ABSTRACT

In multi-issue negotiation, agents might have potentially infinite number of different concessions and proposals to make, it may lead to a very inefficient process before a joint agreement can be reached. In this paper, we assume agents are cooperative negotiators under bounded number of negotiation messages. Namely they are motivated to reach a compromized agreement within the message bound in order to get as close to optimal solution as possible. In particular, we implement agents who could incrementally learn from other agent’s proposal during negotiation in order to speed up the negotiation process. We evaluate their performance in terms of Pareto efficiency, total utility payoffs, and number of negotiating messages. We showed by experiments that negotiation learning agents could reach Pareto efficiency agreement that normally are better off for both agents in a much faster speed than such non-learning negotiating agents as simple random agents, rational agents, and cooperative agents.

1. INTRODUCTION

Recently, the interaction and negotiation among agents is considered as one of the key elements in multi-agent system [4][14]. It is important because the agents are autonomous and have their own goals and negotiation is a way for them to resolve their conflicts and find a joint agreement. The topic of agent negotiation has been studied in various aspects and under various assumptions by many researchers. Certain theoretical backgrounds on negotiation have been well discussed in the game theory literature [7][01][15][17]. Other approaches [6][8][18] allow negotiation agents to make decision on utility payoff functions in order to fit certain constraints of negotiation. In [1][11][12], the agents are designed to make decision on different linear combination of simple functions called “tactics”. However, the negotiating agents may still lead to an inefficient negotiation process. In [2], they adopted Bayesian probabilistic reasoning, agents can predict the behavior of opponent and thus could take advantage of the result. In [19][20], they consider negotiation under time constraint and agent could take the constant discount rate and the cost of delay into consideration of making proposals during negotiation. In [5], they employed case-based reasoning in negotiation that relies on accumulation of prior cases and information. It might not be suitable for on-line incremental learning in dealing with one-shot game. Furthermore, most researches focused on single issue only.

We study the learning problems in bilateral multi-issue negotiation. In multi-issue negotiation, the potential negotiation space to be searched is quite large. Since negotiation agents are normally assumed to be rational, they will tend to resist to concede during the negotiation. So it will lead to a prolong negotiation before an agreement can be reached. However, if the negotiation time is a concern and only finite number of messages is allowed for both agents, they will have to behave strategically in order to reach a good negotiation result within the restricted message bounds. We assume also the negotiation agents know their own multi-issue utility functions but do not know their opponent’s utility functions and they only accept proposals above certain reservation utility level.

Our concerns are:

- Can the incremental learning techniques help to speed up in such a negotiation search process?
- And can the learning techniques guarantee the quality of the negotiation results?

We implemented 4 types of negotiation agents: A random agent, a rational agent, a cooperative agent, and a learning agent. In order to evaluate the learning performance in the negotiation, we conducted 16 different settings of negotiation experiments based on the combinations of these 4 types of negotiation agents.

2. THE AUTOMATED NEGOTIATION MODEL AND NEGOTIATING AGENTS

The common components of an automated negotiation model include negotiation objectives, negotiation protocol, and negotiating agents. In bilateral negotiation, rational agents are assumed seek to find an agreement that can maximize their utility payoffs. But we also want to design the negotiation mechanism that could reach as close to the Pareto optimal frontier as possible. The Pareto efficiency [3] is a measure of how close the final agreement is to the Pareto optimal frontier. In order to model the proposal action space of the negotiating agents, we assume the agents have a preference weight on each issue. As the combination of weights and the issues, the utility function for agents will create many utility levels known as an iso-value curve for level [1]. And all the potential proposals will build up an action space of that agent. In Figure 1, it shows the distribution of the proposal action space of the negotiating agents. We could see that in this example, at the utility levels between 0.5 and 0.55, the agent has the most number of proposals. However, rational agents would only consider those proposals that have a higher
utility level than their reservation utilities. Therefore, the feasible action space should consist of only those proposals whose utilities are above the line of reservation utility (say, RU = 0.6).

The negotiation protocol ensures agents communicate in a common standard. We assume the negotiating agents are rational and initiate the proposals with respect to their maximum utility payoffs. If the other agent accepts the proposal, the negotiation ends with the joint agreement. If not, the agent keeps on proposing the next proposal, which tends to preserve at the same utility payoff level as previous one. If such an alternative proposal does not exist, the agents are forced to make concession. However, in order to keep the maximum utility, the agents will only make as little concession as possible. And finally, if both agents do not accept any proposal and no one wants to propose any more (one of the agents sends a withdrawal message), then it results in a failure negotiation and then both of agents gain nothing. However, if the communication message is a critical cost to the system, the system may claim the message constraint to the negotiating agents. And if the message constraint has reached and no joint agreement exists, the system would drop both agents out and result in a failure negotiation.

![Figure 1. The action space and reservation utility](image)

In the following negotiation protocols of four types of negotiation we use the notations: the subscript s represents self agent; o represents opponent agent; U(.) represent the utility function; RU represents the reservation utility and P represents a proposal respectively.

### 2.1 Simple random agents

The simple random agent constantly makes its proposal at a random utility level in its action space. The only restriction for the suggested proposal is that it must have a better utility payoff than its personal reservation utility level. The algorithm is as follows.

Negotiation Protocol of simple random agents

\[
\begin{align*}
\text{If } U_s(P_o) \geq RU_o & \quad \text{accept } P_o \\
\text{Else} & \\
\text{reject and counter-propose } P_s \text{ randomly from the action space } & \\
\{P_s| U_s(P_s) \geq RU_s\}
\end{align*}
\]

### 2.2 Rational agents

The rational agent acts as a utility maximizer and is a selfish agent. It prefer proposals with the highest utility level to that with the lower utility levels. If the agent finds an alternative proposal with the same utility level as previous one, it will always propose that. If such a proposal does not exist, it will make concession as small as possible with respect to its action space. Namely, it is very reluctant to make concession. The algorithm is summarized as follows.

Negotiation Protocol of rational agents

\[
\begin{align*}
\text{If } U_s(P_o) \geq RU_o \quad \text{then accept } P_o \\
\text{Else} & \\
\text{reject and propose } P_s \text{ that has the same utility level as the } & \\
\text{previous proposal } P_s' \text{ if such a } P_s \text{ exists} & \\
\text{Else} & \\
\text{propose } P_s \text{ at the next lower utility level than the previous } & \\
\text{proposal } P_s' \\
\end{align*}
\]

### 2.3 Cooperative agents

The cooperative agents always choose a proposal that has the shortest Euclidean distance with respect to its opponent’s proposal. This is similar to the notion of similarity of proposals in [12]. Namely, they will try to find a proposal that is likely to be as close to their opponent’s proposal as possible. The algorithm is summarized as follows.

Negotiation Protocol of cooperative agents

\[
\begin{align*}
\text{If } U_s(P_o) \geq RU_o \quad \text{accept } P_o \\
\text{Else} & \\
\text{reject and propose } P_s \text{ at the current utility level that has the } & \\
\text{shortest Euclidean distance to the proposal } P_o \text{ if such a } P_s \text{ exists} & \\
\text{Else} & \\
\text{propose } P_s \text{ at the next lower utility level that has the shortest } & \\
\text{Euclidean distance with respect to } P_o \\
\end{align*}
\]

### 2.4 Learning agents

The learning agents could learn from opponent’s negotiation behaviors. The main difference between learning agents and cooperative agents is that the learning agents could adjust incrementally the weighing models on issues that are proposed by their opponents while the cooperative agents only make fixed estimate (shortest Euclidean distance) in response to opponent’s proposal. The algorithm is summarized as follows.

Negotiation Protocol of learning agents

\[
\begin{align*}
\text{If } U(P_o) \geq RU_o & \quad \text{accept } P_o \\
\text{Else} & \\
\text{reject and propose } P_s \text{ at current utility level that has a minimal } & \\
\text{weighted least square sum with respect to the previous } & \\
\text{proposal } P_s' \text{ if such a } P_s \text{ exists} & \\
\text{Else} & \\
\end{align*}
\]
propose $P_s$ at the next lower utility that has the minimal weighted least square sum with respect to the previous proposal $P_s'$.

[Note: The weightings are incrementally learned by the Perceptrons.]

The agent will tend to choose the proposal that is at the same utility level with respect to previous proposal but with Minimal Weighted Least Square (MWLS) $= \min \{WLS\}$.

The Weighted Least Square (WLS) is computed as:

$$WLS = \sum_{i,j} \text{Weight}_i \times (\{x^i\}_j - y^i_j)^2$$

The $\{x^i\}_j$ means all the proposals $\{X^i\}_j$ of agent $i$ on issue $j$ at the current utility level and $y^i_j$ is the opponent’s proposal on issue $j$. Since the negotiating agent does not know the exact Weight$_i$ for its opponent a priori, it needs to learn during negotiation.

2.4.1 The method of Perceptron learning

The learning agent records the continuous proposal sequence and learns the weights of issues by its opponent agent using a simple Perceptron. The proposal that the agent offers but is rejected by its opponent is a negative instance while the counter proposal from its opponent is a positive instance because the counter proposal must be at least accepted by its opponent. The Perceptron architecture is given in Figure 2. The Perceptron is trained by an iterative Perceptron learning algorithm.

![Figure 2 The Perceptron learning network architecture](image)

The Perceptron Learning Algorithm

Step 0. Initialize the weights (weights are initially uniformly distributed)

Set the learning rate $\alpha$ to 0.1

Step 1. For every self-proposal $s$ that is rejected and receives a counterproposal $p$, for a positive and negative training pair $(p,s)$, do Steps 2-4

Step 2. Set activation of input units:

$$x_i = U_j(s_i)$$

(The function $U_j$ is utility function on issue $j$ for agent $i$)

Step 3. Compute response of Perceptron’s output unit:

$$y_{\text{in}} = \sum x_i w_j$$

$$y = -1 \quad \text{if } y_{\text{in}} > \Phi$$ (If the Perceptron’s prediction $y_{\text{in}}$ is far from the counterproposal)

$$y = 1 \quad \text{if } y_{\text{in}} \leq \Phi$$ (If the Perceptron’s prediction $y_{\text{in}}$ is close to the counterproposal)

($\Phi$ is the threshold for checking if $y_{\text{in}}$ is far from the target $t$, in our simulation, it was set to 0.1)

Step 4. Update the weights if $y \neq t$

$$w_i(\text{new}) = w_i(\text{old}) + \alpha \ast (t-y) \ast x_i$$

2.4.2 The utility functions of agents

The utility functions for each agent we adopted are extended from [9] and are used to model the action space and are assumed to be in a continuous form with respect to each issue $j$. The value $P$ can represent the risk attitude of the negotiating agents over each issue (e.g. $P=1$ risk neutral, $P>1$ risk seeking, $P<1$ risk averse). In our experiment, for simplicity, we set $P$ to 1 (risk neutral). Obviously, the combination of each issue’s utility function and different values of $P$ could result in different kinds of action space of the agents and various zones of the possible agreements.

3. THE SETTING OF EXPERIMENTS

In order to evaluate the performance of our agents during negotiation, we classify our experiments into 2 parts – the negotiating agents without message constraints and the negotiation under message constraints. Agent without message constraint can make infinite number of proposals that eventually could lead to an agreement if it exists. With message constraint, however, negotiation may fail (without finding an agreement) if the message limitation has reached. With message constraint, it is therefore important for agents to choose their proposals more cautiously.

The experiment proceeds as follow:

1) Normalize agents’ weights on the issues so that they are summed to 1. Number of issues is set as a constant 5. Set up agents’ reservation utilities (RU=0.6), and their types (Simple random, Rational, Cooperative, Learning)). So there are total 4 by 4 = 16 types of bilateral negotiations.

2) Agents make proposals and counter proposals in turn according to their strategies and negotiation protocols.

3) The system counts the messages and measures the payoffs for each agent in experiments and ends if the message bound (=500) bound is reached or an agreement is found.
When negotiation ends, evaluate the performance of agents. We evaluate the performance of negotiating agents by three aspects: 1) the number of negotiation messages (proposals and counterproposals) used between the negotiating agents; 2) the total utility payoffs; and 3) the Pareto efficiency. The number of negotiation messages is a measure of communication cost. The total payoff is a measure of satisfactions of rationalities for both agents.

4. EXPERIMENTAL RESULTS

To compare the outcomes of the negotiations among different types of agents, we evaluate the results from three aspects: the numbers of negotiation messages, total utility gains, and the Pareto-optimality.

4.2 The negotiation without message constraint

The agents without message constraint can always reach an agreement as long as the negotiation last for long enough. In our experimental settings, it guarantees the existence of agreements. Figure 3 shows the experimental results in which both agents are of various types and their reservation utility levels are at 0.6.

![Figure 3 Results for negotiation with RU=0.6. The number (1) ~ (4) and symbol (a) ~ (d) represent X and Y as a [simple, rational, cooperative, learning] agent in the bilateral negotiation respectively.](image)

And in Figure 4, we evaluate the performance at the agreements in terms of the number of negotiation messages for different combinations of types of negotiating agents. The experiments of negotiation involving with simple random agents tend to reach an agreement in less number of negotiation messages but will lead to the outcome with less utility gains and Pareto-inefficiency of the negotiation (far from the Pareto optimal frontier). The simple agents act with randomly selecting proposals in their action space. The only advantage for simple agents is that they can always reach the joint agreement fast. The experiments of negotiation involving both rational agents showed that the final agreement always lies on the Pareto optimal frontier but would cost a lot of negotiation messages to come up with an agreement. In addition, the agents who made concessions faster usually got less utility payoff than those who resisted to concede. The experiments of negotiation involving cooperative agents showed that they tended to take less number of messages to reach an agreement than those involving rational agents. But the negotiated results sometimes may be a little away from the Pareto optimal frontier. The payoff results often make no much difference from those obtained by rational agents.

![Figure 4. Message costs for various types of negotiation agents](image)

4.1 The negotiation with message constraint

Imposing the message constraints on negotiating agents, it may cause a failure negotiation, namely, no joint agreement can be made during negotiation. The experimental results are shown in Figure 5 and the learning agents tend to perform better than the rational agents and cooperative agents with respect to message cost and the Pareto-efficiency. Figure 6 shows the performance of various kinds of agents during negotiation in terms of number of message cost.

![Figure 5 and Figure 6](images)
Figure 5 Traces of negotiation among different types of agents at RU=0.6 and message constraint is 500. The number (1) ~ (4) and symbol (a) ~ (d) represent X and Y as a [simple, rational, cooperative, learning] agent in the bilateral negotiation respectively.

When the experiments are involved with simple random agents, the joint agreement is made by randomly selecting the proposals from their action space. Sometimes the simple random agents reach the agreement very fast but sometimes they fail in the negotiation. Even an agreement is made; the result is generally not good in the sense of the total utility gains and the Pareto-efficiency. The traces on the lower right corner represent the negotiation trace of simple random agent X as shown in Figure 5 (1a)(1b)(1c)(1d).

When the experiments are involved with rational agents, the rational agents may still fail in the negotiation under the message constraint. Even the rational agents may reach a joint agreement before the limitation on the number of negotiation message has reached; the total utility gains may not be high enough and the joint agreement usually doesn’t lie on the Pareto optimal frontier. The traces on the lower right corner represent the negotiation trace of the rational agent X as shown in Figure 5 (2a)(2b)(2c)(2d).

The results of experiments with cooperative agents are much better than those with rational agents in terms of the message cost. The cooperative agents reach the agreement with a higher probability and with obviously less transferring messages cost as shown in Figure 6 but with the same total utility gains in comparison to the rational agents. The Pareto-efficiencies of the negotiation results are usually a bit better than those with the rational agents. As in Figure 5 (3b)(3c), the proposal sequence for cooperative agents (lines at the lower right corner representing the traces for cooperative agent X) is much shorter than that for the rational agents and the negotiation result is much closer to the Pareto optimal frontier. It means that the cooperative agent usually choose a better proposals to both agents.

The experiments with learning agents result in a successful negotiation with less message cost than other types of agents above. It also results in an efficient negotiation (the agreement reached is always much closer to the Pareto optimal frontier than that by the cooperative agents’ one) and obtains the same total utility gain as the rational and cooperative agents. And negotiation traces in lower right in Figure 5 (4b)(4c)(4d) represent the negotiation traces of the learning agent X. It shows that proposal sequences of learning agents tend to be shorter and the negotiation result is closer to the Pareto optimal frontier than those of the cooperative and rational agents. In our simulation results as in Figure 6, the learning agents do perform better than the rational agents and cooperative agents in terms of message cost and the Pareto-efficiency.

5. CONCLUSION

In our experiments, we set up the agents’ behaviors via their action space and model how to make concessions under message constraints. In our experiments, we found that if there were no message constraints of the negotiation, the agents would try to reach the final joint agreement at infinite cost of negotiating.
messages. Even the cooperative and learning agents would only perform a little better than the others. But with message constraints, there is motivation for agents to behave smarter! An agent without intelligent strategy may result in a failure negotiation by the time bound or causes an inefficient negotiation. We found that a rational agent would lead to a Pareto-efficient negotiation but will cost a lot of negotiation messages. The cooperative agents or learning agents will reach a better agreement at less message cost in negotiation. From the negotiation trace sequence, it suggested that agents with cooperative and learning abilities can speed up the negotiation and their agreement solution tend to be much closer to the Pareto optimal frontier, namely higher Pareto efficiency. It is important for agents to have some strategies to negotiate to each other, especially when the cost of negotiating time is critical.

This paper contributes in showing how to couple the learning to the bilateral agent negotiation in multi-issue negotiation. Of course the adoption of perceptron learning is just to show that a simple learning algorithm can augment the negotiation. It will inherit the weakness of the perceptron learning. So when dealing with more sophisticated user utility functions, it waits to be seen that whether more sophisticated learning algorithms would get more advantage in speed up the negotiation and at the same time guarantee the quality of the negotiation results.

Acknowledgment
This research is supported in part by MOE Program for Promoting Academic Excellence of Universities under the grant number 89-E-FA04-1-4 and NSC under the grant number NSC91-2213-E-007-036.

REFERENCES