

A New Category of Business Negotiation Primitives for Bilateral Negotiation Agents and Associated Algorithm to Find Pareto Optimal Solutions

Haifei Li

IBM Thomas J. Watson Research Center

PO Box 218, Yorktown Heights, NY 10598

Email: haifeili@us.ibm.com, Phone: (914) 945 1740

Abstract

How to conduct automated business negotiation over the Internet is an important issue for agent research. In most bilateral negotiation models, two negotiation agents negotiate by sending proposals and counterproposals back and forth. Proposals and counterproposals usually use a set of negotiation primitives to define the contents. These primitives deal with complete proposals or the whole negotiation process. With these primitives, a negotiation agent can issue a call-for-proposal (CFP), reject or accept the received proposal, propose a new proposal, or terminate the whole negotiation process. However, there is no primitive about individual attributes within proposals. In this paper, we propose a new category of negotiation primitives that have finer granularities than existing primitives do. They are in the attribute level instead of the proposal level, as other primitives are. These messages can help negotiation agents to find Pareto optimal solutions if both sides agree to use them. In order to eliminate non-Pareto optimal solutions, these primitives should be used before the actual negotiation (bargaining) begins. Pareto optimal solutions help both sides to reduce the negotiation effort and time, and these solutions are theoretically preferable to non-Pareto optimal solutions.

1. Introduction

1.1. Negotiation Primitives and Negotiation Process

According to [10], “negotiation is a process by which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession or search for new alternatives”. Business negotiation is the application of general negotiation principles to business settings. We focus on the business negotiations involving corporate buying and selling. Business

negotiation is an important business activity. How to conduct the business negotiation over the Internet is an active research issue in agent research community. Just as human negotiators in the face-to-face negotiation need a vocabulary of business and technology terms in order to describe the demands verbally, it is equally important to have a collection of negotiation primitives in order to describe the demands in the form of proposals. All primitives in the literature we surveyed are in proposal level, making it difficult to find the Pareto optimal solutions. The new primitives presented in the paper will help agents to find the Pareto optimal solutions. We will use negotiation primitive and negotiation message exchangeably in the following discussion.

In order to conduct automated business negotiation, it is important to formalize the negotiation process to a certain degree. There are many ways to transforming an unstructured negotiation process into a format that is suitable for computer processing. A particularly interesting and natural example is called the proposal-counterproposal procedure (also called offer-counteroffer procedure) as defined in [15] and [17]. Since business negotiation, especially negotiation about corporate buying and selling, usually involves only a buyer and a seller, it is assumed that there are two agents NA and NB. At first, NA issues a CFP (Call For Proposal), and NB responds with a proposal. NA reviews the proposal and decides (may involve a complex decision making process) either (1) send a counterproposal; or (2) accept the proposal; or (3) terminate the negotiation process. If (1) is selected, NA evaluates the received proposal, make a decision and send the reply message (acceptance, or counterproposal or termination) to NB. If (2) is selected, an acceptance message is sent to NA asking for mutual acceptance. After receiving the acceptance message from NB, NA also sends an acceptance message to NB for mutual agreement in most cases. However, NA may take other actions. If (3) is selected, NA does not negotiate with NB anymore. Further messages from NB are ignored. When NB receives messages from NA, it acts in a similar fashion. The process continues until either both NA and

NB mutually agree on the business terms or one side (either NA or NB) unilaterally terminates the negotiation process.

Two aspects of negotiation process are identified from the above description. They are communication language and decision making. Communication language (also called communication primitive, negotiation primitive or negotiation message in other literature) deals with how to represent the demands (requirements) the agent wants. In the context of business negotiation, the demand is a proposal that lists all the goods or services and the corresponding business terms such as product specification, price, quantity, and delivery date. It is vital to realize that the demands are not fixed. They are constantly changing during the negotiation process. The buyer normally increases the price gradually, while the seller normally decreases the price gradually. In the context of business negotiation, decision-making is how to make a decision given the message from the other negotiation agent and the internal states, specifically how to generate the response or counterproposal. [6, 18] are papers about a prototypical implementation of a negotiation server based on a set of common negotiation primitives. How to physically transfer a negotiation message from one site to another is beyond the scope of the paper. Current communication infrastructure using CORBA or Java RMI is able to handle the low-level message passing mechanism.

Negotiation process described above has one important assumption that the preferences about attributes are strictly opposite to each other. For example, sellers always prefer high price, and buyers always prefer low price. Therefore, it is clear to the negotiation agent which direction he / she should go. It is beneficial to have a high demand in order to have a large “negotiation space” for bargaining. During the negotiation phase, the agent gradually concedes in the direction that is beneficial to the other negotiation agent. Only through mutual concessions can a negotiation process move to a possible agreement. If the preferences about attributes are not strictly opposite to each other, some values for the attributes are dominated by others, and these non-Pareto optimal values should not be in the final negotiation outcome. In this case, A program to find Pareto optimal solutions that are beneficial to both sides is helpful to both agents. This paper presents an extension to the traditional negotiation messages. These new messages deal with attributes instead of proposals. These new messages are applicable to attributes that do not have exact opposite interest to both agents. The Pareto optimal solutions for attributes that have exact opposite interest to both agents would include all possible values of the attribute, making the solutions trivial. For example, price is an attribute that has exact

opposite interests to sellers and buyers. Therefore, all possible values for price are Pareto optimal solutions.

1.2. Related Work

Speech act theory [1, 16] is a theory of communication and linguistics. The central idea of the theory is that someone uttering a sentence not only makes statements (i.e., describing a state of affairs), but also performs statements. Some systems, especially agent systems, are built based on the theory. FIPA ACL (Foundation for Intelligent Physical Agents, Agent Communication Language, available at www.fipa.org) is such an example. FIPA ACL defines several categories of communicative acts, and negotiation is one of them. FIPA 97 defined about twenty primitives such as “accept-proposal”, “cfp”, “propose” and “reject-proposal”. These are all proposal level, coarse-granularity primitives. They do not go into the attribute level of the proposals exchanged during the negotiation process. Furthermore, the logic-like representation used in ACL (i.e., SL language) is not easy to understand and process. Recently, there is an effort [5] to define an agent communication language based on the XML syntax.

SANP described in [3] is a negotiation protocol based on speech act theory. The paper describes an example application: department budget preparation within a company. The application is relatively simple in the sense that there is no serious conflict of interest between two parties involved, and higher level management is always able to impose its resolution. We don’t know whether it is applicable to business negotiation scenario where the conflict of interest is explicit and serious. Speech act theory is useful in describing the general conversation among human or artificial agents. Negotiation is unique in that agents uttering a sentence may not necessarily perform the statement. Bluffing, threat, promise and ultimatum may not be executed during the negotiation. Agents even have incentive to lie under certain circumstances.

General negotiation theory and practice [2] deals with how real people negotiate in the real world situation. There are some practical negotiation models [10, 11] proposed by experimental and social psychologists. The language in these models is made up of explicit language (spoken words or written documents) and implicit language (body movement, facial expression and gesture). It is extremely hard, if not impossible, to represent implicit language using agent technology. Only a portion of the explicit language can be expressed in the proposal exchanged among negotiation agents.

NSS (Negotiation Support Systems) [8] and GDSS (Group Decision Support System) [7] are the extension of DSS (Decision Support Systems) to the area

of negotiation. NSS and GDSS are mainly studied by researchers in decision theory and business administration. Since the focus of NSS and GDSS is “support” instead of “automation”, the decision-making is largely left to the human negotiators using the systems. Accordingly proposals exchanged are composed manually for human reading. A proposal (or offer) in most NSS contains a set of constant values for different negotiation attributes (or issues). The description of the attributes, and the possible value for the attribute are assumed to be implicitly known to human negotiators.

DAI (Distributed Artificial Intelligence) and MAS (Multi-Agent Systems) [9] is a branch of AI, and studies coordination, synchronization and interaction of multiple agents. Negotiation is an important aspect of DAI and MAS. Business negotiation is different from low-level negotiation such as resource allocation, task assignment where competition is not a serious issue. [14] contains the designing conventions for negotiation agents, but some of the assumptions made are not valid for business negotiations.

The concept of “Pareto optimality” was invented by Vilfredo Pareto (1848 to 1923). Pareto was an Italian economist and sociologist, and proposed the concept in his most influential work *Manuale d'economia politica* (1906). The modern treatment of Pareto optimality can be found in [4, 19].

The rest of the paper is organized as follows. Section 2 is the problem description and its solution. Section 3 shows the algorithm step by step. Section 4 compares our approach to find Pareto optimal solutions with that of NSS. Section 5 concludes the paper.

2. Problem Description and Negotiation Messages

Bilateral negotiation involving two negotiation agents is considered in the paper. In order to have a clear presentation of the idea, it is assumed that the proposal has a simple model. Although the negotiation attributes (issues) vary from one negotiation to another, it is assumed that during a specific negotiation, the number and nature of the attributes are fixed. The messages exchanged between two negotiation agents include a set of predetermined attributes with their values. Therefore, a proposal is a set of attribute / value pairs. For example, in the negotiation of digital camera purchase, we have the following attributes to be negotiated: price, quantity, delivery_day, pixel resolution. One proposal may look like: (price = 700.00USD, quantity = 500, delivery_day = 3 weeks, pixel resolution = 1280 X 1024).

A proposal becomes a solution if the proposal proposed by one side is accepted by the other side. There are many possible solutions, and some are Pareto optimal

and some are not. Simply speaking, Pareto optimal solutions are solutions that are not “dominated” by other solutions. It is necessary to explain what exactly “domination” means. Our discussion limits to two-agent situation where bilateral negotiation fits in. Pareto optimal solutions deal with two agents that have opposite interests. A (possible) solution is an assignment of values to issues involved. Apparently, there are many possible solutions. If a 2D diagram is used to show the utilities to both agents, the solutions are corresponds to points in the 2D space. Figure 1 shows the two dimensions. The points in the upper right space correspond to the possible solutions. The Pareto frontier is made up of points that can not expand up-right any further. Solution S1 “dominates” solution S2 if S1 would improve both parties, or improve one party and does not harm the other. It is clear from figure 1 that C “dominates” A and D “dominates” B. E “dominates” both A and B. However, A does not “dominates” B and vice versa.

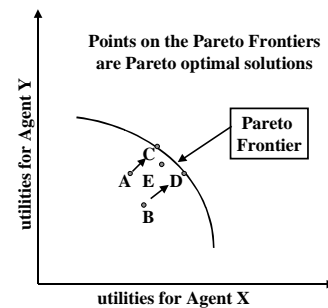


Figure 1. Pareto optimal solutions

Pareto optimal solutions are useful to bilateral negotiation. It is clear that the final agreement should be Pareto optimal. Otherwise, a Pareto optimal solution that dominates the final agreement should be preferable to rational agents. However, it is important to point out the Pareto optimal solutions is usually a set that contains multiple solutions. It is up to the agents to choose which one to be the final agreement. Both C and D are Pareto optimal solutions in Figure 1.

As discussed in the introduction section, the messages exchanged between negotiation agents are usually related to a complete proposal. That is to say, when proposal / counter-proposal are exchanged between two negotiation agents, all attributes must have values. There are two shortcomings with the approach. First, it forces negotiation agents to consider about all negotiation attributes at the very beginning, a difficult task from the computational complexity perspective. Second, there is possibility that the final result is not

Pareto optimal. One metaphor is that “money is left on the negotiation table and nobody claims it”.

In order to remedy the shortcomings, we propose a new category of negotiation messages. These messages deal with individual attributes instead of a complete proposal. These messages are DoYouAgree_ParetoOptimal, IAgree_ParetoOptimal, IDisagree_ParetoOptimal and Process_NegotiationAttributePreference. These messages are exchanged among negotiation agents immediately after the agents are agreed to use the negotiation system. The first three messages are auxiliary messages to establish the consensus between negotiation agents. The last message will pass the attribute preferences around. If agent A gets the preferences of agent B, agent A is able to calculate the Pareto optimal solutions, and vice versa. The negotiation agent that sends DoYouAgree_ParetoOptimal implicitly agrees to use the method. If the other side responds with IAgree_ParetoOptimal, both agents are able to make use of the associated algorithm to do the work. If the other side responds with IDisagree_ParetoOptimal, the algorithm will not be used. The following figure 2 shows the case where the receiver negotiation agent agrees to use the method to find the Pareto optimal solutions. If the receiver does not agree to use the method, two Process_NegotiationAttributePreference messages will not be shown. Please notice that each Process_NegotiationAttributePreference message only carry the preference for one attribute. If there is a need to find Pareto optimal solutions for multiple attributes, multiple rounds of message exchange (of Process_NegotiationAttributePreference) is needed.

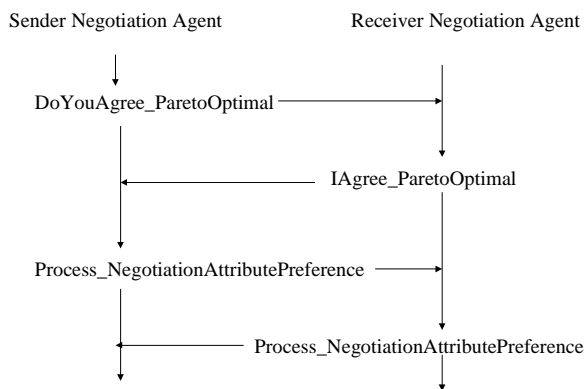


Figure 2. Process flow when both sides agree to use the method.

The purpose of these messages is to communicate the relative importance of the attribute value. In general,

agents are willing to tell the other side his / her preferences. The preference for attributes such as price is implicit. Buyer prefers to a lower price and the seller prefers to a higher price. If the new primitive is used, and both sides are willing to accept the Pareto optimal solution, there is an algorithm to generate the Pareto optimal solution set. The negotiation narrows down to the values in the sets.

Delivery schedule is a good example that may use the new message and the algorithm presented in the paper. Modern enterprises make extensive use of ERP and other enterprise information systems for production planning and execution. It is quite possible that the enterprise's preference over delivery is not as early as possible (for buyer), or as late as possible (for supplier). For example, In JIT (Just-In-Time) manufacturing, the raw materials must arrive at the specified time, therefore, the early arrival time or late arrival time has preferences less than the preference for the optimal arrival time. The consequence is that the preference over the delivery schedule is NOT a monotonically increasing or decreasing function over the time. Figure 3 shows the relationship between buyer's and seller's delivery schedule.



Figure 3. Delivery schedule of seller

The preference from the buyer's side is 2, 6, 1, 5, 4, and 3. The preference from the seller's side is 3, 4, 2, 1, 5, and 6. There is an algorithm to find the Pareto optimal solutions to the problem.

3. Algorithm to Compute the Pareto Optimal solutions

Following is the description of the algorithm:

Input: inputListA and inputListB for agent A and agent B's preferences. List elements are arranged from the most preferable to the least preferable.

Output: outputListA for side A and outputListB for side B.

Steps:

1. Initialize both outputListA and outputListB to empty list,
2. Set pointerA to the first element of inputListA,
3. Select one element from inputListA, call the element eleA, let pointerA point to the next element in inputListA,
4. Find the element in inputListB that has the same value as eleA by scanning inputListB starting from the first element, call the element found in inputListB eleB,
5. Get the ancestor set ancestorSetA of eleA in inputListA. This can be done by traversing from the first element of inputListA down to the element before eleA,
6. Get the ancestor set ancestorSetB of eleB in inputListB. This can be done by traversing from the first element of inputListB down to the element before eleB,
7. Check whether the intersection of ancestorSetA and ancestorSetB is empty. If the intersection is empty, insert eleA to the end of outputListA. Otherwise eleA (also eleB) is dominated by other elements, so it is not added to outputListA,
8. Check whether pointerA points to null. If it is, go to step 9, otherwise go back to step 3,
9. Copy outputListA to outputListB, reverse elements in outputListB,
10. Output outputListA and outputListB.

It does not matter whether the lists are represented as array-based lists or linked lists. However, data structure “set” is not good for the representation since the relative order among elements, which is very important to our algorithm, is not captured. OutputListA and outputListB give the orders that both A and B should follow when it is time for one side to make a concession. However, Pareto-optimal solution list does not tell the agent whether it is good to make the concession or not. The decision to make such a concession is up to the agent.

In the example, the Pareto optimal solution is (2, 4, 3). If buyer is negotiating on this attribute, he / she will gradually relax on the order of 2, 4, and 3. On the other hand, the seller will gradually relax on the order of 3, 4

and 2, exactly the opposite order for the buyer. Other solutions are “dominated” by others. For example, 1 is “dominated” by 2. If we move the solution from 1 to 2, both seller and buyer are better off.

Time complexity analysis is not difficult to figure out. Steps from 3 to 8 forms a loop. Step 7 takes $O(n * n)$ in the worst scenario. Therefore the algorithm takes $O(n * n * n)$ at most, where n is the number of possible values for the attribute.

The following observations are made:

1. The Pareto-optimal solution set is not empty if the value set is not empty.
2. If the preferences of one agent along a dimension are monotonically increasing, and the preferences of the other agent along the same dimension are monotonically decreasing, every value in the dimension is in the Pareto-optimal solutions set.
3. If both agents prefer the same value along a dimension, the Pareto-optimal solutions contains only one element, which is the value preferred by both agents.

4. Comparison with NSS Approach to Pareto Optimal Solutions

In the typical NSS, the Pareto optimal solution is often given at the end of the negotiation process. NSS follows the steps: pre-negotiation, negotiation process, Pareto optimal solution suggestion. The last step is inspired by the ideas proposed in a paper by Howard Raiffa [12]. Even though NSS does not use agent technology, the last step is able to be assigned to a third-party agent as long as the preferences of both sides are known.

We do think it is a good idea to have the last step, but there are two major shortcomings. First, the method does not reduce the negotiation time and effort since time and effort have been spent already before the suggestions are ready. It is a good idea to avoid the generation of non-Pareto optimal solutions from the very beginning. The second shortcoming is closely related the first one. Since negotiation agents have spent much effort on the negotiation, they may be not willing to accept the solution suggested by the computer. The acceptance of the solutions suggested by the computer is a hint that the previous negotiation outcome seems to be less useful.

It is an even better idea if the non-Pareto-optimal solutions are eliminated after the preferences are gathered and before the negotiation begins. Let us have a look at the approach presented in [13]. Preferences of the two parties are presented in Table 1. Only preferences of two issues are shown here:

Currency		Dispute Settlement	
EE	HC	EE	HC
Hungarian	U.S. \$	Hungary	U.S. court
Euro \$	Other hard	ICC	London
U.S. \$	Euro \$	London	ICC
Other hard	Hungarian	U.S.court	Hungary

Table 1: Preference of two issues, a part of Table 1 in paper [13]

Brief background information is needed in order to understand the context. The paper presents a negotiation scenario that takes place between EE (East European Medical Equipment Company) and HC (Healthcare, Inc). EE is the buyer in Hungary and HC is the supplier in US. Four issues are to be negotiated: Price, Delivery, Currency and Dispute settlement. The above table only shows two issues. The preferences are from the top, which is the most preferable, to the bottom, which is the least preferable. For example, the preference for EE over the issue Currency (from the top to the bottom) is Hungarian, Euro \$, U.S. \$, Other hard. "Other hard" means hard currencies other than Hungarian, Euro \$ and U.S. \$. It is understandable that EE, which is a Hungarian company, prefers Hungarian currency most.

The problem lies in the choice of "Other hard" in the issue Currency. It is found that EE and HC have exactly opposite preferences over "Dispute settlement". All choices for "Dispute settlement" are in Pareto-optimal solutions. However, it is not the case for "Currency". It is easy to figure out that "Other hard" is dominated by "U.S. \$". Therefore it is beneficial to both side if "Other hard" is NOT selected. NA (Negotiation Assistant, the system presented in the paper) proposes Post-settlement option at the end of the bargaining phase. Table 2 shows the possible options. It is apparent that option 2 and 6 should be deleted because they are "dominated" by others. However, it would be better if the NA can take advantage of the approach presented above, and tell both sides that "Other hard" should NEVER be selected even before the bargaining phase begins.

Option	Delivery	Price	Currency	Dispute Ratings
1	14 months	225,000 US \$	US	0
2	14 months	225,000 Other hd	London	0
3	12 months	225,000 Euro \$	US	0
4	14 months	195,000 US \$	US	0

5	14 months	210,000 Euro \$	ICC	0
6	12 months	210,000 Other hd	Hungarian	0
7	12 months	195,000 Hungarian	London	0
8	8 months	225,000 US \$	Hungarian	0

Table 2: Ratings of conjoint packages, a part of Figure 1 in paper [13]

5. Conclusion

We present a new set of primitives and associated algorithm to calculate the Pareto optimal solutions in the bilateral negotiation environment. The algorithm is used in the situation where two negotiation agents' preferences are not strictly opposite. We have shown that it is useful to use the algorithm before the bargaining begins. The use of the algorithm can reduce the effort to reach a Pareto optimal solution.

REFERENCES

- [1] J. L. Austin, *How to Do Things with Words*. Oxford University Press. Oxford, England, 1962.
- [2] J. W. Breslin and J. Z. Rubin, *Negotiation Theory and Practice*. The Program on Negotiation at Harvard Law School, Cambridge, MA, 1991.
- [3] M. K. Chang, and C. C. Woo, "A Speech-Act-Based Negotiation Protocol: Design, Implementation, and Test Use," *ACM Transactions on Information Systems*. vol. 12, no. 4, pp. 360-382, 1994.
- [4] A. Goicoechea, D. R. Hansen and L. Duckstein, "Multiobjective Decision Analysis with Engineering and Business Applications", John Wiley & Sons, New York, 1982.
- [5] B. N. Grosz and Y. Labrou, "An Approach to using XML and a Rule-based Content Language with an Agent Communication Language," In *Proceedings of the IJCAI-99 Workshop on Agent Communication Languages (ACL-99)*, Stockholm, Sweden, Aug. 1, 1999.
- [6] J. Hammer, C. Huang, Y. Huang, C. Pluempitwiriwaj, M. Lee, H. Li, L. Wang, Y. Liu, and S. Y. W. Su, "The IDEAL Approach to Internet-Based Negotiation for E-Business," 16th International Conference on Data Engineering, San Diego CA, 2000.

- [7] N. I. Karacapilidis, and C. P. Pappis, "A framework for group decision support systems: Combining AI tools and OR techniques," *European Journal of Operational Research*, vol. 103, no. 2, pp. 373-388, 1997.
- [8] L. Lim, and I. Benbasat, "A Theoretical Perspective of Negotiation Support Systems," *Journal of Management Information Systems*, vol. 9, no. 3, pp. 27-44, 1993.
- [9] G. M. P. O'Hare, and N. R. Jennings, eds., *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, Inc., New York, NY, 1996.
- [10] D. G. Pruitt, *Negotiation Behavior*. Academic Press, New York, NY, 1981.
- [11] H. Raiffa, *The Art and Science of Negotiation*. The Belknap Press of Harvard University Press, Cambridge, MA, 1982.
- [12] H. Raiffa, "Post-settlement settlement," *Negotiation Journal* Vol. 1, No. 1, pp. 9-12, 1985.
- [13] A. Rangaswamy, and G. R. Shell, "Using Computers to Realize Joint Gains in Negotiations: Toward an Electronic Bargaining Table," *Management Science*, vol. 43, no. 8, pp. 1147-1163, 1997.
- [14] J. S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT press, Cambridge, MA, 1994.
- [15] A. Rubinstein, "Perfect Equilibrium in a Bargaining Model," *Econometrica*, vol. 50, pp. 97-110, 1982.
- [16] J. R. Searle, *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, New York, NY, 1969.
- [17] I. Ståhl, *Bargaining Theory*. the Economics Research Institute, Stockholm, Sweden, 1972.
- [18] S. Y. W. Su, C. Huang, and J. Hammer, "A Replicable Web-based Negotiation Server for E-commerce", in *Thirty-third Hawaii International Conference on Systems Science (HICSS-33)*, Wailea Maui, Hawaii, Jan. 4-7, 2000.
- [19] P. Vincke, "Multicriteria Decision Aid", John Wiley & Sons, New York, 1992.