Bayesian Learning based Negotiation Agents for Supporting Negotiation with Incomplete Information

Jeonghwan Gwak and Kwang Mong Sim

***Abstract*—An optimal negotiation agent should have capability for maximizing its utility even for negotiation with incomplete information in which the agents do not know opponent's private information such as reserve price (*RP*) and deadline. To support negotiation with incomplete negotiation, this work focuses on designing learning agents called Bayesian learning (BL) based negotiation agents (BLNA) adopting a time-dependent negotiation model. In BLNA, BL is used for estimating opponent's *RP* and the corresponding deadline is computed using the estimated *RP*. BLNA also has the capability of maximizing its utility using estimated deadline information when its deadline is shorter than or equal to the estimated deadline. To evaluate the performance of BLNA, BLNA is compared to an agent with complete information and an agent with incomplete information. Empirical results showed that BLNA achieved: 1) always higher utility than the agent with incomplete information for all cases, 2) close to or slightly lower utility than the agent with complete information when its deadline is lower and equal to opponent's deadline, and 3) lower utility than the agent with complete information when deadline is higher than opponent's deadline.**

***Index Terms*—Automated negotiation, negotiation agents, intelligent agents, Bayesian learning, negotiation with incomplete information**

I. INTRODUCTION

AUTOMATED negotiation is defined as a process for resolving differences and conflicting goals among interacting agents. There are two types of negotiation environment, one for agents with complete negotiation settings and the other for agents with incomplete negotiation settings. While agents with complete information know opponent private information such as reserve price and deadline, agents with incomplete information do not know opponent private information. For negotiation with complete information, optimal solutions, for an agent can be easily determined using the interacting agent's private information, e.g., [1] and [2]. However, it is not easy for an agent with incomplete information to achieve optimal solutions. For optimal solutions in negotiation with incomplete information, a learning method is required. In this work, Bayesian learning (BL) is adopted for supporting negotiation with incomplete information by finding an opponent's private information.

There are some related works [1], [2], [3] using BL for supporting negotiation with incomplete information. In [3], BL framework was first introduced to support negotiation. In [1] and [2], BLGAN uses the synergy of BL and a genetic algorithm (GA).

A. *Contribution of this work*

The differences between the above related works and this work are as follows:

1) Compared to [1] and [2], this work suggests a new BL method for estimating an opponent's reserve price (*RP*) by designing a new conditional probability of BL.
2) Since estimation of the exact *RP* using BL is not possible, there exist some estimation errors of *RP*. To compensate for the estimation errors, a GA was used in [2] for making tradeoff between an agent's proposal and its opponent proposal. However, negotiation using the GA can be finished quickly without reaching optimal results. Even though the agent adopting a GA-based tradeoff algorithm can increase the negotiation success rates, it should give up some amount of utility. As a primarily research report, this paper is focused on enhancing BL part only but not focusing using a GA-based tradeoff algorithm.
3) Although this paper is based on [1] and [2], there exist several different points. Compared to [1] and [2], this work uses different equations in generating proposals for both learning and no-learning agents. While estimation of opponent *RP* and deadline in [1] and [2] is carried out separately, we calculate opponent deadline using estimated opponent *RP* because estimation of opponent *RP* and deadline is inter-dependent (see Section II-C). Furthermore, the calculated opponent deadline information is used for generating proposals to increase its utility when the calculated opponent deadline is longer than or equal to its deadline.

The paper is organized as follows. The negotiation model in this work is described in Section II, and the design of proposed BL-based negotiation agents (BLNA) are described in Section III. Section IV shows experimental results and analyzes the performance. The final section concludes this work with summary and suggests future works.

Manuscript received January 12, 2011. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-220-D00092).

Jeonghwan Gwak is with the School of Information and Communications (SIC), Gwangju Institute of Science and Technology, South Korea (phone: 82-62-715-2280; fax: 82-62-715-2204; e-mail: gwak@gist.ac.kr).

Kwang Mong Sim is with the School of Information and Communications (SIC), Gwangju Institute of Science and Technology, South Korea (phone: 82-62-715-2235; fax: 82-62-715-2204; e-mail: prof_sim_2002@yahoo.com)

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Negotiation Model

A. Time-dependent Negotiation Model

In this work, we consider bilateral negotiation between two self-interested agents over an issue, e.g., price, quality of service, etc. The two agents have conflicting roles such as seller (*S*) and buyer (*B*). The agents negotiate by exchanging proposals using Rubinstein's alternating-offers protocol [4]. The fixed (no learning) agent $x \in \{B, S\}$ generates proposals at a time round $t, 0 \le t \le \tau_x$, as follows:

$$P_t^x = IP_x + (-1)^\alpha \left(\frac{t}{\tau_x}\right)^{\lambda_x} \left|RP_x - IP_x\right|, \tag{1}$$

where $\alpha = 1$ for *S* and $\alpha = 0$ for *B*. IP_x and RP_x is the initial price (the most favorable price that can afford) and reserve price (the least favorable price that can afford) of *x*, respectively. τ_x is the deadline and λ_x, $0 \le \lambda_x \le \infty$, is the time-dependent strategy of *x*. The concession behavior of *x* is determined by the values the time-dependent strategy [5] and is classified as follows:

1) Conciliatory $(0 \le \lambda_x < 1)$: *x* makes larger concession in earlier negotiation rounds and smaller concessions in later rounds.
2) Linear $(\lambda_x = 1)$: *x* makes a constant rate of concession.
3) Conservative $(1 < \lambda_x \le \infty)$: *x* makes smaller concession in earlier negotiation rounds and larger concessions in later rounds.

Let *D* be the event in which *x* fails to reach an agreement. The utility function of *x* is defined as $U_x : [IP_x, RP_x] \cup D \to [0,1]$ such that $U_x(D) = 0$ and for any $P_t^x \in [IP_x, RP_x]$, $U_x(P_t^x) > U_x(D)$. $U_x(P_t^x)$ is given as follows:

$$U_x(P_t^x) = u_{\min} + (1 - u_{\min}) \frac{\left|RP_x - P_t^x\right|}{\left|RP_x - IP_x\right|}, \tag{2}$$

where $u_{\min}$ is the minimum utility that *x* receives for reaching an agreement at RP_x. For experimental purpose, the value of $u_{\min}$ is set as 0.1. At $P_t^x = RP_x$, $U_x(RP_x) = 0.1 > U_x(D) = 0$.

For the bilateral negotiation in this work, it is assumed that *S* starts the negotiation by making proposals to *B*. The negotiation process between the two agents will be terminated: 1) in making an agreement when an offer or a counter-offer is accepted, or 2) in a conflict when one of the two agents' deadlines is reached. An agreement is reached when one agent proposes a deal that matches or exceeds what another agent asks for, i.e., $U_S(P_t^B) \ge U_S(P_{t-1}^S)$ or $U_B(P_{t-1}^B) \le U_B(P_t^S)$.

B. Optimal Negotiation Strategy with Complete Information

The optimal strategy of *x* is defined as the strategy that maximizes the utility of *x* at agreement time T_c. Let P_c be the agreement price (i.e., $P_c = P_{T_c}^x$). The maximum strategy ensures $U_x(P_c) \ge U_x(P_t^x)$ for all $t \le T_c$.

For negotiation with complete information between *S* and *B*, [1] and [2] proved the following Theorems 1 and 2 (refer [1] and [2] for detailed illustrations):

Theorem 1 $(\tau_B < \tau_S)$: *S* achieves maximal utility when it adopts the strategy

$$\lambda_S = \log_{\frac{\tau_B}{\tau_S}} \frac{IP_S - RP_B}{IP_S - RP_S}.$$

Theorem 2 $(\tau_B > \tau_S)$: *B* achieves maximal utility when it adopts the strategy

$$\lambda_B = \log_{\frac{\tau_S}{\tau_B}} \frac{RP_S - IP_B}{RP_B - IP_B}.$$

C. Negotiation with Incomplete Information

For a negotiation with incomplete information, if an agent can estimate opponent's *RP* and deadline exactly, the agent can find an optimal strategy using its corresponding theorem between Theorems 1 and 2.

The relationship between *RP* and deadline is given as follows. The formulas for calculating *RP* in (3) and deadline in (4) are derived from (1).

$$RP_x = \frac{P_t^x - IP_x}{\left(\frac{t}{\tau_x}\right)^{\lambda_x}} + IP_x \tag{3}$$

$$\tau_x = t \cdot \left(\frac{P_t^x - IP_x}{RP_x - IP_x}\right)^{\lambda_x} \tag{4}$$

For the given IP_x, t, P_t^x and λ^x, the calculation of RP_x is related with τ_x and the calculation of τ_x is related with RP_x. Hence, calculation of RP_x and τ_x is not separable but closely related with each other. If we can estimate either *RP* or deadline, the other can be calculated from the estimated variable using (3) or (4).

If it is assumed that IP_x is known (it can be easily assumed if the first proposal is IP_x), λ^x can be calculated using IP_x and two proposals with different time rounds as follows:

$$P_t^x - IP_x = \left(\frac{t}{\tau_x}\right)^{\lambda_x} (RP_x - IP_x) \tag{5}$$

$$P_{t-1}^x - IP_x = \left(\frac{t-1}{\tau_x}\right)^{\lambda_x} (RP_x - IP_x) \tag{6}$$

By dividing (5) by (6), the following equation is achieved.

$$\frac{P_t^x - IP_x}{P_{t-1}^x - IP_x} = \left(\frac{t}{t-1}\right)^{\lambda_x}$$

Finally, the following λ_x is calculated.

$$\lambda_x = \log_{\frac{t}{t-1}} \frac{P_t^x - IP_x}{P_{t-1}^x - IP_x}, \quad \text{where } t \ge 3, \tag{7}$$

In summary, if IP_x is known, exact λ_x can be calculated by (7) when $t \ge 3$. Then, if either *RP* or deadline is estimated, the other can be computed using (3) or (4). In this work, we estimated *RP* and the corresponding deadline calculation is conducted using (4).

III. Bayesian Learning based Proposed Approach

In this section, we will describe the design of BLNA to support negotiation with incomplete information.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

A. Bayesian Learning of opponent's RP

Let the price range be in [MIN_P, MAX_P]. An agent adopting BL forms a set of hypotheses $\{H_i\}$ of opponent's *RP*, where $H_i = i \times (\mathrm{MAX}_P - \mathrm{MIN}_P) / N_H$ and N_H is the number of hypotheses. The *i*-th hypothesis of an opponent's *RP* is defined as RP_i^{opp} and the estimated opponent *RP* is defined as RP_i^{opp}.

The following relation (domain knowledge) between P_t^x and RP_x is derived from (3).

$$\frac{P_t^x - IP_x}{R_d^x(t)} = RP_x - IP_x, \tag{8}$$

where $R_d^x(t)$ is the discount ratio of *x* and measured by $R_d^x(t) = \left(\frac{t}{\tau_x}\right)^{\lambda_x}$.

For example, given $IP_B = 5$, $RP_B = 85$, $\tau_B = 50$ and $\lambda_B = 5$, the following Figs. 1 and 2 show simulation results of *B*'s proposals and corresponding discounting ratio at $t = 0, 1, \ldots, \tau_B$, respectively. At $t = 25$, the results shows $P_{25}^B = 7.5$ and $R_d^B(25) = \left(\frac{25}{50}\right)^5 = 0.03125$. Hence, $\frac{P_t^x - IP_x}{R_d^x(t)} = \frac{7.5 - 5}{0.03125} = 80$ and it equals to $RP_x - IP_x = 85 - 5 = 80$.

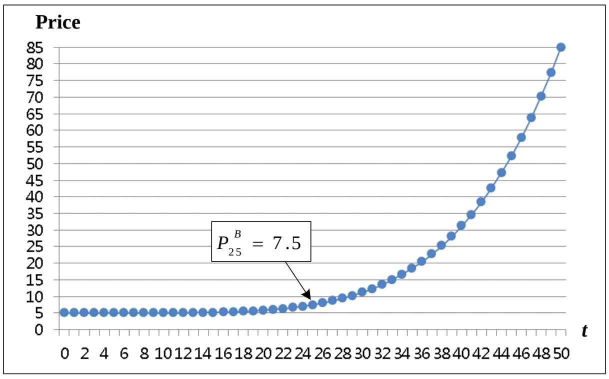

Fig. 1. *B*'s proposals at $t = 0, 1, \ldots, \tau_B$

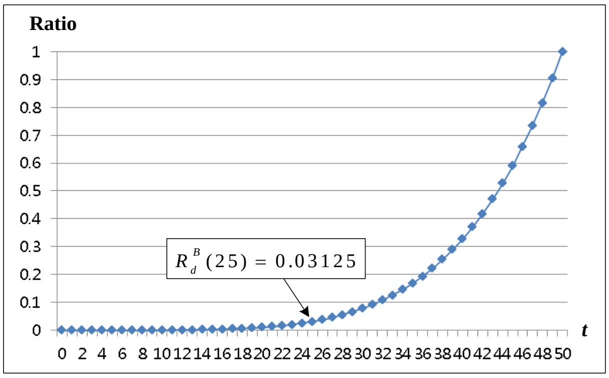

Fig. 2. Discounting ratio $R_d^B(t)$ at $t = 0, 1, \ldots, \tau_B$

Using the formula in (8) representing relation between P_t^x and RP_x, the conditional probability (likelihood) distribution is designed as follows:

$$P(P_t^{opp} \mid RP_i^{opp}) = 1 - \frac{\left| \frac{P_t^{opp} - IP^{opp}}{R_d^x(t)} - (RP_i^{opp} - IP^{opp}) \right|}{\frac{P_t^{opp} - IP^{opp}}{R_d^x(t)}}, \tag{9}$$

where $R_d^x(t)$ is the estimated discounting ratio at time round *t*. To achieve accurate conditional probability in (9), it is crucial to obtain an appropriate $R_d^x(t)$. $R_d^x(t)$ is obtained by the following formula:

$$R_d^x(t) = \left(\frac{t}{\tau_{opp}(t-1)}\right)^{\lambda_{opp}}, \tag{10}$$

where λ_{opp} is calculated by (7) when $t \geq 3$. Due to the same dimensionality and difficulty of estimating opponent deadline as opponent *RP*, $\tau_{opp}(t-1)$ is the estimated opponent deadline at time round $t-1$, and is calculated using (4) from the estimated opponent *RP* at previous time round $t-1$, $RP^{opp}(t-1)$.

In some cases, $P(P_t^{opp} \mid RP_i^{opp})$ should be zero as follows. For *S* estimating opponent *RP* using BL, if $P_t^{opp} > RP_i^{opp}$, then $P(P_t^{opp} \mid RP_i^{opp}) = 0$ because *B* will not generate proposals higher than its *RP*. Similarly, for *B* estimating opponent *RP* using BL, if $P_t^{opp} < RP_i^{opp}$, then $P(P_t^{opp} \mid RP_i^{opp}) = 0$ because *S* will not generate proposals lower than its *RP*.

When *x* receives P_t^{opp}, the Bayesian updating formula revises its belief about opponent *RP* with prior probability distribution is defined as follows:

$$P(RP_i^{opp} \mid P_t^{opp}) = \frac{P_{t-1}(RP_i^{opp}) P(P_t^{opp} \mid RP_i^{opp})}{\sum_{i=1}^{N_H} P_{t-1}(RP_i^{opp}) P(P_t^{opp} \mid RP_i^{opp})}, \tag{11}$$

where the prior probability $P_{t-1}(RP_i^{opp})$ is defined as $P_{t-1}(RP_i^{opp}) = P(RP_i^{opp} \mid P_{t-1}^{opp})$ and initially, it is assumed uniform distribution over all hypotheses.

Finally, the expected value of RP^{opp} at time round *t*, $RP^{opp}(t)$, is computed using the following formula.

$$RP^{opp}(t) = \sum_t P(RP_i^{opp} \mid P_t^{opp}) RP_i^{opp} \tag{12}$$

B. BL-based Negotiation Agents

BLNA *x* generates the next proposal using the following formula [2]:

$$P_t^x = P_{t-1}^x + (-1)^{\alpha} \left(\frac{1}{\tau_x - (t-1)}\right)^{\lambda_x} \left| RP_x - P_{t-1}^x \right|, \tag{13}$$

where $\alpha = 1$ for *S* and $\alpha = 0$ for *B*. Compared to (1), the main difference is that (13) treats the previous proposal P_{t-1}^x as its new initial price at time round *t*.

Until now, all materials for generating (optimal) strategy are studied and prepared. The procedure for generating optimal strategy is described in Algorithm 1.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Set x as BLNA

Part 1 (BL stage): Generating Proposals using BL information

If time round $t < 3$

Compute $RP^{opp}(t)$ as an average value in feasible price range.

Compute $\hat{\tau}_{opp}$ using (4) and $RP^{opp}(t)$.

Compute λ_x using Theorem 1 or 2 with $RP^{opp}(t)$ and $\hat{\tau}_{opp}$.

Generate a proposal P_t^x using (13) with λ_x.

If time round $t = 2$

Set $P_{t-1}(RP_i^{opp})$ as $RP^{opp}(t = 2)$ for all i.

If time round $t \geq 3$

Compute $\hat{\tau}_{opp}(t-1)$ using $RP^{opp}(t-1)$.

Compute $\hat{\lambda}_{opp}$ using (7).

Compute $R_d^x(t)$ using (10).

Compute $P(P_t^{opp} | RP_i^{opp})$ using (9) for all i.

Compute $P(RP_i^{opp} | P_t^{opp})$ using (11) for all i.

Set $P_{t-1}(RP_i^{opp}) = P(RP_i^{opp} | P_{t-1}^{opp})$.

Compute $RP^{opp}(t)$ using (12).

Compute $\hat{\tau}_{opp}$ using (4) and $RP^{opp}(t)$.

Compute λ_x using Theorem 1 or 2 with $RP^{opp}(t)$ and $\hat{\tau}_{opp}$.

Generate a proposal P_t^x using (13) with λ_x.

Part 2: Generating Proposals using Deadline Information

(Case 1: the case when x started the negotiation first)

If $t = \tau_x$ and $\tau_x \leq \hat{\tau}_{opp}$

If x started the negotiation first

If $\left(U(RP_x) \text{ and } U(P_t^x)\right) \leq U(P_{t-1}^{opp}) \leq U(P_{t-1}^x)$

Generate a proposal P_t^x as P_{t-1}^{opp}.

Else,

Generate a proposal P_t^x as its RP_x.

(Case 2: the case when opponent started the negotiation first)

If $t = \tau_x$ and $\tau_x \leq \hat{\tau}_{opp}$

If opponent started the negotiation first

Accept a proposal $P_{\tau_x}^{opp}$.

Algorithm 1. BL-based Negotiation Agents

Algorithm 1 consists of two parts. Part 1 for generating proposals using BL information is the main part of the algorithm as discussed in Section III-A. Part 2 is the procedure for generating proposals using deadline information. If BLNA can learn its deadline is lower than or equal to opponent deadline exactly ($\tau_x \leq \hat{\tau}_{opp}$), it can still maximize its utility even though it reaches its deadline $t = \tau_x$. Depending on which agent started the negotiation first, Part 2 is divided into the following two cases:

(Case 1) If BLNA x started the negotiation first, there is still room for x to achieve its maximum utility at time round $t = \tau_x$. This can be carried out by making the proposal $P_{\tau_x}^x$ as opponent proposal $P_{\tau_x}^{opp}$ at $t = \tau_x$ on the condition that each $U_x(RP_x)$ and $U_x(P_t^x) \leq U_x(P_{t-1}^{opp}) \leq U_x(P_{t-1}^x)$. However, because estimating the opponent proposal will be difficult and there will exist errors in estimating the proposal using BL, BLNA x will generate the proposal P_t^x as the opponent proposal $P_{\tau_x}^{opp}$ at $t = (\tau_x - 1)$.

(Case 2) If opponent started the negotiation first, x will achieve its maximum utility at time round $t = \tau_x$ by not making a proposal $U_x(P_{\tau_x}^{opp}) \leq U_x(P_{t-1}^x)$ and accepting the opponent proposal $P_{\tau_x}^{opp}$ at time round $t = \tau_x$.

IV. Experimental Results

To evaluate the effectiveness and performances of BLNA empirically, three types of negotiation scenarios between S and B were studied as in Table I. Throughout the three negotiation scenarios, B is set as a fixed and no-learning agent in which it generates proposals using (1) and a strategy randomly fixed at start of negotiation. S has three types according to the following scenarios.

TABLE I
THREE NEGOTIATION SCENARIOS

Scenario	Agent *S*	Agent *B*
Complete	Complete information	Incomplete information
Incomplete	Incomplete information	Incomplete information
Incomplete with BL	Incomplete information *S* learns opponent's *RP* using BL	Incomplete information

Scenario 1 (*S* as the optimal agent): *S* has complete information about *B* and *B* has incomplete information about *S*. Since *S* knows *B*'s *RP* and deadline, *S* generates proposals by adopting its optimal strategy using Theorem 1. The negotiation result corresponds to the best-case scenario that BLNA is targeting for.

Scenario 2 (*S* as the fixed and no-learning agent): Both *S* and *B* have incomplete information about each other. *S* does not know *B*'s *RP* and deadline. *S* and *B* generate proposals using a strategy randomly fixed within the possible strategy range at the start of negotiation. The negotiation result corresponds to the worst-case scenario.

Scenario 3 (*S* as BLNA): Both *S* and *B* have incomplete information about each other. However, *S* generates proposals by adopting estimated strategies using Theorem 1 with estimated *B*'s *RP* using BL and the corresponding calculation of *B*' deadline. Furthermore, *S* uses some deadline information for generating proposals (see Part 2 in Algorithm 1)

A. Experimental Settings

The agents' parameter settings are summarized in Table II. Initially, *S* and *B*'s *IP*s, *RP*s and deadlines were randomly selected in the given ranges at start of negotiation. Three type of deadlines 'Short', 'Mid', and 'Long' were used. (Short, Mid), (Mid, Mid) and (Long, Mid) were used for comparing negotiation results with respect to deadline effects (i.e., with different bargaining advantage). The representation (Deadline of *S*, Deadline of *B*) means *S* sets the first element

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and *B* sets the second element as their deadlines. For example, (Short, Mid) means *S* adopts the 'Short' deadline and *B* adopts 'Mid' deadline.

In the experiments of (Short, Mid), (Mid, Mid) and (Long, Mid), 1000 random runs for each scenario were carried out, and in each run, agents used the same *IP*s, *RP*s, deadlines and initial strategies for all the scenarios. In the random generation of *B* and *S*'s strategies at the start of negotiation, the probabilities of generating conciliatory (conceding rapidly) and conservative (conceding slowly) strategies are same.

TABLE II
AGENTS' PARAMETER SETTINGS

Parameter type		Possible Parameter values
Minimum possible price (MIN_P)		1
Maximum possible price (MAX_P)		100
RP_S		[MIN_P + 5, MIN_P+ (MAX_P – MIN_P)/2]
RP_B		[RP_S + 10, MAX_P–5]
IP_S		[RP_B, MAX_P]
IP_B		[MIN_P, RP_S]
Possible strategy range		[0.002, 10]
Deadline	Short	25
	Mid	50
	Long	100
N_H		100

B. *Experimental Results*

The following four performance measures were used: 1) Success rate, 2) Failure type, 3) Average negotiation round, and 4) *S*'s normalized average utility. Success rate (SR) is defined as SR = $N_{success}/N_{total}$, where $N_{success}$ is the number of successful deals and N_{total} is the total number of deals. To identify the reasons of negotiation failure, two types of negotiation failure were considered: 'Type I' and 'Type II' to represent infeasible deals due to 1) *B*'s infeasible high strategy settings and 2) *S*'s wrong strategy estimation, respectively. Average negotiation round (ANR) is measured by the average number of negotiation rounds required to reach an agreement for all successful deals. *S*'s normalized average utility (NAU_S) is defined as

$$NAU_S = \frac{\sum_0^{N_{success}} U_S\left(P^x_{t_{final}}\right)}{N_{success}} \frac{1}{U_S^{\max}},$$

where $P(t_{final})$ is the final proposal at the final negotiation time round t_{final} and $U_S^{\max}$ is the maximum utility of *S*.

TABLE III
RESULTS OF [SHORT, MID] CASE

(Short, Mid)	*Complete*	*Incomplete*	*Incomplete with BL*
Success rate	0.659	0.659	0.659
Failure type [a]	-	Type I	Type I
Average negotiation rounds	25	19.68	24.84
S's normalized average utility	0.5746	0.4661	0.5715

[a] As *B* has a higher strategy value (e.g., 10), there is a higher possibility that the negotiation can fail to generate appropriate proposals before the shorter (*S*'s) deadline reaches (i.e., *B*'s final proposal is lower than RP_S at *S*'s deadline). In this case, *S* cannot make the successful negotiation and 'Type I' counts the kind of negotiation failures.

TABLE IV
RESULTS OF [MID, MID] CASE

(Mid, Mid)	*Complete*	*Incomplete*	*Incomplete with BL*
Success rate	1	1	1
Failure type	-	-	-
Average negotiation rounds	50	36.12	48.13
S's normalized average utility	0.9473	0.5649	0.9032

TABLE V
RESULTS OF [LONG, MID] CASE

(Long, Mid)	*Complete*	*Incomplete*	*Incomplete with BL*
Success rate	1	0.536	0.651
Failure type [b]	-	Type II	Type II
Average negotiation rounds	50	37.83	42.33
S's normalized average utility	0.9997	0.5998	0.7271

[b] If *S* learns inexact *RP* and deadline having higher error rates, the negotiation can fail to generate appropriate proposals before the shorter (*B*'s) deadline reaches (i.e., *S*'s final proposal is higher than RP_B at *B*'s deadline). In this case, *S* cannot make the successful negotiation and 'Type II' counts the kind of negotiation failures.

C. *Analysis and Discussion*

The goal of BLNA is to achieve the results that are close to the "Complete" scenario that shows the optimum results for the given settings. Since in (Short, Mid) and (Mid, Mid) cases, *S* has shorter than or equal to the deadline of *B*, Part 2 in BLNA algorithm will have significant role to improve the performance. Specifically, (Case 2) in Part 2 will have effect in this case because *S* will start the negotiation first. In contrast, in (Long, Mid) case, Part 1 will have significant role for achieving good performance.

Observation 1 - (Short, Mid) Case: "Incomplete with BL" scenario achieved very close results to the "Complete" scenario.

Analysis: Since *S* has a shorter deadline than *B*, if *S* appropriately learns *B*'s *RP* (RP_B) and computes *B*'s deadline $\hat{\tau}_B$ ($> \tau_S$) in "Incomplete with BL" scenario, *S* will propose *B*'s previous proposal at the deadline and successful agreement will be made at the shorter deadline (see Part 2-(Case 1) in Algorithm 1).

As shown in Table III, in "Complete" scenario, the negotiation was terminated at time rounds 25 and *S* achieved normalized average utility 0.5746. Even though in

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

"Incomplete" scenario, *S* achieved normalized average utility 0.4661 at average time rounds 19.68, in "Incomplete with BL" scenario *S* achieved normalized average utility 0.5715 at average time rounds 24.84. The values of "Incomplete with BL" scenario are very close to "Complete" scenario. The results showed that *S* (the BLNA) can learn its opponent's (*B*'s) *RP* and deadline when *B* has longer deadline than *S* $(\tau_S < \hat{\tau}_B)$ almost exactly in (Short, Mid) case.

Observation 2 - (Mid, Mid) Case: "Incomplete with BL" scenario achieved results that are close to the "Complete" scenario.

Analysis: Since *S* and *B* have the same deadline, if *S* learns *B*'s *RP* (RP_B) and computes *B*'s deadline $\hat{\tau}_B$ $(=\tau_S)$ appropriately in "Incomplete with BL" scenario, *S* will propose *B*'s previous proposal at the deadline and successful agreement will be made at the deadline (see Part 2-(Case 1) in Algorithm 1).

As shown in Table IV, in "Complete" scenario, the negotiation was terminated at time rounds 50 and *S* achieved normalized average utility 0.9473. Even though in "Incomplete" scenario, *S* achieved normalized average utility 0.5649 at average time rounds 36.12, in "Incomplete with BL" scenario *S* achieved normalized average utility 0.9032 at average time rounds 48.13. The values of "Incomplete with BL" scenario are close to "Complete" scenario. The results showed that *S* (the BLNA) can learn the opponent's (*B*'s) *RP* and deadline when *B*'s does not have shorter than *S*'s deadline $(\tau_S \le \hat{\tau}_B)$ appropriately with some errors in (Mid, Mid) case.

Observation 3 - (Long, Mid) Case: "Incomplete with BL" scenario achieved results not close to the "Complete" scenario but achieved results better than "Incomplete" scenario.

Analysis: Since *S* has the longer deadline than *B*, if *S* appropriately learns *B*'s *RP* (RP_B) and computes *B*'s deadline $(\hat{\tau}_B < \tau_S)$ appropriately in "Incomplete with BL" scenario, *S* will make proposals using the strategy by Theorem 1 with RP_B and $\hat{\tau}_B$.

As shown in Table V, in "Complete" scenario, the negotiation was terminated at time rounds 50 and *S* achieved normalized average utility 0.9997. Although *S* has longer deadline than *B*, *S* cannot achieved the maximum utility 1. This is because if *S* starts negotiation first, *B* will decide whether it accept *S*'s proposal or not. In "Incomplete" scenario, *S* achieved normalized average utility 0.5998 at average time rounds 37.83. In "Incomplete with BL" scenario *S* achieved normalized average utility 0.7271 at average time rounds 42.33. The values of "Incomplete with BL" scenario are not close to "Complete" scenario but they are higher than "Incomplete" scenario. The results show that *S* (the BLNA) can learn the opponent's (*B*'s) *RP* and deadline with some errors in (Long, Mid) case.

V. Conclusion and Future Works

In this paper, we showed the performance of proposed BLNA by considering the case that one agent uses BL and deadline information for generating its next proposal in the three deadline combinations, (Short, Mid), (Mid, Mid) and (Long, Mid). The performance showed that:

1) In (Short, Mid) case, BLNA can learn the opponent deadline is longer than its opponent almost exactly.
2) In (Mid, Mid) case, BLNA can learn the opponent deadline is not shorter than its opponent with some errors.
3) In (Long, Mid) case, BLNA can learn the opponent's *RP* and deadline with more higher errors.

From the results 1) to 3), we conclude BLNA can support the negotiation with incomplete information. However, we need to improve the performance, especially in the case of (Long, Mid) case, in terms of both the success rate and utility.

Due to the space limitation, we only considered one fixed case of deadline (Mid case) for *B*. There are still plenty of deadline combinations such as (Short, Short), (Mid, Short), (Long, Short), (Short, Long), (Mid, Long) and (Long, Long) to figure out the whole system performance of proposed BLNA. Furthermore, although we only considered the case that one agent (*S*) learns opponent's information using BL and deadline information, it will be interesting to analyze the case that both agents can learn each other. In our future work, we will consider the above issues. Moreover, the research for supporting negotiation with incomplete information is still being carried out to increase the performance of BLNA by incorporating evolutionary algorithms in BL stage of BLNA.

References

[1] K. M. Sim, Y. Guo, and B. Shi, "Adaptive bargaining agents that negotiate optimally and rapidly." pp. 1007-1014.

[2] K. M. Sim, Y. Guo, and B. Shi, "*BLGAN*: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 1, pp. 198-211, Feb. 2009.

[3] D. Zeng, and K. Sycara, "Bayesian learning in negotiation," *Int. J. Hum.-Comput. Stud.*, vol. 48, no. 1, pp. 125-141, 1998.

[4] A. Rubinstein, "A bargaining model with incomplete information about time preferences," *Econometrica*, vol. 53, no. 5, pp. 1151-1172, Sep. 1985.

[5] K. M. Sim, "Equilibria, prudent compromises,and the "waiting" game," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 35, no. 4, pp. 712-724, Aug. 2005.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)