Cluster Computing 7, 85–90, 2004

Analysis and Design of an Agent Searching Algorithm for e-Marketplaces

CHI SING LEUNG *
Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

JOHN PUI-FAI SUM
Department of Computing, Hong Kong Polytechnic University, Hung Hom, KLN, Hong Kong

HONG SHEN
Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Tatsunokuchi, Ishikawa, Japan

JIE WU
Department of Computer Science and Engineering, Florida Atlantic University, 777 W. Glades Rd., Boca Raton, FL 33431-6498, USA

G. YOUNG
Department of Computer Science, CAL POLY, USA

Abstract. Recently, agent techniques in electronic marketplaces (e-marketplaces) bring B-to-B trading into a new era. However, not much analysis on the behavior of agents has been reported. In this paper, based on the ant algorithm in network routing, we introduce a jumping (searching) model for agents in an e-marketplace network. However, we should be aware that if there are too many agents in the e-marketplace network, they will use up all communication bandwidth and computing resource. It is inevitable to investigate the behavior of agents, such as agent population. Based on the existing results in the ant algorithm in network routing, we present the behavior of agents in an e-marketplace network. Hence, we can control the agent population by setting the appropriate agent generation rate.

Keywords: e-marketplaces, agent searching, ant algorithm

1. Introduction

An electronic marketplace (e-marketplace) [5,8], shown in figure 1, is a platform for buyers and sellers exchanging products and services. A buyer first keys in the specification and price range of buying items. The marketplace server then matches trading partners for the buyer and sends out the Request For Quotation (RFQ). Considering the specification and price range, suppliers can return the quotation to the buyer and wait for the confirmation. Once the buyer has received all quotations, the buyer can select the best offer and send back a purchasing order to the selected supplier.

As e-marketplaces can reduce the searching cost, as well as the administration cost, in both buyer and seller sides, it is expected that the trading volume via e-marketplaces will be increased. In Gartner Press Release, January 26, 2000 [8], Gartner Group predicted that the amount would hit 7.27 Trillion USD in 2004. Owing to boost the global Internet trading, many US-based e-marketplaces have been either joinventured or partnered with regional e-market organizations to make super e-marketplaces. For example, the FreeMarkets B-to-B Global Marketplace enables buying from more than 160 supply verticals and more than 9300 suppliers from over 55 countries. A number of e-marketplaces can also be connected together to form an e-marketplace network. For example, the Global Trading Web contains nearly 100 e-marketplaces (both vertical and horizontal) which forms an Internet e-marketplace community consisting of more than 7000 supplies and buyers.

Nowadays, many e-marketplace servers use intelligent agent techniques for web search, product recommendation, and even price negotiation [8,11,15]. Most related research in agent-mediated e-marketplaces focuses on brokering [1,16] and negotiations [13]. In agent brokering, research works focuses on solving the problem of matching buyers and sellers. For example, in [16], an agent-based protocol that uses multiple criteria to match buyers and sellers was proposed.

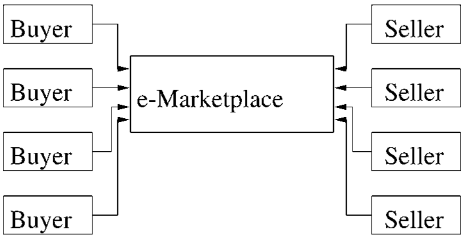

Figure 1. An e-marketplace.

[*] Corresponding author.
E-mail: eeleungc@cityu.edu.hk

Its matching process contains four stages: selection, evaluation, filtering and assignment. In agent negotiation [8], research works focuses on developing automated reasoning mechanisms to simulate human traders. For example, in [8,10], fuzzy reasoning with knowledge representing is used for modeling human behavior. In [9], a negotiation protocol based on the cooperative game theory was proposed for power markets. In [14], a fuzzy game theoretic approach was explored to model some Internet transactions, such as B-to-B price negotiation, B-to-C competition for customers, and C-to-C online auction. Besides, there are a number of other approaches for agent based negotiation [2,7,22]. For example, utility theory [4] was used for trading agents.

When using agent techniques in e-marketplaces, a large number of agents will be generated in the e-marketplace network. In this regard, it will lead to many problems, such as the server loading and the scalability problems [12]. However, less attention has been paid on the analysis of agent behavior in e-marketplaces. In this paper, we model the jumping (searching) mechanism of agents among the e-marketplace network as a modified ant algorithm [3,17,19–21], which was originally used in network routing. To avoid generating agents using up all resources in the e-marketplace network, we use the existing result [17] of the modified ant algorithm to investigate the behavior of agents in e-marketplaces, as well as to control the population of agents by setting the appropriate agent generation rate.

In section 2, the architecture of an agent mediated e-marketplace is sketched. In section 3, we first give an introduction on the ant algorithm in network routing and then model the jumping mechanism of agents in e-marketplaces as a modified ant algorithm. The behavior of agents in e-marketplaces is revealed in section 4. Design guidelines that limit the agent population and the agent migration will be summarized in section 5. The conclusion is presented in section 6.

2. Agent-mediated e-marketplaces

In short, an agent is a program that is executed on an authorized machine on behalf of the agent owner. One advantage of using agent technology is to reduce the bandwidth requirement between the agent and owner. In recent years, many researchers have suggested various designs on agent-mediated e-marketplaces [8,12]. The general idea is to make use of agent technology to help the user *easier to use* e-marketplaces services, such as looking for trading partners and bidding for an auction. Besides, a number of personalized agents also helps users narrow down the search using user-predefined profile. Every time an user logins the web-account and starts a search for trading partners or products, the agent will conduct the search automatically. The search can range from mining a single e-marketplaces to an e-marketplaces network. The search can also range from mining catalog pages to auction pages. The general purpose is to dig as much information as possible for the user.

The mechanism of supplier/buyer agents in B-to-B e-marketplaces is similar to that of an agent in B-to-C marketplaces. Once a client keys in the criteria, such as the specification and the price range of a product, the agent will automatically fetch the web for the suppliers' information from various e-marketplaces and send out the RFQ. If there is any on-going auction has been conducted in e-marketplaces, the agent may even search for the latest market price of the product and give recommendation to the user. Or it can response to the bid-price change and bid immediately.

According to the figure mentioned in the last section, the total number of listed suppliers and buyers in an e-marketplace network may be more than 170 K. This figure does not include merchants who are not listed on the trade directories but are using some software solutions. Hence, the actual number of merchants should be much greater than 170 K. If all merchants access servers simultaneously, those servers will surely be hold-up. Hence, it is necessary to set-up criteria limit the generation and dispatching agents within an e-marketplaces network. If the combinatoric auction is imposed in the e-marketplaces network, the additional load for real-time bidding will be much higher.

3. Searching operation of agents in an e-marketplace network

3.1. Ant algorithm in network routing

Network routing is a problem in network management [18]. Once a packet is required to be sent to a destination (point-to-point) or to multiple destinations (multi-cast), 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 [6], the searching for the optimal path in a faulty network or mobile network will be much more difficult. The ant routing algorithm [3,19–21] is a routing algorithm for these difficult situations. The basic idea of the ant algorithm is similar to the path searching process of an ant. Once a request for sending 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 will send back the information of the path, that it has been passed, to the source host. While the source host obtains sufficient information, it will pick up the best path. The essential idea of the algorithm can be described as follows:

(1) In a host, once a point-to-point connection request has been made, k ant agents (*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 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

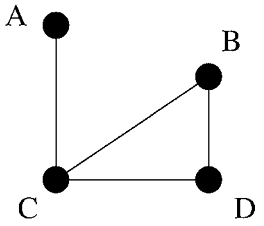

Figure 2. A simple e-marketplace network. A connection between two e-marketplaces means that the respective e-marketplaces are hyperlinked to each other.

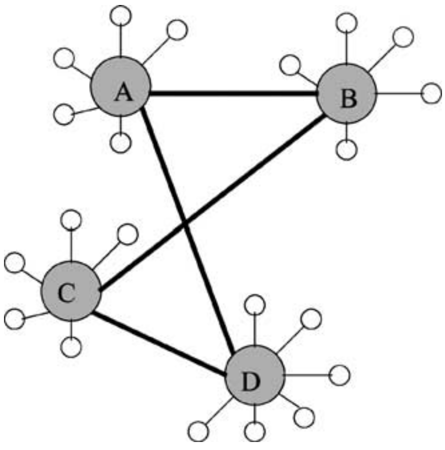

Figure 3. The connectivity of a global e-marketplace. Big circles represent regional e-marketplaces while the small circles represent merchants affiliated. If two big circles are straight-line connected to each other, the corresponding e-marketplaces are hyperlinked to each other.

node B, the allowable moves will be either A → B → C or A → B → D but not A → B → 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.

It should be noticed that the above jumping mechanism can make an explorer agent get stuck in a host. Let us take figure 2 as an example. Suppose an explorer agent is sent out from node B to node D. By chance, it moves to node C in the first jump and then moves to node A in the second jump. As the explorer agent can only move forward, it finally gets stuck in node A. In order to alleviate this problem, we modify the algorithm presented in [20] as a modified ant algorithm in [17].

3.2. E-marketplace networks and jumping mechanism of agents

Usually, an e-marketplace network (global e-marketplace) is assumed to be densely connected because each e-marketplace participated in the community must hyperlinked to each other. Figure 3 shows a simple example. Each e-marketplace will also be hyperlinked to some trading members subscribe the services provided by the e-marketplace. Suppose a search agent from e-marketplace A is going to fetch trading partners for a particular product. It will first search within e-marketplace A. After the search is finished, it will die or randomly jump to either B or D. If it jumps to B and the search at B is finished, it will die or randomly select the next visiting e-marketplace (either A or C). The above jumping mechanism is similar to an ant algorithm in network routing [17,20]. However, this jumping mechanism is one of many methods. A more intelligent method is to memorized those e-marketplaces that have been visited. Hence, the agent can select randomly from the set of e-marketplaces that has not been visited. Since extra memory space is needed for the list of visited e-marketplace and the migration model is different, a separated analysis is required. However, the forward-only jumping method can make the agent get stuck in a host. The essential idea of the jumping mechanism of agents in e-marketplaces can be summarized as follows:

(1) In a host, once a search is requested, k agents are created and sent out to the e-marketplace network. Note that $k \geqslant 1$.

(2) In an e-marketplace, an agent grasps the trading partner information.

(3) After an agent grasps the information in the current e-marketplace, it send back the information to the source e-marketplace and then migrate to a randomly selected e-marketplace that is hyperlinked to the current e-marketplace.

For those e-marketplaces that are hyperlinked to the ith e-marketplace are called the neighbor hosts of the ith e-marketplace. It is denoted by Ω_i. In ith e-marketplace, each agent will randomly select one neighbor host with probability

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

and jump to. The parameter $|\Omega_i|$ is the total number of neighbor hosts of the ith host. It should noticed that probability of an agent that can jump is equal to $|\Omega_i|/(|\Omega_i| + 1)$. That means, in the ith e-marketplace, an agent will die with probability equal to $1/(|\Omega_i| + 1)$. Hence, the agent population does not increase to infinity.

4. Agents population

Let $p_i(t)$ be the number of agents running in the ith e-marketplace at time t and $r_i(t)$ be the number of requests initiated at time t in the ith e-marketplace. Besides, we assume that the network is a connected graph. The dynamical change of agents in the e-marketplace network can be given by the following stochastic equation:

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

where $p_i(0) = 0$ for all i, $\delta_{jil}(t)$ indicates the selection of the lth agent in the ith e-marketplace, i.e.

$$\delta_{jil}(t) = \begin{cases} 1, & \text{if the } l\text{th agent select the } j\text{th e-marketplace,} \\ 0, & \text{otherwise,} \end{cases}$$

and

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

That means the lth agent selects only one neighbor $j \in \Omega_i$. In accordance with the search model,

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

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

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

$$E[p_j(t+1)] = kE[r_j(t)] + \sum_{i\in\Omega_j} \frac{1}{|\Omega_i|+1} E[p_i(t)],$$
$$\hat{p}_j(t+1) = k\hat{r}_j(t) + \sum_{i\in\Omega_j} \frac{1}{|\Omega_i|+1} \hat{p}_i(t), \qquad (2)$$

where $\hat{x}$ is denoted as the expected value of x. In a compact form, equation (2) can be rewritten as follows:

$$\hat{\vec{p}}(t+1) = A\hat{\vec{p}}(t) + k\hat{\vec{r}}(t), \qquad (3)$$

where

$$\hat{\vec{p}}(t) = \begin{bmatrix} \hat{p}_1(t) \\ \hat{p}_2(t) \\ \cdots \\ \hat{p}_N(t) \end{bmatrix}, \quad \vec{r}(t) = \begin{bmatrix} \hat{r}_1(t) \\ \hat{r}_2(t) \\ \cdots \\ \hat{r}_N(t) \end{bmatrix},$$

and $A = (a_{ji})_{N\times N}$ and

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

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

If the number of requests in each e-marketplace is stationary and each e-marketplace has the same expected value of number of requests ($\hat{r}_i(t) = m$ for all i and t), equation (3) can be written as

$$\hat{\vec{p}}(t+1) = A\hat{\vec{p}}(t) + km\vec{e}, \qquad (4)$$

where $\vec{e} = (1, 1, \ldots, 1)^T$. Since the jumping mechanism of agents is identical to the modified ant routing algorithm that has been presented in [17], the convergence result can thus be expressed by the following theorem.

Theorem 1. The expected value of number of agents running in each e-marketplace is less than or equal to $(1 + \max_i\{|\Omega_i|\})km$. If we know the spectral radius of A, denoted as $\sigma(A)$, we can get a bound on the average number of agents in the network. It is less than $N\frac{km}{1-\sigma(A)}$, where N is the number of e-marketplaces. Note that $\|A\|_\infty = \frac{\max_i\{|\Omega_i|\}}{1+\max_i\{|\Omega_i|\}} \leqslant \sigma(A)$ implies $\frac{1}{1-\sigma(A)} \leqslant (1 + \max_i\{|\Omega_i|\})$.

Proof. Firstly, the infinity norm of A, $\|A\|_\infty$, is given by

$$\|A\|_\infty = \frac{\max_i\{|\Omega_i|\}}{\max_i\{|\Omega_i|\} + 1}.$$

Hence,

$$\begin{aligned} \left\| \sum_{\tau=1}^{t} A^{t-\tau} \right\|_\infty &\leqslant \sum_{\tau=1}^{t} \|A^{t-\tau}\|_\infty \\ &\leqslant \sum_{\tau=1}^{t} \|A\|_\infty^{t-\tau} \\ &\leqslant (1 - \|A\|_\infty)^{-1} = \big(1 + \max_i\{|\Omega_i|\}\big). \end{aligned}$$

Referring to equation (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 $\vec{p}(0) = 0$. Hence,

$$\begin{aligned} \|\vec{p}(t+1)\|_\infty &\leqslant km \left\| \sum_{\tau=1}^{t} A^{t-\tau} \right\|_\infty \\ &\leqslant \big(1 + \max_i\{|\Omega_i|\}\big)km. \end{aligned}$$

In the above, if we use the spectral radius as the norm of A rather than the infinity norm, we can get the bound on the average number of agents in the network, $N\frac{km}{1-\sigma(A)}$. □

Let $Q_A(t)$ be the covariance of $\vec{p}(t)$. We can also derive the following recursive equation for $Q_A(t)$:

$$Q_A(t+1) = AQ_A(t)A^T + (km)^2 I, \qquad (5)$$

where $Q_A(0)$ is zero matrix. Since the spectral radius of A is less than 1, $Q_A(t)$ converges and the following theorem can be obtained.

Theorem 2. Each diagonal element of $Q_A(t+1)$, the spectral radius of $Q_A(t+1)$, and the infinity norm of $Q_A(t)$ are less than $\frac{(km)^2(1+\max_i\{|\Omega_i|\})^2}{(1+2\max_i\{|\Omega_i|\})}$. If we know the spectral radius of A, a more tight bound on the spectral radius of $Q_A(t+1)$ is equal to $\frac{(km)^2}{(1-(\sigma(A)^2)}$.

Proof. To prove this theorem, we only need to prove

$$\|Q_A(t+1)\|_\infty \leqslant \frac{(km)^2(1 + \max_i\{|\Omega_i|\})^2}{(1 + 2\max_i\{|\Omega_i|\})}.$$

From equation (5),

$$\begin{aligned} \|Q_A(t+1)\|_\infty &\leqslant \|A\|_\infty^2 \|Q_A(t)\|_\infty + (km)^2 I \\ &\leqslant \frac{(km)^2}{1 - \|A\|_\infty^2}. \end{aligned}$$

Hence,

$$\|Q_A(t)\|_\infty \leqslant \frac{(km)^2(1 + \max_i\{|\Omega_i|\})^2}{(1 + 2\max_i\{|\Omega_i|\})}.$$

Of course, in the above, we use the spectral radius as the norm of A, we get

$$\sigma(\|Q_A(t)\|) \leqslant \frac{(km)^2}{1-\sigma(A)^2}. \qquad \square$$

With the above two theorems, we can use some inequalities in statistics to bound the number of agents. For example, if we further assume that the distribution of agents is normal, we can use the large deviation theory to find a value p such that the probability that the number of agents is greater than p is close to zero.

5. Agents migration

Another concern in the agent-mediated e-marketplace is about the number of agents being migrated from one e-marketplace to another. If there are too many agents migrating between say A and B, it is for sure that the communication between A and B will be jammed. In [17], we have already proved a bound on the number of propagating agents in the ant routing algorithm. For agent search in e-marketplaces, we can have a similar theorem.

Theorem 3. The expected value of number of agents propagating through any communication channel is less than or equal to km.

Proof. The proof is accomplished by induction. Consider the jth row in equation (4),

$$\hat{p}_j(t+1) = km + \sum_{i\in\Omega_j} \frac{\hat{p}_i(t)}{1+|\Omega_i|}.$$

Let $\hat{f}_i(t) = \frac{\hat{p}_i(t)}{1+|\Omega_i|}$ be the average number of agents propagating in jth host from the ith host at time t, we can thus rewrite the above equation as follows:

$$\hat{f}_j(t+1) = \frac{1}{1+|\Omega_j|}\left(km + \sum_{i\in\Omega_j} \hat{f}_i(t)\right). \qquad (6)$$

Obviously, $\hat{f}_i(t) \leqslant km$ for all $i = 1, \ldots, N$ implies $\hat{f}_j(t+1) \leqslant km$. Since the initial value of $\hat{f}_i(0)$ is zero and $\hat{f}_i(1)$ is equal to $\frac{km}{1+|\Omega_i|}$ (which is smaller than km). Therefore, by the principle of induction, the proof is completed. $\square$

With this result and by definition – $\hat{f}_i(t) = \frac{\hat{p}_i(t)}{1+|\Omega_i|}$, it can readily be shown that $\hat{p}_i(t) \leqslant (1+|\Omega_i|)km$ and the following theorem.

Theorem 4. The expected value of number of agents in the ith host is less than or equal to $(|\Omega_i|+1)km$.

Unlike theorem 1, we cannot obtain a tight bound on the average number of agents in the network based on theorem 4 and the spectral radius of A. Theorem 4 shows that the upper bounds on the expected value of number of agents in an e-marketplace is depended on the local parameter $|\Omega_i|$, which is the size of the neighbor set but not depended on the global parameter $\sigma(A)$. If an e-marketplace has to be hyperlinked to the e-marketplace network, we no need to check the global parameter $\sigma(A)$. We can simply check the computational power of the machine and decide the number of hyperlinks in the partner page. Another beauty is the proof of the number of agents being propagating in a link is less than or equal to km. Therefore, simple rules for controlling agent flowing in the network and the number of agents being processed in the network machines can be obtained.

6. Conclusion

In order to provide information and guideline for the design of an effective agent algorithm in e-marketplace, this paper has attempted to model an agent search and thus analyzed its migration behavior. Using theorems being proved, we can obtain some guidelines for agent population and migration control, such as limit the agent generation rate and the number of e-marketplaces being hyperlinked in a page.

Acknowledgement

The work described in this paper was supported by a Strategic Research Grant, City University of Hong Kong (Project No. 7001126 (EE)).

References

[1] V. Almeida, W. Meira, V. Ribeiro Jr. and N. Ziviani, A quantitative analysis of the user behavior of a large e-broker, in: *Proceedings of International Conference on Advance Issues of E-Commerce and Web-Based Information Systems, WECWIS 1999* (1999) pp. 38–45.

[2] M. Barbuceanu and W.K. Lo, A multi-attribute utility theoretic negotiation architecture for electronic commerce, in: *Proceedings of the 4th International Conference on Autonomous Agents* (ACM Press, 2000) pp. 239–248.

[3] G. Di Caro and M. Dorigo, AntNet: A mobile agent approach to adaptive routing, Technical Report IRIDIA/97-12, Université Libre de Bruxelles, Belgium (1997).

[4] P. Faratin, C. Sierra and N.R. Jennings, Negotiation decision functions for autonomous agents, Journal of Robotics and Autonomous Systems 24 (1998) 159–182.

[5] S. Feldman, Electronic marketplaces, IEEE Internet Computing 4 (July–August 2000) 93–95.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, New York, 1979).

[7] R. Kowalczyk and V. Bui, On constraint-based reasoning in e-negotiation agents, in: *Agent-Mediated Electronic Commerce III*, eds. F. Dignum and U. Cortes, Lecture Notes in Artificial Intelligence, Vol. 2003 (Springer, Berlin, 2001) pp. 31–46.

[8] R. Kowalczyk, B. Franczyk, A. Speck, P. Braun, J. Eismann and W. Rossak, InterMarket – towards intelligent mobile agent e-marketplaces, in: *Proceedings of 9th Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems* (2002) pp. 268–275.

[9] V. Krishna and V.C. Ramesh, Intelligent agents for negotiation in market games, IEEE Transactions on Power Systems 13 (1998) 1103–1108.

[10] R. Lai and M.W. Lin, Modeling of agent negotiation via fuzzy constraint processing, in: *Proceedings of the 10th IEEE International Conference on Fuzzy Systems 2001*, Vol. 1 (2001) pp. 445–448.

[11] S. Murugesan, Negotiation by software agents in electronic marketplace, in: *Proceedings of TENCON 2000*, Vol. 2 (2000) pp. 286–290.

[12] H.S. Nwana et al., Agent-mediated electronic commerce: Issues, challenges and some viewpoints, in: *Proceedings of Autonomous Agents 98*, Minneapolis, USA (1998) pp. 189–196.

[13] J.S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers* (MIT Press, Cambridge, MA, 1994).

[14] S. Russell and W.A. Lodwick, Fuzzy game theory and Internet commerce: e-strategy and metarationality, in: *Proceedings of NAFIPS* (2002) pp. 93–98.

[15] C. Sierra, M. Wooldridge and N. Sadeh, Agent research and development in Europe, IEEE Internet Computing 4 (September–October 2000) 81–83.

[16] K.M. Sim and R. Chan, A brokering protocol for agent-based e-commerce, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 30 (2000) 474–484.

[17] J. Sum, H. Shen, C.S. Leung and G. Young, Analysis on mobile-agent based algorithm for network routing and management, accepted for publication in IEEE Transactions on Parallel and Distributed Systems.

[18] A.S. Tanenbaum, *Computer Networks*, 3rd edn. (Prentice-Hall, Englewood Cliffs, NJ, 1996).

[19] T. White and B. Pagurek, Towards Multi-Swarm Problem Solving in Networks, in: *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS '98)* (July 1998) pp. 333–340.

[20] T. White, B. Pagurek and F. Oppacher, ASGA: Improving the ant system by integration with genetic algorithms, in: *Proceedings of the 3rd Conference on Genetic Programming (GP/SGA '98)* (July 1998) pp. 610–617.

[21] T. White, B. Pagurek and F. Oppacher, Connection management using adaptive agents, in: *Proceedings of PDPTA'98* (July 1998) pp. 802–809.

[22] P. Wurman, W. Walsh and M. Wellman, Flexible double auctions for electronic commerce: Theory and implementation, Decision Support Systems 24 (1998) 17–27.

Chi Sing Leung received his B.Sci. in electronics from the Chinese University of Hong Kong in 1989. with a B.Sc. degree in electronics. He received his M.Phil. degree in information engineering and Ph.D. 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.
E-mail: eeleungc@cityu.edu.hk

John Pui-fai Sum received his B.Eng. 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 his M.Phil. and Ph.D. degrees in 1995 and 1998, respectively. He has published many journal and conference papers. In 1992, John received the Hong Kong Institution of Engineers Student Prize in recognition of his academic merit, extra-curricular activities, leadership qualities and personality during his undergraduate study. He has a broad research interest 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.
E-mail: cspfsum@comp.polyu.edu.hk

Hong Shen received his B.Eng. degree from Beijing University of Science and Technology, M.Eng. degree from University of Science and Technology of China, Ph.Lic. and Ph.D. 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 in Griffith University, Australia. Dr. Shen has published over 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.
E-mail: shen@jaist.ac.jp

Jie Wu is with Computer Science and Engineering Faculty, Florida Atlantic University. He is currently a Professor in the Florida Atlantic University. His research deals mainly with parallel/distributed processing and fault tolerant computing. He is currently working on parallelizing a GIS model funded by the South Florida Water Management District.
E-mail: jie@cse.fau.edu

G. Young is with Department of Computer Science, CAL POLY, USA.