Islamic Azad University of Qazvin

ISSN 2345-6582

Journal of Computer & Robotics

Qazvin Islamic Azad University
Vol.5, No. 1, Spring 2014

Directional Stroke Width Transform to Separate Text and Graphics in City Maps 1
A.Ghafari-Beranghar, E.Kabir, K.Kangarloo

A W-band Simultaneously Matched Power and Noise Low Noise Amplifier Using CMOS 0.13μm 9
M.Mohammad-Taheri

Self-cleaning Measurement of Nano-Sized Photoactive TiO2 15
M. Farahmandjou

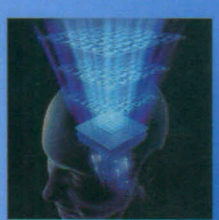
Fast Cellular Automata Implementation on Graphic Processor Unit (GPU) for Salt and Pepper Noise Removal 21
A.Jalalian, B.karasfi, K.Samsudin, M.Iqbal Saripan, S.Mashohor

Advertising keyword suggestion using relevance-based language models from Wikipedia rich articles 29
A.H. Jadidinejad, F.Mahmoudi

MMDT: Multi-Objective Memetic Rule Learning from Decision Tree 37
B. Shaabani, H.Sajedi

PSO-Based Path Planning Algorithm for Humanoid Robots Considering Safety..... 47
R. Shakiba, M. Ersali-Salehi

Improving Agent Performance for Multi-Resource Negotiation Using Learning Automata and Case-Based Reasoning 55
M.Haghighatjoo, B. Masoumi, M. R. Meybodi



Improving Agent Performance for Multi-Resource Negotiation Using Learning Automata and Case-Based Reasoning

Monireh Haghighatjoo [a*], Behrooz Masoumi [a], Mohamad Reza Meybodi [b]

[a] Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Department of Computer Engineering and Information Technology, Amirkabir University, Tehran, Iran

Received 4 September 2012; accepted 8 July 2013

Abstract

In electronic commerce markets, agents often should acquire multiple resources to fulfil 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. In recent years, negotiation has been employed to allocate resources in multi-agent systems. Yet, in most of the conventional methods, negotiation is done without considering past experiments. In this paper, in order to use experiments of agents, a hybrid method is used which employed case-based reasoning and learning automata in negotiation. In the proposed method, the buyer agent would determine its seller and its offered price based on the passed experiments and then an offer would be made. Afterwards, the seller would choose one of the allowed actions using learning automata. Results of the experiments indicated that the proposed algorithm has caused an improvement in some performance measures such as success rate.

Keywords:Multi agent system,Resource allocation, Negotiation,Learning automata,Case -based reasoning

1. Introduction

Today, use of artificial intelligence rather than traditional approaches has improved performance of systems in E-commerce. In electronic 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]. Additionally, the issue of learning in negotiations has become a hot spot within the related studies as a mean for achieving a well-coordinated behaviour. 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. Leaning 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 butalso, 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[2].

In negotiations, agents employ a variety of strategies (negotiation strategy is act of guiding for decision making about different actions in a certain round) to allocate resources; but, in most of them, past experience does not play a role in resolving new problems[3]. For example Peyman Faratin et al. [4] 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. [5] expanded the service-oriented negotiation model by using genetic algorithm and presented the relative success of different tactics against different opponents in different environments. Nyugen et al. [6] 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 later expanded in [7] so that the ability of decision making for commitment/decommitment of

* Corresponding author. Email: monireh.haghighatjoo@gmail.com

the deals was integrated with the one in the previous model. Because, the buyer agents might need to make deals with another agents, so a buyer could reach good deals in an efficient manner. In 2010, Bo An et al. [8] proposed a time-dependent strategy called HBA and compared it with the papers in [7]; this comparison showed increased rate of some efficiency criteria such as success and expected profit.

Considering the type of strategy that was selected, this article aims to evaluate performance of employing case-based reasoning and learning automata as a strategy for agents' decision making in multi-agent systems.

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

2. Multi Resource Negotiation Problem

Resource allocation in multi-agent systems is the process of distributing multiple resources among several agents that can affect distributions. The question is: "How resources are distributed?"[9].To answer this question, negotiation can be mentioned as a method of coordination between competitive agents for the issue of resource allocation. During negotiation, a group of agents will make a mutual decision or reach an acceptable agreement on a specific issue. System of negotiation is illustrated by multiple N = (Ag, S, P) where Ag is a finite set of negotiator agents, S is strategy of negotiation and P is negotiation protocol[10].

The framework of this article was inspired by a time-dependent negotiation model as described in Bo An et al.'s work[8], where the issue of automated negotiation for resource allocation between providers (seller) and consumers (buyers) 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 article includes the 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.

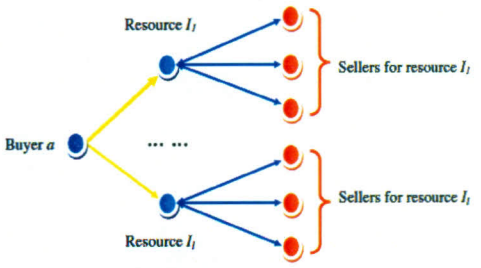


Fig. 1.Buyer a's multi-resource negotiation problem [8].

Figure 1 shows *Buyer a's* multi-resource negotiation problem. Where,I = $\{I_1, I_2,..., I_L\}$ is a set of resources needed by *Buyer a* to accomplish its task and τ is *Buyer a's* negotiation deadline. In this negotiation model, 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 Eq. (1):

$$\varphi^t_{a\to s} = I\,Pj + (RP^t_j - I\,Pj)\delta^t_j \quad (1)$$

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 [4], which are the time-dependent function (linear, conceder, conservative) and the behavior-dependent function (e.g., 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 and making a tentative agreement by paying the penalty. Otherwise, the tentative agreement will be finalized and both parties need to fulfill all their final agreements[8]. (see more details in [8]).

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):

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

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].

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

3. Learning in Multi Agent System

3.1. 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.

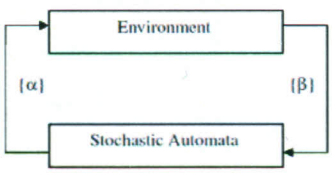


Fig. 2. Relationship between learning automata and environment[11].

The relationship between learning automata and the environment is shown in Figure 2. 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[11]. 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,β_1=1 and β_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[12].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 action , *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:

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

Unfavorable responses from the environment:

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

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}[13].

3.2. Case-Based Reasoning

Case-based reasoning (CBR) 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[14].

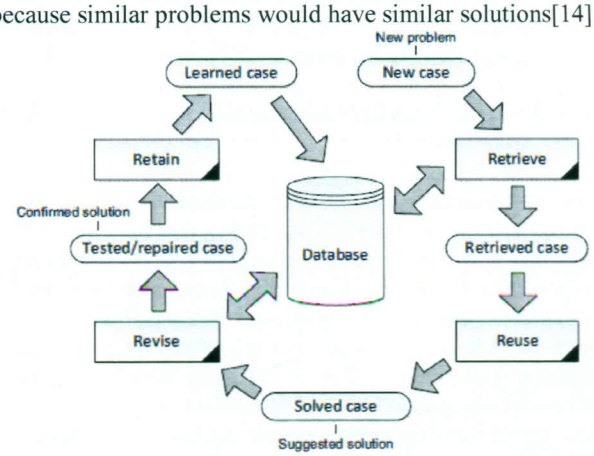


Fig. 3. CBR cycle[15].

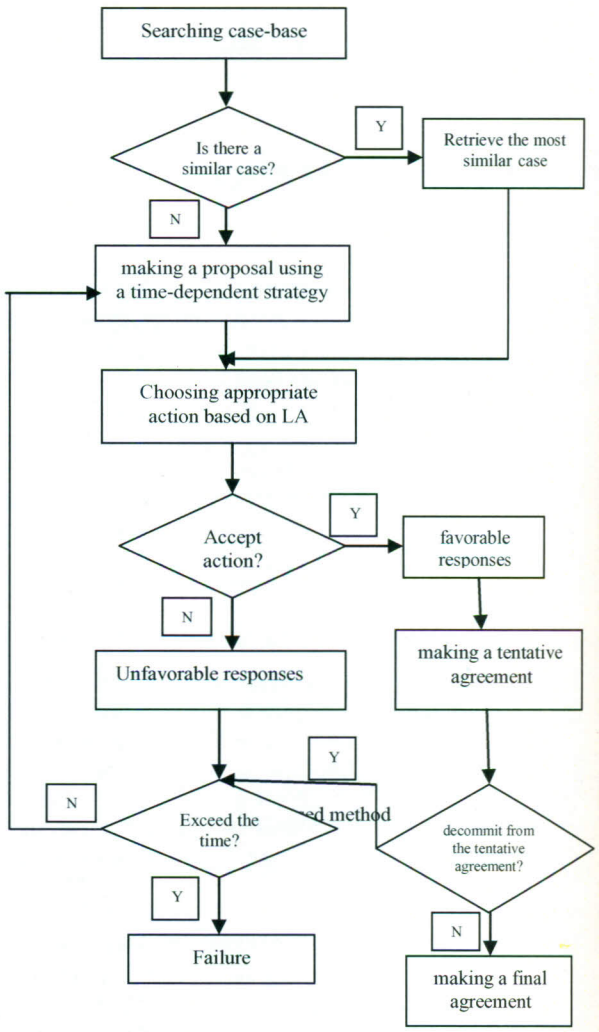
The CBR cycle is illustrated in Fig. 3:

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[1 5].

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 forsolving 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[16].

4. Proposed method

In this section, a method is proposed for solving resource allocation problem using case-based reasoning and learning automata based on negotiation called IHBA. 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} as well as negotiation protocol determine a set of allowable actions for each agent set including accepting the proposal, rejecting the proposal and making a counter- proposal.
In the proposed algorithm, it was assumed that there were two types of case-bases: 1- case-base of the proposals that were 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 examine 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 time-dependent strategy (Equation 1) to make a proposal. Then, the Seller agent must choose one of the allowed actions. In this case, the agent's strategy for negotiation would be CBR algorithm and 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 desirable; if the offer is rejected and no agreement is reached, the output will be known as undesirable. Afterwards, the agent updates probability of action selection based on learning automata algorithm until an agreement is reached among the agents or negotiation of the agents is finished. The automata that has been employed in this article used reward and penalty functions of (4) and (5).



5. Experiment result

Various parameters were considered for evaluation such as performance measures that included[8]:

- Success rate :

$$R_{SUC} = N_{success} / N_{total} \quad (6)$$

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

- Message per resource

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

Where N_{total} is total number of runs, IS_i is number of resources in the i-th run and M^j_i is number of messages for resource j in the i-th 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

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

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

To evaluate such measures a number of experiments was carried, in which agents were subjected to different market types, densities and deadlines.

Table 1
Experiment setting

Market type	Favorable	Balanced	Unfavorable
No. of agents	6-35	36-65	66-95
Deadline	Short	Moderate	Long

According to Table 1, 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 lifespan of an agent in e-market (i.e. its deadline) 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). In the present experiments, HBA and IHBA had the same conditions (for instance, the number of resources they needed to obtain), except that they used different negotiation strategies. Moreover, in IHBA, reward and penalty parameters for learning automata were considered 0.004 and 0.0001, respectively.

5.1. Observation 1

Table 2 shows results of the experiments for 1000 runs. As noted in the table, IHBA has a higher rate of success than HBA. Success rate increases by5.5% and the number of messages sent/received by the buyer for each resource is declined by 6.3%.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 2
Experimental results for 10^3 runs

Strategy	R_{suc}	M_{aver}	Uexp
HBA	0.54	83	0.206
IHBA	0.57	80	0.36

5.2. Observation 2

Fig.5 shows that, the more the number of resources required for fulfil 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 IHBA which use past experience and learning have higher success rate than HBA and, in longer deadlines, such a success rate for learning-based IHBA has more increase than HBA.

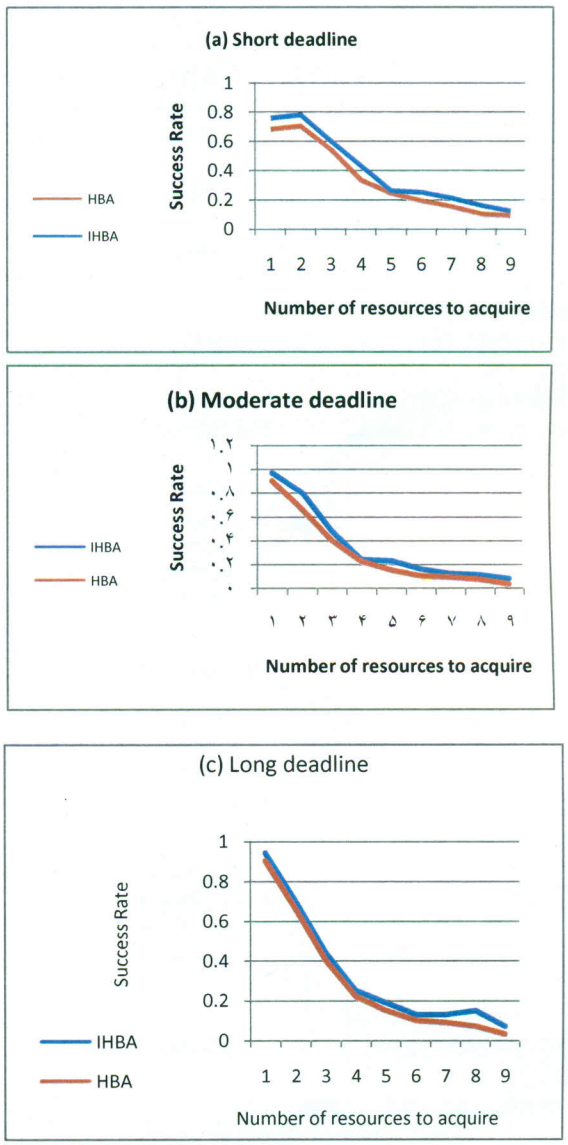


Fig .5. The number of resources and success rate (a: short b:moderate c:long)

As demonstrated in Fig. 6: (1) success rate in IHBA are always higher than of the HBA. In the case of shorter deadline, the possibility of reaching an agreement and thus success rate of the IHBA which used previous experiences increased compared with HBA agent. Furthermore, by increasing the deadline, success rate of IHBA had much higher increase than HBA, because when the deadline is longer, buyers not only use past experiences and learning, but also have sufficient deadline for reaching an agreement.

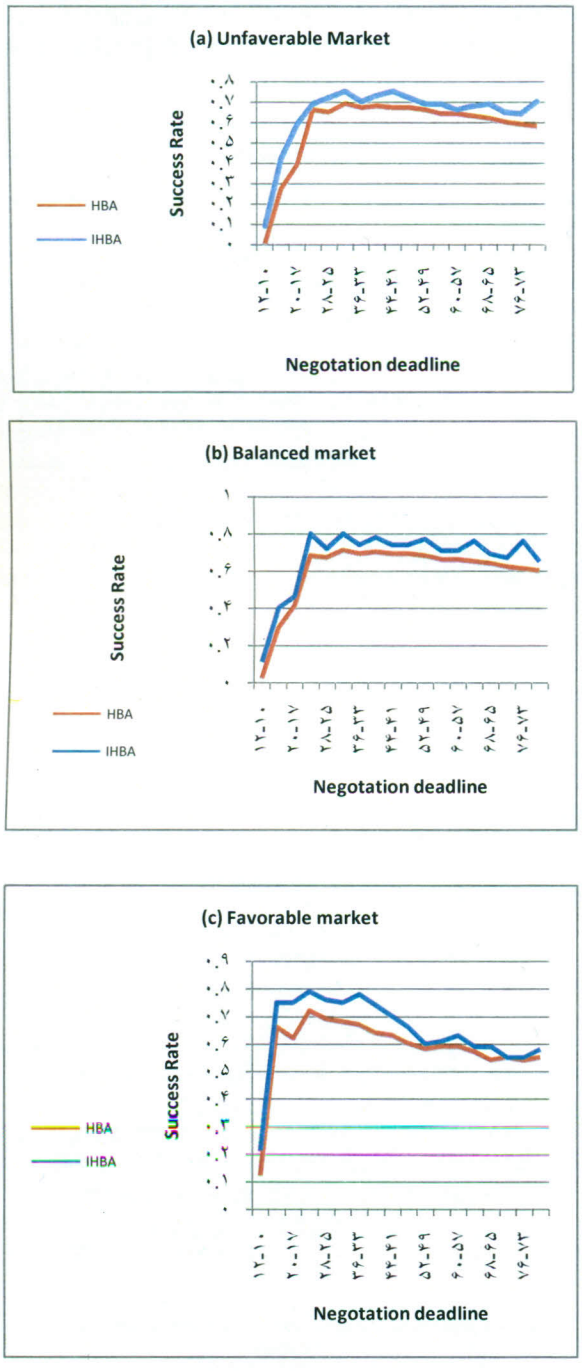


Fig. 6. The deadline and success rate (a:unfavorable b:balanced c:favorable)

In Fig. 7, in two strategies, the more the number of 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.

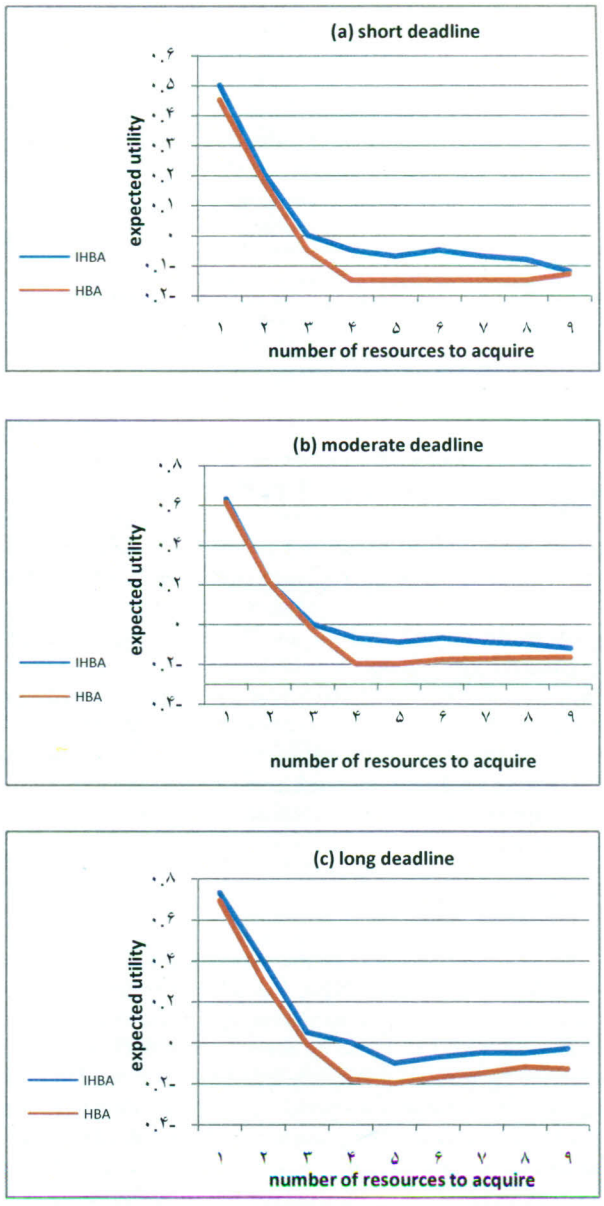


Fig . 7. The number of resources and expected utility (a:short; b:moderate; c:long)

In Fig. 8, results of the experiments show that: (1) Results of the negotiation improved for two kinds of buyers by increasing deadline; (2) Considering equal deadline, IHBA agent gain a higher expected utility than HBA agents, especially when the market is unfavorable and the buyer

agent has sufficient time for reaching an agreement. Also, in shorter deadlines, since IHBA used previous experiences, therefore, they obtained higher expected utility than HBA, because in short deadlines, they could make more agreements, which led to obtaining higher expected utility. Furthermore, this advantage is decreased when the market is favorable.

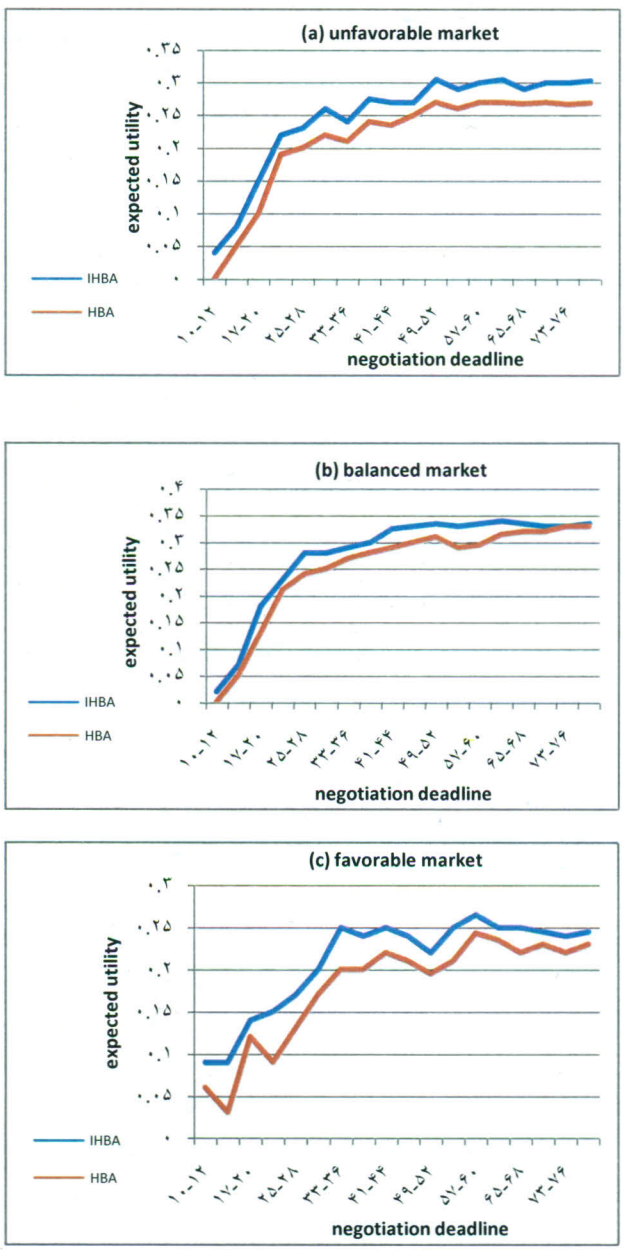


Fig. 8. The deadline expected utility (a:unfavorable b:balanced c:favorable)

6. Conclusion

In negotiation, for making use of agents' experiments, a modelbased on case-based reasoning and learning automata has been proposed. In this model, a learning automata has been placed per agent; these learning automatas lead the agents for opting an appropriate act to reach an acceptable agreement. Besides, it improves performance measures such as success rate, and the number of messages that have been sent/received by each buyer in the negotiation. Moreover, in future, the learning automata can be used for selecting appropriate tactics in order to present a recommendation and recommendation response.

References

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

[2] 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.

[3] Bo An, Nicola Gatti, Victor Lesser, "Bilateral bargaining with one–sided uncertain reserve prices", Journal of Autonomous Agents and Multi-Agent Systems, Vol.26, 2012, pp. 420-455.

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

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

[6] Nguyen, T. D., 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.

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

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

[9] Y. Chevaleyre, et al, "Issues in Multi agent Resource Allocation", Informatica, vol. 30, 2006, pp. 3–31.

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

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

[12] M. A. L. Thathachar, 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.

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

[14] Guorui Jiang, 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.

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

[16] S. Ghanbari, M.R.Meybodi, 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.