Noname manuscript No.
(will be inserted by the editor)

PredictRV : A Prediction Based Strategy for Negotiations with Dynamically Changing Reservation Value

First Author · **Second Author**

Received: date / Accepted: date

Abstract Negotiation is an important component of interaction process among humans. With increasing automation, autonomous agents are expected to take over a lot of this interaction process. Much of automated negotiation literature focuses on agents having a static and known reservation value. In situations involving dynamic environments e.g., an agent negotiating on behalf of a human regarding a meeting, agents can have a reservation value (RV) that is a function of time. This leads to a different set of challenges that may need additional reasoning about the concession behavior. **In this paper, we build upon Negotiation algorithms such as OptimalBidder and Time Dependent Techniques such as Boulware** which work on settings where the reservation value of the agent is fixed and known. Although these algorithms can encode dynamic RV, their concession behavior and hence the properties they were expected to display would be different from when the RV is static, even though the underlying negotiation algorithm remains the same. We therefore propose to use one of Counter, Bayesian Learning with Regression Analysis or LSTM model on top of each algorithm to develop the PredictRV strategy and show that PredictRV indeed performs better on **two different metrics** tested on two different domains on a variety of parameter settings.

Keywords First keyword · Second keyword · More

F. Author
first address
Tel.: +123-45-678910
Fax: +123-45-678910
E-mail: fauthor@example.com

S. Author
second address

1 Introduction

Negotiation is an important component of interaction process among humans [Raiffa(1982),Sandholm et al.(1995)Sandholm, Lesser et al.,Rosenfeld and Kraus(2018)]. A lot of negotiation literature assumes that we have a good amount of information about our own choices [Kraus(2001),Fatima et al.(2004b)Fatima, Wooldridge, and Jennings] and reservation value (**RV**), while not knowing our opponents preferences [Hindriks and Tykhonov(2008),Choi et al.(2001)Choi, Liu, and Chan,Coehoorn and Jennings(2004)]. Note that RV refers to the utility of a bid in the negotiation, below which we would not be willing to accept any bid. Reasons for not accepting a bid whose utility is below RV can be due to a better BATNA - Best Alternative to Negotiated Agreement [Crump and Moon(2017)] (so RV maybe set to BATNA) or that the agent receives a utility it perceives as not being good enough for it to accept. In settings where the environment is dynamic, there can be situations where our own RV can change with time (while the preference profile is static) [Li et al.(2013)Li, Vo, Kowalczyk, Ossowski, and Kersten]. We may not know how the changes would pan out e.g., an agent acting on behalf of a meeting attendee may have varying estimates on when the human may arrive for the meeting [Scerri et al.(2002)Scerri, Pynadath, and Tambe, Chalupsky et al.(2002)Chalupsky, Gil, Knoblock, Lerman, Oh, Pynadath, Russ, and Tambe]. Dynamicity of RV can therefore throw additional challenges when we are unaware of the nature of changes (which is different from RV changing because of a discount factor where the change is computable). Bids that simply react to the dynamicity may not be sufficient since they can change in a random fashion and result in lower utility. For example, it can be hard to agree on a meeting time if an agent acting on behalf of a human declares that the human would arrive in 30 minutes and in a short period re-declare that the human would arrive in 10 minutes and then quickly change to say 20 minutes even though the agent may simply be acting based on its belief of when the human would arrive.

Making concessions to reach an agreement is an important part of negotiation process [Jennings et al.(2001)Jennings, Faratin, Lomuscio, Parsons, Wooldridge, and Sierra, Fatima et al.(2014)Fatima, Kraus, and Wooldridge,Rosenschein and Zlotkin(1994)]. There are a variety of ways in which negotiating agents can concede. One such category of techniques is Time Dependent Tactics (TDT's) [Faratin et al.(1998)Faratin, Sierra, and Jennings, Fatima et al.(2004a)Fatima, Wooldridge, and Jennings] e.g., Boulware and Conceder agents. [Baarslag et al.(2016)Baarslag, Hadfi, Hindriks, Ito, and Jonker] presents an **Optimal Non-Adaptive Concession (ONAC)** and **(cite the ref): presents an OptimalBidder** algorithm with incomplete information where time pressure (amount of time to deadline) is a primary criteria to influence the concession behavior. Negotiation algorithms such as **OptimalBidder** and Boulware [Baarslag et al.(2016)Baarslag, Hadfi, Hindriks, Ito, and Jonker] work on settings where RV of the agent is fixed and known. Although these algorithms can work with (or be modeled as a function of) a dynamic RV, their concession behaviors can have lot more randomness or fluctuations compared to when they have a static RV. For purposes of a more stable bidding behavior,

the agent should therefore make choices based on predicted (RV) values. While quality of agreement is a default metric used in negotiations, popular negotiation frameworks such as the Genius platform [Lin et al.(2014)Lin, Kraus, Baarslag, Tykhonov, Hindriks, and Jonker] do not support modeling of dynamic RV. We therefore had to develop a simple negotiation simulator that can encode dynamic RV. In addition to quality of agreement, we use **Prediction as an additional metric to evaluate the concession behavior.**
We propose to use the following models on top of negotiation algorithms, to handle the effects of a dynamic RV: (a) **Counter model** [Scheaffer and Young(2009)], (b) **Bayesian learning with Regression Analysis** [Yu et al.(2013)Yu, Ren, and Zhang, Zeng and Sycara(1998)] and (c) **LSTM model**. All three models are present in literature and we adapt them here to work suitably with the different negotiation algorithms. While the paper builds on top of **OptimalBidder** and Boulware algorithms, the procedure, in general, would be suitable to apply to algorithms that are sensitive to the dynamicity of RV (which results in fluctuations in bidding). Given that the models help to predict the RV to reduce the effect of dynamicity, we refer to the new strategy as **PredictRV**.

2 Related Work

The paper [Li et al.(2013)Li, Vo, Kowalczyk, Ossowski, and Kersten] talks about negotiations in a setting where the environment is dynamic. This paper integrates important concepts of BATNA and agents concession and resistance strategy into automated negotiation. Instead of focusing on an isolated negotiation process, the proposed framework enables the agents to search for outside options, and thus, proactively improve their BATNA during negotiation. As their new BATNAs becoming available, the agents will then dynamically incorporate this information to update their resistance and concession forces in negotiation, and eventually, leading to a rational change towards their negotiation strategies and decisions. This paper restricts the use of information obtained from the dynamic environment to BATNA that would help the agent in its strategy.

3 Static RV

3.1 Negotiation Model

The negotiation model we use follows the alternating multiple offers protocol [Rubinstein(1982)]. The model follows an all accept strategy that is all other agents apart from the bid offering agent should accept the bid for an agreement. Consider three agents A, B, C with utility functions $U_A(z)$, $U_B(z)$, $U_C(z) \in [0,1]$ where z belongs

to the set of all possible negotiation outcomes for a domain D. The RV's for the agents would be rv_A, rv_B, $rv_C \in [0, 1]$. The agents will propose offers with utility higher than their RV. The model is implemented in the following way: In round j, agent A makes a bid B_A and agent B, C make a decision to either accept it or reject it. If the bid is rejected by atleast one agent, the bid is considered rejected and now agent B makes a bid and its evaluated by agent A, C. If this is rejected by one of the agent then agent C makes a bid and it is evaluated by agent A, B. If now the bid is rejected, the round increases to round j+1 and negotiation continues until a bid is accepted or the deadline is reached. A bid is considered accepted if all the agents accept the bid.

3.2 OptimalBidder

OptimalBidder only decides on the concessions, but is of course good when combined with acceptance components (optimal stop, for example). This algorithm uses the optimal stopping rule (cutoffs) for bidding. It sets a target utility of the bid to be placed at round j. (Text in tweeking the formula also has to be written yet)

$$\begin{aligned} &U_N = 0.25 + 0.25 * rv_A, \\ &U_j = 0.25 * (U_{j+1} + 1)^2, j \in \{1, 2, 3, ..., N-1\} \end{aligned} \tag{1}$$

3.3 Utility Generation for Boulware Algorithm

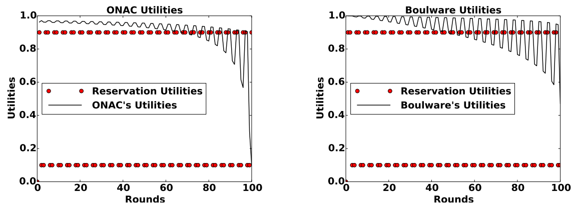
The Boulware algorithm is a TDT [Faratin et al.(1998)Faratin, Sierra, and Jennings, Fatima et al.(2004a)Fatima, Wooldridge, and Jennings], which concedes considerably more as the negotiation deadline approaches. TDTs consist of a family of functions that represent an infinite number of possible tactics, one for each value of β. The formula for this family of functions is as follows where j is the jth round and β should be in the range $(0, 1)$ for Boulware:

$$U_j = rv_A + (1 - rv_A) * (\frac{min(N-j, N)}{N})^{\frac{1}{\beta}} \tag{2}$$

4 Dynamic RV: The PredictRV Strategy

4.1 Negotiation Model

The negotiation model remains same as for static RV case with the following difference: Since agent A's RV is dynamic, it is represented as $rv_A(t)$.



(a) ONAC algorithm (b) Boulware algorithm

Fig. 1 Utilities obtained by using ONAC and Boulware algorithms

4.2 Algorithm One for Dynamic Reservation Values

The OptimalBidder algorithm present in negotiation literature assumes a static RV. For OptimalBidder that works with dynamic RV, utilities can be generated using the following function:

$$\begin{aligned} &U_N = 0.25 + 0.25 * rv_A(t), \text{ where t = j at round j} \\ &U_j = 0.25 * (U_{j+1} + 1)^2 \end{aligned} \tag{3}$$

4.3 Boulware for Dynamic Reservation Values

The Boulware algorithm present in negotiation literature assumes a static RV. For Boulware that works with dynamic RV, utilities can be generated using the following function:

$$U_j = rv_A(j) + (1 - rv_A(j)) * (\frac{min(N - j, N)}{N})^{\frac{1}{\beta}}, \text{ at round j} \tag{4}$$

4.4 Illustrative Example

Consider a toy example, where the RV can be either 0.1 or 0.9 and changes randomly every 2 rounds for a total of 100 rounds. Fig. 1 shows the concession curves obtained by using the **ONAC -¿ replace fig. with Optimal Bidder** and Boulware algorithms. The x-axis of each figure shows the number of rounds from 0 to 100 while the y-axis shows the utility values ranging from 0 to 1. The utilities of the bids at each round are computed using Eq. (3). The figures show that the concession curves are not monotonic due to the dynamic nature of the RV, which results in to and fro concessions being made, where peaks correspond to RV of 0.9 and troughs correspond to 0.1.

4.5 Steps of Strategy for PredictRV

Given a negotiation algorithm (like ONAC or Boulware):

1. **Generate hypotheses about the RV and assign weights to each hypothesis. Compute the utility for each hypothesis T_{x_i} [by setting U_N as the utility of the hypothesis and plugging in Eq. (3 , 4)].**

   For each round j from 1 to N (no. of rounds):
2. **Update weights of hypotheses based on the rv_A at that round i.e., $rv_A(j)$ [using Counter, Bayesian or LSTM approaches presented below]**
3. **Using the utility computed for each hypothesis in Step (1), we now compute the utility of the bid [Using one of Eqs. (6) or (13)].**

   End of for

To generate hypotheses (first step), we divide the range between which the RV can vary, into n number of intervals I_i for $i \in \{1, 2, 3, ...n\}$. A suitable point x_i is selected as a representative value for each interval I_i. If the RV falls within an interval it is classified as having utility of the point that represents the interval. We then compute negotiation algorithm utilities, $T_{x_i} = \langle U_1(x_i), U_2(x_i), ..., U_N(x_i)\rangle$ using Eq. (3). At the start all hypotheses are equally likely, hence each hypothesis is initialized with a probability $\frac{1}{n}$ i.e., uniform distribution over hypotheses. As the negotiation progresses we may have a better prediction over the hypotheses based on the past RVs, hence the probability distribution would change. The second step of the PredictRV strategy is to update weights of the hypotheses as the new round starts. The way in which the weights are updated depends on the actual procedure we use namely Counter, Bayesian Learning or LSTM models presented below.

4.6 Counter Learning

In the Counter based learning procedure, the count for each hypothesis is initialized as $c_{x_i} = 0$, where $i \in \{1, 2, 3, ...n\}$. At a new round j, we obtain a new RV. As step 2 of PredictRV, using the new RV we update the counter for the hypothesis that corresponds to the new RV. We re-compute the probability for each interval as follows:

$$p_{x_i} = \frac{c_{x_i}}{\sum_{i=1}^{n} c_{x_i}} \;, i \in \{1, 2, 3, ...n\} \tag{5}$$

As step 3 of PredictRV, using the probabilities computed on different intervals we compute the utility U_j to be bid by PredictRV as:

$$U_j = \sum_{i=1}^{n} p_{x_i} * T_{x_i j} \;, j \in \{1, 2, 3, ...N\} \tag{6}$$

4.7 Bayesian Learning with Regression Analysis (BLRA)

In the BLRA procedure presented in [Yu et al.(2013)Yu, Ren, and Zhang], the learning agent i has a belief about the p.d. of its opponent's negotiation parameters (i.e., the deadline and RV). As shown in step 1 of PredictRV, we have a belief over the hypothesis of our own (dynamic) RV. By keeping track of the history of values obtained for RV so far and comparing with fitted estimates derived from a regression analysis, the agent can revise its belief over the hypothesis by using a Bayesian updating rule and can correspondingly adapt its concession strategy.

4.7.1 Regression Analysis

As the negotiation proceeds [Yu et al.(2013)Yu, Ren, and Zhang], utility u_t for a TDT decreases according to the following decision function:

$$u_t = 1 - (\frac{t}{T})^{\beta} \tag{7}$$

where T is the deadline and β is the concession parameter. We adapt this terminology to express in terms of the agent A's own dynamic RV. We assume RV to be 0 at start of negotiation and vary according to Eq. (7).

$$u_t = u_0 + (u_T - u_0)(\frac{t}{T})^{\beta} \tag{8}$$

where u_T is the RV at the deadline and u_0 is the RV at the start. For every round we receive an RV for that round. We compute the regression line (fitted utilities) $\hat{RV}_{t_b} = \{\hat{u}_0, \hat{u}_1, \hat{u}_2, ..., \hat{u}_{t_b}\}$ based on the historical RVs $RV_{t_b} = \{u_0, u_1, u_2, ..., u_{t_b}\}$ until round t_b as follows:

Step 1 : Generate the hypotheses and initialize its probabilities as mentioned in Section 4.5 (Steps of strategy) with x_i representing utility of each hypothesis.

Step 2 : Based on Eq. (8), we use the following power regression function to calculate the regression curve:

$$\hat{u}_t = u_0 + (x_i - u_0)(\frac{t}{N})^{\beta} \tag{9}$$

where N is the deadline. Next, β is calculated using Eq. (10) (as proposed in [Yu et al.(2013)Yu, Ren, and Zhang]):

$$\beta = \frac{\sum_{k=1}^{t_b} t_k^* u_k^*}{\sum_{k=1}^{t_b} {t_k^*}^2}, \text{where} u_k^* = ln(\frac{u_0 - u_k}{u_0 - x_i}), t^* = ln(\frac{t}{N}) \tag{10}$$

Step 3 : Based on the calculated regression curve given by Eqs. (9) and (10), the fitted RVs $\hat{RV}_{t_b}$ would be $= \{\hat{u}_0, \hat{u}_1, \hat{u}_2, ..., \hat{u}_{t_b}\}$ at each round (where $\hat{u}_0 = u_0$).

Step 4 : We now calculate the non-linear correlation between RV_{t_b} and the fitted RVs $\hat{RV}_{t_b}$. The coefficient of non-linear correlation γ is given by Eq. (11), where $\overline{u}$ and $\overline{\hat{u}}$ are the average of all the historical and fitted RVs respectively:

$$\gamma = \frac{\sum_{k=1}^{t_b}(u_k - \overline{u})(\hat{u}_k - \overline{\hat{u}})}{\sqrt{\sum_{k=1}^{t_b}(\hat{u}_k - \overline{\hat{u}})^2 \sum_{k=1}^{t_b}(\hat{u}_k - \overline{\hat{u}})^2}}, \gamma_{new} = \frac{\gamma + 1}{2} \tag{11}$$

Step 5 : Parameter γ (-1 $\leq \gamma \leq$ 1) can be used for evaluating resemblance between chosen (x_i) and real RVs (u_t). To use γ as a probability to perform belief update in Bayesian Learning, we normalize to [0,1] (γ_{new} in Eq. (11)).

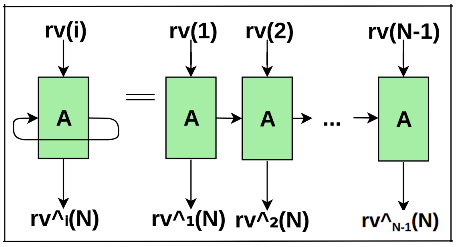


Fig. 2 LSTM Architecture

4.7.2 Bayesian Learning

Step 1: Bayesian Learning can be used if we have a hypothesis about the prediction. Belief about p.d. of these hypotheses can be revised through a posterior probability by observing the RV. Each hypothesis H_i represents that it would be the possible RV at the end of negotiation. The prior p.d., denoted by P(H_i), i ∈ (1,2,3,...,n) signifies the agent's belief about the hypothesis i.e., how likely the hypothesis matches the RV at the end of the negotiation.

Step 2: The agent can initialize the p.d. over hypotheses based on some prior information if available, otherwise a uniform distribution P(H_i) $= \frac{1}{n}$ is assigned. During each round of negotiation t_b the probability of each hypothesis would be computed using the Bayesian updating rule in Eq. (12):

$$P(H_i|RV) = \frac{P(H_i)P(RV|H_i)}{\sum_{k=1}^{n} P(RV|H_k)P(H_k)} \tag{12}$$

Step 3: The observed outcome here is historical RVs $RV_{t_b} = \{u_0, u_1, u_2, ..., u_{t_b}\}$. As presented in [Yu et al.(2013)Yu, Ren, and Zhang], the agent will update the prior probability P(H_i) using the posterior probability P($H_i|RV_{t_b}$), thus a more precise estimate is achieved using Eq. (12).

Step 4: As presented in [Yu et al.(2013)Yu, Ren, and Zhang], conditional probability P($RV_{t_b}|H_i$) is obtained by comparing the fitted points $\hat{RV}_{t_b}$ on the regression line based on each selected RV x_i, with the historical RVs RV_{t_b}. The more correlated fitted RVs are with historical RVs, the higher P($RV_{t_b}|H_i$) will be.

Step 5: Difference between the regression curve and the real RV sequence can be indicated by the non-linear correlation coefficient γ_{new}. Thus, we can use value of γ_{new} as the conditional probability P($RV|H_i$) in Eq. (12). The learning approach will increase probability of a hypothesis when the RV selected (x_i) is most correlated with the RV at end of the negotiation. As mentioned step 4 of PredictRV, using the probabilities computed on different intervals, we compute the utility at that round as:

$$U_j \;= \sum_{i=1}^{n} P(H_i) \; * T_{x_i j} \;,\; j \in \;\{1, 2, 3, ...N\} \tag{13}$$

4.8 LSTM based Prediction

LSTM (Long-Short Term Memory) [Hochreiter and Schmidhuber(1997)] is a popular recurrent neural network architecture to perform deep learning tasks and has been shown to be useful in time-series prediction. The negotiation problem introduced here can be modeled as a time series prediction task

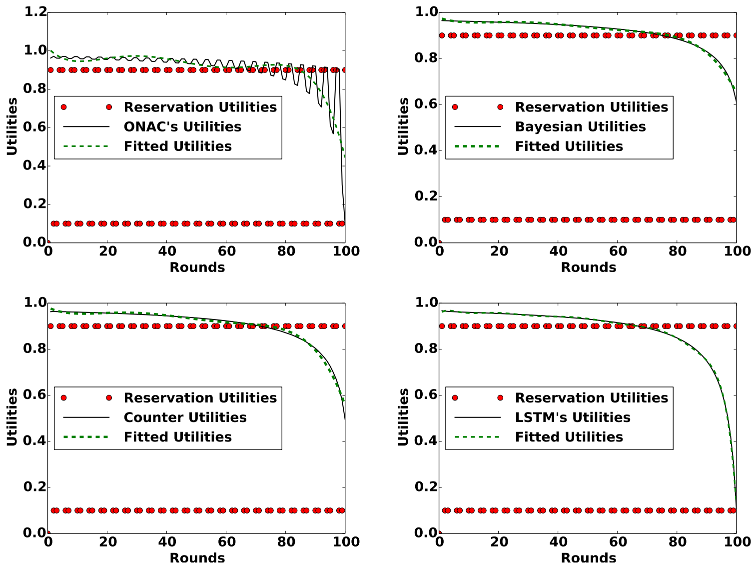


Fig. 3 Algorithms with their fitted curves

wherein the agent learns more information as the negotiation progresses. We therefore propose to use a LSTM based approach to predict the RV at the last time step n of the negotiation, using time-series forecasting. As shown in Fig. 2, the input at each time step t for LSTM is $RV(t)$ (i.e., RV provided by environment at t). Note that there exists a single LSTM cell A to which input is fed repeatedly (one value at every time step) along with output of the previous time step. Output at t is the predicted value for RV at the last time step n denoted by $\hat{RV}_t(n)$. The LSTM is trained using a mean squared error loss function and learns to predict better as the number of epochs increase. There are n hypotheses in our problem whose probability is updated every time step based on the predicted RV for the last time step $\hat{RV}$. This is similar to Counter model where we identify the interval the $\hat{RV}$ falls into and increase count of that hypothesis by 1 (Eq. (5)). Using the probabilities computed for different hypotheses we compute the utility to be bid by PredictRV (Eq. (6)).

5 Example continued

The rest of the example is explained using the ONAC-D algorithm(ONAC-D is ONAC strategy without any changes applied to Dynamic RV). Fig. 3 shows the utility values generated by Counter, BLRA and LSTM models computed using Eqs. (6) and (13) respectively. The x-axes shows the number of rounds from 0 to 100 while the y-axes shows the utility values ranging from 0 to 1.

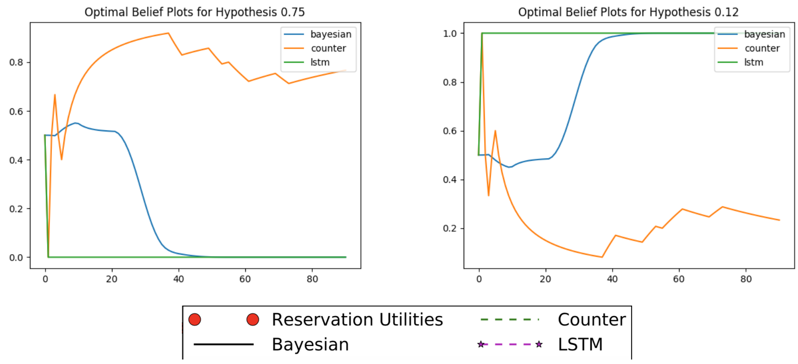


Fig. 4 Belief Plots for two hypotheses

Fig. 4 shows the belief plots for the three models. A belief plot shows how the belief in a particular hypothesis changes as the rounds progress. The figure shows two plots corresponding to the two hypotheses that the RV is 0.1 (hypothesis 0.1) and 0.9 (i.e., hypothesis 0.9). The x-axes for both the figures show the number of rounds from 0 to 100 and the y-axes show the probability of belief in the hypothesis that the figure represents e.g., a y-axis value of 0.3 in figure on left implies that an algorithm believes that the RV is 0.1 with a probability 0.3 which implies that other hypotheses are true with rest of the probability (in this case only other hypothesis is hypothesis 0.9). The belief plots show that: (a) For hypothesis 0.1, while Counter stays close to middle (probability of 0.5), BLRA and LSTM are more clear in their belief for this hypothesis (former converges to close to 0 while the latter converges to close to 1 probability and stay with these probabilities once converged) showing the inherent differences between the models. (b) Counter converges quickly to a belief of 0.5 since RV alternates between the hypotheses every 2 steps, hence the count is more or less balanced. (c) For BLRA, belief in hypothesis 0.1 converges close to 0 since it is not just the count but the time when the RV changes come into play here. (e) For LSTM, belief in hypothesis 0.1 converges to close to 1 faster than other models, however to the opposite belief of BLRA for this example. The outcome utility for **BLRA, Counter, LSTM and Optimal** are **0.929, 0.784, 0.756, 0.765** respectively.

6 Experiments

6.1 Setup for the Experiments

We have number of hypotheses, number of rounds of negotiation N and update rate (frequency of change in RV) as the parameters of our algorithm. N is fixed to 100 for all experiments. Experiments were performed on the Fire Disaster Response and Meeting Scheduling domains. In both these domains, the agent is faced with a dynamic RV. For purposes of experimentation, we model the dynamic RV using a Markov chain model [**we omit the specifics of our modeling due to space constraints**]. The number of hypotheses vary across the domains. Update rate of RV is varied among the values $\{2, 5, 10, 20, 50\}$. We run each experiment for 100 iterations keeping the parameters constant. The x-axis shows the (hypothesis, update rate) while y-axis shows the **Outcome Utility** metric in each plot. **We run the experiments for 4 agents and 5 agents multi agent negotiations**

6.2 Metrics

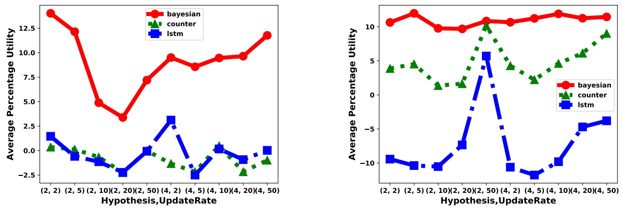
1) **Outcome Utility Metric**: We run negotiations for agent A vs agent B, where A uses one of **ONAC-D** or Boulware-D and B is PredictRV strategy (Counter, BLRA or LSTM). We average the outcome over 100 iterations and compute the outcome utility for each UpdateRate and hypothesis (averaged utility represented as **OD** for **ONAC-D**, C for Counter, B for BLRA and L for LSTM). We then compute the utility of PredictRV w.r.t **ONAC-D** using Eq. (14) (represented in graphs as Average Percentage Utility):

$$\text{percentage utility of i} = \frac{i - OD}{OD} * 100, i \in \{C, B, L\} \tag{14}$$

2) **Prediction Metric**: **We allow each model to train till the end of the negotiation (N rounds). At the last round N, we have a RV predicted by each of the models i.e $\hat{RV}$ for round $N+1$ (which is not part of the negotiation). For each of the models, we then compute the difference between $\hat{RV}$ and the actual RV at round $N+1$ which is used to capture the quality of prediction. This value is averaged over 100 iterations where a lower difference in average value implies a better prediction. (Highlighted this , wanting to remove this metric in JOURNAL.)**

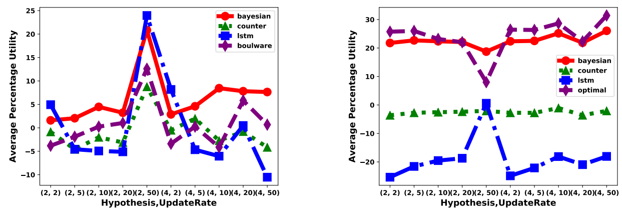
6.3 Fire Disaster Response

Consider a scenario where there is a forest fire and the fire can spread quickly in any of the 4 directions i.e., North, South, East or West. Assume that the forest is modeled as a grid of size $n_1 * n_1$ [Heyman and Sobel(1982)]. Given that the forest is spread over a large area, fire fighting units are dispatched to many locations in the forest to fight the fire. We refer to these as local units. The commander in-charge of the entire operation has a set of resources and the local unit leaders request the commander for resources based on their assessment of fire at the local location. We model this scenario as a negotiation from the perspective of one local unit leader (called the agent) and the commander. The terrain of fire at the local point is such that it is harder to tackle the fire in some directions than the others i.e., more resources would be needed to



(a) Optimal Bidding Strategy (b) Boulware Strategy

Fig. 5 Outcome Utility in Fire Domain with random start points for **4 agents**



(a) Optimal Bidding Strategy (b) Boulware Strategy

Fig. 6 Outcome Utility in Fire Domain with random start points for **5 agents**

stop the fire in certain directions than the others. The minimum number of resources required (i.e., RV, since allocation of resources below this would not be enough to stop the fire) depends on which direction the fire moves.

From the agent's perspective, it would be good to obtain much higher number of resources so as to stop the fire quickly at the local point. Given that the direction of the fire changes in different time steps, the RV's and its utility are dynamic i.e. change with time. Different number of resources are needed to stop the fire when it moves in different directions {North, West, East, South} which determines the RV. The commander on the other hand has a global picture of the fire in mind and would like to reduce the resources given to each local point. More resources committed to one local point would mean less resources for another point and once a resource is committed and moved to that point there is a huge delay and cost involved in moving it to another point. Hence, we have a scenario where there are conflicting preferences and the agent needs to negotiate in a way that gives a reasonable estimate of the needed resources as it negotiates rather than changing bid every couple of steps as the fire changes its direction.

We operationalize the experimental parameters as follows: A negotiation is being carried out with N (= 100) as deadline. The parameters here are number of hypotheses, update rate of the RV and grid size. The number of hypotheses are varied among the values $\{2, 4\}$ i.e., {North, South} or {North, West, East, South} directions with $\{0.75, 0.15\}$ and $\{0.75, 0.57, 0.32, 0.15\}$ (corresponding to number of resources {12, 10, 7, 4}) as the values for RV. Experiments were performed with the local location start point, as a random point around the center of the grid (upto a radius of 4 units from the center). Fig. 5 shows two plots corresponding to ONAC and Boulware with a random start point for fire with grid size 100. Both plots show that the values for outcome utility metric (Sec 6.2) for PredictRV are higher than for ONAC-D or Boulware-D respectively e.g., plot (b) of Fig. 5 shows that the Average Percentage Utility for BLRA varies from 1% at lowest to 20% at highest. **The plot also shows that the overall Average Percentage Utility across all the intervals and update rates for BLRA is 9.05% while it is -0.86% for Counter and -0.26% for LSTM.**

6.4 Meeting Scheduling Domain

Meeting scheduling is a popular multi-agent domain due to the complexity and richness of interactions needed among automated agents. E-Elves is once such meeting scheduling model/application for scheduling meetings between people. In this application, each person has an agent acting on his/her behalf, to perform tasks such as scheduling a meeting, delay a meeting in real-time if needed and so on. We focus on the delay action for our purposes. Many times it can happen that a meeting is scheduled but one of the participants gets delayed to the meet. An agent acting on behalf of this person, now needs to take an action so other participants can continue with productive tasks instead of wasting their time for the meeting to start. However, the delay action is not an easy decision for the agent to take.

In the E-Elves application, the agent can check with the participant for his/her anticipated delay if the participant is not at the meeting by the start time of the meet. If the participant responds with a specific delay, the agent can inform the other participants of the new meeting time. However, if the participant does not respond the situation can be pretty complicated for the agent. The agent would not know how much delay to propose to the other participants. The E-Elves application assumes that the agent can track the participant's location and hence can come to an estimate of the possible delay. Let's assume that the agent estimates that a participant would be delayed by 10 minutes and announces it to other participants. The agent tracks the location for a few minutes and realizes that the participant is actually moving away and hence realizes that the delay will be much larger say 40 minutes. So after a fixed interval (say 3 minutes) it announces to the group the new delay is 40 minutes. However it very quickly realizes that the participant has changed direction and is moving towards the place of meeting. So 3 minutes after the previous announcement, the agent again announces to the group that the participant will be there in 10 minutes. As it tracks the participant, it realizes that an accident came up on that route and there will be 20 minutes additional delay, so it again announces and so on.

This is a scenario where an agent is identifying the best possible estimate every couple of minutes and notifying the group. Unfortunately, too much of fluctuation in the meeting time spoils the productive time for other people

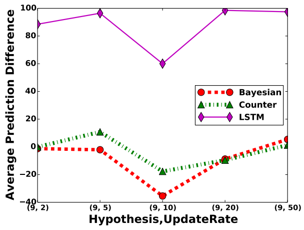
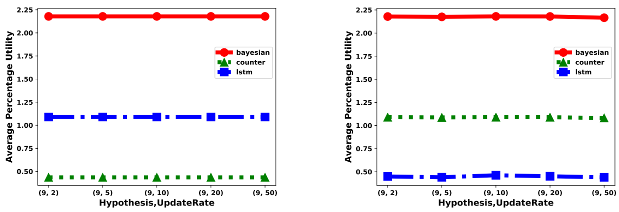
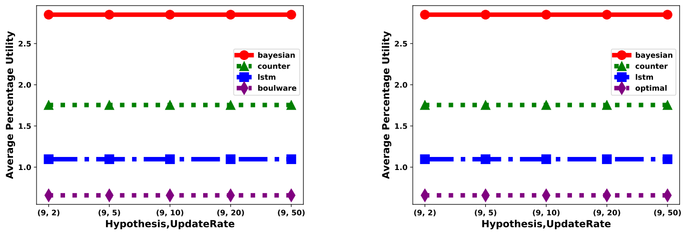


Fig. 7 Prediction in the Meeting Domain



(a) Optimal Bidding Strategy (b) Boulware Strategy

Fig. 8 Outcome Utility in **Meeting for 4 agents**



(a) ONAC Strategy (b) Boulware Strategy

Fig. 9 Outcome Utility in **Meeting for 5 agents**

and inspite of the agent acting rationally from the perspective of information it has, others may not necessarily think that it is a rational behavior i.e., it can be viewed as waveriness in decision-making on part of the agent. Other agents in this scenario can propose alternate meeting times or accept the current bid. The delay estimates determine the RV of the agent since announcing an earlier meeting time would mean that the meet would start without the participant while a much later start time would be bad for other participants although it maybe a safe response for the agent. It is also a bad scenario if the agent keeps changing the meeting time, hence we have a situation where the agent has to balance conflicting preferences while proposing bids on start time of the meeting. The negotiation is assumed to end once the meeting starts or gets canceled.

We operationalized parameters for this domain from the E-Elves[Scerri et al.(2002)Scerri, Pynadath, and Tambe] application. The parameters for the algorithm are update rate of the RV, delay intervals and number of hypothesis. There are 9 possible delay intervals we consider here i.e $\{5, 10, 15, 20, 25, 30, 35, 40, 45\}$ minutes. A delay interval of 10 minutes means that a meeting supposed to start at 10 am is now rescheduled to start at 10:10 am. For this domain each hypothesis corresponds to a delay interval, hence there are 9 hypotheses corresponding to the 9 delay intervals. The overall value of the meet is computed as below:

$$\begin{aligned} &Delaycost = (delay^{alpha}) * 2 \\ &Valueofthemeet = 200 \\ &Overallvalue = Valueofmeet - Delaycost \end{aligned} \tag{15}$$

where $delay$ is delay w.r.t. the scheduled starting time and $alpha \in \{1.0, 1.2, 1.4, 1.6\}$. Utility of the hypothesis is calculated by normalizing the reward obtained using Eq. (15).

Fig. 8 shows two plots corresponding to ONAC and Boulware with a random start point for fire with grid size 100. Both plots show that the values for outcome utility metric (Sec 6.2) for PredictRV are higher than for ONAC-D or Boulware-D respectively e.g., plot (b) of Fig. 8 shows that the Average Percentage Utility for BLRA is 2.25%. The plot also shows that the overall Average Percentage Utility across all the intervals and update rates for BLRA is 2.25% while it is 1.0% for Counter and 0.5% for LSTM.

The prediction measurement for the meeting domain is shown in Fig. 7. Experiments for measuring prediction for the meeting domain were performed with the following summary (we skip graphs due to space issues): The overall average percentage prediction across all the intervals and the update rates for Counter, BLRA and LSTM are -3.05, -8.27 and 88.12 respectively.

6.5 Summary of the Experiments

In Fig. 10, 1 is the best performing model and 2 is the second-best performing model for given metric and domain, $x\%$: how much better the best model is relative to the second-best model. Formulation: a = metric value of best model, b = metric value of 2nd best model. For outcome utility relative performance $= 100 * \frac{(a-b)}{a}$, $(a > b)$. **For the prediction metric, relative performance = 100 ***$\frac{(b-a)}{b}$ **,** $(b > a)$**. ABS - Anti Base Strategy, i.e for PredictRV if OptimalBidder is used ABS is Boulware and vice versa.**

Explanation: For each of the metrics, we measure the relative value of the best performing model w.r.t the second-best performing model for each domain. **For example, in the Fire Random domain for the ONAC algorithm, BLRA is 9.28% better than Counter for** 4 **agents negotiation and and BLRA is 6% better than LSTM for** 5 **agents negotiation on the outcome utility metric.**

Metrics	Domains	Methods		
		Counter	Bayesian	LSTM
Boulware Bidding Outcome Utility	Meeting	2	1(1.06 %)	
	Fire Random	2	1(6.06 %)	
Optimal Outcome Utility	Meeting	2	1(1.75%)	
	Fire Random	2	1(9.28%)	

(a) 3 Agents Negotiation

Metrics	Domains	Methods			
		Counter	Bayesian	A B S	LSTM
Boulware Bidding Outcome Utility	Meeting	2	1(1.06 %)		
	Fire Random		2	1(.9%)	
Optimal Outcome Utility	Meeting	2	1(1.1%)		
	Fire Random		1(6 %)		2

(b) 5 Agents Negotiation

Fig. 10 Relative Performance Table (Ranks

References

[Baarslag et al.(2016)Baarslag, Hadfi, Hindriks, Ito, and Jonker] Baarslag T, Hadfi R, Hindriks K, Ito T, Jonker C (2016) Optimal non-adaptive concession strategies with incomplete information. In: Recent Advances in Agent-based Complex Automated Negotiation, Springer, pp 39–54

[Chalupsky et al.(2002)Chalupsky, Gil, Knoblock, Lerman, Oh, Pynadath, Russ, and Tambe] Chalupsky H, Gil Y, Knoblock CA, Lerman K, Oh J, Pynadath DV, Russ TA, Tambe M (2002) Electric elves: Agent technology for supporting human organizations. AI magazine 23(2):11

[Choi et al.(2001)Choi, Liu, and Chan] Choi SP, Liu J, Chan SP (2001) A genetic agent-based negotiation system. Computer Networks 37(2):195–204

[Coehoorn and Jennings(2004)] Coehoorn RM, Jennings NR (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, pp 59–68

[Crump and Moon(2017)] Crump L, Moon D (2017) Precedents in negotiated decisions: Korea–australia free trade agreement negotiations. Negotiation Journal 33(2):101–127

[Faratin et al.(1998)Faratin, Sierra, and Jennings] Faratin P, Sierra C, Jennings NR (1998) Negotiation decision functions for autonomous agents. Robotics and Autonomous Systems 24(3-4):159–182

[Fatima et al.(2004a)Fatima, Wooldridge, and Jennings] Fatima S, Wooldridge M, Jennings NR (2004a) Optimal negotiation of multiple issues in incomplete information settings. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3, IEEE Computer Society, pp 1080–1087

[Fatima et al.(2014)Fatima, Kraus, and Wooldridge] Fatima S, Kraus S, Wooldridge M (2014) Principles of automated negotiation. Cambridge University Press

[Fatima et al.(2004b)Fatima, Wooldridge, and Jennings] Fatima SS, Wooldridge M, Jennings NR (2004b) An agenda-based framework for multi-issue negotiation. Artificial Intelligence 152(1):1–45

[Gear et al.(2020)Gear, Prakash, Singh, and Paruchuri] Gear AS, Prakash K, Singh N, Paruchuri P (2020) Predictrv: A prediction based strategy for negotiations with dynamically changing reservation value. In: International Conference on Group Decision and Negotiation, Springer, pp 135–148

[Heyman and Sobel(1982)] Heyman DP, Sobel MJ (1982) Stochastic models in operations research: stochastic optimization, vol 2. Courier Corporation

[Hindriks and Tykhonov(2008)] Hindriks K, Tykhonov D (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-Volume 1, International Foundation for Autonomous Agents and Multiagent Systems, pp 331–338

[Hochreiter and Schmidhuber(1997)] Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation 9(8):1735–1780

[Jennings et al.(2001)Jennings, Faratin, Lomuscio, Parsons, Wooldridge, and Sierra] Jennings NR, Faratin P, Lomuscio AR, Parsons S, Wooldridge MJ, Sierra C (2001) Automated negotiation: prospects, methods and challenges. Group Decision and Negotiation 10(2):199–215

[Kraus(2001)] Kraus S (2001) Strategic negotiation in multiagent environments. MIT press

[Li et al.(2013)Li, Vo, Kowalczyk, Ossowski, and Kersten] Li M, Vo QB, Kowalczyk R, Ossowski S, Kersten G (2013) Automated negotiation in open and distributed environments. Expert Systems with Applications 40(15):6195–6212

[Lin et al.(2014)Lin, Kraus, Baarslag, Tykhonov, Hindriks, and Jonker] Lin R, Kraus S, Baarslag T, Tykhonov D, Hindriks K, Jonker CM (2014) Genius: An integrated environment for supporting the design of generic automated negotiators. Computational Intelligence 30(1):48–70

[Raiffa(1982)] Raiffa H (1982) The art and science of negotiation. Harvard University Press

[Rosenfeld and Kraus(2018)] Rosenfeld A, Kraus S (2018) Predicting human decision-making: From prediction to action. Synthesis Lectures on Artificial Intelligence and Machine Learning 12(1):1–150

[Rosenschein and Zlotkin(1994)] Rosenschein JS, Zlotkin G (1994) Rules of encounter: designing conventions for automated negotiation among computers. MIT press

[Rubinstein(1982)] Rubinstein A (1982) Perfect equilibrium in a bargaining model. Econometrica: Journal of the Econometric Society pp 97–109

[Sandholm et al.(1995)Sandholm, Lesser et al.] Sandholm T, Lesser VR, et al. (1995) Issues in automated negotiation and electronic commerce: Extending the contract net framework. In: ICMAS, vol 95, pp 12–14

[Scerri et al.(2002)Scerri, Pynadath, and Tambe] Scerri P, Pynadath DV, Tambe M (2002) Towards adjustable autonomy for the real world. Journal of Artificial Intelligence Research 17(1):171–228

[Scheaffer and Young(2009)] Scheaffer RL, Young L (2009) Introduction to Probability and its Applications. Cengage Learning

[Yu et al.(2013)Yu, Ren, and Zhang] Yu C, Ren F, Zhang M (2013) An adaptive bilateral negotiation model based on bayesian learning.

[Zeng and Sycara(1998)] Zeng D, Sycara K (1998) Bayesian learning in negotiation. International Journal of Human-Computer Studies 48(1):125–141