An Intelligent Agent for Bilateral Negotiation with Unknown Opponents in Continuous-Time Domains

SIQI CHEN and GERHARD WEISS, Department of Knowledge Engineering, Maastricht University

Automated negotiation among self-interested autonomous agents has gained tremendous attention due to the diversity of its broad range of potential real-world applications. This article deals with a prominent type of such negotiations, namely, multiissue negotiation that runs under continuous-time constraints and in which the negotiating agents have no prior knowledge about their opponents' preferences and strategies. A negotiation strategy called *Dragon* is described that employs sparse pseudoinput Gaussian processes. Specifically, *Dragon* enables an agent (1) to precisely model the behavior of its opponents with comparably low computational load and (2) to make decisions effectively and adaptively in very complex negotiation settings. Extensive experimental results, based on a number of negotiation scenarios and state-of-the-art negotiating agents from Automated Negotiating Agents Competitions, are provided. Moreover, the robustness of our strategy is evaluated through both empirical game-theoretic and spatial evolutionary game-theoretic analysis.

Categories and Subject Descriptors: I.2.11 [**Artificial Intelligence**]: Distribute Artificial Intelligence—*Intelligent agents, multiagent systems*

General Terms: Algorithms, Performance, Experimentation, Theory

Additional Key Words and Phrases: Multiagent systems, automated multi-issue negotiation, opponent modeling, sparse pseudoinput Gaussian processes, game theory

ACM Reference Format:
Siqi Chen and Gerhard Weiss. 2014. An intelligent agent for bilateral negotiation with unknown opponents in continuous-time domains. ACM Trans. Autonom. Adapt. Syst. 9, 3, Article 16 (October 2014), 24 pages.
DOI: http://dx.doi.org/10.1145/2629577

1. INTRODUCTION

Automated negotiation, as a fundamental and powerful mechanism for managing interaction among computational agents [Jennings et al. 2001], has become a subject of central interest in distributed artificial intelligence over the past years. The main reason behind it is twofold. First, it can support and facilitate human negotiators in reaching more efficient outcomes by compensating for the limited computational abilities of humans, especially in complicated negotiations. Second, its spectrum of potential

This article is a substantially extended version of our AAMAS main track paper [Chen et al. 2013a]. The extension concerns both the description and evaluation of *Dragon*. With regard to the description, all relevant aspects of our approach are shown in detail. With regard to the evaluation, the range of considered negotiation scenarios and negotiating opponents has been extended significantly. Moreover, the impact of different levels of domain competitiveness, as well as the outcome space on the performance of *Dragon*, is investigated in depth. Furthermore, a comprehensive robustness analysis of *Dragon* is provided, employing empirical game theory as well as spatial evolutionary game theory.

Authors' address: S. Chen (corresponding author) and G. Weiss, Department of Knowledge Engineering, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; emails: {siqi.chen, gerhard.weiss}@maastrichtuniversity.nl.

DOI: http://dx.doi.org/10.1145/2629577

applications in industrial and commercial domains is very broad [Adhau et al. 2012; Moosmayer et al. 2013; Sanchez-Anguix et al. 2013]. Various forms of negotiation can be distinguished [Lopes et al. 2008]. This article deals with bilateral multi-issue negotiation. In this setting, two agents negotiate with the goal to agree on a profitable contract for a product or service, where the contract consists of multiple issues that are of conflictive importance to the negotiators, such as price, delivery time, quantity, and quality. To meet common requirements of real-world negotiation applications, the following four assumptions are made. First, the negotiating agents have no prior information about their opponents—neither about their preferences (e.g., their preferences over issues or issue value ordering) nor about their negotiation strategies. Second, negotiation time is limited—that is, there is a deadline by which an agreement is to be achieved. Furthermore, the deadline is based on the amount of real time that has elapsed. Third, the utility of the negotiation outcome decreases over time—in other words, there is a discounting factor that reduces an agent's payoff over time. Fourth, each agent has a private reservation value below which an opponent's offer is not accepted and an agent obtains its reservation value even if no agreement is reached during negotiation. Negotiation settings that fulfill these assumptions are of relevance to a wide range of practical applications and are also common to many human–human negotiation scenarios. Researchers in the community consider negotiations happening in such settings as complex negotiations to differentiate those in a simple form (single-issue negotiations, negotiations with prior opponent information, etc).

Appropriately modeling an opponent's behavior is known to be a major factor in successful negotiation (e.g., Zeng and Sycara [1997] and Hindriks et al. [2009a]). This modeling, however, is very challenging because negotiators generally are not open about their true preferences and strategies, as they want to avoid exploitation of this information (e.g., Raiffa [1982] and Coehoorn and Jennings [2004]). Existing approaches to tackle this challenge suffer from two main shortcomings: oversimplification and computational complexity. Oversimplification means that simplifying assumptions are made that underestimate an opponent's behavioral space. This typically results in simpler-to-calculate, yet too imprecise, opponent models. Computational complexity means that opponent models, especially those that do not suffer from oversimplification, are too complex to be calculated efficiently in real time.

This work tackles these problems and makes several contributions. The primary contribution is the proposition of a novel negotiation strategy—*Dragon*—for the negotiating agent. This strategy makes use of sparse pseudoinput Gaussian processes (SPGPs) to (1) relax the modeling assumptions of other approaches by employing a nonparametric functional prior, making it capable of modeling highly complex opponent models, and (2) reduce the computation complexity of learning in such a nonparametric setting. The second contribution is the proposition of a new adaptive decision-making strategy. The main advantages of this new decision-making method are (1) allowing the agent to determine the best possible concession degree of the opponent and (2) avoiding the problems related to *irrational concession*.[1] The third contribution is to introduce spatial evolutionary game theory to analyze the strategy robustness in a various ranges of agent interaction.

The remainder of this article is structured as follows. Section 2 provides the reader with background knowledge needed to understand the remainder of the article. Then, the technicalities of *Dragon* are explained in Section 3. An in-depth analysis of

[1] When playing against a tough opponent, a concession-making strategy that is too sensitive to opponent behavior inherently tends to suggest (far) more concession than necessary to reach an agreement with such an opponent. We refer to this phenomenon as irrational concession (making); further considerations on this issue are provided in Section 3.2.

the approach in the context of tournament performance is given in Section 4, and Section 5 launches an innovative scheme to examine the robustness of the negotiation strategy. Section 6 overviews important related work and pinpoints major deficiencies for complex negotiations. Section 7 concludes and identifies interesting future research directions.

2. PRELIMINARIES

In this section, we provide the reader with the background knowledge needed to understand the remainder of the article. First, the negotiation model in which the agents operate is explained. Second, the regression framework, including Gaussian processes (GPs) and SPGPs, is detailed.

2.1. Negotiation Model

In the agents field, a large number of works [Faratin et al. 1998, 2002; Williams et al. 2011; Chen and Weiss 2012; Chen et al. 2013b] adopt the alternating offers protocol formalized in Rubinstein [1982] for bilateral negotiation. During negotiation, the two parties in turn make offers and counteroffers under the alternating offers protocol. This exchange process can be in a form of either discrete or continuous time. Our work is based on the continuous-time version (as this is an element of complex negotiation).

Another key aspect of a negotiation model is to specify the way in which an agent evaluates the utility of an offer to her. We take a common view from the existing literature [Williams et al. 2011; Hao and Leung 2012; Baarslag et al. 2013; Chen et al. 2013a]. Formally, let i be an agent, j be an issue, and k represent the choice of the j^{th} issue. We define the value of issue j as v_{jk}. The lowest payoff of an agent for a negotiation is called the *reservation value* (θ). Further, w^i_j denotes the weighting preference that agent i assigns to issue j. The weights of agent i over the issues are normalized summing to one (i.e., $\sum_{j=1}^{n}(w^i_j) = 1$). These preferences are determined by the interests of the parties to which the agents are on behalf. An offer, O, is a vector of values v_{jk} for each of the issues j. The utility of an offer for agent i is defined as

$$U^i(O) = \sum_{j=1}^{n} \left(w^i_j \cdot V^i_j(v_{jk})\right), \tag{1}$$

where V^i_j is the evaluation function of agent i, mapping the value of an issue j to a real number.

Each agent makes, in turn, an offer in form of a contract proposal to express its demand, although it is free to leave the negotiation table for its own sake, following the notion of negotiation round from Rubinstein [1982]. In this work, we refer to negotiation rounds as the number of offer exchanges between the two parties. A negotiation continues until one party accepts a proposal from the other party, or one party's breaking off, or no more rounds are allowed due to time-out. If no agreement is reached at the end, the provided disagreement solution then takes effect, where each party obtains its own reservation value (θ). This also holds for the case in which one side withdraws from the negotiation in advance.

Moreover, we define a discounting factor δ ($\delta \in [0, 1]$) to capture the discounting effect in negotiation as follows:

$$D_\delta(U, t) = U \cdot \delta^t, \tag{2}$$

where $D_\delta(U, t)$ means the discounted utility at time t, U is the (original) utility, and t is the standardized time (i.e., $t \in [0, 1]$). From the preceding equation, it is clear that the longer it takes for agents to come to an agreement, the lower the utility they can achieve.[2]

2.2. Gaussian Processes

In the field of machine learning, GPs are one of the well-known, nonlinear, nonparametric regression techniques. These models have been successfully applied in negotiation settings by Williams et al. [2011]. Although GPs are a powerful form of function approximators, they suffer from computational problems once dealing with large datasets. GPs present a good candidate for opponent modeling as long as the computational complexity is reduced. To address this accuracy-computation dilemma, *Dragon* proposes a novel learning scheme based on SPGPs. These models (detailed in Section 2.2.2) are able to achieve similar modeling accuracy to GPs but with much less computational effort.

2.2.1. Normal Gaussian Processes. GPs are a form of nonparametric regression techniques. Following the notation of Rasmussen and Williams [2006], given a data set $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{n}$, where $\mathbf{x} \in \mathbb{R}^d$ is the input vector, $y \in \mathbb{R}$ is the output vector, and n is the number of available data points when a function is sampled according to a GP, we write $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $m(\mathbf{x})$ is the mean function and $k(\mathbf{x}, \mathbf{x}')$ the covariance function, fully specifying a GP. Learning in a GP setting involves maximizing the marginal likelihood of Equation (3),

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T\left(\mathbf{K} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K} + \sigma_n^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi, \tag{3}$$

where $\mathbf{y} \in \mathbb{R}^{m\times 1}$ is the vector of all collected outputs, $\mathbf{X} \in \mathbb{R}^{m\times d}$ is the matrix of the dataset inputs, and $\mathbf{K} \in \mathbb{R}^{m\times m}$ is the covariance matrix with $|.|$ representing the determinant. Due to space constraints, we refer the interested reader to Rasmussen and Williams [2006] for a thorough discussion of the topic. To fit the hyperparameters that best suit the available dataset, we need to maximize the marginal likelihood function of Equation (3) with respect to $\mathbf{\Theta}$, the vector of all hyperparameters. Typically, this maximization requires the computation of the derivatives of Equation (3) with respect to $\mathbf{\Theta}$. These derivatives are then used in a gradient-based algorithm to perform the updates. Namely, the update is performed using the following equations:

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|\mathbf{X}, \mathbf{\Theta}) &= \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\right)\\ &= \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \mathbf{K}^{-1})\frac{\partial\mathbf{K}}{\partial\theta_j}\right) \text{ with } \boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{y}.\end{aligned} \tag{4}$$

The problem with GPs is that maximizing Equation (3) is computationally expensive due to the inversion of the covariance matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, where n is the number of data points. The update in each step of the gradient-based optimization algorithm incurs the inversion cost of $\mathcal{O}(n^3)$. Since the covariance matrix is parameterized by the hyperparameters $\mathbf{\Theta}$, this inversion needs to be computed at each step of the gradient-based algorithm as the values of $\mathbf{\Theta}$ are updated.

It is for this specific reason that we employ a fast and more efficient learning technique (i.e., SPGPs). The most interesting feature of SPGPs is that these approximators are capable of attaining very close accuracy in both learning and prediction to normal GPs with only a fraction of the computation cost, primarily because learning is parameterized by a small number of pseudoinputs that are automatically fitted depending

[2] Note that the discounting effect is applied to offer both the utility and the reservation value.

on the variation of the sought function. This property makes them extremely suitable to the negotiation domain, where a complex and low cost function approximation framework is highly demanded. The technicalities of SPGPs are described next.

2.2.2. Sparse Pseudoinput Gaussian Processes. As mentioned previously, normal GPs are computationally expensive to learn, especially in an online setting. SPGPs aim to reduce the complexity of GPs in both learning and prediction. The idea is to parametrize the regression model with the so-called pseudoinputs. The location of these inputs is iteratively fitted by maximizing a new kind of marginal likelihood. Interestingly, using only a small amount of pseudoinputs, SPGPs are capable of attaining very similar fitting and prediction results to normal GPs. To clarify, the idea is to parametrize the model by $M << n$ pseudoinput points while still preserving the full Bayesian framework. This leads to the parametrization of the covariance function by the location of $M <<< n$ pseudoinputs. These are then fitted in addition to the hyperparameters to maximize the following new marginal likelihood:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \boldsymbol{\Theta}) &= \int p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{f}})p(\bar{\mathbf{f}}|\bar{\mathbf{X}})d\bar{\mathbf{f}} \\ &= \mathcal{N}\big(\mathbf{y}|\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \Lambda + \sigma^2\mathbf{I}\big), \end{aligned} \tag{5}$$

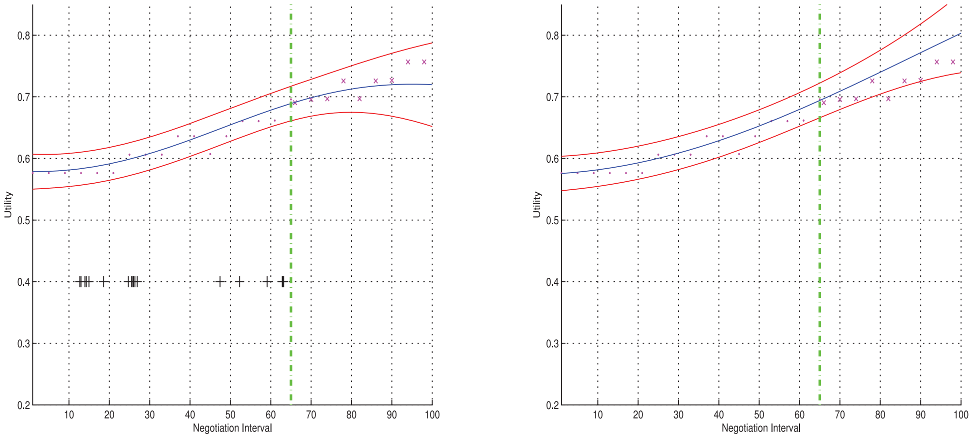
where $\bar{\mathbf{X}}$ is the matrix formed by the pseudoinputs with $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}\}_{m=1}^{M}$. $\mathbf{K}_{NM}$ is the covariance matrix formed by the pseudo and the real inputs as $\mathbf{K}_{MN} = k(\bar{\mathbf{x}}_m, \mathbf{x}_n)$, with $k(.,.)$ being the covariance kernel. $\mathbf{K}_M^{-1}$ is the inverse of the covariance matrix formed among the pseudoinputs with $\mathbf{K}_M = k(\bar{\mathbf{x}}_m, \bar{\mathbf{x}}_m)$. $\boldsymbol{\Lambda}$ is a diagonal matrix having the diagonal entries of $\lambda_n = k_{nn} - \mathbf{k}_n^T\mathbf{K}_M^{-1}\mathbf{k}_n$. The noise variance and the identity matrix are represented by σ and $\mathbf{I}$, respectively.

Results in Snelson and Ghahramani [2006] show a complexity reduction in the training cost (i.e., the cost of finding the parameters of the covariance matrix) to $\mathcal{O}(M^2N)$ and in the prediction cost (i.e., prediction on a new set of inputs) to $\mathcal{O}(M^2)$. The results further demonstrate that SPGPs can fully match normal GPs with small M (i.e., few pseudoinputs), successfully producing very sparse solutions. A full mathematical treatment may be found elsewhere [Snelson and Ghahramani 2006].

To understand the usefulness and powerfulness of SPGPs in the context of automated negotiation, consider a classical negotiation scenario where two agents negotiate with each other about what kinds of sports to do, what time to do them, the intensity, and the price. The task of the prediction model is to forecast the utilities that an agent will receive in the future. With only $M = 20$, SPGPs were able to attain very similar results to normal GPs as shown in Figure 1. It is clear that both learned functions follow a very similar increasing trend. Predictions made at 65 negotiation intervals also show similar predicted values in both cases. The black crosses in Figure 1(a) represent the location of the fitted pseudoinputs. It is also clear that these pseudoinputs were mostly located in critical ranges of the function.

3. PROPOSED METHOD

The overall *Dragon* strategy is shown in Algorithm 1. *Dragon* consists of three functional components that are essential and vital for the agent to operate successfully. First, the opponent-modeling component is described. It adopts a nonparametric and computationally efficient regression technique to approximate the opponent's model. This allows the agent to have more accurate estimates that are used to predict the future behavior of the opponent. After having learned the opponent's model, the concession-making component determines the optimal concession behavior using a novel adaptive decision-making strategy that automatically avoids the problem of



(a) Illustration of the predictive power of SPGPs on a toy experiment.

(b) Illustration of the predictive power of GPs

Fig. 1. The curves in the middle (blue) represent the mean of the approximated function, and the solid lines (red) around them represent the variance. The black crosses in Figure 1(a) show the locations of the pseudoinputs, and the vertical dash-dot lines (green) in both figures denote the time at which the prediction is taking place—that is, they separate historic and predicted data points.

ALGORITHM 1: The *Dragon* approach. Let t_c be the current time, δ the time discounting factor, and t_{max} the deadline of negotiation. O_{opp} is the latest offer of the opponent, and O_{own} is a new offer to be proposed by *Dragon*. χ is the time series including the maximum utilities over intervals. ϱ is the lead time for prediction (i.e., the time between the release of the prediction and the end of the period being predicted), and E_δ denotes the discounted expected utility of incoming counteroffers. E_{low} is the lowest expectation to negotiation, ρ is the compromise point (i.e., the time at which the agent starts to concede), and R is the dynamic conservative expectation function that carefully suggests target utilities to avoid irrational concession. u'_c is the target utility at time t_c.

```
Require: $R, \delta, \varrho, t_{max}$
while $t_c <= t_{max}$ do
    $O_{opp} \Leftarrow receiveMessage$;
    $recordOffers(t_c, O_{opp})$;
    if $TimeToUpdate(t_c)$ then
        $\chi \Leftarrow preprocessData(t_c)$
        $E_\delta \Leftarrow Predict(\chi, \varrho)$;
        $(E_{low}, \rho) \Leftarrow updateParas(t_c)$;
        $R \Leftarrow (E_{low}, \rho)$;
    end if
    $u'_c = getTargetUtility(t_c, E_\delta, \delta, R)$;
    if $isAcceptable(u'_c, O_{opp}, t_c, \delta)$ then
        $accept(O_{opp})$;
    else
        $checkTermination()$;
        $O_{own} \Leftarrow constructOffer(u'_c)$;
        $proposeNewBid(O_{own})$;
    end if
end while
```

irrational concession. Finally, the third and last stage of *Dragon* (i.e., the responding component) responds to those counteroffers and determines the time point at which the negotiation session terminates. Next, each of the preceding components is detailed.

3.1. Opponent-Modeling Component

Modeling the opponent's behavior is done by the first component of *Dragon*. It adopts the SPGPs (detailed in Section 2.2.2) to accurately and efficiently learn the opponent's model. This process of opponent modeling corresponds to the lines 2 through 7 in Algorithm 1. Namely, upon receiving a new proposal from the opponent at the time t_c, the agent records the time stamp t_c and the utility $U(O_{opp})$ that this bid offers according to our agent's own utility function. However, in the setting of multi-issue negotiations with the two negotiation partners having different utility functions, a small change in utility of the opponent may result in a large utility variation for our agent. Making concession at some steps does not necessarily mean that the opponent will increase its concession in the coming rounds. This tends to cause a major misinterpretation of the opponent's behavior if the agent deals with every single counteroffer. Therefore, and to reduce that negative impact, the whole negotiation is divided into a number of equal intervals (denoted as ζ). As the utility is measured in terms of our agent's own utility function, this utility may vary significantly in a given interval. Using maximum offers can alleviate the effect of noise on the prediction model. It is worth pointing out that although minimum or average offers are also useful with respect to noise reduction, they may to some extent lower an agent's expectation on the opponent concession, thereby causing the effect of irrational concession (see Section 3.2). The maximum utilities at each interval with the corresponding time stamps are then provided as inputs to the SPGPs. As SPGPs are more computationally efficient compared to normal GPs, the number of intervals here can be much more (by factors of hundreds) than those used in Williams et al. [2011]. This automatically leads our agent to have more accurate predictions of the future opponent's behavior compared to that work.

After learning a suitable model, SPGPs forecast the future behavior of the opponent as shown in line 7 of Algorithm 1. *Dragon* keeps track of the expected discounted utility based on the predictive distribution at a new input $t_\star$, which is given by

$$p(u_*|t_\star, \mathcal{D}, \bar{\mathbf{X}}) = \int p(u_\star|t_\star, \bar{\mathbf{X}}, \bar{\mathbf{f}})p(\bar{\mathbf{f}}|\mathcal{D}, \bar{\mathbf{X}})d\bar{\mathbf{f}} = \mathcal{N}\big(u_\star|\mu_\star, \sigma_*^2\big), \tag{6}$$

where

$$\mu_\star = \mathbf{k}_\star^T \mathbf{Q}_M^{-1}(\mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1}u$$
$$\sigma_\star^2 = \mathbf{K}_{\star\star} - \mathbf{k}_\star^T\big(\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1}\big)\mathbf{k}_\star + \sigma^2$$
$$\mathbf{Q}_M = \mathbf{K}_M + \mathbf{K}_{MN}(\mathbf{\Lambda} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{NM}.$$

With the given probability distribution over future utilities and the effect of the discounting factor, the expected utility $E_\delta(t_\star)$ is then formulated by

$$E_\delta(t_\star) = \frac{1}{C}\int_{-\infty}^{+\infty} D_\delta(u \cdot p(u; \mu_\star, \sigma_\star), t_\star)du, \tag{7}$$

where $\mu_\star$ and $\sigma_\star$ are the mean and standard deviation at time $t_\star$ and δ is the discounting factor.

In contrast to the work of Williams et al. [2011], we adopt a mathematically valid approach to preserve a probability distribution by introducing C, the normalizing constant, rather than truncating the probability distribution between $[0, 1]$. The latter does not generate a valid probability density function anymore, whereas ours guarantees this.

3.2. Concession-Making Component

Using the approximated model, the concession-making component aims at setting an appropriate concession degree. To deal with uncertainty in negotiation, the concession-making component primarily takes two factors into account. The first relates to the prediction of the opponent's future compromise, whereas the second builds on the agent's own expectation to the negotiation outcome.

As mentioned earlier, *Dragon* makes use of SPGPs to predict the future moves of an opponent following Equation (7). Although successful, the prediction results are sometimes rather overpessimistic due to Boulware behavior [Faratin et al. 1998] of opponents. Such "sophisticated and tough" opponents attempt to avoid (or to postpone as far as possible) making concessions during the bargaining process. In this case, an opponent's offers can easily result in a misleading, too low expectation of an agent about the utility that the opponent will offer in the future. As a consequence, an adaptive agent can be inclined to react irrationally in the sense that it compromises either too much or too early or even both. To solve the problem of irrational concession, *Dragon* employs a dynamic conservative expectation function $R(t)$. Informally, it is a "dynamic conservative expectation function that carefully suggests utilities." Overall, $R(t)$ is sensitive to the remaining time because an agent is under more pressure to settle a negotiation as time is vanishing. Since smaller values of the discounting factor (δ) force rational agents into reaching agreements earlier (otherwise, the final payoff gets more discounted), $R(t)$ is inversely proportional to δ. $R(t)$ further takes the lowest expectation (E_{low}) as its minimum value. Formally, $R(t)$ is written as

$$R(t) = E_{low} + \left(1 - t^{\frac{1}{(1-\rho)^{\beta}}}\right)\left(u^{P}_{max} - E_{low}\right)\cos\left(\frac{1-\delta}{\omega}\right), \tag{8}$$

where β is the concession factor affecting the concession rate, u^{P}_{max} is the maximum utility of the given preference P in a domain, ω is the weight that reflects the impact of the discounting factor to the concession degree, and ρ is the compromise point.

There are two important variables considered in $R(t)$: the compromise point and the lowest expectation. In what follows, we motivate the need for them and detail the technicalities for defining them.

Since the design objective of the negotiation strategy/agent is to maximize its profits, it is highly demanded to exploit the opponent. But the trade-off between exploitation and compromise is also of major importance. To clarify, suppose that the agent never makes any concession, probably no agreement would be reached, or the opponent might even break off somewhere within the negotiation process. Thus, ρ is used to adaptively adjust the time at which *Dragon* should stop exploiting the opponent and rather start to compromise. To this end, the value of ρ should increase with the increasing ratio between the number of new solutions and the total solutions proposed by the other party. This is because a high ratio of new solutions tends to be a good indicator of the opponent's cooperation. Thus, we introduce γ_t to represent the ratio of new counteroffers over the past 10 intervals up to t. The observation of new counteroffers cannot guarantee the concession by the other party (e.g., these new offers could just be the offers with high utility for the opponent but low utility for our agent). Therefore, the effect of γ_t is influenced by the maximum concession until t (λ_t),[3] leading to the following:

$$\rho = 1 - \left(1 - \gamma_t^{(1-\lambda_t)^{\frac{\alpha}{\delta}}}\right)t, \tag{9}$$

where α is the parameter controlling the influence of λ_t.

[3] The maximum concession (until time point t) is the maximum compromise among all counteroffers given by the opponent until time t.

The other variable needed to define $R(t)$ is E_{low}, which represents the lowest expectation to a negotiation session. Formally, E_{low} is defined as

$$E_{low} = \begin{cases} \theta & if\ \theta \geq {}_{maxU}(E_{\delta=1}(0, t_l)) \\ {}_{maxU}(E_{\delta=1}(0, t_l)) & otherwise, \end{cases} \tag{10}$$

where θ is the reservation value specified by the preference, $maxU$ returns the maximal utility from counteroffers, and t_l is the last time the update was carried out.

Based on the preceding definitions, the decision of how to counter the opponent is made as follows. If the future expectation obtained from $E_\delta(t)$ is optimistic (i.e., there exists an interval $\{T | T \neq \varnothing, T \subseteq [t_c, t_s]\}$), that is,

$$E_\delta(t) \geq D_\delta(R(t), t), \quad t \in T, \tag{11}$$

with t_s being the end point of the prediction and $t_s \leq t_{max}$, then the time $\hat{t}$ at which the maximum expectation $\hat{u}$ is reached is set according to

$$\hat{t} = \underset{t \in T}{argmax}\ E_\delta(t). \tag{12}$$

And $\hat{u}$ is defined as

$$\hat{u} = E_\delta(\hat{t}). \tag{13}$$

Conversely, in the pessimistic case where the estimated opponent concession is below the agent's expectations, we define the probability of accepting the best possible utility, φ, to be inversely proportional to the minimum difference between $E_\delta(t)$, $D_\delta(R(t), t)$, and the discounting factor, as follows:

$$\varphi = 1 - \sqrt[5]{\frac{\delta \cdot (D_\delta(R(t_v), t_v) - E_\delta(t_v))}{D_\delta\left(u^P_{max}, t_v\right)}}, \quad t_v \in [t_c, t_s], \tag{14}$$

where t_v is given by

$$t_v = \underset{t \in [t_c, t_s]}{argmin}\,(|E_\delta(t) - D_\delta(R, t)|). \tag{15}$$

According to the probability φ, the best possible outcome in the "pessimistic" scenario is chosen as the target utility. The rationale here is that if the agent rejects the "locally optimal" counteroffer, it may lose the opportunity to reach a fairly good agreement earlier.

In the acceptance case, $\hat{u}$ and $\hat{t}$ are defined as $E_\delta(t_v)$ and t_v, respectively. Otherwise, $\hat{u}$ is defined as -1, meaning that it does not have an effect, and $R(t_c)$ is used for the target utility u'_c. When the agent expects to achieve a better outcome (see Equation (11)), the optimal expected utility $\hat{u}$ is chosen as its target utility (see Equations (12) and (13)).

Obviously, conceding immediately to $\hat{u}$ is not rational for the agent when $u_l \geq \hat{u}$ (u_l is the utility of last bid before t_l). Neither is it to shift to $\hat{u}$ without delay if $u_l < \hat{u}$, especially because the predication may be not completely accurate. To deal with this, *Dragon* simply concedes linearly. More precisely, the concession rate is dynamically adjusted to be able to "grasp" every chance to maximize profit. Overall, the process to set u'_c is shown in line 11, which is calculated as follows:

$$u'_c = \begin{cases} R(t_c) & if\ \hat{u} = -1 \\ \hat{u} + (u_l - \hat{u})\frac{t_c - \hat{t}}{t_l - \hat{t}} & otherwise. \end{cases} \tag{16}$$

3.3. Responding Component

The responding component is the last component of the *Dragon* strategy and corresponds to lines 12 through 18 of Algorithm 1. After the expected utility u'_c has been determined, the agent needs to examine one of two conditions in response to the opponent. According to the first condition, the agent has to validate whether the utility of the counteroffer $U(O_{opp})$ is better than u'_c, whereas according to the second condition, the agent has to determine whether it had already proposed this offer (i.e., the opponent's counteroffer) earlier in the negotiation process. If either one of these two conditions is satisfied, the agent accepts it and terminates the session as shown in line 12.

On the other hand, if none of them are met, the agent proposes a new offer depending on an ϵ-greedy strategy—that is, to select either a greedy action (i.e., exploit) with 1-ϵ probability or to select a random action with an ϵ probability, where $0 \leq \epsilon \leq 1$. The greedy action is determined based on a frequency analysis. Although simple, such a method has been successfully applied by some state-of-the-art negotiating agents, like Hardheaded and CUHKAgent (refer to Baarslag et al. [2013] and Hao and Leung [2012]). In the present work, *Dragon* considers that the opponent is rational. More precisely, *Dragon* assumes that the sequence of counteroffers is in line with a decreasing order of satisfaction. Thus, for a value of an issue j, the more frequent and earlier it is proposed by the negotiation partner, the more contribution it makes to the opponent's overall utility.

Formally, let $F(\cdot)$ be the frequency function defined as

$$F^n(v_{jk}) = F^{n-1}(v_{jk}) + (1-t)^{\psi} \cdot g(v_{jk}), \tag{17}$$

where the superscript of $F(\cdot)$ indicates the number of negotiation rounds, ψ is the parameter reflecting the discounting effect of time, and $g(\cdot)$ is the two-valued function, whose return is 1 if the specific issue value (i.e., v_{jk}) appears in the counteroffer and 0 otherwise.

With a probability $1 - \epsilon$, *Dragon* then picks the offer whose issue values have the maximal sum of frequencies according to the frequency function. In the case of the random action, *Dragon* constructs a new offer that has an utility within some range around u'_c. The main motivation behind this choice is twofold: (1) it is possible, in multi-issue negotiations, to generate a number of offers whose utilities are the same or very similar to the offering agent, with granting the opposing negotiator different utilities, and (2) it is sometimes not possible to make an offer whose utility is exactly equivalent to u'_c. Thus, it is reasonable that an agent selects an offer whose utility is in the narrow range $[(1-0.005)u'_c, (1+0.005)u'_c]$. If no such solution can be found, the agent repeats the latest bid again in the next round.

One additional step is needed to cope with terminating the negotiation in advance when $\theta > 0$ and $\delta \neq 1$, because in such a case, failing to reach an agreement leads to a lower payoff (not that the reservation value is also discounted), which is more likely to happen in a negotiation against a tough opponent. The responding component investigates whether the maximum expectation obtained from SPGPs is larger than θ. If that is true, the agent could expect to gain a better outcome than what the disagreement solution generates. Otherwise, the agent breaks off the bargaining if the opponent does not make any compromise in the incoming 10 intervals.

4. EMPIRICAL EVALUATIONS

The performance of *Dragon* is evaluated by means of the official platform of the ANAC—GENIUS [Hindriks et al. 2009b]. This simulation environment helps to facilitate comparing different agents across a variety of application domains under real-time constraints. Section 4.1 describes the experimental technicalities including the implementation details of *Dragon* (referred to as *DragonAgent*). The competition results are reported in Section 4.2. Moreover, Section 4.3 summarizes the performance of negotiating agents.

Table I. Overview of Test Domains in Ascending Order of Competitiveness

Domain	*Issues*	*Domain Size*	*Competitiveness*
DogChoosing	5	270	0.051
Kitchen	6	15,625	0.063
Animal	5	1,152	0.110
Acquisition	5	384	0.117
Icecream	4	720	0.148
Laptop	3	27	0.160
Planes	3	27	0.165
Outfit	4	128	0.198
Camera	6	3,600	0.219
SmartPhone	6	12,000	0.237
HouseKeeping	5	384	0.281
Wholesaler	7	56,700	0.308
DefensiveCharms	3	36	0.322
Lunch	6	3,840	0.420
Coffee	3	112	0.486
Ultimatum	2	9	0.545
Fifty2013	1	11	0.707
NiceOrDie	1	3	0.840

4.1. Experimental Setup

To assess *Dragon* in a highly competitive setting, we set up two tournaments where its implementation (*DragonAgent*) plays against the well-performing agents of the 2010–2012 ANAC,[4] excluding those best ones (primary tournament), and the winners of ANACs (advanced tournament). Moreover, the negotiations are conducted in the whole set of domains created for ANAC 2013. Such a setting avoids advantageous bias, as they are unknown to any participant. To capture the influence of the discounting factor δ and the reservation value ϑ on the agents' performance, different values for these two parameters are considered. Thus, the experiments are conducted for each domain with three discounting factors (i.e., $\delta = \{0.5, 0.75, 1.0\}$) and three reservation values (i.e., $\vartheta = \{0, 0.25, 0.5\}$), which result in nine (3×3) different scenarios for every domain.

All domains used in this work are overviewed in Table I, where issue refers to the number of items under negotiation, competitiveness represents the minimum distance from all points in the outcome space of a domain to the point leading to a complete satisfaction for both sides (note that such an ideal solution may not be always available), and the domain size means the scale of the outcome space of a domain (i.e., the number of possible agreements). For the results to be statistically significant, each scenario is repeated 20 times.

DragonAgent simply sets ζ, which is the number of equal intervals into which the agent divides the whole negotiation, to 180. Note that ζ can be larger than the value; we use it for the purpose of convenience and robustness. The lead time ϱ is limited to 25 intervals, and the parameter α for compromise point is set to 0.1. The parameters for the expectation function R are set to $\beta = 1.5$ and $\omega = 1.2$. Note that this set of parameters is chosen according to our experience without systematically optimizing it. In our experiments, Dragon performed well and was very robust for a quite broad range of parameters.

[4] Because of the significant change with respect to the competition rules, the agents of ANAC 2013 are allowed to use historical knowledge. Introduction of them in the experiments will cause unfairness to others, and this new feature moreover does not meet the requirements of complex negotiation.

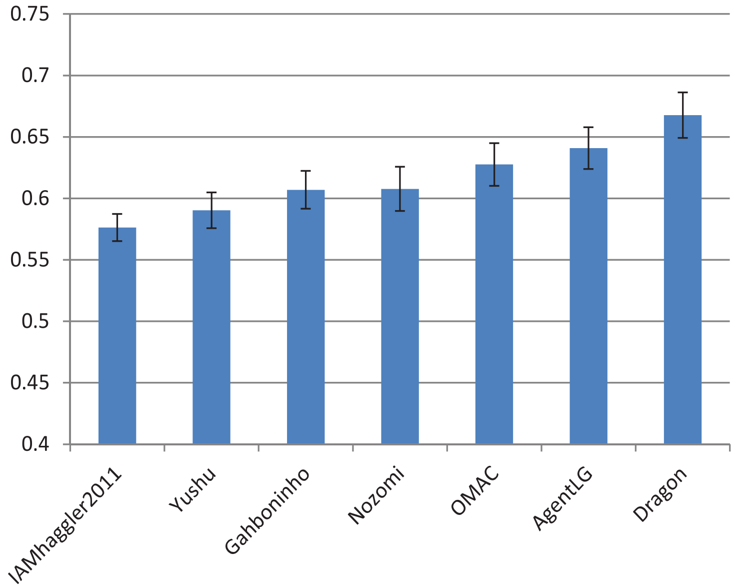


Fig. 2. Average normalized score of negotiating agents in the primary tournament. The error bar indicates a single standard deviation.

4.2. Experimental Results

4.2.1. Results of Primary Tournament. The results achieved by the agents in the primary tournament are shown in Figure 2, where a total number of 162 scenarios are considered, resulting from 18 domains with nine different combinations of discounting factors and reservation values. As depicted in the figure, *DragonAgent* demonstrated excellent performance against a variety of opponents and was the best-performing agent from the perspective of the mean normalized[5] score. More precisely, *DragonAgent* achieved a score 10% higher than the mean normalized score of opponents. AgentLG and OMAC (the second and third best agents of ANAC 2012) followed our agent and on average obtained a score of 0.634, which is 5% below that of *DragonAgent*. Gahboninho and IAMhaggler2011, the second and third in ANAC 2011, presented a lower performance, at around 87% of ours, and there was a similar performance difference for the agents from ANAC 2010 (i.e., Nozomi and Yushu).

4.2.2. Results of Advanced Tournament. Figure 3 gives the results of the advanced tournament where only the strongest opponents—the winners of each year's competition and an additional agent EMAR [Chen and Weiss 2013]—are included. This setting is much more challenging than the former. Due to this fact, the performance of *DragonAgent* to some extent dropped. It, however, was still ranked first among other contenders with a score of 0.627, leading the mean performance of others by a margin of 7.5%. In addition, its performance experienced the smallest variance. The second best agent of the advanced tournament was CUHKAgent (the 2012 winner). EMAR lagged behind CUHKAgent with a tiny difference (within 0.005) and finished third. Then, Hardheaded (the 2011 winner) and Agent_K (the 2010 winner) took the fourth to fifth place in sequence. To sum up, *DragonAgent* managed to outperform other best agents in this very competitive setting with a notable advantage.

[5] For convenience of comparing performance across domains, normalization is adopted for the results of all scenarios and done in the standard way, using the maximum and minimum raw score obtained by all agents.

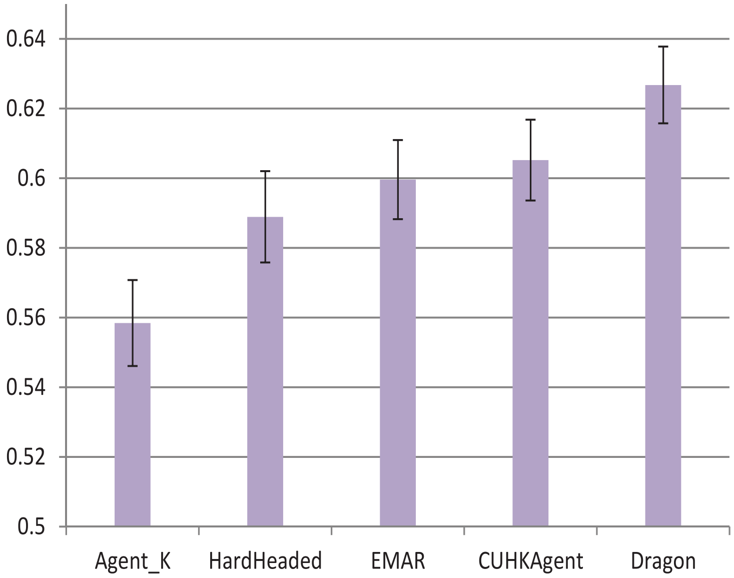


Fig. 3. Average normalized score of negotiating agents in the advanced tournament. The error bar indicates a single standard deviation.

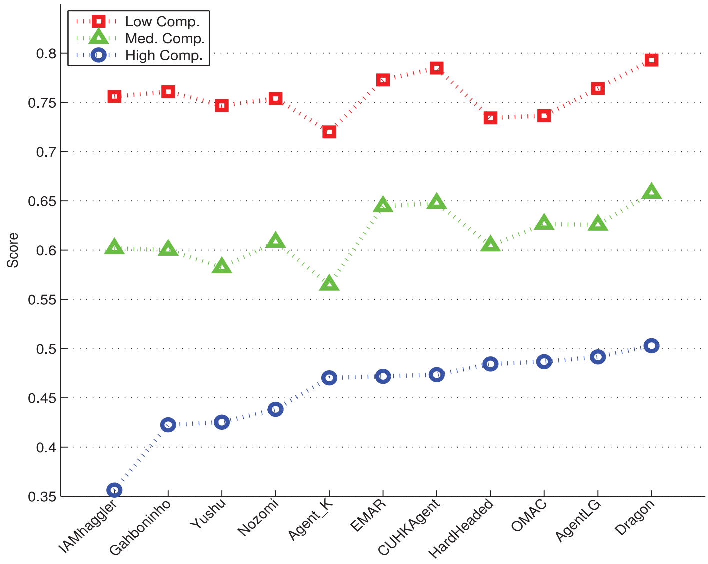


Fig. 4. Comparing performance of the agents in different levels of competitiveness.

4.2.3. Impact of Different Competitiveness. For evaluating how competitiveness affects agents' performance, the domains are classified into three groups (each with the same number of domains) to represent the different level of competitiveness as follows: low (competitiveness ≤ 0.160), medium ($0.160 <$ competitiveness < 0.32), high (competitiveness ≥ 0.32). Figure 4 demonstrates the comparison of all agents' performance under low, medium, and high domain competitiveness. The influence of the discounting factor and the reservation value has been already taken into consideration in the

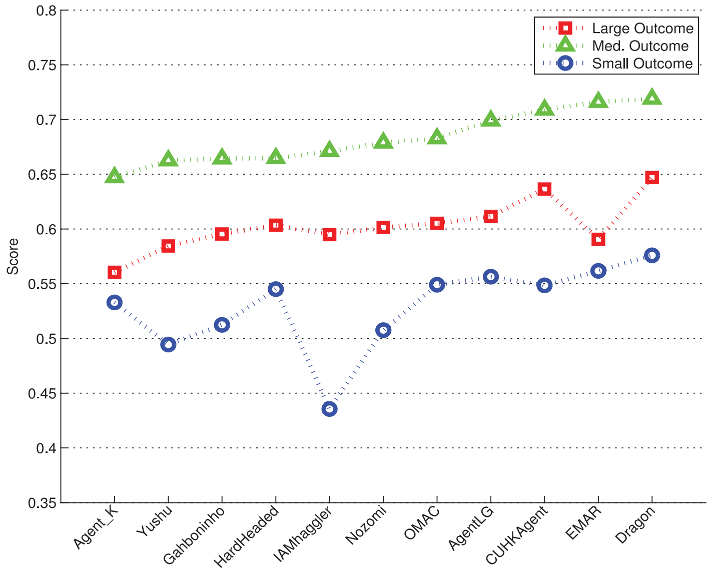


Fig. 5. Comparing performance of the agents in different levels of domain size.

manner as mentioned before. Not surprisingly, all agents managed to increase their profit with a decreasing competitiveness level. *DragonAgent* was the most successful agent in all three cases, where the difference to the other agents grew from 5.5% to 11.5% as the level of competitiveness increased. CUHKAgent made second place in the case of low and medium competitiveness, whereas AgentLG made second place in highly competitive domains. The performance of IAMhaggler2011 dropped most significantly as the competitiveness gets stronger. The results showed that *DragonAgent* was the most effect agent regardless of the level of competitiveness.

4.2.4. Impact of Different Outcome Space. To assess the effects of domain outcome space on agents' performance, the domains are classified into three classes to represent the different level of outcome space as follows: small (outcome space $\leq$ 100), medium (100 $<$ outcome space $<$ 3600), large (outcome space $\geq$ 3600). The results are shown in Figure 5, which are averaged over all scenarios of the included domains, as done in the previous subsection. As can be seen from the figure, these agents tended to achieve better negotiation results (e.g., higher scores) in medium-size domains than in small domains. *DragonAgent* advanced other competitors in all three classes of outcome space domains. In large domains, CUHKAgent and AgentLG finished in second and third place, respectively. EMAR and CUHKAgent performed quite well in medium outcome space domains, which were only second to *DragonAgent*. EMAR and AgentLG, following *DragonAgent*, were ranked second and third in small domains. By considering the agents' performance in a plenty of scenarios with various classes of domain outcome space, *Dragon* was well suited to domains with a broad range of outcome spaces.

4.3. Performance Summary

According to the overall performance shown in Table II, *Dragon* was the best negotiating agent considered in the experiments. The top three agents of ANAC 2012 achieved a mean score of 0.626, and the best agents of ANAC 2011 and 2010 obtained a similar mean score around 0.59, which in general presents a decreasing trend. With an average normalized score of 0.651, our agent led a margin of 8% over the mean score of these

Table II. Overall Performance of All Agents Across All Scenarios in Descending Order

Agent	*Normalized Utility*	*95% Confidence Interval*	
		Lower Bound	*Upper Bound*
Dragon	0.651	0.625	0.677
CUHKAgent	0.635	0.611	0.659
EMAR	0.629	0.604	0.654
AgentLG	0.627	0.601	0.652
OMAC	0.616	0.591	0.642
HardHeaded	0.607	0.584	0.630
Nozomi	0.600	0.574	0.625
Gahboninho	0.594	0.572	0.616
Agent_K	0.585	0.561	0.608
Yushu	0.584	0.563	0.606
IAMhaggler2011	0.571	0.552	0.590

Note: The letter in bold of each strategy is taken as its identifier for the later EGT analysis.

opponents. Then, CUHKAgent, EMAR, AgentLG, and OMAC took the second through fifth place in sequence. In addition, as the confidence intervals of several agents overlap with Dragon's confidence interval, Welch's t-test (with 95% confidence) was applied to the results to investigate whether the score differences are significant. The test showed that the score of Dragon was significantly better than the scores achieved by all other agents. To summarize, *Dragon* clearly outperformed, with a considerable margin, the state-of-the-art automated negotiators in a variety of application scenarios. *Dragon-Agent*'s ability of learning opponent behavior with high precision and avoiding irrational concession may account for this success.

Moreover, the noticeable performance gap between *DragonAgent* and IAMhaggler2011 is also interesting. Both agents employ GPs or a variant of GPs to model opponents, but IAMhaggler2011 was the worst-performing one in our experiments. Unlike our agent, IAMhaggler2011 (1) applies GP as a prediction tool and (2) adapts its concession rate fully on the basis of global predictions. The empirical evaluation suggests that one reason for this performance gap lies in the global prediction view. In more detail, this view seems to be vulnerable to irrational concession induced by pessimistic predictions. *Dragon* already avoided such a behavior as explained in Section 3.2. The phenomenon of irrational concession becomes increasingly apparent when IAMhaggler2011 bargains with opponents in nondiscounting domains where other players have no pressure to make early concession. Another reason may be because SPGPs are more suitable than GPs in the context of automated negotiation, as SPGPs are computationally efficient and thus more information can be processed to build the opponent model.

5. STRATEGY ROBUSTNESS ANALYSIS

Until now, the performance of strategies was studied only from the traditional mean-scoring perspective. This, however, did not reveal anything about the strategy robustness. For example, how would the results change if the players were free to switch their strategies for better individual payoffs, or if the mixture of opponent strategies were different? To address this issue, we analyze strategy robustness of the top six strategies ranked in the Table II with two cases. The first case concerns a two-player negotiation game, whereas the second analysis focuses on a spatial game in which a large number of players switch among the set of negotiation strategies to be better suited for the environment. For the spatial game, three versions of increasing range of agent interaction are considered.

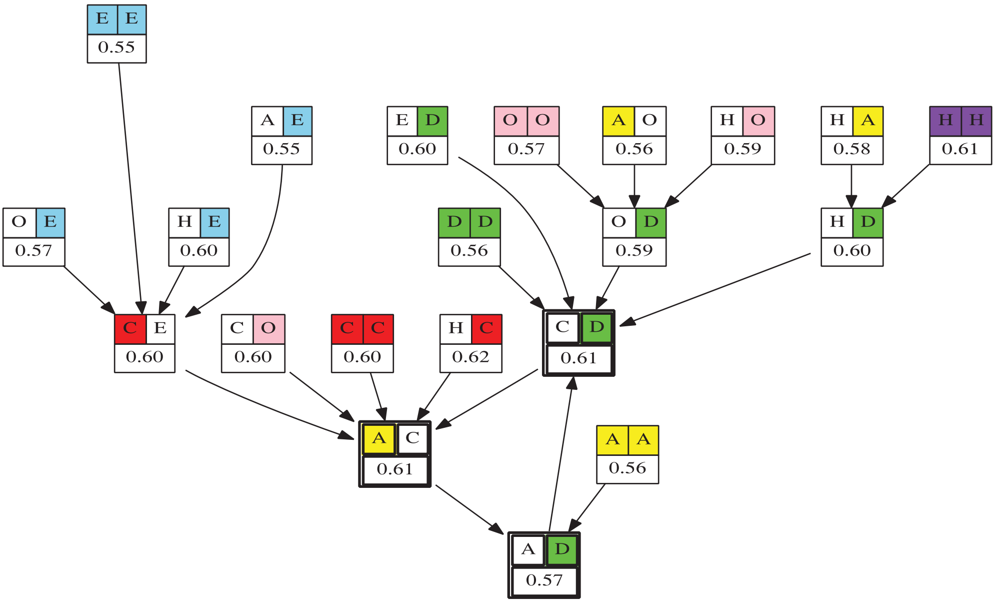


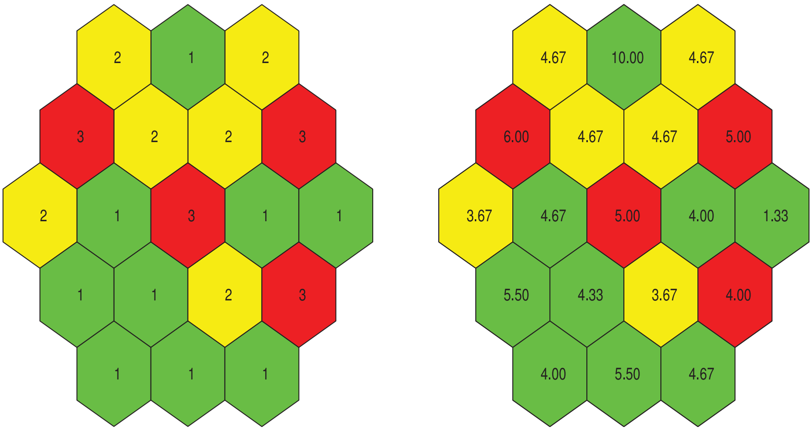
Fig. 6. Deviation analysis for the two-player negotiations among top six strategies. Each node shows a strategy profile, where the winning strategy that achieves a higher score is marked with a color background. An arrow indicates a statistically significant deviation to a different strategy profile. The nodes of the best reply cycle are indicated by a bold frame.

5.1. Robustness Analysis for the Two-Player Negotiation Game

For dealing with strategy robustness in the two-player negotiation game, the technique of empirical game-theoretic (EGT) analysis [Jordan et al. 2007] is applied to the tournament results. The aim of using EGT is to search for pure Nash equilibria, in which no agent has an incentive to deviate from its current strategy, or best reply cycle [Young 1993], where there exist a set of profiles (i.e., the combination of strategies chosen by players) for which a path of deviations exists that connect them, with no deviation leading to a profile outside of the set. The profiles in both Nash equilibria and best reply cycle are called *empirical stable states*. Strategy deviation is conducted following the ways mentioned in Williams et al. [2011], Baarslag et al. [2013], and Chen and Weiss [2013]. In such a deviation, one player unilaterally changes the strategy to statistically improve its own profit given the configuration of opponent strategies.

The abbreviation for each strategy is indicated by its first letter (in bold) in Table II. A profile (or node) in the resulting EGT graph is defined by the mixture of strategies used by the players in a state. The first row of a profile lists the strategies employed by the two players, and the second row shows the average score of the profile. The winning strategy of a profile is indicated by a color background in the sense that it achieves a higher score. Because we just concern the strategy mixture, the player order is not taken into account; otherwise, there are many redundant nodes (e.g., [A‖C] and [C‖A] in Figure 6) to be denoted, which makes the graph less visible.

The result under this EGT analysis is shown in Figure 6. As can be seen, there only exists a best cycle that includes three nodes. This best cycle is symmetric, consisting of three strategies: AgentLG, *Dragon*, and CUHKAgent. For any other strategy profile out of this cycle, there exists a path of statistically significant deviations (i.e., strategy changes) that lead to a profile within the cycle. Moreover, despite the fact that EMAR



(a) Strategy of each cell (each strategy is marked with a different color).

(b) Fitness of each cell.

Fig. 7. A locality example in a hexagon model.

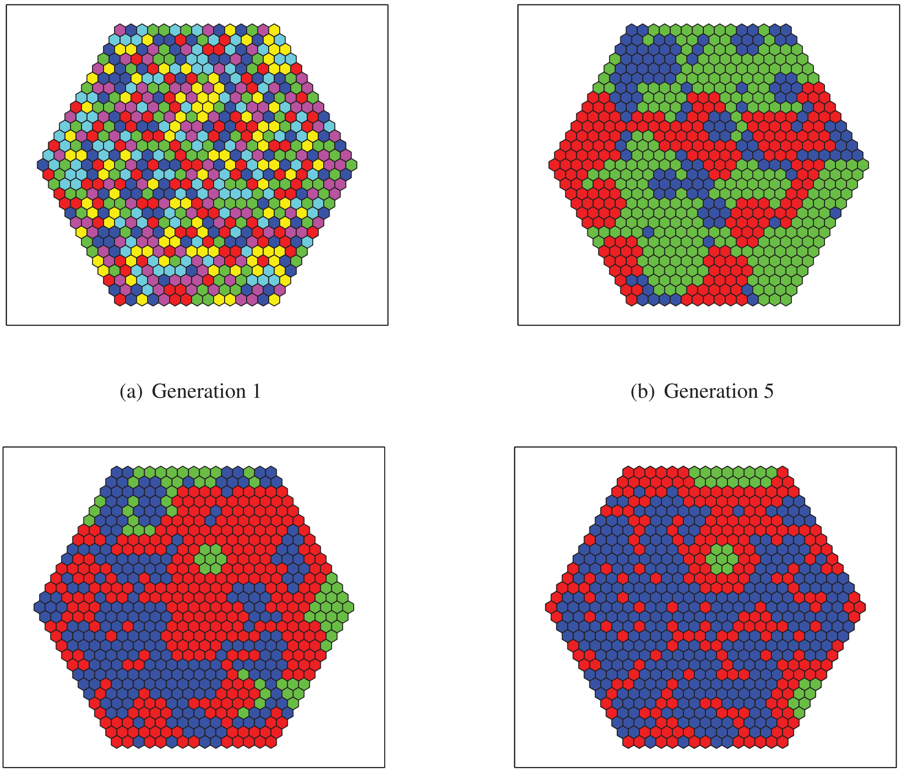
is the third best strategy (refer to Table II), it is not contained by any node in the best reply cycle. *Dragon*, as the best-performing strategy in the tournaments, is still very successful since it exists in two thirds of the best-cycle profiles, which is also the winning strategy of those profiles. By contrast, AgentLG is the winning strategy in one stable state, and CUHKAgent is not a winning strategy in any empirical stable states. The EGT analysis suggests that *Dragon* is strongly robust in the two-player negotiation game.

5.2. Robustness Analysis in the Spatial Negotiation Game

Locality is an important factor in negotiation problems where neighborhood needs to be taken into consideration. However, strategy robustness with respect to locality has not been well studied so far. Thus, this work introduces spatial negotiation games to discuss the strategy robustness in different ranges of agent interaction. A spatial negotiation game is a game in which a population of players/agents in a certain area, for the sake of higher fitness, use a set of negotiation strategies to compete against each other. To analyze spatial negotiation games, we apply evolutionary game theory—more precisely, spatial evolutionary game theory [Killingback and Doebeli 1996; Szabó and Fáth 2007]—to the tournament results. This allows us to study the impact on fitness (i.e., how well a individual is adapted to a dynamic environment) of each species (strategy) competing against others locally.

The fitness of an individual or a cell (as it is located at a certain environmental position) is determined by the average payoff of its own strategy playing against its neighbors. Take a simple case with three strategies as an example, where the center cell choosing strategy 1 meets its neighbors as shown in Figure 7(a). The fitness of the center cell is the average payoff of playing against three opponents using strategy 1 and three opponents using strategy 2. To put it differently, it has the neighbor distribution $x = (0.5,0.5,0)$. The payoff matrix of the three strategies is given by matrix A:

$$A = \begin{bmatrix} 4 & 10 & 0 \\ 1 & 4 & 9 \\ 3 & 7 & 4 \end{bmatrix},$$



(a) Generation 1

(b) Generation 5

(c) Generation 15

(d) Generation 45

Fig. 8. Evolution of strategy distribution.

where an entry $A(i, j)$ is the payoff of strategy i against strategy j. Thus, the fitness (ρ) of the center cell is 5.0, following the equation

$$\rho = e_i A x^T, \tag{18}$$

where e_i denotes the i^{th} row of a unit matrix e with the size of the number of strategies and A denotes the payoff matrix. For clarity, the fitness of every cell is shown in Figure 7(b).

Our analysis assumes that the players in the game have the freedom to choose one of the six negotiation strategies as in the two-player negotiation game. There also exists a payoff matrix suggesting utilities of any pair of strategies (i.e., their scores averaged across all available scenarios). Each strategy is initialized with a equal number of 100 players, randomly distributed over a two-dimensional hexagon lattice Λ. A cell (I) in the lattice represents a player and is occupied by one strategy and bordered with six adjacent cells—that is, each cell has six neighbors in its local scale. Calculating the fitness of each cell in the field is simultaneously performed. Each cell then imitates which one has the highest fitness in its neighborhood (including itself). In this way, the natural selection process (i.e., how to update the strategy of the cell for the next generation) is well defined.

Figure 8 provides an example illustrating how the players switch between these strategies in such a environment. In the first generation (as shown in Figure 8(a)),

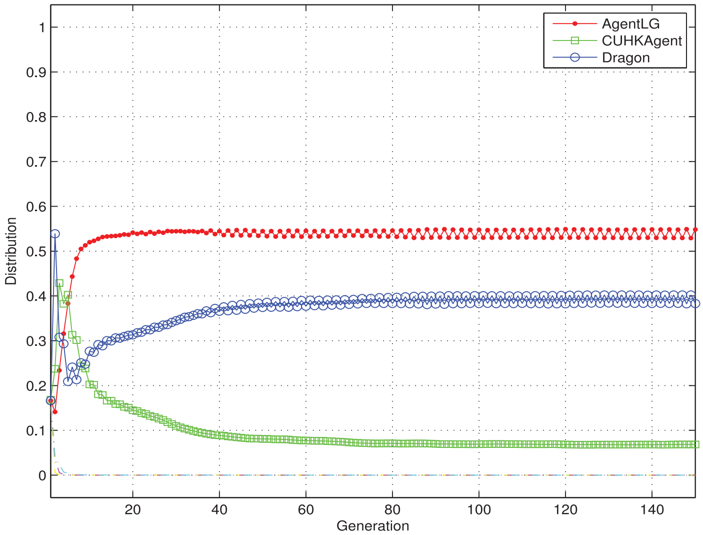


Fig. 9. Strategy distributions over generations when players are allowed to interact with their neighbors.

every cell is randomly assigned with a strategy, and each strategy is indicated by a distinct color. The shares of the strategies in the population are made equivalent. After four generations (see Figure 8(b)), there are three strategies surviving in the game, which correspond to those appearing in the best reply cycle that we have found in Section 5.1. In the 15th generation (Figure 8(c)), the competition between these three strategies results in the fading away of CUHKAgent (green), whereas AgentLG (red) and *Dragon* (blue) together achieve more shares in the population. As the evolution continues (see Figure 8(d)), CUHKAgent nearly vanishes, and the other two strategies coexist well and eventually become the dominant strategies.

The dynamics of strategy distribution, as a topic of central interest for evolutionary game theory, serve as a very important indicator of strategy robustness in this case. To guarantee results with high statistical significance, the simulation was run 10,000 times with random initialization of the location arrangement of the six strategies. Figure 9 shows the variation of strategy proportions over generations when players interact with their neighbors. In the beginning, *Dragon* soars up to its maximal share and then experiences a slump in the population. After this, its proportion slowly grows again. In contrast, there is a prompt increase in the number of players choosing AgentLG; later, it oscillates with a rather low amplitude. Other strategies, including EMAR, OMAC, and HardHeaded, are exterminated in a very short period. After a large number of generations, when only very few players are left that have an incentive to change their strategies for better fitness, around 52% of the population choose the strategy AgentLG, whereas 39% of the agents switch to *Dragon*. Strategy proportions reach a “stable” state (in which almost no players would like to change their strategies) around the 120th generation. Moreover, no equilibrium states, where no players have the incentive to switch to a different strategy, could be found, as there is only tiny oscillation in the percentages of the two dominate strategies.

It is also interesting to investigate the results when the interaction range of a cell is extended to its neighbors’ neighbors. In doing so, we could have a clearer insight on the impact of interaction range. Toward this end, the nature selection process is also modified such that a player is permitted to mimic the strategy used by the most successful player in this wider neighborhood. The results are depicted in the Figure 10.

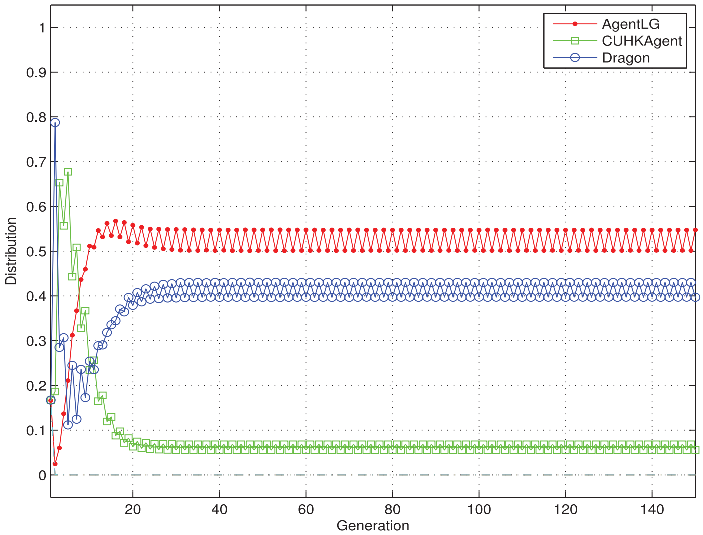


Fig. 10. Strategy distributions over generations when players are allowed to interact with their neighbors' neighbors.

In contrast to the former version, the shares of the three strategies more quickly reach a stable state. The oscillation between the shares of the two coexisting strategies, however, becomes more obvious. Moreover, the proportion of *Dragon* grows larger in this case. The result is caused by its competitive advantage over the opponents. As a matter of fact, when further extending the interaction range of a player to all other players existing in the game, we observe that all cells promptly switch to *Dragon*—that is to say, the game converges to a equilibrium state soon where only a single strategy (*Dragon*) exists.

6. RELATED WORK

Approximating the opponent's model in negotiation has been of growing interest in the agents community (e.g., Zeng and Sycara [1997], Coehoorn and Jennings [2004], Saha et al. [2005], Narayanan and Jennings [2006], Lin et al. [2008], Hindriks et al. [2009a], and Hendrikx [2011]). However, most of the proposed approaches are either restrictive in their assumptions or computationally expensive. For instance, Faratin et al. [2002] designs a trade-off strategy to increase the offer acceptance rate. Coehoorn and Jennings [2004] uses kernel density estimation to approximate the opponent's preferences. Although successful, the effectiveness of these methods highly depends on the availability of extra information, such as the negotiation history, the opponent's strategy, or certain domain knowledge. Other research efforts adopt a Bayesian setting to aid learning in automated negotiation. In Lin et al. [2008], for instance, a reasoning model is introduced to learn the likelihood of an opponent profile. Hindriks and Tykhonov [2008] present a generic framework to learn the outcome preference order of an opponent. Further, Rahman et al. [2011] develop an order statistics Bayesian-mining agent to learn the opponent's preference in automated negotiation. To reduce the hypothesis space and enlarge the scalability, previous negotiation traces are assumed to be accessible for the agent. The main problem behind these approaches is the computational effort needed to learn in problems of high dimensionality, where the computational complexity of Bayesian learning increases exponentially with the increase in the problem's dimensionality.

Furthermore, Saha et al. [2005] applies Chebychev polynomials to estimate the offer acceptance probability in repeated single-issue negotiations. Brzostowski and Kowalczyk [2006] make use of differentials to perform online prediction of future counteroffers based on the negotiation history, assuming that the opponent strategy is a weighted combination of two basic negotiation tactics introduced in Faratin et al. [1998]. Alternatively, Narayanan and Jennings [2006] employ a combined technical scheme built on Markov chains and Bayesian learning to model the opponent's strategy in single-issue negotiations. But it makes the assumption that the set of opponent strategies is prior knowledge. In Carbonneau et al. [2008], an artificial neural network is constructed to predict future counteroffers. But the training process places demands on a large amount of previous encounters as well as computational resources. Although the existing work together advances the field of automated negotiation, it still suffers from several limitations. These could be summarized as (1) restrictive structural assumption or (2) high computational efforts.

Recent research aiming at solving the assumption problem for the complex negotiation that we define in Section 1 includes Williams et al. [2011]. In this work, the authors attempt to predict the opponent's future behavior using GPs. The resulting model is then used by the agent to adjust its own concession strategy. In addition, they use empirical game theory to verify the robustness of their strategy against the best agents from ANAC 2010. Hao and Leung [2012] introduce a negotiation strategy named *ABiNes* to deal with negotiations in complex environments. To operate the negotiation process well, ABiNeS exploits the behavior of the other negotiation party to determine where to start compromising. In addition, it employs a reinforcement learning approach to improve the acceptance probability of its proposals. Its implementation under the name of CUHKAgent was the champion of ANAC 2012. Another noteworthy work is Chen and Weiss [2012], in which a negotiation approach—OMAC—is proposed that learns the opponent's strategy to achieve better outcomes. In more detail, OMAC first decomposes the received utilities of counteroffers through discrete wavelet decomposition and then it makes use of cubic smoothing splines to predict future opponent compromise. With the assistance of the learned model, an effective counterstrategy to the opponent can be decided. This agent finished second in the qualifying round of ANAC 2012 and made third place in the final. In a more recent work [Chen and Weiss 2013], the authors solve complex negotiations by means of empirical mode decomposition, which, unlike wavelet decomposition, automatically transforms a time series into several independent subcomponents (i.e., intrinsic mode functions) with different frequencies. Each subcomponent is then dealt with an appropriate regression model to learn the opponent strategy. A detailed comparison to the preceding methods was conducted in Section 4, where the proposed strategy outperformed the aforementioned strategies.

Among the currently available negotiation approaches, the one most related to *Dragon* is IAMhaggler2011 in that both employ GPs to learn opponent behavior. As can be seen from the reported results, however, *Dragon* clearly exceeds IAMhaggler2011 for three main reasons: *Dragon* adopts a more efficient prediction model (based on SPGPs), a more sophisticated concession-making strategy, and a more effective responding scheme. Specifically, SPGPs employed by *Dragon* are capable of delivering very similar results but at lower computational cost. Furthermore, the concession-making mechanism is more reasonable and robust. By using the dynamic conservative expectation function, *Dragon* concedes toward an opponent's demand in a very cautios manner, which helps to avoid the problem of irrational concession from which

IAMhaggler2011 suffers. Moreover, unlike IAMhaggler2011, which generates offers in a simple random way, *Dragon* applies a greedy strategy based on a frequency learning mechanism to determine best possible future offers.

7. CONCLUSIONS

This work introduced an effective strategy called *Dragon* for a negotiating agent in complex bilateral multi-issue, no prior knowledge, time-constrained, low computational load, influence of reservation value scenarios. The comprehensive experiments conducted in our work clearly demonstrated the effectiveness of the proposed negotiating agent, which is built on SPGPs and an adaptive decision-making scheme. Precisely, the performance of *Dragon* was carefully evaluated with respect to the following three aspects:

—Two tournaments were run to investigate how well *DragonAgent* performed against a large number of well-known negotiating agents (the primary tournament) and the champions of each year's ANAC plus an additional novel agent (the advanced tournament) in a broad range of negotiation scenarios. Experimental results showed that *DragonAgent* generated a mean score statistically higher than all benchmark agents in both tournaments. The main competitor to the new strategy (i.e., IAMhaggler2011) was ranked last (see Table II). Based on our investigations, we believe that a major reason for this performance difference lies in the irrational concession problem that *DragonAgent* automatically avoids.

—The impact of different levels of domain competitiveness and outcome space on *DragonAgent*'s performance was assessed as well. According to the results, *DragonAgent* won all sub-tournaments regardless of the level of domain competitiveness or outcome space.

—Strategy robustness was discussed in the two-player negotiation and spatial negotiation game, respectively. The first game highlights the robustness of a strategy in bilateral negotiation encounters; the second one concerns the case where a population of players compete for a higher individual payoff. The analysis based on game theory manifested that *Dragon* is a robust strategy in both games. Moreover, this strategy became increasingly successful as the agent interaction range grew in the spatial negotiation game.

Overall, based on an advanced technical framework, *DragonAgent* outperformed state-of-the-art negotiating agents, both from the standard mean-score perspective as well as a game-theoretic perspective.

The exceptional results justify and investment of further research effort into this approach. In the future, we plan on comparing the opponent modeling scheme with other approaches available to complex negotiation. Second, the extension of this framework to other negotiation settings, such as concurrent negotiation or multilateral negotiation, is another interesting avenue to exploit. As well, it is of great interest to investigate the performance of this automated negotiation strategy against human negotiators. Then, integrating a more powerful preference learning technique into our agent may lead to a significant improvement in negotiation power. Last, but not least, to further aid the learning performance, we consider incorporating transfer learning techniques into the proposed strategy.

ACKNOWLEDGMENTS

We would like to thank the China Scholarship Council (CSC) for providing a PhD scholarship (Grant number: 2010607022) to Siqi Chen. Moreover, we appreciate the fruitful discussions with the members of the DKE Swarmlab. Special thanks also go to the anonymous reviewers of this work for their valuable comments.

REFERENCES

Sunil Adhau, M. L. Mittal, and Abhinav Mittal. 2012. A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence* 25, 8, 1738–1751.

Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. 2013. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence* 198, 73–103.

Jakub Brzostowski and Ryszard Kowalczyk. 2006. Predicting partner's behaviour in agent negotiation. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, New York, NY, 355–361.

Réal Carbonneau, Gregory E. Kersten, and Rustam Vahidov. 2008. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications* 34, 2, 1266–1273.

Siqi Chen, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. 2013a. Optimizing complex automated negotiation using sparse pseudo-input Gaussian processes. In *Proceedings of the 12th International Joint Conference on Automomous Agents and Multi-Agent Systems*. ACM, New York, NY, 707–714.

Siqi Chen, Haitham Bou Ammar, Karl Tuyls, and Gerhard Weiss. 2013b. Using conditional restricted Boltzmann machine for highly competitive negotiation tasks. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. AAAI, Palo Alto, CA, 69–75.

Siqi Chen and Gerhard Weiss. 2012. An efficient and adaptive approach to negotiation in complex environments. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS, Amsterdam, The Netherlands, 228–233.

Siqi Chen and Gerhard Weiss. 2013. An efficient automated negotiation strategy for complex environments. *Engineering Applications of Artificial Intelligence* 26, 10, 2613–2623.

Robert M. Coehoorn and Nicholas R. Jennings. 2004. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th International Conference on Electronic Commerce*. ACM, New York, NY, 59–68.

Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. 1998. Negotiation decision functions for autonomous agents. *Journal of Robotics and Autonomous Systems* 24, 4, 159–182.

Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. 2002. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence* 142, 2, 205–237.

Jianye Hao and Hofung Leung. 2012. ABiNeS: An adaptive bilateral negotiating strategy over multiple items. In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, Vol. 2 (WI-IAT'12)*. IEEE, Los Alamitos, CA, 95–102.

Mark Hendrikx. 2011. *A Survey of Opponent Models in Automated Negotiation*. Technical Report. Delft University of Technology, The Netherlands.

Koen Hindriks, Catholijn Jonker, Sarit Kraus, Raz Lin, and Dmytro Tykhonov. 2009b. GENIUS: Negotiation environment for heterogeneous agents.. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, New York, NY, 1397–1398.

Koen Hindriks, Catholijn M. Jonker, and Dmytro Tykhonov. 2009a. The benefits of opponent models in negotiation. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT'09)*. 439–444.

Koen Hindriks and Dmytro Tykhonov. 2008. Opponent modelling in automated multi-issue negotiation using Bayesian learning. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, New York, NY, 331–338.

Nick R. Jennings, Peyman Faratin, Alessio R. Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. 2001. Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation* 10, 2, 199–215.

Patrick R. Jordan, Christopher Kiekintveld, and Michael P. Wellman. 2007. Empirical game-theoretic analysis of the TAC supply chain game. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*. ACM, New York, NY, 1188–1195.

Timothy Killingback and Michael Doebeli. 1996. Spatial evolutionary game theory: Hawks and Doves revisited. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 263, 1374, 1135–1144.

Raz Lin, Sarit Kraus, Jonathan Wilkenfeld, and James Barry. 2008. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence* 172, 6–7, 823–851.

Fernando Lopes, Michael Wooldridge, and Augusto Novais. 2008. Negotiation among autonomous computational agents: Principles, analysis and challenges. *Artificial Intelligence Review* 29, 1, 1–44.

Dirk C. Moosmayer, Alain Yee-Loong Chong, Martin J. Liu, and Bjoern Schuppar. 2013. A neural network approach to predicting price negotiation outcomes in business-to-business contexts. *Expert Systems with Applications* 40, 8, 3028–3035.

Vidya Narayanan and Nicholas R. Jennings. 2006. Learning to negotiate optimally in non-stationary environments. In *Cooperative Information Agents X*. Springer, 288–300.

Samir Abdel Rahman, Reem Bahgat, and George M. Farag. 2011. Order statistics Bayesian-mining agent modelling for automated negotiation. *Informatica: An International Journal of Computing and Informatics* 35, 1, 123–137.

Howard Raiffa. 1982. *The Art and Science of Negotiation*. Harvard University Press, Cambridge, MA.

Carl E. Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

Ariel Rubinstein. 1982. Perfect equilibrium in a bargaining model. *Econometrica* 50, 1, 97–109.

Sabyasachi Saha, Anish Biswas, and Sandip Sen. 2005. Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, New York, NY, 397–403.

Victor Sanchez-Anguix, Vicente Julian, Vicente Botti, and Ana Garła-Fornes. 2013. Tasks for agent-based negotiation teams: Analysis, review, and challenges. *Engineering Applications of Artificial Intelligence* 26, 10, 2480–2494.

Edward Snelson and Zoubin Ghahramani. 2006. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 1257–1264.

György Szabó and Gábor Fáth. 2007. Evolutionary games on graphs. *Physics Reports* 446, 4, 97–216.

Colin R. Williams, Valentin Robu, Enrico H. Gerding, and Nicholas R. Jennings. 2011. Using Gaussian processes to optimise concession in complex negotiations against unknown opponents. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. AAAI, Palo Alto, CA, 432–438.

H. Peyton Young. 1993. The evolution of conventions. *Econometrica: Journal of the Econometric Society* 61, 1, 57–84.

Dajun Zeng and Katia Sycara. 1997. Benefits of learning in negotiation. In *Proceedings of the 14th National Conference on Artificial Intelligence*. AAAI, Palo Alto, CA, 36–41.

Received November 2013; revised March 2014; accepted May 2014