This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

BLGAN: Bayesian Learning and Genetic Algorithm for Supporting Negotiation With Incomplete Information

Kwang Mong Sim, *Senior Member, IEEE*, Yuanyuan Guo, and Benyun Shi

***Abstract*—Automated negotiation provides a means for resolving differences among interacting agents. For negotiation with complete information, this paper provides mathematical proofs to show that an agent's optimal strategy can be computed using its opponent's reserve price (*RP*) and deadline. The impetus of this work is using the synergy of Bayesian learning (*BL*) and genetic algorithm (*GA*) to determine an agent's optimal strategy in negotiation (*N*) with incomplete information. *BLGAN* adopts: 1) *BL* and a deadline-estimation process for estimating an opponent's *RP* and deadline and 2) *GA* for generating a proposal at each negotiation round. Learning the *RP* and deadline of an opponent enables the *GA* in *BLGAN* to reduce the size of its search space (*SP*) by adaptively focusing its search on a specific region in the space of *all* possible proposals. *SP* is dynamically defined as a region around an agent's proposal P at each negotiation round. P is generated using the agent's optimal strategy determined using its estimations of its opponent's *RP* and deadline. Hence, the *GA* in *BLGAN* is more likely to generate proposals that are closer to the proposal generated by the optimal strategy. Using *GA* to search around a proposal generated by its current strategy, an agent in *BLGAN* compensates for possible errors in estimating its opponent's *RP* and deadline. Empirical results show that agents adopting *BLGAN* reached agreements successfully, and achieved: 1) higher utilities and better combined negotiation outcomes (*CNO*s) than agents that only adopt *GA* to generate their proposals, 2) higher utilities than agents that adopt *BL* to learn only *RP*, and 3) higher utilities and better *CNO*s than agents that do not learn their opponents' *RP*s and deadlines.**

***Index Terms*—Automated negotiation, Bayesian learning (*BL*), genetic algorithms (GAs), intelligent agents, negotiation agents.**

I. Introduction

IN SYSTEMS involving the interactions of artificial or human agents, automated negotiation [1] provides a means for agents to resolve differences and conflicting goals. Although designing negotiation agents that only optimize utility (e.g., buyer agents negotiating for the lowest possible price) may be sufficient for generic e-commerce applications [2], [3], in some applications (e.g., Grid resource management [4]–[9]), negotiation agents should be designed such that they are more likely to acquire resources more rapidly and perhaps with more certainty (in addition to optimizing utility). For instance, in a Grid computing environment, failure to obtain the necessary computing resources before a deadline will lead to a delay in job executions.

Whereas [10], [11] devised negotiation models for bilateral negotiation and proved that the strategies optimize the utilities of agents, this work devises negotiation strategies that attempt to optimize the combined negotiation outcomes (*CNO*s) of agents in terms of their average utilities, success rates, and negotiation speed (measured in negotiation rounds) in negotiation with incomplete information.

While game-theoretic models (e.g., [12]–[17]) have long been used as mathematical tools for modeling and analyzing negotiation processes, this work contributes to the literature in automated negotiation by developing a procedure called *BLGAN* that uses the synergy of Bayesian Learning (*BL*) and Genetic Algorithm (*GA*) for generating *N*egotiation solutions. Game-theoretic models for negotiation are generally categorized into the following: 1) negotiation with complete information in which agents know other agents' parameters (e.g., their reserve prices (*RP*s) and deadlines) and 2) negotiation with incomplete information in which negotiation agents are not endowed with complete information about their opponents (e.g., their *RP*s and deadlines). This work deals with the more difficult problem of finding solutions for negotiation with incomplete information when agents can only deduce the private information of their opponents by studying their moves. The novel feature of this work is that, whereas a Bayesian updating method (Section III-A) and a deadline-estimation process (Section III-B) are adopted for estimating an opponent's *RP* and deadline which guide an agent in evolving its strategy (Section III-C), its *CNO* is enhanced by using *GA* to search for a possibly better proposal (Section III-D).

In some game-theoretic models of negotiation with incomplete information (e.g., [18]), it is assumed that agents only have probabilistic information about the private information of other agents. These models generally adopt the assumption that all agents start with the same probability distribution on the private information about other agents, and this probability distribution is known to all agents [11, p. 22]. For instance, assuming that

Manuscript received February 5, 2008; revised May 31, 2008. This work was supported by the DASAN Project (GIST International Faculty Start-up Fund) from the Gwangju Institute of Science and Technology. This paper was recommended by Associate Editor Prof. A. F. Gomez Skarmeta.

K. M. Sim and B. Shi were with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. They are now with the Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju 500-712, Korea (e-mail: prof_sim_2002@yahoo.com; benyunsh@gmail.com).

Y. Guo is with the Department of Computer Science, University New Brunswick, Saint John, NB E2L 4L5, Canada (e-mail: p61r0@unb.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.2004501

the probability distribution over agents' deadlines is known to all agents, Sandholm and Vulkan [10] addressed the problem of splitting the price–surplus between two negotiating agents. Negotiation agents in [10] adopt the "sit-and-wait" strategy, and it is assumed that both agents know the price–surplus. It was shown in [10] that an agent's optimal strategy is to wait until the first deadline of both agents, at which the agent with the shorter deadline concedes everything (giving the entire price–surplus) to the other agent with longer deadline. Hence, in [10], agents only make two proposals: demanding either the entire surplus or no surplus. By adopting the "sit-and-wait" strategy, the deadline effect almost always suppresses the time-discounting effect (i.e., the devaluation of goods over time). For the negotiation model in this work, the price–surplus can be described as the difference between the reserve prices, RP_B and RP_S, of two agents, a buyer agent B and a seller agent S, such that $RP_B - RP_S$ if $RP_B \geq RP_S$; otherwise, if $RP_B < RP_S$, there is no price–surplus between B and S. If both B and S know RP_B and RP_S, the "sit-and-wait" strategy is also an optimal strategy based on the results in [10]. However, to apply the "sit-and-wait" strategy to agents in this work, one needs to assume that RP_B and RP_S are known to both B and S. For instance, suppose that $RP_S \in [IP_B, RP_B]$, $RP_B \in [RP_S, IP_S]$, and $\tau_B > \tau_S$ (where τ_B and τ_S are the deadlines of B and S, respectively, and IP_B and IP_S are their initial prices). If both B and S adopt the "sit-and-wait" strategy when τ_S is reached, S will give up the entire price–surplus (i.e., propose RP_S). However, if B does not know RP_S, it will still maintain its proposal IP_B at τ_S. Hence, if $IP_B < RP_S$, then B and S may not reach an agreement. To optimize utility and yet guarantee successful negotiation, it is necessary for the agent with the longer deadline (i.e., B) to concede to the ***RP*** of its opponent (i.e., S) when its opponent's deadline is reached.

Similar to [11], this paper devises a set of optimal negotiation strategies for bilateral negotiation that takes into consideration the uncertainty of deadlines and ***RP***s of both agents and models time discounting (Section II). In [11], there are three classes of strategies: *Boulware* (maintaining the initial price until an agent's deadline is almost reached), *Conceder* (conceding rapidly to the ***RP***), and *Linear* (conceding linearly). Whereas [11] showed that there is an optimal strategy class for different scenarios, the question of "what *specific* optimal strategy should an agent adopt to optimize its utility and still guarantee successful negotiation?" has not been answered. Based on the ***RP*** and deadline of B (respectively, S), this work determines the *specific* strategy λ_S (respectively, λ_B) that S (respectively, B) should adopt. For negotiation with complete information, it is proven in Section II that the λ_S (respectively, λ_B) optimizes the utilities of S (respectively, B) and guarantees that agreements are reached.

For negotiation with incomplete information, the following scenarios are studied empirically: 1) when S adopts *BLGAN* to learn RP_B *only* and uses *GA* to search for an appropriate proposal, taking into consideration RP_B, 2) when S adopts *BLGAN* to learn *both* RP_B and τ_B and uses *GA* to search for an appropriate proposal, taking into consideration *both* RP_B and τ_B, and 3) both agents do not have an estimation of each other's ***RP*** and deadline (Section IV). Experiments have also been conducted to compare the performance between agents adopting *BLGAN* and negotiation agents in other works that adopt: 1) *BL* (e.g., [19]) and 2) *GA* (e.g., [20]) (see Section IV). Empirical results in Section IV show that agents adopting *BLGAN* achieved: 1) much higher average utilities and much better *CNO*s than negotiation agents that adopt only *GA* to generate their proposals (e.g., [20]), 2) higher average utilities than negotiation agents that adopt only *BL* to learn ***RP*** [19], and 3) much higher average utilities and generally much better *CNO*s than agents that do not learn their opponents' ***RP***s and deadlines.

II. Optimal Negotiation Strategies

To analyze negotiation with complete information between two agents B and S, it is assumed that $RP_S \in [IP_B, RP_B]$ and $RP_B \in [RP_S, IP_S]$. This is because if $RP_B < RP_S$, no agreement can be reached regardless of the strategies that both agents adopt. Let D be the event in which agent $A \in \{B, S\}$ fails to reach an agreement with its opponent. The utility function of A is defined as $U_A : [IP_A, RP_A] \cup D \to [0, 1]$ such that $U_A(D) = 0$ and for any $l_A \in [IP_A, RP_A]$, $U_A(l_A) > U_A(D)$. Furthermore, if $A = B$, we have $U_B(l_B^1) > U_B(l_B^2)$ when $l_B^1 < l_B^2$; and if $A = S$, then we have $U_S(l_S^1) < U_S(l_S^2)$ when $l_S^1 < l_S^2$.

This section devises a set of optimal strategies that maximizes the utilities of agents while ensuring that agreements are successfully reached because not reaching an agreement is the worst outcome for both B and S [21]. In this work, an *optimal strategy* is defined as follows.

Definition (Optimal Strategy): Let O_t be the final agreement price of agent A that results from its strategy λ_A at agreement time t. λ_A is the optimal strategy for agent A if it maximizes the final utility $U_A(O_t)$.

In this work, both agents adopt the time-dependent strategies in [22], such that the proposal of agent A at time t, $0 \leq t \leq \tau_A$, is determined as follows:

$$P_t^A = IP_A + \left(\frac{t}{\tau_A}\right)^{\lambda_A} (RP_A - IP_A) \tag{1}$$

where $0 \leq \lambda_A \leq \infty$.

Using (1), an agent concedes to its ***RP*** at its deadline, i.e., A will make a proposal RP_A at time $t = \tau_A$ as follows: $P_{\tau_A}^A = IP_A + (\tau_A/\tau_A)^{\lambda_A}(RP_A - IP_A) = RP_A$. Initially, at $t = 0$, $P_0^A = IP_A + (0/\tau_A)^{\lambda_A}(RP_A - IP_A) = IP_A$.

In negotiation with complete information, an agent knows its opponent's ***RP*** and deadline. Hence, finding an optimal strategy for A is to find an exact value of λ_A such that A's final utility is maximized. The problem of finding an optimal strategy for an agent can be analyzed by considering two cases as follows.

Case 1—($\tau_B > \tau_S$): In this case, B's strategy determines whether both agents can reach an agreement before their deadlines. Since S will propose RP_S at τ_S, B must propose a price which is higher than or equal to RP_S at or before τ_S to ensure that an agreement is reached. Hence, it follows that $P_{\tau_S}^B \geq RP_S$, and hence

$$IP_B + \left(\frac{\tau_S}{\tau_B}\right)^{\lambda_B} (RP_B - IP_B) \geq RP_S.$$

Consequently, λ_B must satisfy the condition

$$0 \leq \lambda_B \leq \log_{\frac{\tau_S}{\tau_B}} \frac{RP_S - IP_B}{RP_B - IP_B}$$

to guarantee a successful negotiation.

Theorem 1: Agent B achieves maximal utility when it adopts the strategy

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

Proof: B's utility function is monotonically decreasing. Hence, to maximize its final utility, B needs to minimize its final agreement-price.

The minimal possible agreement-price for B is RP_S. B knows that S must concede to RP_S at τ_S because S makes concession following (1). Therefore, it is advantageous for B to propose a price RP_S at τ_S. This is because of the following: 1) if B proposes RP_S before τ_S, its proposal at τ_S will be higher than RP_S and 2) if B proposes RP_S after τ_S, it will fail to reach an agreement with S. Hence, it follows that the optimal strategy for agent B satisfies $P^B_{\tau_S} = RP_S$, i.e., $\lambda_B = \log_{(\tau_S/\tau_B)}(RP_S - IP_B)/(RP_B - IP_B)$.

Since $\tau_B > \tau_S$, B's strategy will determine if an agreement can be reached regardless of S's strategy. The strategies that S can adopt are as follows: 1) S can adopt the "sit-and-wait" strategy [10] by making its initial proposal at IP_S, maintaining the same proposal until τ_S, and conceding to RP_S at τ_S; 2) S makes its initial proposal at RP_S, then "sit-and-wait" until τ_S; 3) S adopts λ_S, $1 < \lambda_S < \infty$ (maintaining IP_S until *almost* τ_S); 4) S adopts $\lambda_S = 1$ (conceding linearly); and 5) S adopts λ_S, $0 < \lambda_S < 1$ (conceding rapidly to RP_S). However, like the analysis in [10] where the agent with the shorter deadline concedes the entire surplus to the agent with the longer deadline, regardless of the strategy that S adopts, if an agreement is reached, the agreement price for S will always be RP_S.

Case 2: ($\tau_B < \tau_S$): In this case, S's strategy determines whether both agents can reach an agreement before their deadlines. Using a similar analysis as in case 1, the following theorem is established.

Theorem 2: Agent S obtains the maximal utility when it adopts the strategy

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

Symmetrically, regardless of the strategy that B adopts, if an agreement is reached, its agreement price will always be RP_B.

Theorem 1 (respectively, Theorem 2) specifies and proves the optimal strategy of a buyer (respectively, seller) agent for negotiation in complete information (where an agent knows its opponent's *RP* and deadline). The idea of the optimal strategy derived using Theorems 1 and 2 can be adopted to devise new negotiation strategies for bilateral negotiation with incomplete information (i.e., when agents only have estimated values of its opponent's *RP* and deadline through learning). Section III presents an algorithm for estimating an opponent's *RP* and deadline and evolving an agent's strategy during negotiation for bilateral negotiation with incomplete information.

III. *BLGAN* Algorithm

BLGAN has two main procedures for generating an agent's next proposal: a *BL*-procedure and a *GA*-procedure (see Fig. 1). In the *BL*-procedure, a Bayesian updating method is adopted for estimating an opponent's *RP* (Section III-A). Additionally, there is also a process for estimating an opponent's deadline (Section III-B). Based on the estimated reserve price $\widehat{RP}^B_t$ (respectively, $\widehat{RP}^S_t$) and deadline $\hat{\tau}^B_t$ (respectively, $\hat{\tau}^S_t$), S (respectively, B) adjusts its strategy λ^S_t (respectively, λ^B_t) and generates a possible proposal P^{bl}_t (Section III-C). To compensate for possible errors in estimating $\widehat{RP}^B_t$ (respectively, $\widehat{RP}^S_t$) and deadline $\hat{\tau}^B_t$ (respectively, $\hat{\tau}^S_t$), an agent in *BLGAN* adopts a *GA*-procedure to search for a possibly better proposal P^{ga}_t within a dynamic search space (*SP*) confined to an area around P^{bl}_t (Section III-D). Then, the better of the two proposals (P^{bl}_t and P^{ga}_t) is adopted as the new proposal P^S_t (respectively, P^B_t) (Section III-E). The negotiation process terminates when an agreement is reached or if either agent's deadline is reached.

A. Learning Opponent's RP

The *BL*-procedure (based on [21]) for learning an opponent's *RP* is shown in Fig. 2. Suppose that the price range is $[\mathrm{MIN}_P, \mathrm{MAX}_P]$. An agent forms H hypotheses of its opponent's *RP*, where $H = \mathrm{MAX}_P - \mathrm{MIN}_P$. The ith hypothesis of an opponent's *RP* is denoted as RP^{opp}_i, the estimated *RP* of an opponent at round t as $\widehat{RP}^{\mathrm{opp}}_t$, and the opponent's proposal at round t as P^{opp}_t. Denoting the prior probability of RP^{opp}_i as $P(RP^{\mathrm{opp}}_i)$ and the conditional probability of P^{opp}_t given RP^{opp}_i as $P(P^{\mathrm{opp}}_t|RP^{\mathrm{opp}}_i)$, the objective here is to compute $P(RP^{\mathrm{opp}}_i|P^{\mathrm{opp}}_t)$ (the conditional probability of RP^{opp}_i given P^{opp}_t, also called the *posterior probability*, because it is derived from or depends on the specified value of P^{opp}_t).

Initially, it is assumed that the hypotheses follow a uniform distribution [Fig. 2, step 1.b.i.]. At time t ($t > 0$), when an agent receives P^{opp}_t from its opponent, it will update its belief about P^{opp}_t [Fig. 2, steps 1.b.iii. and 1.b.iv.]. The Bayesian updating formula is as follows:

$$P\left(RP^{\mathrm{opp}}_i|P^{\mathrm{opp}}_t\right) = \frac{P_{t-1}\left(RP^{\mathrm{opp}}_i\right) \times P\left(P^{\mathrm{opp}}_t|RP^{\mathrm{opp}}_i\right)}{\sum_{i=1}^{H} P_{t-1}\left(RP^{\mathrm{opp}}_i\right) \times P\left(P^{\mathrm{opp}}_t|RP^{\mathrm{opp}}_i\right)} \tag{2}$$

where

$$P_{t-1}\left(RP^{\mathrm{opp}}_i\right) = P\left(RP^{\mathrm{opp}}_i|P^{\mathrm{opp}}_{t-1}\right) \tag{3}$$

and $P(P^{\mathrm{opp}}_t|RP^{\mathrm{opp}}_i)$ is assumed to be following a normal distribution $N(\mu_i, 1)$. The conditional probability of P^{opp}_t, given RP^{opp}_i [Fig. 2, step 1.b.ii.], can be computed as follows:

$$P\left(P^{\mathrm{opp}}_t|RP^{\mathrm{opp}}_i\right) = \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{\left(P^{\mathrm{opp}}_t - \mu_i\right)^2}{2}}}{\int_{\mathrm{MIN}_P}^{\mathrm{MAX}_P} \frac{1}{\sqrt{2\pi}} e^{-\frac{\left(P^{\mathrm{opp}}_t - \mu_i\right)^2}{2}}} \tag{4}$$

$$\mu_i = RP^{\mathrm{opp}}_i \times \left[1 + (-1)^\beta \times \alpha(t)\right] \tag{5}$$

1. Set the rounds counter t =0.
2. S generates its proposal P_0^S using its initial λ_S, and sends P_0^S to B.
3. If B accepts P_0^S, EXIT.
   Else, B generates its proposal P_0^B using its initial λ_B, and sends P_0^B to S.
4. Increment t by 1.
5. If S accepts P_0^B, EXIT.
   Else, S generates its proposal P_1^S using its initial λ_S, and sends P_1^S to B.
6. If B accepts P_1^S, EXIT.
   Else, B generates its proposal P_1^B using its initial λ_B, and sends P_1^B to S.
7. Increment t by 1.
8. While the negotiation process has not terminated,
   a. If S accepts P_{t-1}^B, EXIT.
   b. If S is not programmed to learn, generate proposal P_t^S using (16) Else,
      I. Execute ***BL*-procedure** to obtain an estimated value of B's reserve price $\widehat{RP}_t^B$.
      II. Compute an estimated value of B's deadline $\hat{\tau}_t^B$ using (10).
      III. Compute a new λ_t^S using (15).
      IV. Generate a possible proposal P_t^{bl} using (16).
      V. Define a search space SP_t^S.
      VI. Execute ***GA*-procedure** on SP_t^S, and return a new proposal P_t^{ga}.
      VII. Set P_t^S to be the better proposal between P_t^{ga} and P_t^{bl}.
   c. Revise the new proposal P_t^S (see III-E).
   d. Send P_t^S to agent B.
   e. If B accepts P_t^S, EXIT.
   f. If B is not programmed to learn, generate proposal P_t^B using (16) Else,
      I. Execute ***BL*-procedure** to get an estimated value of S's reserve price $\widehat{RP}_t^S$.
      II. Compute an estimated value of S's deadline $\hat{\tau}_t^S$ using (12).
      III. Compute a new λ_t^B using (14).
      IV. Generate a possible proposal P_t^{bl} using (16).
      V. Define a search space SP_t^B.
      VI. Execute ***GA*-procedure** on SP_t^B, and return a new proposal P_t^{ga}.
      VII. Set P_t^B to be the better proposal between P_t^{ga} and P_t^{bl}.
   g. Revise the new proposal P_t^B (see III-E).
   h. Send P_t^B to agent S.
   i. Increment t by 1.

Fig. 1. *BLGAN* algorithm.

where $\alpha(t)=|1-P_t^{\text{opp}}\times[1+(-1)^\beta\times\alpha(t-1)]/P_{t-1}^{\text{opp}}|$ when $t>0$, $\alpha(0)=|1-P_0^{\text{opp}}/P_0^{\text{own}}|$ (P_0^{own} represents the agent's own proposal at round $t=0$) and $\beta=1$ for S ($\beta=0$ for B).

1. For each hypothesis RP_i^{opp},
   a. If RP_i^{opp} is impossible, set $P(RP_i^{opp})$ =0.
   b. Else,
      i. If t=0, compute $P(RP_i^{opp})$ using a uniform distribution.
      ii. Else,
         I. Compute μ_i using (5).
         II. Compute $P(P_t^{opp} \mid RP_i^{opp})$ using (4).
      iii. Update $P(RP_i^{opp} \mid P_t^{opp})$ using (2).
      iv. Update $P(RP_i^{opp})$ using (3).
2. Compute $E_{RP}(t)$ using (6).
3. Set $E_{RP}(t)$ as $\widehat{RP}_t^{opp}$.

Fig. 2. *BL*-procedure.

Using (5), it is assumed that, initially, it is very likely for an agent to generate a proposal that is far from its ***RP***. As time passes, it will generate a proposal that is closer to its ***RP***. It should be noted that, in some cases, $P(RP_i^{\text{opp}}|P_t^{\text{opp}})$ is zero. For example, suppose B learns S's ***RP***, then $P(RP_i^{\text{opp}}|P_t^{\text{opp}})$ is equal to zero when $P_{t-1}^S < RP_i^{\text{opp}} \leq \text{MAX}_P$, because $P(P_t^{\text{opp}}|RP_i^{\text{opp}})=0$ when $P_{t-1}^S < RP_i^{\text{opp}} \leq \text{MAX}_P$ (from B's perspective, S will not generate a proposal that is lower than its ***RP***). Similarly, if S is programmed to learn its opponent's ***RP***, $P(RP_i^{\text{opp}}|P_t^{\text{opp}})=0$ when $\text{MIN}_P \leq RP_i^{\text{opp}} < P_{t-1}^S$.

Then, an expected value of RP^{opp} at round t can be computed (Fig. 2, step 2) by

$$E_{\text{RP}}(t)=\sum_t RP_i^{\text{opp}}\times P\left(RP_i^{\text{opp}}|P_t^{\text{opp}}\right). \tag{6}$$

When $t=0$, $P(RP_i^{\text{opp}})$ is used instead of $P(RP_i^{\text{opp}}|P_t^{\text{opp}})$ in (6).

B. Estimating Opponent's Deadline

This section discusses how an agent estimates its opponent's deadline. Suppose agent S is programmed to learn agent B's deadline. In ***BLGAN***, both agents (B and S) generate their proposals as follows: $P_t=P_{t-1}+(-1)^\beta\times(t/\tau)^\lambda\times|RP-P_{t-1}|$, where $\beta=1$ for S ($\beta=0$ for B). Since both agents use the same model for generating proposals, it is plausible for S to assume that B uses the same model to generate proposals as follows:

$$P_t^B-P_{t-1}^B=\left[\frac{t}{\tau_B}\right]^{\lambda_B}\times\left|RP_B-P_{t-1}^B\right|. \tag{7}$$

Equation (7) follows the form in (1) (see Section II) by replacing initial price (IP_A) in (1) by B's proposal P_{t-1}^B at the previous round (i.e., at round $t-1$). The rationale is that, at every round t when B adjusts its strategy, it starts a "new" negotiation process by treating P_{t-1}^B as its new "initial price" (see Section III-C).

For a series of three negotiation rounds $t-2$, $t-1$, and $t(t \geq 2)$, the following equations can be obtained from (7):

$$P_{t-1}^B - P_{t-2}^B = \left[\frac{t-1}{\tau_B}\right]^{\lambda_B} \times |RP_B - P_{t-2}^B| \quad (8)$$

$$P_t^B - P_{t-1}^B = \left[\frac{t}{\tau_B}\right]^{\lambda_B} \times \left|RP_B - P_{t-1}^B\right|. \quad (9)$$

In negotiation with incomplete information, S does not know RP_B. Hence, $\widehat{RP_B}$ (an estimated value of RP_B obtained from the *BL*-procedure) can be used to replace RP_B in (8) and (9). From (8) and (9), $\hat{\tau}_t^B$ (an estimated value of τ_B) can be computed by substituting τ_B with $\hat{\tau}_t^B$ and RP_B with $\widehat{RP_B}$ as follows:

$$\hat{\tau}_t^B = \frac{t}{\left[\frac{P_t^B - P_{t-1}^B}{\left|\widehat{RP_B} - P_{t-1}^B\right|}\right]^{1/\lambda_B}} \quad (10)$$

where λ_B is determined as follows:

$$\lambda_B = \log_{\left(\frac{t-1}{t}\right)}\left(\frac{P_{t-1}^B - P_{t-2}^B}{P_t^B - P_{t-1}^B} \times \frac{\left|\widehat{RP_B} - P_{t-1}^B\right|}{\left|\widehat{RP_B} - P_{t-2}^B\right|}\right). \quad (11)$$

However, at both $t=0$ and $t=1$, S (respectively, B) generates its proposals using its initial strategy λ_S (respectively, λ_B) and waits until $t=2$ before it can start to estimate B's (respectively, S's) deadline and adjust its own strategy (see Fig. 1, steps 2–7). If $P_{t-1}^B = P_{t-2}^B$, or $P_t^B = P_{t-1}^B$, or $P_{t-1}^B = \widehat{RP_B}$, or $P_{t-2}^B = \widehat{RP_B}$, (i.e., (10) and (11) have no solution), then the previous estimated value of τ_B is used.

Similarly, when B is programmed to learn S's deadline, it follows the same computation method given earlier. An estimated value of τ_S can be computed by substituting τ_S with $\hat{\tau}_t^S$ and RP_S with $\widehat{RP_S}$ as follows:

$$\hat{\tau}_t^S = \frac{t}{\left[\frac{P_{t-1}^S - P_t^S}{\left|\widehat{RP_S} - P_{t-1}^S\right|}\right]^{1/\lambda_S}} \quad (12)$$

where $\widehat{RP_S}$ is an estimation of S's *RP* and λ_S is determined as follows:

$$\lambda_S = \log_{\left(\frac{t-1}{t}\right)}\left(\frac{P_{t-2}^S - P_{t-1}^S}{P_{t-1}^S - P_t^S} \times \frac{\left|\widehat{RP_S} - P_{t-1}^S\right|}{\left|\widehat{RP_S} - P_{t-2}^S\right|}\right). \quad (13)$$

If $P_{t-1}^S = P_{t-2}^S$, or $P_t^S = P_{t-1}^S$, or $P_{t-1}^S = \widehat{RP_S}$, or $P_{t-2}^S = \widehat{RP_S}$, (i.e., (12) and (13) have no solution), then the previous estimated value of τ_S is used.

C. Adjusting Negotiation Strategy

B adjusts its strategy as follows. At each round $R_t = t$, if an agreement is not reached, B will first estimate S's *RP*, $\widehat{RP}_t^S$, (Section III-A) and deadline $\hat{\tau}_t^S$ using previous proposals of S (Section III-B). Using these estimated parameters of S and under the assumption that $\widehat{RP}_t^S \in (P_{t-1}^B, RP_B)$, B treats its proposal P_{t-1}^B at $t-1$ as its new "initial price" and adjusts its strategy by setting λ_t^B to

$$\log_{\frac{\hat{\tau}_t^S - R_t + 1}{\tau_B - R_t + 1}}\left(\frac{\widehat{RP}_t^S - P_{t-1}^B}{RP_B - P_{t-1}^B}\right)$$

to start a new negotiation process. However, there are two special cases for the estimated value $\widehat{RP}_t^S$ of agent S, $\widehat{RP}_t^S \leq P_{t-1}^B$ and $\widehat{RP}_t^S \geq RP_B$, that need to be considered.

1) $\widehat{RP}_t^S \leq P_{t-1}^B$. In this case, the value $(\widehat{RP}_t^S - P_{t-1}^B)$ is negative. To deal with this situation, $(\widehat{RP}_t^S - P_{t-1}^B)$ is set to $\max\{0, (\widehat{RP}_t^S - P_{t-1}^B)\}$. Hence, at round $R_t = t$, when $\widehat{RP}_t^S < P_{t-1}^B$, the value of λ_t^B can be determined as $\log_{(\hat{\tau}_t^S - R_t + 1)/(\tau_B - R_t + 1)} 0 = +\infty$, i.e., B adopts the "sit-and-wait" strategy. Since the minimal possible agreement price for B is S's *RP*, if $\widehat{RP}_t^S \leq P_{t-1}^B$, B can maintain its previous proposal P_{t-1}^B and wait for S to decrement its price to P_{t-1}^B. This is because B believes that S may still decrease its price to a price that is equal to or lower than P_{t-1}^B because B's estimation of S's *RP*, $\widehat{RP}_t^S$, is equal to or lower than P_{t-1}^B.
2) $\widehat{RP}_t^S \geq RP_B$. If the *exact RP*, RP_S, of S is higher than RP_B, S and B can never reach an agreement. The best course of action for B is to terminate the negotiation immediately to avoid wasting computational resources in haggling in a fruitless negotiation. However, when $\widehat{RP}_t^S \geq RP_B$, it is still possible that the exact *RP* of S is in the price range of B, i.e., $RP_S \in [IP_B, RP_B]$. Hence, in this case, the absolute value of λ_t^B is adopted, i.e.,

$$\lambda_t^B = \left|\log_{\frac{\hat{\tau}_t^S - R_t + 1}{\tau_B - R_t + 1}}\left(\frac{\widehat{RP}_t^S - P_{t-1}^B}{RP_B - P_{t-1}^B}\right)\right|.$$

Hence, at each round $R_t = t$, λ_t^B can be calculated by

$$\lambda_t^B = \left|\log_{\frac{\hat{\tau}_t^S - R_t + 1}{\tau_B - R_t + 1}}\left(\max\left\{0, \frac{\widehat{RP}_t^S - P_{t-1}^B}{RP_B - P_{t-1}^B}\right\}\right)\right|. \quad (14)$$

Similarly, using the estimated parameters of B, $\widehat{RP}_t^B$, and $\hat{\tau}_t^B$ and treating P_{t-1}^S as S's new "initial price," λ_t^S can be determined as follows:

$$\lambda_t^S = \left|\log_{\frac{\hat{\tau}_t^B - R_t + 1}{\tau_S - R_t + 1}}\left(\max\left\{0, \frac{P_{t-1}^S - \widehat{RP}_t^B}{P_{t-1}^S - RP_S}\right\}\right)\right|. \quad (15)$$

An agent determines its next proposal as follows:

$$P_t = P_{t-1} + (-1)^\beta \times K_t \times \left(\frac{1}{\tau - t + 1}\right)^\lambda \quad (16)$$

1. Define the search space SP_t.
2. Set the generations counter g=0.
3. Set the population size N.
4. Initialize the population $P(g)$.
5. Calculate the fitness for each individual in $P(g)$.
6. While g is smaller than the maximum number of generations,
    a. Copy all the individuals to a temporary population $TP(g)$.
    b. Perform crossover on $TP(g)$.
    c. Perform mutation on $TP(g)$.
    d. Calculate the fitness of each individual in $TP(g)$.
    e. Use tournament selection to select N individuals from $TP(g)$ and $P(g)$, to form a new population $P(g+1)$.
    f. Increment g by 1.
7. Return the best individual in the last generation.

Fig. 3. *GA*-procedure.

where $\beta = 1$ for S ($\beta = 0$ for B), and $K_t = |RP - P_{t-1}|$ is the difference between its *RP* and its proposal at $t-1$. Equation (16) follows the form in (1) (see Section II) by substituting IP_A (initial price of an agent) in (1) with P_{t-1}, since an agent treats its proposal P_{t-1}^B at $t-1$ as its new "initial price." Since an agent starts a "new" negotiation at $t-1$, $(t/\tau_A)^\lambda$ in (1) (where τ_A is agent A's deadline) is substituted by $((t-(t-1))/(\tau-(t-1)))^\lambda = (1/(\tau-t+1))^\lambda$.

D. GA-Procedure

The *GA*-procedure is shown in Fig. 3. Using a real coding mechanism, each individual (representing a possible proposal) is a real number in the *SP*. For S (respectively, B) at the beginning of each execution of *GA*, the *SP*, SP_t^S, (respectively, SP_t^B) is (dynamically) defined after P_t^{bl} is calculated using (16) (in the negotiation process at each negotiation round t). For B, the *SP* at round t is $[\max\{P_{t-1}^B, P_t^{bl}-\delta\}, \min\{P_{t-1}^S, P_t^{bl}+\delta, RP_B\}]$. For S, the *SP* at round t is $[\max\{P_{t-1}^B, P_t^{bl}-\delta, RP_S\}, \min\{P_{t-1}^S, P_t^{bl}+\delta\}]$. Hence, at different negotiation rounds, the *SP* for agent S (respectively, B) is dynamically changed to an area around P_t^{bl}. Based on experimental tuning, for $P_t^{bl} \in [1, 100]$, δ is set to 10.

Let an individual represent a proposal value P^o of an agent. Denoting the agent's opponent's proposal as P^{opp}, the fitness of P^o is determined as follows:

$$\text{fitness}(P^o) = w(t) \times U(P^o) + [1-w(t)] \times [1-\text{Dist}(P^o, P^{\text{opp}})]$$

where $U(P^o)$ determines the utility generated by P^o (see Section IV for the definition of utility function in this work), $w(t)$ is a weight parameter computed by $w(t) = \alpha \times [1-(t/\tau)^2]$, t is the number of rounds, τ is the agent's deadline, and $\text{Dist}(P^o, P^{\text{opp}})$ is the normalized distance between P^o and P^{opp}. $\text{Dist}(P^o, P^{\text{opp}})$ is computed as follows:

$$\text{Dist}(P^o, P^{\text{opp}}) = \frac{|P^o - P^{\text{opp}}|}{\text{MAX}_P - \text{MIN}_P}$$

where $[\text{MIN}_P, \text{MAX}_P]$ is the price range of an agent.

By varying α (where $\alpha \in [0,1]$), $w(t)$ can be used to control different proposals (individuals) with different negotiation outcomes to be generated. With a larger α (and correspondingly $w(t)$), proposals that achieve higher utilities for an agent are more likely to be generated (see Section IV). When α is small, proposals that are closer to an opponent's proposal are more likely to be generated—i.e., striving to enhance negotiation success rates by placing more emphasis on $\text{Dist}(P^o, P^{\text{opp}})$. Furthermore, empirical results reported in [21] show that this also facilitates reaching faster agreements.

E. Revising Possible Proposals

For each possible proposal P_t^S of S, it is prudent to set P_t^S to be higher than or equal to its own *RP* and B's last proposal P_{t-1}^B and to be lower than or equal to its own last proposal P_{t-1}^S. That is, it is prudent to set P_t^S in the region $[\max(P_{t-1}^B, RP_S), P_{t-1}^S]$. Similiarly, for B, it is prudent to set each possible proposal P_t^B in the region $[P_{t-1}^B, \min(P_{t-1}^S, RP_B)]$). If P_t^S (respectively, P_t^B) exceeds the edge points of $[\max(P_{t-1}^B, RP_S), P_{t-1}^S]$ (respectively, $[P_{t-1}^B, \min(P_{t-1}^S, RP_B)]$), it should be set to the value of the nearest edge point.

IV. Empirical Results

To evaluate *BLGAN* empirically, three sets of experiments were carried out between two negotiation agents B and S to compare negotiation with incomplete information that adopts *BLGAN* to adjust an agent's strategy with the following negotiation scenarios:

1) Negotiation with complete information in which an agent adopts its optimal strategy (determined using Theorems 1 and 2) and negotiation with incomplete information in which an agent does not learn its opponent's *RP* and deadline (called the "no-learn" strategy).
2) Negotiation with incomplete information in which an agent adopts *BL* [19] to learn its opponent's *RP* and adjusts its proposals based on its estimations of its opponent's *RP*.
3) Negotiation with incomplete information in which an agent adopts *GA* [20] to generate its best proposal.

BLGAN versus "no-learn" Strategy: In the first set of experiments, the following scenarios were studied.

1) "*CompleteInfo*": B and S know each other's *RP* and deadline, and they adopt the optimal strategy in Theorems 1 and 2, respectively.
2) "*IncompleteInfo*": B and S adopt the "no-learn" strategy, i.e., they do not know each other's *RP* and deadline, and their strategies λ_B and λ_S remain fixed throughout the negotiation.
3) "*BLGAN-S-learns-RP*": S adjusts λ_S by adopting *BLGAN* to learn RP_B, and B's strategy λ_B remains fixed throughout the negotiation.
4) "*BLGAN-S-learns-RP-deadline*": S adjusts λ_S by adopting *BLGAN* to learn RP_B and τ_B, and B's strategy λ_B remains fixed throughout the negotiation. This was compared with scenarios 1) and 2).

Nevertheless, bilateral negotiation with complete information (i.e., both agents know each other's deadline and *RP*) is an ideal scenario that is extremely rare in real world situations. It is NOT the intention of this paper to compare *BLGAN* strategy in bilateral negotiation with incomplete information with the optimal strategy in bilateral negotiation with complete information. Instead, in Fig. 4, the empirical results of the optimal strategy in bilateral negotiation with complete information (i.e., "*CompleteInfo*") is used as a yardstick for comparing and evaluating the empirical results of the following: 1) "*BLGAN-S-learns-RP-deadline*," 2) "*BLGAN-S-learns-RP*," and 3) "*IncompleteInfo*."

BLGAN versus *BL*: In the second set of experiments, the following scenarios were studied.

1) "*BL-S-learns-RP*": S adjusts λ_S by adopting the *BL* in [19] to learn RP_B, and B's strategy λ_B remains fixed throughout the negotiation.
2) "*BLGAN-S-learns-RP-deadline*" (as described above).

BLGAN versus *GA*: In the third set of experiments, the following scenarios were studied.

1) "*GA-S*": S adopts the *GA* in [20] to generate a proposal at each negotiation round, and B adopts a strategy λ_B that remains fixed throughout the negotiation.
2) "*BLGAN-S-learns-RP-deadline*" (as described above).

Implementation: The experiments were carried out using a testbed implemented using C++. The testbed consists of two negotiation agents, and each agent was coded as an instance of a user-defined class. Besides its private information (*RP* and deadline), each agent was also designed to record historical proposals of its opponent. To evaluate the *BLGAN* algorithm, the following software components were developed for each agent: 1) a component for learning its opponent's *RP*, 2) a component for estimating its opponent's deadline, and 3) a component for generating its proposal based on the learning results (this includes adjusting the negotiation strategy of the agent and revising possible proposals).

Experimental Settings: In each of the three sets of experiments, 50 random runs for each scenario were carried out, and in each run, S (respectively, B) was programmed with the same deadline, *RP*, and initial price for all the scenarios. Initially, λ_B and λ_S were randomly selected from [0.2, 10]. In the *BL*-procedure, the range of possible prices for each agent's proposal and *RP* is from 1 to 100 units, and both agents' deadlines τ_B and τ_S are between 9 to 100 rounds. The *GA*-procedure executes for a maximum number of 200 generations with a population size of 50, a tournament size of 7, and crossover and mutation rates of 0.8 and 0.1, respectively. Through experimental tuning, α is set to be 0.9 when only S learns (see the Appendix).

Performance Measures: The following four performance measures were used: 1) Success Rate (R_{success}), 2) Average Negotiation Speed (*ANS*), 3) Average Utility (*AU*), and 4) *CNO*. Whereas *ANS* is measured by the average number of rounds needed to reach an agreement, $R_{\text{success}} = N_{\text{success}}/N_{\text{total}}$, where N_{success} is the number of successful deals and N_{total} is the total number of deals. An agent's utility function is defined as follows. Let $l_{\min}$ and $l_{\max}$ (respectively, $l_{\max}$ and $l_{\min}$) be the initial and *RP*s, respectively, for B (respectively, S), and l_c be the price that a consensus is reached by B and S. An agent's utility $U(l_c)$ for reaching a consensus at l_c is given as follows:

$$U(l_c) = \begin{cases} u_{\min} + (1-u_{\min})[(l_{\max}-l_c)/(l_{\max}-l_{\min})], & \text{for } B \\ u_{\min} + (1-u_{\min})[(l_c-l_{\min})/(l_{\max}-l_{\min})], & \text{for } S \end{cases}$$

where $u_{\min}$ is the minimum utility that an agent receives for reaching a deal at its *RP*. For experimentation purpose, the value of $u_{\min}$ is defined as 0.1. Assigning zero or a value that is too close to zero does not distinguish the utilities between deals and no deals (since an agent receives a utility of zero if negotiation fails). However, assigning a value that is too high may not significantly distinguish the preference orderings of agents.[1]

Hence, $AU = (1/N_{\text{success}}) \sum_i^{N_{\text{success}}} U(l_c)$.

An agent's *CNO* is determined by $CNO = R_{\text{success}} \times AU \times (NNS)^{-1}$ where *NNS* is an agent's normalized *ANS* defined as follows. In a single negotiation process, if an agent S (respectively, B) reaches a consensus with its opponent B (respectively, S) at round t_c, then the normalized negotiation speed is $t_c/(\min\{\tau_S, \tau_B\})$. Hence, for N_{total} negotiation, *NNS* is determined as follows:

$$NNS = \frac{1}{N_{\text{success}}} \sum_{i=1}^{N_{\text{total}}} \frac{t_i}{\min\left\{\tau_S^i, \tau_B^i\right\}}.$$

Results: Empirical results recorded from S's perspective are shown in Figs. 4–6. From these results, five observations are drawn as follows.

Observation 1: S in "*BLGAN-S-learns-RP*" achieved: 1) *much* higher average utilities and better *CNO*s than S in "*IncompleteInfo*," and 2) faster *ANS* than S in "*CompleteInfo*."

Analysis: Fig. 4(a)–(d) show the comparison of "*BLGAN-S-learns-RP*" (and "*BLGAN-S-learns-RP-Deadline*") with "*CompleteInfo*" and "*IncompleteInfo*," and it is observed that, in "*BLGAN-S-learns-RP*," S generally achieved a 100% success rate (except for very short deadlines, i.e., 9 to 20 rounds) [Fig. 4(a)] and much higher average utilities and better combined outcomes than in the "*IncompleteInfo*" situation [Fig. 4(b) and (d)]. With a larger α (see Section III-D), S in "*BLGAN-S-learns-RP*" strives to achieve higher average utilities at the expense of using more negotiation rounds than in the "*IncompleteInfo*" situation to reach agreements. On the other hand, the average utilities and combined outcomes of S in "*BLGAN-S-learns-RP*" were better than the "*IncompleteInfo*" situation and closer to the optimal results in the "*CompleteInfo*" situation (proven in Section II). In "*BLGAN-S-learns-RP*," S achieved much faster *ANS* than in the "*CompleteInfo*" situation [Fig. 4(c)]. This is because, in the "*CompleteInfo*" situation, S will *only* reach an agreement either at its own deadline or the deadline of B, whichever is shorter, while S in "*BLGAN-S-learns-RP*" can reach an agreement earlier than the shorter deadline of B and S.

[1] The authors acknowledge that the utility function used in this work is simple. However, the intention of this work is to focus on designing a learning method for negotiation in the presence of incomplete information, and not on the utility function *per se*.

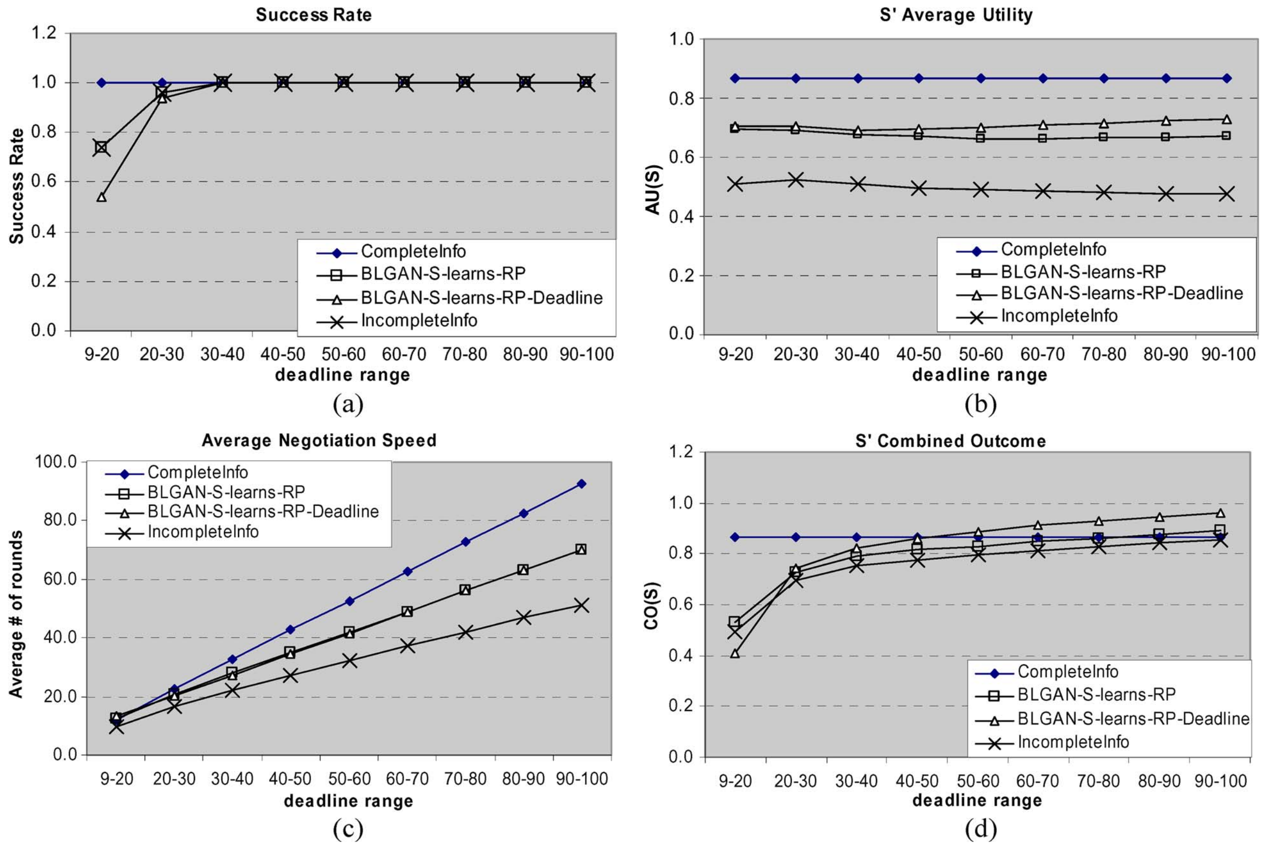

Fig. 4. *BLGAN* (S learns) versus *Optimal* and *"no-learn" Strategies*.

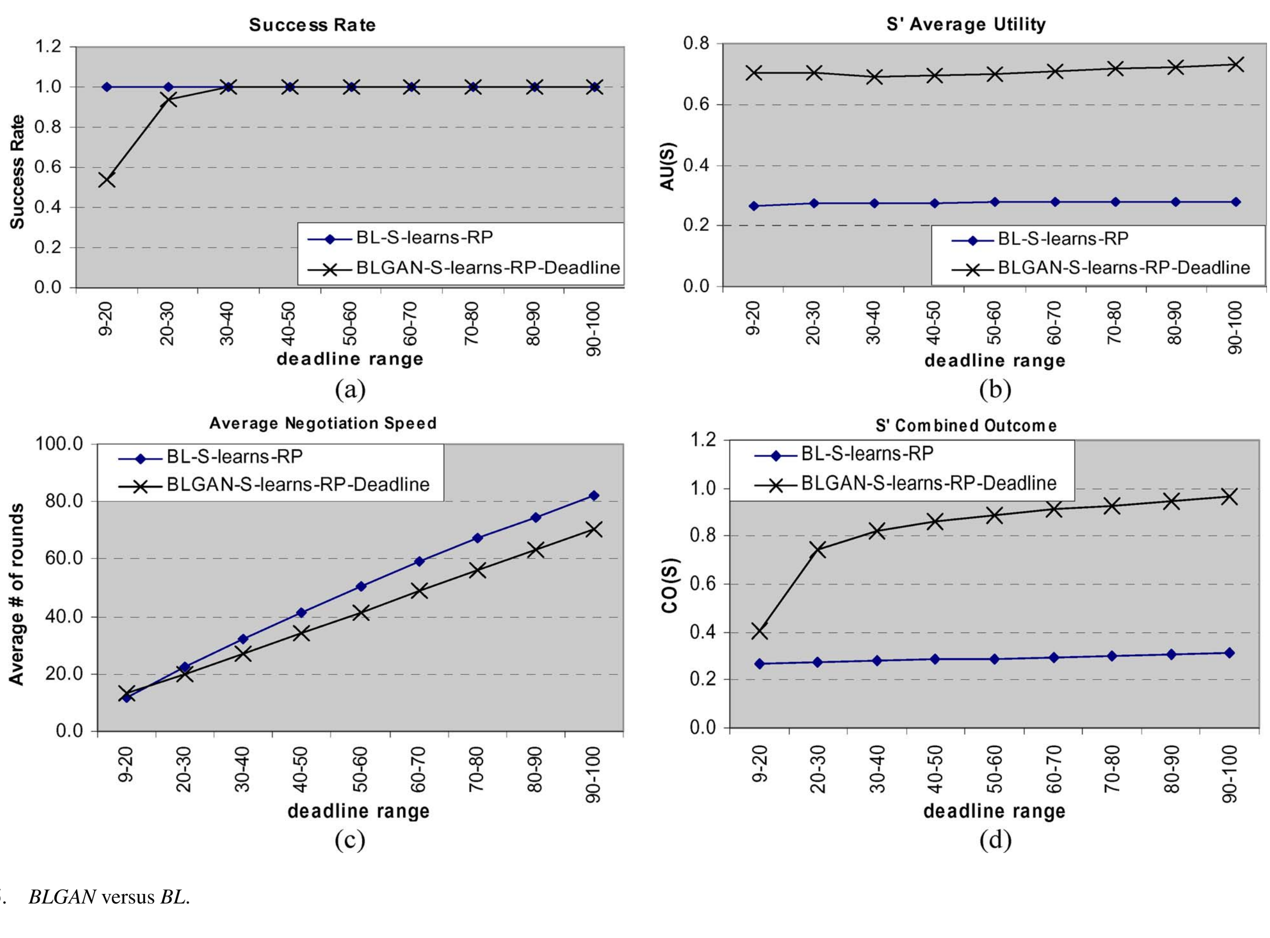

Fig. 5. *BLGAN* versus *BL*.

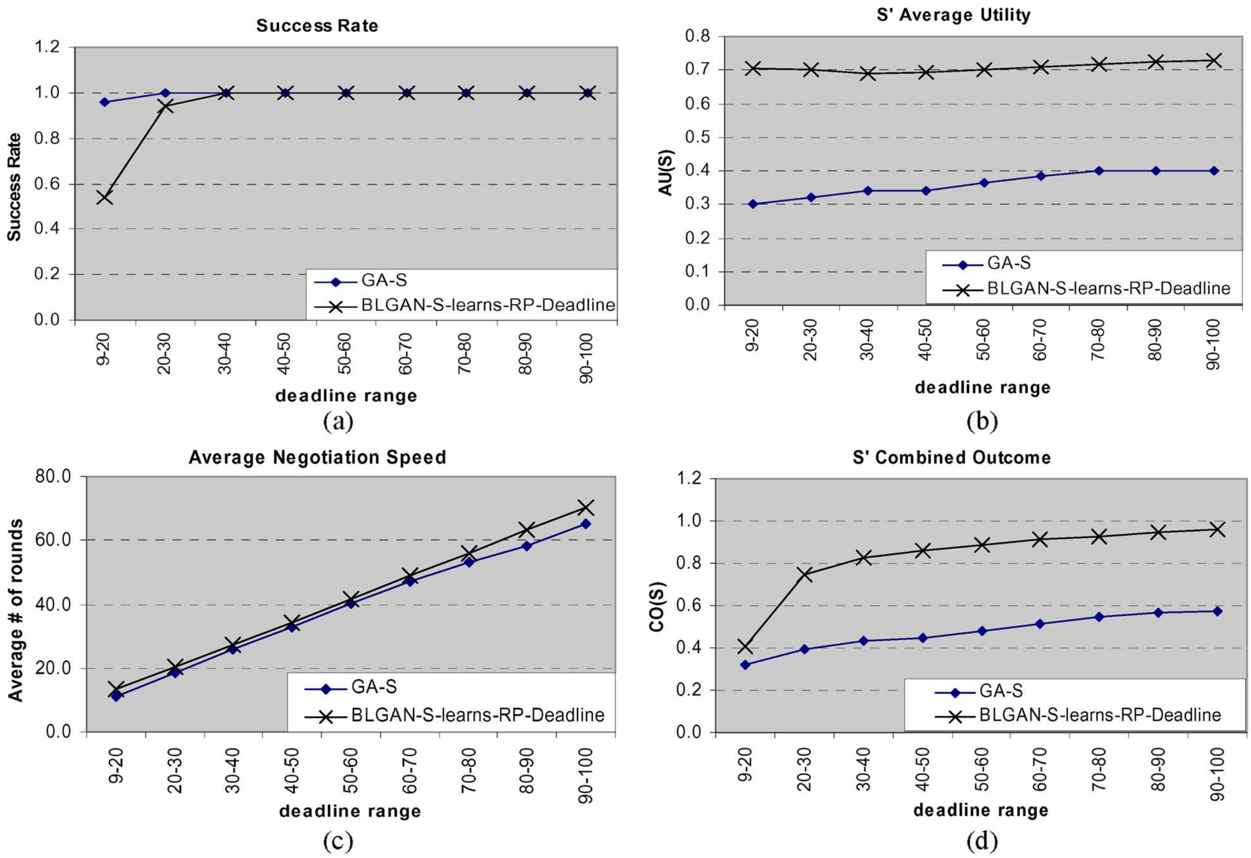

Fig. 6. *BLGAN* versus *GA*.

Observation 2: S in "BLGAN-S-learns-S-Deadline" achieved the following: 1) *much* higher average utilities and better *CNO*s than S in "*IncompleteInfo*," and 2) faster *ANS* and better combined outcomes (for longer deadlines) than S in "*CompleteInfo*."

Analysis: It can be observed from Fig. 4(a)–(d) that, in "*BLGAN-S-learns-RP-Deadline*," S generally achieved a 100% success rate (except for very short deadlines, i.e., 9 to 20 rounds) [Fig. 4(a)] and much higher average utilities, and generally, much better combined outcomes than in the "*IncompleteInfo*" situation [Fig. 4(b) and (d)]. Similar to "*BLGAN-S-learns-RP*," with a larger α, S in "*BLGAN-S-learns-RP-Deadline*" strives to achieve higher average utilities at the expense of using more negotiation rounds than in the "*IncompleteInfo*" situation to reach agreements. For the same reason described in "*BLGAN-S-learns-RP*," in "*BLGAN-S-learns-RP-Deadline*," S also achieved much faster *ANS* than in the "*CompleteInfo*" situation [Fig. 4(c)]. Furthermore, for longer deadlines, S in "*BLGAN-S-learns-RP-Deadline*" achieved better combined outcomes than S in "*CompleteInfo*."

Observation 3: S in "BLGAN-S-learns-RP-Deadline" achieved higher average utilities and better *CNO*s than S in "*BLGAN-S-learns-RP*."

Analysis: It can be observed from Fig. 4(b) and (d) that S in "*BLGAN-S-learns-RP-Deadline*" achieved higher average utilities and better *CNO*s than S in "*BLGAN-S-learns-RP*." From Fig. 4(a) and (c), it is observed that both S in "*BLGAN-S-learns-RP-Deadline*" and S in "*BLGAN-S-learns-RP*" generally achieved a 100% success rate (except for short deadlines) and used the same number of negotiation rounds to reach agreements. These observations show that an agent adopting *BLGAN* achieved better performance when it learns more private information about its opponent. By having estimations of *both* the *RP* and deadline of its opponent, S (respectively, B) is *more likely* to adjust its strategy λ_t^S (respectively, λ_t^B) in each negotiation round to a value that is closer to the optimal strategy for complete information negotiation (proven in Section II) than when it *only* has an estimation of its opponent's *RP* [21]. Theorems 1 and 2 (Section II) show that B (respectively, S) achieves the optimal utility if it adopts a strategy λ_B (respectively, λ_S) determined using a formula that depends on *both* the *RP* and deadline of B's (respectively, S's) opponent. On this account, this paper has enhanced the work in [21] by obtaining significantly better empirical results.

Observation 4: S in "BLGAN-S-learns-RP-Deadline" achieved *much* higher average utilities and *much* better *CNO*s than S in "*BL-S-learns-RP*."

Analysis: Fig. 5(a)–(d) show the comparison of "*BLGAN-S-learns-RP-Deadline*" with "*BL-S-learns-RP*." It is observed that, in "*BLGAN-S-learns-RP-Deadline*," S generally achieved a 100% success rate (except for very short deadlines, e.g., 9 to 20 rounds) [Fig. 5(a)], *much* higher average utilities and *much* better combined outcomes than in "*BL-S-learns*-RP" [Fig. 5(b) and (d)]. "*BL-S-learns-RP*" achieved much lower average utilities (between 0.2 and 0.3) as compared to between 0.7 and 0.8 for S in "*BLGAN-S-learn-RP-Deadline*" [Fig. 5(b)]. This is because, in [19], S generates its proposal Pro_S between its own *RP*, RP_S, and the estimated value $\widehat{RP}_B$ of its opponent's *RP* as follows: $\text{Pro}_S = \delta \times \widehat{RP}_B + (1-\delta)RP_S$. In [19, p. 18], δ was set to 0.1, and the experimental settings in this work followed the same settings given in [19, p. 18]. This means that $\text{Pro}_S = 0.1 \times \widehat{RP}_B + (1-0.1)RP_S$ and the proposals of S in "*BL-S-learns-RP*" are close to its own *RP*, and hence, this also means that it made very large amounts of concessions. Even though S in "*BLGAN-S-learns-RP-Deadline*" used fewer negotiation rounds than S in "*BL-S-learns-RP*" to reach agreements [Fig. 5(c)], the authors acknowledge that agents in [19] are designed for enhancing joint utilities and negotiation speed of both agents by learning each other's *RP* using *BL*. Experiments in this work were also conducted to compare the following scenarios: 1) when both agents (B and S) adopt *BLGAN* to learn *RP* and deadline with 2) when both B and S adopt *BL* to learn each other's *RP*. Whereas space limitation precludes these empirical results from being included here, the authors would like to summarize that, when B and S learn each other's private information, agents in [19] achieved faster negotiation speed and better *CNO*s than *BLGAN* agents. When both *BL* agents in [19] make large amounts of concessions, they are more likely to reach faster agreements (at the expense of achieving lower average utilities). However, in [23, p. 145], it was noted that if an agent concedes too much, it "wastes" some of its utility, and this is inefficient. Hence, *BLGAN* agents are designed to maintain a balance among: 1) optimizing utilities, 2) obtaining reasonably good negotiation speed (even though when both B and S learn about the opponent's private information, they used more negotiation rounds than agents in [19]), and 3) reaching agreements successfully.

Observation 5: S in "*BLGAN-S-learns-RP-Deadline*" achieved much higher average utilities and much better *CNO*s than S in "*GA-S*."

Analysis: Fig. 6(a)–(d) show the comparison of "*BLGAN-S-learns-RP-Deadline*" with "*GA-S*," and it is observed that, in "*BLGAN-S-learns-RP-Deadline*," S generally achieved a 100% success rate (except for very short deadlines, i.e., 9 to 20 rounds) [Fig. 6(a)], much higher average utilities and much better combined outcomes than in "*GA-S*" [Fig. 6(b) and (d)]. With a larger α (see Section III-D), S in "*BLGAN-S-learns-RP-Deadline*" strives to achieve high utilities, but in this case, it used almost the same average number of negotiation rounds as "*GA-S*" to reach agreements [Fig. 6(c)]. At each negotiation round t, S in "*BLGAN-S-learns-RP-Deadline*" confined its *SP* *only* to the possible proposals around the proposal computed from its strategy λ_t^S determined using the estimated values of an opponent's deadline and *RP*. λ_t^S is derived from the *same* formula as the agent's optimal strategy (Theorem 2, Section II) by replacing its "initial price" with its proposal in round $t-1$. By treating its current proposal as its "initial price," at each round t, S starts a "new" negotiation process by attempting to adjust λ_t^S to its optimal strategy based on its updated estimation of its opponent's *RP* and deadline. Hence, if agreements are reached at S's proposals determined using *BLGAN*, then S is *more* likely to obtain higher average utilities since the proposals generated using *BLGAN* are more likely to be closer to the proposal generated using S's optimal strategy. On the other hand, the *GA* of S in "*GA-S*" searched the entire *SP* of all possible proposals in the interval between S's proposal at round $t-1$ and B's proposal at round $t-1$. Hence, it is comparatively less likely than S in "*BLGAN-S-learns-RP-Deadline*" to generate proposals that are close to the proposal generated by S's optimal strategy.

V. Related Work

The literature that relates to this work includes the following: 1) negotiation agents adopting *GA* [20], [24]–[27] (Section V-A) and 2) negotiation agents adopting *BL* [19], [28], [29] (Section V-B). Space limitation precludes all these works from being introduced here, and this section only discusses some of the more closely related works (e.g., [19], [20], [24]–[27]).

A. Negotiation Agents Adopting GA

In the literature on applying *GA* to enhancing automated negotiation, *GA*s are used to: 1) evolve the best strategies [24], 2) generate proposals at every round [20], 3) track shifting tactics and changing behaviors [25], and 4) learn effective rules for supporting negotiation [26]. Furthermore, [27] presented a novel *GA* with a new genetic operator for concession making in negotiation.

Reference [24] utilized *GA* for learning the most successful class of bargaining strategies in different circumstances (e.g., when an agent is facing different opponents). In their negotiation model, an agent's strategy is based on time-dependent, resource-dependent, and behavior-dependent negotiation decision functions (*NDF*s). An agent adopting time-dependent *NDF*s considers both deadlines and time preferences. Whereas resource-dependent *NDF*s generate proposals based on how a resource (e.g., remaining bandwidth) is being consumed; in behavior-dependent *NDF*s, an agent generates its proposal by replicating (a portion of) the previous attitude of its opponent. Represented as a gene in [24], an agent's strategy is based on a combination of the time-dependent, resource-dependent, and behavior-dependent *NDF*s. By placing different weightings on the time-dependent, resource-dependent, and behavior-dependent *NDF*s, different strategies can be composed. The basic genetic operators: reproduction, crossover, and mutation were used in [24] for generating new (and better) strategies. In their *GA*, tournament selection is used to create the mating pool of the genes that form the basis for the next population. While *GA* is used in [24] for evolving the most successful strategy classes against different types of opponents in different environments, this work uses the synergy of *GA* and *BL* for determining an agent's optimal strategy and generating the best proposal at each negotiation round. In this work, an agent determines its optimal strategy using its estimations of an opponent's *RP* and deadline using a *BL*-procedure and a deadline-estimation process, respectively. To compensate for possible errors in the *BL*-procedure and the deadline-estimation process, *GA* is used to search for a possibly better proposal within a dynamic *SP* confined to an area around a proposal generated by an agent's current strategy determined using the estimated values of an opponent's deadline and *RP*.

Reference [20] devised an evolutionary learning approach for designing negotiation agents. Negotiation agents in [20] are designed not only to optimize an agent's individual payoff but also to strive to ensure that a consensus is reached. A subset of feasible offers at a negotiation round is represented as a population of chromosomes, and each chromosome encodes an offer using a fixed number of fields. In their *GA*, the effectiveness of a negotiation solution is evaluated using a fitness function that determines both the following: 1) the similarity of a negotiation solution to an opponent's proposal according to a *weighted Euclidean distance function* and 2) the optimality of the negotiation solution. Similar to [24], the *GA* in [20] utilizes reproduction, crossover, and mutation, and at each iteration, either tournament or Roulette-wheel selection is used to select chromosomes from the current population for creating a mating pool. Empirical results in [20] seem to indicate that their *GA*-based negotiation agents can acquire effective negotiation tactics. Whereas the *GA* in [20] searches the entire *SP* of all possible proposals in the interval between an agent's proposal and its opponent's proposal, the *GA* in this work focuses its search *only* around the proposal generated by an agent's strategy derived from the *same* formula as the agent's optimal strategy. This enables the *GA* in this work to focus its search on a specific region in the *SP* with proposals that are more likely to be closer to the proposal that is generated by the optimal strategy. In doing so, empirical results (Section IV) have shown that agents in this paper are more likely to achieve higher average utilities than agents in [20].

In the adaptive negotiation agents (*ANA*s) by Krovi *et al.* [25], decision making of a negotiator is modeled with computational paradigms based on *GA*. *GA* is used for tracking the shifting negotiation tactics and changing preferences of negotiators. Novel features of *ANA* are as follows: 1) adopting different tactics in response to opponents' tactics, 2) modeling the knowledge of opponents' preferences, 3) considering the cost of delaying settlements, 4) achieving different levels of goals in negotiation, and 5) considering the different magnitudes of initial offers. The *GA*-based negotiation mechanism was used to model the dynamic concession-matching behavior arising in bilateral-negotiation situations. Representing the set of feasible offers of an agent as a population of chromosomes, the "goodness" of each chromosome (i.e., each feasible offer) was measured by a fitness function derived from Social Judgment Theory (*SJT*). With a predefined number of iterations, reproduction, crossover, and mutation were used to operate on the population of chromosomes, and the fittest chromosome from the current population was selected as tentative solution which represents the counteroffer. However, since the fitness function of *ANA* is based on *SJT*, an agent's evaluation of its opponent's counteroffer(s) may be subjective.

Reference [26] is one of the earliest works that utilize a *GA* for bolstering negotiation support systems. In [26], *GA* is used to learn effective rules for bolstering a bilateral negotiation process. Unlike the work in [20], [24], and [25] where chromosomes are used to encode strategies and offers, respectively, chromosomes in [26] represent (classification) rules. In [26], the fitness of a rule (chromosome) is determined by the frequency that it is used to contribute to a successful negotiation process (i.e., the number of times the rule is used to contribute to reaching a consensus). The basic genetic operators of reproduction, crossover, and mutation were used. Empirical results seem to indicate that genetically learned rules are effective in supporting users in several bilateral negotiation situations. The results in [26] also show that, in a bargaining process, an effective negotiation rule is one that prescribes small step concessions and introduces new issues into the negotiation process. On this account, [26] and this paper have different focuses and adopt different approaches.

Trade GA [27] is an approach that employs *GA* for finding solutions for multilateral negotiation involving multiple attributes. Trade *GA* is characterized by having a new genetic operator called *Trade* (in addition to crossover and mutation) for addressing problem-specific characteristics. *Trade* models a concession-making mechanism that is often used in negotiation systems and simulates the exchange of a resource of one negotiator for a resource of another negotiator. By applying the trade operator, negotiators and resources are randomly selected based on their willingness to trade. Empirical results seem to suggest that Trade *GA* outperformed all the other approaches such as traditional *GA*, random search, hill-climbing algorithm, and nonlinear programming.

B. Negotiation Agents Adopting BL

The work in [19] attempted to demonstrate that learning an opponent's *RP* is beneficial in bilateral negotiation. A *BL* algorithm is used during a negotiation process to update an agent's belief of its opponents *RP*. Two performance measures were used: joint utility (*JU*) and number of proposals exchanged (*NPE*s) during negotiation. Similar to this work, three scenarios were simulated in [19]: 1) when only one agent learns its opponent's *RP*, 2) when both agents learn each other's *RP*, and 3) neither agent learns. During negotiation, each learning agent makes a proposal computed from a linear combination of its own *RP* and the expected value of the opponent's *RP*. An agent that does not learn will change its proposal with a fixed percentage but above (respectively, below) its *RP* for a seller (respectively, buyer) agent. For simplicity, [19] designated a distribution function to each element (prior probability or conditional probability) in the Bayesian updating rule. Empirical results showed that agents obtain the most favorable *JU* when both agents learn [scenario 2)], less favorable *JU* when neither agent learns [scenario 3)], and the least favorable *JU* when only one agent learns [scenario 1)]. Scenario 2) has the smallest *NPE*s, scenario 1) a larger *NPE*s, and scenario 3) the largest *NPE*s. While [19] showed that adopting *BL* to estimate the *RP*s of their opponents enhances agents' negotiation speed and joint utilities, this work has shown that agents adopting *BLGAN* achieved an almost 100% success rate in negotiation and obtained higher utilities and much better combined outcomes than agents without learning capabilities. Whereas agents in [19] used *BL* to learn their opponents' *RP*s, an agent adopting *BLGAN* learns both the *RP* and deadline of its opponent.

The work in [28] attempted to construct a Bayesian Classifier from past exchanges of messages for updating an agent's beliefs of other agents' preferences in multilateral negotiation. On this account, Bui *et al.* [28] have quite different focuses from this work. In [28], all possible agreements of a negotiation are represented as nodes of an agreement tree. Negotiation is considered as a coordinated search through the agreement tree to find a leaf (agreement) that is acceptable to all agents. The distributed meeting schedule domain was chosen to test the performance of their learning agents. Two-agent and three-agent negotiations were simulated. Performance measures in their work are as follows: group utilities of the final agreements, number of messages needed for reaching agreements in each run, average prediction error, and entropy of the learned probabilities for the learning agents. Experimental results seem to suggest that their learning agents are able to either reach agreements with equal utilities but fewer number of messages exchanged or make better predictions but with less uncertainty over time.

VI. Conclusion

Based on the theoretical results obtained in Section II for finding an optimal strategy for negotiation with complete information, this work has devised a procedure called *BLGAN* to search for solutions that are close to optimal solutions in negotiation with incomplete information.

For negotiation with complete information, this work determines the *specific* strategy (i.e., the exact value of λ^A) that an agent A should adopt and proves that λ^A maximizes the utility of A and guarantees that agreements are reached (Theorems 1 and 2, Section II). This contribution distinguishes this work from [11] which only showed that there is an optimal *strategy class* [among three strategy classes: *Boulware* ($\lambda^A > 1$), *Linear* ($\lambda^A = 1$), and *Conceder* ($\lambda^A < 1$)] for different scenarios.

The main and novel contribution of this work is that *BLGAN* is one of the earliest works that use the synergy of *GA* and *BL* to deal with the difficult problem of determining an agent's optimal strategy in negotiation with incomplete information by learning the private information of its opponent. Since an opponent's *RP* and deadline (which are unknown to another agent) are used as two of the variables for computing an agent's optimal strategy (Section II), any error in learning an opponent's *RP* or deadline would mean that an agent will generate a proposal that deviates from the proposal that corresponds to its optimal strategy. To compensate for possible errors in estimating an opponent's *RP* using the *BL*-procedure and in estimating an opponent's deadline (Section III-A and B), the *GA*-procedure (Section III-D) is used to search for a possibly better proposal within a dynamic *SP* confined to an area around a proposal generated by an agent's current strategy determined using the estimated values of an opponent's deadline and *RP*. On the other hand, the *BL* procedure and the deadline-estimation process are used to allow an agent to adaptively focus its search *only* on an appropriate area in the *SP* of the *GA*-procedure. As time passes and as both agents exchange more proposals, an agent is more likely to obtain better estimations of the *RP* and deadline of its opponent. This in turn enables the *GA*-procedure in *BLGAN* to reduce the size of its *SP* by adaptively focusing its search on a specific region in the space of all possible proposals [dynamically defined by each proposal generated by an agent at each negotiation round using its revised estimations of an opponent's *RP* and deadline (see Section III-D)]. In doing so, empirical results have shown that agents adopting *BLGAN* are more likely to achieve much higher average utilities and much better *CNO*s than agents that adopt only *GA* for generating their proposals (e.g., [20]) (see the analysis of observation 5 in Section IV). This is because the *GA*-procedure in *BLGAN* focuses its search around a proposal generated by an agent's strategy that is derived from the *same* formula as the agent's optimal strategy (proven in Section II). At each negotiation round t, an agent adopting *BLGAN* attempts to adjust its strategy to the optimal strategy using its updated estimations of its opponent's *RP* and deadline and by treating its proposal at round $t-1$ as its "initial price" (Section III-C).

Whereas [19] showed that adopting *BL* to estimate the *RP*s of their opponents enhances agents' negotiation speed and joint utilities, empirical results in Section IV show that agents adopting *BLGAN* achieved higher average utilities than agents in [19]. This is because an agent in *BLGAN* compensates for possible errors in estimating an opponent's *RP* and deadline by using *GA* to search around a proposal generated by its current strategy (derived from the same formula as its optimal strategy).

For negotiation with incomplete information, empirical results in Section IV have shown that agents adopting *BLGAN* were highly successful in reaching agreements and achieved much higher average utilities and generally much better *CNO*s than agents that do not learn their opponents' *RP*s and deadlines.

Moreover, this work has considerably and significantly enhanced the authors' preliminary work in [21] as follows.

1) Whereas agents in [21] *only* learn the *RP*s of their opponents, agents in this paper are programmed to learn *both* the *RP*s and deadlines of their opponents. In particular, this work extends [21] by enhancing the *BLGAN* procedure (Fig. 1 in Section III) with a deadline-estimation process (Section III-B) and provides a more detailed analysis of the process for adjusting an agent's strategy by considering two special cases (Section III-C). The *BL*-procedure and the *GA*-procedure in Figs. 2 and 3 (Section III) have also been improved, and more detailed descriptions for both these procedures have been added in Section III-A and D, respectively.
2) Empirical results in Section IV also show that an agent adopting *BLGAN* achieved better performance when it learns both the *RP* and deadline of its opponent (in this work) than just learning its *RP* (in the preliminary work in [21]). By having estimations of *both* the *RP* and deadline of its opponent, an agent A is *more likely* to adjust λ^A to a value that is closer to its optimal strategy, since its optimal strategy is determined using *both* the *RP* and deadline of its opponent.
3) Whereas [21] only compared *BLGAN* with the following negotiation scenarios: a) negotiation with complete information when agents adopt the optimal strategy and b) negotiation with incomplete information when agents do not learn their opponents' private information,

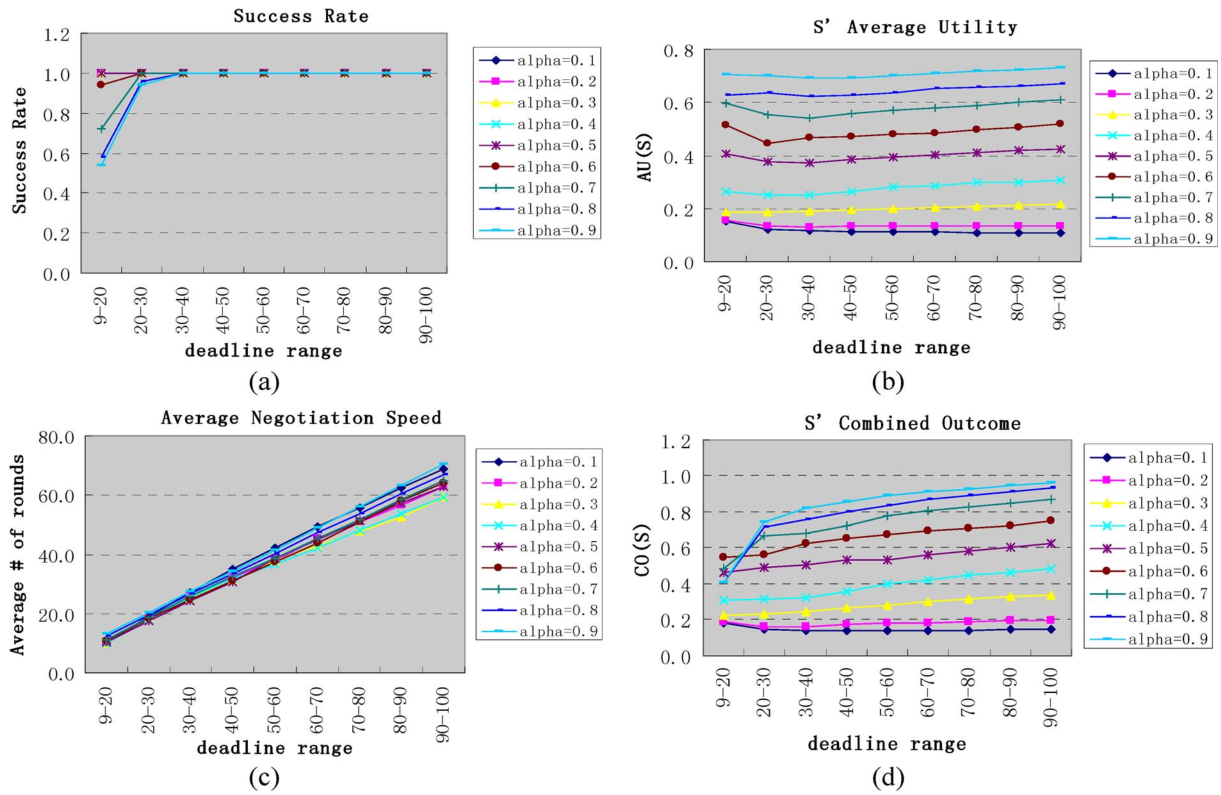

Fig. 7. *Influence* of α.

this work conducted considerably and significantly *much more* empirical studies (Section IV) than the preliminary results in [21]. The more extensive empirical results in this paper compare the relative performance of agents in *BLGAN* with that of the following: i) agents that do not learn their opponents' private information, ii) agents adopting only *GA* [20], and iii) agents adopting only *BL* [19].

4) In comparison with [21], this paper has provided much more detailed discussions of related works on negotiation agents adopting *GA* and *BL* (Section V).

While there is an enormous volume of works on game-theoretic models of bargaining (e.g., [12]–[18]), there are comparatively fewer works on applying evolutionary computation techniques (e.g., *GA*) and *BL* for finding solutions in negotiation problems. On this account, this work does not compete with the existing related literature but, rather, it supplements the very few works on applying *GA* and *BL* to solving negotiation problems by providing a novel approach that is a hybrid of *GA* and *BL*.

Nevertheless, the authors acknowledge that, in its present form, this work only considers bilateral negotiation. A future agenda of this paper is to apply *BLGAN* to enhancing the performance of adaptive bargaining agents that consider outside options [30]–[32] in multilateral negotiation.

Appendix

The results obtained from experimental tuning to determine an appropriate value of α when only S learns are shown in Fig. 7. From Fig. 7, it can be concluded that S achieved a higher *CNO* with an increase in the value of α. It is reminded that *CNO* is a function of the following: 1) R_{success}, 2) *AU*, and 3) *ANS*. It can be observed from Fig. 7 that, whereas both R_{success} and *ANS* are not significantly influenced by an increase (or decrease) in the value of α, S achieved a higher *AU* with an increase in the value of α. Furthermore, since *CNO* is directly proportional to *AU*, an increase in value in *CNO* is attributed to the increase in *AU* as α increases.

Acknowledgment

The authors would like to thank the Editor-in-Chief, Associate Editor, and the anonymous referees for their comments and suggestions.

References

[1] J. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. Cambridge, MA: MIT Press, 1994.

[2] R. Guttman, A. Moukas, and P. Maes, "Agent-mediated electronic commerce: A survey," *Knowl. Eng. Rev.*, vol. 13, no. 2, pp. 147–159, Jun. 1998.

[3] A. Chavez and P. Maes, "Kasbah: An agent marketplace for buying and selling goods," in *Proc. 1st Int. Conf. Practical Appl. Intell. Agents Multi-Agent Technol.*, London, U.K., 1996, pp. 159–178.

[4] K. Czajkowski, I. Foster, and C. Kesselman, "Agreement-based resource management," *Proc. IEEE*, vol. 93, no. 3, pp. 631–643, Mar. 2005.

[5] R. Wolski *et al.*, "Grid resource allocation and control using computational economies," in *Grid Computing—Making the Global Infrastructure a Reality*, F. Berman, A. Hey, and G. Fox, Eds. New York: Wiley, 2003, pp. 747–771.

[6] K. M. Sim, "A survey of bargaining models for grid resource allocation," *ACM SIGECOM: E-Commerce Exchange*, vol. 5, no. 5, pp. 22–32, Jan. 2006.

[7] K. M. Sim, "G-commerce, market-driven G-negotiation agents and grid resource management," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1381–1394, Dec. 2006.
[8] K. M. Sim, "Relaxed-criteria G-negotiation for grid resource co-allocation," *ACM SIGECOM: E-Commerce Exchanges*, vol. 6, no. 2, pp. 37–46, Jan. 2007.
[9] K. M. Sim and K. F. Ng, "A relaxed-criteria bargaining protocol for grid resource management," in *Proc. 6th IEEE Int. Symp. CCGRIDW*, Singapore, May 2006, p. 5.
[10] T. Sandholm and N. Vulkan, "Bargaining with deadlines," in *Proc. Nat. Conf. Artif. Intell.*, Orlando, FL, 1999, pp. 44–51.
[11] S. Fatima, M. Wooldridge, and N. Jennings, "Bargaining with incomplete information," *Ann. Math. Artif. Intell.*, vol. 44, no. 3, pp. 207–232, 2005.
[12] J. F. Nash, "The bargaining problem," *Econometrica*, vol. 18, no. 2, pp. 155–162, Apr. 1950.
[13] M. J. Osborne and A. Rubinstein, *Bargaining and Markets*. New York: Academic, 1990.
[14] A. Rubinstein, "A bargaining model with incomplete information about time preferences," *Econometrica*, vol. 53, no. 5, pp. 1151–1172, Sep. 1985.
[15] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, Jan. 1982.
[16] J. Harsayi, *Rational Behavior and Bargaining Equilibrium in Games and Social Situations*. Cambridge, U.K.: Cambridge Univ. Press, 1977.
[17] A. Muthoo, *Bargaining Theory With Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[18] J. Harsanyi and R. Selten, "A generalized Nash solution for two-person bargaining games with incomplete information," *Manage. Sci.*, vol. 18, no. 5, pp. 80–106, Jan. 1972.
[19] D. Zeng and K. Sycara, "Bayesian learning in negotiation," *Int. J. Human-Comput. Stud.*, vol. 48, no. 1, pp. 125–141, Jan. 1998.
[20] R. Lau *et al.*, "An evolutionary learning approach for adaptive negotiation agents," *Int. J. Intell. Syst.*, vol. 21, no. 1, pp. 41–72, Jan. 2006.
[21] K. M. Sim, Y. Guo, and B. Shi, "Adaptive bargaining agents that negotiate optimally and rapidly," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 1007–1014.
[22] 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.
[23] M. Wooldridge, *An Introduction to MultiAgent Systems*. Hoboken, NJ: Wiley, 2002.
[24] N. Matos, C. Sierra, and N. Jennings, "Determining successful negotiation strategies: An evolutionary approach," in *Proc. ICMAS*, 1998, pp. 182–189.
[25] R. Krovi, A. Graesser, and W. Pracht, "Agent behaviors in virtual negotiation environments," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 1, pp. 15–25, Feb. 1999.
[26] S. Matwin, T. Szapiro, and K. Haigh, "Genetic algorithms approach to a negotiation support system," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 102–114, Jan./Feb. 1991.
[27] B. Rubenstein-Montano and R. Malaga, "A weighted sum genetic algorithm to support multiple-party multiple-objective negotiations," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 366–377, Aug. 2002.
[28] H. Bui, S. Venkatesh, and D. Kieronska, "Learning other agents' preferences in multi-agent negotiation using the Bayesian classifier," *Int. J. Cooperative Inf. Syst.*, vol. 8, no. 4, pp. 275–293, 1999.
[29] S. Buffett and B. Spencer, "Learning opponents' preferences in multi-object automated negotiation," in *Proc. 7th IEEE Int. Conf. E-Commerce Technol.*, 2005, pp. 300–305.
[30] K. M. Sim, "A market-driven model for designing negotiation agents," *Comput. Intell.—Special Issue in Agent Technology for E-Commerce*, vol. 18, no. 4, pp. 618–637, Nov. 2002.
[31] K. M. Sim and C. Y. Choi, "Agents that react to changing market situations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 2, pp. 188–201, Apr. 2003.
[32] N. Jin and E. Tsang, "Co-adaptive strategies for sequential bargaining problems with discount factors and outside options," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 2149–2156.

Kwang Mong Sim (SM'07) received the B.Sc. degree (*summa cum laude*) from the University of Ottawa, Ottawa, ON, Canada, and the M.Sc. and Ph.D. degrees from the University of Calgary, Calgary, AB, Canada.

He is currently with the Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju, Korea. He has served as a Referee for several national research grant councils, including the National Science Foundation in USA, and as Program Vice Chair, Area Chair, Session Chairman, and PC Member in many conferences. He has supervised over 30 researchers/graduate students.

Dr. Sim is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C and the *International Journal of Applied Systemic Studies*. He is also an editorial/advisory board member of the *International Journal of Hybrid Intelligent Systems*, the *Open Cybernetics and Systemics Journal*, and the *System and Information Sciences Notes*. He is also the Guest Editor of five (IEEE) journal Special Issues in agent-based Grid computing and automated negotiation, including a Special Issue on Grid Resource Management in the IEEE SYSTEMS JOURNAL (IEEE Systems Council).

Yuanyuan Guo is currently working toward the Ph.D. degree in the Department of Computer Science, University New Brunswick, Saint John, NB, Canada.

Her research interests include machine learning, Bayesian learning, and data mining.

Benyun Shi received the M.Phil. degree from Hong Kong Baptist University, Kowloon Tong, Hong Kong.

He recently joined the Department of Information and Communications, Gwangju Institute of Science and Technology, Gwangju, Korea, as a Research Assistant. His research interests include automated negotiation and agent-based grid computing.