of success, P_s, that at least s agents from k agents can find the destination in d jumps satisfies the following inequality*:

$$\frac{\triangle^s}{\sqrt{2\pi s}} < P_s < \frac{\triangle^s}{\sqrt{2\pi s}} \cdot \frac{1}{1 - \lambda/(s+1)}, \tag{14}$$

where $\lambda = kP(d)$ and $\triangle = \frac{\lambda}{s} e^{1-\lambda/s}$.

Proof As we know, this probability satisfies binomial distribution, so we have

$$P_s = \sum_{i=s}^{k} C_k^s P(d)^s [1 - P(d)]^{k-s}.$$

When k is big enough, this distribution can be approximated by Poisson distribution as follows

$$P_s \approx \sum_{i=s}^{k} \frac{\lambda^s e^{-\lambda}}{s!}.$$

where $\lambda = kP(d)$. Therefore, we have

$$\begin{aligned}
e^{-\lambda}\frac{\lambda^s}{s!} &< \sum_{i=s}^{k} \frac{\lambda^s e^{-\lambda}}{s!} \\
&= e^{-\lambda}\frac{\lambda^s}{s!} \cdot \left[1 + \frac{\lambda}{s+1} + \frac{\lambda^2}{(s+1)(s+2)}\right] \\
&\quad + \cdots + e^{-\lambda}\frac{\lambda^s}{s!} \cdot \left[\frac{\lambda^{k-s}}{(s+1)\cdots k}\right] \\
&\leq e^{-\lambda}\frac{\lambda^s}{s!} \cdot \left[1 + \frac{\lambda}{s+1} + \left(\frac{\lambda}{s+1}\right)^2\right] \\
&\quad + \cdots + e^{-\lambda}\frac{\lambda^s}{s!} \cdot \left[\left(\frac{\lambda}{s+1}\right)^{k-s}\right] \\
&\leq e^{-\lambda}\frac{\lambda^s}{s!} \cdot \frac{1}{1-\lambda/(s+1)}.
\end{aligned}$$

Applying Stirling's formula

$$n! \approx \sqrt{2\pi n}n^n e^{-n} \text{for large } n,$$

we have

$$\frac{\triangle^s}{\sqrt{2\pi s}} < P_s < \frac{\triangle^s}{\sqrt{2\pi s}} \frac{1}{1-\lambda/(s+1)}.$$

□

In particular, P_1, the probability that at least one agent among k agents can find the destination, is $1 - [1 - P(d)]^k$.

5.1 Settleable case

In the settleable case, both the host node and its neighboring nodes have the same possibility to be selected since each direction is equally likely to connect with the destination. Thus, the probability $\Pr\{i \to j\}$ equals to $1/(d_i + 1)$ where $j \in NB(i)$ and d_i is the number of neighboring nodes of node i. Therefore, the probability $p(t)$ in Lemma 4 satisfies

$$p(t) = \sum_{l \in NB(J^{(t-1)})} \frac{d_l - 1}{n} \cdot \frac{1}{d_{J^{(t-1)}} + 1} \cdot \left[1 - \sum_{k=0}^{t-1} p(k)\right], \tag{15}$$

where $p(0) = (d_{J^{(0)}} + 1)/n$ and $0 < t \le d$. In particular, when the average connectivity of the network, denoted by θ, is available, we can further estimate $p(t)$ based on (15) as follows:

$$\begin{aligned} p(t) &= \sum_{l \in NB(J^{(t-1)})} \frac{\theta - 1}{n(\theta + 1)} \cdot \left[1 - \sum_{k=0}^{t-1} p(k)\right], \\ &= \frac{\theta(\theta - 1)}{n(\theta + 1)} \cdot \left[1 - \sum_{k=0}^{t-1} p(k)\right], \end{aligned} \tag{16}$$

$$p(t-1) = \frac{\theta(\theta - 1)}{n(\theta + 1)} \cdot \left[1 - \sum_{k=0}^{t-2} p(k)\right], \tag{17}$$

where $\theta = E[d_i]$. Denote $\theta(\theta - 1)/[n(\theta + 1)]$ by a, then we have

$$p(t) - p(t-1) = a \cdot [-p(t-1)]. \tag{18}$$

Therefore,

$$\begin{aligned} p(t) &= (1-a)p(t-1) = (1-a)^{t-1} p(1) \\ &= a(1-a)^{t-1}\left(1 - \frac{\theta + 1}{n}\right), \end{aligned} \tag{19}$$

since $p(1) = a[1 - p(0)]$ and $p(0) = (\theta + 1)/n$. Furthermore, the probability of success, $P(d)$, that an agent can find its destination in d jumps satisfies

$$\begin{aligned} P(d) &= \sum_{t=0}^{d} p(t) = p(0) + \sum_{t=1}^{d} p(t) \\ &= \frac{\theta + 1}{n} + \sum_{t=1}^{d} a(1-a)^{t-1}\left(1 - \frac{\theta + 1}{n}\right) \\ &= \frac{\theta + 1}{n} + a\left(1 - \frac{\theta + 1}{n}\right)\frac{1-(1-a)^d}{a} \\ &= 1 - \left(1 - \frac{\theta + 1}{n}\right)(1-a)^d. \end{aligned} \tag{20}$$

5.2 Nonsettleable case

In the nonsettleable case, the probability $\Pr\{J^{(t)} \to l\}$ equals to $1/d_{J^{(t)}}$ since all neighboring nodes have the same possibility to be selected. Substituting this value in (12), the probability of success that an agent can find its destination at the t-th jump can be estimated as follows:

$$p(t) = \sum_{l \in NB(J^{(t-1)})} \frac{d_l - 1}{n \cdot d_{J^{(t-1)}}} \left[1 - \sum_{k=0}^{t-1} p(k)\right], \tag{21}$$

where $p(0) = (d_{J(0)} + 1)/n$ and $0 < t \le d$. Similar to the analysis for the settleable case, when the average connectivity θ is available, the probability can be estimated as follows

$$p(t) = b(1-b)^{t-1}\left(1 - \frac{\theta+1}{n}\right), \tag{22}$$

where $b = (\theta - 1)/n$. The probability of success, $P(d)$, that an agent can find its destination in d jumps satisfies

$$P(d) = 1 - (1-b)^{d}\left(1 - \frac{\theta+1}{n}\right). \tag{23}$$

6 Simulation studies

In this section, we describe the simulation results for performance evaluation. We have performed extensive simulation experiments to evaluate the performance of our model and compare the results with those of existing models. The network topology was randomly generated by the Tier program [8]. We have conducted experiments for many topologies with different parameters and found that the performance of our model was relatively insensitive to topology changes. Here, we list only the experimental results for one topology of a network with 10, 000 nodes and a small network with 57 nodes (the Japanese backbone).

In Fig. 3, we plot $\alpha(\sigma_1, \sigma_n, t)$ and $\beta(\sigma_1, \sigma_n, t)$ as functions on σ_1 and σ_n, respectively. It can be seen that both of them are increasing functions on σ_1 and decreasing functions on σ_n. $\beta(\sigma_1, \sigma_n, t)$ is greater than $\alpha(\sigma_1, \sigma_n, t)$ as a function on the connectivity of the network.

In Fig. 4, we plotted results of the number of mobile agents running on each node for $n = 10{,}000$, $r = 100$, and $k = 50$. The figures on the upper row show the number

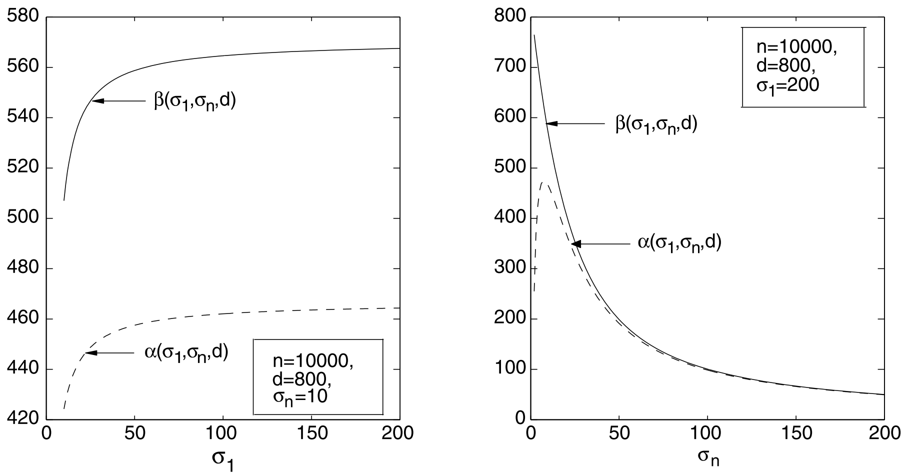

Fig. 3 The changes of $\alpha(\sigma_1, \sigma_n, d)$ and $\beta(\sigma_1, \sigma_n, d)$

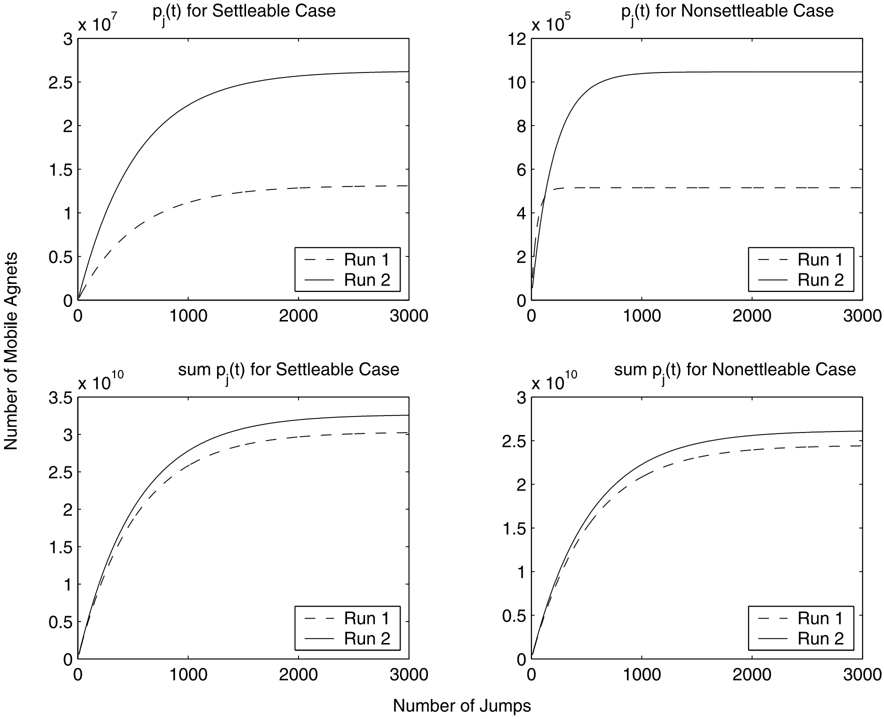

Fig. 4 Variations of $p_j(t)$ with n

of mobile agents running on a node at time t, $p_j(t)$, and the figures on the lower row show the number of mobile agents running in the network at time t, $\sum_{j=1}^{n} p_j(t)$. The figures on the left column are for the number of mobile agents for the settleable case, and the figures on the right column are for the number of mobile agents for the nonsettleable case. In each graph, there are two curves, which indicate the following two runs, respectively:

- Run 1: $\sigma_1 = 50$, $\sigma_n = 20$;
- Run 2: $\sigma_1 = 200$, $\sigma_n = 20$.

From the figures on the upper row we can see that $p_j(t)$ is a monotonically increasing function on t. For both cases, $p_j(t)$ increases more quickly for large connectivity. We can also see that $p_j(t)$ for the settleable case is less than that for the nonsettleable case, which conforms to our analysis proposed previously for each case. From the figures on the lower row we can see that $\sum_{j=1}^{n} p_j(t)$ is a monotonically increasing function on t. It can also be seen that for a network which has larger connectivity, the number of mobile agents is larger. $\sum_{j=1}^{n} p(t)$ for settleable case is less than that for nonsettleable case, which conforms to our analysis for each case. We also compare the number of agents between settleable and nonsettleable cases in Table 1.

The experimental results for the probabilities of success are depicted in Fig. 5. The figures in the upper row are for the probability of success that an agent can find its

Table 1 The comparison of the number of agents between settleable case and nonsettleable case

			$d = 100$	$d = 500$	$d = 1000$	$d = 2000$	$d = 3000$
Run 1	$p_j(t)$	settleable	4.0911e+005	9.5692e+005	1.0386e+006	1.0462e+006	1.0462e+006
		nonsettleable	2.2431e+006	8.0491e+006	1.1175e+007	1.2849e+007	1.3099e+007
	$\sum_{j=1}^{n} p_j(t)$	settleable	4.2410e+009	1.5026e+010	2.0831e+010	2.3941e+010	2.4406e+010
		nonsettleable	5.2550e+009	1.8618e+010	2.5812e+010	2.9666e+010	3.0241e+010
Run 2	$p_j(t)$	settleable	4.4737e+005	5.1502e+005	5.1504e+005	5.1504e+005	5.1504e+005
		nonsettleable	4.4711e+006	1.6083e+007	2.2334e+007	2.2583e+007	2.6183e+007
	$\sum_{j=1}^{n} p_j(t)$	settleable	4.5343e+009	1.6065e+010	2.2272e+010	2.5597e+010	2.6094e+010
		nonsettleable	5.6592e+009	2.005e+010	2.7798e+010	3.1948e+010	3.2568e+010

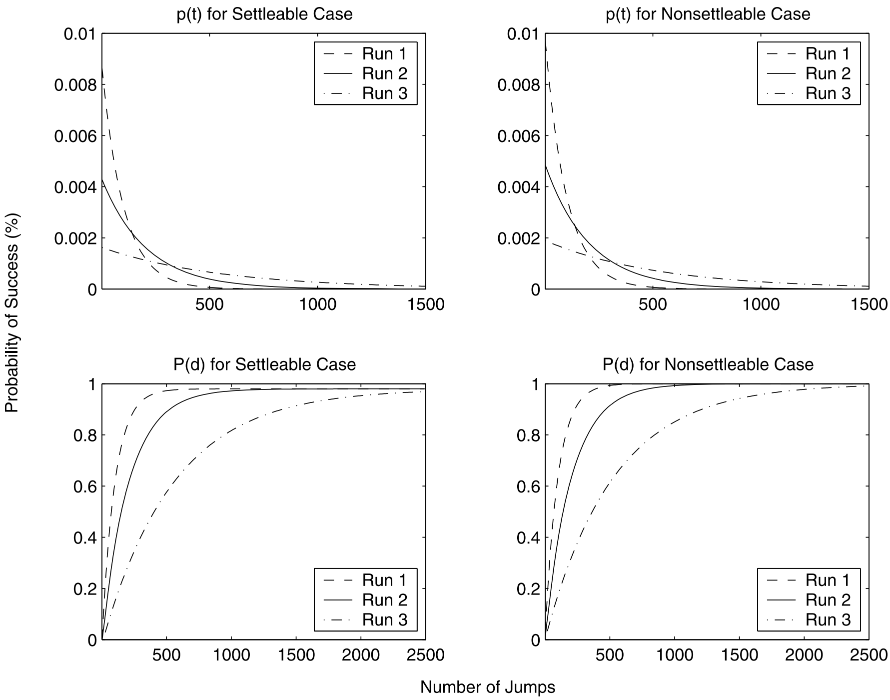

Fig. 5 Probabilities of success

destination at the t-th jump, $p(t)$, and those in the lower row are for the probability of success that an agent can find its destination in t jumps, $P(t)$. The figures in the left column are the probabilities of success in the settleable case, and those in the right are for probabilities of success in the nonsettleable case. In each figure, there are three curves, which indicate the following three runs, respectively:

Table 2 The comparison of the probability of success between settleable case and nonsettleable case

			$t = 100$	$t = 200$	$t = 500$	$t = 1500$	$t = 2500$
Run 1	$p(t)$	settleable	0.0033	0.0012	6.4040e−005	3.3768e−009	1.7806e−013
		nonsettleable	0.0037	0.0014	6.8403e−005	3.2669e−009	1.5603e−013
	$P(t)$	settleable	0.6067	0.8406	0.9727	0.9800	0.9800
		nonsettleable	0.6340	0.8647	0.9932	1.0000	1.0000
Run 2	$p(t)$	settleable	0.0027	0.0016	3.8908e−004	3.1528e−006	2.5548e−008
		nonsettleable	0.0030	0.0018	4.2022e−004	3.0917e−006	2.2747e−008
	$P(t)$	settleable	0.3624	0.5984	0.8900	0.9793	0.9800
		nonsettleable	0.3882	0.6257	0.9142	0.9994	1.0000
Run 3	$p(t)$	settleable	0.0014	0.0011	6.5825e−004	1.0760e−004	1.7589e−005
		nonsettleable	0.0016	0.0013	7.3400e−004	1.0958e−004	1.6361e−005
	$P(t)$	settleable	0.1459	0.2841	0.5758	0.9139	0.9692
		nonsettleable	0.1734	0.3165	0.6137	0.9423	0.9914

- Run 1: $n = 10{,}000$, $\theta = 100$;
- Run 2: $n = 10{,}000$, $\theta = 50$;
- Run 3: $n = 10{,}000$, $\theta = 20$.

From the figures in the upper row, we can see that $p(t)$ is a monotonically decreasing function of t and decreases more quickly for large θ. We can also see that $p(t)$ for the nonsettleable case is larger than that for the settleable case, which conforms to the analysis presented previously for each case. From the figures on the lower row, we can see that $P(t)$ is a monotonically increasing function of t. It can also be seen that for a network which has larger θ, a higher probability of success, $P(t)$, can be gained in a shorter time. $P(t)$ for the nonsettleable case is larger than that for the settleable case, which conforms to our analysis for each case. We also compare the probability of success between settleable case and nonsettleable case in Table 2.

Figures 7 and 6 compare our model to the one proposed in [31] ("blind case" for short). We use the proposed model for settleable case in the comparison since it is almost the same as the "blind case" except mobile agents' ability to check neighboring nodes' information. We plotted results of the number of mobile agents running on each node for $n = 10{,}000$, $r = 100$, and $k = 50$. The topologies are generated by the Tier algorithm, which is wide accepted as reasonable. In Fig. 7, figures on the upper row show the number of mobile agents running on a node at time t, $p_j(t)$, and the figures on the lower row show the number of mobile agents running in the network at time t, $\sum_{j=1}^{n} p_j(t)$. The figures on the left column are for the number of mobile agents for $\sigma_1 = 50$, and the figures on the right column are for the number of mobile agents for $\sigma_1 = 200$. We can see that both $p_j(t)$ and $\sum_{j=1}^{n} p_j(t)$ are monotonically increasing functions on t. For both cases, we can also see that $p_j(t)$ for the settleable case is less than that for blind case. The reason is that mobile agents of settleable case have a higher probability of success at each step, therefore, a higher death rate on each node.

The experimental results for the probabilities of success are depicted in Fig. 6. The figures in the left column are for the probability of success that an agent can find its

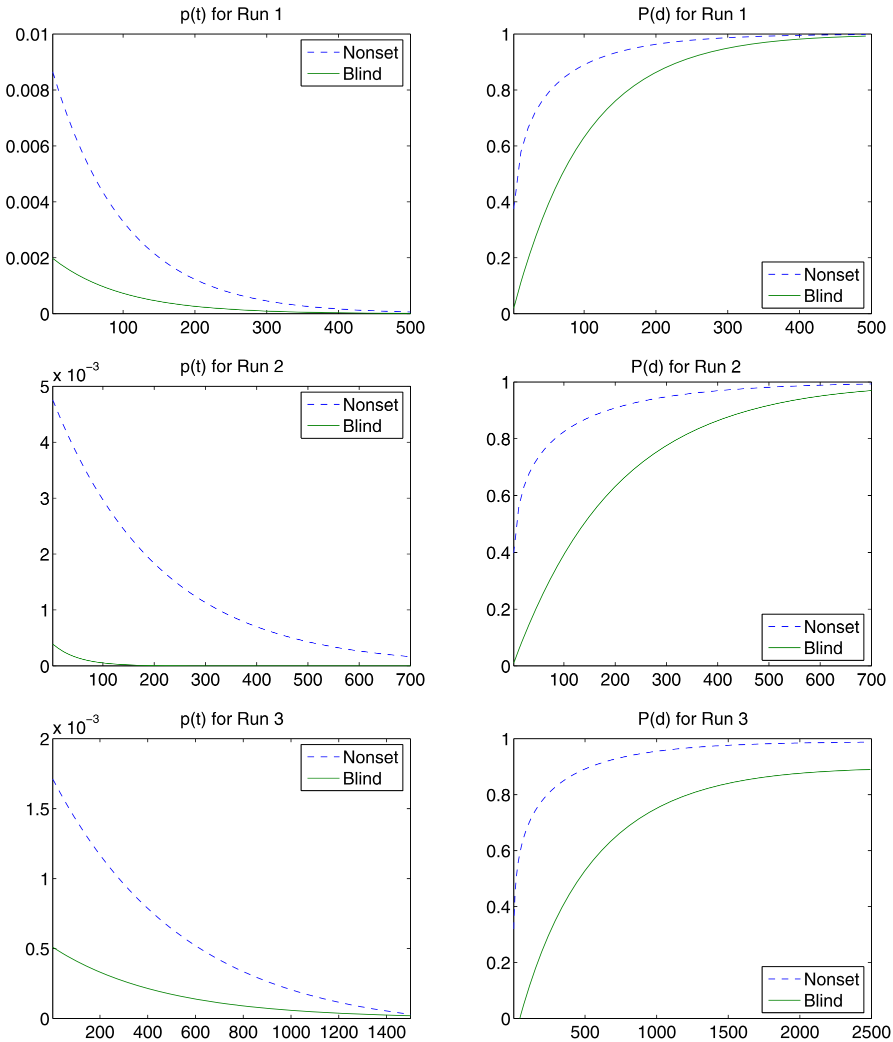

Fig. 6 The probability of success

destination at the t-th jump, $p(t)$, and those in the right column are for the probability of success that an agent can find its destination in t jumps, $P(t)$. The figures in the first row are the probabilities of success for case Run 1:$\theta = 100$, those in the second row are for Run 2:$\theta = 50$, and those in the third row are for Run 3:$\theta = 20$, which are described below:

It can be seen that $p(t)$ is a monotonically decreasing function on t and decreases more quickly for large θ. It can also be seen that $p(t)$ for the settleable case is larger than that for the blind case. This is because that mobile agents in the settleable case can check more candidate nodes at each step, which illustrate the rationality of our

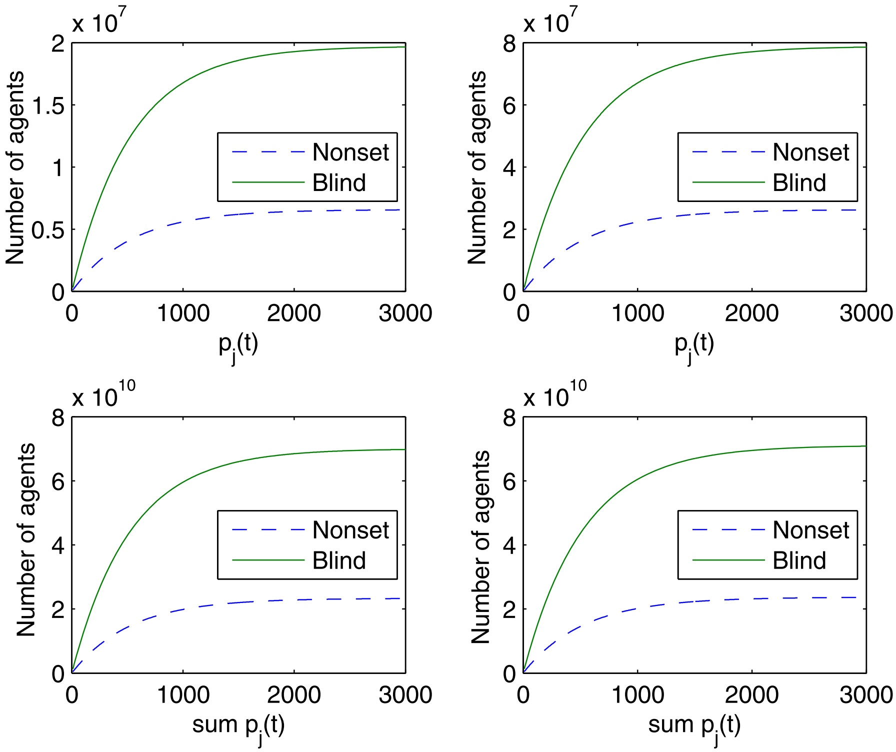

Fig. 7 The population distribution

assumption. Figures on the right column show that $P(t)$ is a monotonically increasing function on t and $P(t)$ for the settleable case is larger than that for the blind case.

We have also implemented some of our results on the Japanese backbone, NTTNET, as a small network scenario in the OMNeT++ simulator [37]. The network is composed of 57 nodes and 162 bidirectional links. The topology is showed in Fig. 8. The bandwidth is set to be 6 Mbits/sec for every link and the propagation delay is from 2 to 5 milliseconds. The assigned link or node fault probability is null, local buffers have a gigabyte of capacity and links are accessed through statistical multiplexing. The size of a mobile agent is set to be 24 bytes. In the simulation, 20 nodes in the network are randomly selected as the sink nodes of requests and 20 destinations are also randomly selected accordingly. The number of agents generated per request is 5, and the life-span limit of a mobile agent is set to be 30 hops.

Figure 9 reports experimental results for the population distribution and probability of success. The time length of the simulation has been set to 200 seconds. We observed that after this time interval the behavior of the algorithm is already well characterized. The left figure described the results for the population distribution. Reported data are averaged over 10 trials. From the left figure, we can see that the number of agents of nonsettleable case is lower than that of settleable case, which coincides with our theoretical results. The right figure shows the results for probability of success. Reported data are averaged over 50 trials. We observe the number of

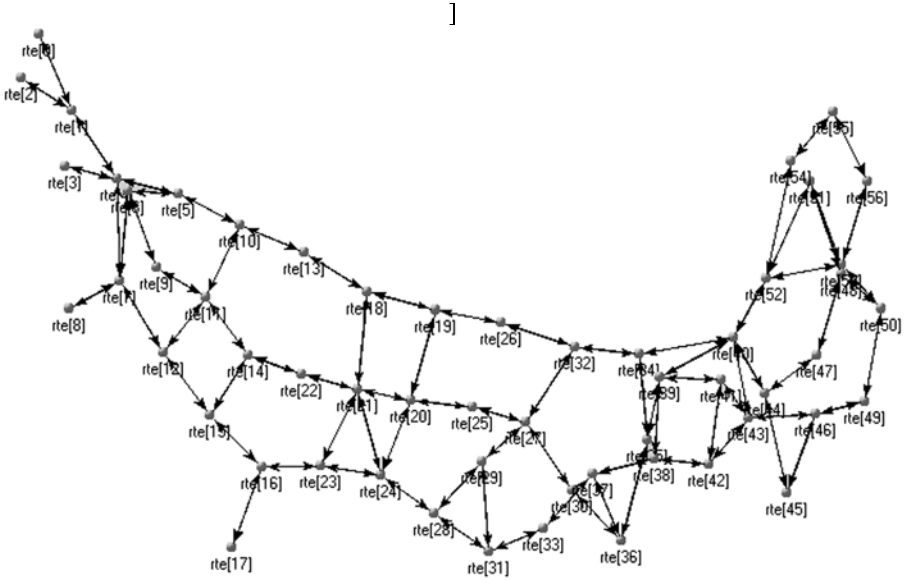

Fig. 8 NTT backbone network

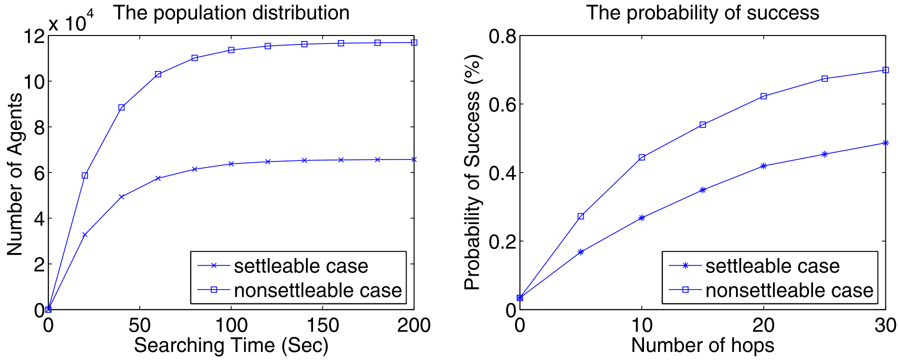

Fig. 9 Simulation results for NTT backbone network

agents that find their destinations. The probability of success is calculated as the ratio between the number of agents that have found its destination and the total number of launched agents. From the right figure, we can see that the probability of success of nonsettleable case is higher than that of settleable case as we verified previously.

7 Conclusion

In this paper, we described a general agent-based routing model in which the neighborhood information of each node is known in the node's routing table. We further classify the model into settleable and nonsettleable cases according to whether an agent is allowed to remain in the current host node during the next step of searching. For each case, we analyzed both the population distribution and the probability of success of mobile agents. We also conducted extensive simulation studies to validate our theoretical results. Both the theoretical and experimental results showed that the behaviors of mobile agents can be characterized by the population of mobile agents and the probability of success, and both these parameters can be controlled by tuning the number of agents generated per request and the number of jumps each mobile agent can make. Our results reveal new theoretical insights into the statistical behaviors of mobile agents and provide useful tools for effectively managing mobile agents in large networks.

Acknowledgements This research is conducted as a program for the National Natural Science Foundation of China (Grant No. 90718030).

Appendix: proof of Theorem 2 and Theorem 5

Assume that $\overrightarrow{p}(0) = 0$, the number of mobile agents running on a node can be estimated as follows based on (1) and the fact $a_{ji} \leq 0$:

- When $0 \leq t \leq d$,

$$\begin{aligned}\overrightarrow{p}(t) &= \sum_{i=0}^{t-1} B^i kr(I + A)\overrightarrow{e} + B^t \overrightarrow{p}(0) \\ &\leq \sum_{i=0}^{t-1} B^i kr\overrightarrow{e} = kr\overrightarrow{e} + B\overrightarrow{p}(t-1).\end{aligned}$$

- When $t \geq d$,

$$\begin{aligned}\overrightarrow{p}(t) &= \sum_{i=0}^{t-1} B^i kr(I + A)\overrightarrow{e} + B^t \overrightarrow{p}(0) - \sum_{j=d}^{t-1} B^j kr\overrightarrow{e} \\ &= \sum_{i=0}^{d-1} B^i kr(I + A)\overrightarrow{e} + B^t \overrightarrow{p}(0) + \sum_{j=0}^{t-1} Akr\overrightarrow{e} \\ &\leq \sum_{i=0}^{d-1} B^i kr\overrightarrow{e}.\end{aligned}$$

From the analysis, it is easy to see that

$$p_j(t) \leq kr + \sum_{i \in NB(j)} \Pr^{(2)}\{i \to j\} p_j(t-1). \tag{24}$$

In the following, we will present analysis on the number of mobile agents running on a node for each case, respectively.

Proof of Theorem 2

Proof Substitute the value of $\Pr^{(2)}\{i \to j\}$ for settleable case in inequality (24), we have

$$p_j(t) \le kr + \sum_{i \in NB(j)} \frac{1}{d_i}\left(1 - \frac{d_i - 1}{n}\right) p_j(t-1).$$

Since $p_j(0) = 0$ and $p_j(t) = 1$, we have

$$\begin{aligned} p_j(t) &\le (d_j + 1)kr \\ &\quad - \left[\frac{\sigma_n - 1}{n} - \frac{\sigma_n - 1}{\sigma_n}\left(1 - \frac{\sigma_n - 1}{n}\right)\right] d_j kr \\ &= (d_j + 1)kr - \mu d_j kr, \end{aligned}$$

where $\mu = \frac{\sigma_n - 1}{n} + \frac{\sigma_n - 1}{\sigma_n}(1 - \frac{\sigma_n - 1}{n})$. If the following inequality is held,

$$\begin{aligned} p_j(t) \le (d_j + 1)kr - \left[\frac{\sigma_n - 1}{n} - \frac{1}{\sigma_n}\left(1 - \frac{\sigma_n - 1}{n}\right)\right] \\ \times \sum_{i=0}^{t-2}\left(1 - \frac{\sigma_n - 1}{n}\right)^i d_j kr - \mu\left(1 - \frac{\sigma_n - 1}{n}\right)^{t-1} d_j kr. \end{aligned}$$

Then it can be easily proved that the inequality will also hold for $t + 1$. Therefore, we have

$$\begin{aligned} p_j(t) &\le (d_j + 1)kr - \mu\left(1 - \frac{\sigma_n - 1}{n}\right)^{t-1} d_j kr \\ &\quad - \frac{\sigma_n^2 - n - 1}{n\sigma_n} \sum_{i=0}^{t-2}\left(1 - \frac{\sigma_n - 1}{n}\right)^i d_j kr \\ &= (d_j + 1)kr - \frac{\sigma_n^2 - n - 1}{\sigma_n(\sigma_n - 1)} d_j kr \\ &\quad + \left[\frac{\sigma_n^2 - n - 1}{\sigma_n(\sigma_n - 1)} - \mu\right]\left(1 - \frac{\sigma_n - 1}{n}\right)^{t-1} d_j kr \\ &= kr + \xi d_j kr + \zeta\left(1 - \frac{\sigma_n - 1}{n}\right)^{t-1} d_j kr, \end{aligned}$$

where $\xi = \frac{n - \sigma_n + 1}{\sigma_n(\sigma_n - 1)}$ and $\zeta = \frac{\sigma_n^2 - n - 1}{\sigma_n(\sigma_n - 1)} - \mu = \frac{2}{\sigma_n} - \frac{n^2 + (\sigma_n - 1)^2}{n\sigma_n(\sigma_n - 1)}$. Therefore, a convenient upper bound of the number of mobile agents is as follows

$$p_j(t) \le \left[\frac{n d_j}{(\sigma_n - 1)^2 + 1}\right] kr. \qquad \square$$

Proof of Theorem 5

Proof Similar to that in the proof of Theorem 2, substitute the value of $\Pr^{(2)}\{i \to j\}$ for unsettleable case in inequality (24), we have

$$p_j(t) \le kr + \sum_{i \in NB(j)} \frac{1}{d_i - 1}\left(1 - \frac{d_i - 1}{n}\right) p_j(t-1).$$

By induction, it can be proved that

$$\begin{aligned} p_j(t) &\le kr + \frac{n}{(\sigma_n - 1)^2} d_j kr \left[1 - \left(1 - \frac{\sigma_n - 1}{n}\right)^t\right] \\ &\le kr + \frac{n}{(\sigma_n - 1)^2} d_j kr. \end{aligned}$$

□

References

1. Baek JW, Yeo JH, Kim GT, Yeom HY (2001) Cost effective mobile agent planning for distributed information retrieval. In: Proc of the 21st int'l conf on distributed computing systems (ICDCS'01), pp 65–72, 2001
2. Baek JW, Kim GT, Yeom HY (2002) Cost-effective planning of timed mobile agents, In: Proc of the int'l conf information technology: coding and computing (ITCC'02), pp 536–541, 2002
3. Baek JW, Yeo JH, Yeom HY (2002) Agent chaining: an approach to dynamic mobile agent planning. In: Proc the 22nd int'l conf on distributed computing systems (ICDCS'02), pp 579–586, 2002
4. Beckers R, Deneubourg JL, Goss S (1992) Trails and U-turns in the selection of the shortest path by the ant lasius niger. Theor Biol (159): 397–415
5. Bergadano F, Puliafito A, Riccobene S, Ruffo G (1999) Java-based and secure learning agents for information retrieval in distributed systems. Inf Sci 113(1–2):55–84
6. Bieszczad A, Pagurek B, White T (1998) Mobile agents for network management. IEEE Commun Surv 1(1):2–9
7. Brewington B, Gray R, Moizumi K, Kotz D, Cybenko G, Rus D (1999) Mobile agents in distributed information retrieval. In: Klusch M (ed) Intelligent information agents: agents-based information discovery and management on the Internet. Springer, Berlin, pp 355–395, Chapter 15
8. Calvert KL, Doar MB, Zegura EW (1997) Modelling Internet topology. IEEE Commun Mag 35(6):160–163
9. Claessens J, Preneel B, Vandewalle J (2003) How can mobile agents do secure electronic transactions on untrusted hosts? A survey of the security issues and the current solutions. ACM Trans Internet Technol 3(1):28–48
10. Caro GD, Dorigo M (1998) AntNet: distributed stigmergetic control for communications networks. J Artif Intell Res 9:317–365
11. Caro GD, Dorigo M (1998) Mobile agents for adaptive routing. In: Proc of the 31st Hawaii int conf on system sciences, pp 74–83
12. Curran K, Woods D, McDermot N, Bradley C (2003) The effects of badly behaved routers on Internet congestion. Int J Netw Manag 13(1):83–94
13. Dorigo M, Maniezzo V, Colorni A (1996) The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B 26(1):29–41
14. Du TC, Li EY, Chang AP (2003) Mobile agents in distributed network management. Commun ACM 46(7):127–132
15. Garey M, Johnson D (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, New York
16. Goss S, Aron S, Deneubourg JL, Pasteels JM (1989) Self-organized shortcuts in the Argentine ant. Naturwissenschaften 79:579–581

17. He M, Jennings NR, Leung H (2003) On agent-mediated electronic commerce. IEEE Trans Know Data Eng 15(4):985–1003
18. Heck PS, Ghosh S (2000) A study of synthetic creativity through behavior modeling and simulation of an ant colony. IEEE Intell Syst 15(6):58–66
19. Hőlldobler B, Wilson EO (1990) The Ants. Springer, Berlin
20. Jayasimha DN, Schwiebert L, Manivannan D, May JA (2003) A foundation for designing deadlock-free routing algorithms in wormhole networks. J ACM 50(2):250–275
21. Kim SH, Robertazzi TG (2000) Mobile agent modeling. SUNY at Stony Brook Technical Report 786
22. Kotz D, Gray RW (1999) Mobile agents and the future of the Internet. ACM Oper Syst Rev 33(3):7–13
23. Karjoth G, Lange D, Oshima M (1997) A security model for aglets. IEEE Internet Comput 1(4):68–77
24. Lange D, Oshima M (1998) Programming and Developing Java Mobile Agents with Aglets. Addison-Wesley, Reading
25. Lange D, Oshima M (1999) Seven good reasons for mobile agents. Commun ACM 42:88–89
26. Lee Y, Tien JM (2002) Static and dynamic approaches to modeling end-to end routing in circuit-switched networks. IEEE/ACM Trans Netw 10(5):693–705
27. Lou W, Wu J (2002) On reducing broadcast redundancy in ad hoc wireless networks. IEEE Trans Mobile Comput 1(2):111–123
28. Maes P, Guttman RH, Moukas AG (1999) Agents that buy and sell. Commun ACM 42(3):81–91
29. Milojicic D (2000) Guest editor's introduction: agent systems and applications. IEEE Concurr 8(2):22–23
30. Pleisch S, Schiper A (2003) Fault-tolerant mobile agent execution. IEEE Trans Comput 52(2):209–222
31. Qu W, Shen H, Sum J (2003) New analysis on mobile agents based network routing. In: Proc of the 3rd int'l conf on hybrid intelligent systems (HIS'03), pp 769–778
32. Roughgarden T, Tardos E (2002) How bad is selfish routing? J ACM 49(2):236–259
33. Sheldon T, Linktionary. http://www.linktionary.com/
34. Stojmenovic I, Seddigh S, Zunic J (2002) Domination sets and neighbor elimination based broadcasting algorithms in wireless networks. IEEE Trans Parallel Distrib Syst 13(1):14–25
35. Sum J, Shen H, Leung C, Young G (2003) Analysis on a mobile agent-based ant algorithm for network routing and management. IEEE Trans Parallel Distrib Syst 14(3):193–202
36. Theilmann W, Rothermel K (2000) Optimizing the dissemination of mobile agents for distributed information filtering. IEEE Concurr, pp 53–61
37. Varga A, OMNeT++: Discrete Event Simulation System: User Manual. http://www.omnetpp.org/
38. Webopedia Online Computer Dictionary for Computer and Internet Terms and Definitions. http://www.webopedia.com/
39. White T, Pagurek B, Oppacher F, ASGA: improving the ant system by integration with genetic algorithms. In: Proc of the 3rd conf on genetic programming (GP/SGA'98), pp 610–617
40. Xu D, Yin J, Deng Y, Ding J (2003) A formal architectural model for logical agent mobility. IEEE Trans Softw Eng 29(1):31–45

Wenyu Qu is a researcher at the Center for Information Fusion at the Institute of Industrial Science of the University of Tokyo, Japan. She is also a Professor at the College of Computer Science and Technology, Dalian Maritime University, China. She got her bachelor and master degree both from Dalian University of Technology, China in 1994 and 1997, and her doctor degree from Japan Advanced Institute of Science and Technology in 2006. She was a lecturer in Dalian University of Technology from 1997 to 2003. Wenyu Qu's research interests include mobile agent-based technology, distributed computing, computer networks, and grid computing. Wenyu Qu has published more than 30 technical papers in international journals and conferences. She is on the committee board for a couple of international conferences.

Keqiu Li is a Professor at the School of Electronic and Information Engineering, Dalian University of Technology, China. He got his bachelor and master degree both from Dalian University of Technology, China in 1994 and 1997, and his doctor degree from Japan Advanced Institute of Science and Technology in 2005. He had also five-year experience in industry and university. Keqiu Li's research interests include Web technology, distributed computing, computer networks, grid computing, and multimedia application. Keqiu Li has published more than 40 technical papers in international journals and conferences, such as IEEE TPDS, ACM TOIT, and ACM TOMAPP. He is on the committee board for several internationals and serves as organization chair/program chair/publication chair/program committee member for a couple of international conferences. He is a member of IEEE.

Masaru Kitsuregawa received a Ph.D. degree from the University of Tokyo in 1983. He is currently a professor and a director of the Center for Information Fusion at the Institute of Industrial Science of the University of Tokyo. His current research interests include database engineering, Web mining, parallel computer architecture, parallel database processing/data mining, storage system architecture, digital earth, and transaction processing. He had been a VLDB Trustee and served as the general chair of ICDE 2005 at Tokyo. He is currently an Asian coordinator of the IEEE Technical Committee on Data Engineering, and a steering committee member of PAKDD and WAIM. In Japan, he chaired the data engineering technical group of IEICE, and served as ACM SIGMOD Japan Chapter Chair. He is currently an adviser to the Storage Networking Industry Association Japan and a director of the Database Society of Japan. He is a member of ACM and the IEEE Computer Society, and a fellow of IEICE and the Information Processing Society of Japan.

Weilian Xue received the B.S. degree and M.S. degree in computer science from the Liaoning Normal University in 1989 and 2003, respectively. She is currently working toward the Ph.D. degree in Dalian University of Technology. Her research interests are in sensor network and artificial immune system, information management, and information system.