Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 12 (12): 1290-1296, 2012
ISSN 1812-5654 / DOI: 10.3923/jas.2012.1290.1296

Evaluating Time Performance Optimization Analysis for Mobile Agent Message Communication Using Assignment Computing Agent

Faiz Al Shrouf, Hesham Abusaimeh, Khalil Al Shqeerat and Mahmoud Omari
Faculty of Information Technology, Applied Science University, Amman, 11931, Jordan

Abstract: This study presents an evaluation process of performance optimization between two different approaches based on mobile agent's technology. Results are simulated using wireless environment network in which different mobile agents communicate through performance messaging model. The environment involves set of master mobile agents working on clients, set of slave agents and communication messaging scheme. The first approach utilized a stationary agent that records slave response time for one message sent by a master agent. The stationary agent carries out a mathematical model based on Vogel's Approximation Method (VAM) and finally it computes the optimized time and distributes messages among master agents to their slaves. The second approach is studied via a computing agent that utilizes an Assignment Computation Development Method (ACDM) that computes the final minimized time. This approach carries out an algorithm and is analyzed by comparing optimality between agents.

Key words: Mobile agents, assignment computing agent, performance optimization, performance messaging model

INTRODUCTION

Numerical Computations for software development was introduced very early (Rice, 1993). Therefore, numerical software and development becomes complex where a set of features has to be taken into account (Dumont *et al.*, 2007; Dumont and Mourlin, 2010); efficiency: the challenge becomes to find best use of computing resources; reliability: means that resources can be unavailable and the developer does not accept to lost computational time due to hardware failure; portability: means resource ought to be used if it is not the same as the previous one; optimality: measures the performance of computational algorithms in numerical software development.

In that context, our paper presents several numerical approaches that use mobile agents to explore optimality of time minimization. This directive drove our definition of numerical platform based on mobile agent paradigm.

While many researchers investigated the development of mobile agent systems, very few proposed evaluation of the technology through actual measurement in real large scale environment implementation of such systems (Al-Shrouf *et al.*, 2008).

Performance optimization analysis has been demonstrated by the need to accommodate an increasing degree of efficiency, flexibility and adaptability. Optimization analysis for mobile agents introduces the study of message communication and computational algorithms with the aim to optimize traffic load and time management. Message communication affects communication performance while computational algorithms affect time performance.

This study presents an evaluation for different computational algorithms to optimize time performance for set of mobile agents. The evaluation process was carried out in Java environment and Aglet software development kit (Lange and Oshima, 1998). The performance evaluation was studied to compute the optimum time of the basic communication involved in the implementation of mobile agent Aglet platform on top of Java.

MOTIVATION

Agents are autonomous and problem solving software components which can carry out their tasks in open and dynamic environment. Mobile agents are a particular kind of agents, which extend the above definition by adding mobility, which is the capability to move from one host to another in a Mobile Area Network (MANET). Mobile agents decide when and where to move to discover the network, searching and filtering information (Kitsuregawa and Li, 2007) and acting as active network entities (Genco, 2008; Mekki, 2010).

Figure 1 shows mobile agents acting in network environment.

Corresponding Author: Faiz Al Shrouf, Faculty of Information Technology, Applied Science University, Amman, 11931, Jordan

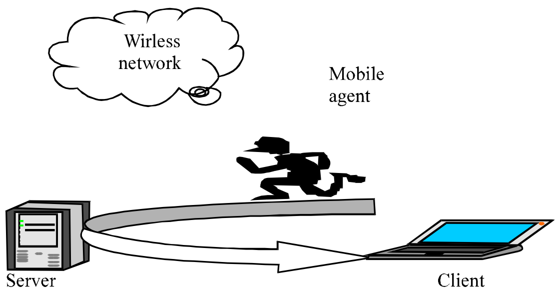

Fig. 1: Architectural diagram of mobile agents model

Communication is the base mechanism for message coordination and collaboration in mobile agent systems. However, current mobile agent systems are not based on the message content, but they focus on three components: message transport, message delivery and response time and code around messages (Shivani and Dinakaran, 2010). Our approach exploits the main capability of mobility with communication through messages. This is presented by using a set of mobile agents working on clients and set of master agents working on servers with the aim to optimize response time.

Our approach focuses on evaluating time performance optimization in mobile agent systems using different computation algorithms in operation research techniques. The first algorithm was implemented using VAM by means of V-Agent (Al-Shrouf *et al*., 2008) and the second algorithm is presented in this paper which exploits Assignment Computation Development Method (ACDM) by means of iterative computing agent. This can be used to improve the performance optimization in message time response of mobile agent systems.

PERFORMANCE OPTIMISATION MODELS

Performance evaluation for mobile agents was studied by different authors and they invented different approaches and models. Such models were investigated for transfer time optimization to improve agent communication (Dikaiakos and Samaras, 2001; Yeh and Wang, 2011). A performance model was investigated by Hagimont and Ismail (1999); (client/server and mobile agents) through two application scenario. Results show that significant time performance improvements can be obtained using mobile agents. Another performance model for communication among mobile hosts in large ad hoc wireless network was studied in (Bandyopadhyay and Paul, 1999). The evaluation in this model is concentrated on the validity of the scheme using selected arbitrary source destination and the time taken by an agent initiated by source to reach within the transmission range of the location. A six layering performance communication model ICM was introduced by Lin *et al*. (2010), to overcome the drawbacks in the model presented by Bandyopadhyay and Paul (1999). The ICM was implemented and executed on IBM Aglets, where mobile agent transfer and time is analyzed under different loads in (MANET). The results showed that communication performance time can be improved under different loads.

Several authors have presented computational algorithms for performance optimization analysis. A computational performance model called Mobile Agent Parallel Based Computing (MABPC) (Manwade and Patil, 2008), using a family of mobile agents. MABPC model uses a matrix multiplication task and divided on row wise basis, a number of available servers and dimension of block size is assigned to each mobile agent. The MABPC model experiments an implementation where a small amount of data gives little performance time than large matrices on large number of servers, which gives better performance time. MABPC model uses master-slave pattern for mobile agents in which a slave agent constructs the message and sends it to a master agent proxy. The master agent waits for the result from slave agent and it extracts the results and combines the results together. A performance messaging model was proposed by Al-Shrouf and Turani (2009) based on mobile agent systems. The optimized time was computed using Least Time computational algorithm. Furthermore, an optimized mobile agent probabilistic model was given by Ali *et al.* (2007), which uses Decision Tree Learning based on Based Routing Algorithm (ABRA). The optimized time is improved using ABRA over OSFP algorithm.

In this study, the authors present a novel algorithm that optimizes slave agents response time using the mathematical computation model developed by Al-Shrouf *et al.* (2008).

Starting from the above considerations, this study proposes an evaluation performance analysis of computational model for mobile agent message communication, where all the capabilities of mobile agents can be exploited to optimize message time in a communication system.

AGENT MODELS

Master-slave task model: Master slave task pattern (Lange and Oshima, 1998; Yang *et al*., 2002) is parallel computing application in which master agents are working on clients in order to create slave agents that dispatch to remote servers. Slave agents visit specified server to perform the required task. Master agents connect with

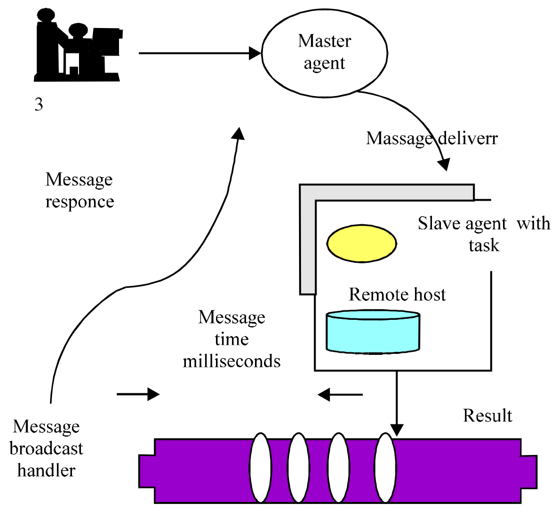

Fig. 2: Architecture of master slave task model

slave agents using proxy and sends messages in a broadcast paradigm. A user performs and submits task to master agent that divides and assigns it to individual slave agent. Each slave agent records the message time in milliseconds and informs its master by its message response time. Figure 2 shows the block architecture of master slave task pattern.

The agent-client model: Master agents are task handlers in the system. They are created and working on set of clients. The user uses Java Execution Environment (JEE) platform consists of Tahiti server and Java runtime of Aglet development Kit. After the user instantiates interface, it creates slave agents, reads the database for the list of servers and checks the availability of servers for their slaves.

The agent-server model: Slave agents are mobile agents which migrate and perform set of functions on servers. They response to master agent's messages, at remote and receive multiple messages for given tasks and send results to master agents by embedding it into message handler. Each message registered with the message handler, records response time for receiving the message and delivers it to the master agent.

ASSUMPTIONS AND CONSTRAINTS

The performance messaging model is a proposed mathematical model that based on a Message Delivery/Response Matrix (MDRM). To constitute MDRM, we considered a set of clients C_k which have a set of master agents M_k, where C_i = {C_{imi}, I = 1,2,….k}

	S_{1s1}	S_{2s2}	...	S_{nsn}	MD
C_{1m1}	t_{mlsl} λ_{mlsl}	t_{2s2l} λ_{2s2l}	...	t_{m1sn} λ_{m1sn}	MD_{m1}
C_{2m2}	t_{m2sl} λ_{m2sl}	t_{2s2l} λ_{2s2l}	...	t_{m2sn} λ_{m2sn}	MD_{m2}
⋮	⋮	⋮	⋮ ⋮ ⋮	⋮	⋮
C_{kmk}	t_{m2kl} λ_{m2kl}	t_{2k2l} λ_{2k2l}	...	t_{mksn} λ_{mksn}	MD_{mk}
MR	MR_{s1}	MR_{s2}	...	MR_{sn}	

Fig. 3: Message Delivery Response Matrix (MDRM)

represents a set of servers S_n are distributed across the network which have set of slave agents, S_j = {S_{jsj}, j = 1,2,….n}. Suppose, MD_{mi} is the total number of messages delivered from the master agent (m_i) to the slave agent carrying out tasks on server side (s_j) and MR_{sj} is the total number of response messages that a slave agent (s_j) gets from all master agents. In addition t_{misj} is the estimated response time of the slave agent (s_j) performing a task on behalf of the master agent (m_i), λ_{misj} is the number of messages delivered by a master agent (m_i) to a slave agent (s_j). Furthermore, t_{misj} is the calculated time in milliseconds of the slave (s_j) that received the message from (m_i). Based on this assumption, we propose MDRM as shown in Fig. 3.

The performance messaging model follows a linear programming model (Al-Shrouf *et al*., 2008) with set of equations:

- Determine the objective function
- Set out model constraints
- Validate non negative values

Constraints, objective function and non-negative condition are given by Eq. 1, 2, 3 and 4, respectively as follows:

$$\text{Min}\,(t)=\sum_{i=1}^{k}\sum_{j=1}^{n}\lambda_{misj}t_{ij} \tag{1}$$

Subject to (constraints):

$$\sum_{j=1}^{n}\lambda_{misj}=MD_{mi}\quad i=1,2,\ldots\ldots,k \tag{2}$$

$$\sum_{i=1}^{n} \lambda_{misj} = MR_{sj} \quad j=1,2,......,n \tag{3}$$

and:

$$\lambda_{misj} \geq 0 \text{ For all i and j} \tag{4}$$

Furthermore, we proposed that total delivered messages by all masters is equal to total response messages of MDRM, in other words:

$$\sum_{i=1}^{k} MD_{mi} = \sum_{j=1}^{n} MR_{sj}$$

Objective of this study was to simulate and optimize results based on two different agents: V-agent and Assignment Computational Agent.

PERFORMANCE OPTIMISATION ALGORITHMS

V-Agent: V-agent implemented VAM approach (Al-Shrouf *et al*., 2008) and acts as a distributer of messages from master agent to slave agents. It computes the total optimized time based on filling sells of the performance messaging model following VAM algorithm.

VAM algorithm: V-Agent implements the VAM algorithm for calculating the optimized time. V-agent suggested a good initial solution. VAM algorithm is given below:

- Find Penalty Time (PT) between two successive time values in each row and column
- Choose the largest PT amongst rows and columns
- Fill the cell with master message delivery and slave message response
- Repeat the process until all cells have been filled
- Calculate the optimal time using Eq. 1

Assignment computing agent: Assignment computing agent is an agent that follows ACDM algorithm. This agent optimizes the calculated time and best fit messages from master agents to their slaves. An iterative process is obtained from the optimization process via this agent over the V-agent.

ACDM Algorithm: The ACDM algorithm follows the Hungarian method to find the proper assignment (Hillier and Lieberman, 2005). Implementation of ACDM algorithm in MDRM proposes that number of slave agents should equal number of master agents. In case that this condition is not satisfied, we suppose a dummy master agent or slave agent with zero time entry, for every t_{misj}=0. The computational ACDM algorithm is given below:

- Subtract the minimum entry (t_{misj}) of each row from all the entries of respective row of MDRM
- Subtract the minimum entry (t_{misj}) of each column from all the entries of the respective column of MDRM
- Starting with the first row of MDRM in the first step, scan the rows one by one until a row containing one zero is found. The assignment computing agent marked this zero and it crosses all zeros in the column in which the assignment is made. The assignment computing agent should be adopted for each row assignment
- When the set of rows has been completely examined, the assignment agent scanned columns, starting with column one, scanned all columns, until a column containing one zero is found. The agent makes an experimental assignment in that position and cross other zeros in the row in which the assignment was made
- The assignment agent continues successive operations on rows and columns of MDRM until all zero's have either been crossed out
- The assignment agent excludes columns vertically or horizontally of MDRM starting with row or column contains the maximum number of zero's with minimum lines to cover zeros
- In case that number of covered lines less than n where n is the number of master agents, the assignment agent chooses the smallest uncovered number of MDRM, subtract this smallest number from all those elements which are not covered and add this element to all those elements which are at the intersection of two lines
- The assignment agent continues marking rows and columns with maximum zero's till excluded lines equals number of master agents, otherwise repeat the previous step
- If the excluded lines equals number of master the assignment agent terminates the process and fit the appropriate assignment beginning with the row or column contains the smallest number of zero's and exclude the master agent in that row with the column of slave agent
- Calculate the optimum time using Eq. 1

SIMULATION EXPERIMENTS AND EVALUATION

The following section presents a simulation experiment conducted in a wireless network lab in the faculty of Information Technology of the Applied Science

	s_1	s_2	s_3	MD
m_1	10 λ_{m1s1}	9 λ_{m1s2}	8 λ_{m1s3}	1000
m_2	8 λ_{m2s1}	6 λ_{m2s2}	11 λ_{m2s3}	1000
m_3	10 λ_{m3s1}	3 λ_{m3s2}	5 λ_{m3s3}	1000
m_4	7 λ_{m4s1}	9 λ_{m4s2}	8 λ_{m4s3}	1000
MR	1000	1200	1800	4000

Fig. 4: Slave agents response time of MDRM

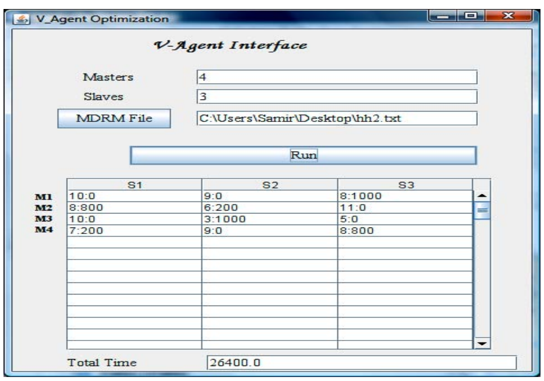

Fig. 5: Simulation results using V-Agent

University (ASU). The study proposes installation of Aglet Software Development Kit (ASDK) on set of clients and defines four master agents, three slave agents and 4000 delivered messages. Each master agent delivered 1000 messages. Slave agents records time response to each delivered message from a master agent in milliseconds. The total response messages from slave agents $\{s_1,s_2,s_3\}$ are recorded, respectively {1000,1200,1800}. Based on these assumptions, MDRM is given in Fig. 4.

Now, we simulate the results that best fit delivered messages from each master agent to the slave agents, (i.e., λ_{misj} for all j = 1, 2, 3), that minimize time according to Eq. 1.

Simulation results using V-Agent: V-agent starts the process by computing PT for rows and columns which chooses the largest PT, if more than one PT values are equals, V-agent chooses arbitrary PT. Therefore, the calculated PT is 3 in the second column has been chosen. V-agent allocates messages from the third master agent to the second slave agent $(m_3 \rightarrow s_2)$. Therefore, V-agent allocates 1000 messages from m_3 to s_2.

V-agent repeats the process till all cells is filled and the total time is calculated using Eq. 1. The optimum time is 26,400 msec according to the result of (1).

The final results of V-agent process given by our simulation are shown in Fig. 5.

Table 1: First simulation set of assignment computing agent

	s_1	s_2	s_3	s_4
m_1	3	200 3	800 0	0
m_2	2	1000 0	3	0
m_3	6	0	1000 0	3
m_4	0 1000	3	0	0

Table 2: Second simulation set of assignment computing agent

	s_1	s_2	s_3	s_4
m_1	3	3	1000 0	0
m_2	2	200 0	800 3	0
m_3	6	1000 0	0	3
m_4	0 1000	3	0	0

Simulation results using assignment computing agent: The assignment computing agent examines that number of master agents equals to number of slave agents. In last experiment a proposed slave agent is added with expected time equals to zero entry. Following the ACDM algorithm, the assignment computing agent iterates the solution and it computes two assignment sets given in Table 1 and 2, respectively:

The first simulation results are given from the assignment computing agent is: $(m_1 \rightarrow s_4)$, $(m_2 \rightarrow s_2)$, $(m_3 \rightarrow s_3)$, $(m_4 \rightarrow s_1)$. The corresponding assignment computation of MDRM given in Fig. 4 is 18. The assignment computing agent follows distributing of messages based on ACDM algorithm which is given in

Table 1. The computed time is 26,200 msec. The assignment agent computed time is optimum compared with the second simulation set.

The second simulation results are given from the assignment computing agent is: $(m_1 \rightarrow s_3)$, $(m_2 \rightarrow s_4)$, $(m_3 \rightarrow s_2)$, $(m_4 \rightarrow s_1)$. The corresponding assignment computation of MDRM given in Fig. 4 is 18. The assignment agent distributes messages from master agents to slave agents based on ACDM algorithm as given in Table 2. The computed time from the second set simulation is 28,000 msec, which is approved by the assignment agent.

Analysis, conclusions and perspectives: The first optimization analysis of V-agent (26,400 msec), is given in Fig. 5, presents results which corresponds to three slave agents. The first slave agent (s_1) performs 1000 messages in 7,800 msec when it responds to messages created from the second master agent (m_2) and the fourth master agent (m_4) respectively, while the second slave agent (s_2) performs 1200 messages in 4,200 msec when it responds to messages created from the second master agent (m_2) and the third master agent (m_3) respectively and finally, the third slave agent (s_3) carries out 1800 messages in 14,400 msec when it responds to messages created from the first master agent (m_1) and the fourth master agent (m_4).

The second optimization analysis of assignment computing agent given in Table 1 is considered the optimum computed time or best optimization (26,200 msec). The first slave agent (s_1) performs 1000 messages in 7,000 msec when it responds to messages created from the fourth master agent (m_4), while the second slave agent (s_2) carries out 1200 messages in 7,800 msec when, it responds to messages created to the first master agent (m_1) and second master agent (m_2) respectively, Finally, the third slave agent performs 1800 messages in 11,400 msec when it responds to the first master agent (m_1) and the third master agent (m_3), respectively.

The third optimization analysis of assignment computing agent given in Table 2 is the worst optimization (28,000 msec). The first slave agent (s_1) performs 1000 messages in 7,000 msec when it responds to messages created from the first master agent (m_1), while the second slave agent (s_2) performs 1200 messages in 4,200 msec when it responds to messages created from the second master agent (m_2) and the third master agent (m_3) and finally, the third slave agent responds in 16,800 msec when it responds to messages created from the first master agent and the second master agent. The analysis of the results is given in Table 3. A comparison between the three cases is presented in Fig. 6.

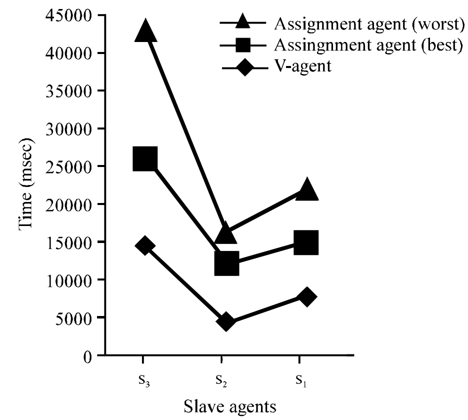

Fig. 6: V-Agent versus assignment computing agent

Table 3: Total computed time for V-Agent and assignment agent

Slaves agent	S_1	S_2	S_3
V-Agent	7,800	4,200	14,400
Assignment agent (Best)	7,000	7,800	11,400
Assignment agent (Worst)	7,000	4,200	16,800

From the analysis process, we conclude that the assignment computing agent (best case) represents an increasing function, i.e.,

$$F(s_i)=t_i \quad (5)$$

where, s_i is the slave agent that responses to messages from a master agent in time t_i. This function is normalized compared to the other functions representations of V-agent and (worst case).

As estimated, the number of response messages for the second slave agent is critical while the average response time is 6.75 msec. In addition, the number of response messages of the third slave agent is bigger and the response average time is 8 msec. The behavior of the function, which is given by Eq. 5, would prompt us to find out the following future perspectives:

Find the function that represents the assignment agent for the best case by applying numerical analysis approaches.

Use this function for time optimization forecasting that measures optimality.

Apply other operation research approaches for optimization such as sinuous path optimization or coefficient of multipliers.

ACKNOWLEDGMENT

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the generous support of this research project.

REFERENCES

Al-Shrouf, F. and A. Turani, 2009. Mobile agent optimization analysis of least time approach versus V-agent. Int. J. Comput. Intell. Res., 5: 97-108.

Al-Shrouf, F., M. Eshtay and K.A. Humaidan, 2008. Performance optimization for mobile agent message broadcast model using V-agent. Int. J. Comput. Sci. Network Security, 8: 285-291.

Ali, Y.K., H.N. Elmahdy and S.E.O.H. Ahmed, 2007. Optimizing mobile agents migration based on decision tree learning. World Acad. Sci. Eng. Technol., 33: 326-332.

Bandyopadhyay, S. and K. Paul, 1999. Evaluating the performance of mobile agent-based message communication among mobile hosts in large ad hoc wireless network. Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, August 20, 1999, Seattle, WA, USA.

Dikaiakos, M.D. and G. Samaras, 2001. Performance evaluation of mobile agents: Issues and approaches. Performance Eng., 2047: 148-166.

Dumont, C. and F. Mourlin, 2010. Jacobi computation using mobile agent. Int. J. Comput. Sci. Inform. Technol., 1: 392-401.

Dumont, C., F. Mourlin and A. Mobile, 2007. Computing architecture for numerical simulation. Proceedings of International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, November 4-9, 2007, Papeete, French Polynesia, pp: 68-74.

Genco, A., 2008. Mobile Agents: Principles of Operation and Applications. WIT Press, Southampton, UK, ISBN: 9781845640606, Pages: 271.

Hagimont, C. and L. Ismail, 1999. A performance evaluation of mobile agent paradigm. ACM SIGPLAN Notices, 34: 306-313.

Hillier, F.S. and G.J. Lieberman, 2005. Introduction to Operations Research. 8th Edn., McGraw-Hill, New York, ISBN: 9780071238281, Pages: 1061.

Kitsuregawa, W.Q. and M.K. Li, 2007. Performance analysis on mobile agent-based parallel information retrieval approaches. Int. Conf. Parallel Distrib. Syst., 2: 1-8.

Lange, D.B. and M. Oshima, 1998. Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley, Boston, MA, USA, ISBN: 9780201325829, Pages: 225.

Lin, Z., L. Xiao-Ping and Z. Chun-Xiao, 2010. An improved communication efficiency optimized model based on mobile agent. Proceedings of the International Forum on Information Technology and Applications (IFITA), July 16-18, 2010, Kunming, pp: 280-283.

Manwade, K.B. and G.A. Patil, 2008. Performance analysis of parallel client-server model versus parallel mobile agent model. World Acad. Sci. Eng. Technol., 38: 374-377.

Mekki, R., 2010. Interoperable mobile agent with JXTA. J. Applied Sci., 10: 2154-2160.

Rice, J.R., 1993. Numerical Methods, Software and Analysis. Academic Press, New York, ISBN: 9780125877558, Pages: 720.

Shivani, G. and M. Dinakaran, 2010. Forwarding of multiple messages to multiple destinations by using mobile gents. Int. J. Distrib. Parallel Syst., 1: 48-55.

Yang, B., D.Y. Liu and K. Yang, 2002. Communication performance optimization for mobile agent system. Proc. Int. Conf. Machine Learn. Cybernet., 1: 327-335.

Yeh, T.Y. and T.I. Wang, 2011. Minimizing message delivery cost in mobile agent location management. Int. J. Adv. Comput. Technol., 3: 41-51.