Available online at www.sciencedirect.com

SCIENCE DIRECT®

Applied Soft Computing 6 (2005) 108–118

Applied Soft Computing

www.elsevier.com/locate/asoc

New analysis on mobile agents based network routing

Wenyu Qu [a,*], Hong Shen [a], John Sum [b]

[a] Graduate School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Tatsunokuchi, Ishikawa 923-1292, Japan

[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong

Received 26 April 2004; received in revised form 14 January 2005; accepted 8 February 2005

Abstract

This paper focuses on behavior analysis on mobile agents used in network routing. We describe a general agent-based routing model and classify it into two cases based on the reaction of mobile agents to a system failure, namely mobile agents with weak reaction capability (MWRC) and mobile agents with strong reaction capability (MSRC). For each case, we analyze the probability of success (the probability that an agent can find the destination) and the population distribution (the number of mobile agents) of mobile agents. The probability of success serves as an important measure for monitoring network performance, and the analysis of population distribution provides a useful tool for reducing the computational resource consumption. Our analysis reveals theoretical insights into the statistical behaviors of mobile agents and provides useful tools for effectively managing mobile agents in large networks.

Keywords: Mobile agents; Routing; Probability of success; Population distribution

1. Introduction

Mobile agent, a relatively new paradigm for network software development, has become an accessible technology in recent years. The potential benefits of this technology, including the reduction of network bandwidth consumption and latency, have drawn a great deal of attention in both academia and industry [3,11,19,20]. 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.

In recent years, many intelligent mobile agent-based network management techniques have been proposed and implemented [1,6,10,14]. When a mobile agent is encapsulated with a task, it can be dispatched to a remote node. Once the agent has completed its tasks, the summary report for its trip is sent back to the source node. Since there are very few communications between the agent and the source node during the process of searching, the network traffic generated by mobile agents is very light. So, mobile agent is an effective way for improving network performance.

* Corresponding author. Tel.: +81 761 51 1285; fax: +81 761 51 1149.

E-mail address: qu_wenyu@jaist.ac.jp (W. Qu).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2005.02.005

Network routing is an important issue for network performance. Advanced research in mobile agent has brought in some new methods for network routing [5,15]. Ant routing algorithm is a recently proposed routing algorithm for use in large dynamic networks [7,13,17,21]. The idea is similar to the shortest path searching process of ants. For an agent-based network, agents can be generated from every node in the network, and each node in the network provides mobile agents an execution environment. A node that generates mobile agents is called the server of these agents. Once a request for sending a packet is received from a server, the server will generate a number of mobile agents. These agents will then move out from the server to search for the destination. Once a mobile agent finds the destination, the information will be sent back to the server along the same path. When all (or some of) the mobile agents come back, the server will determine the optimal path and send the packet to the destination along the optimal path. At the same time, the server will update its routing table.

Since mobile agents will be generated frequently in the network, there will be many agents running in the network. On one hand, if there are too many mobile agents running in the network, they will consume too much computational resource, which will affect the network performance due to the limited network resource and ultimately block the entire network; on the other hand, if the number of generated agents per request is too small, we cannot get a high probability of success. Therefore, analysis on mobile agents is necessary and important for network management. Unfortunately, few works have been done on this aspect.

In [18], an ant routing model was proposed and the number of mobile agents was estimated under the assumption that nodes in the network will not fail. Thus, it can be viewed as a special case of the model in this paper. In [16], a smaller upper bound of the number of mobile agents was provided based on the same model in [18], and for the first time the probability of success was considered. In this paper, we describe a general mobile agent-based routing model and classify it into two cases based on the reaction capability of mobile agents to a system failure. For each case, we analyze both the probability of success and the population distribution of mobile agents. Our contributions are summarized as follows:

- A general agent-based routing model is described and is classified into two cases based on the reaction of mobile agents to a system failure: MWRC and MSRC.
- The probability of success is analyzed for each case, which serves as an important measure for monitoring network performance.
- Analysis on population distribution of mobile agents is presented for both cases, providing a useful tool to reduce the computational resource consumption by adjusting the number of agents to be generated at individual nodes and the life span of these mobile agents.

In any mobile agent-based routing models, mobile agents must be generated and dispatched to the network frequently. Although a large number of agents generated per request would bring a high success probability, an excessive number of agents will consume too much computational resources due to per-agent overhead. Our results provide a guideline for choosing a suitable propagating rate to benefit both the probability of success and the network performance (illustrated by the population distribution). The results are extremely useful when the computational power of the host servers is limited, which is unable to handle large amount of processing requests and/or the network channel capacity is limited for large volume of mobile agents propagating in the network.

The rest of this paper is organized as follows: Section 2 discusses related work; Section 3 describes our model; Section 4 introduces the notations used in this paper and presents the analytical results for mobile agents, including the probability of success and the population of agents; Section 5 concludes the paper.

2. Related work

A mobile agent is an autonomous object that possesses the ability for migrating autonomously from node to node in a computer network. Usually, the main task of a mobile agent is determined by specified applications of users, which can range from E-shopping and distributed computation to real-time device control. In recent years, a number of research institutions and industrial entities have been engaged in the development of elaborating supporting systems for this technology [11,23]. In [11], several merits for mobile agents are described, including network load and latency reduction, protocol encapsulation, adaption, heterogeneity, robustness and fault-tolerance. Successful examples using mobile agents can be found in [10,12].

Network routing is a problem in network management. Ant routing is a recently proposed mobile agent based network routing algorithm for use in these environments [21,22]. The continuing investigation and research of naturally occurring social systems offer the prospect of creating artificial systems that are controlled by emergent behavior and promise to generate engineering solutions to distributed systems management problems such as those in communication networks [5,17].

Real ants are able to find the shortest path from a food source to the nest without using visual cues. Also, they can adapt to changes in the environment, for example, finding a new shortest path once the old one is no longer feasible due to a new obstacle [2,9]. In the ant routing algorithm described in [7,18], artificial ants are agents which concurrently explore the network from node to node and exchange collected information when they meet each other. They irrespectively choose the node to move by using a probabilistic function which was proposed here to be a function of the connecting situation of each node. Artificial ants probabilistically prefer nodes that are connected immediately. Initially, a number of artificial ants are placed on randomly selected nodes. At each time step, they move to new nodes and select useful information. When an ant has completed its task, it will send a message back to the server.

In [4], Brewington et al. formulated a method of mobile agent planning, which is analogous to the traveling salesman problem [8] to decide the sequence of nodes to be visited by minimizing the total execution time until the destination is found. In the preliminary work of this paper [16], both the population distribution of mobile agents and the probability of success are analyzed. The model can be seen as a special case of the one in this paper.

3. Mobile agent-based routing model

Once a request is received by a server, the server generates a number of mobile agents. These agents will then move out from the server searching for the destination. Once an agent finds the destination, it will traverse back to the server following the searched path and leave marks on the nodes along the path. When a certain number of agents have come back (others may have dead, or are still in the searching process), the server will evaluate the costs of those collected paths and pick up the optimal one. The main idea of our algorithm is as follows:

1. In a network with n nodes, agents can be generated from every node in the network. Each node in the network provides to mobile agents an execution environment.
2. Initially, there are piles of requests for sending packets in the network. Then, a number of mobile agents are generated for each request.
3. At any time t, the expected number of requests received from one node is m. Once a request arrives, k agents are created and dispatched into the network.
4. Those agents traverse the network from the server to search for the destination. Once an agent reaches a node, it will check whether the node is its destination or not. If so, the agent will turn back to the server with information of the searched path. Otherwise, it will select a neighboring node to move on.
5. The server will compare all the path collected and pick up the optimal path. Then, the packet is sent out to the destination along the optimal path. At the same time, the server updates its routing table.
6. To avoid the user from waiting for a too long time, an agent will die if it cannot find its destination within a given time bound, which is called the agent's life-span limit in this paper.

4. Mathematical analysis

Since any component of the network (machine, link or agent) may fail at any time, we classify mobile agents into two kinds based on their reaction to a failure: weak and strong. A mobile agent with weak reaction capacity (MWRC) will die if it subjects to a failure, while one with strong reaction capacity (MSRC) will go back to the previous node, reselect another node, and go on its trip. In this section, we

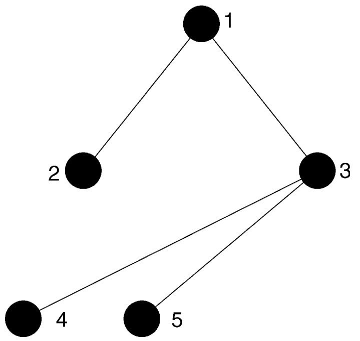

Fig. 1. An example of a small network.

analyze both the probability of success and the population distribution for each case, respectively.

Suppose that the network topology we used in this paper is a connected graph so that there is at least one path (directly or indirectly) between any two nodes. Matrix $\Phi = (\varphi_{ij})_{n\times n}$ is the connectivity matrix which describes the connectivity of the graph, i.e., if there is a direct link between node i and node j, then $\varphi_{ij} = \varphi_{ji} = 1$; otherwise, $\varphi_{ij} = 0$. Let φ_j be the j-th column vector of matrix Φ: $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_n)$. $c_j = ||\varphi_j||_1 = \sum_{i=1}^{n} |\varphi_{ij}|$, $\sigma_1 = \max_{1\le j\le n} c_j$, $\sigma_n = \min_{1\le j\le n} c_j$. $C = \mathrm{diag}(c_1, c_2, \ldots, c_n)$ is a diagonal matrix. It is easy to see that c_j is the number of neighboring nodes of the j-th node including itself, and $||\Phi||_1 = \max_{1\le j\le n} ||\varphi_j||_1 = \sigma_1$. For example, suppose that the graphical structure of a network is shown in Fig. 1. Accordingly, $n = 5$, $\sigma_1 = 4$, $\sigma_5 = 1$, matrix Φ and matrix C can be given as follows:

$$\Phi = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

4.1. The probability of success for MWRC

For a network with n nodes (i.e., $n_1, n_2, \ldots, n_n$), every node can be the destination of a request, and each node has an independent error rate. Let X_i be a binary valued variable defined as follows:

$$X_i = \begin{cases} 1, & \text{agent dies in the } i\text{-th node due to a failure;} \\ 0, & \text{otherwise,} \end{cases}$$

with a probability $P\{X_i = 1\} = p$. Then, the parameter p measures the incidence of failure in the network. We say a node is down if it is out of work; otherwise, it is up. Once a point-to-point request[1] is made, a number of agents are generated and dispatched into the network. Once an agent reaches an up node, it will find its destination locally with a probability $1/n$. If the agent cannot find its destination here, it will select a neighboring node and move on. Assume that the probability of jumping to any neighboring nodes or die in the current node is same. Regarding to the probability that an agent can find the destination in d jumps, we have the following theorem.

Theorem 1. *The probability, $P^*(n, d, p, k)$, that at least one agent among the k agents can find the destination in d jumps satisfies the following equality*:

$$P^*(n, d, p, k) = 1 - \left[1 - \frac{a(1-\tau^d)}{1-\tau}\right]^k, \qquad (1)$$

where $a = (1 - p)/n$, $b = E[1/c_i]$, *and* $\tau = (1 - a)(1 - b)$.

Proof. Denote the sequence number of node that the agent entered at i-th jump by J_i and the probability that an agent can find its destination at the i-th jump by $P(i)$. The probability that an agent can find its destination at the first jump is $P(1) = (1 - p)/n$ and the probability that it cannot find the destination is $1 - (1 - p)/n$. If the agent cannot find its destination, the probability that it can jump out and search on is $[1 - (1 - p)/n]\lfloor (c_{J_1} - 1)/c_{J_1} \rfloor$, and the probability that it can find its destination at the second jump is $P(2) = [(1 - p)/n][1 - (1 - p)/n]\lfloor (c_{J_1} - 1)/c_{J_1} \rfloor$, and the probability it cannot find the destination at the second jump is $[1 - (1 - p)/n]^2[(c_{J_1} - 1)/c_{J_1}]$. If the agent cannot find its destination at the second jump, the probability that it takes the third jump is $[1 - (1 - p)/n]^2 \lfloor (c_{J_1} - 1)/c_{J_1} \rfloor \lfloor (c_{J_2} - 1)/c_{J_2} \rfloor$ and $P(3) = [(1 - p)/n][1 - (1 - p)/n]^2[(c_{J_1} - 1)/c_{J_1}][(c_{J_2} - 1)/c_{J_2}]$. Similarly, the probability that an agent can find its destination node at the i-th jump is:

$$P(i) = \frac{1-p}{n}\left(1 - \frac{1-p}{n}\right)^{d-1} \prod_{i=1}^{d-1} \frac{c_{J_i} - 1}{c_{J_i}}.$$

[1] For point-to-multiple-point requests, the idea is intrinsic same.

Assume that the number of neighboring nodes of each node in the network is independent with each other with the same distribution, then take expectation on both side of the above equation, and denote $(1-p)/n$ by a, we have:

$$\widehat{P}(i) = a(1-a)^{i-1}\prod_{j=1}^{i-1}\left[1 - E\left(\frac{1}{c_{J_i}}\right)\right] = a(1-a)^{i-1}\left[1 - E\left(\frac{1}{c_{J_i}}\right)\right]^{i-1}.$$

Denote $E(1/c_{J_i})$ by b and $(1-a)(1-b)$ by τ, we have:
$\widehat{P}(i) = a\tau^{i-1}$.

So, the probability, $P^*(n, d, p, k)$, that at least one agent among k agents can find the destination node in d jumps satisfies the following:

$$P^*(n,\ d,\ p,\ k) = \sum_{s=1}^{k} C_k^s\left[\sum_{i=1}^{d} P(i)\right]^s\left[1 - \sum_{i=1}^{d} P(i)\right]^{k-s} = 1 - \left[1 - \sum_{i=1}^{d} P(i)\right]^k.$$

Due to

$$\sum_{i=1}^{d} P(i) = \sum_{i=1}^{d} a\tau^{i-1} = a\frac{1-\tau^d}{1-\tau},$$

we have:

$$P^*(n,\ d,\ p,\ k) = 1 - \left[1 - \frac{a(1-\tau^d)}{1-\tau}\right]^k.$$

Hence, the theorem is proven.

The value of b is depended on the probability distribution of c_i. For example, if $c_i (1 \le i \le n)$ are independent and satisfy the uniform distribution, we have:

$$b = E\left[\frac{1}{c_i}\right] = \int_1^n \left(\frac{1}{c_i}\frac{dc_i}{n-1}\right) = \frac{\ln n}{n-1}.$$

From Theorem 1, it is easy to estimate $P^*(n, d, p, k)$ as follows.

Corollary 1. *The probability $P^*(n, d, p, k)$ satisfies the following inequalities*:

$$1 - \left(\frac{1-a}{1+a\sigma_n - a}\right)^k \le \lim_{d\to\infty} P^*(n,\ d,\ p,\ k) \le 1 - \left(\frac{1-a}{1+a\sigma_1 - a}\right)^k,$$

where $a = (1-p)/n$, σ_1 and σ_n are the maximum and minimum number of c_i.

Proof. Since

$$P(d) = a(1-a)^{d-1}\prod_{i=1}^{d-1}\frac{c_{J_i}-1}{c_{J_i}} \le a(1-a)^{d-1}\prod_{i=1}^{d-1}\frac{\sigma_1 - 1}{\sigma_1} = a(1-a)^{d-1}\left(\frac{\sigma_1 - 1}{\sigma_1}\right)^{d-1},$$

we have:

$$\sum_{t=1}^{d} P(t) \le \sum_{t=1}^{d} a(1-a)^{t-1}\left(\frac{\sigma_1 - 1}{\sigma_1}\right)^{t-1} \le a\frac{1}{1-(1-a)\left(1-\frac{1}{\sigma_i}\right)} = \frac{a\sigma_1}{1+a\sigma_1 - a}.$$

Therefore,

$$P^*(n,\ d,\ p,\ k) = 1 - \left[1 - \sum_{t=1}^{d} P(t)\right]^k \le 1 - \left(\frac{1-a}{1+a\sigma_1 - a}\right)^k.$$

Similarly, the second inequality can be proved.

The probability that an agent can find its destination is decided by the connectivity of network and parameters k and d, which coincides with practice. From the theorem above, we can easily get that the probability that none of those k agents can find the destination is less than $[(1-a)/(1+a\sigma_1 - a)]^k$ and the probability that all the k agents can find the destination is less than $[(a\sigma_1)/(1+a\sigma_1 - a)]^k$ (Fig. 2).

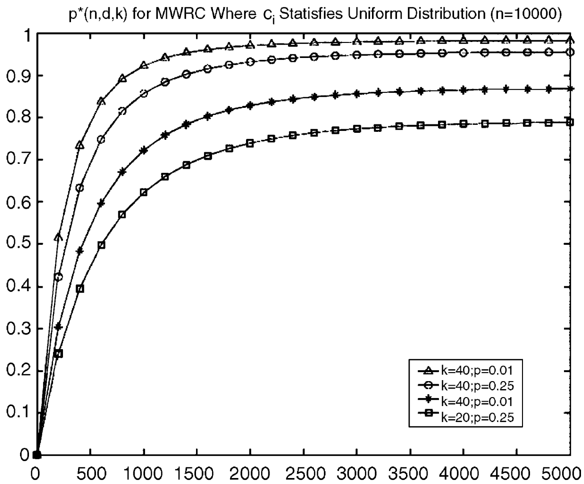

Fig. 2. The change of $P^*(n, d, p, k)$ over d where c_i satisfies uniform distribution. It is easy to see that $P^*(n, d, p, k)$ is an increasing function on d with a loose upper bound 1. When $p \neq 0$, $P^*(n, d, p, k)$ will not reach 1 no matter how long time the agent can search. The reason is that there is a possibility that the agent will die before it finds its destination. From the figure, it also can be seen that $P^*(n, d, p, k)$ is an increasing function on k and a decreasing function on p.

4.2. The probability of success for MSRC

Since for MSRC an agent will not die if it has not reached its destination within its life span, the probability of success for MSRC equals to r/n, where r is the number of nodes that the agent has entered and checked. Denote the i-th node that an agent enters by h_i, the number of neighboring nodes of the i-th node c_i, and the number of neighboring nodes that the agent has selected by v_i (i.e., the agent fails to enter the first $v_i - 1$ nodes and can only enter the v_i-th selected node). Regarding the average number of nodes selected, we have the following result.

Lemma 1. *The average number of neighboring nodes selected by an agent at each node equals to*:

$$E(v_i) = \frac{1 - E[p^{c_i}]}{1 - p} - E[c_i p^{c_i}].$$

Proof. The probability that an agent can enter the first selected node, h_i^1, in NB(i), equals to $1 - p$, and the probability that the agent can enter the second selected node equals to $p(1 - p)$. By recursion, the probability that the agent enters the v_i-th node equals $p^{v_i-1}(1 - p)$. Therefore, the average number of nodes the agent selected in NB(i) satisfies:

$$\begin{aligned} E(v_i|\mathrm{NB}(i)) &= \sum_{v_i=1}^{c_i} v_i p^{v_i-1}(1 - p) \\ &= \frac{1-p}{p}\frac{c_i p^{c_i+2} - (c_i + 1)p^{c_i+1} + p}{(1 - p)^2} \\ &= \frac{1 - p^{c_i}}{1 - p} - c_i p^{c_i}. \end{aligned}$$

Thus, the average number of nodes the agent selected at each node during the agent's trip satisfies:

$$E(v_i) = E[E(v_i|\mathrm{NB}(i))] = \frac{1 - E[p^{c_i}]}{1 - p} - E[c_i p^{c_i}].$$

Hence, the lemma is proven.

Regarding the estimation of r, we have the following result.

Lemma 2. *Let r be the number of nodes that the agent visits, then the average number of nodes that an agent enters satisfies*:

$$E(r) = \left\lfloor \frac{d}{2E(v_i) - 1} \right\rfloor,$$

where $\lfloor x \rfloor$ indicates the greatest integer less than or equal to x (i.e., $x - 1 < \lfloor x \rfloor \leq x$).

Proof. Denote the j-th selected node from the neighboring nodes of node h_i by h_i^j, the path the agent traverse from h_i to h_{i+1} can be expressed as $h_i, h_i^1, h_i, h_i^2, \ldots, h_i, h_i^{v_i}$. The v_i-th selected node is the node h_{i+1}. Inside this process, there are $2(v_i - 1) + 1$ jumps the agent takes. Since an agent will die if it cannot find its destination in d jumps, we have:

$$r = \max\left\{ l : \sum_{i=1}^{l} (2v_i - 1) \leq d \right\}.$$

Taking expectation on the inequality, we have:

$$d \geq E\left[\sum_{i=1}^{l} (2v_i - 1)\right] = E(l)[2E(v_i) - 1],$$

since l and v_i are independent to each other, and the distributions of v_i are same for $1 \leq i \leq 1$. Let $r = \{\max\}\{l\}$, then the lemma is proven.

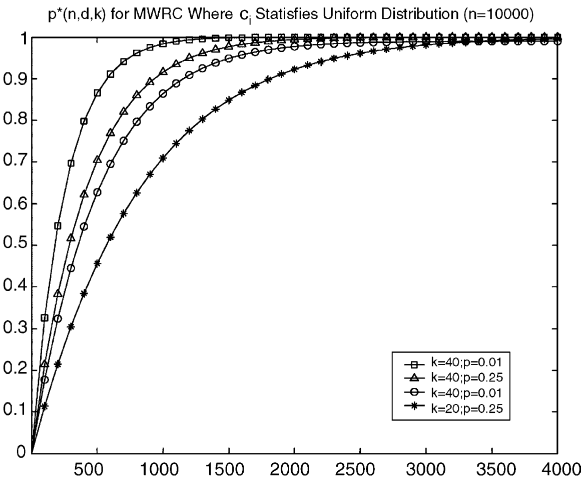

Fig. 3. The change of $P^*(n, d, p, k)$ over d where c_i satisfies uniform distribution. From the figure, it can be seen that $P^*(n, d, p, k)$ is an increasing function on k and a decreasing function on p.

Table 1
The comparison of the probability of success between MWRC and MSRC

	k			
	1	2	5	10
$n = 6000$, $p = 0.001$				
$d = 500$				
MWRC	0.0571	0.1110	0.2548	0.4447
MSRC	0.0832	0.1594	0.3522	0.5803
$d = 1000$				
MWRC	0.0826	0.1583	0.3501	0.5776
MSRC	0.1663	0.3050	0.5973	0.8378
$n = 10000$, $p = 0.001$				
$d = 500$				
MWRC	0.0391	0.0767	0.1809	0.3292
MSRC	0.0499	0.0973	0.2258	0.4006
$d = 1000$				
MWRC	0.0626	0.1213	0.2763	0.4762
MSRC	0.0998	0.1896	0.4089	0.6505

From Lemmas 1 and 2, it is readily to get the following theorem (Fig. 3).

Theorem 2. *The probability, $P^*(n, d, p, k)$, that at least one agent among k agents can find the destination in d jumps equals to $1 - [1 - E(r)/n]^k$, where $E(r) = d/(2E(v_i) - 1))$ and $E(v_i) = E[E(v_i|NB(i))] = \lfloor (1 - E\lfloor p^{c_i} \rfloor)/(1 - p) \rfloor - E\lfloor c_i p^{c_i} \rfloor$.*

Table 1 compares the probability of success $P^*(n, d, p, k)$ between MWRC and MSRC with different n, d, and k. Since a node failure is a rare event, we set $p = 0.001$ in this simulation. From the table, it can be seen that $P^*(n, d, p, k)$ for MSRC is greater than that for MWRC with the same parameters n, d, p, and k.

4.3. The population distribution of mobile agents for MWRC

Firstly, we consider the situation that the agents run in the network with infinite life span. Assume that at time $t - 1$, there are $p_i(t - 1)$ agents in the i-th node, these agents search for the destination locally, and the expected number of agents that cannot find the destination is equal to $(1 - 1/n)p_i(t - 1)$, and the expected number of mobile agents that move from the i-th node to the j-th node is equal to $(1 - 1/n)((1 - p)/c_i)p_i(t - 1)$. The total number of agents that move to the j-th node at time t is $\sum_{i \in \mathrm{NB}(j)}(1 - 1/n)((1 - p)/c_i)p_i(t - 1)$, where NB($j$) denotes the set of the neighboring nodes of the j-th node. Consider new generated agents in the j-th node at time t, we have the following equation:

$$p_j(t) = km - \Omega_j(t) + \sum_{i \in \mathrm{NB}(j)} \left(1 - \frac{1}{n}\right)\frac{1 - p}{c_i} p_i(t - 1),$$

where m is the average number of requests initiated at time t at a node, k the number of agents generated per request, and $\Omega_j(t)$ indicates the number of mobile agents on the j-th node at time t that are generated at time $t - d$. We eliminate the number of these agents from $p_j(t)$ because these agents will die at time t.

Let $A = (1 - p)(1 - 1/n)(\Phi - I)C^{-1} = (a_1, a_2, \ldots, a_n)$ be a $n \times n$ matrix, where a_j is the j-th column vector of A. Obviously, we have $||a_j||_1 = (1 - p)(1 - 1/n)(c_j - 1)/c_j$. Let $\vec{p}(t) = (p_1(t), p_2(t), \ldots, p_n(t))^T$, and $\vec{e} = (1, 1, \ldots, 1)^T$. At any time, the distribution of newly generated mobile agents is $km\vec{e}$ based on the assumption that the average number of requests received by a node is m. After searching d nodes, the distribution of survival agents among these agents is $A^d km\vec{e}$. Therefore, we have $\vec{\Omega}(t) = (\Omega_1(t), \ldots, \Omega_n(t))^T = A^d km\vec{e}$. Thus, the population distribution of mobile agents can be expressed in vector–matrix format as follows:

$$\vec{p}(t) = A\vec{p}(t - 1) + km\vec{e} - A^d km\vec{e}. \quad (2)$$

Regarding to the population distribution of mobile agents with limited life span, we have the following theorem based on the above analysis.

Theorem 3. *The distribution of mobile agents can be expressed as follows*:

$$\vec{p}(t) = \begin{cases} 0, & t = 0; \\ \sum_{i=0}^{t-1} A^{i-1} km\vec{e}, & 0 < t \le d; \\ \sum_{i=0}^{d-1} A^{i-1} km\vec{e}, & t > d. \end{cases}$$

Proof. If the distribution of mobile agents generated at time 0 is $\vec{p}(0)$, then after time t, the distribution of the agents is $A^d \vec{p}(0)$. Thus, according to the assumption that an agent will die if it cannot find the destination in d steps, we have:

(1) When $t \le d$,

$$\begin{aligned} \vec{p}(t) &= km\vec{e} + A\vec{p}(t-1) \\ &= km\vec{e} + A(km\vec{e} + A\vec{p}(t-2)) \\ &= (I + A)km\vec{e} + A^2\vec{p}(t-2) = \cdots \\ &= (I + A + \cdots + A^{t-1})km\vec{e} + A^t\vec{p}(0). \end{aligned}$$

Since the initial population of mobile agents $\vec{p}(0) = 0$, the result for $t \le d$ is proven.

(2) When $t > d$, then at time t, all the survival agents generated at time $t - d$ will die, so the distribution of agents under this case can be illustrated as:

$$\begin{aligned} \vec{p}(t) &= km\vec{e} + A\vec{p}(t-1) - A^d km\vec{e} \\ &= \sum_{i=0}^{t-d-1} A^i km\vec{e} + A^{t-d}\vec{p}(d) \\ &\quad - A^d \sum_{i=0}^{t-d-1} A^i km\vec{e} = \sum_{i=0}^{t-d-1} A^i km\vec{e} \\ &\quad + A^{t-d}\left[\sum_{i=0}^{d-1} A^i km\vec{e} + A^d\vec{p}(0)\right] \\ &\quad - A^d \sum_{i=0}^{t-d-1} A^i km\vec{e} = \sum_{i=0}^{d-1} A^i km\vec{e}. \end{aligned}$$

Hence, the theorem is proven.

From the theorem above, we can easily see that the distribution of mobile agents will not exceed $\sum_{i=o}^{d-1} A^i km\vec{e}$, that is, the number of mobile agents in our model will not increase infinitely.

Since mobile agents are generated frequently and dispatched to the network, it is important to estimate the maximum number of mobile agents running in the network and in each node. When there are too many agents in the network, they will introduce too much computational overhead to node machines, which will eventually become very busy and indirectly block the network traffic.

Regarding to the number of agents running in the network, we have the following theorem.

Theorem 4. *The number of agents running in the network can be estimated as follows*:

$$\sum_{j=1}^{n} p_j(t) \le \frac{n^2\sigma_1 km}{n + \sigma_1 - 1}.$$

Proof. By the definition of matrix 1-norm, we have:

$$\begin{aligned} \sum_{j=1}^{n} p_j(t) &= ||\vec{p}(t)||_1 \le ||km\vec{e}||_1 \cdot \left\| \sum_{i=0}^{d-1} A^i \right\|_1 \\ &\le nkm \sum_{i=0}^{d-1} (||A||_1)^i \le \frac{nkm}{1 - ||A||_1}. \end{aligned}$$

Due to $||A||_1 = (1 - p)(1 - 1/n)[(\sigma_1 - 1)/\sigma_1]$, it is easy to prove that

$$\begin{aligned} ||\vec{p}(t)||_1 &\le \frac{nkm}{1 - (1 - p)(1 - 1/n)(1 - 1/\sigma_1)} \\ &= \frac{n^2\sigma_1 km}{pn\sigma_1 + (1 - p)(n + \sigma_1 - 1)}. \end{aligned}$$

Since $n\sigma_1 \ge n + \sigma_1 - 1$, the theorem is proven.

Regarding to the number of agents running in a node, we have the following theorem.

Theorem 5. *The number of agents running in the j-th node can be estimated as follows*:

$$\begin{aligned} p_j(t) &\le km + \frac{nc_j km}{(np + 1 - p)\sigma_n}(1 - \alpha^t) \\ &\le km + \frac{nc_j km}{(np + 1 - p)\sigma_n}, \end{aligned}$$

where $\alpha = (1 - 1/n)(1 - p)$.

Proof. The theorem can be proved by induction. First, for $t = 0$, it is easy to see that the theorem is hold. Assume that for any t, the theorem is hold, that is,

$$p_j(t) \leq km + \frac{nc_jkm}{(np+1-p)\sigma_n}(1-\alpha^t) \leq km + \frac{c_jkm}{\sigma_n}km\sum_{l=1}^{t-1}\alpha^l,$$

then for $t + 1$, we have:

$$\begin{aligned} p_j(t+1) &= km + \sum_{i \in \mathrm{NB}(j)} \left(1-\frac{1}{n}\right)\frac{1-p}{c_i}p_i(t-1) \\ &= km + \alpha \sum_{i \in \mathrm{NB}(j)} \frac{p_i(t-1)}{c_i} \\ &\leq km + \alpha \sum_{i \in \mathrm{NB}(j)} \left(\frac{km}{c_i} + \frac{km}{\sigma_n}\cdot\sum_{l=1}^{t-1}\alpha^l\right) \\ &\leq km + \frac{c_jkm}{\sigma_n}\sum_{l=1}^{t}\alpha^l \\ &\leq km + \frac{nc_jkm}{(np+1-p)\sigma_n}(1-\alpha^{t+1}) \\ &\leq km + \frac{nc_jkm}{(np+1-p)\sigma_n}. \end{aligned}$$

Hence, the theorem is proven.

From the above analytical results, we can claim that both the number of mobile agents in the network and the number of mobile agents on each node will not increase infinitely over time t. The upper bounds of these two numbers can be controlled by tuning the number of mobile agents generated per request.

4.4. The population distribution of mobile agents for MSRC

For MSRC, a mobile agent will not die when it moves to a failing node. It will return back to the previous node and reselect another neighboring node to move. Thus, similar to the analysis for MWRC, the population distribution of mobile agents can be expressed as follows.

$$\vec{p}(t) = \begin{cases} 0, & t = 0; \\ km\vec{e}, & t = 1; \\ A\vec{p}(t-1) + p\cdot\vec{p}(t-2) + km\vec{e}, & 2 \leq t \leq d; \\ A\vec{p}(t-1) + p\cdot\vec{p}(t-2) + km\vec{e} - A^d km\vec{e}, & t \geq d. \end{cases} \tag{5}$$

From Eq. (5), we can estimate the number of mobile agents running in the network as follows.

Theorem 6. *The total number of mobile agents running in the network is no more than* $(n^2\sigma_1 km)/[(n+\sigma_1-1)(1-p)]$.

Proof. By the definition of vector norm, the total number of mobile agents running in the network can be expressed as $\sum_{j=1}^{n} p_j(t) = ||\vec{p}(t)||_1$. Therefore, from Eq. (5), it is easy to see that

$$||\vec{p}(t)||_1 \leq ||A||_1||\vec{p}(t-1)||_1 + p||\vec{p}(t-2)||_1 + ||km\vec{e}||_1,$$

since all the parameters a_{ji}, k, m are positive. For $t = 0$ and $t = 1$, we have:

$$\left.\begin{aligned} ||\vec{p}(0)||_1 &= 0 \\ ||\vec{p}(0)||_1 &= nkm \end{aligned}\right\} \leq \frac{n^2\sigma_1 km}{(n+\sigma_1-1)(1-p)}.$$

If the theorem holds for $t-1$ and $t-2$, then for t, we have:

$$\begin{aligned} ||\vec{p}(t)||_1 &\leq \left[(1-p)\left(1-\frac{1}{n}\right)\left(1-\frac{1}{\sigma_1}\right)+p\right] \times \frac{n^2\sigma_1 km}{(n+\sigma_1-1)(1-p)} + nkm \\ &= \frac{n^2\sigma_1 km}{(n+\sigma_1-1)(1-p)}, \end{aligned}$$

where $||A||_1 = (1-p)(1-1/n)(\sigma_1-1)/\sigma_1$. Hence, the theorem is proven.

Regarding to the maximum number of mobile agents running on a node, we have the following theorem.

Theorem 7. *The number of mobile agents running on a node is no more than* $(nc_jkm)/[(1-p)\sigma_n]$.

Proof. From Eq. (5), we know that when $t \leq d$,

$$p_j(t) = km + p \cdot p_j(t-2) + \sum_{i \in \mathrm{NB}(j)} \left(1 - \frac{1}{n}\right) \frac{1-p}{c_i} \cdot p_i(t-1).$$

Define $f_i(t) = (1 - 1/n)\lfloor(1 - p)/c_j\rfloor p_j(t)$ and substitute it in the above function, we have:

$$f_j(t) = \left(1 - \frac{1}{n}\right) \frac{1-p}{c_j} \cdot km + p \cdot f_j(t-2) + \left(1 - \frac{1}{n}\right) \frac{1-p}{c_j} \sum_{i \in \mathrm{NB}(j)} f_i(t-1).$$

By induction, we can prove that

$$f_j(t) \leq \frac{n-1}{\sigma_n} km.$$

Therefore, it can be easily proved that for all $0 \leq t \leq d$,

$$p_j(t) \leq \frac{\frac{n-1}{\sigma_n} km}{\left(1 - \frac{1}{n}\right) \frac{1-p}{c_j}} = \frac{nc_j km}{\sigma_n (1-p)}.$$

When $t \geq d$, since

$$\vec{p}(t) = A\vec{p}(t-1) + p \cdot \vec{p}(t-2) + km\vec{e} - A^d km\vec{e},$$

we have,

$$p_j(t) \leq km + p \cdot p_j(t-2) + \sum_{i \in \mathrm{NB}(j)} \left(1 - \frac{1}{n}\right) \frac{1-p}{c_i} \cdot p_i(t-1).$$

Similar to the analysis for $0 \leq t \leq d$, the upper bound $p_j(t) \leq (nc_j km)/[(1 - p)\sigma_n]$ also holds. Hence, the theorem is proven.

It is easy to see that both the total number of mobile agents running in the network and the number of mobile agents running on a node are greater than that for MWRC. The reason is because mobile agents in MWRC case have a higher death rate than in MSRC case. It also can be seen that the number of mobile agents can be justified by tuning the number of mobile agents generated per request.

5. Concluding remarks

In this paper, we addressed the problem of network routing and management by deploying mobile agents. We analyzed the probability of success and the population growth of mobile agents under our agent-based routing model. For mobile agents with weak reaction capability (MWRC), we obtained the following analytical results: (1) The probability of success, $P^*(n, d, p, k)$, that at least one agent among k agents can find the destination in d jumps equals to $1 - [1 - a(1 - c^d)/1 - c]^k$, where $a = (1 - p)/n$, $b = E[1/c_i]$, $c = (1 - a)(1 - b)$, d is the maximum number of jumps an agent can make, k is the number of agents generated per request, p is the probability that a node may fail, n is the number of nodes in the network, and c_i is the connectivity of the i-th node. (2) The total number of agents running in the network is less than $(n^2\sigma_1 km)/(n + \sigma_1 - 1)$, where $\sigma_1 = \max_{1\leq j\leq n} c_j$, and m is the average number of requests keyed in one node once. (3) The number of mobile agents running in each node, $p_j(t)$, is less then $km + [(nc_j km)/((np + 1 - p)\sigma_n)]$. For mobile agents with strong reaction capability (MSRC), we obtained the following analytical results. (1) $P^*(n, d, p, k) = 1 - [1 - E(r)/n]^k$, where $E(r) = d/(2E(v_i) - 1))$ and v_i is the number of selected nodes i-th node an agent selected. (2) $\sum_{j=1}^{n} p_j(t) \leq (n^2\sigma_1 km)/[(n + \sigma_1 - 1)(1 - p)]$. (3) $p_j(t) \leq p_j(t) \leq (nc_j km)/[(1 - p)\sigma_n]$, where $\sigma_n = \max_{1\leq j\leq n} c_j$.

We can see that the probability of success $P^*(n, d, p, k)$ is a monotonically increasing function on k and d, and a monotonically decreasing function on p and n; while the number of agents is a monotonically increasing function on k, n, d and a monotonically decreasing function on p. For the same p, routing in a smaller network may get a greater probability of success. For a network with a lot of nodes, this probability can be enlarged by increasing k and/or d. {We can also see that for the same k and p, both the probability of success and the number of agents for MWRC are less than those for MSRC.} Based on these results, we can dispatch a small number of mobile agents and achieve a good probability of success by selecting an optimal number of mobile agents generated per request and giving them an optimal life-span limit.

Acknowledgements

This research is conducted as a program for the "21st Century COE Program" by Ministry of Education, Culture, Sports, Science and Technology.

References

[1] F. Abbattista, A. Paradiso, G. Semeraro, F. Zambetta, An agent that learns to support users of a web site, Appl. Soft Comput. 4 (1) (2004) 1–12.

[2] R. Beckers, J.L. Deneubourg, S. Goss, Trails and U-turns in the selection of the shortest path by the ant Lasius niger, J. Theor. Biol. 159 (1992) 397–415.

[3] F.M.T. Brazier, B.J. Overeinder, M. Steen, N.J.E. Wijngaards, Agent factory: generative migration of mobile agents in heterogeneous environments, in: Proceedings of the ACM Symposium on Applied Computing (SAC02), 2002, pp. 101–106.

[4] B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, D. Rus, Mobile agents in distributed information retrieval, in: M. Klusch (Ed.), Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet, Springer-Verlag, Berlin, 1999 , pp. 355–395(Chapter 15).

[5] G.D. Caro, M. Dorigo, AntNet: a mobile agents approach to adaptive routing, Tech. Rep. IRIDIa/97-12, Universite Libre de Bruxelles, Belgium, 1997.

[6] J. Claessens, B. Preneel, J. Vandewalle, (How) can mobile agents do secure electronic transactions on untrusted hosts? A survey of the security issues and the current solutions, ACM Trans. Internet Technol. 3 (1) (2003) 28–48.

[7] M. Dorigo, L.M. Gambardella, Ant colonies for the traveling salesman problem, BioSystems 43 (1997) 73–81.

[8] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, 1979.

[9] S. Goss, S. Aron, J.L. Deneubourg, J.M. Pasteels, Self-organized shortcuts in the Argentine Ant, Naturwissenschaften 76 (1989) 579–581.

[10] G. Karjoth, D. Lange, M. Oshima, A security model for aglets, IEEE Internet Comput. 1 (4) (1997) 68–77.

[11] D. Lange, M. Oshima, Seven good reasons for mobile agents, Commun. ACM 42 (1999) 88–89.

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

[13] Z.J. Lee, C.Y. Lee, S.F. Su, An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem, Appl. Soft Comput. 2 (1) (2002) 39–47.

[14] T. Li, K. Lam, An Optimal Location Update and Searching Algorithm for Tracking Mobile Agent, AAMAS'02, 15–19 Bologna, Italy, July, 2002.

[15] J.H. Moore, L.W. Hahn, M.D. Ritchie, T.A. Thornton, B.C. White, Routine discovery of complex genetic models using genetic algorithms, Appl. Soft Comput. 4 (1) (2004) 79–86.

[16] W. Qu, H. Shen, J. Sum, New analysis on mobile agents based network routing, in: Proceedings of the Third International Conference on Hybrid Intelligence Systems (HIS'03), 2003, pp. 769–778 (Best Student Paper Award).

[17] R. Schoonderwoerd, O. Holland, J. Bruten, Ant-like agents for load balancing in telecommunications networks, in: Proceedings of the Agents'97, ACM Press, Marina del Rey, CA, 1997, pp. 209–216.

[18] J. Sum, H. Shen, C.S. Leung, G. Young, Analysis on mobile-agent based algorithm for network routing and management, IEEE Trans. Parallel Distributed Syst. 14 (3) (2003) 193–2002.

[19] L. Tang, B. Pagurek, A comparative evaluation of mobile agent performance for network management, in: Proceedings of the Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'02), 2002, pp. 258–267.

[20] Y. Wang, Dispatching multiple mobile agents in parallel for visiting E-shops, in: Proceedings of the Third International Conference on Mobile Data Management (MDM'02), 2002, pp. 61–68.

[21] T. White, B. Pagurek, F. Oppacher, ASGA: improving the ant system by integration with genetic algorithms, in: Proceedings of the Third Conference on Genetic Programming (GP/SGA'98), 1998, pp. 610–617.

[22] T. White, B. Pagurek, F. Oppacher, Connection management using adaptive agents, in: Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98), 1998, pp. 802–809.

[23] D. Wong, N. Paciorek, D. Moore, Java-based mobile agents, Commun. ACM 42 (1998) 92–102.