Strategic agents for Multi-Resource Negotiation Using Learning Automata and Case-Based Reasoning

Monireh Haghighatjoo
Faculty of Computer and Information Technology Engineering, Islamic Azad university, Qazvin Branch
Qazvin, Iran
monireh.haghighatjoo@Gmail.com

Behrooz Masoumi
Faculty of Computer and Information Technology Engineering, Islamic Azad university, Qazvin Branch
Qazvin, Iran
masoumi@Qiau.ac.ir

Mohamad reza Meybodi
Department of Computer Engineering and Information Technology, Amirkabir University
Tehran, Iran
mmeybodi@Aut.ac.ir

Abstract—**In electronic commerce markets, agents often should acquire multiple resources to fulfill a high-level task. In order to attain such resources they need to compete with each other. In multi-agent environments, in which competition is involved, negotiation would be an interaction between agents in order to reach an agreement on resource allocation and to be coordinated with each other. During recent years, many strategies have been used for negotiation; but, their performance and success are not the same in different conditions. This paper presents a method base on case-based reasoning method and learning automata for agent negotiations. In the proposed method, case-based reasoning method and learning automata are used for selecting an efficient seller and successful strategy, respectively. Results of the experiments indicated that the proposed method has caused an improvement in some performance measures such as success rate and expected utility.**

Keywords—multi agent system; resource allocation; negotiation; *learning automata; case -based reasoning*

I. Introduction

Todays, use of artificial intelligence rather than traditional approaches has improved performance of systems in electronic commerce. In E-commerce markets where agents are usually selfish, attempts are made to acquire multiple resources in order to accomplish a high-level task with the highest utility. Thus, agents would try to negotiate with others to obtain the demanded resources[1]. System of negotiation illustrated by multiple N = (Ag, S, P) where Ag is a finite set of negotiator agents, S is strategy of negotiation (negotiation strategy is act of guiding for decision making about different actions in a certain round) and P is negotiation protocol (Type of Negotiation Protocol specifies the mechanism and the specific negotiation rules it uses for a particular negotiation)[2].

In negotiations, agents employ a variety of strategies. For example Peyman Faratin et al. [3] proposed a service-oriented negotiation in 1998, in which a range of strategies and tactics such as time-dependent tactic was presented in bargaining stage for proposal and counter proposal. Then, Jennings et al. [4] expanded the service-oriented negotiation model by using genetic algorithm and presented the relative success of different tactics against different opponent in different environments. Nyugen et al. [5] had presented a heuristic model for concurrent bi-lateral negotiations in incomplete information settings in which the agents had no a priori knowledge about the preferences of their opponents. It was afterward expanded in [6] so that the ability of decision making for commitment/decommitment of the deals was integrated with the one in previous model. Because, the buyer agents might need to make deals with another agents, so the buyer could reach good deals in an efficient manner. In 2010, Bo An et al. [7] proposed a time-dependent negotiation strategy called HBA, in which the reserve price of each resource is dynamically determined by (1) the likelihood that negotiation will not be successfully completed, (2) the expected agreement price of the resource and (3) the expected number of final agreements. In 2012, Abu naser [8] found out “best solution for negotiation” by using genetic algorithm; one for the seller and one for the buyer. since the final solution found out, the agents begin the negotiation protocol time bound negotiation framework to reach a mutually acceptable agreement. In [9] a negotiation model has been presented in which sellers will propose a service to buyer. The service can be replaced with another ones in the occasions that the buyer is not satisfied with the service. Such a model could lead to validation of exploited services, security of transactions engaged with services, trustworthiness, creation and negotiation of Service Level Agreements with these services. Athirai [10] presented an appropriate seller selection method for doing transaction by querying other buyers called advisors; also, in order for the cost of querying advisors not to exceed that of a successful transaction with a seller, an optimal design was proposed for selecting advisors based on querying. Moreover, some models are environment-dependant in terms of selecting an appropriate seller and do not properly act in unknown environments. Athirai et al. [11] proposed a model for selecting an appropriate seller for unknown environments.

Some strategies act more efficiently in some conditions and environments. For example, an agent which has a limited time for reaching an agreement would have a different behavior from another agent which is not pressed for time. Furthermore, in order for the models to be applicable in practical environments, agents must be able to demonstrate flexible behaviors in different conditions, because in negotiations, agents encounter with complex environments, different deadlines and incomplete

978-1-4799-5487-2/14/$31.00 ©2014 IEEE

information about the opponents and their different preferences[4]. Thus, a negotiation system must be equipped with effective learning mechanisms to be able to adopt an appropriate strategy. Since, agents do not recognize the factor rewards that are associated with different actions in the environment, selection of the actions would be difficult task**.** Learning through setting out selection of actions of the agents based on the collected data over time, could meet such purpose. In multi-agent systems, utility of one agent is usually influenced by those of others. It will make the issue of learning in multi-agent systems more complicated, since the agents have to learn not only the effects of their actions but also, coordination manner of their actions with others. Current works have proven that learning often ends up to a coordinated behavior. Learning automata is one of the learning algorithms[12]. Considering the type of strategy that is selected, this paper aims to evaluate performance of employing case-based reasoning method and learning automata as a strategy for agents' decision making in multi-agent systems.

The remainder of this paper organized as follows: Section 2 introduces the multi-resource negotiation problem. Section 3 presents learning automata and case-based reasoning method. Section 4 presents the proposed method. Section 5 reports experimental results and presents an analysis of the properties of proposed method and Section 6 concludes the paper.

II. MULTI RESOURCE NEGOTIATION PROBLEM

The framework of this paper is inspired by a time-dependent negotiation strategy called HBA as described in Dr. Bo An et al.'s work[7], where the issue of automated negotiation for resource allocation between providers (seller) and consumers (buyer) in E-market is studied. In this model, consumer agents may require multiple resources to successfully accomplish their tasks. Therefore, they need to participate in multiple negotiations. If all of these negotiations are not successful, consumers gain nothing.

The issue of negotiation in this paper included three following features:
1. Buyer agents only knew the total reserve price (known as maximum price that could be spent for all resources).
2. Agents could decommit from tentative agreements at cost of paying a penalty.
3. Negotiation agents were assumed to have incomplete information about other agents. For example, status of the negotiations (set of proposals it received) and negotiation strategy are private information.

Fig. 1 shows *Buyer a's* multi-resource negotiation problem. Where, I = {I_1, I_2,..., I_L} is a set of resources that needed by *Buyer a* to accomplish its task and τ is *Buyer a's* negotiation deadline.

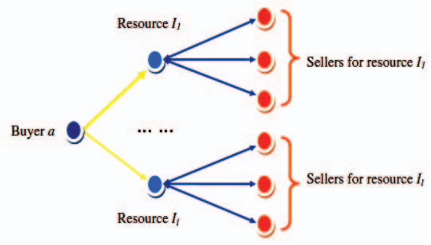


Figure 1. Buyer *a*'s multi-resource negotiation problem[7].

In this negotiation strategy, a pair of buyer and seller agents bargain by making proposals to each other. Assuming that *Buyer a* is negotiating with *Seller s* about resource I_j. First, *Buyer a* makes a proposal based on (1):

$$\varphi^{t}_{a\rightarrow s} = I\,Pj + (RP^{t}_{j} - I\,Pj\,)\delta^{t}_{j} \qquad (1)$$

Where Rp^t_j is *Buyer a's* current reserve price of resource I_j at round *t, IP_j* is *Buyer a*'s initial proposal price for resource I_j and δ^t_j is *Buyer a*'s concession rate with respect to resource I_j at round *t*. Subsequently, *Seller s* would have three choices: (1) accepting the proposal, (2) rejecting the proposal, or (3) making a counter proposal by randomly choosing a negotiation strategy from a set of tactics outlined in [3], which are the time-dependent tactic (linear, conceder, conservative), the resource-dependent tactic(impatient, steady, patient) and the behavior-dependent tactic (relative tit for tat, random tit for tat, average tit for tat). If *Seller s* accepts the proposal of *Buyer a*, negotiation terminates with a tentative agreement. If *Seller s* rejects the proposal, negotiation terminates with no agreement. If *Seller s* makes a counter-proposal, bargaining proceeds to another round and the buyer can accept and reject the proposal, or make a counter-proposal. Bargaining between two agents terminates (1) when an agreement is reached or (2) with a conflict (i.e. no agreement is made), when one of the two agents' deadline is reached or one agent quits the negotiation. Both parties are able to decommit from the tentative agreement within λ rounds after reaching an agreement by paying the penalty. Otherwise, the tentative agreement will be finalized and both parties need to fulfill all their final agreements. (see more details in [7]).

Buyer a tries to make agreements for all its resources and *buyer a* gains nothing if it fails to make an agreement for any resource in *I*. The utility function of *a* when *a* makes at least one final agreement for each resource is defined as (2):

$$U_a = RP - \sum_{I_j \in I} \sum_{Ag \in FAg_j^{\tau+\lambda}} \Pr c(Ag) + \sum_{t=0}^{\tau+\lambda} \left(\rho^{t}_{in} - \rho^{t}_{out}\right) \qquad (2)$$

where $\tau + \lambda$ is the maximum period that *buyer a* was involved in negotiation and decommitment, $Fag_j^{\tau+\lambda}$ is the set of final agreements for resource I_j at $\tau + \lambda$, ρ^t_{out} is the penalty *buyer a* pays to other agents at *t* when it decommits, and ρ^t_{out} is the payment of penalty *buyer a* receives from other agents at *t* if they decommit[7].

If *buyer a* fails to make a final agreement for at least one resource, *buyer a* gains nothing and its utility is defined as (3):

$$U_a = -\sum_{I_j \in I} \sum_{Ag \in FAg_j^{\tau+\lambda}} \Pr c(Ag) + \sum_{t=0}^{\tau+\lambda} \left(\rho_{in}^t - \rho_{out}^t\right) \qquad (3)$$

III. LEARNING IN MULTI AGENT SYSTEM

A. Case-Based Reasoning

Case-based reasoning (CBR) method that was proposed by Prof. Roger of the United States for the first time in 1982 is an important issue in the field of artificial intelligence. In CBR, there is a set of cases stored in the case-base as primary source of knowledge. The very primary idea of this method is making an experience from past and choosing the most similar case to the current problem, simply because similar problems would have similar solutions[13].

The CBR cycle is illustrated in Fig. 2:

1. Retrieve the most similar case or cases;
2. Reuse the retrieved information and knowledge;
3. Revise the proposed solution;
4. Retain the revised solution to be useful for future problem solving[14].

Re-using the results of the problems that have been solved in the past can lead to an increase in efficiency for resolving new problems rather than deriving them from the beginning. Moreover, CBR is a method for continuous incremental learning because each new experiment will be stored by resolving a problem and its results will be immediately accessible for solving the upcoming problems. This feature of CBR leads to learning. In fact learning in CBR occurs after resolving a problem. Hence, as the problem is successfully solved, its experiment will be stored for re-using in future similar problems[15].

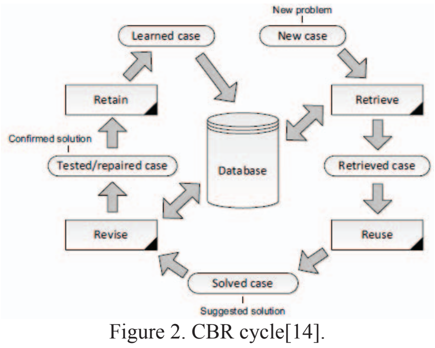


Figure 2. CBR cycle[14].

B. Learning Automata

Learning automata is a machine that can perform a finite set of actions. Each selected action will be evaluated by a random environment and a respond will be sent to learning automata. Learning automata uses such responses to select its action for the next round. The relationship between learning automata and the environment is shown in Fig. 3.

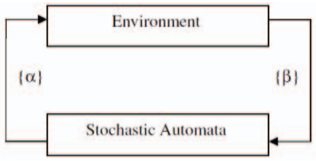


Figure 3. Relationship between learning automata and environment[16].

The ultimate aim is that automata can learn how to choose the most optimal action from its set of actions. The optimal action is the one which maximizes probability of wining reward from the environment[16]. The environment can be illustrated by E=<a,b,c>, where $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is set of inputs, $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ is set of outputs, and $c \equiv \{c_1, c_2, ..., c_r\}$ is set of penalty probabilities. When β is a two-member series, the environment would be of type p. In such an environment, $\beta_1=1$ and $\beta_2=0$ are considered as penalty and reward, respectively. In the Q-type environment, β(n) can discretely take a value within finite values of [0.1]. In the S-type environment, β(n) is a random variable within [0.1]. c_i is the probability that act α_i has an undesirable result. In static environments, c_i values remain unchanged while in non-static environments, these values change over time.

Learning automata can be classified into two main categories: (1) fixed structure learning automata and (2) variable structure learning automata[17]. In the first category, Markov chain theory is the main tool and, in the most of cases, an appropriate behavior is obtained by choosing state transaction probabilities in response to the output environment. Given that this paper used variable structure learning automata, below, are a few descriptions have given about variable structure learning automata.

Variable structure learning automata is a quintuple < *α, β, p, T(α,β,p)* >, where *α* is set of actions , *β* is an environment response set(input automata) and p is the probability set containing *r* probabilities, each being the probability of performing every action in the current internal automaton state. Function *T* is the reinforcement algorithm, which modifies the action probability vector *p* with respect to the performed action and received response. In this type of automata, if action α_i on step n is chosen, and this action has received favorable responses from the environment, probability p_i (n) increases and other probabilities are decreased. The possibility of unfavorable responses P_i(n) decreases and other probabilities are increased.

Favorable responses from the environment:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \qquad \forall j \quad j \neq i \end{aligned} \qquad (4)$$

Unfavorable responses from the environment:

$$p_i(n+1) = (1-b)p_i(n)$$
$$p_j(n+1) = (b/r-1) + (1-b)p_j(n) \quad \forall j \quad j \neq i \tag{5}$$

Where *a* and *b* are reward and penalty parameters, respectively. Values for a and b can be considered in three modes :when a and b are equal, the algorithm is called L_{RP}, when b is much less than a, the algorithm is $L_{R\varepsilon P}$ and when b is 0, the algorithm is L_{RI}[18].

IV. PROPOSED METHOD

This section presented a negotiation strategy based on CBR and learning automata, in which seller and negotiation strategy are selected based on CBR and learning automata, respectively.

To illustrate the issue of resource allocation in multi-agent systems that utilize negotiation, it can be said that the set of agents Ag = {buyer agents, seller agents}, negotiation protocol determine a set of allowable actions for each agent set including accepting the proposal, rejecting the proposal and making a counter proposal. Moreover, allowable strategies for making a counter proposal include: the time-dependent strategy (linear, conceder, conservative), the resource-dependent strategy (impatient, steady, patient) and the behavior-dependent strategy (relative tit for tat, random tit for tat, average tit for tat).

In the proposed method, learning automata are used in two steps: the first step for selecting one of the allowable actions and the second step for selecting the best strategy for making a counter proposal. In the proposed method, it is assumed that there are two types of case-bases: (1) case-base of the proposals that are accepted and (2) the case-base of the proposals that have been rejected while each case consisted of resource name, price and the resource provider's name. The buyer agent's goal is to search accepted proposals of the case-bases, find out the price of its required resources whose negotiations are successfully done in advance and finally to offer the lowest price of the selected price to the resource provider. If it does not find a similar case, it will use a HBA strategy (Equation 3) to make a proposal. Then, the Seller agent must choose one of the allowed actions based on learning automata. In this case, the agent's strategy for negotiation would be learning automata so that the seller agent could choose one of the allowed actions. Then, it will be checked whether the action is appropriate or not and, based on the given answer, automata will give a reward or a penalty to the action. Considering that the environment is p model, if this proposal is accepted and the parties reach an agreement, the output will be considered as favorable; if the offer is rejected and no agreement is reached, the output will be known as unfavorable. Afterwards, the automata updates probability of action selection based on learning automata algorithm.

The second step of using automata is as follows: if an agent selects the action of making a counter proposal at the previous step, then automata is used to select an appropriate strategy for making a counter proposal. Thus, the seller agent selects one of the strategies based on learning automata and makes a counter proposal. The agent calculates the difference between the received proposal and the counter proposal would send at this point. If this difference exceeds the satisfaction of the seller agent, then this strategy is considered as unfavorable; otherwise, it is considered as favorable and probabilities would be updated based on learning automata algorithm. Negotiation would continue in this way until agents achieve at least one final agreement for all resources or the negotiation deadline expires and it fails.

V. EXPERIMENTAL RESULTS

This paper presented a negotiation strategy based on CBR and learning automata, in which the seller agent applied learning automata in two steps and this strategy is called HBA_{LA12}. Then, this strategy is compared with other three strategies including HBA_{LA1}, HBA_{LA2}, and HBA. In HBA, which is described in Section 2, CBR and learning automata are not used. In HBA_{LA1}, CBR is used to select the appropriate seller and learning automata is only used for selecting the appropriate action from the set of allowable actions (accepting the proposal, rejecting the proposal and making a counter proposal) and the strategy for making a counter proposal is selected on a random basis. In HBA_{LA2}, CBR is applied to select the appropriate seller and learning automata is only used to select the appropriate strategy for making a counter proposal. In HBA_{LA12}, learning automata is used for selecting both of the appropriate action and an appropriate strategy for making a counter proposal.

Various parameters are considered for evaluation such as performance measures that included[7]:

- Success rate :

$$R_{suc} = N_{success} / N_{total} \tag{6}$$

Where N_{total} is total number of runs and $N_{success}$ is number of runs reaching consensus.

- Message per resource

$$M_{aver} = \frac{\sum_{i=1}^{N_{total}} \sum_{j=1}^{IS_i} M_i^j}{\sum_{i=1}^{N_{total}} IS_i} \tag{7}$$

Where N_{total} is total number of runs, IS_i is number of resources in the *ith* run and M^j_i is number of messages for resource *j* in the *ith* run. Since the number of resources that each buyer obtains at anytime could be different, comparison between the number of messages that have been sent/accepted for each resource is essential.

- Expected utility

$$U_{\exp} = \left(\sum_{i=1}^{N_{total}} U_i\right) \Big/ N_{total} \qquad (8)$$

Where N_{total} is total number of runs and U_i is utility of the *ith* run.

To evaluate such measures a number of experiments is carried, in which buyer agents are subjected to different market types, deadlines and resources.

According to Table I, when the number of agents ranges between 6 and 35, the market is sparse (similarly in the ranges of 36 to 65 and 66 to 95, in which market is moderate and dense, respectively). The deadline of an agent in e-market is randomly selected from (10, 80). In the experiments, such deadlines ranges between 10 and 30 that are considered short (similarly in the ranges between 35 and 55 and 60 and 80 which are considered moderate and long respectively). Also the number of resources each task needs, is randomly selected between 1 and 9, where 1–3 (respectively, 4–6 and 7–9) is considered as lower range (respectively, mid-range and upper range).

In the experiments the penalty function is $0.06\times prc(Ag)\times((t'-t)/\lambda)^{1/2}$ where λ (λ=4) is the decommitment period[7]. Moreover, reward and penalty parameters for learning automata are considered 0.004 and 0.0001, respectively. The automata that has being employed in this paper used reward and penalty functions of (4) and (5). In the present experiments, the different kinds of agents had the same conditions (for instance, the number of resources they needed to obtain), except that they used different negotiation strategies.

Table I. Experimental setting[7]

Market type	Favorable	Balanced	Unfavorable
Market density	Sparse	Moderate	Dense
No. of agents	6-35	36-65	66-95
Deadline	Short	Moderate	Long
T_{max}	10-30	35-55	60-80
Resources	Lower range	Mid range	High range
L	1-3	4-6	7-9

A. Observation 1

Table II shows results of the experiments for 10^6 runs. As noted in the table, HBA_{LA12} buyer agent has a higher rate of success and obtained better utility than other agents. While using learning automata and CBR, agent tries to make a proposal using past experiences that has been accepted. This process results in reaching a faster agreement among agents.

Table II. Experimental results for 10^6 runs

strategy	R_{suc}	U_{exp}	M_{aver}
HBA	0.59	0.206	86
HBA_{LA1}	0.62	0.36	83
HBA_{LA2}	0.65	0.37	84
HBA_{LA12}	0.7	0.43	80

B. Observation 2

In Fig. 4, results of the experiments show that: (1) Results of the negotiation improved for all kinds of buyers by increasing deadline; (2) Considering equal deadline, HBA_{LA12} buyer agent gain a higher expected utility than other agents, especially when the market is unfavorable and the buyer agent has sufficient time for reaching an agreement. Also, in shorter deadlines, since HBA_{LA2}, HBA_{LA12}, and HBA_{LA1} used previous experiences, therefore, they obtained higher expected utility than HBA, because in short deadline, they could make more agreements, which led to obtaining higher expected utility. Furthermore, this advantage is decreased when the market is favorable.

As demonstrated in Fig. 5: (1) By increasing the deadline, success rate of all the agent increased; (2) Success rate of HBA_{LA12} is always higher than that of other agents, especially when the market is unfavorable. Also, in the case of shorter deadline, The possibility of reaching an agreement and thus success rate of the agents which used previous experiences increased compared with HBA agent. Furthermore, by increasing the deadline, success rate of HBA_{LA12}, HBA_{LA1}, and HBA_{LA2} agents has much higher increase than HBA agent, because when the deadline is longer, buyers not only use past experiences and learning, but also have sufficient deadline for reaching an agreement.

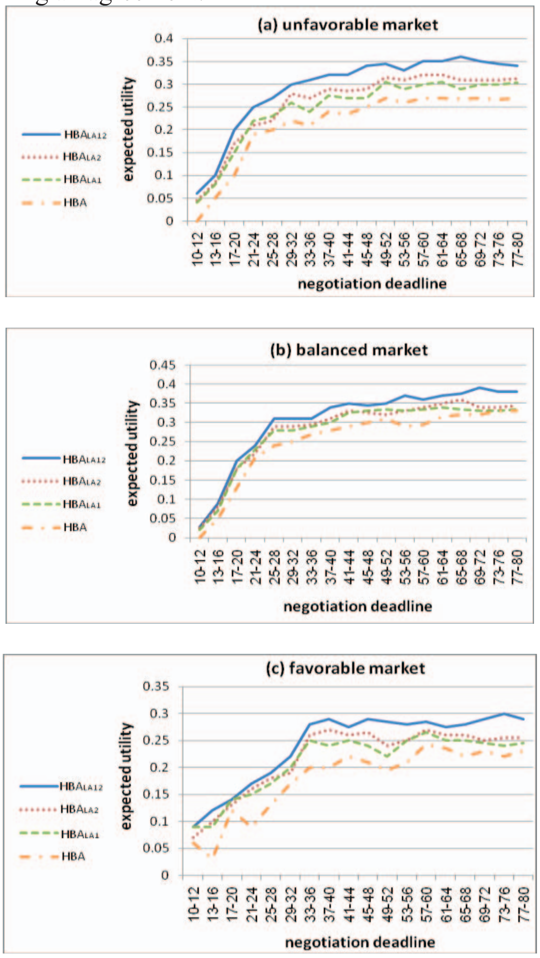


Figure 4. Deadline and expected utility (a: unfavorable market b: balanced market c: favorable market)

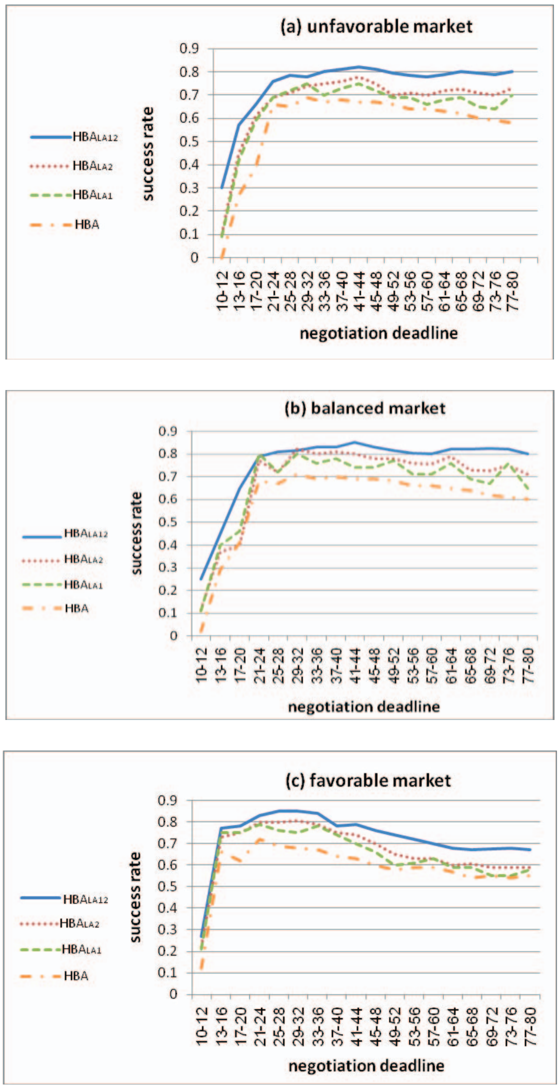


Figure 5. Deadline and success rate (a: unfavorable market b: balanced market c: favorable market)

In Fig. 6, in all the strategies, the more the number of required resources, the less the agent expected utility would be. In longer deadlines, since agents have more time for making the agreement and use their learning, therefore, they make more utility than the time they are faced with shorter deadlines.

Fig. 7 shows that, the more the number of resources required for fulfill the task, the less the success rate of agents would be, because it is difficult for the agent to manage a large number of resources and the failure possibility of negotiations which is directly related to success rate. Also, when the deadline is short, the agent does not have enough time to make an agreement; however, the strategies which use past experience and learning have higher success rate than HBA and, in longer deadlines, such a success rate for learning-based strategies has more increase than HBA.

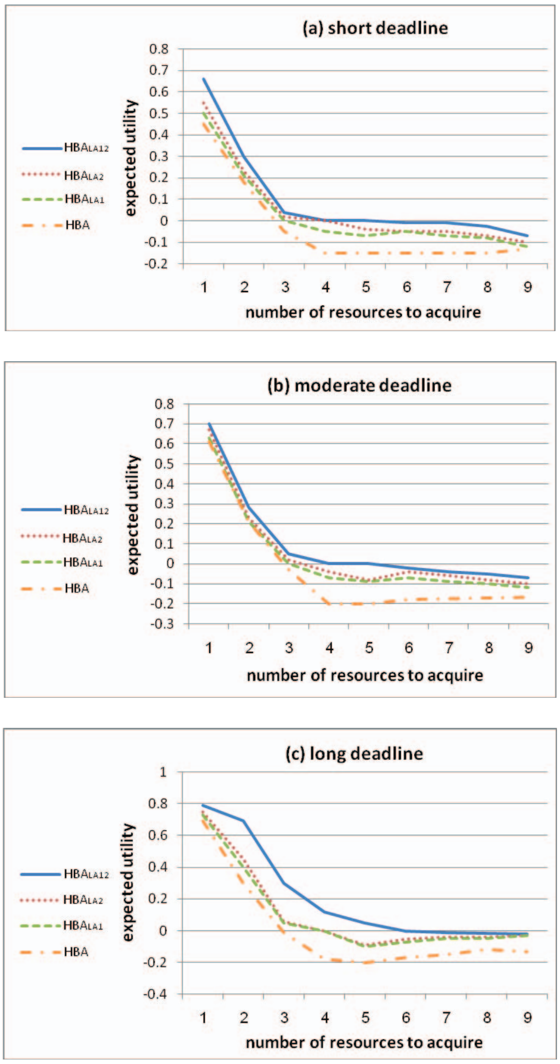
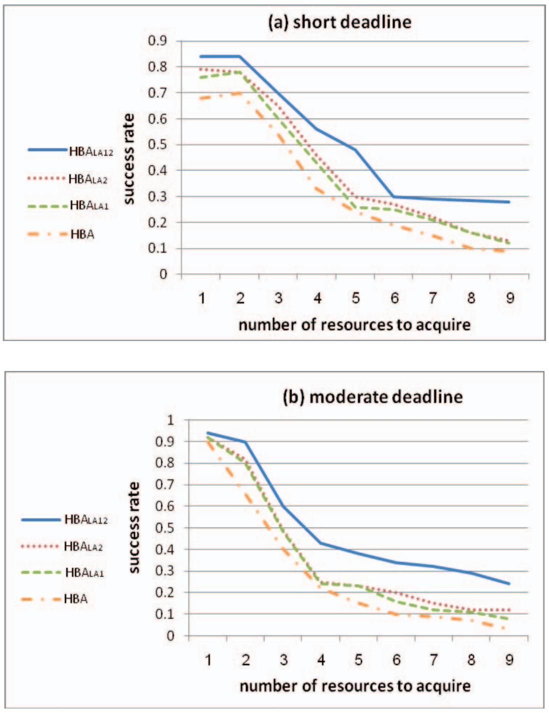


Figure 6. Number of resources and expected utility (a: short deadline b: moderate deadline c: long deadline)



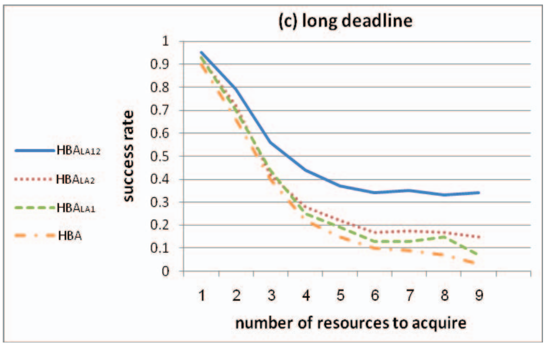


Figure 7. Number of resources and success rate (a: short deadline b: moderate deadline c: long deadline)

VI. CONCLUSION

This work proposes an approach to multiple resource negotiation, which is designed to assist agents in acquiring multiple resources to successfully fulfill high-level tasks in electronic commerce markets. In order to improve negotiation efficiency, this approach employs case-based reasoning and learning automata; especially, case-based reasoning is used to select appropriate resources sellers by reusing an experience from past and choosing the most similar case to the current task, and learning automata helps make negotiation actions and select negotiation strategies among a set of strategies from the previous work [Faratin 1998]. Experimental results provide by the work show that the proposed method has made an improvement in three performance measures (success rate, the number of messages and expected utility) over several other candidate methods. it is possible to develop a set of strategies and make use of new negotiation strategies that have been proposed to tackle bilateral automated negotiation in order to make counter proposal. Additionally, by utilizing learning automata agents would be able to learn how to choose a combination of strategies instead of using a single strategy for making the counter proposal. This method can be employed in other environments such as cloud computing and be extended in other issues rather that price solely.

REFERENCES

[1] Guoling Lao, and Chengkun Zhong,"A Negotiation Model and its Simulation based on Qlearning Algorithm," IEEE, Computational Intelligence and Software Engineering (CiSE), 2010, pp. 1-4.

[2] Ahmad Abdolahzade, Behrooz Masumi, and Mohamadreza Ayatollah zadeh, An Introduction for Allocated Artificial Intelligence, 1 th ed,Tehran, iran, 2008.

[3] Peyman faratin, carles sierra, and nich r.jennings, "Negotiation decision functions for autonomous Agents," International Journal of Robotics and Autonomous Systems, 1998, pp.159–182.

[4] Noyda Matos, Carles Sierra, and Nick R. Jennings, "Determining Successful Negotiation Strategies: An Evolutionary Approach," international conference in Paris, france, 1998, pp.182-189.

[5] Nguyen, T. D., and Jennings, N. R., " A heuristic model of concurrent bi-lateral negotiations in incomplete information settings," In Proceedings of the eighteenth international joint conference on artificial intelligence, 2003, pp. 1467–1469.

[6] Nguyen, T. D., and Jennings, N. R., "Managing commitments in multiple concurrent negotiations," Electronic Commerce Research and Applications, pp. 362–376.

[7] Bo An, Victor Lesser, and Kwang Mong Sim, "Strategic agents for multi-resource negotiation," Journal of Autonomous Agents and Multi-Agent Systems, Vol.23, 2011, pp.114-153.

[8] Amal M. Abu Naser, "Genetic Algorithm Strategies and Tactics Model for Agent Negotiation in E-Commerce Systems," ABHATH AL-YARMOUK: "Basic Sci. & Eng." Vol. 21, No. 1, 2012, pp. 51-58.

[9] Elisabetta Di Nitto, and Massimiliano Di Penta, "Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach," Springer-Verlag Berlin Heidelberg, ICSOC 2007, LNCS 4749, pp. 295–306, 2007.

[10] Athirai A. Irissappane, "Robust Trust Management," (Doctoral Consortium) Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.

[11] Athirai A. Irissappane, Siwei Jiang, and Jie Zhang, "A Framework to Choose Trust Models for Different E-Marketplace Environments," proceeding IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, pp 213-219, 2013.

[12] G. Chalkiadakis, and Boutilier C., "Coordination in Multiagent Reinforcement Learning: A Bayesian Approach," Proceedings of 2nd Intl. Conf. on Autonomous Agents and Multi agent Systems (AAMAS-03), 2003, pp. 709-716.

[13] Guorui Jiang, and Lin Wu, "Research on Method of Multi-Agent Negotiation Strategy Selection," IEEE, In Computing in the Global Information Technology (ICCGI), 2010 Fifth International Multi-Conference on,2010, pp. 110-115.

[14] Ivo Pereira and Ana Madureira, "Self-Optimizing through CBR Learning" ,IEEE, evolutionary computation (CEC), 2010, PP.1-8.

[15] S. Ghanbari, M.R.Meybodi, and K. Badie, "A Case-Based Recommender for Task Assignment in Heterogeneous Computing Systems," Proceedings of the Fourth IEEE International Conference on Hybrid Intelligent Systems, 2004, pp. 110- 115.

[16] K. Narenrdra, S., M. A. L. Thathachar, "Learning Automata: An Introduction," Prentice Hall, 1989.

[17] M. A. L. Thathachar, and Sastry P.S., "Varieties of Learning Automata: An Overview," IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, Vol. 32, No. 6, 2002, pp.711-722.

[18] M. R. Meybodi, and S.Lakshmivarahan, "Optimality of a Generalized Class of Learning Algorithm", Information Science, Vol. 28, 1982, pp. 1-20.