



Journal of Theoretical and Applied Electronic
Commerce Research

E-ISSN: 0718-1876

ncerpa@utalca.cl

Universidad de Talca
Chile

Jazayeriy, Hamid; Azmi-Murad, Masrah; Sulaiman, Nasir; Izura Udizir, Nur
The Learning of an Opponent's Approximate Preferences in Bilateral Automated Negotiation
Journal of Theoretical and Applied Electronic Commerce Research, vol. 6, núm. 3, diciembre, 2011,
pp. 65-84
Universidad de Talca
Curicó, Chile

Available in: <http://www.redalyc.org/articulo.oa?id=96520915006>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in redalyc.org

redalyc.org

Scientific Information System
Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal
Non-profit academic project, developed under the open access initiative

The Learning of an Opponent's Approximate Preferences in Bilateral Automated Negotiation

Hamid Jazayeri¹, Masrah Azmi-Murad², Nasir Sulaiman³ and Nur Izura Udizir⁴

¹ Noshirvani University of Technology, Babol, Iran, jhamid@nit.ac.ir
² University Putra Malaysia, Faculty of Computer Science and IT, Serdang, Malaysia
³ masrah@fsktm.upm.edu.my, ⁴ nasir@fsktm.upm.edu.my, ⁵ izura@fsktm.upm.edu.my

Received 21 June 2010; received in revised form 13 April 2011; accepted 26 April 2011

Abstract

Autonomous agents can negotiate on behalf of buyers and sellers to make a contract in the e-marketplace. In bilateral negotiation, they need to find a joint agreement by satisfying each other. That is, an agent should learn its opponent's preferences. However, the agent has limited time to find an agreement while trying to protect its payoffs by keeping its preferences private. In doing so, generating offers with incomplete information about the opponent's preferences is a complex process and, therefore, learning these preferences in a short time can assist the agent to generate proper offers. In this paper, we have developed an incremental on-line learning approach by using a hybrid soft-computing technique to learn the opponent's preferences. In our learning approach, first, the size of possible preferences is reduced by encoding the uncertain preferences into a series of fuzzy membership functions. Then, a simplified genetic algorithm is used to search the best fuzzy preferences that articulate the opponent's intention. Experimental results showed that our learning approach can estimate the opponent's preferences effectively. Moreover, results indicate that agents which use the proposed learning approach not only have more chances to reach agreements but also will be able to find agreements with greater joint utility.

Keywords: Bilateral negotiation, Learning preferences, Uncertain information, Genetic algorithm, E-marketplace

1 Introduction

Automated negotiation is a multi-disciplinary area of research consisting of multi-agent systems (MAS) [9], [22], [24], game theory [3], optimization [1], [26] e-commerce [11], [19], [27] and decision support systems [14], [15]. Although researchers from distinct fields have different points of views on automated negotiation and its applications, it has been categorized as a search problem. Accordingly, *it can be viewed as a search problem where autonomous agents try to optimize their own utility by finding the best possible joint offer from the search space.*

Autonomous agents can play important roles in e-marketplaces [2], [19]. They can behave as a seller/buyer, supplier/consumer, client/server or even as a mediator. These agents can negotiate with each other to increase their owner's utility. Unfortunately, current e-marketplaces hardly support automated negotiation. The main reason is that negotiation is a complex process.

Automated negotiation is a basic element in multi-agent systems (MAS) which enables autonomous agents to track their goal by providing communication among them. Autonomous agents (that are able to control their own behavior in the furtherance of their own goals [30]) have their own intentions to interact with other agents and usually these intentions are conflicting. Automated negotiation is a process of resolving conflicts to reach a mutual agreement by proposing a series of offers and counter-offers. To this end, an agent should be able to generate an offer that satisfies other agents and motivates them to continue the negotiation. However, generating offers is a complex process because, firstly, agents do not have much information about their opponent and, secondly, finding a proper offer from the huge search space is time consuming and needs some sort of intelligence. Moreover, solution for automated negotiation is not unique and agents may reach to different agreements according to their generated offers. It means that the quality of the negotiation outcome is highly related to the agent's information about its opponent that can be learned during the negotiation process by using artificial intelligence techniques as well as offer-generating strategies.

In bilateral automated negotiation, learning an opponent's preferences can be considered as a challenging problem. Firstly, agents usually do not pass a training phase to participate in the negotiation. Thus, they should learn online especially when they encounter a new scenario and environment. Secondly, negotiators try to protect self-payoffs by keeping their preferences private. That is, the opponent's preferences are usually hidden, and no information is provided at the beginning and, thereby, learning should be conducted during the negotiation process incrementally. In fact, each round of the negotiation will provide a single item for learning. Thirdly, there is a time constraint to complete the negotiation process which means that the amount of learning data is very limited. Last but not least, learning in automated negotiation is usually unsupervised because the exact opponent's utility is uncertain when the agent receives an offer. This uncertainty makes it very hard to explore the opponent's preferences.

Although extensive academic research has explored the characteristics and dynamics of automated negotiation from different perspectives, learning an opponent's preferences with uncertain information needs more attention. Yet, to date, most presented learning methods in multi-issue negotiation require prior information about the opponent's preferences, which is not available necessarily. For example, some initial information about the probability distribution of the negotiation likely outcome is needed in Bayesian learning negotiation [4], [10], [33]. Similarly, kernel density estimation (KDE) method involves an offline process of the previous negotiation encounters [7]. Providing such information is hardly possible because of the privacy of opponents' preference and negotiation circumstances.

The following assumptions are used in this paper. Firstly, we assume that agents have incomplete information about their opponents. Secondly, similar to the real world negotiation, agents have limited time to find an agreement which makes the learning problem more challenging. These assumptions imply that agents should concede and make trade-offs among negotiation issues simultaneously. Finally, agents are computationally bounded meaning that they need time (and resources) to find a solution.

In this respect, learning an opponent's preferences by using soft computing techniques with incomplete information in multi-issue bilateral negotiation is the main goal of this study. To this end, simplified concepts of genetic algorithm, fuzzy membership functions and constraint satisfaction method are mixed to construct a hybrid learning approach.

The paper is organized as follows. Section 2 reviews the related works in the area of learning on opponent's preferences in automated negotiation. Section 3 describes the negotiation model used in this study by detailing the negotiation protocol and some basic concepts. Section 4 describes our proposed learning method. In section 5, we empirically evaluate the proposed method in a range of negotiation settings and scenarios. Finally, Section 6 outlines the conclusions and our plans for future study.

2 Related Work

During the past few years, interest in multi-issue automated negotiation has been growing. There has been a considerable body of work aiming to improve the negotiation outcome by proposing an effective strategy or by learning opponents' preferences. In particular, it has been shown that applying a learning method in automated negotiation has a great impact on the negotiation outcome [32]. Learning in automated negotiation not only helps agents to explore opponents' preferences [4], [7], [24], [33], it also empowers agents to find the optimal interaction strategies [20] and the opponents' future moves [5].

There have been several attempts to learn the different aspects of an opponent's preferences, such as learning an opponent's reservation point [33], learning issues order or priority (relative importance of issues) [7], [24], and estimating of an opponent's importance degrees over issues [7], [13]. Our learning approach also is an attempt to estimate an opponent's importance degrees in a form of fuzzy values by applying a set of constraints and a genetic algorithm.

Although Bayesian learning is one of the most popular methods to learn an opponent's behavior in the field of automated negotiation [4], [10], [33], agents require a priori information about the probability distribution of the negotiation likely outcome. Given the previous negotiation encounters and historical information can help agents to provide some prior probabilities about the opponents' behavior. This can be also considered as a drawback for Bayesian learning because of the privacy of information needed to compute these probabilities.

Zeng and Sycara [33] used Bayesian learning to reveal an opponent's reservation point in a single issue (price) negotiation. They have shown that Bayesian learning can reveal the opponent preferences and consequently, improve the negotiation outcome. However, our approach can explore an opponent's preferences during the multi-issue negotiation process. Buffett and Spencer [4] have used a Bayesian classifier to determine a class in which the opponents' preference relation likely fits during the negotiation process. The success of their method is highly related to the initial set of classes that has been determined by a utilized k-mean method. They assumed that agents use a concession-based strategy to reduce their offers' utility and consequently find an agreement. Similarly, Hindriks and Tykhonov [10] applied the concession assumption in their Bayesian learning and attempted to explore an opponent's preferences. In our approach, similarly, the concession assumption is applied to form some constraints that show an opponent's utility-boundaries during the negotiation process. However, in contrast with Bayesian learning, our approach doesn't need any prior information about an opponent that makes it more effective and practical.

Kernel density estimation (KDE) is used by Coehoorn and Jennings [7] to learn issues' priorities of an opponent. This method needs an offline process of the previous negotiation encounters to estimate an initial probability density function over the opponent's importance degrees for the negotiation issues. Then, new information can be augmented by online learning from the ongoing negotiation. They found that having a small error in estimation of an opponent's weights will not significantly degrade the performance of the trade-off algorithm [9]. However, in our approach, it is assumed that an agent has no information about its opponent's previous encounters. Consequently, the agent only can explore its opponent's preferences by online learning. We consider the opponent's issue importance degrees (weights) as fuzzy values and then try to learn preferences that have the minimum error based on some constraints. Similar to [7], our work finds an estimation of the opponent's preferences that can be used in trade-off.

Moreover, learning the order of issues' importance degrees is studied by Ros and Sierra [24]. They attempted to learn the order of issues' importance degrees based on a simple statistical analysis of the received offers. Issues with fewer changes considered as more important than those with more changes during the negotiation process. Ros and Sierra used the order of importance weights to improve the trade-off algorithm [9]. Similarly, we use an online learning during the negotiation; however, our work differs in that it is not limited to learn the order of weights. In our approach, estimating issues' importance degree empowers agents to generate offers more effectively.

Constraint satisfaction problem has been extensively studied to model the negotiation process [16]-[18], [29], [31]. Particularly, Luo et al. [18] developed a fuzzy constraint based model for bilateral multi-issue negotiation. They used prioritized fuzzy constraints to find trade-offs among the possible values of the negotiation issues which ensure that agents reach a fair agreement (Pareto-optimal). In our work, constraints are used to bound an opponent utility, and then, to explore its preferences by using the genetic algorithm.

In automated negotiation, agents usually have incomplete information about their opponents and, therefore, their perception about opponents is uncertain. This imperfect information motivates researchers to apply fuzzy techniques in automated negotiation. In doing so, different aspects of uncertainty and fuzzy techniques are widely addressed in the automated negotiation. For example, Faratin, Sierra and Jennings [9] presented a trade-off strategy, for multi-issue negotiation that enables agents to make trade-offs among negotiation issues (decision variables) by using fuzzy similarity approach. This fuzzy similarity is based on the opponent's importance degrees. They showed that quality of negotiation will decline if an agent has incomplete information about its opponent. Our work is a

complementary of their study, as we use a learning approach to explore an opponent's preferences and then apply the trade-off strategy to increase the negotiation success rate.

In other respects, however, a fuzzy scoring function used by Teuteberg [28] to evaluate the utility of each individual issue in multilateral negotiation in an environment of limited negotiation time. They showed that early concession in the negotiation process by both seller and buyer sides can increase the negotiation success rate. Though, our work differs from this as we use fuzzy preferences to learn negotiation issues' weight in bilateral negotiation by forecasting the opponent's utility value.

A prototype in autonomous multi-issue negotiation was presented by [16] where negotiation was considered as a form of a distributed decision making problem in the presence of incomplete information and uncertain constraints that could be modeled as a distributed fuzzy constraint satisfaction problem. However, unlike our work, they didn't use an explicit learning method in their approach. Moreover, their approach needs a search process guided by ordering and pruning the search space to find a mutual agreed offer that makes it complex and time consuming. An extended approach presented in [23] by applying a fuzzy Markov decision process to obtain an adaptive strategy in the single issue negotiation. In contrast, we try to explore an opponent's preferences in multi-issue negotiation.

Fuzzy preferences can be used to generate offers. Cheng et al. [6] used a fuzzy rule based on the concept of trade-offs over issues by assuming that perfect information about an opponent's issues' weights is available. Although agents do not need to learn their opponent's importance degrees, the strategy used in their work for offer generation is effective and fast. The main drawback for their strategy is the randomness of offer generation. However, our work differs from this as we present a learning method to explore a series of fuzzy preferences that can be used in fuzzy rules for offer-generation.

3 Negotiation Model

In this section, the negotiation model used in our study is explained. An extension of the alternating-offers protocol presented by Rubinstein [25] and also Faratin et al. [8] is used to describe the multi-issue bilateral negotiation in the context of e-commerce.

Negotiation Scenario refers to the environment and settings in which negotiation should be carried out. It mainly includes the negotiation context (e-commerce, politics...), negotiation parties and negotiation object.

Negotiators in automated bilateral negotiation refer to two parties presented by intelligent agents that can make decisions on behalf of their owners. In the context of e-commerce, according to the negotiation scenario, these agents can be a seller/buyer, supplier/customer or server/client. We use a, b to represent agents involved in negotiation.

Negotiation object is the set of conflicting issues (attributes) over which agreement must be reached [12]. Negotiation object may have many attributes such as price, delivery time, warranty duration, and so on. Let $A = \{a_1, a_2, \dots, a_n\}$ be the set of issues under negotiation. For each issue $a_j \in A$, agents have a domain $D_j = [\min_j, \max_j]$ where \min_j and \max_j are the lower and upper reservation values. These values are private and considered as part of agents' preferences. The agent $i \in \{a, b\}$ can evaluate an issue's value x_j ($j \in A$) by using its scoring function $f_j^i: D_j \rightarrow [0, 1]$ over the given domain. It is assumed that issues are independent and have different importance for agents. A weight vector $W^i = \langle w_1^i, w_2^i, \dots, w_n^i \rangle$ is used to represent the relative importance degrees among the issues for the agent i . This weight vector is normalized ($\sum_{j=1}^n w_j^i = 1$ where $i \in \{a, b\} \wedge j \in A$). Similar to reservation values, issues' scoring function and importance degree are private and considered as the agent's preferences. Agents assign a value $x_j \in D_j$ to each issue to make an offer $X = \{x_1, x_2, \dots, x_n\}$. The utility of a given offer x for the agent i is an additive function over negotiation issues [21]:

$$u^i = \sum_{j=1}^n w_j^i \cdot f_j^i(x_j) \quad (1)$$

Agents need to evaluate the utility of the received offer and make a decision about the ongoing negotiation. Then, agents will agree with the received offer if its utility is greater than an *aspiration-level* (θ). *Aspiration-level* is agents' desire utility that they wish to achieve it based on the negotiation time, and their opponent behavior. Agents change their aspiration-level during the negotiation process.

Initiation – the agent who starts the negotiation by sending the first offer can be chosen randomly. Let's say a is the starter and $x_{a \rightarrow b}^t$ is the sent offer from a to b at time t . Whenever agents receive an offer they have the right to make a decision whether to accept the offer, withdraw from the negotiation or propose a counter-offer (like $x_{b \rightarrow a}^t$). Each *round* (t) of the negotiation contains a pair of offers that are sent by agents (an offer from starter and a counter-offer from the other agent).

Termination – the agent who receives the last offer can terminate the negotiation by accepting the opponent's offer or by withdrawing from the negotiation. Accepting an offer is based on the agent's *aspiration level* (θ). The agent will accept the received offer as an agreement if the utility of the received offer is greater than or equal to its aspiration level. Generally, an agent's aspiration level is close to 1 at the beginning of the negotiation and close to its utility threshold (u_{min}) at end of the negotiation ($u_{min} \leq \theta \leq 1$). If an agent reaches to its maximum considered time (t_{max}) without any agreement then it will withdraw from the negotiation. Figure 1 shows agents' decision making procedure when it receives an offer. At first, they update the best received offer (br) based on the last received offer (lr). It is obvious that br has the highest utility among the received offers. Agents check whether they should accept the br whenever they receive an offer and make a concession on their aspiration level. If they cannot accept the br and no time is left to continue the negotiation then they will withdraw from negotiation, otherwise they will learn their opponent's preferences and generate a counter-offer to continue the negotiation. We assume that agents have the learning capability to explore their opponent's preferences. This ability will help them to generate near Pareto-optimal offers.

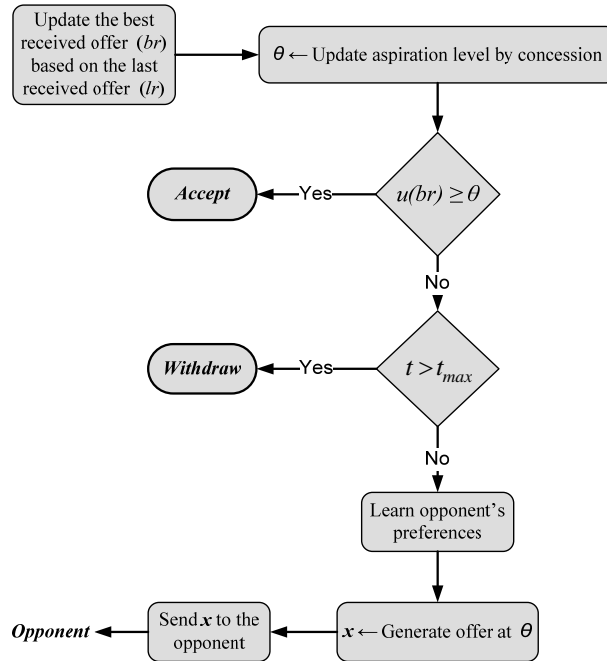


Figure 1: Decision making procedure when agents receives an offer by considering the learning capability

Pareto-optimal offer refers to a generated offer at the given aspiration level (θ) which has maximum utility for the opponent.

Iso-utility can be defined as a set of offers that have the same utility. Formally, given a desire aspiration level of utility θ , the iso-utility at level θ for the agent a can be defined as [8]:

$$iso^a = \{x | u^a(x) = \theta\} \quad (2)$$

Practically, generating offers at a given aspiration level, θ , is not always possible when there are some qualitative negotiation issues (with nominal values). Thus, we extend this definition by considering an aspiration area like $\bar{\theta} = [\alpha, \beta]$ instead of using a single value as θ . Now, *iso-utility* can be defined over a given area $\bar{\theta}$ as a set of offers that their utilities for the agent a are bounded between α and β . Formally, given a desire area of utility $\bar{\theta}$, the iso-utility in $\bar{\theta}$ for the agent a can be defined as:

$$\bar{\theta} = [\alpha, \beta] \quad (3)$$

$$iso^a(\bar{\theta}) = \{x | \alpha \leq u^a(x) \leq \beta\} \quad (4)$$

Iso-utility curve is a two dimensional graph that shows Pareto-frontier. This graph can be used to analyze the quality of generated offers and to trace agents' movement toward the joint agreement. Each point (u^a, u^b) at *iso-utility curve* is associated with a level of utility $u^a = \theta$ (or $u^a \in \theta$) and the opponent's maximum possible utility.

$$u^b = \max \{u^b(x) | x \in iso^a(u^a)\} \quad (5)$$

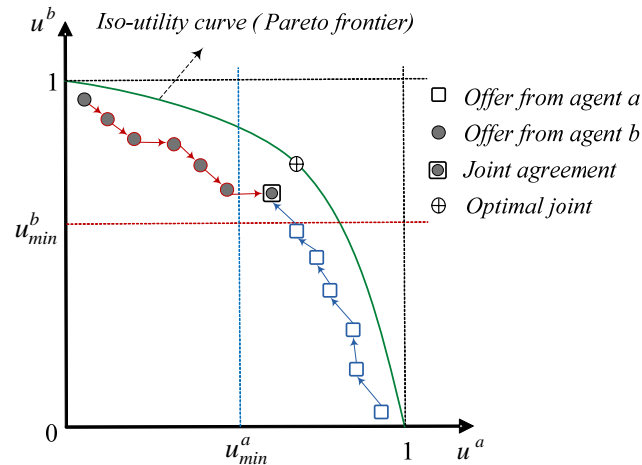


Figure 2: A sample iso-utility curve. Agents try to generate offers close to pareto-frontier

Figure 2 shows a sample iso-utility curve. At the beginning of the negotiation, agents' utility is close to 1 and their opponents' utility is almost zero. As time passes agents concede their aspiration level gradually and try to find an agreement. Learning the opponent's preferences helps the agent to generate offers close to Pareto-frontier and increases the opponent's satisfaction.

Preferences articulate agents' interest and intention in the negotiation. To select an offer among a set of different alternatives an agent uses its preferences. Moreover, the agent uses its preferences to evaluate a received offer and make a decision whether it should agree with or not. Agents' preferences are usually private to get more payoffs from the opponent. Having information about an opponent's importance weights over negotiation issues can help agents to generate high quality offers. In the following sections, preferences refer to agents' importance weights over the negotiation issues. We assume that these preferences are private and agents should learn each other preferences. This assumption (privacy) makes the automated negotiation more similar to the real world negotiation.

4 Learning Approach

Negotiators usually hide their preferences to protect their payoffs. Our goal is to present a learning approach to explore an opponent's importance degrees under the condition of incomplete information. Figure 3 shows our proposed learning procedure. The agent considers a population of the tentative chromosomes (preferences) and then refines it by omitting the low fitted chromosome during the negotiation. The best fitted chromosome will be selected as the opponent's preferences. Before starting the negotiation, agents decode the crisp values to fuzzy values to shrink the search space. Then, the fact that $\sum w_j = 1$ is applied to omit some chromosomes from the population. Later, we show that this refinement will downsize more than 80% of the population. Next, in each round of the negotiation, agents receive an offer and update their belief about their opponent's importance weight. As time passes, agents make more refinements and select the best fitted chromosome as their opponent's weight vector. The selected vector, in each round, is used to generate a near Pareto-optimal offer. The following sections explain the search process in more details.

4.1 Encoding the Preferences

Agents do not need to know their opponent's exact preferences [7], and learning an approximation of the importance degrees can help them to perceive their opponent's interest.

To learn an opponent's importance degrees, agents may have two different views. In case they want to learn the crisp value of an importance degree they should answer to this question:

What is the opponent's importance degree w_j when $w_j \in (0,1)$ and $\sum w_j = 1$? To answer this question they need to consider that there are many points in $(0,1)$ and it will be very hard to find an exact value for w_j .

In case they want to learn an approximate (uncertain) value of an importance degree, the following question should be answered:

How important is the j -th issue for the opponent? Or, is the j -th issue low-important or is it moderate/high-important for the opponent?

It is obvious that answering to the later question is easier because it is limited to three choices and the search space is much smaller. Encoding the importance degrees into the fuzzy sets can reduce the search space effectively, but we will lose the accuracy. In other words, by considering the fuzzy values for an opponent's importance weights, we can reduce the complexity of learning problem by searching in a set of fuzzy values.

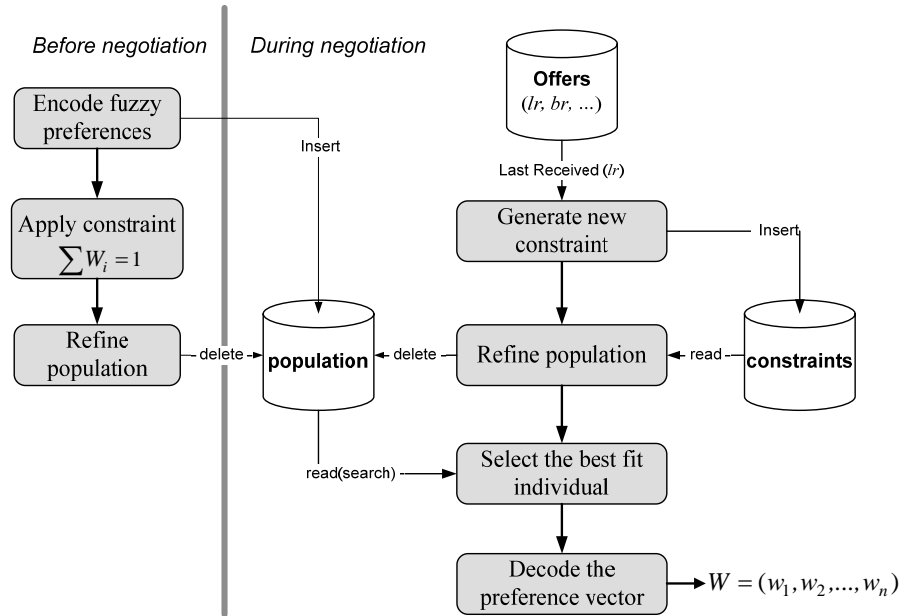


Figure 3: Learning procedure is modeled by a search process.

Figure 4 shows a general trapezoid membership function used in this study to encode importance degrees. Importance weights can be categorized as *very low* (VL), *low* (L), *moderate* (M), *high* (H), *very high* (VH), and so on. Each trapezoid is determined by four points A, B, C, D and a parameter, a . The width of trapezoid ($A - D$) depends on the number of negotiation issues (n) and is presented by $5a$, where $a = \frac{2}{9n}$ and $B = A + 2a$; $C = A + 3a$; $D = A + 5a$.

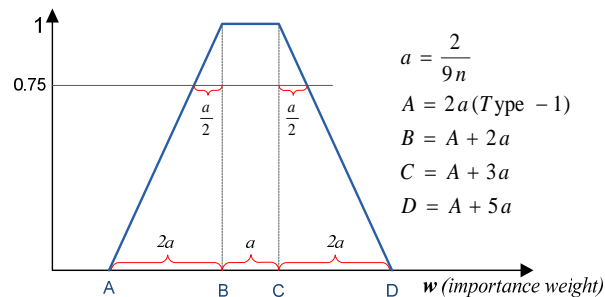


Figure 4: A sample trapezoid membership function to represent an uncertain importance weight. It is presented by four points A, B, C, D and a parameter, a . The support area is $5a$ (from A to D)

The starting point of the trapezoid (A) is determined by $A = 2a(\text{Type} - 1)$, where $\text{Type} = 0$ represents the *very low* membership function, $\text{Type} = 1$ represents the *low* membership function and so on. The number of membership functions N_{mf} ($0 \leq \text{Type} < N_{mf}$) depends on the number of negotiation issues (n). We use the following formula to set the number of membership functions:

$$N_{mf} = \left\lceil \frac{9n + 2}{4} \right\rceil \quad (6)$$

When a negotiation has three issues, we will have 7 membership functions shown in Figure 5. Issues' importance degrees can be expressed by a vector such as [VL, M, VH], [M, M, M], [H, VL, M] and so on. There are 7^3 possible vectors to express the importance degrees, but most of them (For example, [VL, VL, VL] or [VH, H, M]) cannot satisfy the main constraint ($\sum w_j = 1$). Thus, we need an initial refinement to omit some vectors which reduces the search space size.

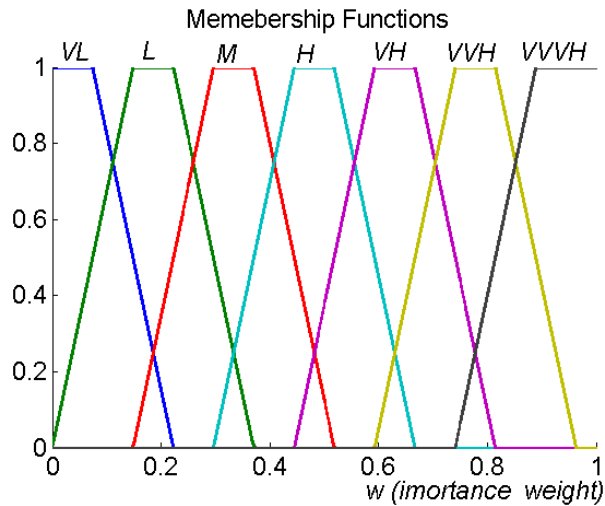


Figure 5: Seven membership functions to express the importance degrees in a negotiation over 3 issues

4.2 Decoding the Preferences

Decoding is used to convert a given uncertain value to a crisp value. A given membership function (known by position of four points A, B, C, D) can be initially decoded by $w = 0.5(B + C)$. In most cases, these initial values cannot satisfy the main constraint which is $(\sum w_j = 1)$. Therefore, the crisp value can be determined by considering an error value.

Now, let's say each vector is a chromosome (individual) and each membership function is a gene. In case the negotiation has three issues ($n = 3$) we will have seven genes $N_{mf} = 7$ and initial population contains $N_{mf}^n = 7^3 = 343$ individuals. Individuals can simply be decoded by the following pseudo-code (Figure 6):

```
//Decoding the genes to crisp value in chromosome
for each gene  $w_j$  in chromosome  $W$ 
     $w_j \leftarrow 0.5(B_j + C_j)$ 
error =  $1 - \sum w_j$ 
for each gene  $w_j$  in chromosome  $W$ 
     $w_j \leftarrow w_j + \frac{error}{n}$ 
```

Figure 6: Decoding the genes to the crisp values by considering the $\sum w_j = 1$

4.3 Refining the Population

After decoding, the gene value should be in $(0, 1)$. Otherwise, it is invalid and the chromosome containing this gene can be omitted from the initial population. Moreover, chromosomes which have a gene with membership function less than 0.75 will be omitted from the initial population. The refinement process is given in Figure 7. The preferences' population size will significantly shrink after the refinement process.

```
// Refining the initial population
for each chromosome  $W$  in population
    Decode( $W$ )
for each gene  $w_j$  of  $W$ 
    if  $w_j \notin (0, 1)$  or  $MF(w_j) < 0.75$ 
        delete  $W$  from the population
```

Figure 7: Refining the initial population

Table 1 shows that refining the initial population by applying the main constraint will reduce the search space effectively. For example, in the negotiation over three issues, 81% of individuals will be pruned. Thus, agents need to consider just 64 chromosomes as their opponent's importance degrees at the beginning of the negotiation.

Table 1: Applying the main constraint on the initial population

#Issues	#Genes	Population before refinement	Population after refinement	Reduction
3	7	343	64	81.3%
4	9	6,561	552	91.5%
5	11	161,051	3857	97.6%

After refinement of the initial population, agents are ready to start the negotiation. Whenever they receive an offer, they can update their beliefs about their opponent's preferences. This exploration continues during the negotiation process and helps them to generate near Pareto-optimal offers.

4.4 Online Exploration

So far, we have shown how agents can prepare an initial population of possible preferences. This initial population can be formed at the beginning of the negotiation. Having some information about the opponent's concession tactic may help an agent to learn its opponent's preferences.

Agents may behave as a *conceder*, *Boulware* or they may use a *fixed concession* tactic to change their aspiration level. Figure 8 shows concession tactics, while a *Boulware* agent hardly reduces its aspiration-level at the beginning of the negotiation. On the other hand, a conceder agent reduces its aspiration-level very fast to satisfy its opponent.

An agent with a fix concession tactic reduces its aspiration-level in a monotonic manner. It is obvious that agents start the negotiation with a high aspiration-level. Then, during the negotiation, they reduce their aspiration-level by making concession. Therefore, an agent's utility can be bounded $\alpha_t \leq \theta_t \leq \beta_t$ in each round of the negotiation where $\beta_1 = 1$ and $\alpha_{t_{max}} = u_{min}$. Agents can compute their aspiration-level by using the following formula:

$$\alpha_t = 1 + (u_{min} - 1) \left(\frac{t}{t_{max}} \right)^\lambda; \quad \beta_t = 1 + (u_{min} - 1) \left(\frac{t-1}{t_{max}} \right)^\lambda \quad (7)$$

where $\lambda = 1$ presents the fixed concession while ($\lambda < 1$ and $\lambda > 1$) values identify *conceder* and *Boulware* tactics respectively.

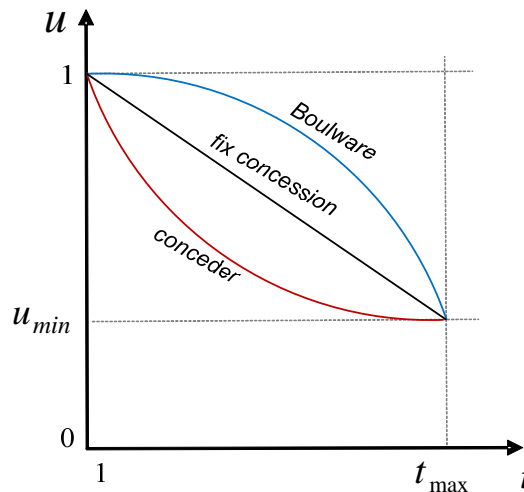


Figure 8: Concession tactics: conceder, fixed and Boulware

We assumed that agents are aware of their opponent's concession tactic and the expected time (t_{max}) for negotiation. Thus, they can have an estimation of their opponent's utility, which helps to learn the importance degrees.

Let's say u' is the opponent utility and the agent wants to explore the opponent importance degrees W' (prime notation is used to show the opponent). Agents negotiate over conflicting issues meaning that increasing the scoring function of j -th issue for an agent will decrease the scoring function of its opponent, and vice versa. Therefore, the opponent's scoring function f'_j can model by $1 - f_j$, and the opponent's utility by:

$$u'(x_t) = \sum_{j=1}^n w'_j \cdot (1 - f_j(x_t)) \quad (8)$$

where x_t is the received offer at round t . Having an opponent's concession tactic helps the agent to know the boundaries of the opponent's utility.

$$\alpha'_t \leq u'_t \leq \beta'_t$$

$$\alpha'_t \leq \sum_{j=1}^n w'_j \cdot (1 - f_j(x_t)) \leq \beta'_t \quad (9)$$

Agents can evaluate a given chromosome, $w = \langle w_1, w_2, \dots, w_n \rangle$, in the population by finding its error $E_w(x_t)$.

$$E_w(x_t) = |u'(x_t) - 0.5(\alpha + \beta)| + \begin{cases} \sqrt{\alpha' - u'(x_t)} & u'(x_t) < \alpha' \\ 0 & \alpha' \leq u'(x_t) \leq \beta' \\ \sqrt{u'(x_t) - \beta'} & u'(x_t) > \beta' \end{cases} \quad (10)$$

Given a set of received offers, S_r , the accumulative error \bar{E}_w can be calculated by the following formula:

$$\bar{E}_w = \sum_{x \in S_r} E_w(x) \quad (11)$$

Agents can evaluate the fitness of a given chromosome $w \in D_w$ by using fitness function $fitness: D_w \rightarrow [0,1]$ based on the accumulative error.

$$fitness = \frac{1}{1 + \bar{E}_w} \quad (12)$$

Finally, agents search among chromosomes and select the one which has the highest fitness as the opponent preferences. Then, agents decode the genes' crisp value to estimate their opponent's importance weights.

The number of chromosomes in a population affects the learning complexity, because agents need to update the fitness of all individuals and select the best fitted chromosome. Although, the learning algorithm is not too complex, agents can further reduce this complexity by pruning the very low fitted chromosomes from the population (For example, they can delete 10% of the population after updating the fitness of the chromosomes).

In this study which is similar to recent studies on bilateral negotiation, we considered scenarios where the number of negotiation issues is 3 or 4. As the population is very small we do not need to regenerate the population by *cross-over* and *mutation* techniques. In high dimensional negotiation, the population size will be large, and thus, agents can select an initial population and regenerate the population by using *cross-over* and *mutation* techniques.

5 Experimental Evaluation

This section illustrates a series of experiments which were carried out to evaluate the impact of our proposed learning approach. First, the negotiation strategy used in these experiments is presented. Next, the experimental setting and negotiation scenarios are described. Then, the negotiation metrics which used to evaluate the efficiency of the proposed approach will be explained.

5.1 Offer-Generating Strategy

Learning the opponent's preferences is the main concern in this study. However, to evaluate the impact of the presented learning approach on the negotiation outcome, we need an offer-generating algorithm. That is, the well-known *trade-off algorithm* with fuzzy similarity [9] is used. Trade-off algorithm tries to keep an agent utility while maximizes the opponent's utility by making trade-off between issues. In other words, it makes trade-off by reducing the scoring function for low importance issues and increasing the scoring function for high importance issues. It generates a set of random offers (called children) and then tries to find the most similar one to the last received offer by using a fuzzy similarity function:

$$Similarity(x, y) = \sum_{j=1}^n Sim_j(x_j, y_j) \quad (13)$$

where x is a candidate offer to be sent to the opponent and y is the last received offer and sim_j is the similarity over j -th issue. Success of the trade-off algorithm depends on the number of random offers (children) and the agent's information about the opponent's importance degrees w' .

We use the trade-off algorithm to generate offers in the given aspiration level area $[\alpha, \beta]$ (according to Equation 1). Therefore, agents can make a concession and then generate an offer according to the learned preferences. Faratin *et al.* [9] presented the trade-off algorithm without any learning over the opponent's importance degrees. Thus, learning the opponent's importance weights can be considered as a complementary to their work. Although, our work is complementary to the trade-off algorithm, it is independent of the offer-generating strategy and can be combined with any other strategies as well.

5.2 Experimental Settings and Scenarios

To remove the effect of the negotiation scenario on the negotiation outcome, we had chosen two scenarios from the former studies [6], [24]. The starter agent was selected randomly to remove the advantage/disadvantage of the first mover. Agents initiate the negotiation by sending an offer with the highest utility. We have used two linear functions S, Z to describe the increasing and the decreasing scoring functions as follows:

$$S(x, a, c) = \frac{x - a}{c - a}; \quad Z(x, a, c) = \frac{x - c}{a - c} \quad (14)$$

The following independent variables are used in the experiments:

- *Agents information* about their opponent's importance weights. This information was divided into three groups: *i) perfect information*, where agents knows the exact importance weight; *ii) no-information*, where agent has no idea about its opponent's importance weights and, therefore, we used random values as the opponent importance weights in each repetition of negotiation; *iii) elicited information*, where agent learns the opponent importance weights during the negotiation.
- The number of random children (offers) in *trade-off* algorithm. Agents in our experiments used the trade-off algorithm to generate offers. The performance of the trade-off algorithm is related to the number of steps and children in each step. We used a single step in all repetitions of the trade-off algorithm, but we considered 10 different values to set the number of random children (offers) which were $\{2^1, 2^2, \dots, 2^{10}\}$.

5.2.1 Scenario 1

This scenario is almost similar to the scenario reported in [24]. In this scenario, agents negotiate over four issues (price, color, material, delivery time). Two samples of the importance weights were provided to evaluate agents' ability to learn these importance degrees. The following general settings were applied to the negotiation and involving agents:

- *Issues' domain*- Agents could choose issues' value from the following domains: price [30, 70], color [0, 5], material [0, 4], delivery time [5, 15].
- *Scoring functions*-Table 2 shows the scoring functions for the given issues.
- *Threshold utility*- The threshold utility u_{min} was set to 0.55 for both agents.
- *Maximum rounds*- Both agents supposed to have the maximum of 8 rounds.

To evaluate an agent's ability to learn its opponent's preferences we considered two samples where agents had different importance weights. The importance weights for agents a and b are given in Table 3. Each sample determines a new negotiation.

Table 2: Scoring functions used in scenario 1

Negotiation Issue	scoring function	
	Agent a	Agent b
price	$S(x, 30, 70)$	$Z(x, 30, 70)$
color	$Z(x, 0, 5)$	$S(x, 0, 5)$
material	$Z(x, 0, 4)$	$S(x, 0, 4)$
delivery time	$S(x, 5, 15)$	$Z(x, 5, 15)$

Table 3: Agents' importance weights in scenario 1

Sample	Agent	price	color	material	delivery time
Sample 1	Agent a	0.50	0.13	0.17	0.20
	Agent b	0.30	0.20	0.15	0.35
Sample 2	Agent a	0.30	0.10	0.20	0.40
	Agent b	0.45	0.17	0.13	0.25

5.2.2 Scenario 2

This scenario is almost similar to the scenario reported in [6]. In this scenario, agents negotiate over three abstracted issues (issue1, issue2, issue3). Similar to scenario 1, two samples of the importance weights were provided to evaluate agents' ability to learn these importance degrees. The following general settings were applied to the negotiation and involving agents:

- *Issues' domain*- Agents could choose issues' value from the following domains: *issue1* [5000,10000], *issue2* [30,90], *issue3* [1,3].
- *Scoring functions*- Table 4 shows the scoring functions for the given issues.
- *Threshold utility*- The threshold utility u_{min} was set to 0.55 for both agents.
- *Maximum rounds*- Both agents supposed to have the maximum of 8 rounds.

To evaluate an agent's ability to learn its opponent's preferences we considered two samples where agents had different importance weights. The importance weights for agents *a* and *b* are given in Table 5. Each sample determines a new negotiation.

Table 4: Scoring functions used in scenario 2

Negotiation Issue	scoring function	
	Agent a	Agent b
issue 1	$S(x, 5000, 10000)$	$Z(x, 5000, 10000)$
issue 2	$Z(x, 30, 90)$	$S(x, 30, 90)$
issue 3	$S(x, 0, 3)$	$Z(x, 0, 3)$

Table 5: Agents' importance weights in scenario 2

Sample	Agent	issue 1	issue 2	issue 3
Sample 1	Agent a	0.35	0.50	0.15
	Agent b	0.55	0.35	0.10
Sample 2	Agent a	0.42	0.48	0.10
	Agent b	0.55	0.33	0.12

5.3 Evaluation metrics

The aim of learning an opponent's preferences is to increase the agent's chance to reach a high quality agreement. To this end, the following metrics were used to measure agents' ability of learning its opponent's preferences and its effects on the negotiation outcome.

5.3.1 Learning error

This metric can show the distance between the learned weights and the real weights. As the importance degrees are expressed by a vector, therefore, the distance between the real vector and the learned vector can articulate the learning error.

$$e = \sqrt{\sum_{j=1}^n (w_j - w'_j)^2} \quad (15)$$

where w_j is the opponent's real weight for j -th issue and w'_j is the learned weight.

5.3.2 Probability of success (*pos*)

To remove the effects of the random offers in the trade-off algorithm, we need to repeat the negotiation samples and observe the outcomes. *pos* shows the chance of reaching an agreement in the given negotiation setting by counting the number of times that the agent reaches agreements. In our experiment, a given negotiation was repeated 1000 times to remove the effects of the random offers in the trade-off algorithm.

$$pos = \frac{\# \text{ agreement}}{\# \text{ repetition of the negotiation}} \quad (16)$$

5.3.3 Joint utility and quality of agreement (*QoA*)

The following metrics show the overall outcome of the negotiation by considering the benefit of both agents. In other words, these metrics can show the quality of the success (agreement). Given the utility of agents *a*, *b* the geometric mean can show the joint utility of the negotiation $G: \leftarrow [0, 1] \times [0, 1] \rightarrow [0, 1]$ where:

$$G = \begin{cases} \sqrt{u^a \cdot u^b} & \text{agree} \\ 0 & \text{withdraw} \end{cases} \quad (17)$$

Similar to u^a, u^b the joint utility is also a value in $[0, 1]$. When agents cannot reach any agreement, the joint utility will be considered as zero. Now, let's say G_i is the joint utility of *i*-th run of a negotiation. Then, the quality of agreement (*QoA*) can be defined as the average joint utilities of all agreements.

$$QoA = \frac{1}{k} \sum_{i=1}^N G_i \quad (18)$$

where *N* is the number of negotiation runs and *k* is the number of times that agents reach to agreements.

5.4 Evaluation of the learning approach

The aims of these experiments were to test the proficiency of the presented learning method and its effect on the negotiation outcome. To this end, first, we tried to evaluate the proficiency of the learning method. For simplicity, in these experiments, our learning method is called *hybrid* method.

Before evaluating the hypotheses, we would like to demonstrate agents' actions based on the *Pareto-frontier* curves in Figure 9. Each row in this figure refers to a negotiation sample. Graphs A, B and C are related to the sample negotiation 1 of the scenario 1 and graphs D, E and F are related to the sample 2 of the scenario 1. Similarly, Graphs G, H and I are related to the sample negotiation 1 of the scenario 2 and graphs J, K and L are related to the sample 2 of the scenario 2. The left column (graphs A, D, G and J) refers to agents without any learning capability while the middle column (graphs B, E, H, K) refers to agents which used the hybrid learning approach and the right column (graphs C, F, I, L) refers to agents with perfect information. It can be perceived that agents equipped with the hybrid learning capability can generate near Pareto-optimal offers. To analyze the results in more detail a series of hypotheses were proposed and tested as follows:

Hypothesis 1. The hybrid method can approximately learn an opponent's importance weights during the negotiation process.

To test the proficiency of the *hybrid* method, it was assumed that agents had no information about their opponent's importance weights at the beginning of the negotiation, but they had tailored with the *hybrid* approach to learn their opponent's preferences. Hence, opponents' weights were initialized with random values. Then, the distance between the learned weights and the real weights (learning error) was recorded in each round of the negotiation. Negotiations were repeated 1000 times for the given number of random offers (*children* = 128). Figure 10 shows the average learning error in each round of the sample negotiations. As time passes, the learning errors are reduced. In this figure, graphs A, B, C, D are related to Scenario 1 sample negotiation 1, Scenario 1 sample negotiation 2, Scenario 2 sample negotiation 1, and Scenario 2 sample negotiation 2, respectively.

The maximum of negotiation rounds was set to 8 in all runs which provides the maximum of 8 samples for learning. The experiment showed that at the beginning of the negotiation agents had their maximum learning error, but as time passed agents could learn their opponent's weights and reduce the learning error. Actually, in each round of the negotiation, agents could receive an offer and use it as a new sample for the incremental learning. Figure 10 shows that agents can reduce the learning error during the negotiation process and explore an approximate weight vector.

Now the question is, *how can the hybrid method affect the quality of the negotiation outcome?* In other words, it is important to know whether learning approximate weights by the *hybrid* method is effective enough to reach a high

quality agreement or not. As Faratin *et al.* [9] have shown, the trade-off algorithm can provide better offers if an agent has perfect (or partial) information about its opponent.

The following hypotheses were tested to elucidate the effects of learned weights on the negotiation outcome.

Hypothesis 2. Learning by the hybrid method can increase the chance of reaching agreements.

To evaluate the effects of *hybrid* approach, we considered three types of agents: (i) *perfect*: an agent with perfect information about its opponent's importance weights, (ii) *hybrid*: an agent equipped with the *hybrid* learning approach which had no information about its opponent's importance weights, and finally (iii) *none*: an agent without any learning capability and information about its opponent.

Then, the probability of success (*pos*) was measured for each pair of agents (for example (*perfect-none*) or (*perfect-hybrid*)). Since, the performance of the trade-off algorithm depends on the number of random offers (children), we conducted our experiments with different number of random offers.

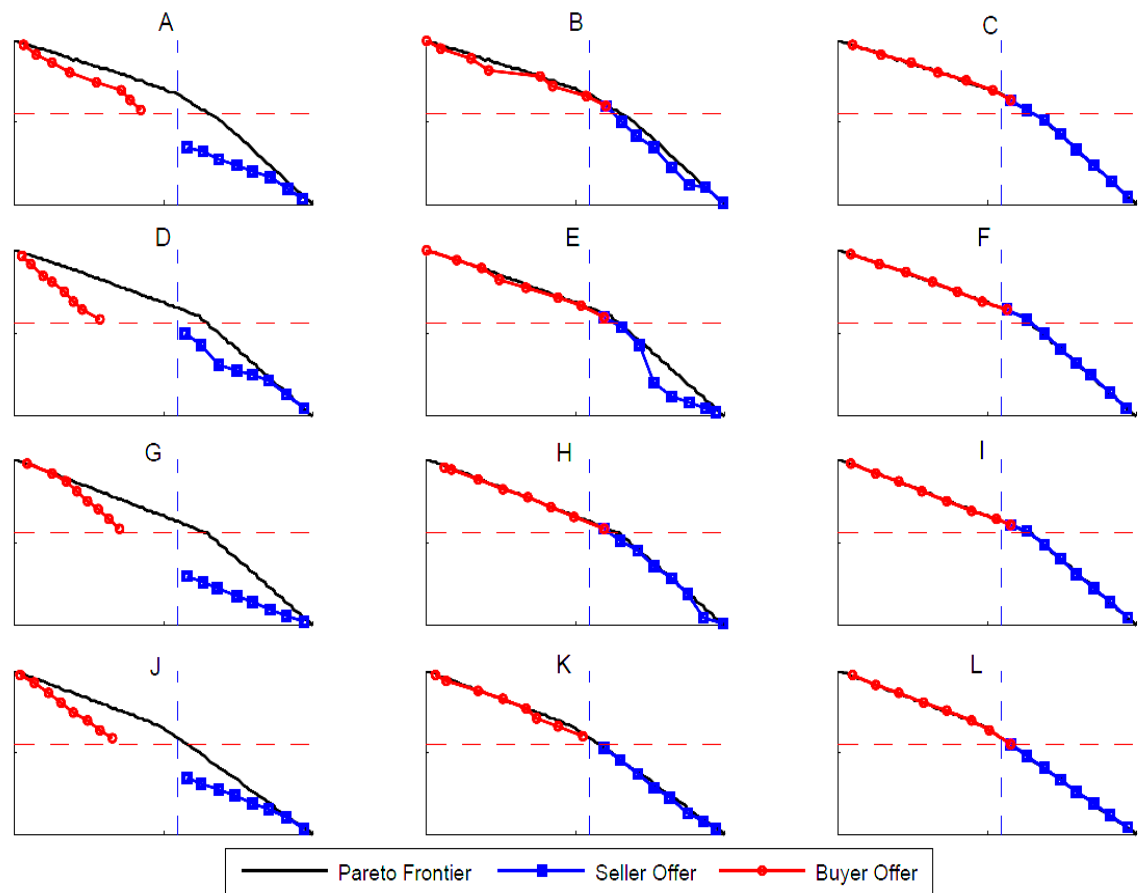


Figure 9: Agents' actions based on Pareto-frontier curves in four sample negotiations.

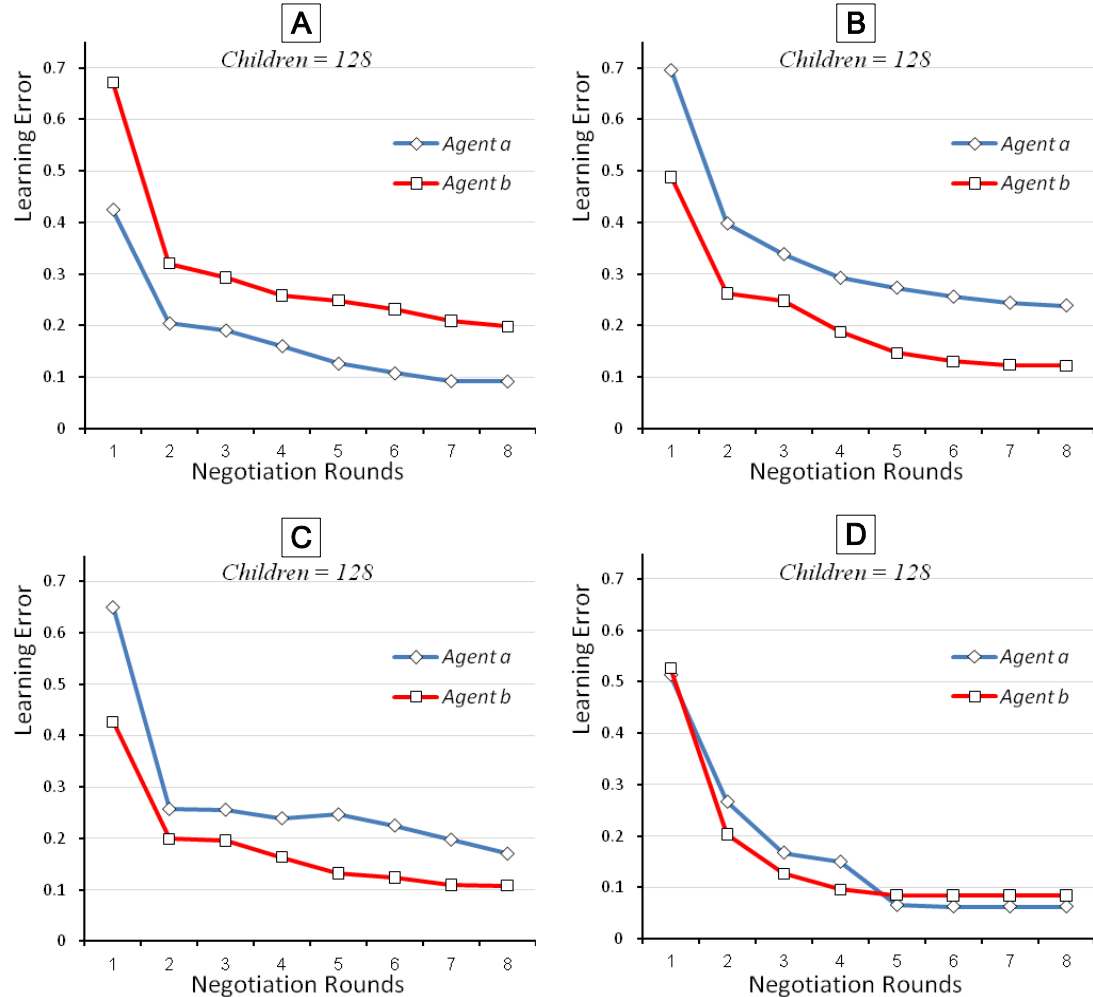


Figure 10: Learning errors in 8 negotiation rounds. As time passes, the learning errors are reduced

Figure 11 shows the *pos* metric for five possible pairs of agents. Agents with learning capability have higher *pos* when compared to agents without learning. In this figure, graphs A, B, C, D are related to Scenario 1 sample negotiation 1, Scenario 1 sample negotiation 2, Scenario 2 sample negotiation 1, and Scenario 2 sample negotiation 2, respectively. It can be perceived that the maximum *pos* had occurred when both agents had perfect information about each other. In contrast, when agents had no information about each other, the chance of reaching an agreement had its minimum value. It can also be seen in both scenarios that pair of (*hybrid- hybrid*) outperformed the pairs of (*none-none*), (*none- hybrid*), (*perfect-none*).

It is perceivable that the probability of reaching an agreement depends on two parameters: (*i*) the number of random offers which is related to the nature of the trade-off algorithm, and (*ii*) the provided information to agents. If we show the ">" as an operator which articulates the greater chance of reaching an agreement, then according to Figure 12, the following precedence can be clearly concluded:

$$(Perfect - Perfect) > (Perfect - Hybrid) > (Hybrid - Hybrid) > (Hybrid - None) > (None - None) \quad (19)$$

Thus, the *hybrid* approach can improve the probability of reaching an agreement and hypothesis 2 is accepted.

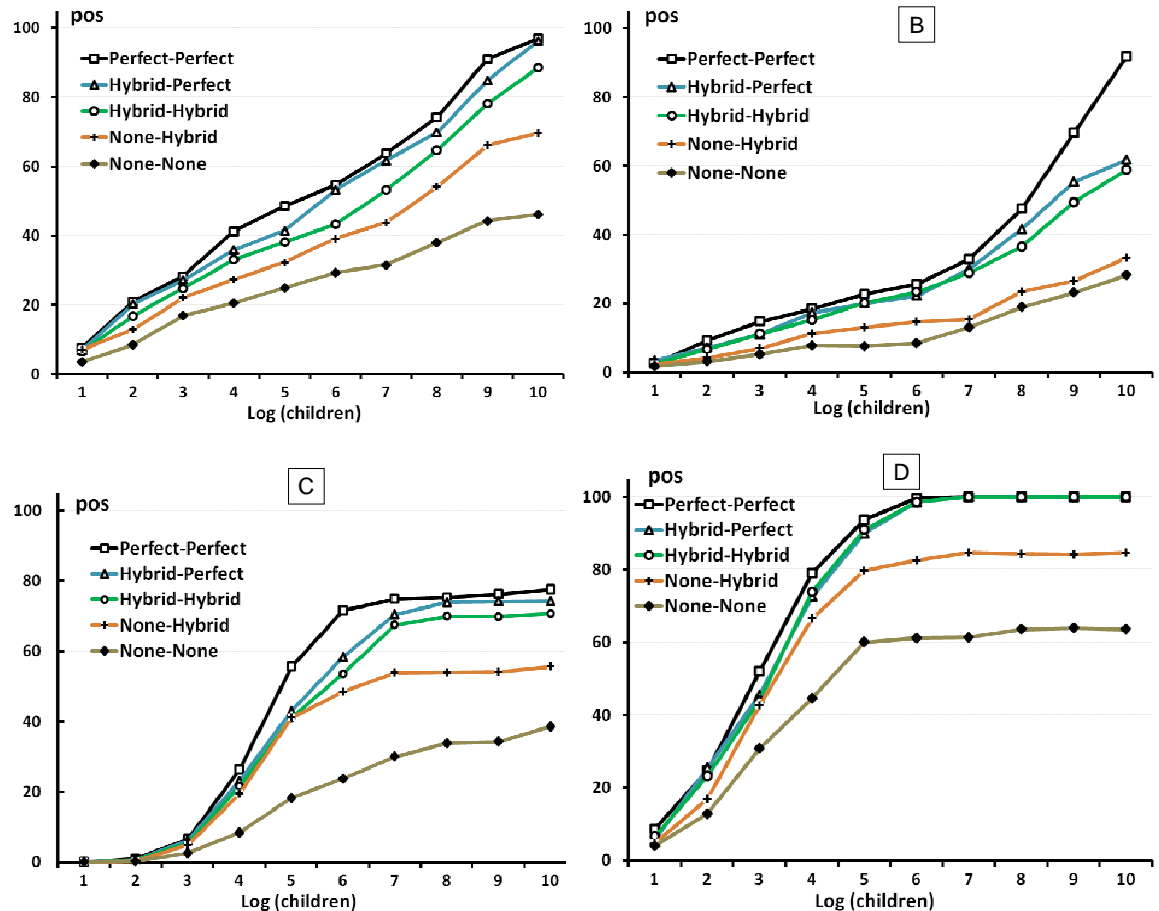


Figure 11: Possibility of success (*pos*) in four sample negotiations. Agents with learning capability have higher *pos* when compared to agents without learning

Hypothesis 3. Two agents equipped with the hybrid learning ability have higher chances than a pair of agents where one has perfect information and the other one has no information about its opponent.

In this experiment the pairs of (*hybrid-hybrid*) and (*perfect-none*) were examined to find their chance to reach an agreement. The results are presented in Figure 12. In this figure, graphs A, B, C, D are related to Scenario 1 sample negotiation 1, Scenario 1 sample negotiation 2, Scenario 2 sample negotiation 1, and Scenario 2 sample negotiation 2, respectively. In all negotiation samples it can be clearly seen that the pair of (*hybrid-hybrid*) outperforms the pair of (*perfect-none*). In other words, this result tells us that having an opponent equipped with the *hybrid* learning approach is better than having perfect information about the opponent's preferences because bilateral negotiation is a process in which both agents should try to satisfy each other. When one of the negotiation parties has no learning capability it will reduce the chance of reaching an agreement, even if the other side has perfect information about its opponent.

So far, we have shown that the *hybrid* learning approach can increase the chance of reaching agreements, but the quality of the agreement is not evaluated yet. The following hypothesis has been used to evaluate the quality of the negotiation's outcome.

Hypothesis 4. Learning by the hybrid method can increase the quality of agreement (QoA).

To evaluate the quality of agreements, we considered the pairs of (*hybrid-hybrid*) and (*none-none*).

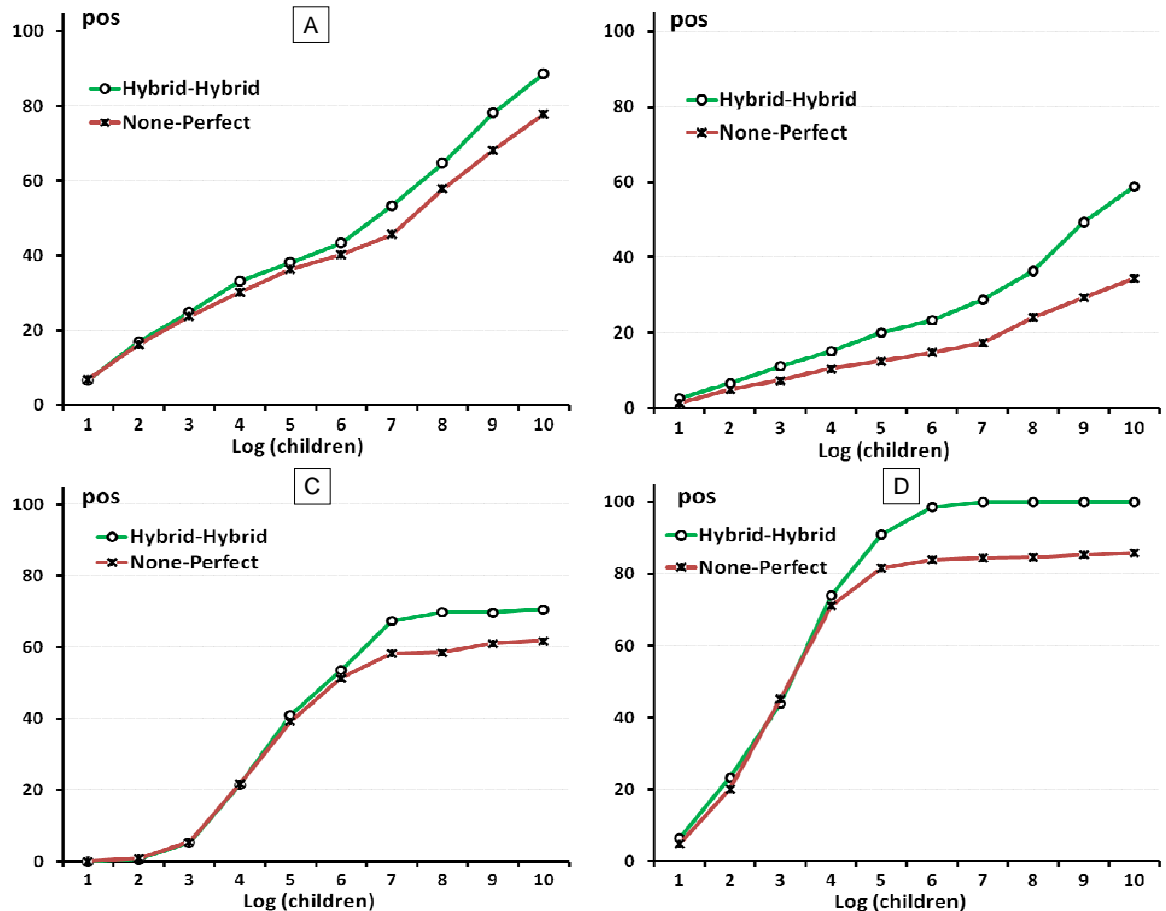


Figure 12: Comparing the possibility of success in the pairs of (*Hybrid-Hybrid*) and (*perfect-none*)

Figure 13 shows the quality of agreement according to the different number of random children in the trade-off algorithm. In this figure, graphs A, B, C, D are related to Scenario 1 sample negotiation 1, Scenario 1 sample negotiation 2, Scenario 2 sample negotiation 1, and Scenario 2 sample negotiation 2, respectively. We applied statistical *T-test* to evaluate the effect of the *hybrid* learning on the negotiation outcome. A series of paired-samples *T-test* with the significance level of 0.05 were conducted to compare the quality of agreements in four sample negotiations.

- Experiment in sample negotiation 1 of scenario 1 (graph A of Figure 13) showed that there was significant difference in the quality of agreement for pair of (*hybrid-hybrid*) ($M = 0.5872$) and pair of (*none-none*) ($M = 0.5863$); $t(9) = 2.05, p = 0.035$.
- Experiment in sample negotiation 2 of scenario 1 (graph B of Figure 13) showed that there was significant difference in the quality of agreement for pair of (*hybrid-hybrid*) ($M = 0.5878$) and pair of (*none-none*) ($M = 0.5859$); $t(9) = 4.19, p = 0.001$.
- Experiment in sample negotiation 1 of scenario 2 (graph C of Figure 13) showed that there was significant difference in the quality of agreement for pair of (*hybrid-hybrid*) ($M = 0.5109$) and pair of (*none-none*) ($M = 0.5102$); $t(9) = 2.52, p = 0.016$.
- Experiment in sample negotiation 2 of scenario 2 (graph D of Figure 13) showed that there was significant difference in the quality of agreement for pair of (*hybrid-hybrid*) ($M = 0.5783$) and pair of (*none-none*) ($M = 0.5764$); $t(9) = 6.65, p = 0.000$.

The results suggest that agents with the *hybrid* learning capability can find agreements with higher joint utility in compare to agents without any learning capability. Although, the quality of a generated offer depends on the offer-generating algorithm, our experiments showed that learning an opponent's preferences not only increases the chance of reaching an agreement but also helps agents to improve their social welfare.

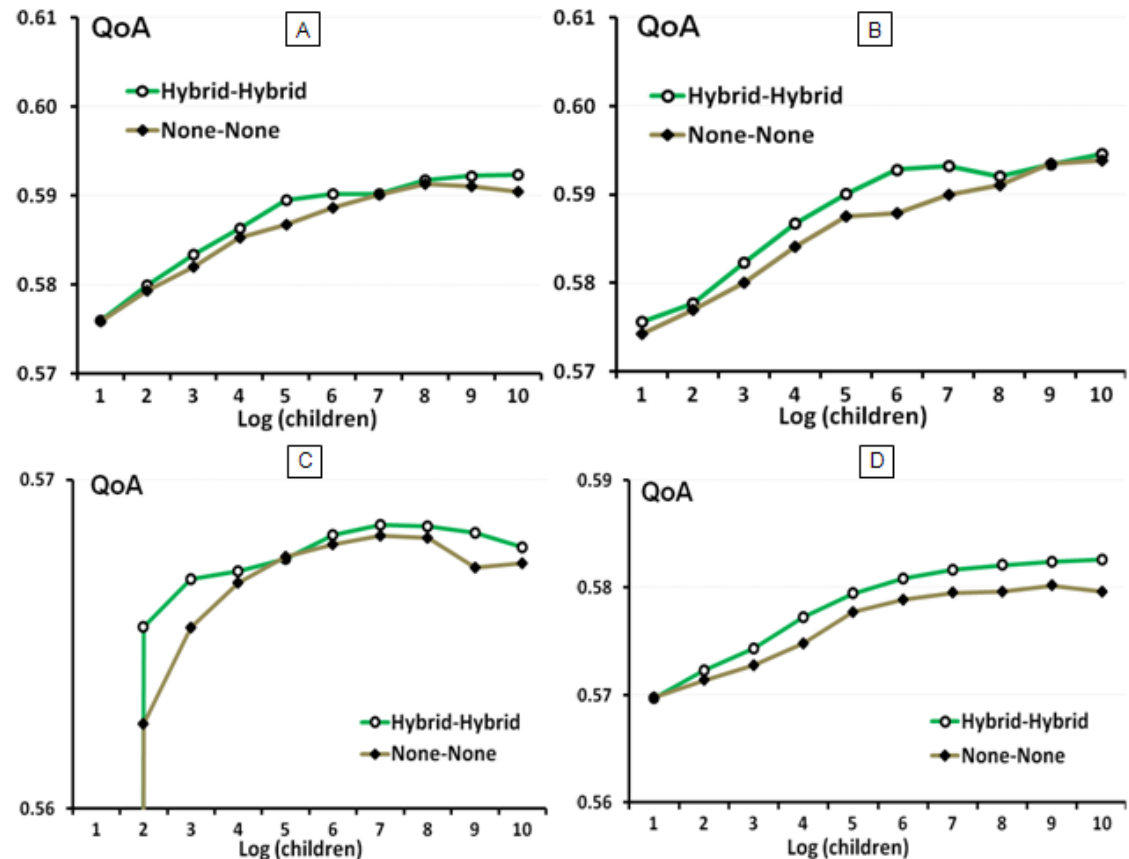


Figure 13: Comparing possibility of success in the pairs of (*Hybrid- Hybrid*) and (*none-none*)

6 Conclusions

This paper has presented a learning approach, based on hybrid soft-computing techniques, to estimate an opponent's preferences with incomplete information in automated bilateral negotiation. The presented learning approach is based on fuzzy techniques, genetic algorithm and constraint satisfaction. In particular, fuzzy membership functions have been used to encode/decode uncertain information about opponents' preferences. These functions reduce the possible preferences in the search space and enable agents to explore information with limited online samples. Moreover, reducing the population of possible preferences by using fuzzy encoding brought us a simpler genetic algorithm where the population (search space) is small enough to ignore the *mutation* and *cross-over* operations. In each round of the negotiation, agents use the received offer to form a new constraint and update the fitness of individuals in the given population of preferences. From this basis, agents can learn/explore their opponent's preferences incrementally and, consequently, choose the best fitted weight vector as the learned preferences to generate a high quality offer.

Our presented learning method is independent of offer-generating strategy and can be mixed with any other offer-generating algorithms. In this work, the trade-off algorithm with fuzzy similarity [9] is used to generate offers. The empirical evaluation showed that agents can learn an estimation of their opponent's preferences. Moreover, it has been shown that the learning method increases agents' chance to reach an agreement while improves the quality of the outcome. It has also been shown that two agents with the learning ability have greater chances to find an agreement than a pair of agents where one has perfect information and the other has neither information nor a learning ability.

For the future, this work can be improved in many ways. Firstly, our learning method provides a near optimal estimation of the opponent's preference, and this estimation can be improved by developing a secondary search algorithm. Secondly, as the presented learning method provides fuzzy preferences, developing an algorithm to generate offers based on the fuzzy preferences can be the best match with it. Hence, proposing an offer-generating algorithm based on the fuzzy preferences is suggested. Thirdly, this study presents a hybrid method that uses a simple form of the genetic algorithm without *cross-over* and *mutation* operators. Designing proper crossover/mutation operators in high-dimensional negotiations would be an interesting study in the future. Finally, this study assumed that agents have information about their opponent's concession tactic, which may be unavailable at the beginning of

the negotiation. Therefore, developing an algorithm to reveal an opponent's concession tactic and its preferences simultaneously can be a challenging area of research.

Acknowledgments

Financial support from the *School of Graduate studies* (GSO) at University Putra Malaysia (UPM) is gratefully acknowledged. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped to improve the quality of the paper.

References

- [1] C. Arbib and F. Rossi, Optimal resource assignment through negotiation in a multi-agent manufacturing system, *IIE Transactions*, vol. 32, no. 10, pp. 963-974, 2000.
- [2] C. Beam and A. Segev, Automated negotiations: A survey of the state of the art, *Wirtschaftsinformatik*, vol. 39, no. 3, pp. 263-268, 1997.
- [3] K. Binmore and N. Vulkan, Applying game theory to automated negotiation, *Netnomics*, vol. 1, no. 1, pp. 1-9, 1999.
- [4] S. Buffett and B. Spencer, A bayesian classifier for learning opponents' preferences in multi-object automated negotiation, *Electronic Commerce Research and Applications*, vol. 6, no. 3, pp. 274-284, 2007.
- [5] R. Carbonneau, G. E. Kersten, and R. Vahidov, Predicting opponent's moves in electronic negotiations using neural networks, *Expert Systems with Applications: An international Journal*, vol. 34, no. 2, pp. 1266-1273, 2008.
- [6] C.-B. Cheng, C.-C. H. Chan, and K.-C. Lin, Intelligent agents for e-marketplace: Negotiation with issue trade-offs by fuzzy inference systems, *Decision Support Systems*, vol. 42, no. 2, pp. 626-638, 2006.
- [7] R. Coehoorn and N. Jennings, Learning on opponent's preferences to make effective multi-issue negotiation trade-offs, in *Proceedings of the 6th International Conference on Electronic Commerce*, Delft, Netherlands, October 25-27, 2004, pp. 59-68.
- [8] P. Faratin, C. Sierra, and N. Jennings, Negotiation decision functions for autonomous agents, *Robotics and Autonomous Systems*, vol. 24, no. 3, pp. 159-182, 1998.
- [9] P. Faratin, C. Sierra, and N. R. Jennings, Using similarity criteria to make issue trade-offs in automated negotiations, *Artificial Intelligence*, vol. 142, no. 2, pp. 205-237, 2002.
- [10] K. Hindriks and D. Tykhonov, Opponent modelling in automated multi-issue negotiation using bayesian learning, in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, May 12-16, 2008, pp. 331-338.
- [11] C. Huang, W. Liang, Y. Lai, and Y. Lin, The agent-based negotiation process for B2C e-commerce, *Expert Systems With Applications*, vol. 37, no. 1, pp. 348-359, 2010.
- [12] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, M. Wooldridge, and C. Sierra, Automated negotiation: Prospects, methods and challenges, *Group Decision and Negotiation*, vol. 10, no. 2, pp. 199-215, 2001.
- [13] C. Jonker, V. Robu, and J. Treur, An agent architecture for multi-attribute negotiation using incomplete preference information, *Autonomous Agents and Multi-Agent Systems*, vol. 15, no. 2, pp. 221-252, 2007.
- [14] G. Kersten and H. Lai, Negotiation support and e-negotiation systems: An overview, *Group Decision and Negotiation*, vol. 16, no. 6, pp. 553-586, 2007.
- [15] G. Kersten and G. Lo, Aspire: An integrated negotiation support system and software agents for e-business negotiation, *International Journal of Internet and Enterprise Management*, vol. 1, no. 3, pp. 293-315, 2003.
- [16] R. Kowalczyk and V. Bui, On constraint-based reasoning in e-negotiation agents, in *Proceedings Agent-Mediated Electronic Commerce III, Current Issues in Agent-Based Electronic Commerce Systems*, Barcelona, Catalonia, Spain, 2000, pp. 31-46.
- [17] R. Lin, S. Kraus, J. Wilkenfeld, and J. Barry, Negotiating with bounded rational agents in environments with incomplete information using an automated agent, *Artificial Intelligence*, vol. 172, no. 6-7, pp. 823-851, 2008.
- [18] X. Luo, N. Jennings, N. Shadbolt, H. Leung, and J. Lee, A fuzzy constraint based model for bilateral, multi-issue negotiations in semi-competitive environments, *Artificial Intelligence*, vol. 148, no. 1-2, pp. 53-102, 2003.
- [19] P. Maes, R. H. Guttman, and A. G. Moukas, Agents that buy and sell, *Communications of the ACM*, vol. 42, no. 3, pp. 81-91, 1999.
- [20] N. Matos, C. Sierra, N. R. Jennings, and Y. Demazeau, Determining successful negotiation strategies: An evolutionary approach, in *Proceedings of the 3rd International Conference on Multi-Agent Systems*, Paris, France, 1998, pp. 182-189.
- [21] H. Raiffa, *The Art and Science of Negotiation*. Cambridge: Harvard University Press, 1982.
- [22] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings, Negotiating using rewards, *Artificial Intelligence*, vol. 171, no. 10-15, pp. 805-837, 2007.
- [23] J. Richter, R. Kowalczyk, and M. Klusch, Multistage fuzzy decision making in bilateral negotiation with finite termination times, in *Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial intelligence*, Melbourne, Australia, December 1-4, 2009, pp. 21-30.
- [24] R. Ros and C. Sierra, A negotiation meta strategy combining trade-off and concession moves, *Autonomous Agents and Multi-Agent Systems*, vol. 12, no. 2, pp. 163-181, 2006.

- [25] A. Rubinstein, Perfect equilibrium in a bargaining model, *Econometrica: Journal of the Econometric Society*, vol. 50, no. 1, pp. 97-109, 1982.
- [26] T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions, *Artificial Intelligence*, vol. 135, no. 1-2, pp. 1-54, 2002.
- [27] M. Schoop, A. Jertila, and T. List, Negoisst: A negotiation support system for electronic business-to-business negotiations in e-commerce, *Data & Knowledge Engineering*, vol. 47, no. 3, pp. 371-401, 2003.
- [28] F. Teuteberg, Experimental evaluation of a model for multilateral negotiation with fuzzy preferences on an agent-based marketplace, *Electronic Markets*, vol. 13, no. 1, pp. 21-32, 2003.
- [29] M. Wang, J. Liu, H. Wang, W. Cheung, and X. Xie, On-demand e-supply chain integration: A multi-agent constraint-based approach, *Expert Systems with Applications*, vol. 34, no. 4, pp. 2683-2692, 2008.
- [30] M. Wooldridge, *An Introduction to Multiagent systems*. Hoboken: John Wiley and Sons, 2009.
- [31] M. Yokoo and K. Hiramaya, Algorithms for distributed constraint satisfaction: A review, *Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 2, pp. 185-207, 2000.
- [32] D. Zeng and K. Sycara, Benefits of learning in negotiation, in *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference*, Providence, Rhode Island, 1997, pp. 36-42.
- [33] D. Zeng and K. Sycara, Bayesian learning in negotiation, *International Journal of Human-Computer Studies*, vol. 48, no. 1, pp. 125-141, 1998.