Smart Computing Review

A Smart and Automated Negotiation System based on Linear Programming in a Multilateral Environment

Sheetal Vij[1], Madhur Patrikar[1], Debajyoti Mukhopadhyay[1]*, and Avinash Agrawal[2]

[1]Maharashtra Institute of Technology, Pune 411038, India / {sheetal.sh, mapatrikar, debajyoti.mukhopadhyay}@gmail.com

[2]RCOEM, Nagpur, India / agrawalaj@rknec.edu

*Corresponding Author: Debajyoti Mukhopadhyay

Received October 9, 2015; Revised November 11, 2015; Accepted December 17, 2015; Published December 31, 2015

Abstract: Automated negotiation is a branch of e-commerce. Successful management of purchase and sale activities depends not only on selecting the right products or services but also on choosing the best method for buying and selling them. We propose a linear programming and pattern matching–based multilateral automated negotiation system. We compared some methods of automated negotiation. A multilateral negotiation system gives better results for participants than bilateral automated negotiation. The technique of pattern matching–based automated negotiation gives fast results and reduces the overhead from calculations. Automated negotiations are not an easy task without the interactions of human beings. Automated negotiation can be done in bilateral or multilateral environments. In this paper, we consider many-to-many negotiations where agents concurrently negotiate with many other agents. In this paper, we also propose a matrix method to manage the concurrent negotiations conducted by multiple buyer agents with multiple seller agents.

Keywords: utility theory; multilateral negotiation; linear programming–based negotiation; pattern matching–based negotiation

Introduction

Nowadays, the view of organizations towards negotiation has changed. E-commerce has changed the nature of business interactions, whether it is B2B or B2C trading or online shopping. E-procurement is called supplier exchange. It is used in the sale of supplies and services via the Internet. It enhances business relationships and leads to cost savings. Software agents assigned the job of negotiation should be able to simulate human behavior, be able to learn from past experience in negotiation and improve their respective negotiation strategies.

DOI: 10.6029/smartcr.2015.06.004

In automated multi-attribute negotiation, time-based, resource-based and behavior-based strategies are used. A heuristic method can be used to learn private information about agents. In the process of negotiation, additional information about a user or an offer can be exchanged by using an argumentation-based negotiation method [12]. Attributes of negotiation can be dependent or independent. For example, suppose a buyer agent and a seller agent are willing to negotiate the purchase of a product with several attributes; in a situation where attributes are dependent on the value of another attribute, if the value of one attribute increases for the buyer or seller, then the other attribute's value is automatically decreased.

The number of issues, the dependency between issues, the representation of utility, negotiation protocols, the form of negotiation and time constraints, are the main factors of negotiation. Negotiation takes place when two or more people, with differing views, come together to attempt to reach agreement on an issue. It is persuasive communication, or bargaining. Attributes of negotiation can be dependent or independent. Negotiation is conducted in various ways, such as bidding, bargaining or by using auctions. In bilateral negotiation, two agents are involved in negotiation over a single issue or multiple issues. When more than two agents are involved with multiple issues, this complex phenomenon is called multilateral automated negotiation. Multilateral automated negotiation is possible in one-to-many, many-to-one or many-to-many formats.

The organization of this paper is as follows: i) Introduction to automated negotiation, ii) Related work on multilateral automated negotiation and its methods, iii) Linear programming–based multilateral automated negotiation, iv) Proposed architecture of multilateral negotiation and the matrix method, and v) the results of negotiation: a comparison of methods for multilateral automated negotiation with respect to time needed and efficiency. A summary of the forms of automated negotiation is in Table1.

Table 1. Forms of negotiation

Sr. No.	Bidding	Auctions	Bargaining
1.	Buyer starts	Seller starts	Buyer starts
2.	Bilateral/Multilateral	Multilateral	Bilateral/Multilateral
3.	Simplest negotiation	Medium negotiation	Complex negotiation

In bilateral automated negotiation, maximum utility for a single agent can become minimum utility for the opponent agent; therefore, the chance of agreement is low. This problem is avoided with multilateral automated negotiation. A major challenge in negotiation using the bilateral protocol is that the agents hide their preferences. So each agent does not know which preferences the opponent will prefer.

Related Work

There are many methods for multilateral automated negotiation. The commonly used methods are utility theory, heuristic-based algorithms, argumentation-based negotiation, fuzzy-based negotiation and game theory. Wang and Shen discussed how a mixed integer programming algorithm can provide optimal solutions [7], but scalability and computation time would increase when the number of agents increases. Lijuan et al. were inspired by a bio-inspired algorithm based on genetic theory. Lijuan et al. proposed ant-inspired negotiation. Autonomy, scalability and adaptability are features of the ant-inspired–based automated negotiation algorithm. They need to improve technical strategies to improve offers.

Mukhopadhyay et al. [28] and Vij et al. [27] have given comparative studies in bilateral negotiations. For Buttner, automated negotiation is classified mainly as structure, theoretic foundation and restriction, as shown in Figure 1 [21]. We are going to focus on the protocol for the structure. The protocols can be classified into bilateral, one-sided and double-sided protocols. One-sided and double-sided negotiations are also called multilateral automated negotiation. The Kasbah model was built on only a single issue, such as price. The system was built on multiple issues with two agents, but it has not provided good results for buyers and sellers.

Klaus et al. [19] gave an overview of game theory–based negotiation, multi-attribute utility theory–based negotiation and auction-based negotiation. As per their paper, there is a scoring function problem and a user-dependent problem in many-to-many multilateral negotiation. For a linear scoring function, an optimal solution can be found, but for a non-linear scoring function, mathematical analysis is very difficult. How to construct the negotiation strategy is not clear in this paper. As per this paper, multilateral negotiation using game theory is very difficult to use. Utility theory can give better results than game theory.

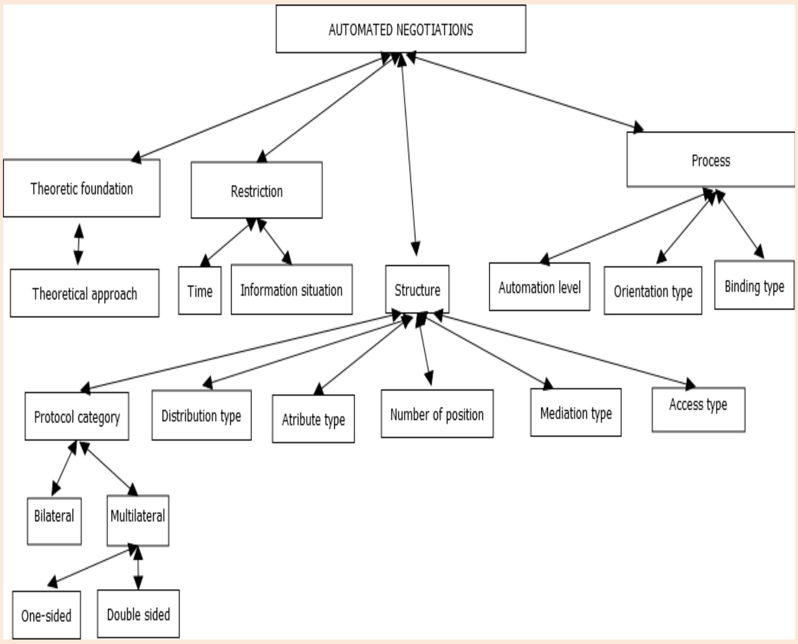

Figure 1. Main criteria for classifying automated negotiation

Park and Yang [23] proposed a negotiation agent system based on incremental learning in order to increase the efficiency of bilateral negotiations and to improve applicability towards multilateral negotiations. For the system, they also introduced a framework for multilateral negotiations in an e-marketplace in which the components can dynamically join and disjoin. They proposed an automated negotiation system that can efficiently carry out multilateral negotiations with multiple attributes in pervasive computing environments. The effects of learning ability are investigated, with a focus on reciprocity of participants and on the execution time of negotiation. The issues in relation to the improvement in incremental learning and the development of delicate protocols for agent interoperability are not included in this system. They also developed a linear programming–based automated negotiation system. They used the concept of a mediator agent, and two bilateral automated negotiation schemes based on linear programming. The experimental results show that the proposed system produces higher joint profits and is faster at reaching agreements, on average, under the condition of agreement for reciprocity than a negotiation system based on a trade-off mechanism [22].

A multiple issue negotiation model with distributed problem solving was presented by Faratin et al. [14]. In this, they developed a fully autonomous agent that coordinates both agents' interactions and also handles an individual agent.

Monotonic Concession Protocol for Multilateral Negotiation was described by Endriss [24]. It is a deadlock-free protocol which restricts the utility function. It is not applicable for all cases of negotiation. When the participant does not share his preference in the negotiation, the agent needs to analyze the behavior of the opponent. Performance of the negotiation can be measured in two ways: using an agent's performance as a benchmark for the model's quality, and directly evaluating its accuracy by using similarity measures.

As per Raz (2012), there is an almost linear correspondence between accuracy and performance of the system. This study measured accuracy of a system over timing, but did not consider a system based on resource dependence. Dong proposed a multi-attribute negotiation model based on internal factor argumentation; the system can achieve a Pareto efficiency solution, promotes cooperation between agents, and then reaches a win-win result. In a multilateral multiple issue negotiation protocol, a multiple agent system (MAS) is used for decision making [15].

Considering these papers, we can say multilateral automated negotiation gives better results to buyers and sellers. A negotiation protocol is a general rule that can be used by anybody in the negotiations. The protocol determines the flow of messages between the negotiating parties. Request-based negotiation protocols and sequential bilateral negotiation protocols are used for bilateral negotiation. Automated mediation, baseline mediation, multiple bilateral, feedback-based mediation and contract net protocol are used for multilateral negotiation. A win-win strategy gives better outcomes to buyer and seller. Intelligent techniques such as neural networks, genetic programming, fuzzy logic theory and Bayesian theory are used to learn the opponent's behavior, for decision-making and for generating offers. Fuzzy systems, multithreading, game theory, genetic algorithms and linear programming are some of the methods that can be used for multilateral automated negotiations [8].

One of the most important criteria is the protocol category. It describes the number of negotiating partners. There, bilateral and multilateral are separated. One-sided and double-sided are two types of multilateral automated negotiation.

There are two environment forms of automated negotiations.

1. Bilateral Automated Negotiation: Bilateral negotiations are done between only two parties. Bilateral negotiations can be done using single issues or multiple issues. Two agents negotiate on behalf of their respective owners. It is a simple environment form.

2. Multilateral Automated Negotiation: In multilateral automated negotiation, there are more than two agents involved with multiple issues. One-to-many, many-to-one and many-to-many are three forms of multilateral negotiation. In one-to-many negotiation, a buyer is represented by a combination of one coordinating agent and multiple sub-buyer agents, and each supplier is represented by a seller agent.

They may use one of the coordination strategies: the desperate strategy or the patient strategy. In the desperate strategy, agents want to complete the negotiation process as early as possible. The negotiation process is terminated as soon as any one of the sub-buyer agents is successful in the negotiation. In case several proposals are found at the same time, the proposal with the highest utility gain is accepted. This is called one-side multilateral negotiation. It is the same as one-to-many or many-to-one negotiation, but the difference is that it has two coordinating agents on both sides. This is also called double-sided multilateral negotiation. Multilateral negotiations are differentiated from bilateral negotiations because of their wider size, greater complexity and greater heterogeneity. Multilateral negotiation often presents very different dynamics of development from bilateral negotiations, according to three dimensions: i) wider size, ii) greater complexity, and iii) stronger diversity. Multilateral negotiations are constituted by several parties, and among them, heterogeneous ones, too, each with their own configuration of interests and issues [22].

It is clear that a change in the number of parties and issues, especially in the case of an increase in abundance, leads to complications in the negotiation process.

■ Utility Theory

There are many methods that take preferences from participants. Preferences can be taken in several forms: conditional, unconditional, qualitative and/or quantitative. Utility theories, fuzzy methods and the cp-net with heuristics method, are all methods where preferences are taken. We used utility theory to assign a weight to preferences. Weight/priority should lie between 0 and 1. A weight that is greater than 1 is rounded down to 1. Weight less than 0 is rounded to 0. Each object has a number of different properties. Each property has sub-properties. Consider the example of a laptop, as shown in Figure 2.

Table 2. Detail working of matrix

Seller1*Buyer1	**Seller2*Buyer1**	**Seller3*Buyer1**	**.....**	**SellerN*Buyer1**
Seller1*Buyer2	Seller2*Buyer2	Seller3*Buyer2	.....	SellerN*Buyer2
Seller1*Buyer3	Seller2*Buyer3	Seller3*Buyer3	.....	SellerN*Buyer3
Seller1*Buyer N	Seller2*Buyer N	Seller3*Buyer N	.....	Seller N*Buyer N

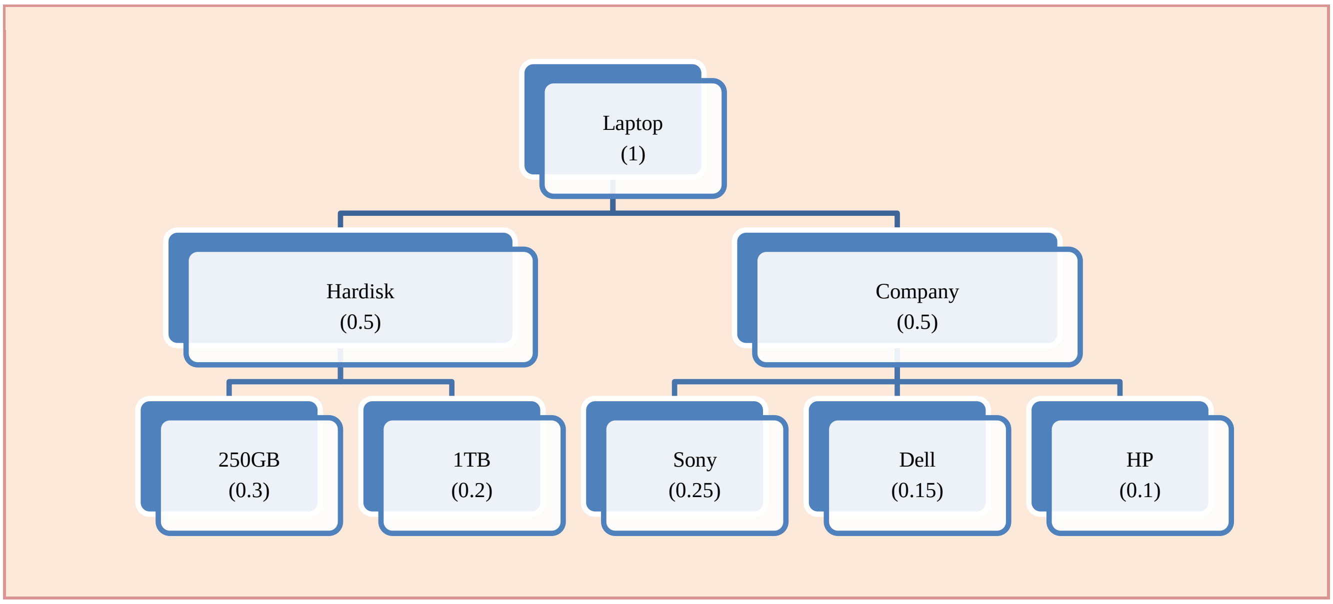

Figure 2. How to assign weight to preferences

The laptop is the object, which has a different number of properties. Here, we consider a laptop to have two properties: hardisk and company. Again, hardisk and company each have properties, called sub-properties of the laptop. Utility is assigned in a form such that the sum of the weight of all properties should be 1.

Linear programming based multilateral automated negotiation

According to Park and Yang, a linear programming system produces higher joint profits in negotiation and is faster at reaching agreement, on average, under the condition of agreement of reciprocity rather than the trade-off mechanism–based negotiation system [22]. The seller agent and buyer agent are connected to each other through the proxy of a mediator. The mediator agent evaluates the profits of all negotiation participants and sorts them in decreasing order of joint profit. Thereafter, the final couples are determined from the sorted list. The time complexity determines the final couples, where the maximum profit criterion is O (N log N), where N is the number of negotiation partners, since sorting takes longer than any other operation. So, we propose a pattern-matching method to reduce time complexity. The value of the utility function can be regarded as the profit of either buyer or seller. The utility function Profits (x_i) of a participant can be expressed as follows:

$$\text{Profit}(x_i) = \sum_{i=1}^{n} w_i.\text{E}(x_i),\ \sum_{i=1}^{n} w_i=1,$$

where n is the number of attributes, x_i is a variable representing the offer value of the i^{th} attribute, w_i is the weight of the i^{th} attribute, and finally, the evaluation function $\text{E}(x_i)$ of the i^{th} attribute is expressed in terms of the request values (request_value_i) and the allowable values (allowable_value_i) as follows:

$$\text{E}(x_i) = \frac{xi - allowabl_{value}i}{request\_value\ i - allowable_{value}i}$$

where the allowable value means the maximum value that the participant concedes for negotiation, and the request value means the maximum value the participant wants in negotiation. If $x_i = \text{request\_value}_i$, the degree of satisfaction of the i^{th} attribute is assumed to be the highest, and $\text{E}(x_i)$ representing the degree of satisfaction of the i^{th} attribute becomes 1. On the other hand, if $x_i = \text{allowable\_value}_i$, the degree of satisfaction of the i^{th} attribute is the lowest, and $\text{E}(x_i)$ is set to 0. Therefore, when x_i is within the range between request_value_i and allowable_value_i, $\text{E}(x_i)$ ranges between 0 and 1. If x_i is outside the range between request_value_i and allowable_value_i, $E(x_i)$ is set to either 0 or 1, depending on the value of $E(x_i)$. That is, if $E(x_i) < 0$, it is set to 0, and if $E(x_i) > 1$, it is set to 1.

Proposed architecture

A client/server architecture is used to design a negotiation system on multiple nodes. Organizations often seek opportunities to maintain service and quality competition to sustain market position, with the help of technology, where the client/server model makes an effective impact. Deployment of client/server computing in an organization will positively increase productivity through the use of cost-effective user interfaces, enhanced data storage, vast connectivity and reliable application services.

- Improved Data Sharing

Data retained by usual business processes and manipulated on a server are available for designated users (clients) over an authorized access.

- Security

Servers have better control access and resources to ensure that only authorized clients can access or manipulate data, and server updates are administered effectively. All databases are stored on the server side. Participant and Admin are two types of user who can use the system. The administrator has the responsibility to start the negotiation process, as shown in Figure 2. The requirements of buyers and sellers are passed to the product and preferences module. Buyers and sellers have their own preferences, as per their priority of items; the weight will be assigned to each item. Each buyer and seller selects agents manually. The multiple agent module is used to select the agent and finalize pairs between buyers and sellers for negotiation. An agent performs negotiation on behalf of a buyer or seller. There should be one agent for each participant, and this is a constraint in our model. Details of the model are described in the procedure section. The system can support both bilateral and multilateral negotiation. Here, we used three different methods to generate offers, and to finalize the pairs between participants. Linear programming with a matrix, linear programming with sorting, and pattern-matching methods are used to generate counter-offers. We compared the efficiency and the time of each method. The flow activities of negotiation methods are shown in Figure 3 [26].

Decision function

Decision making is a process that moves an individual or a group towards a common goal. A decision function is used to take the decision about the negotiation process. After checking the constraints of the system, the decision function decides whether an offer will be accepted or rejected. The decision function saves each offer in the database, which is generated by the offer generator. The automated negotiation processes between two agents, or more than two agents, are bilateral interactions that consist of a succession of offers or counter-offers. Let a1, a2, a3, a4….. be the agents. Each agent has to define a range of values of objects. Let $x_o \in [MIN^a_o, MAX^a_o]$, where objects are negotiated by participants. Each agent has a utility function: $U^a(x_o)$. To negotiate on the multiple objects (say o), a normalized weight represents the relative importance of each object. We are using the variable x to denote the offer that agent a has received from agent b at time t. Round (R) is the number of attempts, which is directly proportional to time.

The decision function has three main cases. 1) Reject: $t > t_{max}$, ~ (min ∩ max) , ~ ($Object_a \cap Object_b$), where, $Object_a$ is the object of the buyer, and $Object_b$ is the object of the seller; 2) Accept: $U_a(x^t_{\{a-b\}}) <= U^a(x^{\{t\}}_{\{b-a\}})$; and 3) Otherwise: offers are generated.

- Negotiation determines the value of x such that

$$(x \in [min^a, max^a] \cap [min^b, max^b]).$$

where x is the offer, min^a is the minimum value of the buyer, max^a is maximum value of the buyer, min^b is the minimum value of the seller, and max^b is the maximum value of the seller.

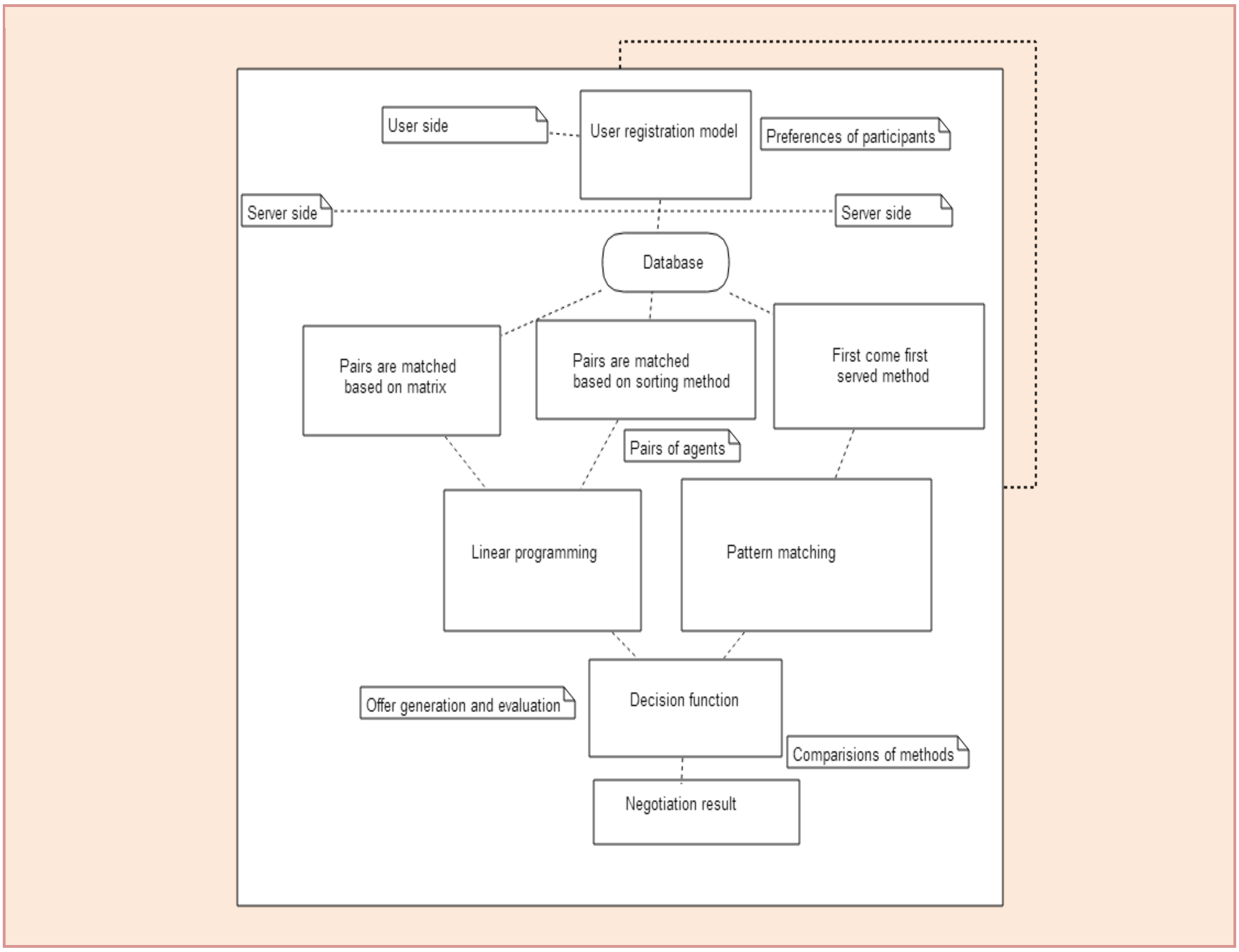

Figure 3. Architecture of multilateral automated negotiation

- Proposals and counterproposals are evaluated using a scoring function:

$$response^a\left(t^n, x_{b\to a}^{t_{n-1}}\right) = \begin{cases} withdraw(a,b)\ if \quad t^n > t_{max}^a \\ accept(a,b,x_{b\to a}^{t_{n-1}}\ \ if\ \ V^a\begin{pmatrix} t_{n-1} \\ x_{b\to a}\end{pmatrix} > V^a(x_{a\to b}^{t_n}) \\ offer(a,b,x_{a\to b}^{t_n}\ \ otherwise \end{cases}$$

■ Offer generator

An offer generator is used to generate offers for the buyer and seller. For offer generation, we use a linear programming method.

Offer = (min_utility+0.1) U_B+ (Max_Utility – S $_{Discount;)}$ $U_{S.}$

■ Pattern match-based negotiation

This is used by the decision function in case of emergency (e.g., not enough time). The case for emergency depends on the time factor. If the required time of the buyer is decreased, then the decision function sends a direct available offer to the participants. Among these offers, the buyer's agent selects the one that is the most suitable for the buyer's preferences. The offer will be available on the basis of market basket analysis. Market basket analysis is the discovery of relations or correlations among a set of items. The decision function sends the final output of the negotiation process to the negotiation result module. The data set is a large collection of files. No structure for datasets of automated negotiation is available on which we can apply the negotiation methods and analyze different test cases.

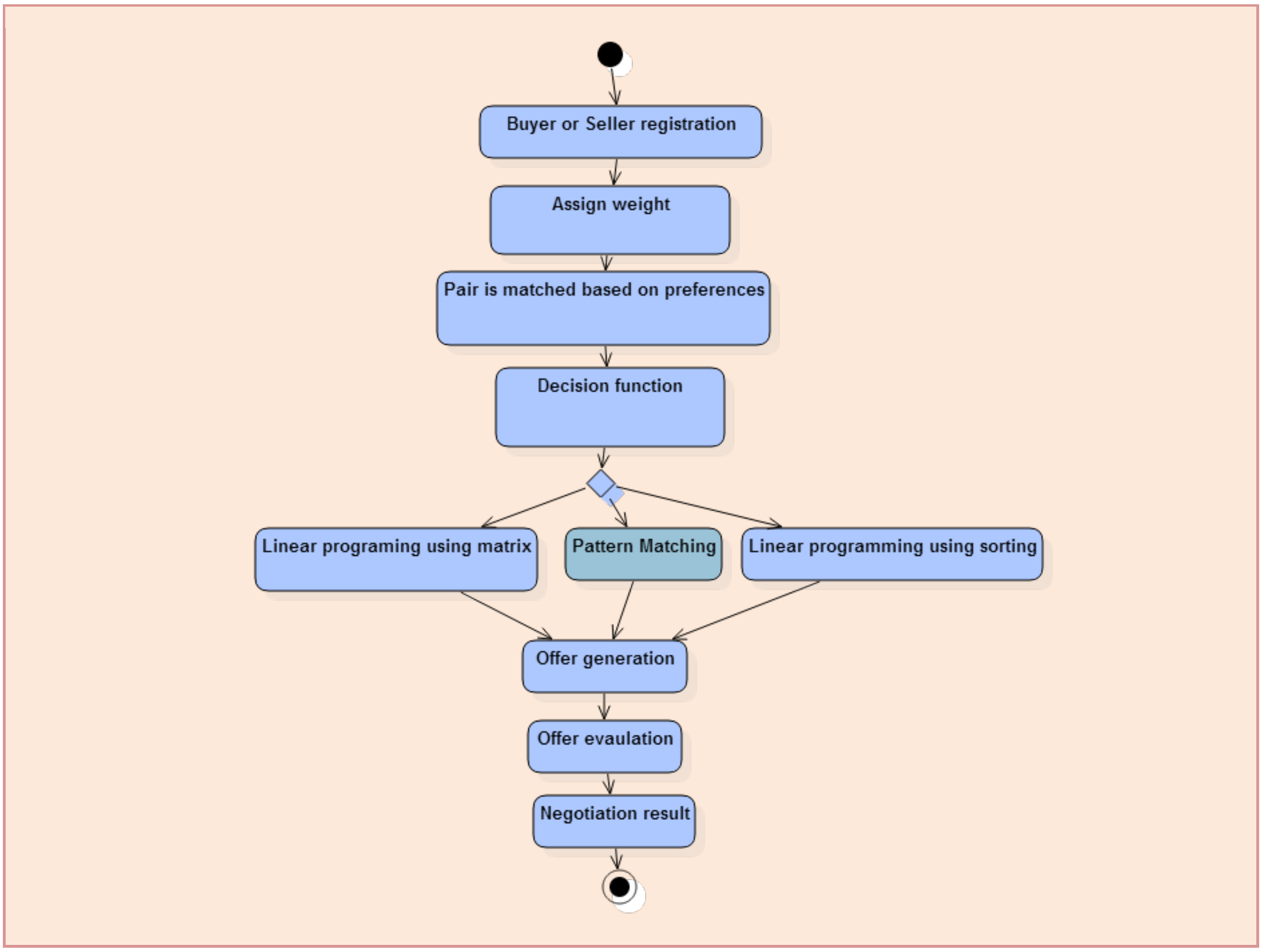

Figure 4. Activity flow diagram of multilateral automated negotiation

Negotiation Result

To compare and analyze the effectiveness of our proposed system, we conducted some experiments. Graph 1 shows that participants who have fewer rounds than their opponents get maximum profit, and vice versa, for case III (see Graph 3).

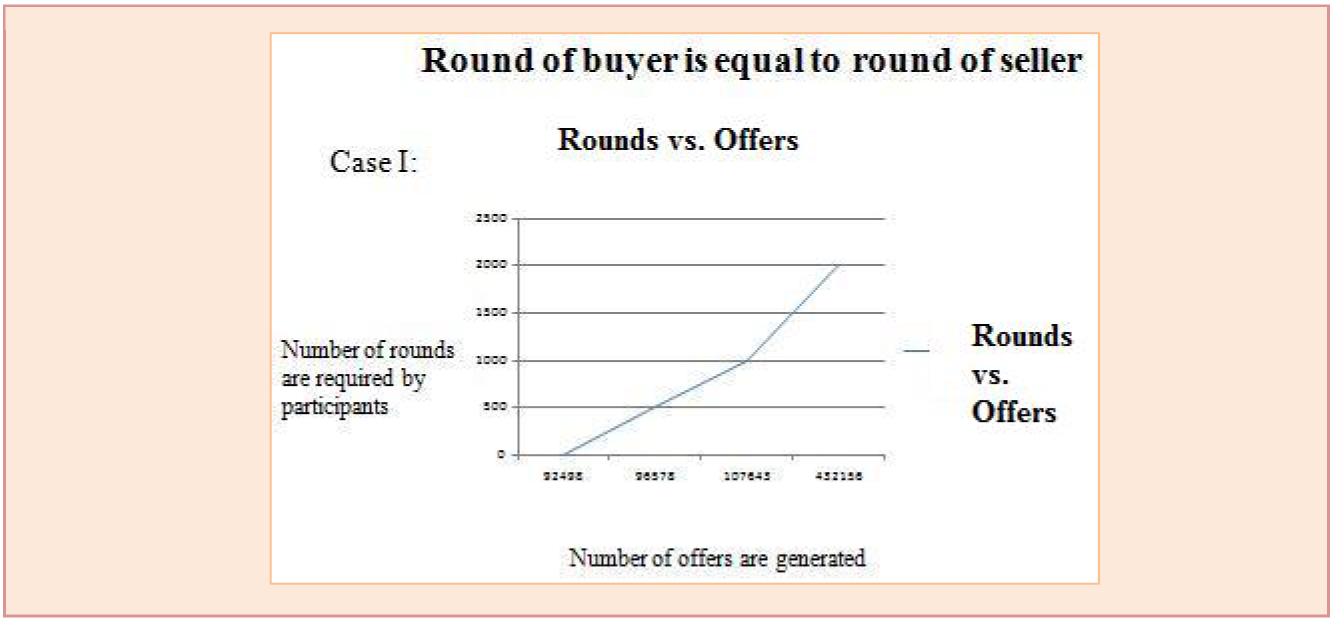

Figure 5. Round vs. Offer (buyer and seller have different values for the object)

If buyer and seller both have the same marginal value, then all participants get the same offer values (see Graph 2).

$$\text{Efficiency}=\frac{\text{Total number of participants}-\text{Number of participants are rejected}}{\text{Total number of participants}}$$

We compared our three proposed methods (after registering more than 500 participants, an appropriate result was obtained). Pattern matching gives fast results in less time, but a linear programming method gives greater efficiency than the pattern matching method, as shown in Graph 6. Linear programming using a matrix, and linear programming with a sorting method required approximately the same time to generate the offers.

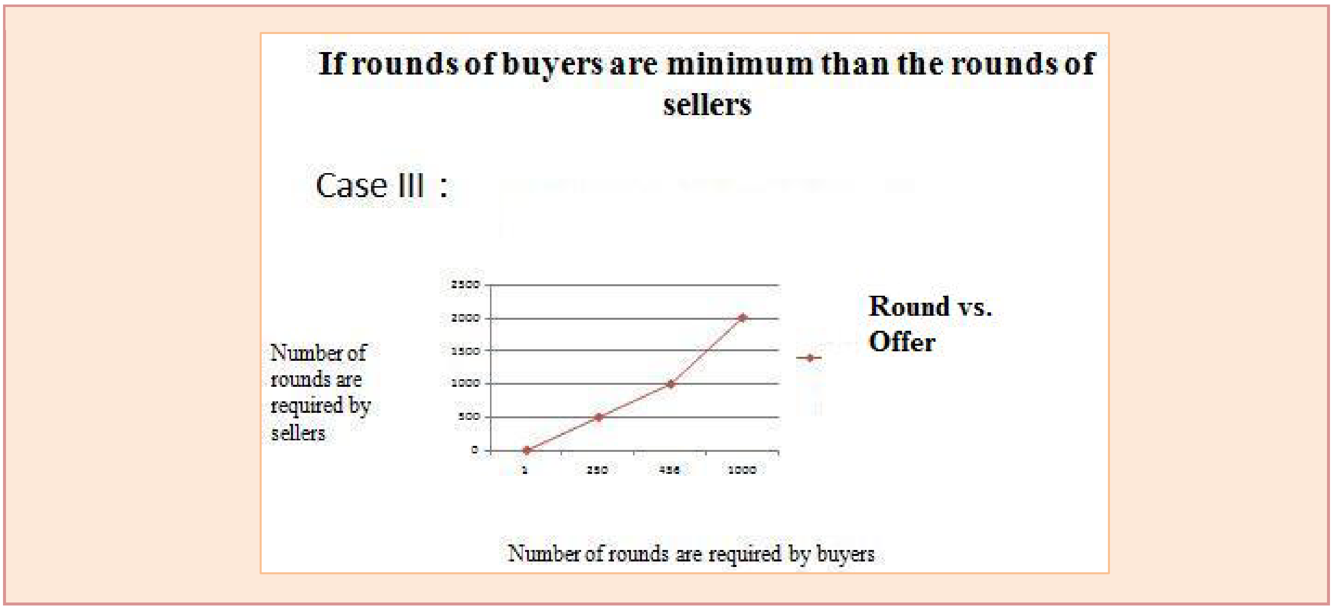

Figure 6. If the buyer's rounds are fewer than the seller's rounds

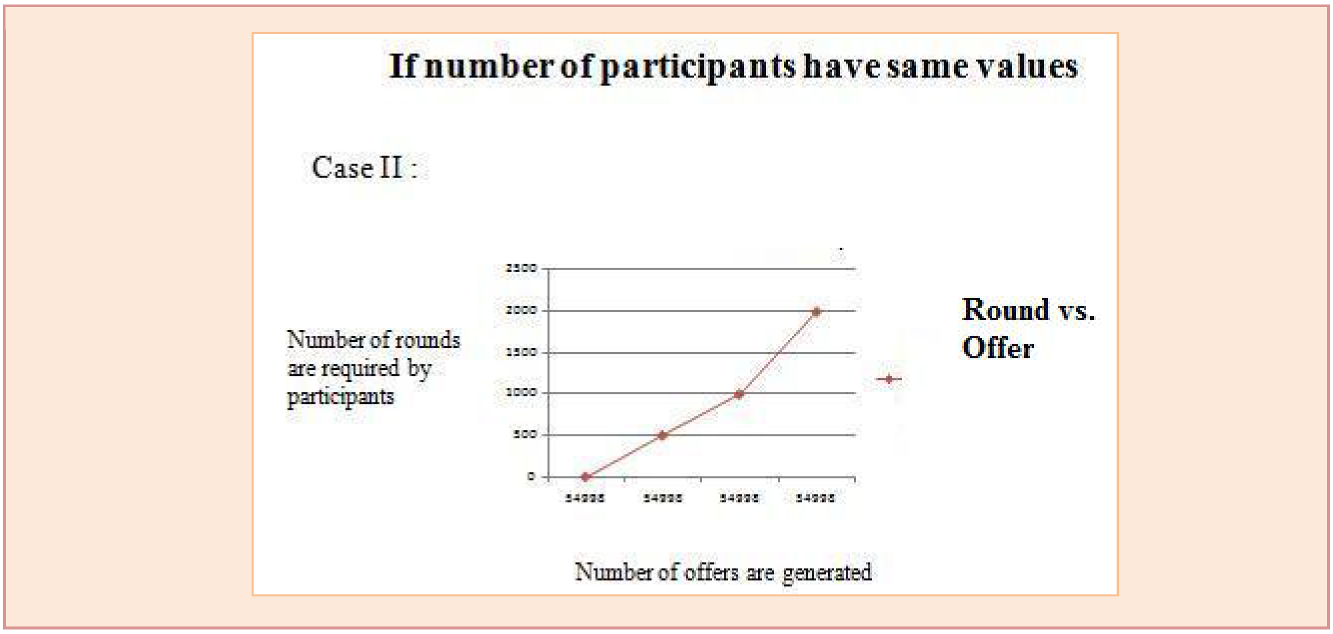

Figure 7. Round vs. Offers (buyer and seller have same values of the object)

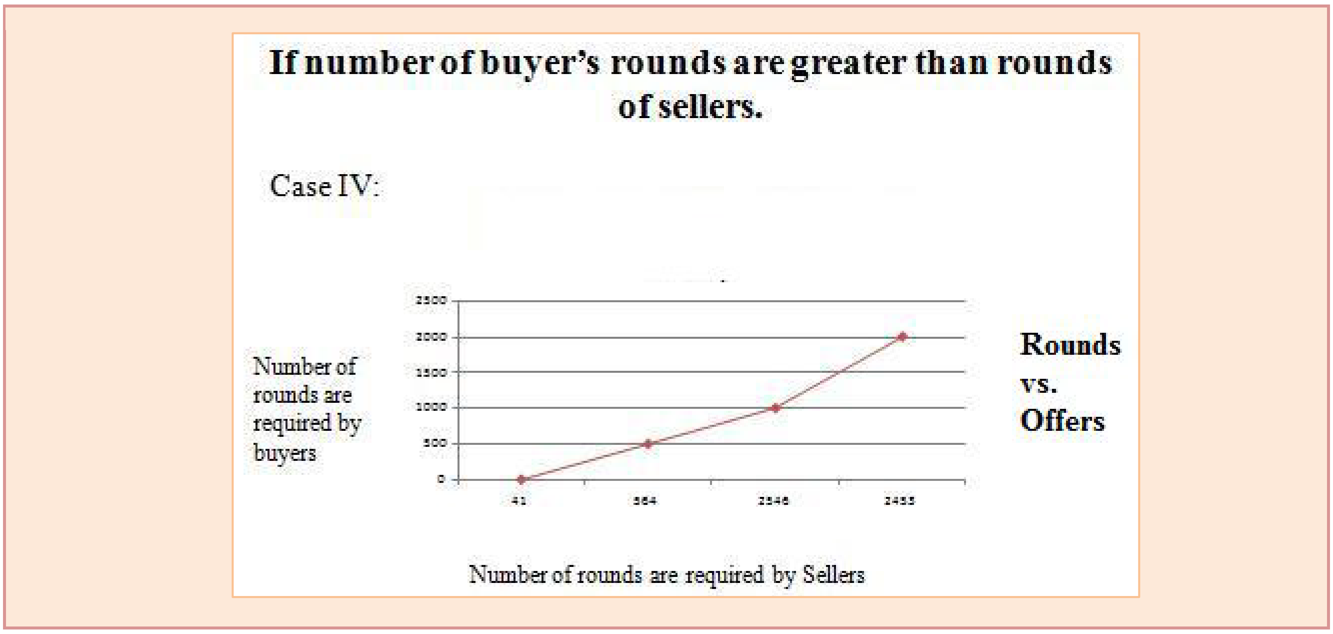

Figure 8. If the buyer's rounds are greater than the seller's rounds

Table 3. Difference between bilateral automated negotiation and multilateral automated negotiation

Sr. No.	Bilateral automated negotiation	Multilateral automated negotiation
1	Only two parties are involved.	More than two parties are involved.
2	It is simpler than multilateral negotiation.	It is complex negotiation.
3	The less number of issues under considerations.	The high number of issues under considerations.
4	Less level of satisfaction to achieve agreements.	High level of satisfaction to achieve agreements.
5	Less time required to do negotiation process.	More time required for negotiation.

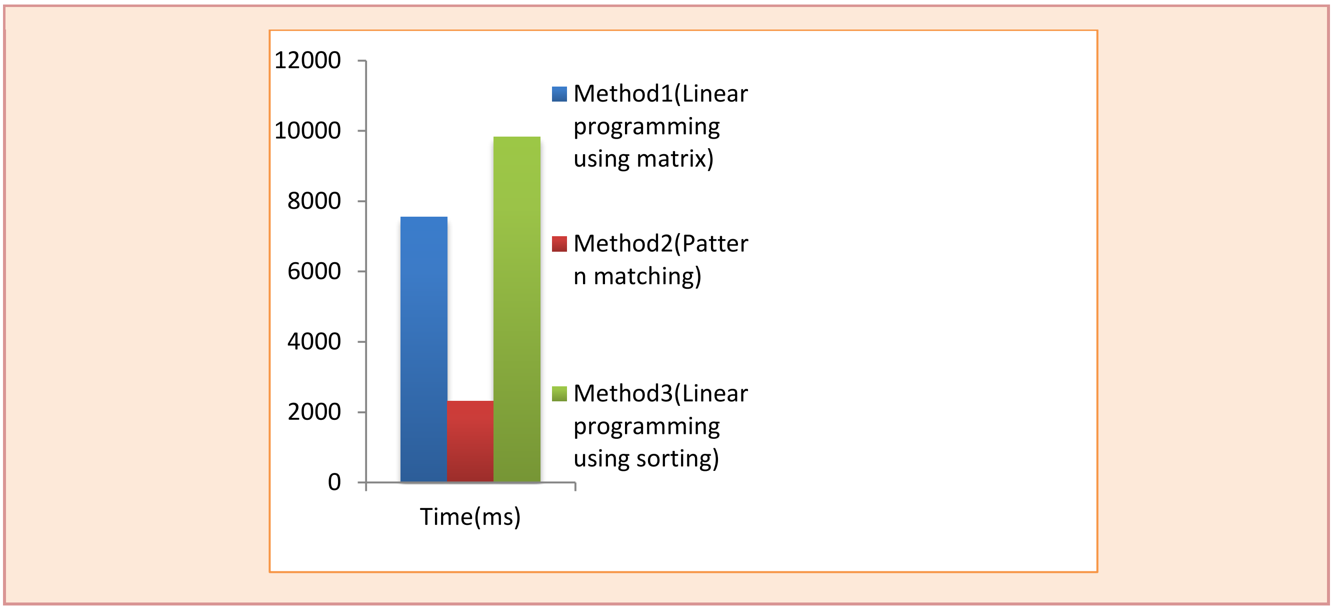

Figure 9. Comparison between methods with their respective times

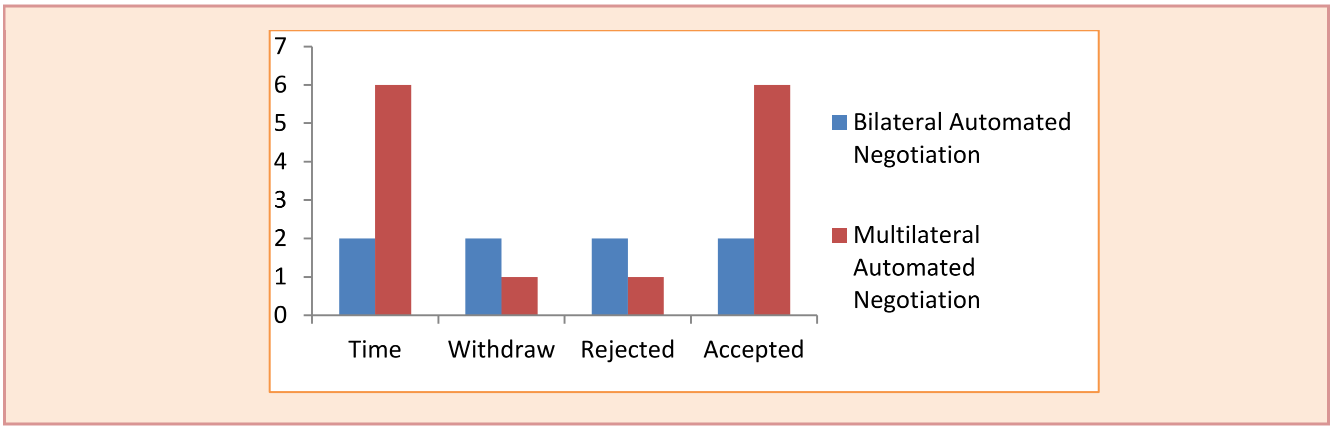

Figure 10. Comparison between bilateral negotiation and multilateral automated negotiation (using random offer generation method: on x-axis no. of participants are plotted (range hundred)).

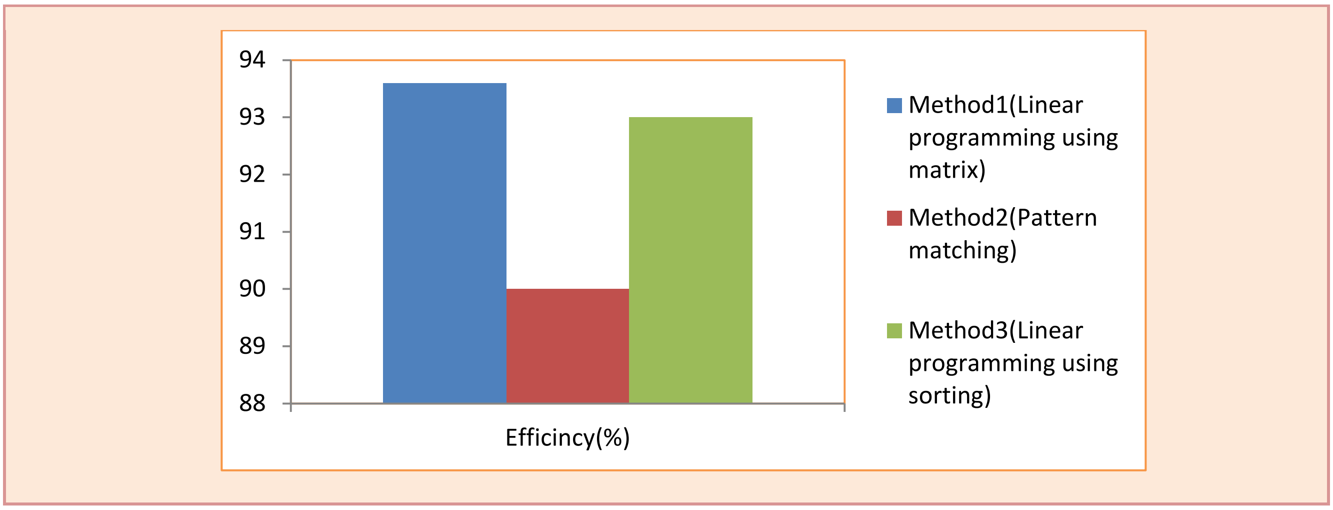

Figure 11. Efficiency of methods

Conclusion

The system is capable of negotiating in a multilateral environment without human intervention. A user is only required at the beginning to specify requirements, which are given as input to the agents. Multilateral negotiation is more complicated and time consuming than bilateral negotiation, because in multilateral automated negotiation, we must do multiple matching between the participants. A multilateral automated negotiation system gives better results in terms of profit than a bilateral automated negotiation system. A pattern-matching technique will give fast results and will reduce the overhead from calculations.

Considering some of the previous papers, we found that a multilateral negotiation system can be developed in fuzzy systems, multithreading, time-dependent systems, and systems in linear programming and genetic algorithms. Various experimental results show that predictive decision-making gives better results in terms of profit for the adaptive negotiation agent, as compared to the range of non-predictive negotiation strategies.

For real-time multilateral automated negotiation, the cloud will be more helpful. The cloud requires low maintenance on data and is more secure, but it is useful for large applications because the cloud is costly.

References

[1] A. More, S. Vij, D. Mukhopadhyay, “Agent Based Negotiation using Cloud - An Approach in E-Commerce”, *48th Annual Convention of the Computer Society of India, CSI Proceedings*, Visakhapatnam, India, Springer-Verlag Germany, pp. 489-496, ISBN 978-3-319-03107-1, 2013.

[2] B. Shojaiemehr, M. K. Rafsanjani, “A fuzzy system approach to multilateral automated negotiation in B2C e-commerce,” *Journal, Neural Computing and Applications*, Springer-Verlag London, UK, Volume 25 Issue 2, pp. 367-377, 2014. Article (CrossRef Link)

[3] Charles J. Thomas, Federal Trade Commission, Harvard Law School, Bart J. Wilson, “A Comparison of Auctions and Multilateral Negotiations,” *Bureau of Economics Federal Trade Commission Washington*, Working pp. 231, 2000.

[4] D. Mukhopadhyay, S. Vij, S. Tasare, “NAAS:Negotiation Automation Architecture with Buyer”s Behavior Pattern Prediction Component,” *The Fourth International Conference on Web & Semantic Technology, NeCoM Proceedings*, Chennai, India, Springer-Verlag, Germany, ISBN 978-3-642-31513-8, pp.425-434, 2012

[5] D. Xue-jie, C. Jian, H. Ying-lan, JIANG, Guo-rui, HUANG Ti-yun, “Multi-attribute Negotiation Model Based on Internal Factor Argumentation,” *IEEE Conference*, Harbin China, ISBN 978-1- 4799-0473-0 , pp. 20-27, 2013.

[6] K. Kurbel, I. Loutchko, F. Teuteberg, “Fuzzy MAN: An Agent-Based Electronic Marketplace with a Multilateral Negotiation Protocol,” *Springer-Verlag Berlin Heidelberg*, pp. 29-30, 2004.

[7] L. Wang, J. Shen, “Multi-phase ant colony system for multi- party data-intensive service provision,” *journal, IEEE Transactions on services computing*, , pp-1-14, 2014.

[8] M. Arun P, S. Vij, R A. Rane, “An Architecture for Multilateral Automated Negotiation: AMAN,” *International Journal of Engineering Research & Technology*, India, Vol. 3, pp. 192-198, ISSN:2278-0181, 2014.

[9] M. Li, P. Li, L. shijun, W. Lei, “A win-win Multi-issue Service Negotiation Model Based on Bayesian Learning,” *Conference on Cloud Computing and Big Data*, pp. 155-162, ISBN: 978-1-4799-2829-3, 2013.

[10] M. I. Bala, S. Vij, “Automated Negotiation and Behavior Prediction,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 2 no. 6, pp. 1832-1838, ISSN: 2278-0181, 2013.

[11] M. I. Bala, S. Vij, D. Mukhopadhyay, “Intelligent Agent for Prediction in E-Negotiation: An Approach,” *Conference on CUBE2013 International IT, CUBE 2013 Proceedings*, Pune, India, pp 183-187, 2013. Article (CrossRef Link)

[12] M. I. Bala, S. Vij, D. Mukhopadhyay, “Negotiation Life Cycle: An Approach in E-negotiation with Prediction,” *48th Annual Convention of the Computer Society of India*, CSI 2013 Proceedings, Visakhapatnam, India, Springer-Verlag Germany, ISBN 978-3-319-03107-1, 2013.

[13] M. Hemaissia, A. E. Fallah, C. Labreuche, J. Mattioli, “A Multilateral Multi-issue Negotiation Protocol,” *Conference on Autonomous Agent and Multiagent System*, Honolulu, Hawaii, USA, pp. 939-946, 2007. Article (CrossRef Link)

[14] P. Faratin, C. Sierra, N. R. Jennings, P. Buckle, “Designing Responsive and Deliberative Automated Negotiators,” *Proc. AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, Orlando, pp. 12-18, 2000.

[15] R. Lin, S Kraus, T. Baarslag, D. Tykhonov, K. V. Hindriks, C. M. Jonker, “An integrated environment for supporting the design of generic automated negotiators,” *Computational Intelligence*, 2012.

[16] R. Aydogan, T. Baarslag, K. V. Hindriks, C. M. Jonker, P. Yolum, “Heuristics for using CP-nets in utility-based negotiation without knowing utilities,” *Springer-Verlag London*, 2014. Article (CrossRef Link)

[17] S. Kraus, “Automated Negotiation and Decision Making in Multiagent Environments,” *ACAI*, *Spinger Verlag*, Barllin, Heidalberg, pp. 150-172, 2001.

[18] S. Deochake, S. Kanth, S., S. Sarode, V. Potdar, D. Mukhopadhyay, “HENRI: High Efficiency Negotiation-based Robust Interface for Multi-party Multi-issue Negotiation over the Internet,” *Conference on CUBE2012*, ACM Pune, India, pp.647-652 ISBN: 978-1-4503-1185-4/12/09, 2012. Article (CrossRef Link)

[19] S. Klaue, K. Kurbel, I. Loutchko, “Automated Negotiation on Agent-Based E-Marketplaces: An Overview,” *14th Bled Electronic Commerce Conference*, Bled, Slovenia, pp 25-26, 2001.

[20] T. Ito, M. Zhang, V. Robu, S. Fatima, Tokuro, “Advances in Agent-Based Complex Automated Negotiations,” *Springer-Verlag Berli Heidelberg*, Tokyo, ISBN 978-3-642-03189-2, 2009.

[21] R. Buttner, “A classification Structure for Automated Negotiations,” *IEEE/WIC/ACM international conference on web intelligence and intelligent technology*, Hong Kong, China, pp. 523-530, 2006. Article (CrossRef Link)

[22] S. Park, S.-B. Yang, “An efficient multilateral negotiation system for pervasive computing environments,” *SICE-ICASE International Joint Conference in Bexco*, Busan, Korea, pp.433-443, 2006.

[23] S. Park, S.-B. Yang, “An Automated System based on Incremental Learning with Applicability Toward Multilateral Negotiations,” *SICE-ICASE International Joint Conference*, Bexco, Busan, Korea, pp. 6001-6006, 2006. Article (CrossRef Link)

[24] U. Endriss, “Monotonic Concession Protocols for Multilateral Negotiation,” *ACM Conference on Autonomus Agent and Multiagent System*, Hokkaido Japan, ISBN:1- 59593-303-4, pp. 392-399, 2006. Article (CrossRef Link)

[25] X. Luo, N. R. Jennings, N. Shadbolt, “Acquiring Tradeoff Preferences for Automated Negotiations: A Case Study,” *Workshop on Agent-Mediated Electronic Commerce V*, Melbourne, Australia, 2003.

[26] M. Patrikar, S. Vij, D. Mukhopadhyay, “An Approach on Multilateral Automated Negotiation;” *4th International Conference on Advances in Computing*, Communication & Control - 2015, ICAC3 2015 Proceedings; Mumbai, India; Elsevier Proceedia Computer Science, USA; April 1 - 2, 2015.

[27] S. R. Vij, A. More, D. Mukhopadhyay, A. Agrawal; “An E-Negotiation Agent using Rule Based and Case Based Approaches: A Comparative study with Bilateral E-Negotiation with Prediction;” Journal of Software Engineering and Applications, Scientific Research, USA; vol.8, no.10; pp.521-530, Oct. 2015.

[28] D. Mukhopadhyay, S. Vij, A. More; “Book Chapter 16: An Efficient E-Negotiation Agent Using Rule Based and Case Based Approaches: An E-Negotiation Agent,” *Advanced Research on Cloud Computing Design and Applications - Book*, IGI Global, USA, pp.245-261, Sep. 2015. Article (CrossRef Link)

Madhur Patrikar has just submitted her ME thesis from the Department of Computer Engineering, Maharashtra Institute of Technology, Pune, Maharashtra, India. Her current research interests include E-commerce.

Prof. Sheetal Vij is an Assistant Professor, Department of Computer Engineering, Maharashtra Institute of Technology, Pune, Maharashtra, India. She has published papers in various international journals and conferences. Her areas of research interest are in automated negotiation, bio-informatics, high-performance computing, and HCI. She holds a BE (Computer Engineering) and an ME (Computer Engineering) from the University of Pune, India. She is currently pursuing her PhD at Nagpur University, India.

Dr. Avinash Agrawal is an Assistant Professor at RCOEM Engineering College, Nagpur Area, India. He holds a B.Tech. from Nagpur University in 1998, an M.Tech. from NIT Raipur in 2005, and a PhD in Computer Science & Engineering from VNIT Nagpur in 2012. His research interests include Natural Language Processing.

Prof. Debajyoti Mukhopadhyay is the Dean (R&D) of the MIT Group of Institutions, and is Head of Information Technology at Maharashtra Institute of Technology, Pune, India. He is the Founder of MITCERI (MIT Center of Excellence for Research & Innovation) to encourage and facilitate R&D activities within the MIT Group. He had earlier assumed the position of Director of the Balaji Institute of Telecom & Management in Pune. He is the Founding Director of the Web Intelligence & Distributed Computing Research Lab. From 2008 to 2010 (for almost three years), he was the Founding Head and a Professor of Information Technology & MIS at Calcutta Business School. He was a Visiting Scholar at George Mason University, Virginia, USA, during June and July 2014. Prof. Mukhopadhyay is a Distinguished Adjunct Professor at Curtin University, Perth, Australia. He holds Adjunct Professorship at the Monarch Business School, Switzerland, the College of Engineering, Pune, India, and Thapar University, Patiala, India. He worked as a full Professor of Computer Science & Engineering at the West Bengal University of Technology's affiliated Engineering Colleges from 2001 to 2008. He was a Visiting Professor at Chonbuk National University in the Republic of Korea (2006-2007). He also taught at Stevens Institute of Technology, New Jersey, USA (1982-1984) and at Bengal Engineering & Science University (1980-1981). He worked as a Research Fellow at the Indian Statistical Institute, Calcutta (1979-1980). From 1982 to 1994 and in 1999, he was in the USA. He worked at Bell Communications Research, USA, in its Computing Systems and Architecture Lab (1987-1994). He has published nearly 160 research articles in international journals, in conference proceedings and as research reports. Prof. Mukhopadhyay holds a BE (Electronics) from the University of Calcutta (India), a DCS (Computer Science & Applications) from The Queen's University of Belfast (UK), an MSc (Computer Science) from the Stevens Institute of Technology (USA) and a PhD (Engineering) in Computer Science from Jadavpur University (India). Prof. Mukhopadhyay is an SMIEEE (USA), SMACM (USA), FIE (India), FIETE (India), C.Eng., SMCSI, MIMA, and an Elected Member of Eta Kappa Nu