Analysis on a Mobile Agent-Based Algorithm for Network Routing and Management

John Sum, Hong Shen, Chi-sing Leung, and G. Young

Abstract—Ant routing is a method for network routing in the agent technology. Although its effectiveness and efficiency have been demonstrated and reported in the literature, its properties have not yet been well studied. This paper presents some preliminary analysis on an ant algorithm in regard to its population growing property and jumping behavior. For synchronous networks, three main results are shown. First, the expected number of agents in a node is shown to be no more than $(1+\max_i\{|\Omega_i|\})km$, where $|\Omega_i|$ is the number of neighboring hosts of the $i\text{th}$ host, k is the number of agents generated per request, and m is the average number of requests. Second, the expected number of jumps of an agent is shown to be no larger than $(1+\max_i\{|\Omega_i|\})$. Third, it is shown that for all $p \geq (1+\max_i\{|\Omega_i|\})km$, the probability of the number of agents in a node exceeding p is not greater than $\int_p^\infty \mathcal{P}(x)dx$, where $\mathcal{P}(x)$ is a normal distribution function with mean and variance given by Mean = $(1+\max_i\{|\Omega_i|\})km$, Var. = $2km(1+\max_i\{|\Omega_i|\})$ + $\frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{(1+2\max_i\{|\Omega_i|\})}$. The first two results are also valid for the case when the network is operated in asynchronous mode. All these results conclude that as long as the value $\max_i\{|\Omega_i|\}$ is known, the practitioner is able to design the algorithm parameters, such as the number of agents being created for each request, k, and the maximum allowable number of jumps of an agent, in order to meet the network constraint.

Index Terms—Mobile agents routing algorithms, networking routing, ant algorithm.

1 Introduction

Due to the rapid growth of network-centric programming [1], [2], [3], [4], [5], [6] and WWW applications [7], [8], [9], use of mobile agents is one of the new techniques evolved recently. A mobile agent is a program that acts on behalf of a user to perform intelligent decision-making tasks. It is capable of migrating autonomously from node to node in an information network. Its tasks are determined by the user specified agent applications, such as online shopping and distributed computation to real-time device control. Successful examples using mobile agents can be found in the new program paradigms, such as Aglets [10], [11], Voyager [12], Agent Tcl [13], Tacoma [14], Knowbots [15] and Telescript [16]. As the process required for the communication between the server agent and the user agent is established within one single host and does not involve the message exchange between the server machine and the user machine, one advantage of using agent programming is that the network traffic for the message exchange is greatly reduced [9]. Owing to the tremendous growth of the size of the Internet and the latest development of mobile computing [17], [18], [19], [20], [21], a low traffic management technology, such as mobile agent-based network management, will be increasingly demanded.

In recent years, many intelligent mobile agent-based network management techniques have been proposed and implemented [22], [23], [24], [25]. In fault diagnosis [23], the server first dispatches to the network and generates mobile agents. Those agents then wonder around the network and gather the information about the current status of the network. Once an agent has traversed back to the server, it hands in its summary reports. The report of an agent is given to the server only when the agent's trip is over, but not in the middle of the trip, during the course of the trip there are very few communications between the agent and the server. Thus, based on the concept of mobile agents, the network traffic generated by mobile agents is very light.

Network routing is another problem in network management [26]. Once a packet is required to be sent to a destination (point-to-point) or to multiple destinations (multicast), the router should recommend a good path (or even 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 [27], the searching for the optimal path in a faulty network or mobile network will be much more difficult. The ant routing algorithm [25], [28], [29], [30], [31], [32] is a recently proposed routing algorithm for use in these environments.[1] The idea is similar to the path searching process of an ant. Once a request for sending a message is received from a server, the server will generate a number of mobile agents like ants. Those agents will then move out from the server to search for the corresponding destination host. Once an agent has reached the destination, it traverses all the way back to the source host, the server, following the path searched and leaves marks (just like the pheromone) on the

- *J. Sum is with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Hong Kong. E-mail: csfsum@comp.polyu.edu.hk.*
- *H. Shen is with the Graduate School of Information Science, Japan Advanced Institue of Science and Technology, Tatsunokuchi, Ishikawa 923-1292, Japan. E-mail: shen@jaist.ac.jp.*
- *C.-s. Leung is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. E-mail: eeleungc@ee.cityu.edu.hk.*
- *G. Young is with the Department of Computer Science, Cal Poly Pomona, CA.*

Manuscript received 27 Apr. 2000; revised 7 Mar. 2002; accepted 14 May 2002.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 112017.

1. There are other related works having similar idea as ant-routing, such as randomized routing algorithms and hot-potato routing.

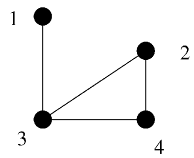

Fig. 1. A simple network structure.

hosts along the path. When all agents have been back, the source host will evaluate the costs of those paths collected and pick up the best path. If a connection is required, the server will send out an allocator agent to reserve resources from the hosts along the best path.

It can be seen that in either fault management or message routing, agents have to be generated frequently and dispatched to the network. Thus, they will certainly hold a certain amount of computation resources in each host machine in the network. If there are too many agents in the network, they will introduce too much computational overhead to host machines. Eventually, those host machines will be busy and indirectly block the network traffic. Therefore, the analysis on the agents' population and growing behavior is necessary and important for network management.

This paper shows that with a proper selection on the number of agents created for each request, the agents' population can be controlled in a probabilistic sense. In the next section, the network model and the ant routing algorithm will be discussed. Section 3 analyzes the ant population if the network is operated in synchronous mode. Guidelines on the control of the agent population will also be suggested in this section. The average number of hops will then be analyzed in Section 4. In Section 5, two results presented in Section 3 will be extended to the case when the network is operated in asynchronous mode. The conclusion will be presented in the last section.

2 Network Model and Ant Routing

Let $G = \{V, E\}$ be a graph corresponding to a fixed network, where V is the set of hosts and E is the edge set. The connectivity of the graph is described by a connection matrix C. For example, for a network shown in Fig. 1, its connection matrix C is given by

$$C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

In this paper, we assume that the topology of a network is a connected graph in order to ensure that communication is able to be made between any two host machines.

White et al. [28] proposed an adaptive mobile agents algorithm for network routing and connection management. The essential idea of the algorithm can be described as follows:

1. Once a point-to-point connection[2] request has been made, m ant agents (the *explorer* agents) are created and are sent out to the network.
2. Those explorer agents traverse the network from the source to destination. At each immediate node, each explorer agent will randomly select a neighbor node to move forward. For example, if an agent has moved from node A to node B, and both node C and node D are also connected to node B, the allowable moves will be either $A \to B \to C$ or $A \to B \to D$ but not $A \to B \to A$.
3. Once an explorer agent reaches the destination node, it will traverse backward to the source node and then will report its explored path to the source node.
4. In the source node, the host will compare all explored paths from return-back[3] explorer agents. If the cost of a traverse path of a return-back explorer agent is acceptable, then an *allocator* agent will be sent out immediately and allocate network resources on the nodes and links used in the path.
5. When the path is no longer required, a *deallocator* agent traverses the path and deallocates the network resources used on the nodes and links.

It should be noted that the performance of this ant routing algorithm is determined by the costs of paths being searched by those explorer agents. Besides, the forward-only move can make the explorer agent get stuck in any terminal node. Let us take Fig. 1 as an example. Suppose an explorer agent is sent out from Node-2 to Node-4. By chance, it moves to Node-3 in the first jump and then moves to Node-1 in the second jump. As the explorer agent can only move forward, it finally gets stuck in Node-1. In order to alleviate the problem of being-stuck, we modify the algorithm presented in [28] as follows:

Suppose that a request for sending message is received in the ith host at time t; the host will thus generate k ant agents. Each agent will randomly select one neighbor host with probability

$$P(j|i) = P(i \to j) = \frac{1}{|\Omega_i| + 1},$$

and move to this host. The parameter $|\Omega_i|$ is the total number of neighbor hosts of the ith host.[4] Suppose that when an ANT reaches the jth host, it will check whether the host is its destination. If so, it will turn back to the source host and report the path being searched if the jth host is its destination. Otherwise, the agent will randomly select one neighbor host of the jth host and move on. In the source machine, the server will pick up the path which is of minimum number of hops. Then, the message is sent along this path to the destination machine. At the same time, the server updates its routing table by including the information of this new path.

3 Agents Population: Synchronous Mode

Let $p_i(t)$ be the number of agents running in the ith host at time t and $r_i(t)$ be the number of requests initiated at time t

2. For point-to-multiple-point connection, the idea is essentially the same.

3. In White-Pagurek-Oppacher algorithm, if an explorer agent cannot reach the destination node within a predefined number of move, the agent will die automatically. Therefore, not all agent can return to the source node.

4. It should noted that probability of an agent that can jump is less than one, $\frac{|\Omega_i|}{|\Omega_i|+1}$. It is to ensure that the agent population does not increase to infinity.

in the ith host. Besides, we assume that the network is a connected graph. The dynamical change of the agents in the network can be given by the following stochastic equation:

$$p_j(t+1) = kr_j(t) + \sum_{i\in\Omega_j}\sum_{l=1}^{p_i(t)} \delta_{jil}(t), \quad (1)$$

where $\delta_{jil}(t)$ indicates the selection of the lth agent in the ith host and k is the number of agents generated per request, i.e.,

$$\delta_{jil}(t) = \begin{cases} 1 & \text{if the } l\text{th agent select the } j\text{th host to go} \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\sum_{j\in\Omega_i} \delta_{jil}(t) = 1.$$

This means that the lth agent selects only one neighbor $j \in \Omega_i$. In accordance with the ant routing algorithm,

$$P(\delta_{jil}(t) = 1) = P(j|i) = \frac{1}{|\Omega_i| + 1}$$

for all $l = 1, 2, \cdots, p_i(t)$.

Taking the expectation on both side of (1), it is readily shown that the evolution behavior of the ant routing follows Markov property as,

$$E\{p_j(t+1)|\vec{p}(t), \vec{r}(t)\} = kr_j(t) + \sum_{i\in\Omega_j} \frac{1}{|\Omega_i|+1} p_i(t),$$

where

$$\vec{p}(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ \cdots \\ p_N(t) \end{bmatrix} \quad \vec{r}(t) = \begin{bmatrix} r_1(t) \\ r_2(t) \\ \cdots \\ r_N(t) \end{bmatrix}.$$

In compact form, it can be rewritten as follows:

$$E\{\vec{p}(t+1)|\vec{p}(t), \vec{r}(t)\} = A\vec{p}(t) + k\vec{r}(t), \quad (2)$$

where $A = (a_{ji})_{N\times N}$ is an N by N matrix with the elements defined as follows:

$$a_{ji} = \begin{cases} \frac{1}{|\Omega_i|+1} & \text{if } j \in \Omega_i, j \neq i \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that a survival agent is an agent running after the current step of transition, a_{ji} is the expected delivery rate of a survival agent transmitted from i to $j \in \Omega_i$. By defining the following recurrent relation,

$$\hat{\vec{p}}(t+1) = E\{\vec{p}(t+1)|\hat{\vec{p}}(t), \hat{\vec{r}}(t)\},$$

the dynamical equation for the expected number of agents in the network is given by

$$\hat{\vec{p}}(t+1) = A\hat{\vec{p}}(t) + k\hat{\vec{r}}(t) \quad (3)$$
$$= A\hat{\vec{p}}(t) + km\vec{e}, \quad (4)$$

where $\vec{e} = (1, 1, \cdots, 1)^T$ and m is the average number of requests. To analyze the convergence behavior of the above equation, we need the following Lemma from Matrix Theory [33].

Lemma 1. *If λ is an eigenvalue of a matrix A and $\sigma(A)$ is the spectral radius of the matrix A, then $|\lambda| \leq \sigma(A)$.*

Since the largest column sum of A in (4) is strictly less than one, $|\lambda| < 1$.

$$\sigma(A) \leq \|A\|_\infty = \frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\} + 1},$$

where $\|A\|_\infty$ is the infinity norm of the matrix A, i.e., the largest row sum of matrix A. Referring to (4), we can express $\hat{\vec{p}}(t+1)$ in terms of A and $km\vec{e}$, $\hat{\vec{p}}(t+1) = \sum_{\tau=1}^{t} A^{t-\tau} km\vec{e}$, assuming that $\vec{p}(0) = 0$.

Considering the limiting case,

$$\lim_{t\to\infty} \hat{\vec{p}}(t) = km[I - A]^{-1}\vec{e}.$$

Note that $[I - A]$ is a diagonal dominant matrix. Hence, $[I - A]^{-1}\vec{e}$ is a positive vector and

$$\lim_{t\to\infty} p_i(t) \geq 0$$

for all $i = 1, \cdots, N$. Hence, the convergence result can thus be expressed by the following theorem.

Theorem 1. *Using the ant routing algorithm, the expected number of agents running in each host is less than or equal to $(1 + \max_i\{|\Omega_i|\})km$.*

Proof. Since, for any matrix A and vector y, $\|Ay\|_\infty \leq \|A\|_\infty \|y\|_\infty$. Here, $\|A\|_\infty$ is the largest row sum of the matrix while $\|y\|_\infty$ is the largest element in vector y. Let us denote $\vec{p}(\infty)$ be $\lim_{t\to\infty} \vec{p}(t)$,

$$\begin{aligned} \|\vec{p}(\infty)\|_\infty &\leq km\|[I - A]^{-1}\|_\infty \|\vec{e}\|_\infty \\ &\leq km(1 - \|A\|_\infty)^{-1} \\ &\leq (1 + \max_i\{|\Omega_i|\})km. \end{aligned}$$

The expected number of agents running in each host is then less than or equal to $(1 + \max_i\{|\Omega_i|\})km$, and the proof is completed. □

To derive the covariance matrix for the vector $\vec{p}(t)$, let us first consider the following cases when $t = 1$ and $t = 2$. For $t = 1$, we have

$$\vec{p}(1) = k\vec{r}(0)$$
$$E_{r0}\{\vec{p}(1)\} = km\vec{e},$$

since $\vec{p}(0) = 0$. Besides, the covariance of $\vec{p}(1)$ is given by

$$Q(0) = E_{r0}\{(\vec{p}(1) - E_{r0}\{\vec{p}(1)\})(\vec{p}(1) - E_{r0}\{\vec{p}(1)\})^T\},$$

which is equation to $(km)^2 I$. Here, the expectation, E_{r0}, is taken on the random variable $\vec{r}(0)$. Besides,

$$Q(0) = (km)^2 I. \quad (5)$$

For $t = 2$, we have

$$E_A\{\vec{p}(2)\} = A\vec{p}(1) + k\vec{r}(1),$$

where the expectation E_A is taken on the random variables $\delta_{jil}(1)$. Using $E_A\{\vec{p}(2)\}$, it is able to discuss the covariance matrix for $E_A\{\vec{p}(2)\}$. For simplicity, we denote this by $\vec{p}_A(2)$. Furthermore, we let $\vec{p}_A(1) = E_A\{\vec{p}(1)\}$,

$$\vec{p}_A(2) = A\vec{p}_A(1) + k\vec{r}(0).$$
$$E_{r1}\{\vec{p}_A(2)\} = A\vec{p}_A(1) + km\vec{e}.$$
$$E_{r0}\{E_{r1}\{\vec{p}_A(2)\}\} = km(I + A)\vec{e}.$$

Now, let us define

$$\langle\vec{p}_A(t)\rangle = E_{r0}\{E_{r1}\{\cdots E_{r.t-1}\{\vec{p}_A(t)\}\cdots\}\}.$$

It is readily shown that

$$\langle\vec{p}_A(t+1)\rangle = A\langle\vec{p}_A(t)\rangle + km\vec{e}. \quad (6)$$

Further define that

$$Q_A(t) = E_{r0}\Big\{\cdots E_{r,t-1}\Big\{(\vec{p}_A(t) - \langle\vec{p}_A(t)\rangle)(\vec{p}_A(t) - \langle\vec{p}_A(t)\rangle)^T\Big\}\cdots\Big\},$$

we can thus derive the following recursive equation for $Q_A(t)$:

$$Q_A(t+1) = AQ_A(t)A^T + (km)^2 I. \quad (7)$$

Since $Q_A(t+1) - Q_A(t) \trianglelefteq \sigma^2(A)(Q_A(t) - Q_A(t-1))$, $Q_A(t+1)$ converges and the following theorem can be obtained. **Theorem 2.** *The limit of* $\sigma(Q_A(t))$ *is smaller than*

$$\frac{(km)^2(1 + \max_i\{|\Omega_i|\})^2}{(1 + 2\max_i\{|\Omega_i|\})}.$$

Proof. Since $AQ_A(t)A^T \trianglelefteq \sigma^2(A)Q_A(t)$ and $Q_A(t)$ converges,

$$Q_A(t+1) \trianglelefteq \sigma^2(A)Q_A(t) + (km)^2 I$$
$$\lim_{t\to\infty} Q_A(t) \trianglelefteq \frac{(km)^2}{1 - \sigma^2(A)} I.$$

The notation $A \trianglelefteq B$ means that the matrix $(A - B)$ is negative semidefinite. Hence,

$$\lim_{t\to\infty} \sigma(Q_A(t)) \le \frac{(km)^2(1 + \max_i\{|\Omega_i|\})^2}{(1 + 2\max_i\{|\Omega_i|\})},$$

and the proof is completed. □

The significance of the results stated in Theorem 1 and Theorem 2 can be visualized from Fig. 2. In the limiting case, the distribution of the $\lim_{t\to\infty}\vec{p}_A(t)$ can be approximated by an normal distribution.

Therefore, for all p which is greater than $(1 + \max_i\{|\Omega_i|\})km$, the probability of the expected[5] number of agents exceeding p in any host is given by

$$\text{Prob}(\lim_{t\to\infty} p_A(t) \ge p) \le \int_p^\infty \mathcal{P}(x)dx,$$

where $\mathcal{P}(x)$ is a normal density function given by

$$\text{N}\left((1 + \max_i\{|\Omega_i|\})km, \frac{(km)^2(1 + \max_i\{|\Omega_i|\})^2}{(1 + 2\max_i\{|\Omega_i|\})}\right).$$

Now, let us define a covariance matrix $Q(t)$ similar to that of $Q_A(t)$, i.e.,

5. The expectation is taken on the random variables δ_{jil}.

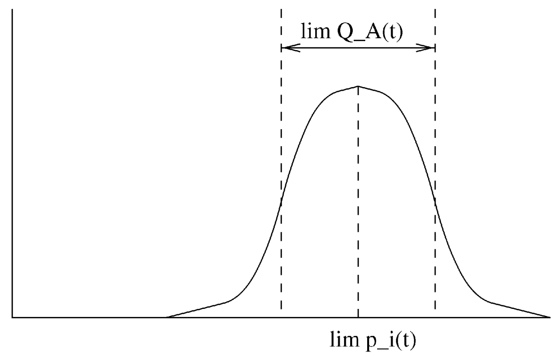

Fig. 2. Distribution of the number of agents in a host approximated by a normal distribution. The horizontal axis corresponds to the number of agents, while the vertical axis corresponds to the probability.

$$Q(t) = E_{r0}\Big\{\cdots E_{r,t-1}\Big\{\big(\vec{p}(t) - \langle\langle\vec{p}(t)\rangle_\delta\rangle_r\big)\big(\vec{p}(t) - \langle\langle\vec{p}_A(t)\rangle_\delta\rangle_r\big)^T\Big\}\cdots\Big\}.$$

Noting that $\{\delta_{jil}(t)\}$, selection of the jth host to go for the lth agent in the ith node, is independent of the vector random variables $\vec{p}(t)$ and $\vec{r}(t)$, it can be readily shown that

$$\langle\langle\vec{p}(t)\rangle_\delta\rangle_r = \langle\vec{p}_A(t)\rangle,$$

$$\vec{p}(t) - \langle\langle\vec{p}(t)\rangle_\delta\rangle_r = \vec{p}(t) - \vec{p}_A(t) + \vec{p}_A(t) - \langle\vec{p}_A(t)\rangle$$

and

$$Q(t) = Q_p(t) + Q_A(t), \quad (8)$$

where $Q_p(t)$ corresponds to the covariance matrix of $(\vec{p}(t) - \vec{p}_A(t))$, i.e.,

$$Q_p(t) = E_{r0}\Big\{\cdots E_{r,t-1}\Big\{(\vec{p}(t) - \vec{p}_A(t))(\vec{p}(t) - \vec{p}_A(t))^T\Big\}\cdots\Big\}. \quad (9)$$

Consider the jth element of the vector $(\vec{p}(t+1) - \vec{p}_A(t+1))$, say $\Delta p_j(t+1)$,

$$\Delta p_j(t+1) = \sum_{i\in\Omega_j}\sum_{l=1}^{p_i(t)}\left\{\delta_{jil}(t) - \frac{1}{|\Omega_i| + 1}\right\}. \quad (10)$$

In the compact form, this can be rewritten as follows:

$$\Delta\vec{p}(t+1) = \sum_l^{p_1(t)}\vec{a}_{1l}(t) + \sum_l^{p_2(t)}\vec{a}_{2l}(t) + \cdots + \sum_l^{p_N(t)}\vec{a}_{Nl}(t) - \vec{p}_A(t+1), \quad (11)$$

where $\vec{a}_{il}(t)$ is a vector indicating the jumping process of the lth agent in the ith host at the tth step.

Consider the graph structure as shown in Fig. 1, suppose that there is an agent in the third host (i.e., node 3), there are three possible jumping vectors for $\vec{a}_{3l}(t)$ (see Fig. 3):

$$(1,0,0,0)^T \quad (0,1,0,0)^T \quad (0,0,0,1)^T.$$

The jumping vector $(0,0,1,0)^T$ is impossible because the agent will die if it jumps back to the same host.

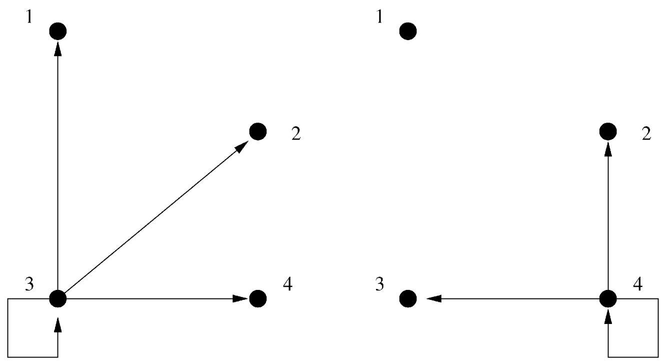

Fig. 3. Possible hosts to jump for an agent in the third host and an agent in the fourth host.

Each of them has probability $1/4$. Similarly, for the fourth host (node 4), there are two possible jumping vectors for $\vec{a}_{4l}(t)$:

$$(0,1,0,0)^T \quad (0,0,1,0)^T.$$

Each of them has probability $1/3$.

Since $\vec{a}_{3l}(t)$ and $\vec{a}_{4l}(t)$ are independent random vectors, the following condition holds:

$$E\Big\{(\vec{a}_{j_1l_1} - E\{\vec{a}_{j_1l_1}\})(\vec{a}_{j_2l_2} - E\{\vec{a}_{j_2l_2}\})^T\Big\} = \begin{cases} A_{j_1} & \text{if } j_1 = j_2 \text{ and } l_1 = l_2 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Let $(A_j)_{ii}$ be the ith diagonal element of matrix A_j,

$$(A_j)_{ii} = \begin{cases} \frac{|\Omega_j|^2+|\Omega_j|-1}{(|\Omega_j|+1)^3} & \text{if } i \in \Omega_j \text{ and } i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The off-diagonal element of $(A_j)_{i_1i_2}$ will be given by

$$(A_j)_{i_1i_2} = \begin{cases} \frac{-|\Omega_j|^2}{(|\Omega_j|+1)^3} & \text{if } i_1, i_2 \in \Omega_j \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

With reference to Fig. 3, the matrices A_3 and A_4 are with structures as follows:

$$A_3 = \begin{bmatrix} + & - & 0 & - \\ - & + & 0 & - \\ 0 & 0 & 0 & 0 \\ - & - & 0 & + \end{bmatrix},$$

where "+" sign corresponds to the value $\frac{|\Omega_3|^2+|\Omega_3|-1}{(|\Omega_3|+1)^3}$ and "−" sign corresponds to the value $\frac{-|\Omega_3|^2}{(|\Omega_3|+1)^3}$.

$$A_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & + & - & 0 \\ 0 & - & + & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where "+" sign corresponds to the value $\frac{|\Omega_3|^2+|\Omega_3|-1}{(|\Omega_3|+1)^3}$ and "−" sign corresponds to the value $\frac{-|\Omega_3|^2}{(|\Omega_3|+1)^3}$.

Now, we can express $E\Big\{\Delta\vec{p}(t+1)\Delta\vec{p}(t+1)^T\Big\}$ in terms of $\vec{p}(t)$. Taking the expectation on both sides of (11) and using the result in (12), we can readily obtain that

$$E\Big\{\Delta\vec{p}(t+1)\Delta\vec{p}(t+1)^T\Big\} = \sum_{l}^{p_1(t)} A_1 + \sum_{l}^{p_2(t)} A_2 + \cdots + \sum_{l}^{p_N(t)} A_N \tag{15}$$

$$= \sum_{j=1}^{N} A_j p_{Aj}(t). \tag{16}$$

where $p_{Aj}(t)$ is the jth element of the vector $\vec{p}_A(t)$.

In accordance with (12), the jth diagonal element of

$$E\Big\{\Delta\vec{p}(t+1)\Delta\vec{p}(t+1)^T\Big\}$$

is given by

$$\sum_{i\in\Omega_j}\sum_{l}^{p_i(t)} (A_i)_{jj} = \sum_{i\in\Omega_j} p_i(t)\frac{|\Omega_i|^2 + |\Omega_3| - 1}{(|\Omega_i|+1)^3}.$$

The j_1j_2 off-diagonal element is given by

$$\sum_{i\in\Omega_{j_1}\bigcap\Omega_{j_2}}\sum_{l}^{p_i(t)} (A_i)_{j_1j_2} = \sum_{i\in\Omega_{j_1}\bigcap\Omega_{j_2}} p_i(t)\frac{-|\Omega_i|^2}{(|\Omega_i|+1)^3}.$$

Therefore, using Lemma 1, we can obtain the following Lemma:

Lemma 2. *The largest eigenvalue of the covariance matrix $\sigma(Q_p)$ is at most $2km\max_i\{|\Omega_i|\}$.*

Proof. Since

$$0 \le \sum_{i\in\Omega_{j_1}\bigcap\Omega_{j_2}} p_i(t)\frac{|\Omega_i|^2}{(|\Omega_i|+1)^3} \le \sum_{i\in\Omega_{j_1}} p_i(t)\frac{|\Omega_i|^2}{(|\Omega_i|+1)^3} \le \max_i\{p_i\},$$

and

$$\sum_{i\in\Omega_j} p_i(t)\frac{|\Omega_i|^2 + |\Omega_3| - 1}{(|\Omega_i|+1)^3} \le \max_i\{p_i\},$$

by Lemma 1, we get that $\sigma(Q_p) \le 2km\max_i\{|\Omega_i|\}$, and the proof is completed. □

Hence, using Theorem 1, the following theorem can also be obtained:

Theorem 3.

$$\lim_{t\to\infty}\sigma(Q(t)) \le 2km\max_i\{|\Omega_i|\} + \frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{(1+2\max_i\{|\Omega_i|\})}. \tag{17}$$

Remarks:

- The major results obtained in this section indicate that the population of agents in the network will reach an upper bound. The relation between $Q(t)$ and $Q_A(t)$ can be visualized from Fig. 4.
- The significance of Theorem 3 indicates that the probability of the number of agents exceeding p, where $p \ge (1+\max_i\{|\Omega_i|\})km$, is less than

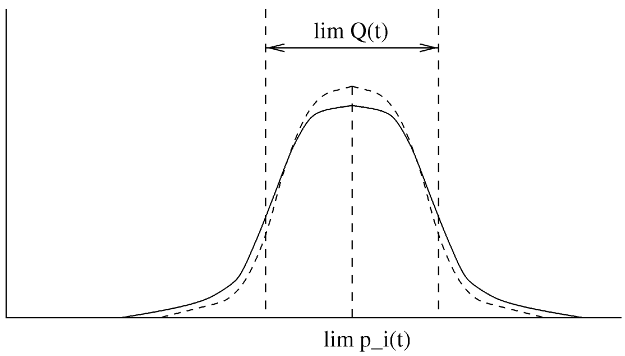

Fig. 4. Distribution of the number of agents in a host. The solid line corresponds to the upper bound $P(\vec{p})$ while the dash line corresponds to $P(\vec{p}_A)$.

$$\int_p^\infty \mathcal{P}(x)dx,$$

where $\mathcal{P}(x)$ is a normal distribution function with mean and variance given by

$$\text{Mean} = (1+\max_i\{|\Omega_i|\})km,$$

$$\text{Var.} = 2km(1+\max_i\{|\Omega_i|\}) + \frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{(1+2\max_i\{|\Omega_i|\})}.$$

- The expected number of agents in each host and the covariance matrix derived can be treated as a guideline for choosing the value of k, which is the number of agents created for each request. For example, let p_{max} be the maximum number of agents which can be processed in any host. Setting

$$km(1+\max_i\{|\Omega_i|\}) + 1.65 \left[2km(1+\max_i\{|\Omega_i|\}) + \frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{(1+2\max_i\{|\Omega_i|\})}\right]^{1/2} = p_{max},$$

it will ensure that $P\{p_i(t) \geq p_{max}\} \leq 0.05$.

4 Average Number of Jumping Hops

Another issue concerning ant routing is the expected number of hops required for an ant agent to reach its destination. This factor is important when one has to define the maximum number of hops for an agent in order to reduce the agents' population in the network. To estimate this value, we can first define a binary random variable s_t as follows:

$$s_t = \begin{cases} 1 & \text{if the agent still survives at the } t\text{th step} \\ 0 & \text{if the agent dies at the } t\text{th step.} \end{cases} \quad (18)$$

Then, the probability that $s_{t+1} = 1$ given $s_t = 1$ can be expressed as follows:

$$P\{s_{t+1}|s_t = 1\} = \begin{cases} \frac{|\Omega_t|}{|\Omega_t|+1} & \text{if } s_{t+1} = 1 \\ \frac{1}{|\Omega_t|+1} & \text{if } s_{t+1} = 0, \end{cases} \quad (19)$$

and $P\{s_0 = 1\} = 1$. So that the probability that the agent will jump exactly t steps is given by

$$P\{jump = t\} = P\{s_{t+1} = 0, s_t = \cdots = s_1 = s_0 = 1\} \quad (20)$$

$$= \frac{1}{|\Omega_{t+1}|+1}\prod_{\tau=1}^{t}\frac{|\Omega_\tau|}{|\Omega_\tau|+1} \quad (21)$$

$$\leq \frac{1}{\min_i\{|\Omega_i|\}+1}\frac{\max_i\{|\Omega_i|\}^t}{(\max_i\{|\Omega_i|\}+1)^t}, \quad (22)$$

where i corresponds to the host where the agent visits at the ith step. While Ω_i is the corresponding neighborhood set. Therefore, the expected number of jumps (jumping hops) $E\{jump\}$ can be given by the following theorem:

Theorem 4. *The expected number of jumps (jumping hops) cannot be greater than the maximum number of* $|\Omega_i|$, *i.e.,*

$$E\{jump\} \leq \max_i\{|\Omega_i|\}\frac{\max_i|\Omega_i|+1}{\min_i|\Omega_i|+1}.$$

Proof.

$$E\{jump\} \leq \sum_{k=1}^{\infty}\frac{k}{\min_i\{|\Omega_i|\}+1}\left(\frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\}+1}\right)^k \quad (23)$$

$$= \frac{\max_i\{|\Omega_i|\}}{(\max_i\{|\Omega_i|\}+1)(\min_i|\Omega_i|+1)}\sum_{k=1}^{\infty}k \left(\frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\}+1}\right)^{k-1} \quad (24)$$

Due to the fact that $\sum_{k=1}^{\infty}kx^{k-1} = \frac{d}{dx}(1-x)^{-1}$ and the proof is completed. □

One can also derive a range for the probability that the number of jumping hops being greater than t.

Theorem 5. *The probability for the number of jumping hops being larger than t,* $P\{jump \geq t\}$, *satisfies the following inequality:*

$$P_L(t) \leq P\{jump \geq t\} \leq \min\{P_U(t), 1\},$$

where

$$P_L(t) = \frac{\min_i\{|\Omega_i|\}+1}{\max_i\{|\Omega_i|\}+1}\left(\frac{\min_i\{|\Omega_i|\}}{\min_i\{|\Omega_i|\}+1}\right)^t,$$

$$P_U(t) = \frac{\max_i\{|\Omega_i|\}+1}{\min_i\{|\Omega_i|\}+1}\left(\frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\}+1}\right)^t.$$

Proof. Since $P\{jump \geq t\} = \sum_{k=t}^{\infty}\frac{1}{|\Omega_k|+1}\prod_{\tau=1}^{k}\frac{|\Omega_{\tau-1}|}{|\Omega_{\tau-1}|+1}$, it is obvious that

$$P_L(t) \leq P\{jump \geq t\} \leq P_U(t),$$

where

$$P_U(t) = \sum_{k=t}^{\infty}\frac{1}{\min_i\{|\Omega_i|\}+1}\left(\frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\}+1}\right)^k = \frac{\max_i\{|\Omega_i|\}+1}{\min_i\{|\Omega_i|\}+1}\left(\frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\}+1}\right)^t, \quad (25)$$

$$P_L(t) = \sum_{k=t}^{\infty}\frac{1}{\max_i\{|\Omega_i|\}+1}\left(\frac{\min_i\{|\Omega_i|\}}{\min_i\{|\Omega_i|\}+1}\right)^k = \frac{\min_i\{|\Omega_i|\}+1}{\max_i\{|\Omega_i|\}+1}\left(\frac{\min_i\{|\Omega_i|\}}{\min_i\{|\Omega_i|\}+1}\right)^t \quad (26)$$

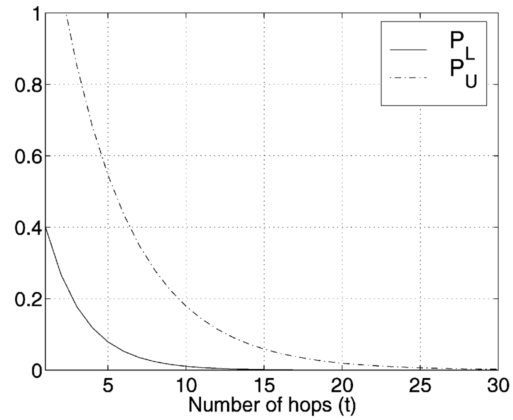

Fig. 5. The changes of $P_L(t)$ and $P_U(t)$ against the number of hops t for a simple network.

Since $P\{jump \geq t\}$ cannot be greater than one, $P\{jump \geq t\} \leq \min\{P_U(t), 1\}$ and the proof is completed.□

For illustration, Fig. 5 shows the changes of $P_L(t)$ and $P_U(t)$ against the number of jumping hops t for the network shown in Fig. 1. Here, $\max_i\{|\Omega_i|\}$ is four and $\min_i\{|\Omega_i|\}$ is two.

Remark: Note that the results derived in this section are independent from the agent being propagated in synchronous or asynchronous mode. Therefore, they are general enough for setting guidelines for designing the maximum number of jumping hops. For example, in accordance with Fig. 5, the probability that an agent can have more than 20 jumps is very small. Then, we can set the maximum number of jumping hops to be 20 in order to reduce the computational resource.

5 AGENTS POPULATION: ASYNCHRONOUS MODE

Discussion in the previous sections relies on the assumption that the agents are propagating in the synchronous mode. In this section, we will present a preliminary result on the average case when the network is operated in the asynchronous mode. This might happen when there are uneven transition delays. To simplify the discussion, we first give a simple example showing our approach to analysis.

An example is shown in Fig. 6. The transition delay between any two neighboring hosts is depicted by the number labelled on the edge connecting them.

Let $\langle p_i(t)\rangle_\delta$ be the expectation of the number of agents taking over the random variables δ_{jil}s in the ith host $(i = 1, \cdots, 4)$ at the tth step, given $\langle p_1(\tau)\rangle, \cdots, \langle p_4(\tau)\rangle$ for $\tau < t$.

$$\langle p_1(t+1)\rangle_\delta = kr_1(t) + \frac{\langle p_3(t-1)\rangle_\delta}{|\Omega_3|+1}, \quad (27)$$

$$\langle p_2(t+1)\rangle_\delta = kr_2(t) + \frac{\langle p_3(t)\rangle_\delta}{|\Omega_3|+1} + \frac{\langle p_4(t-2)\rangle_\delta}{|\Omega_4|+1}, \quad (28)$$

$$\langle p_3(t+1)\rangle_\delta = kr_3(t) + \frac{\langle p_1(t-1)\rangle_\delta}{|\Omega_1|+1} + \frac{\langle p_2(t)\rangle_\delta}{|\Omega_2|+1} + \frac{\langle p_4(t)\rangle_\delta}{|\Omega_4|+1}, \quad (29)$$

$$\langle p_4(t+1)\rangle_\delta = kr_4(t) + \frac{\langle p_2(t-2)\rangle_\delta}{|\Omega_2|+1} + \frac{\langle p_3(t)\rangle_\delta}{|\Omega_3|+1}. \quad (30)$$

Next, let us define $\langle \vec{p}_A(t)\rangle_\delta = \vec{p}_A$ where

$$\vec{p}_A = \langle (p_1, p_2, p_3, p_4)\rangle_\delta^T,$$

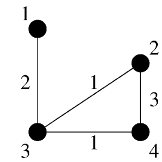

Fig. 6. A simple network with uneven transition delays.

(27) to (30) can be rewritten as the following model:

$$\vec{p}_A(t+1) = T_{11}\vec{p}_A(t) + T_{12}\vec{p}_A(t-1) + T_{13}\vec{}_A(t-2) + k\vec{r}(t), \quad (31)$$

where

$$T_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{|\Omega_3|+1} & 0 \\ 0 & \frac{1}{|\Omega_2|+1} & 0 & \frac{1}{|\Omega_4|+1} \\ 0 & 0 & \frac{1}{|\Omega_3|+1} & 0 \end{bmatrix},$$

$$T_{12} = \begin{bmatrix} 0 & 0 & \frac{1}{|\Omega_3|+1} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{|\Omega_1|+1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$T_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{|\Omega_4|+1} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{|\Omega_2|+1} & 0 & 0 \end{bmatrix}.$$

Taking the expectation on both sides w.r.t. the random vectors $\vec{r}(t)$, $\vec{r}(t-1)$, $\cdots$, $\vec{r}(1)$, the following equation is readily obtained:

$$\begin{aligned}\langle \vec{p}_A(t+1)\rangle_r =& T_{11}\langle \vec{p}_A(t)\rangle_r + T_{12}\langle \vec{p}_A(t-1)\rangle_r \\ &+ T_{13}\langle \vec{p}_A(t-2)\rangle_r + km\vec{e},\end{aligned} \quad (32)$$

for all $t \geq 2$. As T_{11}, T_{12}, and T_{13} are all with nonnegative elements, it can be derived that

$$\begin{aligned}&\|\langle \vec{p}_A(t+1)\rangle_r - \langle \vec{p}_A(t)\rangle_r\| \\ &\leq (T_{11}+T_{12}+T_{13})\max_{1\leq\tau\leq 3}\left\{\|\langle \vec{p}_A(t-(\tau-1))\rangle_r - \langle \vec{p}_A(t-(\tau))\rangle_r\|\right\}.\end{aligned}$$

Hence,

$$\lim_{t\to\infty} \|\langle \vec{p}_A(t+1)\rangle_r - \langle \vec{p}_A(t)\rangle_r\| = 0,$$

which means the limit exists for $\lim_{t\to\infty}\langle \vec{p}_A(t+1)\rangle_r$. Let this limiting vector be p_0, using (32) it can readily be shown that

$$\lim_{t\to\infty}\langle \vec{p}_A(t+1)\rangle_r = km[I - T_{11} - T_{12} - T_{13}]^{-1}\vec{e} \quad (33)$$

$$= km[I - A]^{-1}\vec{e} \quad (34)$$

$$\leq km(1 + \max_i\{|\Omega_i|\})\vec{e}. \quad (35)$$

Using this approach, we can see that for any finite propagation delay τ, an equation similar to that of (32) can be written as follows:

$$\langle \vec{p}_A(t+1)\rangle_r = \sum_{i=1}^{\tau} T_{1i}\langle \vec{p}_A(t-(i-1))\rangle_r + km\vec{e}, \quad (36)$$

TABLE 1
Summary of the Theorems Being Proven in This Paper

Theorem	Result	Mode
1	$\langle p \rangle = \langle p_A \rangle \leq (1 + \max_i\{\lvert\Omega_i\rvert\})km$	Syn.
2	$\sigma(Q_A) \leq \frac{(km)^2(1+\max_i\{\lvert\Omega_i\rvert\})^2}{(1+2\max_i\{\lvert\Omega_i\rvert\})}$	Syn.
3	$\sigma(Q) \leq 2km(1 + \max_i\{\lvert\Omega_i\rvert\}) + \frac{(km)^2(1+\max_i\{\lvert\Omega_i\rvert\})^2}{(1+2\max_i\{\lvert\Omega_i\rvert\})}$	Syn.
4	$E\{jump\} \leq \max_i\{\lvert\Omega_i\rvert\}\frac{\max_i \lvert\Omega_i\rvert+1}{\min_i \lvert\Omega_i\rvert+1}$	Syn./Asyn.
5	$P_L(t) \leq P\{jump \geq t\} \leq \min\{P_U(t), 1\}$	Syn./Asyn.
6	$\langle p \rangle = \langle p_A \rangle \leq (1 + \max_i\{\lvert\Omega_i\rvert\})km$	Asyn.
7	$\sigma(Q_A) \leq \frac{(km)^2(1+\max_i\{\lvert\Omega_i\rvert\})^2}{(1+2\max_i\{\lvert\Omega_i\rvert\})}$	Asyn.

$$\sum_{i=1}^{\tau} T_{1i} = A, \tag{37}$$

where the matrix $A = (a_{ji})_{N\times N}$ is defined in a similar way as in synchronious mode, i.e.,

$$a_{ji} = \begin{cases} \frac{1}{|\Omega_i|+1} & \text{if } j \in \Omega_i, j \neq i \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the following theorem can be stated for the general case when τ is any finite integer.

Theorem 6. *Using the ant routing algorithm in asynchronous mode, the expected number of agents running in each host is at most* $(1 + \max_i\{|\Omega_i|\})km$.

That is to say, as long as the propagation delay is finite, the expected number of agents in each host server is finite and bounded. Let us define the variance of $\vec{p}_A(t)$ as that

$$Q_A(i,j|k) = E_{r0}\Big\{\cdots E_{r,k}\Big\{(\vec{p}_A(i) - \langle\vec{p}_A(i)\rangle)(\vec{p}_A(j) - \langle\vec{p}_A(j)\rangle)^T\Big\}\cdots\Big\},$$

we can show that the variance $Q_A(t,t|t-\tau)$ is bounded above for all $t > 0$.

Theorem 7. *The values* $\sigma(Q_A(t,t|t-\tau))$ *is bounded by*

$$\frac{(km)^2(1 + \max_i\{|\Omega_i|\})^2}{1 + 2\max_i\{|\Omega_i|\}},$$

for all $t \geq 0$.

Proof. See Appendix. □

It should be noted that Theorem 7 does not show that $\sigma(Q_A(t,t|t-\tau))$ converges, in contrast to Theorem 2, which has shown that $\lim_{t\to\infty}\sigma(Q_A(t))$ exists.

6 CONCLUSION

This paper has analyzed on the growing behavior and the jumping behavior of an agent based routing algorithm. The theorems given in this paper are summarized in Table 1. The subscripts i, j are dropped for simplicity. It is found that as long as the network topology is fixed and the number of agents being created for each request is finite, the expectation and the covariance of the expected number of agents in each host server must also be finite, independent of whether the network is operated in synchronous or asynchronous mode, as shown in Theorems 1 and 2 and Theorems 6 and 7. If the network is operated in synchronous mode, we further show that the covariance of the number of agents in each host server is also finite (Theorem 3). Finally, we have also analyzed the jumping behavior of an agent and showed that the expected number of jumping hops is smaller than $\max_i\{|\Omega_i|\}$ (Theorems 4 and 5). All these results provide guidelines for the design of an agent based routing system, particularly when the computational power of the host servers is not able to handle large amount of processing.

It should be further noted that the searching behavior of ant routing is similar to the searching behavior of "SupplierSearch" agent or "BuyerSearch" agent in the electronic marketplace [34]. Once a buyer would like to buy certain product, the buyer will set the criteria to a mobile agent. The agent thus searches the suppliers' information available on the global marketplace, say Global Trading Web^{TM} for instance, and checks whether any product listed on the supplier's catalogue matches the buyer criteria. Obvious, this mechanism is indeed the same as the routing algorithm presented in this paper. The buyer agent needs to search all the suppliers' site in order to collect all the information for price comparison. As we can see from analyst forecast, there will be an increasing amount of trading conducted via Internet and particularly the electronic marketplace, a good design for the platform is inevitable. One should realize that the results presented in this paper provide guidelines not only for agent routing design but also for this agent-mediated electronic marketplace design.

APPENDIX

PROOF OF THEOREM 7

Proof. Using the inequality, $\sigma(AB^T) \leq \frac{1}{2}(\sigma(AA^T) + \sigma(BB^T))$, it can readily be deduced that

$$\begin{aligned} & T_{1i}Q_A(t+1-i, t+1-j|t-\tau)T_{1j}^T \\ \trianglelefteq\ & T_{1i}\sigma(Q_A(t+1-i, t+1-j|t-\tau))T_{1j}^T \\ \trianglelefteq\ & \frac{1}{2}T_{1i}\sigma(Q_A(t+1-i, t+1-i|t-\tau))T_{1j}^T \\ & + \frac{1}{2}T_{1i}\sigma(Q_A(t+1-j, t+1-j|t-\tau))T_{1j}^T. \end{aligned}$$

Hence,

$$Q_A(t+1,t+1|t-\tau)=\sum_{i=1}^{\tau}\sum_{j=1}^{\tau}T_{1i}Q_A(t+1-i,t+1-j|t-\tau)T_{1j}^T+(km)^2I.$$

$$\leq \frac{1}{2}\sum_{i=1}^{\tau}\sum_{j=1}^{\tau}T_{1i}\sigma(Q_A(t+1-i,t+1-i|t-\tau))T_{1j}^T + \frac{1}{2}\sum_{i=1}^{\tau}\sum_{j=1}^{\tau}T_{1i}\sigma(Q_A(t+1-j,t+1-j|t-\tau))T_{1j}^T + (km)^2I.$$

It can then be proven by the principle of mathematical induction that if

$$\sigma(Q_A(k+1-i,k+1-j|k-\tau)) \leq \frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{1+2\max_i\{|\Omega_i|\}}$$

for all $k=1,\cdots,t-1$,

$$\sigma(Q_A(t+1,t+1|t-\tau)) \leq \sigma\left(\sum_{i=1}^{\tau}\sum_{j=1}^{\tau}T_{1i}T_{1j}^T\right)\frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{1+2\max_i\{|\Omega_i|\}}+(km)^2I$$

$$=\sigma(A)^2\frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{1+2\max_i\{|\Omega_i|\}}+(km)^2I$$

$$\leq \frac{\max_i\{|\Omega_i|\}^2}{(1+\max_i\{|\Omega_i|\})^2}\frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{1+2\max_i\{|\Omega_i|\}}+(km)^2I.$$

Therefore,

$$\sigma(Q_A(t+1,t+1|t-\tau)) \leq \frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{1+2\max_i\{|\Omega_i|\}},$$

and the proof is completed. □

ACKNOWLEDGMENTS

This work was partially supported by the Competitive Earmarked Grant Research Grant (Project: 9040603), the Research Grants Council, and the Hong Kong Special Administration Region.

REFERENCES

[1] L. Cardelli, "A Language with Distributed Scope," *Proc. ACM Symp. Principles of Programming Languages,* pp. 286-297, 1995.
[2] F. Douglis and J. Ousterhout, "Transparent Process Migration: Design Alternatives and the Sprite Implementation," *Software Practice and Experience,* vol. 21, no. 8, pp. 757-785, Aug. 1991.
[3] C.G. Harrison, D.M. Chess, and A. Kershenbaum, "Mobile Agents: Are They a Good Idea?," technical report, IBM Research Division, T.J. Watson Research Center, (http://www.research.ibm.com/massdist/mobag.ps), Mar. 1995.
[4] E. Jul, H. Levy, N. Hutchinson, and A. Black, "Fine-Grained Mobility in the Emerald System," *ACM Trans. Computer Systems,* vol. 6, no. 1, pp. 109-133, Feb. 1988.
[5] M. Ranganathan, A. Acharya, S. Sharma, and J. Saltz, "Network-aware Mobile Programs," *Proc. USENIX,* Winter 1997.
[6] T. Thorn, "Programming Languages for Mobile Code," *ACM Computing Surveys,* vol. 29, no. 3, pp. 213-239, Sept. 1997.
[7] A. Lingnau, O. Drobnik, and P. Domel, "An HTTP-Based Infrastructure for Mobile Agents," *Proc. Fall 1995 WWW Conf.,* Dec. 1995.
[8] C. Munday, J. Dangedej, T. Cross, and D. Lukose, "Motivation and Perception Mechanisms in Mobil Agent for Electronic Commerce," *Proc. First Australian Workshop Distributed Artificial Intelligence,* vol. 1087, pp. 144-158, Nov. 1996.
[9] P. Wayner, "Agents Away," *Byte,* May 1994.
[10] G. Karjoth, D. Lange, and M. Oshima, "A Security Model for Aglets." *IEEE Internet Computing,* pp. 68-77, July/Aug. 1997.
[11] D.B. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets,* Addison Wesley, 1998.
[12] "ObjectSpace. ObjectSpace Voyager Core Package Technical Overview,"technical report, ObjectSpace, Inc., July 1997.
[13] R.S. Gray, "Agent TCL: A Flexible and Secure Mobile-Agent System," *Proc. Fourth Ann. TCL/TK Workshop, (TCL '96),* July 1996.
[14] D. Johansen, R. van Renesse, and F.B. Schneider, "An Introduction to the TACOMA Distributed System," Technical Report 95-23, Dept. of Computer Science, Univ. of Tromso, June 1995.
[15] J. Hylton, K. Manheimer, F.L. Drake Jr., B. Warsaw, R. Masse, and G. van Rossum, "Knowbot Programming: System Support for Mobile Agents," *Proc. Fifth Int'l Workshop Object Orientation in Operating Systems (IWOOOS '96),* Oct. 1996.
[16] J.E. White, "Mobile Agents," technical report, General Magic, available at URL http://www.genmagic.com/Telescript/, Oct. 1995.
[17] B. Bakshi, P. Krishna, N. Vaidya, and D. Pradhan, "Improving Performance of TCP Over Wireless Networks, Dept. of Computer Science, Texas A&M Univ., Technical Report 96-014, May 1996.
[18] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP Performance Over Wireless Networks," *Proc. First ACM Int'l Conf. Mobile Computing and Networking (Mobicom),* Nov. 1995.
[19] C. Perkins, "Providing Continuous Network Access to Mobile Hosts Using TCP/IP," *Computer Networks and ISDN Systems,* vol. 26, pp. 357-370, Nov. 1993.
[20] J.D. Soloman, *Mobile IP: The Internet Unplugged.* Prentice Hall, 1998.
[21] M. Taylor, W. Waung, and M. Banan, *Internetworking Mobility: The CDPD Approach.* Prentice Hall, 1997.
[22] A. Bieszczad, T. White, and B. Pagurek, "Mobile Agents for Network Management," *IEEE Comm. Surveys,* Sept. 1998.
[23] M. El-Darieby and A. Bieszczad, "Intelligence Mobile Agents: Towards Network Fault Management Automation," *Proc. Sixth IFIP/IEEE Int'l Symp. Integrated Network Management,* May 1999.
[24] C. Schramm, A. Bieszczad, and B. Pagurek, "Application-Oriented Network Modeling with Mobile Agents," *Proc. IEEE/IFIP Network Operations and Management Symp., NOMS '98,* Feb. 1998.
[25] T. White, "Routing with Swarm Intelligence," Technical Report SCE-97-15, Systems and Computer Eng., Carleton Univ., Sept. 1997.
[26] A.S. Tanenbaum, *Computer Networks,* third ed., Prentice Hall, 1996.
[27] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: Freeman, 1979.
[28] T. White, B. Pagurek, and F. Oppacher, "ASGA: Improving the Ant System by Integration with Genetic Algorithms," *Proc. Third Conf. Genetic Programming (GP/SGA '98),* pp. 610-617, July 1998.
[29] T. White, B. Pagurek, and F. Oppacher, "Connection Management Using Adaptive Agents," *Proc. PDPTA '98,* pp. 802-809, July 1998.
[30] T. White and B. Pagurek, "Towards Multi-Swarm Problem Solving in Networks," *Proc. Third Int'l Conf. Multi-Agent Systems (ICMAS '98),* pp. 333-340, July 1998.
[31] B. Bullnheimer, R.F. Hartl, and C. Strauss, "Applying the Ant System to the Vehicle Routing Problem," *Proc. Second Metaheuristics Int'l Conf. (MIC-97),* 1997.
[32] G. Di Caro and M. Dorigo, "AntNet: A Mobile Agents Approach to Adaptive Routing," Technical Report IRIDIA/97-12, Univ. Libre de Bruxelles, Belgium, 1997.
[33] R. Bronson, *Matrix Operation.* Schaum's Outline Series, McGraw Hill, 1989.
[34] H. William, K. Cheung, C.H. Li, E.C.M. Lam, and J. Liu, "Customized Electronic Commerce with Intelligent Software Agents," *Internet Commerce and Software Agents,* S.M. Rahman, ed., Idea Group Publishing, 2000.

John Sum received the BEng in electronic engineering from the Hong Kong Polytechnic University in 1992. After that, he went to the Department of Computer Science and Engineering of the Chinese University of Hong Kong studying and researching on the area of Neural Networks and received the MPhil and PhD degrees in 1995 and 1998, respectively. He has published many journal and conference papers. He is a referee for international journals and conferences, an organizing committee member, a program committee member, and session chair of several international conferences. In 1992, John received the Hong Kong Institution of Engineers Student Prize in recognition of his academic merit, extra-curricular activities, leasdership qualities, and personality during his undergraduate study. Twice he has received two times the Certificate of Merits from the Department of Computer Science and Engineering of CUHK in appreciation for his excellent teaching assistantship, in 1996 and 1997, respectively. Dr. Sum has been selected for inclusion in the Year 2000 Marquis Who's Who in the World. He has a broad research interests including Web Commerce, Information Economy, and some others. Since 1999, John has also been affiliated with the Laboratory of Internet Software Technologies, Institute of Software, Chinese Academy of Science, working on the analysis and design of an Internet platform for procuring over active supply chain. Currently, he is serving as a advisory board member in various MNC providing technological consultant service.

Hong Shen received the BEng degree from Beijing University of Science and Technology, China, the MEng degree from University of Science and Technology of China, the PhLic and PhD degrees from Abo Akademi University, Finland, all in computer science. He is currently a full professor in the Graduate School of Information Science, Japan Advanced Institute of Science and Technology. Previously, he was a professor at Griffith University, Australia. Dr. Shen has published more than 140 technical papers on algorithms, parallel and distributed computing, interconnection networks, parallel databases and data mining, multimedia systems, and networking. He has served as an editor of *Parallel and Distributed Computing Practice*, associate editor of *International Journal of Parallel and Distributed Systems and Networks*, editorial board member of *Parallel Algorithms and Applications*, *International Journal of Computer Mathematics*, and *Journal of Supercomputing*, and chaired various international conferences. Dr. Shen is a recipient of 1991 National Education Commission Science and Technology Progress Award and 1992 Sinica Academia Natural Sciences Award.

Chi-sing Leung received the BSci degree in electronics from the Chinese University of Hong Kong in 1989. He received the MPhil degree in information engineering and the PhD degree in computer science from the same university in 1991 and 1995, respectively. He is currently an associate professor in the Department of Electronic Engineering, City University of Hong Kong. His research interests include neural computing, data mining, and computer graphics.

G. Young is with the Department of Computer Science, CAL POLY, Pomona, California.