

# An Efficient Automated Negotiation System Using Multi-attributes in the Online Environment

Sanghyun Park and Sung-Bong Yang

Dept. of Computer Science, Yonsei University  
134 Shinchon-Dong, Seodaemun-Gu, Seoul, 120-749, Korea  
{psh, yang}@cs.yonsei.ac.kr

**Abstract.** In this paper we propose an efficient negotiation agent system that guarantees the reciprocity of the attendants in a bilateral negotiation on the e-commerce. The proposed negotiation agent system exploits incremental learning based on artificial neural networks to generate counter-offers and is trained by the previous offers that have been rejected by the other party. During a negotiation, the software agents on behalf of the buyer and the seller negotiate each other by considering the multi-attributes of a product. The experimental results show that the proposed negotiation system achieves better agreements than other negotiation agent systems that can be operable under the realistic and practical environment. Furthermore, the proposed system carries out negotiations about twenty times faster than other negotiation systems on the average.

## 1 Introduction

Software agents have been playing an important role in e-commerce. Nowadays, agent mediated negotiation has received considerable attention in the field of automated trading [1,2]. However, they have not been developed sufficiently at the level of robust systems applicable to commercial web sites for e-commerce considering the importance of software agents that carry out negotiations on behalf of buyers or sellers. Such underdevelopment is due to the facts that the attributes, which may influence the buyer's behavior, cannot be defined precisely and that the success of a negotiation is dependent not only on the price but on certain abstract factors such as diverse inclination of buyers or sellers.

The Persuader system in the work of Sycara [3] utilizes the concepts of argumentation. In this system, mediation has been modeled by considering an iterative negotiation, multiple issues and multiple agents. However, a negotiation, as defined in the Persuader system, is a mutual selection of outcome and precludes any intervention by outside parties [4]. The persuasion mechanisms operate on the beliefs of agents with the aim of changing one or both parties' beliefs. Therefore, the Persuader system has no concern about the fact that it is not necessary for the agents to have similar beliefs at the end of the negotiation.

Kasbah [5] has proposed some negotiation strategies involved in negotiations for the real world applications. In Kasbah, however, a negotiation is carried out over an only single attribute such as price and the agents could not adopt any negotiation

strategy for themselves. The number of strategies that are delegated to the agents is limited to only three and even their selections are not autonomous.

Negotiation agent architecture for structured interactions in the real world environment has been proposed by Faratin et al. [4]. Several negotiation systems have been implemented with the architecture in the paper of Faratin et al. [4,6]. These negotiation systems are capable of dealing with multiple attributes in a bilateral negotiation and of reaching an agreement through interactions between the agents. Among the various strategies for negotiations, the strategies based on the personal information of the other party could guarantee high profits for both participants in a negotiation [6]. However, under the environment of the online and real-time negotiations with unspecific persons, it is unrealistic that an agreement has been reached under the assumption that a negotiation is carried out with the personal information of the other party. Furthermore, the negotiation systems operating without the information of the other party cannot guarantee the profits for reciprocity.

Other negotiation systems considering interactions between agents have been studied by Fatima et al. [1] and Faratin et al. [7]. In these studies, time constraint was considered as an additional attribute. For adaptive agent based negotiation, Oliver [8] showed that agents could learn strategies using a genetic algorithm-based learning technique and Oprea [9] proposed a negotiation system that used a feed forward artificial neural network as the learning ability of a negotiation model in the context of agent-based e-commerce. These studies have shown satisfying results on negotiation under long-term deadlines; however, their systems require longer time interval to obtain better deals.

In this paper, we propose a negotiation agent system based on the incremental learning method. The aim of the proposed system is to increase the efficiency of negotiation in terms of both the execution time and the profits of the participants for reciprocity. The proposed system guarantees to obtain agreements within reasonable time. Note that time is not an issue of a contract. The proposed negotiation agent system has been implemented using interactions among the agents as in the Faratin's systems [4,6].

For the experiment, the negotiation agent systems considering interactions between the buyer and the seller agents [4,6] have also been implemented and experimented with the various datasets to compare with the proposed negotiation system. The experimental results show that the average relative error of the proposed negotiation agent system with respect to the negotiation system under *ideal* assumptions that one party knows the personal negotiation information about the other party, is within only about 2% and that the proposed system achieves better agreements than other negotiation agent systems that can be operable under the realistic and practical environment with unspecific persons. Furthermore, the proposed system carries out negotiations about twenty times on average faster than other negotiation systems implemented in this paper.

The remainder of the paper is organized as follows. Section 2 defines a trading model in which a negotiation carries on. Section 3 describes various negotiation systems including the proposed negotiation agent system. Section 4 provides the experimental results that compare the proposed system with the previous negotiation agent systems. Finally, the conclusions are made in Section 5.

## 2 The Negotiation Agent Model

### 2.1 The Multi-attribute Utility Theory

In this paper, the multi-attribute utility theory (MAUT) [10,11] is applied to evaluate the profits of buyers and sellers considering multi-attributes of the merchandise. Therefore, each attribute of the merchandise has a weight indicating the relative preference to each of other attributes.

The utility function in MAUT can be expressed in terms of the weights of the attributes and the values of the evaluation function at *the offer values* of the attributes, where the offer value of an attribute is the value of the attribute in a contract. Therefore, a contract that consists of the offer values of all the attributes can be regarded as a “proposal” in the real negotiations. The value of the utility function can be denoted as the profit of either a buyer or a seller. The utility function  $V^k(X)$  can be expressed as follows.

$$V^k(X) = \sum_{i=1}^n \left( w_{j_i}^k \cdot v_{j_i}^k(x[j_i]) \right), \quad \sum_{1 \leq i \leq n} w_{j_i}^k = 1, \quad 0 \leq v_{j_i}^k \leq 1 \quad (1)$$

where  $k$  indicates either a buyer or a seller agent,  $J = \{j_1, j_2, \dots, j_n\}$  is the set of attributes  $j_i$  ( $1 \leq i \leq n$ ),  $X = \{x[j_1], x[j_2], \dots, x[j_n]\}$  is a contract composed of the values of all the attributes,  $x[j_i]$  is the *offer value* of attribute  $j_i$ ,  $w_{j_i}^k$  is the weight of attribute  $j_i$  for agent  $k$ , and finally  $v_{j_i}^k(x[j_i])$  is the evaluation function for  $x[j_i]$  for agent  $k$ .

### 2.2 The Evaluation Function

The evaluation function for an attribute can be expressed in terms of *the request value* and *the allowable value* of the attribute, where the request value of the attribute is the maximum value for the attribute that an agent wants to acquire from the other party in the negotiation and the allowable value of the attribute is the maximum value for the attribute to which an agent concedes during the negotiation. The evaluation function can be written as follows.

$$v_{j_i}^q(x[j_i]) = \frac{x[j_i] - \text{allowable value}}{\text{request value} - \text{allowable value}}, \quad 0 \leq v_{j_i}^q \leq 1 \quad (2)$$

If the offer value,  $x[j_i]$ , is out of the range from the allowable value to the request value, the value of the evaluation function is set to zero or one depending on  $v_{j_i}^q$ . That is, if  $v_{j_i}^q < 0$ , it is set to zero, and if  $v_{j_i}^q > 1$ , it is set to one. Let the set of values  $\{v_{j_1}^q(x[j_1]), v_{j_2}^q(x[j_2]), \dots, v_{j_n}^q(x[j_n])\}$  be a *normalized value set*. The following equation obtained from Equation (2) computes the *offer value*  $x[j_i]$  of attribute  $j_i$ .

$$x[j_i] = (1 - v_{j_i}^q(x[j_i])) \times (\text{allowable value}) + v_{j_i}^q(x[j_i]) \times (\text{request value}) \tag{3}$$

Therefore, from Equation (3) we can obtain an offer from its corresponding normalized value set.

2.3 The Trading Model

In the trading model adopted in this paper, we deal only with bilateral negotiations, because a multilateral negotiation is an extension of a bilateral one. We also concern mainly with finding a faster and better agreement in a bilateral negotiation. By the better agreement we mean the agreement for reciprocity; the profit is not partial to one over the other participant and rather the agreement guarantees as much profits as possible for both participants.

Although various commodities can be traded in a negotiation agent system, in this paper, we have chosen used cars for trading, because used cars could reflect various propensities to consume. In the trading model, the buyer wants to purchase a used car from the seller who deals various types of used cars. Although there may be several cars with the same model, a negotiation can still be carried on, because each car has different values on attributes in general.

The attributes of a used car for a negotiation are set to its price, the mileage and the warranty of the car. In the trading model, we assume that the buyer wants to purchase a specific car that the seller has presented. The seller has his or her own values on the attributes such as the request values, the allowable values and the weights. The buyer has also such negotiation information. The agent representing a party has all the negotiation information that the party has. However, the agent for one party does not know any negotiation information about the other party except the information regarding a contract.

The input datasets for bilateral negotiations in this paper are prepared so that each attribute could reflect all possible variations. Tables 1 and 2 show sample datasets for the buyer and the seller, respectively.

Table 1. A sample dataset for the buyer

Attributes	Request values	Allowable values	Weights
Price (Dollars)	2,000	5,500	0.5
Mileage (Miles)	5,000	100,000	0.3
Warranty (Months)	24	12	0.2

Table 2. A sample dataset for the seller

Attributes	Request values	Allowable values	Weights
Price (Dollars)	6,000	3,000	0.5
Mileage (Miles)	40,000	10,000	0.2
Warranty (Months)	6	18	0.3

Table 1 shows that the buyer wants to purchase a car with \$2,000. The buyer may make a concession to the seller by \$5,500 if the seller proposes a contract that is more

profitable to the buyer in the mileage or the warranty than the previous contract. Furthermore, the buyer wants a car with 5,000 miles and may also make a concession to the seller by 100,000 miles if better offers in other attributes are proposed. The warranty is requested for 24 months from the buyer and also allowed up to 12 months according to the values of the other attributes. The weights of the attributes shown in Table 1 indicate that the buyer conceives the price as the most important factor in purchasing a car, the mileage as the next and the warranty as the least significant factor. Table 2 shows the negotiation information for the seller and can be explained similarly.

## 2.4 The Negotiation Process

A negotiation with multi-attributes involves various contracts with the same value of utility, so called the *iso-value curve* [12]. Each agent proposes a contract or provides counter-offers that lie on the iso-value curve with respect to the combinations of the values of attributes. The generation of a contract has a significant effect on the negotiation efficiency such as the profits of the attendants in the negotiation and the execution time of the negotiation. In the negotiation systems in the work of Faratin et al. [6], a contract, which seems to be the closest to the opponent's last offer among the contracts on the iso-value curve, is proposed to the other party. In this case, the negotiation efficiency can be very high when the agent of one party knows the negotiation information of the other such as the opponent's weight for each attribute. However, since the negotiation information of one party cannot be exposed to the other party in a practical commercial transaction, it is not easy to find a contract that is the closest to the opponent's last offer. Fig.1 shows an overview of the process of a negotiation for the negotiation systems implemented in this paper [4].

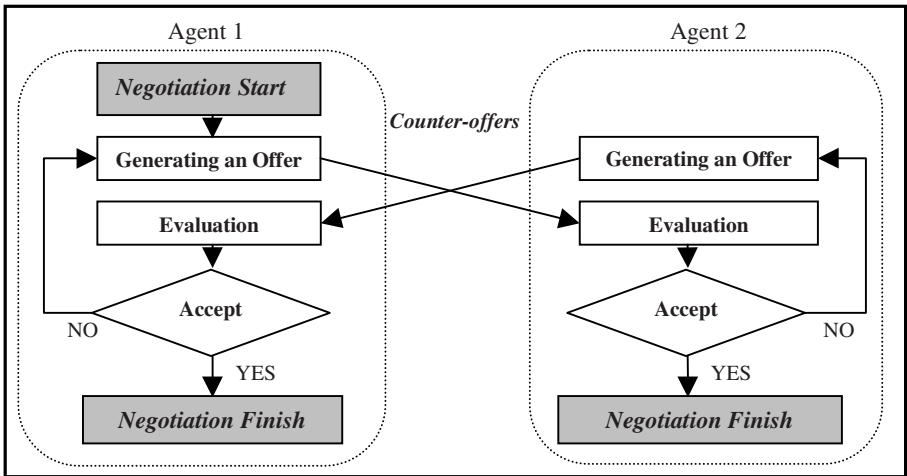


Fig. 1. The process of a negotiation in the agent systems

One agent as a participant in a bilateral negotiation offers a contract (an offer) to the other agent, and then the agent who receives the offer may accept it or generates a counter-offer as another contract. The utility functions for the profits of both participants are evaluated as described in the previous section.

### 3 The Negotiation Agent Systems

The trade-off mechanism is a kind of deliberative mechanism [4], and has been suggested to implement the transactions of the two agents participated in a bilateral negotiation. The trade-off mechanism enables an agreement to be reached through the transactions of agents and is capable of reflecting a practical environment for negotiations. However, since the previous negotiation systems based on the trade-off mechanism have never considered the execution time for a negotiation, they cannot handle negotiations well in the online environment and may not perform multi-lateral negotiations efficiently in the online environment as well.

#### 3.1 The Trade-Off Mechanism

A *trade-off* can be defined informally as the mechanism in which one party lowers its values on some attributes and demands more on other attributes at the same time. Therefore, a trade-off is a search for a contract that is equally valuable to the previously offered contract, but that may benefit the other party.

Let  $X$  be an offer from agent  $q$  to the other party, and  $Y$  be a subsequent offer from the other party to agent  $q$ . A trade-off for agent  $q$  with respect to  $Y$  can be defined as follows

$$tradeoff_q(X, Y) = Z \quad (4)$$

where  $q$  is either the buyer agent or the seller agent,  $Z$  is the contract that satisfies  $V^k(Z) = V^k(X)$  and is assumed to be the offer that is the most similar to  $Y$ .

When the similarity between two offers is evaluated with the information of the other party such as the weights under the assumption that an agent is acquainted with the information about the other party, an agreement can be found under the guarantee of high profits of the participants in a negotiation [6]. However, the similarity must be evaluated under a practical condition that one participant in a negotiation is not aware of the other party's information. Therefore, the similarity based on the Euclidean distance is used for evaluation in order to implement the negotiation systems considering interactions between the buyer and the seller agents [4, 6, 12]. In the process of a negotiation, a value of the utility function is reduced by a predetermined amount, if a deadlock during the negotiation occurs due to keeping up the values of the utility function [6].

## 3.2 The Negotiation System Based on Incremental Learning

### 3.2.1 An Artificial Neural Network

In this paper, the effects of the learning agents are investigated with focusing on the reciprocity and on the execution time in a negotiation system. An online learning method such as the incremental learning method, therefore, is more appropriate for a negotiation agent system in the e-commerce than an offline learning method such as the batch learning, considering the environment of the online and real-time negotiations with unspecific persons [13]. The negotiation agent system proposed in this paper is designed to conduct a negotiation by using the incremental learning method in generating a contract, in contrast to the negotiation systems with trade-off mechanisms based on the similarity [4, 6]. Fig. 2 illustrates the structure of an artificial neural network for a trading model proposed in this paper. This model can be applicable to a general trading model by creating the same number of nodes in each input and output layer as that of attributes. The input and the output layers have three nodes each, since we consider three attributes for a product in the proposed system. Each node corresponds to an attribute of a product. The sigmoidal function is established as the activation functions of both the hidden layer and the output layer [14].

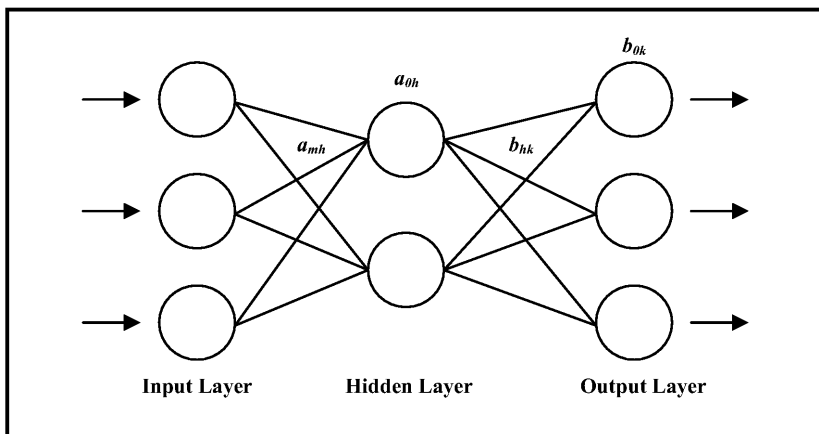


Fig. 2. The structure of an artificial neural network for the proposed negotiation system

### 3.2.2 The Learning Process

The negotiation agent system proposed in this paper follows the negotiation protocol illustrated in Fig. 1. Incremental learning process is employed in order to generate counter-offers. The learning process consists of two processes; *the initial learning process* and *the run-time learning process*. Fig. 3 shows the proposed negotiation procedures to which these two learning processes for *generating an offer* are added. The initial learning process initializes an artificial neural network through repetitive learning. The run-time learning process simultaneously generates a contract (a counter-offer) and carries out learning with the artificial neural network based on the initial learning process.

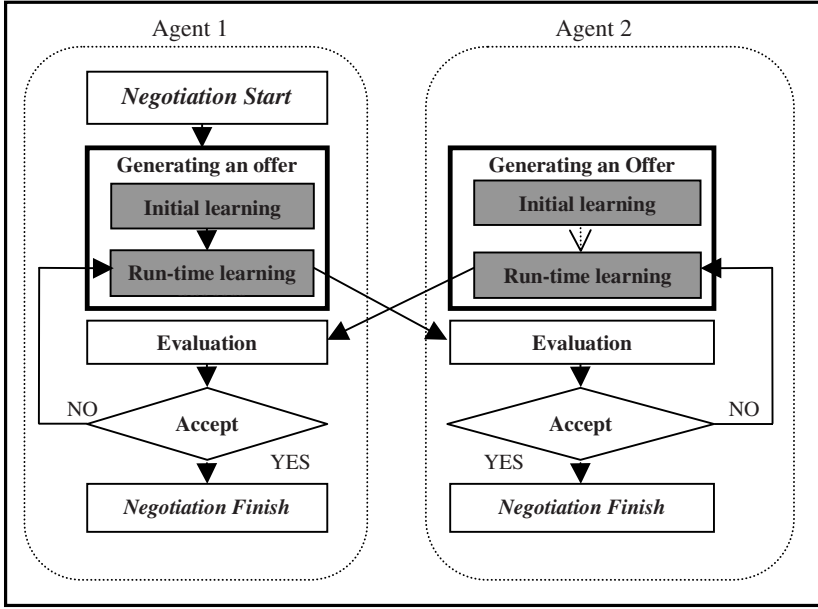


Fig. 3. Two learning processes for generating an offer in the proposed negotiation agent system

### 3.2.3 The Initial Learning Process

In the initial learning process, the input layer node  $m$  has the value of  $v_{j_m}^q$  ( $m = 1, 2, \dots, n$ ) which is the evaluation value of the initial offer from the other party. The target of the output layer node  $k$  is  $v_{j_k}^q$  ( $k = 1, 2, \dots, n$ ) which is the evaluation value of an offer that is very close to the first contract proposed from the other party. The offer is chosen out of offers on the *iso value curve*. Therefore, each node in the input and output layers has a value between zero and one. The criterion of the similarity in the initial learning process is based on the distance between a pair of contracts. The run-time learning process, however, is not dependent on the similarity.

In Fig. 2 the error terms  $\delta_k$  and  $\delta_h$  in the output and the hidden layers can be expressed as follows, respectively.

- 1) The error term  $\delta_k$  of the output layer node  $k$  can be written as

$$\delta_k = o_k(1 - o_k)(t_k - o_k) \quad (k = 1, 2, \dots, n) \quad (5)$$

- 2) The error term  $\delta_h$  of the hidden layer node  $h$  can be written as

$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs}} b_{hk} \delta_k \quad (h = 1, 2, \dots, n-1 \text{ and } k = 1, 2, \dots, n) \quad (6)$$

where  $o_k$  and  $t_k$  are the output and target values of the output layer node  $k$ , respectively,  $o_h$  is the output values of hidden layer node  $h$ , and  $b_{hk}$  is the weight from the hidden layer node  $h$  to the output layer node  $k$ .



The weight  $w_{xy}$  from node  $x$  to node  $y$  can be updated by  $w_{xy} \leftarrow w_{xy} + \eta \delta_y p_{xy}$  ( $\eta$  = the learning rate,  $p_{xy}$  = the input from node  $x$  to node  $y$ ). Therefore, the weights of the artificial neural network for the proposed system are updated as follows.

$$\begin{cases} a_{mh} \leftarrow a_{mh} + \eta \delta_h p_{mh} \\ b_{hk} \leftarrow b_{hk} + \eta \delta_k p_{hk} \end{cases} \quad (7)$$

where  $a_{mh}$  is the weight from the input layer node  $m$  to the hidden layer node  $h$ .

In the beginning, all the weights of the artificial neural network in the initial learning process are set to zero. Note that the weights of the artificial neural network after the initial learning process will become non-zeroes because the weights are updated repeatedly. The initial learning process establishes the *initial weights* of the artificial neural network for the run-time learning process through repetitive learning.

### 3.2.4 The Run-Time Learning Process

The run-time learning process generates a contract and carries out a learning process on the artificial neural network after the initial learning process. In this process the input layer node  $m$  has the value of  $v_{j_m}^q$  ( $m=1, 2, \dots, n$ ) which is the evaluation value of the previous contract that was rejected by the other party. The output value of the artificial neural network is converted to the *offer value* for each attribute  $j_k$  ( $k=1, 2, \dots, n$ ) by Equation (3), and these *offer values* for the entire attributes constitute a counter-offer for the other party.

For example, when agent 1 rejects a contract {\\$200, 20,000km, 18mos.} offered by agent 2, the contract are converted to the normalized value set {0.766, 0.684, 0.612}, which was computed with Equation (2). This normalized value set is then fed into the neural network of agent 2. Assume that the output of the neural network is {0.728, 0.691, 0.545}. Then this output is now converted to {\\$210, 19,000km, 15mos.} with Equation (3) as a counter-offer, and the weights of the network are updated with the backpropagation of the neural network by Equation (7).

The utility of the target is preserved as the current utility of the iso-value curve. If an agreement cannot be reached until a certain number of rounds in a negotiation, the utility is decreased with a predetermined rate. If the agreement cannot be reached within 100 rounds in a negotiation, the utility is decreased with 0.01 and carries out the negotiation. In the run-time learning process, the number of iterations for learning is increased monotonically as the number of rounds in the negotiation is increased. In the experiment, the learning rate is set to 0.01 during the negotiation process.

The proposed negotiation system utilizes the incremental learning method to generate counter-offers and is trained by the previous offer that has been rejected by the other party. Therefore, under the realistic and practical environment of negotiations, the incremental learning agent system is able to perform negotiations more rapidly and to guarantee more profits of the participants compared with the negotiation systems based on the similarity and the trade-off mechanism [6].

## 4 The Experimental Results

### 4.1 The Input Datasets

The proposed negotiation system is compared with the negotiation systems using the trade-off mechanism in [6]. The seller’s negotiation data are generated randomly within the range from Min to Max values in Table 3. The weights of the price and the mileage for a seller are chosen randomly within the ranges of the preferences in Table 3, respectively. The weight of the warranty is determined with the equation in the table entry after the weights of the price and the mileage have been chosen. Two hundred sellers are created for the experiments.

The buyer’s negotiation data are generated according to Tables 4 and 5. In these tables, Pmax, Mmax, and Wmin are the request values of the price, the mileage, and the warranty for a seller created in Table 3, respectively. Similarly, Pmin, Mmin, and Wmax are the allowable values of the price, the mileage, and warranty for a seller created from Table 3, respectively. In Tables 4 and 5,  $\alpha$  is introduced to indicate how much the negotiation range of an attribute for one party intersects with that of the same attribute of the other party. By a negotiation range we mean the range from a request value to an allowable value of an attribute for a party.  $\alpha$  is chosen randomly within the range of  $\alpha$  in Tables 4 or 5 to create the buyer’s request and allowable values for each attribute. We create one hundred buyers from Table 4 and another one hundred buyers are created with Table 5. A buyer created from Table 4 is likely to have negotiation ranges that are less similar to those of a seller, while the negotiation ranges of a buyer created from Table 5 are more similar to those of a seller. We call the buyer data created from Table 4 the Negotiation Dataset I and the buyers data from Table 5 the Negotiation Dataset II.

**Table 3.** The constructions of seller’s negotiation data

Attribute	Request values [Min, Max]	Allowable values [Min, Max]	The range of preference
Price (Dollars)	[5000, 10000]	[1500, 3000]	[0.4, 0.6]
Mileage (Miles)	[30000, 100000]	[10000, 20000]	[0.2, 0.4]
Warranty (Months)	[6, 12]	[14, 36]	$1 - \{\text{the weight of price} + \text{the weight of mileage}\}$

**Table 4.** The Negotiation Dataset I

Attribute	Request value	Allowable value	The range of $\alpha$
Price (Dollars)	$P_{min} + (P_{max} - P_{min}) \cdot \alpha$	$P_{max} + (P_{max} - P_{min}) \cdot \alpha$	[-0.4, 0.1]
Mileage (Miles)	$M_{min} + (M_{max} - M_{min}) \cdot \alpha$	$M_{max} + (M_{max} - M_{min}) \cdot \alpha$	[-0.4, 0.1]
Warranty (Months)	$W_{max} + (W_{max} - W_{min}) \cdot \alpha$	$W_{min} + (W_{max} - W_{min}) \cdot \alpha$	[-0.1, 0.4]

**Table 5.** The Negotiation Dataset II

Attribute	Request value	Allowable value	The range of $\alpha$
Price (Dollars)	$P_{min}+(P_{max}-P_{min})\cdot\alpha$	$P_{max}+(P_{max}-P_{min})\cdot\alpha$	$[-0.25, 0.1]$
Mileage (Miles)	$M_{min}+(M_{max}-M_{min})\cdot\alpha$	$M_{max}+(M_{max}-M_{min})\cdot\alpha$	$[-0.25, 0.1]$
Warranty (Months)	$W_{max}+(W_{max}-W_{min})\cdot\alpha$	$W_{min}+(W_{max}-W_{min})\cdot\alpha$	$[-0.1, 0.25]$

## 4.2 The Experiment Environment

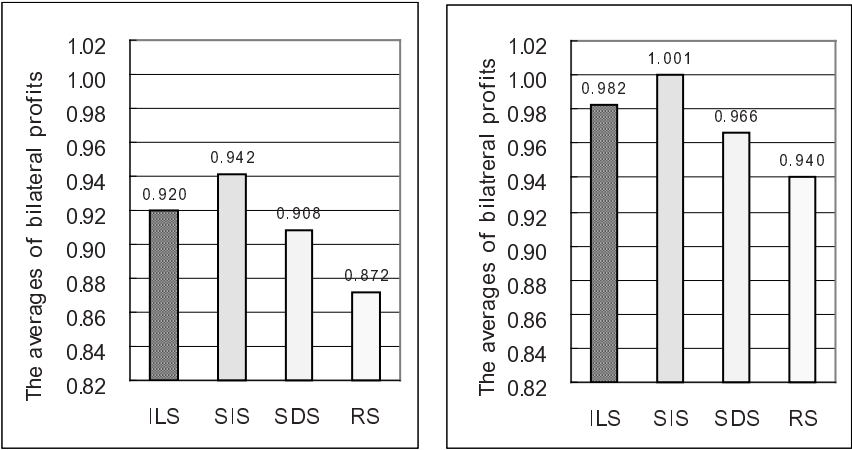
The Negotiation Datasets I and II include various inclinations of buyers and sellers. The experiments are carried out on these datasets to compare the performance of the proposed negotiation agent system with those of other negotiation systems based on the trade-off mechanism. At each agreement for all the negotiation systems, the difference between the profits of the buyer and the seller in a bilateral negotiation has never exceeded 0.03. Therefore, the performance of a negotiation agent system can be determined with the summation of the profits of both the buyer and the seller.

In order to compare the incremental learning system proposed in this paper (Incremental Learning System, *ILS*), three negotiation agent systems proposed by Faratin et al. [6] have also been implemented. The first system is the negotiation agent system using the similarity based on the information of the other party (Similarity-Information System, *SIS*). The second one is the negotiation agent system using the similarity based on the distance between contracts, without using the information of the other party (Similarity-Distance System, *SDS*). The last system is the negotiation agent system generating offers randomly, without considering the other party (Random System, *RS*). Note that, among these negotiation agent systems, *SIS* is more advantageous in utility than other systems, because *SIS* operates under the assumption that the agent of one party knows the personal information of the other party such as the weights of attributes.

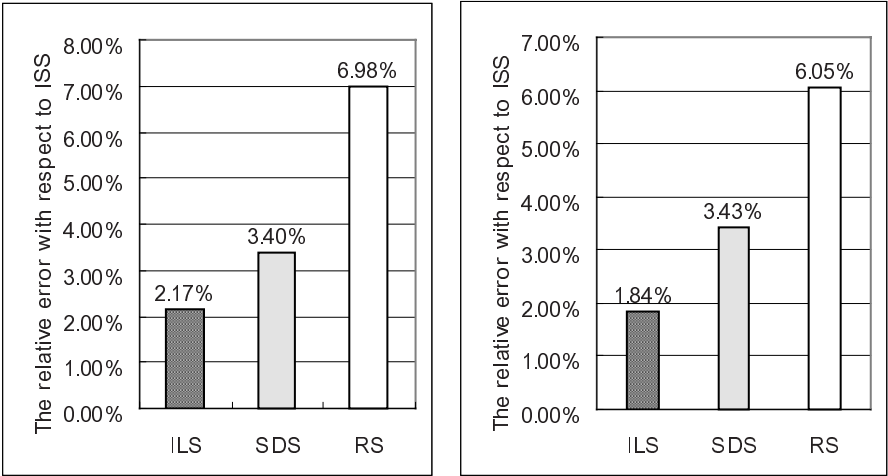
## 4.3 The Experimental Results

Fig. 4 shows the average bilateral profits for the negotiation systems on the Negotiation Datasets I and II. The average bilateral profits indicate the summation of the buyer's and the seller's profits. As shown in this figure, *SIS* obviously achieves the best agreements for reciprocity, because *SIS* operates under the assumption that the personal information of one party such as the weights of the attributes is available to the other party in a negotiation. The figure also shows that *ILS* produces better agreements for reciprocity than other negotiation agent systems in the practical environment.

Fig. 5 shows the average relative errors of *ILS*, *SDS* and *RS* with respect to *SIS* on the Negotiation Datasets I and II. For these two datasets, the average relative errors of *ILS* are 2.17% and 1.84%, which are lower than those of *SDS*, respectively. Observe that *ILS* provides better agreements on the Negotiation Dataset II than on the Negotiation Dataset I, since the negotiation range of a buyer created from the Negotia-



**Fig. 4.** The average bilateral profits for the negotiation systems on the Negotiation Dataset I (left) and Dataset II (right)

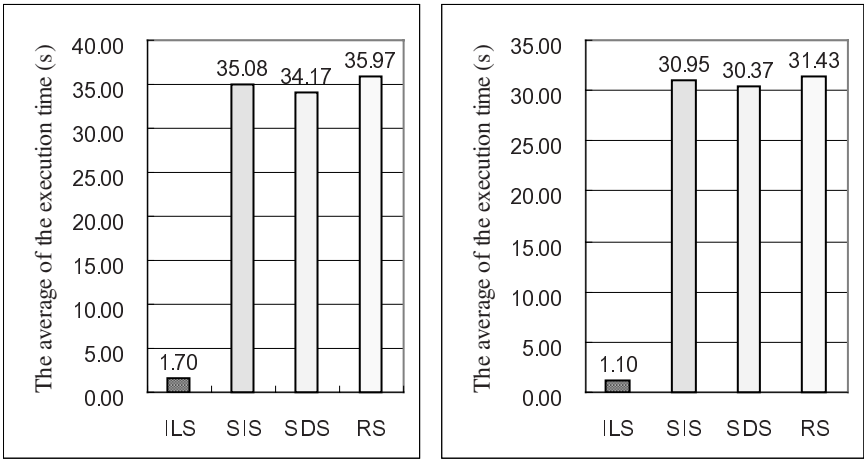


**Fig. 5.** The average relative errors of ILS, SDS and RS to SIS on the Negotiation Dataset I (left) and Dataset II (right)

tion Dataset II is more similar to those of a seller. Considering both datasets, ILS is superior to SDS in bilateral negotiations. Fig.4 also shows that RS has higher relative errors than other negotiation systems for both datasets.

Fig. 6 shows the average execution times for the negotiation agent systems on the datasets. For the Negotiation Dataset I, the average execution time of ILS is 1.7 seconds; ILS is at least twenty times faster than others. For the Negotiation Dataset II, the average execution time of ILS is 1.1 seconds; ILS is at least twenty seven times faster than other systems. The execution times for the Negotiation Dataset I are longer

than those for the Negotiation Dataset II throughout all the negotiation agent systems, because the negotiation range of a buyer created from the Negotiation Dataset II is more similar to that of a seller. From the viewpoint of the online and real-time negotiation with unspecific persons for the e-commerce, ILS performs more efficient negotiations than other negotiation systems considering both the execution time and negotiation results.



**Fig. 6.** The average execution times of the negotiation agent systems on the Negotiation Datasets I (left) and Dataset II (right)

## 5 Conclusions

In this paper, we have proposed an efficient negotiation agent system that guarantees the reciprocity of the attendants in a bilateral negotiation. The proposed negotiation system exploits incremental learning in generating counter-offers and is trained by the previous offers that have been rejected by the other party.

An agreement for reciprocity in this paper is reached under the assumption that the profit is not partial to any single participant in a bilateral negotiation and that the contract of the agreement guarantees as much profits as possible to both participants. Therefore, since the difference between the profits of the buyer and the seller is maintained under a predetermined threshold, a negotiation system can be evaluated by the sum of the profits of both the buyer and the seller.

We have implemented three negotiation agent systems based on the work from Faratin et al. (2000) to compare the performance of the proposed negotiation agent system. The experimental results show that under the realistic and practical environment of negotiations the proposed system is more efficient in negotiations than others in terms of both the profits for reciprocity and the execution time. The improvement in the incremental learning method and the expansion to multilateral negotiations are left as future researches.

## References

1. Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, Vol. 10, Issue 2. (2002) 199-215
2. Fatima, S. S., Wooldridge, M., Jennings, N. R.: Multi-Issue Negotiation Under Time Constraints. *Proc. of 1<sup>st</sup> Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, July 15-19, Bologna, Italy (2002) 143-150
3. Sycara, K.: Multi-Agent Compromise via Negotiation. *Distributed Artificial Intelligence*, Vol. 2. (1989) 119-139
4. Faratin, P., Sierra, C., Jennings, N. R., Buckle, P.: Designing Responsive and Deliberative Automated Negotiators. *Proc. AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, Orlando, USA (1999) 12-18
5. Chavez, A., Maes, P.: Kasbah: An Agent Marketplace for Buying and Selling Goods. *Proc. 1<sup>st</sup> Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK (1996)
6. Faratin, P., Sierra, C., Jennings, N. R.: Using Similarity Criteria to Make Negotiation Trade-offs. *Proc. 4th Int. Conf. on Multi-Agent Systems*, Boston, USA (2000) 119-126
7. Faratin, P., Sierra, C., Jennings, N. R.: Negotiation Decision Functions for autonomous agents. *Int. J. of Robotics and Autonomous Systems*, Vol. 24, Issue 3-4. (1998) 159-182
8. Oliver, J.: A Machine Learning Approach to Automated Negotiation and Prospects for Electronic Commerce. *Journal of Management Information Systems*, Winter 1996/1997
9. Oprea, M.: An Adaptive Negotiation Model for Agent-Based Electronic Commerce. *Studies in Informatics and Control*, Vol. 11, No. 3 (2002)
10. Barbuceanu, M., Lo, W.: A Multi-Attribute Utility Theoretic Negotiation Architecture for Electronic Commerce. *Proc. 4th Int. Conf. on Autonomous Agents*, Barcelona, Catalonia, Spain (2000)
11. Keeney, R. L., Raiffa, H.: *Decisions with Multiple Objectives*. Cambridge University Press (1993)
12. Raiffa, H.: *The Art and Science of Negotiation*. Harvard University Press (1982)
13. Soo, V., Hung, C.: On-Line Incremental Learning in Bilateral Multi-Issue Negotiation. *Proc. 1<sup>st</sup> Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy (2002)
14. Smith, M.: *Neural Networks for Statistical Modeling*. International Thomson Computer Press (1996)