An Execution Prototype of Mobile Agent-Based Peer-to-Peer systems

Wenyu Qu[1], Masaru Kitsuregawa[1], Keqiu Li[2] and Hong Shen[3]

[1] Institute of Industrial Science
University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan
[2] College of Computer Science and Technology
Dalian Maritime University
No. 1, Linghai Road, Dalian, 116026, China
[3] School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Tatsunokuchi, Ishikawa, 923-1292, Japan

Abstract

Peer-to-peer networks are one of the trends in the field of internetworking. Mobile agent-based technology is a newly proposed technology that can be used in peer-to-peer networks. Mobile agents are small software entity that can migrate freely from node to node in heterogeneous networks and has attracted considerable interest in both academia and industry in recent years. In this paper, we focus on the problem of using mobile agents executing in peer-to-peer networks. In our proposed mobile agent-based execution model, mobile agents are classified into three kinds: inner agents, outer agents, and worker agents. Inner agents roam inside peer groups, outer agents routing among peer groups, and outer agents execute users' tasks on behalf of the users. During agents routing process, the network traffic is considered. For each step, mobile agents select their targets based on a probability distribution. We mathematically modeled the search process of mobile agents and provide some theoretical analysis on the defined cost function.

Keywords: Mobile agents, peer-to-peer, traffic information, unbiased distribution.

1 Introduction

With the far-reaching significance of the Internet and dramatic advances in computer technology, computers are no longer isolated computational machines. People communicate with the outer world through wireless networks, LANs, and the Internet. The exponential expansion of the Internet and the widespread popularity of the World Wide Web demands new paradigms for building computer systems.

Peer-to-peer systems are one of the trends in the field of internetworking. In peer-to-peer systems, the hosts, or peers, act as client and server. Milojicic et al. [18] lists some characteristics of peer-to-peer system. They are: decentralization; scalability; anonymity; self-organization; cost of ownership; ad-hoc connectivity; performance; security; transparency and usability; fault resilience; and interoperability. Currently peer-to-peer networks are mainly used for connecting nodes via largely ad hoc connections. Such networks are useful for many purposes. Sharing content files containing audio, video, data or anything in digital format is very common, and realtime data, such as telephony traffic, is also passed using P2P technology. Peer-to-peer systems can be divided into pure peer-to-peer systems and hybrid systems. In pure peer-to-peer networks, there is no kind of central servers as in hybrid model where some servers are offered for e.g. locating the resources. The most known peer-to-peer systems are probably file-sharing networks, such as Napster, kazaa, DC++, Gnutella and Bittorrent. But peer-to-peer is also used in e-commerce, distributed computing and in instant messaging (such as MSN Messenger). The concept of peer to peer is increasingly evolving to an expanded usage as the relational dynamic active in distributed networks, i.e. not just computer to computer, but human to human.

In recent years, mobile agent technology has drawn a lot of attention in both academia and industry. The use of mobile agents can be found in various areas [13], such as electronic commerce [11, 16], network management [3, 7], and information retrieval [2, 24]. As defined in [17], mobile agents are executing software entities that are capable of migrating from node to node in heterogeneous networks on behalf of network users. The key idea underlying mobile agents is to bring the computation to the data rather than the data to the computation [21]. In [14], Lange et al. concluded that mobile agents can reduce the network load, overcome network latency, encapsulate protocols, execute asynchronously and autonomously, and dynamically adapt to changes. Although some of these strengths can be realized with combinations of many traditional distributed-computing techniques, no competing technique shares all of them [10]. The merits have led a number of leading companies and research institutions to develop mobile agent systems. The existing systems include Ara, D'Agents, Aglets, Concordia, Gypsy, Mole, JatLite, Voyager and others [19].

Based on the characterization of peer-to-peer networks and the merits of mobile agents, the using of mobile agents in peer-to-peer networks will benefit a lot (see Section 2 for detail discussion). In this paper, we apply mobile agent technology in peer-to-peer applications and propose a mobile agent-based execution model. There are three kinds of agents in our model, including inner agents, outer agents, and worker agents. Inner agents and outer agents are responsible for routing, while outer agents execute users' tasks on behalf of the users. Our model is described in Section 3.

In a large agent-driven communication network, mobile agents will be generated frequently and dispatched to the network. Thus, they will certainly consume a certain amount of bandwidth of each link in the network. If there are too many agents migrating through one or several links at the same time, they will introduce too much transferring overhead to the links. Eventually, these links will be busy and indirectly block the network traffic. Therefore, there is a need of developing routing algorithms that consider about the traffic load. Since the state of different links may change dynamically over time, the agents have to dynamically adapt themselves to the environment, which increases the difficulty for both algorithm design and theoretical analysis. In [5], the network state is monitored by launching an agent at regular intervals from a source to a certain destination. In [6], the agent was enabled to estimate queuing delay without waiting inside data packet queues. In [15], the authors showed that the information needed in [5, 6] for each destination is difficult to obtain in real networks. In [20], we modeled the routing process of mobile agents by a min-max problem. In [1], a mechanism of handling routing table entries at the neighbors of crashed routers was proposed which significantly improved the algorithm proposed in [5, 6]. In [4], the authors formulated a method of mobile agent planning, which is analogous to the traveling salesman problem [9] to decide the sequence of nodes to be visited by minimizing the total execution time until the desired information is found.

In this paper, to balance the network traffic load, we introduce the maximum entropy theory to find an optimal probability distribution that both makes inference on the known traffic information and balances the traffic load. The rest of the paper is structured as follows. Section 2 presents the motivation of using mobile agents in peer-to-peer networks. Section 3 presents the proposed model. Section 4 mathematically models the cost function between two immediate nodes. Section 5 introduces the maximum entropy theory. Section 6 provides theoretical analysis and Section 7 concludes this paper.

2 Motivation of using Mobile Agents in Peer-to-Peer Systems

As summarized in [8], the reason of using mobile agents in peer-to-peer networks comes from the following aspects:

First, mobile agents reduce the need for bandwidth. Very often peers using a distributed protocol establish a communication channel between themselves, and then perform multiple interactions over this channel. Each of these interactions generates network traffic. Mobile agents allow these interactions to be packaged together, and sent as a discrete piece of network traffic. This then allows all the interactions to take place locally. Mobile agents also encapsulate all the required data within themselves. Therefore when a mobile agent arrives on a computer it has all its data with it, and does not need to communicate with any other computers. In a conventional search protocol all the raw data travels over the network to be processed, even though only a subset of this data may be needed. In this scenario, mobile agents reduce the network traffic by moving the processing to the raw data, instead of moving the raw data to the processing. Finally, mobile agents can be very small in size, but can grow dynamically as they need to accommodate more data.

Second, mobile agents are asynchronous. Therefore when a mobile agent is dispatched there is no need to wait for it to return. Indeed the original peer does not even need to remain connected to the network while the mobile agents are out. The mobile agents can wait until the original peer is back on the network before attempting to return to it.

Third, mobile agents are autonomous. This particularly suits peer-to-peer networks, because the mobile agent is learning about the network as it progresses through it. The mobile agent will visit peers that were unknown when it was originally dispatched. At each peer it can make decisions based on its history of visited peers and the current peer.

Fourth, information is being disseminated at every peer that the mobile agent visits. Every peer benefits from accepting a visiting mobile agent, because the mobile agent will have either new or more recent information about resources. Also, every mobile agent benefits from visiting a peer because it will learn of either new or updated resources . If the mobile agents do not contain any new information they may be destroyed. Accepting and hosting mobile agents requires the use of physical resources, such as memory and computer cycles. Should thee become critically limited, it is easy for the peer to refuse further requests to accept mobile agents until more physical resources become available.

Fifth, mobile agents may easily be cloned and dispatched in different directions. This allows them to function in parallel. Although this causes more mobile agents to be active on the network, it does ensure that the network resource discovery is completed sooner, and therefore the mobile agents spend less time on the network.

Sixth, a mobile agent based solution is very fault tolerant. Even if some of the mobile agents are destroyed, all surviving ones will have a positive impact. Indeed, the destroyed mobile agents will have benefited every peer up to the point where they were destroyed.

Finally, a mobile agent based solution can be combined with successful features from other peer-to-peer based systems to provide an improved final solution.

3 Execution Model

As we know, large scale distributed system consists of many different virtual organizations and private networks which often shows social or organizational structure. Therefore they do not naturally support a centralized point of control, which results in that the resource descriptors can't be stored at a single node in the system. Because of this peer-to-peer computing is well suited for this kind of systems. Agent-based computing is used because it offers the desired characteristics to implement a peer-to-peer system.

In this section, we will briefly describe the organizational principles of our model. Our algorithm focuses on how the mobile agents will behave while they roam around the network. The network is organized into peer groups. Note that a peer in a peer group is only only a peer, but also a node in the network. For a specified peer (also called a node), peers in the same peer group consists of its neighborhood and called neighbor nodes of the peer. When a node does not belongs to any peer group, we say the peer group consists of only one node.

There are three kinds of agents employed in our algorithm, namely inner agents, outer agents, and worker agents. Inner agents are periodically generated by a peer/node. They roam inside the peer group, collect and disseminate information. Outer agents are generated for when a user's request cannot be completed in the current peer group. They are dispatched into the network, roam among peer groups and search for destination peer group for executing the request. Worker agents are generated after the outer agents find the destination. They carry the requests to be completed to the destination peer group, follow a desired route that the outer agents have explored, and send the results to the users. A peer is known as the "creator peer" for the mobile agents and all future clones of these mobile agents.

Each peer periodically sends an inner agent, by broadcasting replicas of it to each neighbor node. For a inner agent, a journey-time is set in terms of the maximum number of peers that may be visited before it must return to its creator peer. Also a branching factor is set, that is used to determine how many times the inner agent may be cloned at any one peer. These two parameters control the depth and breadth of the search, and therefor the maximum number of peers that may be visited.

Initially the inner agent is given the addresses of some other peers participating in the network. Typically the number of addresses provided should be small, and should be less than the branching factor. For best results these addresses should come from different sources. This increases the chances that these peers are operating in separate sub networks that are as yet unaware of each other. After arriving at each peer the inner agent decrements its journey-time, and updates the peer with information about its creator peer, and other peers encountered along the route so far. The inner agent also updates itself with information on the current peer. If two inner agents from the same creator peer arrive at the current peer within a preset time period, then the second mobile agent destroys itself. This may lead to to less information being acquired about the network. However, it keeps the information up to date, and yet prevents peers that form cycles from having to deal repeatedly with inner agents from the same creator peer. If the inner agent's journey-time has expired, then it returns to its creator peer, and updates its creator peer with all the information it has collected on its journeys. Otherwise the inner agent clones itself enough times to allow a clone of itself to be sent to each peer that was known to the current peer before the inner agent arrived. By excluding previously known peers, the incidence of revisits is reduced.

When a peer as a node in the network receives a request, it firstly check whether the request can be fulfilled inside its peer group. If so, it generates a worker agent and forward it to the suitable site to execute the request. Otherwise, a number of outer agents are generated and launched to nodes outside its neighborhood. For a outer agent, a journey-time is also set in terms of the maximum number of nodes that may be visited before it must return to its creator peer. Each outer agent while traveling, collects and carries path information, and that it leaves, at each node visited, the trip time estimate for reaching its creator node from this node over the incoming link. Once an outer agent reaches a node, the node tells it whether the request can be fulfilled in its peer group. If so, the outer agent goes back to the creator peer along the same route it has explored, updates the routing table along its route, and submits its report of its journey. Thus each node in the network maintains current routing information for reaching nodes outside its neighborhood. This mechanism enables a node to route a data packet (whose destination is beyond the neighborhood of the creator node) along a path toward the destination node.

After a certain number of outer agents has come back, the creator peer selects a desirable route based on some rules, generates a worker agent and forward it to the destination together with the request. The worker agent executes the request on the destination node and returns back to the creator peer with the result. Finally, the creator peer submits the result to the user.

Inner agents and outer agents make routing decisions using the specialized routing strategy by selecting the next target to visit in their exploration. The routing strategy is based on the consideration of traffic balance of the network, that is, the next hop for an agent is selected in a probabilistic manner according to the quality measure of the traffic situation. We will introduce our adopted probabilistic in detail in the following.

4 Probability Distribution for Mobile Agents Routing in the Network

Let $G = \{V, E\}$ be a graph corresponding to a fixed network, where $V = \{\nu_1, \nu_2, \cdots\}$ is the set of vertices (hosts) and E is the set of edges. We assume that the topology of a network is a connected graph in order to ensure that communication are able to be made between any two host machines. For an inner agent, $NB(i)$ is the set of peers in the peer group of the creator peer ν_i and $|NB(i)|$ is the number of peers in $NB(i)$. For an outer agent, $NB(i)$ is the set of neighboring nodes outside the peer group of the creator peer ν_i and $|NB(i)|$ is the number of nodes in $NB(i)$. Originally, each peer has no information about the traffic situation of its peer group and the network. Therefore, each vertex in set $NB(i)$ has the same probability to be selected by the inner agent as a target to move to, i.e., $1/|NB(i)|$. This uniform probability distribution of agents' neighboring node-selection will be updated with time going. The new probability distribution should satisfies two constrains:

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

To find a probability distribution that both makes inference on the known information and approximates to the unbiased (uniform) distribution, We mathematically model the constrains as follows. The effect of the known traffic information on the agent's migrating decision making can be expressed by a minimum problem as follows:

$$\begin{cases} \min\limits_{j \in NB(i)} & f_i(j) \\ s.t. & g_k(i,j) \le 0, k = 1, 2, \cdots \\ & h_l(i,j) = 0, l = 1, 2, \cdots . \end{cases} \tag{1}$$

where $f_i(j)$ is a cost function that measures the cost that an agent migrates from node ν_i to ν_j, $g_k(i,j)$ and $h_l(i,j)$ are constraint functions which define the feasible region of function $f_i(j)$.

At the same time, the unbiased requirement is expressed by the maximum entropy function (as shown in the next section), which will be introduced in Section 5.

5 Maximum Entropy Theory

In [22], Shannon first introduced the concept of entropy into informatics as a measurement of uncertainty. Suppose that there are a set of possible events whose probabilities of occurrence are $\lambda_1, \lambda_2, \cdots, \lambda_n$. These probabilities are known but that is all we know concerning which event will occur. Can we find a measure of how much "choice" is involved in the selection of the event or of how uncertain we are of the outcome? Shannon pointed that if there is such a measure, say $H(\lambda_1, \lambda_2, \cdots, \lambda_n)$, it should have the following properties:

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

In [22], it is proved that the entropy function $H = -k\sum_{i=1}^{n}\lambda_i \ln \lambda_i$ is the only function that can satisfies all the requirements, where k is a positive constant decided by measurement units. Usually, k is set to be 1. The Shannon entropy has the following properties:

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

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

E. T. Jaynes found that in many probabilistic executions, the resulting probability distribution cannot foreknown; thus, the entropy cannot be calculated. But he also claimed that the probability distribution could be induced by the accumulated test data such as the mean and the variance. In [12], E.T.Jaynes proposed the maximum entropy theory: "in making inference on the basis of partial information we must use that probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we do not have". Notice that "entropy" is a measurement of the degree of uncertainty and the great the entropy's value, the less known information, the maximum entropy theory can be mathematically expressed as follows:

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

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

Templeman et al. [23] first applied maximum entropy theory to solve optimization problems in which the objective function is unanimously approximated by a smooth one. By solving the resulting problem, an approximate solution of the original problem can be obtained. The purpose of deploying maximum entropy theory in agents' searching process is to find a probability distribution that both satisfies the known routing information and mostly approximate to the unbiased (uniform) distribution.

6 Mathematical Analysis

Now, let's look at the Problem (1). Firstly, we consider about a subproblem of (1) as follows:

$$\begin{cases} \min\limits_{j \in NB(i)} f_i(j) \\ \quad s.t. \quad g_k(i,j) \leq 0, k = 1, 2, \cdots \end{cases} \tag{3}$$

where $f_i(j)$ and $g_k(i,j)$ are continuous on R^n. Denote the feasible region by $\Omega = \{j | g_k(i,j) \leq 0, i = 1, 2, \cdots\}$, the discussion will be given under the following hypothesis:

Hypothesis 1 *Set Ω for (3) satisfies int$\Omega \neq \emptyset$.*

Define function $G(j)$ as

$$G(j) = \max_k g_k(i,j), i = 1, 2, \cdots, j \in NB(i), \tag{4}$$

we have the following theorem:

Theorem 1 *Problem (5) is equivalent to (3).*

Proof Assume that j satisfies $g_k(i,j) \le 0, \forall\, k$ then $G(j) = \max_k g_k(i,j) \le 0, \forall\, k$. On the other hand, if j satisfies $G(j) = \max_k g_k(i,j) \le 0, \forall\, k$, then for any k, we have $g_k(i,j) \le G(j) \le 0$. Thus, the theorem is proven. □

Then (3) is equivalent to

$$\begin{cases} \min\limits_{j\in NB(i)} & f_i(j) \\ s.t. & G(i,j) \le 0 \end{cases} \tag{5}$$

Thus, the solving of problem (3) can be transferred into the solving of the following problem

$$\begin{cases} \max\limits_{j\in NB(i)} & \left(f_i(j), \sum\limits_{j\in(NB(i))} \lambda_{ij} g_k(ij) \right) \\ s.t. & \sum\limits_{j\in NB(i)} \lambda_{ij} = 1 \\ & \lambda_{ij} \ge 0, j \in NB(i). \end{cases} \tag{6}$$

Here, λ_{ij} discloses the possibility that node ν_j is selected as the target from the neighboring set of ν_i, $NB(i)$. Therefore, by the entropy function introduced in Section 5, we have

$$\lambda_{ij} = \frac{\exp(\theta \cdot g_k(ij))}{\sum_{j\in NB(i)} \exp[\theta \cdot g_k(ij)]} \tag{7}$$

where θ is a given constant.

Now, let's look at Problem 1. Define the feasible regions Ω_1 and Ω_2 as follows:

$$\begin{array}{l} \Omega_1 = \{j | g_k(ij) \le 0, k = 1, 2, \cdots\}, \\ \Omega_2 = \{j | h_l(ij) = 0, k = 1, 2, \cdots\} \end{array} \tag{8}$$

the feasible region of Problem 1 can be obtained as $S = \Omega_1 \bigcap \Omega_2$.

Hypothesis 2 *The feasible region of Problem 1,* $S = \Omega_1 \bigcap \Omega_2$*, satisfies* $int\Omega_1 \bigcap \Omega_2 \neq \emptyset$

Define

$$H(i,j) = \max_k h_k^2(ij) \tag{9}$$

then

$$\Omega_2 = \{j | H(i,j) = 0\} \tag{10}$$

Further define a function

$$H_\theta(i,j) = \frac{1}{\theta} \ln \left\{ \sum_k \exp\left[\theta \cdot h_k^2(ij)\right] \right\} \tag{11}$$

where θ is a given constant. Then when θ approximates to $+\infty$, function $H_\theta(i,j)$ is uniformly convergence to function $H(i,j)$. Meanwhile, these two functions satisfy the following inequality:

$$H(i,j) \le H_\theta(i,j) \le H(i,j) + \frac{\ln L}{\theta} \tag{12}$$

Here, L is the number of constrains in $H(i,j)$, that is, $l = 1, 2, \cdots, L$.

Similarly,

$$G_\vartheta(i,j) = \frac{1}{\vartheta} \ln \left\{ \sum_k \exp\left[\vartheta \cdot g_k(ij)\right] \right\} \tag{13}$$

is uniformly convergence to

$$G(i,j) = \max_k g_k(ij) \tag{14}$$

when $\vartheta \to +\infty$. Therefore, Problem (1) can be approximated by the following problem:

$$\begin{cases} \min f_i(j) \\ s.t.\ \ G_\vartheta(i,j) \le 0 \\ \qquad H_\theta(i,j) \le r\dfrac{\ln L}{\theta}, r \in (1, +\infty) \text{ is a constant.} \end{cases} \tag{15}$$

or

$$\begin{cases} \min f_i(j) \\ s.t.\ \ G_\vartheta(i,j) \le 0 \\ \qquad H_\theta(i,j) - r\dfrac{\ln L}{\theta} \le 0. \end{cases} \tag{16}$$

Let

$$E(i,j) = \max_{k,l} \left\{ g_k(ij), h_l^2(ij) \right\} \tag{17}$$

and construct an assistant function as follows

$$\begin{aligned} E_\xi(i,j) = \frac{1}{\xi} \ln \Bigg[& \sum_k \exp(\xi \cdot g_k(ij)) \\ & + \sum_l \exp(\xi \cdot h_k^2(ij))/r \Bigg] \end{aligned} \tag{18}$$

we have

$$E(i,j) \le E_\xi(i,j) \le E(i,j) + \frac{K+L}{\xi} \tag{19}$$

where K is the number of constrains in $G(i,j)$, e.g., $k = 1, 2, \cdots, K$. Therefore, the Problem (16) can be transferred as

$$\begin{cases} \min f_i(j) \\ s.t.\ \ E_\xi(i,j) \le 0 \end{cases} \tag{20}$$

which can be solved by the method similar to Problem (3) as presented above.

Definition 1 *For $\alpha \in R^1$, denote the level set by $L(\alpha) = \{j | f_i(j) \le \alpha\}$.*

Obviously, if $f_i(j)$ is a convex function, then $L(\alpha)$ is a convex set. Thus, we have the following lemma:

Lemma 1 *If $f_i(j)$ is a convex function, S is a convex set, $S \bigcap L(\alpha) \neq \emptyset$ with a boundary, then there exists a $\beta > \alpha$ such that $S \bigcap L(\beta) \neq \emptyset$ with a boundary.*

Denoted the feasible region of Problem (20) by $\Omega_\xi = \{j | E_\xi(i,j) \le 0\}$, we have the following lemma:

Lemma 2 *If $int\Omega_1 \bigcap \Omega_2 \neq \emptyset$, when for any $\xi > 0$, we have $\Omega_\xi \subset int\Omega_1 \bigcap \Omega_2$; for arbitrary $x \in int\Omega_1 \bigcap \Omega_2$, there exists $\xi_0 > 0$, such that when $\xi > \xi_0$, $j \in int\Omega_\xi$.*

Lemma 3 *If $j_\xi \in \Omega_\xi$, then when $\xi \to +\infty$, any limit point of $\{j_\xi\}$, j_0, satisfies $j_0 \in S$.*

Proof Since $j_\xi \in \Omega\xi$, $E_\xi(i, j_\xi) \le 0$. Therefore, $G_\xi(i, j_\xi) \le 0$ and $H'_\xi(i, j_\xi) \le 0$. Let

$$g(i,j) = \max_k \{g_k(ij)\}, h(i,j) = \max_l \{h_l^2(ij)\},$$

we have

$$\begin{aligned} &g(i, j_\xi) \le G_\xi(i, j_\xi) \le 0 \\ &-r\frac{\ln l}{\xi} \le h(i,j) - r\frac{\ln l}{\xi} \le H'_\xi(i, j_\xi) \le 0 \end{aligned}$$

From the continuity of $g(i,j)$ and $h(i,j)$, we have $g(i, j_0) \le 0$ and $h(i, j_0) = 0$. Thus, $j_0 \in S$, the lemma is proven. □

Lemma 4 *If $f_i(j)$, $g_k(i,j), k = 1, 2, \cdots$ are convex functions and $h_l(i,j), j = 1, 2, \cdots$ are linear functions, $E_\xi(i,j)$ must be a convex function.*

From the above hypothesis, definition, and lemmas, we can give the following theorem:

Theorem 2 *(Convergence Theorem) Suppose that in Problem (1), $f_i(j)$ and $g_k(ij), k = 1, 2, \cdots, K$ are convex function, $h_l(ij), l = 1, 2, \cdots, L$ are liner function. There is a level set $L(\alpha)$ such that $S \bigcap L(\alpha) \neq \emptyset$ and is bounded. Thus, under the Hypothesis 2, all solutions of () are bounded and any limit of this solution sequence is a solution of (1).*

Proof From Lemma 4, it is easy to see that both (1) and (20) are convex programming problems. First of all, we will prove that j_ξ is bounded by a limitation. Since $S \bigcap L(\alpha) = \emptyset$ with boundary, from Lemma 1, there is a $\beta > \alpha$ such that $S \bigcap L(\beta)$ is also unequal to $\emptyset$ with a limited boundary. Thus, there is a $j_1 \in S$ satisfies $f_i(j_1) \le \alpha < \beta$. From the Hypothesis 2, $\text{int}\Omega_1 \bigcap \Omega_2 \neq \emptyset$, there exists $j_2 \in \text{int}\Omega_1 \bigcap \Omega_2$. Therefore, for any $\lambda \in (0,1)$, we have $(1-\lambda)j_1 + \lambda j_2 \in \text{int}\Omega_1 \bigcap \Omega_2$. When λ is small enough, $(1-\lambda)j_1 + \lambda j_2 \in N(j_1, \delta_1)$. Thus, there exists a $\bar{\lambda} \in (0,1)$, such that

$$j_3 = (1-\bar{\lambda})j_1 + \bar{\lambda} j_2 \in (\text{int}\Omega_1 \bigcap \Omega_2) \bigcap N.$$

From $j_3 \in \text{int}\Omega_1 \bigcap \Omega_2$ and Lemma 2, there exists a $\xi_0 > 0$, when $\xi > \xi_0$, $j_3 \in \Omega_\xi$. If there exists an infinite subsequence, also denoted by j_ξ, such that j_ξ does not belong to $L(\beta)$, then when $\xi > \xi_0$, there is a relationship that $f_i(j_3) \ge f_i(j_\xi) \ge \beta > f_i(j_3)$, which is incompatible. Therefore, it can be proved that when ξ is big enough, $j_\xi \in L(\beta)$.
If sequence $\{j_\xi\}$ without a boundary, then there must be a subsequence of $\{j_\xi\}$, also denoted by $\{j_\xi\}$, such that $\lim_{\xi \to +\infty} |j_\xi| = +\infty$. Since $j_\xi \in \Omega_\xi$, thus $E_\xi(i, j_\xi) \le 0$. From the proof of Lemma 3, we have

$$\begin{gathered} G_\xi(i, j_\xi) \le 0 \\ -r\frac{\ln L}{\xi} \le h(i, j_{xi}) - r\frac{\ln L}{\xi} \le 0 \end{gathered}$$

Thus, $g(i, j_\xi) \le 0$. Let $D = \{j | d(j; S \bigcap L(\beta)) = 1\}$, where $d(j; S \bigcap L(\beta)) = \inf\{\|j - m\| \,|m \in S \bigcap L(\beta)\}$. As $S \bigcap L(\beta)$ is limited, D is a tight set. For j_ξ, find a $j'_\xi \in S \bigcap L(\beta)$, denote

$$\|j_\xi - j'_\xi\| = d(j_\xi; S \bigcap L(\beta))$$

Due to $\lim_{\xi \to +\infty}$, there exists a $j_\xi" \in D$ and $\lambda_\xi \in (0,1)$ such that $j_\xi" = \lambda_\xi j'_\xi + (1 - \lambda_\xi j_\xi)$. Since $j_\xi, j'_\xi \in L(\beta)$ and $L(\beta)$ is a convex set, we have $j_\xi" \in L(\beta)$. On the other hand, since $j_\xi"$ doesn't belong to the set $S \bigcap L(\beta)$, it follows by a result that $j_\xi"$ doesn't belong to S. Furthermore, since $h(i,j) = \max_k h_k^2(ij)$ is convex, we have

$$h(i, j_\xi") \le \lambda_\xi h(i, j'_\xi) + (1 - \lambda_\xi) h(i, j_\xi)$$

As $h(i, j'_\xi) = 0$, we have

$$h(i, j_\xi") \le (1 - \lambda_\xi) h(i, j_\xi)$$

Since both sequences $\{j_\xi"\}$ and $\{\lambda_\xi\}$ are bounded, they have convergent subsequences, also denoted by original symbols, and further denote $\lim_{\xi \to +\infty} j_\xi"$ by $\bar{j}$, then $\bar{j} \in D$. Besides, since $j_\xi" \in L(\beta)$ and $L(\beta)$ is a tight set, we have $\bar{j} \in L(\beta)$, therefore, $\bar{j}$ does not in S. Consider about the fact that $g(i,j)$ is a convex function, we have

$$g(i, j_\xi") \le \lambda_\xi g(i, j'_\xi) + (1 - \lambda_\xi) g(i, j_\xi).$$

Since $j'_\xi \in S$, we have $g(i, j'_\xi) \leq 0$. Combining the above two aspects, we have $g(i, j_\xi") \leq 0$, which results in a conclusion that $\bar{j} \in \Omega_1$, $\bar{j} \notin \Omega_2$, and $h(i, \bar{j}) > 0$. Take limitation on both side of $h(i, j_\xi") \leq (1 - \lambda_\xi)h(i, j_\xi)$ results in $h(i, \bar{j}) \leq 0$, which is conflicted with $h(i, \bar{j}) > 0$, therefore, it is proved that sequence $\{j_\xi\}$ is limited by a boundary.

Now we will prove that any limitation point j_0 of $\{j_\xi\}$ is the optimal solution of (1). It can be known from Lemma 3 that $j_0 \in S$. If the optimal solution of (1) is not j_0, but j^*, then we have $f_i(j_0) > f_i(j^*)$. Let $\epsilon = \frac{1}{2}(f_i(j_0) - f_i(j^*))$, since $f_i(j)$ is continuous, there exists a neighbor region of j^*, $N(j^*, \delta_2)$ such that

$$|f_i(j) - f_i(j^*)| < \epsilon$$

when $j \in N(j^*, \delta_2) \bigcap S$ and therefore

$$f_i(j) < f_i(j^*) + \epsilon = \frac{1}{2}(f_i(j^*) + f_i(j_0)) < f_i(j_0)$$

Set $\delta = \min\{\delta_1, \delta_2\}$, when $j_3 \in N(j^*, \delta) \bigcap S$, we have $f_i(j_3) < f_i(j_0)$. On the other hand, from Lemma 2, for j_3, there is a $\xi_0 > 0$. When $\xi > \xi_0$, $j_3 \in \Omega_\xi$, therefore $f_i(j_3) > f_i(j_\xi)$. From the continuity of $f_i(j)$, we have $f_i(j_3) \geq f_i(j_0)$, and thus raises a contradiction. Thus, the theorem is proven. □

7 Conclusion

In this paper, we proposed a mobile agent-based execution model for use in peer-to-peer networks in which the traffic congestion is considered. We defined a traffic cost function for each link based on known traffic information of the network by which mobile agents can unbiasedly select a neighboring node to move to. We also provided some theoretical analysis on the defined traffic cost function.

References

[1] B. Baran and R. Sosa, "A new approach for antnet routing," *Proceedings of the 9th International Conference on Computer, Communications and Networks*, 2000.

[2] F. Bergadano, A. Puliafito, S. Riccobene, and G. Ruffo, "Java-based and secure learning agents for information retrieval in distributed systems," *Information Sciences*, Vol. 113, NO. 1-2, pp.55-84, 1999.

[3] A. Bieszczad, B. Pagurek, and T. White, "Mobile Agents for Network Management," *IEEE Communication Surveys*, Vol. 1, No. 1, pp. 2-9, 1998.

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

[5] G. D. Caro G. and M. Dorigo, "AntNet: A Mobile Agents Approach to Adaptive Routing," *Technical Report IRIDIa/97-12,* Universite Libre de Bruxelles, Belgium, 1997.

[6] G. D. Caro and M. Dorigo, "Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks," *Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98)*, pp. 541-546, 1998.

[7] T.C. Du, E.Y. Li, and A.P. Chang, "Mobile Agents in Distributed Network Management," *Communications of the ACM,* Vol. 46, No. 7, pp. 127-132, 2003.

[8] C. R. Dunne, "Using Mobile Agents for Network Resource Discovery in Peer-to-Peer Networks", *SIGecom Exchanges, Newsletter of the ACM Special Interest Group on E-Commerce*, Vol. 2, No. 3, pp. 1-9, 2001.

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

[10] R. Gray, D. Kotz, G. Cybenko, and D. Rus, "Mobile Agents: Motivations and State-of-the-Art Systems," *Technical Report TR2000-365,* Dartmouth College, 2000.

[11] M. He, N.R. Jennings, and H. Leung, "On Agent-Mediated Electronic Commerce," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 15, No. 4, pp. 985-1003, 2003.

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

[13] D. Kotz and R.W. Gray, "Mobile Agents and the Future of the Internet," *ACM Operating Systems Review*, Vol. 33, No. 3, pp. 7-13, 1999.

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

[15] S. Liang, A. Zincir-Heywood, and M. Heywood, "The effect of routing under local information using a social insect metaphor," *Proceedings of the IEEE Congress on Evolutionary Computing*, pp. 1438-1443, 2002.

[16] P. Maes, R.H. Guttman, and A.G. Moukas "Agents that Buy and Sell," *Communications of the ACM*, Vol. 42, No. 3, pp. 81-91, 1999.

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

[18] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-peer Computing," *Technical Report,* HP Laboratories Palo Alto, 2002.

[19] H. Paulino, "A Mobile Agent Systems' Overview," *Technical Report DCC-2002-1,* New University of Lisbon, 2002.

[20] W. Qu and H. Shen, "Theoretical Analysis on A Traffic-Based Routing Algorithm of Mobile Agents", *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)*, pp. 520-526, 2005.

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

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

[23] A. B. Templeman and X. Li, "A maximum entropy approach to constrained non-linear programming," *Engineering Optimization*, Vol. 12, No. 3, pp. 191-205, 1987.

[24] W. Theilmann and K. Rothermel, "Optimizing the Dissemination of Mobile Agents for Distributed Information Filtering," *IEEE Concurrency,* pp. 53-61, 2000.