Optimizing Complex Automated Negotiation using Sparse Pseudo-input Gaussian processes

Siqi Chen, Haitham Bou Ammar, Karl Tuyls, Gerhard Weiss
Department of Knowledge Engineering, Maastricht University
P.O. Box 616, 6200 MD
Maastricht, The Netherlands
{siqi.chen, haitham.bouammar, k.tuyls, gerhard.weiss}@maastrichtuniversity.nl

ABSTRACT

Complex negotiations among rational autonomous agents is gaining a mass of attention due to the diversity of its possible applications. This paper deals with a prominent type of complex negotiations, namely, multi-issue negotiation that runs under real-time constraints and in which the negotiating agents have no prior knowledge about their opponents' preferences and strategies. We propose a novel negotiation strategy called **Dragon** which employs sparse pseudo-input Gaussian processes (SPGPs) to model efficiently the behavior of the negotiating opponents. Specifically, with SPGPs **Dragon** is capable of: (1) efficiently modeling unknown opponents by means of a non-parametric functional prior; (2) significantly reducing the computational complexity of this functional prior; and (3) effectively and adaptively making decisions during negotiation. The experimental results provided in this paper show, both from the standard mean-score perspective and the perspective of empirical game theory, that **Dragon** outperforms the state-of-the-art negotiation agents from the 2012 and 2011 Automated Negotiating Agents Competition (ANAC) in a variety of negotiation domains.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distribute Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Algorithms, Performance, Experimentation, Theory

Keywords

Multi-agent systems; Automated Multi-issue Negotiation; Opponent Modeling; Sparse Gaussian Process; Empirical game theory

1. INTRODUCTION

Because of the broad spectrum of potential applications in industrial and commercial domains, automated negotiation is achieving steadily growing attention as a fundamental and powerful mechanism for managing interaction among computational agents, which are in a consumer-provider or buyer-seller relationship and thus typically have different interests over possible joint agreements on some matter.Various forms of negotiation can be distinguished [15].

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

Research described in this paper deals with bilateral multi-issue negotiation, which is widely used. 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 which are of conflictive importance to the negotiators, such as price, delivery time, quantity, and quality.

To have a more realistic setting, the work described here studies the scenarios in which the agents have no prior information about their opponents – neither about their preferences (e.g., their preference over issues or issue value ordering) nor about their negotiation strategies. Moreover, this paper focuses on negotiation settings with deadline and discount and in which the negotiating agents do not know the remaining number of the negotiation rounds. With that, the negotiators have to take into account (1) the remaining negotiation time and (2) the fact that the achievable profit through an agreement decreases over time (i.e., is discounting). Furthermore, we assume that each agent has a private reservation value below which an offered contract is not accepted, and we adopt the common view that an agent obtains the reservation value even if no agreement is reached during negotiation. Despite being complex, this overall setting is of relevance to a wide range of practical applications, and is also common to many human-human negotiation scenarios.

A major factor to a successful negotiation is modeling the opponent's behavior, however it is indeed challenging because negotiators generally are not open about their true preferences and strategies to avoid exploitation [6, 16]. Although there exist methods for solving this problem, two issues still stand out. The first one relates to the simplifying assumptions made about the opponent's model. Typically, simplifying assumptions about the structure of the modeled function, and/or the rate of change in the function are made. These allow using simpler approximation methods. The problem is that these assumptions are usually overly simplified and thus underestimate the opponent's model. Others tried to avoid these simplifications by adopting more sophisticated approximators, which relate to the second issue: computational complexity or the availability of computational resources in the negotiation setting. Specifically, these sophisticated approximation techniques perform well in capturing the structure of the underlying latent model, but face problems dealing with higher dimensions and/or larger data sets, prohibiting their applications in more complex scenarios.

We tackle these problems and make a number of contributions. The primary contribution is the proposition of a novel negotiation strategy called **Dragon**. This strategy makes use of SPGPs to: (1) relax the modeling assumptions of other approaches by employing a non-parametric functional prior, making it capable of modeling highly complex opponent models, and (2) reduce the computation complexity of learning in such a non-parametric 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 optimal concession degree of the opponent, and (2) avoid the problems related to "irrational concession", see Section 4.2.

The presented experiments, performed in negotiating against state-of-the-art opponents clearly demonstrate the effectiveness of the proposed strategy. More precisely, **Dragon** outperforms the top-ranked negotiation agents of the 2012 and 2011 ANAC from the mean score perspective. Further studies conducted using empirical game theory show that **Dragon** not only outperforms the other agents, but is also robust when the mix of opposing strategies was different.

The remainder of this paper is structured as follows. Section 2, overviews important related work and pin-points their major deficiencies. Section 3, provides the reader with background knowledge needed to understand the remainder of the paper. Then, the technicalities of **Dragon** are explained in Section 4. A careful experimental analysis of the approach is given in Section 5. Finally, Section 6, concludes and identifies interesting future research directions.

2. RELATED WORK

Approximating the opponent's model in negotiations has been of growing interest in the agents community, see [10] for an overview. However, most of the proposed approaches are either restrictive in their assumptions, or computationally expensive. For instance, Coehoorn and Jennings [6] use Kernel Density Estimation to approximate the opponent's preferences. Faratin et al. [8], design a trade-off strategy to increase the offer acceptance rate. Though successful, the effectiveness of these methods highly depend on the availability of extra information, such as the negotiation history, the opponent's strategy, or other domain knowledge. Other research efforts adopt a Bayesian setting to aid learning in automated negotiations. In [14], for instance, a reasoning model is introduced to learn the likelihood of an opponent profile. Hendrikx and Tykhonov [12] present a more generic framework for opponent modeling to learn opponent preferences. 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. [19] applies Chebychev Polynomials to estimate the offer acceptance probability in repeated single-issue negotiations. Brzostowski et al. [3] makes use of differentials to perform online prediction of future counter-offers based on the previous negotiation history, with the assumption that the opponent strategy is only based on a combination of two basic negotiation tactics introduced in [7]. In [4] an artificial neural network (ANN) was constructed to predict future counter-offers, which places demands on a large amount of previous encounters and computational resources to complete the training. Although this existing work advances the field of automated negotiations, 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 includes William et al. [21]. In this work the authors applied Gaussian processes to predict the opponent's future concession. The resulting model was then used by the agent to adjust its own concession strategy. Towards the same end, Hao et al. [9] introduced a negotiation strategy named ABiNes[1] to deal with negotiations in complex environments. To successfully perform negotiations, ABiNeS adjusts the time to stop exploiting the negotiating partner and also employs a reinforcement-learning approach to improve the acceptance probability of its proposals. Another noteworthy work is [5], where Chen and Weiss proposed the negotiation approach "OMAC" that learns the opponent's strategy to predict utilities of future counter-offers through discrete wavelet decomposition and cubic smoothing splines. A detailed comparison to the above methods is performed in Section 5, where the proposed strategy outperforms the aforementioned strategies.

[1]Its implementation is called CUHKAgent, which achieved the champion of ANAC 2012.

3. BACKGROUND

In this section we provide the reader with the background knowledge needed to understand the remainder of the paper. Firstly, the negotiation model in which the agents operate is explained. Secondly, the regression framework including both Gaussian Processes and Sparse Pseudo-Input Gaussian Processes is detailed.

3.1 Negotiation Model

A basic bilateral multi-issue negotiation setting which is widely used in the agents field (e.g., [7, 8]) is adopted. The negotiation protocol is based on the standard alternating offers formalized in [18]. Let i be a specific agent, j be a particular issue and k represent the choice of the j^{th} issue. We define the value of issue j as v_{jk}. Each agent has a lowest expectation for the negotiation outcome called the reservation value (θ). Further, w_j^i denotes the weighting preference which agent i assigns to issue j. The weights of a specific agent i over the issues are normalized summing to one (i.e., $\sum_{j=1}^{n}(w_j^i) = 1$). During negotiation the two agents act in conflictive roles specified by their preferences. 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:

$$U^i(O) = \sum_{j=1}^{n}(w_j^i \cdot V_j^i(v_{jk})) \tag{1}$$

where, V_j^i 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. The negotiation continues until one party accepts a counter-offer or an agent breaks off during the process. If no agreement is reached at the end, the disagreement solution specified by the scenario is then adopted. This is also applied for the case in which an agent withdraws from the negotiation in advance.

Discounted domains are also considered. We define a discounting factor δ ($\delta \in [0, 1]$) which is used to calculate the discounted utility as follows:

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

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

3.2 Gaussian Processes

In the field of machine learning, Gaussian Processes (GPs) are one of the well-known, non-linear, non-parametric regression techniques. These models have been successfully applied in negotiation settings by [21]. Although GPs are a powerful form of function approximators, they suffer from computational problems once dealing with large data sets. 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 strategy based on Sparse Pseudo-inputs Gaussian Processes (SPGPs). These models detailed in Section 3.2.2, are able to achieve similar modeling accuracy to GPs but with much less computational effort.

3.2.1 Gaussian Processes

Gaussian Processes (GPs) are a form of non-parametric regression techniques. Following the notation of [17], 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, $y \in \mathbb{R}$ the output vector and m 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.

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

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 data set 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 [17] for a thorough discussion of the topic. To fit the hyperparameters that best suit the available data set 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,

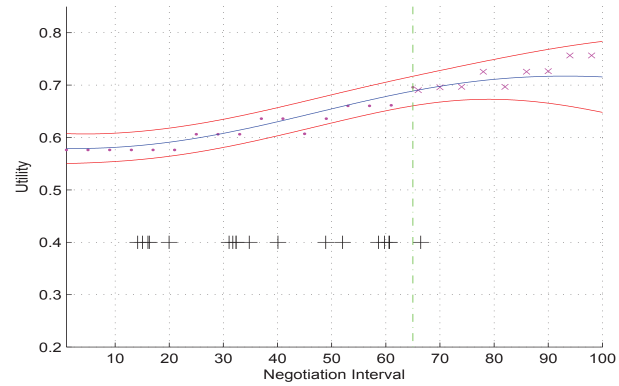
$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|\mathbf{X},\mathbf{\Theta}) &= \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}\right)\\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

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.

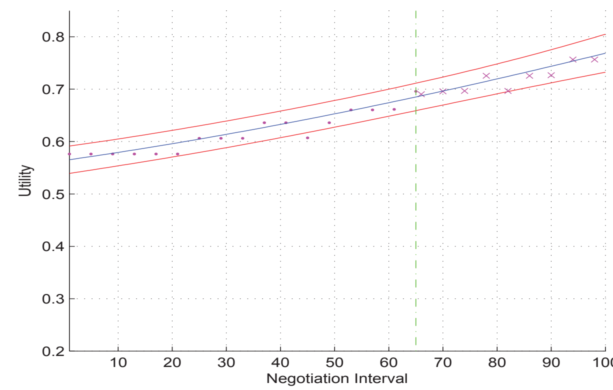
It is for this specific reason that we employ a fast and more efficient learning technique (i.e., Sparse Pseudo-input Gaussian Processes (SPGPs)). The most interesting feature of SPGPs is that these approximators are capable of attaining very close accuracy in both learning and prediction to normal GPs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

3.2.2 Sparse Pseudo-input Gaussian Processes

As mentioned previously, GPs are computationally expensive to learn especially in an online setting. SPGPs aim at reducing the complexity of GPs in both learning and prediction. The idea is to parametrize the regression model with the so-called pseudo-inputs. The location of these inputs is iteratively fitted by maximizing a new kind of marginal likelihood. Interestingly, using only a small



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



(b) Illustration of the predictive power of GPs

Figure 1: The blue curves represent the mean of the approximated function while the red lines represent the variance. The black crosses in Figure 1(a) show the locations of the pseudo-inputs and the vertical dash-dot lines in both figures denote the interval at which the prediction is taking place.

amount of pseudo-inputs, SPGPs are capable of attaining very similar fitting and prediction results to normal GPs. To clarify, the idea is to parametrize the model by $M << n$ pseudo-input points, while still preserving the full Bayesian framework. This leads to the parametrization of the covariance function by the location of $M <<< n$ pseudo-inputs. These are then fitted in addition to the hyperparameters in order to maximize the following new marginal likelihood:

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

where, $\bar{\mathbf{X}}$ is the matrix formed by the pseudo-inputs 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 pseudo inputs with $\mathbf{K}_M = k(\bar{\mathbf{x}}_m, \bar{\mathbf{x}}_m)$. $\mathbf{\Lambda}$ 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 [20] 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

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, E_δ denotes the discounted expected utility of incoming counter-offers. E_{low} is the lowest expectation to negotiation, ρ is the compromise point, and R is the dynamic conservative expectation function. u' is the target utility at time t_c.

```
1:  Require: R, δ, ξ, t_max
2:  while t_c <= t_max do
3:     O_opp ⇐ receiveMessage;
4:     recordOffers(t_c, O_opp);
5:     if TimeToUpdate(t_c) then
6:        χ ⇐ preprocessData(t_c)
7:        E_δ ⇐ Predict(χ, ξ);
8:        (E_low, ρ) ⇐ updateParas(t_c);
9:        R ⇐ (E_low, ρ);
10:    end if
11:    u' = getTargetUtility(t_c, E_δ, δ, R);
12:    if isAcceptable(u', O_opp, t_c, δ) then
13:       accept(O_opp);
14:    else
15:       checkTermination();
16:       O_own ⇐ constructOffer(u') ;
17:       proposeNewBid(O_own);
18:    end if
19: end while
```

inputs) to $\mathcal{O}(M^2)$. The results further demonstrate that SPGPs can fully match normal GPs with small M (i.e., few pseudo-inputs), successfully producing very sparse solutions. A full mathematical treatment may be found elsewhere [20].

In order to show the usefulness and powerfulness of SPGPs we conducted an experiment on artificially generated data. With only $M = 20$, SPGPs where able to attain very similar results to normal GPs as shown in Figure 1. It is clear that both learned function follow a very similar increasing trend. Predictions made at 50 negotiation intervals also show similar predicted values in both cases. The black crosses in Figure 1(a) represent the location of the fitted pseudo-inputs. It is clear that these pseudo-inputs were mostly located in critical ranges of the function.

4. PROPOSED METHOD

The overall **Dragon** strategy is shown in Algorithm 1. **Dragon** consists of three functional components, which are essential and vital for the agent to operate successfully. Firstly, the opponent-modeling component is described. It adopts a non-parametric and computationally efficient regression technique in order 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 "irrational concession". Finally, the third and last stage of **Dragon** (i.e., the responding component) responds to the offers of the opponent and determines the time at which the negotiation session terminates. Next, each of the above components is detailed.

4.1 Opponent-modeling component

Modeling the opponent's behavior is done by the first component of **Dragon**. It adopts the SPGPs, detailed in Section 3.2.2, in order to accurately and efficiently learn the opponent's model. This process of opponent modeling corresponds to the lines 2 to 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, a small change in utility of the opponent can result in a large utility variation for our agent leading to a fatal misinterpretation of the opponent's behavior. Therefore and in order to reduce that negative impact, the whole negotiation is divided into a fixed number (denoted as ζ) of equal intervals. 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 [21]. This automatically leads our agent to have more accurate predictions of the future opponent's behavior compared to [21].

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:

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

where

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

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 $t_\star$, δ is the discounting factor.

On the contrary to the work of [21] 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 way doesn't generate a valid probability density function anymore while ours guarantees that.

4.2 Concession-making component

Using the approximated model, the concession-making component aims are setting the optimal concession degree. Two factors are taken into account to determine the optimal concession degree. The first, is based on the prediction of the opponent's future compromise, while the second builds on the agent's own reservation function.

Though successful, the results are sometimes rather over-pessimistic due to "boulware" [7] behavior of the opponent. To clarify, "sophisticated and tough" negotiation opponents are likely to escape concession in bargaining. Thus, in this case the prediction results could lead to a misleading, and very low expectation about the utility offered by the opponent resulting in an adverse concession behavior. To solve "irrational concession", **Dragon** therefore employs a dynamic conservative expectation function $R(t)$. Informally, it is a "dynamic conservative expectation function that carefully suggests utilities". The main idea behind $R(t)$ is to lower the utility expectation with time. Since smaller values of the discount factor, δ, force rational agents to reach agreements earlier, $R(t)$ is inversely proportional to δ. $R(t)$ further takes the lowest expectation as its minimum value. To define $R(t)$ we introduce two new variables, the compromise point – ρ and the lowest expectation – E_{low}. In what follows, we motivate the need for these two variable and detail the formal technicalities for defining $R(t)$.

The main objective of the strategy is to achieve as high profit as possible. It is therefore highly demanded to exploit the opponent. However, a trade-off between exploitation and compromise, is also of major importance. To clarify, if the agent never makes any concession, probably no agreement will be reached, or the opponent might even break-off somewhere within the negotiation process. Thus ρ is used to adaptively specify the time at which **Dragon** should stop exploiting the opponent and rather start to compromise.

The value of ρ should further increase with the increasing ratio between the number of new solutions to the total solutions proposed by the other party. Thus, we introduce γ_t to represent the ratio of new to all counter-offers over the past ten intervals up to t. The observation of new counter-offers cannot guarantee the concession by the other party (e.g., these new offers could just be the offers with high utility for the opponent while low utility for our agent). Therefore, the effect of γ_t is influenced by the maximum concession λ_t, leading to the following:

$$\rho = 1 - (1 - \gamma_t^{\alpha(1-\lambda_t)})t^{\varepsilon\delta^2} \tag{8}$$

where α is the parameter determining the impact of δ over time and ε controls the influence of λ_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:

$$E_{low} = \begin{cases} \theta & if \;\; \theta \geq maxU(E_\delta(0, t_l)) \\ maxU(E_\delta(0, t_l)) & otherwise \end{cases} \tag{9}$$

where θ is the reservation value specified by the preference, *maxU* returns the maximal utility from counter-offers, t_l is the last time the update was carried out.

Based on the above definitions, $R(t)$ is now defined as:

$$R(t) = E_{low} + \left(1 - t^{\frac{1}{(1-\rho)^\beta}}\right)(u_{max}^P - E_{low})\cos\left(\frac{1-\delta}{\omega}\right) \tag{10}$$

where β is the concession factor affecting the concession rate, u_{max}^P is the maximum utility of the given preference P in a domain, and ω is the weight which reflects the impact of the discounting factor to the concession degree.

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:

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

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:

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

where, $\hat{u}$ is defined as:

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

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:

$$\varphi = 1 - \frac{D_\delta(R(t_\nu), t_\nu) - E_\delta(t_\nu)}{\xi \cdot \sqrt{1-\delta} D_\delta(u_{max}^P, t_\nu)}, \quad t_\nu \in [t_c, t_s] \tag{14}$$

where ξ indicates the acceptance tolerance for the pessimistic forecast (i.e. a larger value enables our agent to bear with worse expectation) and t_ν is given by

$$t_\nu = argmin_{t \in [t_c, t_s]}(|E_\delta(t) - D_\delta(R, t)|) \quad . \tag{15}$$

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" counter-offer, it may lose the opportunity to reach a fairly good agreement earlier.

In the acceptance case, $\hat{u}$ and $\hat{t}$ are defined as $E_\delta(t_\nu)$ and t_ν, respectively. Otherwise, $\hat{u}$ is defined as -1, meaning it does not have an effect, and $R(t_c)$ is used for the target utility u'. 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, it is not rational to concede immediately to $\hat{u}$ when $u_l \geq \hat{u}$, where u_l is the utility of last bid before **Dragon** performs predictions at time t_l. It is also not appropriate for an agent 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 in order to be able to "grasp" every chance to maximize profit. Overall, the process to set u' is shown in line 11, which is calculated as follows :

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

4.3 Responding component

This is the last component of the **Dragon** strategy and corresponds to lines $12 - 18$ of Algorithm 1. After the expected utility u' has been determined, the agent needs to examine one of two conditions in response to the opponent. In the first the agent has to validate whether the utility of the counter-offer $U(O_{opp})$ is better than u', while in the second the agent has to determine whether the opponent had already proposed this offer 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 [2, 9]). In this work, **Dragon** considers that the opponent is rational. More precisely, **Dragon** assumes that the sequence of counter-offers 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 $g(\cdot)$ the two-valued function, whose return is 1 if the specific issue value (i.e., v_{jk}) appears in the counter-offer 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 which has an utility within some range around

u'. 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'. Thus it is reasonable that an agent selects an offer whose utility is in the narrow range [(1-0.005)u',(1+0.005)u']. 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 $\delta \neq 1$. Here, the responding component investigates whether the maximum expectation obtained from SPGPs is larger than θ. If this is the case, the agent expects to gain a better outcome than what the disagreement solution generates. Therefore, it would continue the bargaining. On the other hand, if the previous condition is not met, **Dragon** sets η as the probability of terminating the negotiation session which is calculated according to:

$$\eta = \theta^{\delta} \tag{18}$$

Given this probability, **Dragon** decides whether to leave the current negotiation session or not. Moreover, when the breaking-off decision is made and to be on the safe side, the agent waits a silence period, where none of counter-offers of the most recent 3 intervals are better than the previous best counter-offer. **Dragon** then quits the negotiation process (i.e., before the deadline).

5. EMPIRICAL EVALUATIONS

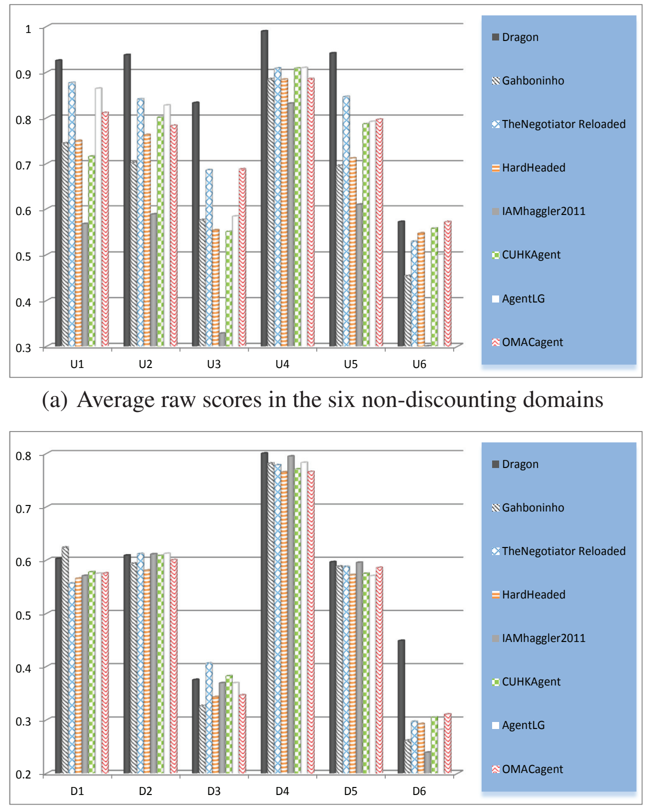
The performance evaluation of **Dragon** is done with the GENIUS simulation environment [11]. GENIUS allows to compare different agents across a variety of application domains under real-time constraints. The preference profiles of two negotiating parties are specified in correspondence to the individual domains. The assessment quality under which we evaluate the performance of the proposed strategy is the mean score in the context of tournaments. The competition results are reported in Section 5.2. Moreover, an empirical game theoretic evaluation is used to study the robustness of the strategy. The details of the latter evaluation are described, as needed, in Section 5.3.

The experimental technicalities including the implementation details of **Dragon** referred to as **Dragon-agent** are described next.

5.1 Experiment setup

In order to evaluate **Dragon** in a highly competitive setting, we run its implementation (**Dragon-agent**) against the top four agents of the ANAC-2012 competition and the top three of the ANAC-2011 competition. Moreover, the negotiations are conducted in the same six domains as in [21]. Additionally, in order to have an even more comprehensive evaluation scheme, three domains of the 2012 ANAC final round are also considered, namely, *IS BT Acquisition*, *Fitness*, and *Fifty fifty*. They are chosen to serve as the respective representatives of easy, medium, and hard domains of the 2012 competition (classfied according to the mean score of all participants achieved in a domain, see [1] for more details). To evaluate the agent's performance under no and high time pressure, both a non-discounting and discounting version of the domains are considered.

For convenience, we refer to the non-discounting domain *Travel* as U_1, *England vs Zimbabwe* as U_2, *Itex vs Cypress* as U_3, *IS BT Acquisition* as U_4, *Fitness* as U_5, *Fifty fifty* as U_6. Their corresponding discounting domains are labeled as $D_1 \ldots D_6$. All discounting domains are equipped with the identical discounting factor of 0.5. Please note that similar results for other values of discounting factor are observed. For the results to be statistically significant, we ran the GENIUS tournament mode 20 times for each domain.



(a) Average raw scores in the six non-discounting domains

(b) Average raw scores in the six discounting domains

Figure 2: Average raw scores of all agents in the twelve domains. The vertical axis represents utility and horizontal axis represents domain.

Dragon-agent simply sets ζ to be 180 which is equal to the number of seconds. The lead time ϱ is limited to 25 intervals, and the two parameters α and ε for compromise point are set to 0.5 and 10, respectively. The parameters for the expectation function R are set to $\beta = 2$ and $\omega = 1.2$. Finally $\xi = 0.95$. It is worth noting, that the agent shows robust behavior to different parametric settings.

5.2 Simulation results

The results achieved by every agent in terms of raw score in non-discounting domains and discounting domains are shown in Figure 2(a) and Figure 2(b), respectively.

As depicted in the figures, **Dragon-agent** demonstrates excellent performance against a variety of opponents in various scenarios. Clearly the **Dragon-agent** is ranked as number one in nine domains. For the remaining three where it does not take the first place, the performance of **Dragon-agent** is only marginally (around 3.8% on average) below the best performer. More precisely, **Dragon-agent** achieves the first place with 33.9% higher performance than the mean normalized score[2] of opponents in non-discounting domains. In contrast, the second ranked agent – TheNegotiator Reloaded, only obtains 85% score of ours. For discounting-domains, **Dragon-agent** also finishes first and has an advantage of 11.2% over the

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

Table 1: Performance in non-discounting and discounting domains.

Agent	*Discounting*	*Non-discounting*
	mean	*mean*
Dragon-agent	**0.776**	**0.835**
TheNegotiator Reloaded	0.698	0.712
CUHKAgent	0.695	0.662
OMACagent	0.686	0.693
AgentLG	0.681	0.663
Gahboninho	0.669	0.593
HardHeaded	0.668	0.624
IAMhaggler2011	0.667	0.416

Table 2: Overall Performance over domains. The bounds are based on the 95% confidence interval.

Agent	*Normalized Score*		
	Mean	*Lower Bound*	*Upper Bound*
Dragon-agent	**0.806**	**0.781**	**0.830**
TheNegotiator Reloaded	0.706	0.690	0.721
OMACagent	0.690	0.674	0.706
CUHKAgent	0.678	0.653	0.703
AgentLG	0.672	0.648	0.696
HardHeaded	0.646	0.611	0.652
Gahboninho	0.631	0.611	0.651
IAMhaggler2011	0.542	0.524	0.560

second best agent. More details can be found in the Table 1, which illustrates the mean normalized score of all agents in terms of normalized results averaged over the discounting domains and non-discounting domains, respectively.

According to the overall performance shown in Table 2, **Dragon** is the best strategy. With the average normalized score of 0.806, it leads a margin of 23.5 over the mean score of opponents across all domains. OMACagent, CUHKagent and AgentLG come in subsequent places, followed by the best agents of the 2011 ANAC. Interestingly, **Dragon-agent** leads by 17.5% over the mean score of the group consisting of the four best agents from the 2012 ANAC. This margin is even larger for the best agent group of the 2011 ANAC, which exceeds 30%. It is worth noting that this ranking is different from the final results of the 2012 ANAC. We speculate that the reason relates to the advantages gained by **Dragon** once competing with these opponents. To summarize, **Dragon** significantly outperforms with a high margin the state-of-the-art automated negotiators in a variety of application scenarios.

Another interesting observation to be made is the noticeable gap between **Dragon-agent** and IAMhaggler2011. In more details, this agent on average achieves less than half the performance of ours. Unlike **Dragon**, IAMhaggler: (1) applies Gaussian process as a prediction tool and (2) adapts its concession rate fully on the basis of global predictions. The experimental studies suggest that a reason for this performance gap lies in the global prediction view. Namely, this view seems to be vulnerable to "irrational concession making" induced by pessimistic predictions. **Dragon** already avoided such a behavior as explained in Section 4.2. The phenomenon of irrational concession becomes increasingly apparent when IAMhaggler2011 bargains with opponents in non-discounting domains where other players have no pressure to make concession earlier.

5.3 Empirical game theoretical analysis

In the previous subsection, we investigated the strategy performance from the usual mean-score perspective. This, however, does not reveal information about the robustness of these strategies. To address robustness appropriately, empirical game theory (EGT) analysis [13] is applied to the competition results. Here, we consider the best single-agent deviations as in [21], where there is an incentive for one agent to unilaterally change the strategy in order to statistically improve its own profit. The aim of using EGT is to search for pure Nash equilibria, in which no agent has an incentive to deviate from its strategy by choosing another one.

We applied the EGT technique to the scenarios where exactly two players are involved (which corresponds to the common format of bilateral negotiation) and each agent is allowed to choose one strategy from the eight strategies considered in our experiments. For brevity, let the initial letter of each strategy be the identifier (e.g., **O** means OMACagent) and S be the strategy set, that is, $S = \{\mathbf{D}, \mathbf{T}, \mathbf{O}, \mathbf{C}, \mathbf{A}, \mathbf{H}, \mathbf{G}, \mathbf{I}\}$. A profile is defined as the two strategies used by players in the game. Furthermore, the score of a specific strategy in a specific profile is calculated as its averaged payoff achieved when playing against the other strategy in all domains. The results are depicted in Figure 3. Under this EGT analysis, there exists only one pure Nash equilibrium, namely, the strategy profile ($\mathbf{D}^\star$ v.s. **T**), i.e., **D**ragon versus **T**heNegotiator Reloaded. This observation is of great interest, as it indicates that this strategy profile is the most stable profile among all possible profiles. For any non-Nash equilibrium strategy profile there exist a path of statistically significant deviations (strategy changes) that leads to this equilibrium. When compared with the other strategy in the equilibrium, **Dragon** is always preferred unless the current profile already contains it, which creates for a player an incentive to deviate for 80% of the state transitions. Moreover, the equilibrium profile constitutes a negotiation solution with the highest social welfare (i.e., the largest sum of scores achieved by two strategies). This is desirable because, as a measure of the negotiation benefit for all participants rather than the benefit for an individual agent, higher social welfare results in a better overall value of a negotiation.

6. CONCLUSIONS

This work introduced an effective strategy called **Dragon** for automated negotiation in complex bilateral multi-issue, no prior knowledge, time-constrained, low computational load, influence of reservation value scenarios. This strategy, based on Sparse Pseudo-Inputs Gaussian processes and an adaptive decision-making scheme, outperformed the best agents of the most recent International Automated Negotiation Agents Competition (ANAC). It is clear that the **Dragon-agent** is more efficient than the others. Experiments show that **Dragon** generates a higher mean scoring compared to the state-of-the-art negotiation agents. The main competitor to the new strategy (i.e., IAMHaggler2011) is ranked last. We speculate that the main reason for this performance difference relates to the "irrational concession" problem that **Dragon** automatically avoids. Further analysis based on empirical game theory clearly manifest the robustness of the proposed strategy. The exceptional results justify to invest further research efforts into this approach. In the future, we plan on comparing the opponent modeling scheme with the other available approachers and further, extend this framework to other negotiation settings, such as concurrent negotiation or multilateral negotiation.

7. ACKNOWLEDGMENTS

We would like to thank the China Scholarship Council (CSC) for providing a PhD scholarship to Siqi Chen. Also, we are grateful for the kind help from Shuang Zhou, useful discussions provided by the DKE Swarmlab members and the valuable comments of the referees.

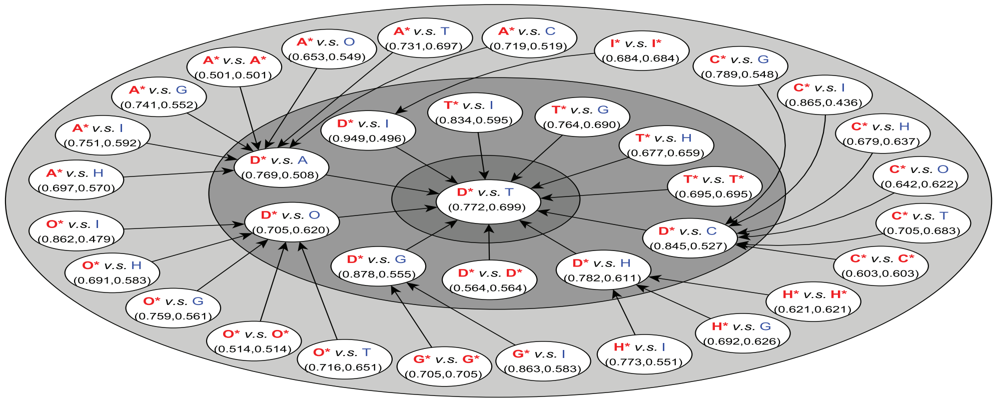


Figure 3: Deviation analysis for two-player negotiation. Each node shows a strategy profile and the scores of two involved strategies with the higher scoring one marked by a star. The arrow indicates the statistically significant deviation between strategy profiles.

8. REFERENCES

[1] The Third International Automated Negotiating Agent Competition (ANAC 2012). `http://anac2012.ecs.soton.ac.uk/results/final/`, 2012.

[2] T. Baarslag, K. Fujita, E. H. Gerding, K. Hindriks, T. Ito, N. R. Jennings, C. Jonker, S. Kraus, R. Lin, V. Robu, and C. R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 2013.

[3] J. Brzostowski and R. Kowalczyk. Predicting partner's behaviour in agent negotiation. In *Proceedings of the Fifth Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 355–361. ACM, 2006.

[4] R. Carbonneau, G. E. Kersten, and R. Vahidov. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Syst. Appl.*, 34:1266–1273, February 2008.

[5] S. Chen and G. Weiss. An efficient and adaptive approach to negotiation in complex environments. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, pages 228–233, Montpellier, France, 2012. IOS Press.

[6] R. M. Coehoorn and N. R. Jennings. Learning on opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th Int. conf. on Electronic commerce (ICEC 2004)*, pages 59–68. ACM, 2004.

[7] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(4):159–182, 1998.

[8] P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

[9] J. Hao and H. fung Leung. ABiNeS: An adaptive bilateral negotiating strategy over multiple items. In *Proceedings of the 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2012)*, China, 2012.

[10] M. Hendrikx. A survey of oppnent models in automated negotiation. Technical report, Delft University of Technology, The Netherlands, September 2011.

[11] K. Hindriks, C. Jonker, S. Kraus, R. Lin, and D. Tykhonov. Genius: negotiation environment for heterogeneous agents. In *Proceedings of the 8th international joint conference on Autonomous agents and multiagent systems (AAMAS 2009)*, pages 1397–1398. ACM, 2009.

[12] 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 (AAMAS 2008)*, pages 331–338. ACM, 2008.

[13] P. R. Jordan, C. Kiekintveld, and M. P. Wellman. Empirical game-theoretic analysis of the tac supply chain game. In *Sixth International Joint Conference on Automomous Agents and Multi-Agent Systems (AAMAS 2007)*, pages 1188–1195. ACM, 2007.

[14] 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*, 172:823–851, April 2008.

[15] F. Lopes, M. Wooldridge, and A. Novais. Negotiation among autonomous computational agents: principles, analysis and challenges. *Artificial Intelligence Review*, 29:1–44, 2008.

[16] H. Raiffa. *The art and science of negotiation*. Harvard University Press Cambridge, Mass, 1982.

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[18] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.

[19] S. Saha, A. Biswas, and S. Sen. Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the Fourth international joint conference on Autonomous agents and multiagent systems (AAMAS 2005)*, pages 397–403. ACM, 2005.

[20] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18 (NIPS 2006)*, pages 1257–1264. MIT press, 2006.

[21] C. Williams, V. Robu, E. Gerding, and N. Jennings. Using gaussian processes to optimise concession in complex negotiations against unknown opponents. In *Proceedings of the 22nd Internatioanl Joint Conference on Artificial Intelligence (IJCAI 2011)*. AAAI Press, 2011.