Evolving the Best-response Strategy to Decide When to Make a Proposal

Bo An, Kwang Mong Sim and Victor Lesser

Abstract— **This paper designed and developed negotiation agents with the distinguishing features of 1) conducting continuous time negotiation rather than discrete time negotiation, 2) learning the response times of trading parties using Bayesian learning and, 3) deciding when to make a proposal using a multi-objective genetic algorithm (*MOGA*) to evolve their best-response proposing time strategies for different negotiation environments and constraints. Results from a series of experiments suggest that 1) learning trading parties' response times helps agents achieve more favorable trading results, and 2) on average, when compared with SSAs (Static Strategy Agents), BRSAs (Best-Response proposing time Strategy Agents) achieved higher average utilities, higher success rates in reaching deals, and smaller average negotiation time.**

I. Introduction

Automated negotiation among software agents is becoming increasingly important and research on engineering e-negotiation agents [7], [12] has received a great deal of attention in recent years. Even though there are many extant negotiation agents dealing with negotiation with multiple trading parties (e.g., [9], [20], [15], [3]), negotiation in these systems is conducted in discrete time. This gives rise to the problem that during negotiation, no matter how long an agent has to wait and how many proposals have been received, the agent cannot propose *until* having received proposals from all its trading parties. Therefore, the general negotiation mechanism is not flexible when agents have different response times. Thus, it is essential for agents to use a flexible strategy (called *proposing time strategy*) to decide when to make a proposal in response to negotiation dynamics. Furthermore, to operate successfully in open environments, bargaining agents must be capable of evolving their proposing time strategies to adapt to prevailing circumstances and constraints.

To overcome the limitation of the general negotiation mechanism, [1] have developed a flexible negotiation mechanism for software agents and it has been shown that the flexible strategy achieved better negotiation results as compared with the general discrete time strategy. However, the flexible strategy in [1] is not designed with capabilities that can enhance their performance by evolving their proposing time strategies for negotiation in different environments and constraints. Nevertheless, it is assumed in [1] that agents have complete information about their trading parties' response times. However, in real-world negotiation, an agent's response time is often its private information. To this end, this work will supplement and complement existing literature by developing bargaining agents that can 1) learn the response times of their trading parties using Bayesian learning (Section III) and 2) evolve their best-response proposing time strategies in different market situations and constraints using a multi-objective GA (Section IV) in which negotiation results are not only evaluated in terms of utility, but also in terms of negotiation speed and success rate.

Additionally, the negotiation model is presented in Section II. We conducted a series of experiments (see Section V) to 1) determine the most successful proposing time strategies for different negotiation environments, and 2) evaluate the performance of negotiation agents capable of learning trading parties' response times and evolving best-response proposing time strategies. Section VI summarizes related work and Section VII concludes this paper.

II. Negotiation Model

For ease of analysis, this work focuses on single-issue (e.g., price-only) negotiation. Each agent consists of a manager agent and several sub-agents, and negotiation is composed of multiple interactive sub-negotiation threads. When negotiation begins, the manager agent creates a number of sub-agents corresponding to the number of trading parties, and each sub-agent negotiates with a trading party. All the sub-negotiation threads mutually influence one another. This section presents 1) the negotiation protocol, and 2) sub-agents' concession strategy.

A. Negotiation Protocol

This work adopts the alternating offers protocol (see [16, p.100]) so that a pair of buyer and seller agents negotiates by making proposals in alternate rounds. Many buyer-seller pairs can negotiate deals simultaneously. At round $t = 0$, an agent sends its proposals to its trading parties. During negotiation, each agent uses its proposing time strategy discussed in Section IV to decide when to make a proposal. If no agreement is reached, negotiation proceeds to another round. Negotiation between two agents terminates 1) when an agreement is reached or 2) with a conflict when one of the two agents' deadlines is reached.

B. Sub-agents' concession strategy

A sub-agent's concession rate at each round of negotiation is determined by four decision functions [1]: *time dependent*

Bo An is with the Computer Science Department, University of Massachusetts, Amherst, MA 01003, USA (e-mail: ban@cs.umass.edu).

Kwang Mong Sim is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (e-mail: bsim@comp.hkbu.edu.hk). Kwang Mong Sim gratefully acknowledges financial support from the Hong Kong Research Grant Council under project code: RGC/HKBU210906.

Victor Lesser is with the Computer Science Department, University of Massachusetts, Amherst, MA 01003, USA (e-mail: lesser@cs.umass.edu).

function T, *trading parties' strategies dependent function* O, *other negotiation threads dependent function* P, and *competition dependent function* C. Let $k_i^t[T]$, $k_i^t[O]$, $k_i^t[P]$ and $k_i^t[C]$ be the set of concession rates of agent i in round t according to the decision functions T, O, P and C respectively. The agent i's concession rate k_i^t in round t is given by:

$$k_i^t = f\Big(k_i^t[T], k_i^t[O], k_i^t[P], k_i^t[C]\Big)$$

If agent i is a buyer, its proposal b_i^t at round t is $b_i^t = (1+k_i^t) \times b_i^{t-1}$; If agent i is a seller, its proposal s_i^t at round t is $s_i^t = (1-k_i^t) \times s_i^{t-1}$.

Time dependent function T: Function T takes the remaining negotiation time's effect on negotiation strategies into account. When an agent negotiates with a number of trading parties, from its perspective, the negotiation process consists of several rounds, but each round may be of different cycle time (the time spent in a round of negotiation) and the agent may only propose to some of its trading parties in each round. For the sub-negotiation thread i, let B_i^t be the beginning time at round t and F_i^t be the finishing time in round t. The proposal $p_i^t[T]$ of agent i at round t is modeled as follows:

$$p_i^t[T] = IP_i - \phi_i(B_i^t) \times (IP_i - RP_i)$$

where time-dependent function $\phi_i(B_i^t)$ is determined with respect to time preference (*eagerness*) $\beta > 0$ and deadline T_{max}^i and is given as

$$\phi_i(B_i^t) = (B_i^t / T_{max}^i)^\beta$$

With infinitely many values of β, there are infinitely many possible strategies in making concession with respect to remaining time. However, they are classified by [18] into:

1) Linear: $\beta = 1$ and $\phi_i(B_i^t) = B_i^t/T_{max}^i$. $b_i^t[T]$ linearly increases and the remaining time has a consistent effect on the concession rate.

2) Conciliatory: $\phi_i(B_i^t) = (B_i^t/T_{max}^i)^\beta$, where $0 < \beta < 1$. An agent makes larger concessions in the early trading rounds and smaller concessions at the later stage.

3) Conservative: $\phi_i(B_i^t) = (B_i^t/T_{max}^i)^\beta$, where $\beta > 1$. An agent makes smaller concessions in early rounds and larger concessions in later rounds.

Let $kb_i^t[T]$ (respectively, $ks_i^t[T]$) be the concession rate of sub-buyer (respectively, sub-seller) i according to function T at round t. Let $b_i^t[T]$ (respectively, $ks_i^t[T]$) be sub-buyer i's proposal at round t according to function T. Since $b_i^t[T] = (1 + kb_i^t[T]) \times b_i^{t-1}[T]$ and $s_i^t[T] = (1 - ks_i^t[T]) \times s_i^{t-1}[T]$,

$$kb_i^t[T] = \frac{(RP_i - IP_i) \times (B_i^t/T_{max}^i)^\beta + IP_i}{(RP_i - IP_i) \times (B_i^{t-1}/T_{max}^i)^\beta + IP_i} - 1$$

$$ks_i^t[T] = 1 - \frac{IP_i - (IP_i - RP_i) \times (B_i^t/T_{max}^i)^\beta}{IP_i - (IP_i - RP_i) \times (B_i^{t-1}/T_{max}^i)^\beta}$$

Trading parties' strategies dependent function O: It is reasonable for an agent to make concessions based on the behaviors of its trading parties as 1) to maximize utility, an agent may choose to use imitative tactics to protect itself from being exploited by other agents [5]; 2) making large concessions to a conservative agent makes no sense; and 3) an agent may have more than one chance to reach an agreement.

Let $kb_i^t[O]$ (respectively, $ks_i^t[O]$) be the concession rate of sub-buyer (respectively, sub-seller) i according to function O at round t ($t > 2$).

$$kb_i^t[O] = \eta^n \times (1 - s_i^{t-1}/s_i^{t-2})$$

$$ks_i^t[O] = \eta^n \times (b_i^{t-1}/b_i^{t-2} - 1)$$

where $0 < \eta \leq 1$ and n is the number of the trading parties. An agent partly, in percentage of η^n, reproduces the behavior that its trading party performed. The parameter η reflects agents' *optimism* toward negotiation results when their trading parties increase. η with a little value (respectively, larger value) means that an agent is very optimistic (respectively, pessimistic) about negotiation results with the increase of trading parties.

Other negotiation threads dependent function P: Function P models how negotiation situations of multiple threads influence each other. Let $nums_{t-1}$ be the number of trading parties to whom the buyer proposes at round t, and $s_{min}^{t-1} = min_{i=1}^{nums_{t-1}} s_i^{t-1}$ represents the lowest proposal the buyer has received in round $t-1$. Similarly, when a seller negotiates with n buyers, $0 < numb_{t-1} \leq n$ represents the number of the proposals the seller has received at round $t-1$, and $b_{max}^{t-1} = max_{i=1}^{numb_{t-1}} b_i^{t-1}$ represents the highest proposal the seller has received at round $t-1$.

Suppose that a buyer negotiates with a number of sellers. At round $t-1$, sub-buyer i receives a proposal s_i^{t-1} from its party – seller i. For sub-buyer i, 1) if $s_{min}^{t-1} > b_i^{t-1}$, it should make some concession. In this case, if s_{min}^{t-1}/s_i^{t-1} is very small, i.e., seller i's offer s_i^{t-1} is very high as compared with the other $nums_{t-1} - 1$ proposals, sub-buyer i would make a small concession as their negotiation seems "hopeless". If s_{min}^{t-1}/s_i^{t-1} is closed to 1, i.e., seller i's proposal s_i^{t-1} is very "favorable", sub-buyer i would make a large concession to seller i according to function O as the sub-negotiation thread seems "hopeful". Therefore, the lower seller i's proposal as compared with the other $nums_{t-1} - 1$ proposals, the larger concession sub-buyer i will make. 2) Otherwise, i.e., $s_{min}^{t-1} \leq b_i^{t-1}$. Since at least one of the trading parties' proposals is higher than its proposal at round $t-1$, sub-buyer i will choose to raise its expectation, i.e., let $b_i^t < s_{min}^{t-1} \leq b_i^{t-1}$. Since $b_i^t = (1 + kb_i^t[P]) \times b_i^{t-1}$, where $kb_i^t[P]$ is the concession rate of sub-buyer i according to function P at round t, then $(1 + kb_i^t[P]) \times b_i^{t-1} < s_{min}^{t-1}$, thus $kb_i^t[P] < s_{min}^{t-1}/b_i^{t-1} - 1$. Hence, we have: $kb_i^t[P] = s_{min}^{t-1}/b_i^{t-1} - 1 - \sigma$, $\sigma > 0$, and sub-buyer i can decide the value of σ according to its desire to maximize utility (*greed*). σ with a large value means that the agent will greatly raise its expectation if $s_{min}^{t-1} \leq b_i^{t-1}$.

Similarly, let $ks_i^t[P]$ be the concession rate of sub-seller i according to function P at round t. Hence,

$$kb_i^t[P] = \begin{cases} min(kb_i^t[O], s_{min}^{t-1}/s_i^{t-1}) & \text{if } s_{min}^{t-1} > b_i^{t-1} \\ s_{min}^{t-1}/b_i^{t-1} - 1 - \sigma & \text{otherwise} \end{cases}$$

$$ks_i^t[P] = \begin{cases} min(ks_i^t[O], b_i^{t-1}/b_{max}^{t-1}) & \text{if } b_{max}^{t-1} < s_i^{t-1} \\ 1 - b_{max}^{t-1}/s_i^{t-1} - \sigma & \text{otherwise} \end{cases}$$

Competition dependent function C: A negotiator's bargaining "power" is affected by the number of competitors, and trading alternatives. The competition situation of an agent is determined by the probability that it is being (not being) considered as the most preferred trading party [19], [17]. Suppose agent i has $m-1$ competitors and n trading parties. The probability that agent i is not the most preferred trading party of any trading party is $(m-1)/m$. The probability of agent i not being the most preferred trading party of all the trading parties is $[(m-1)/m]^n$. Hence,

$$k_i^t[C] = [(m-1)/m]^n$$

where $k_i^t[C]$ is the concession rate of sub-agent i according to the decision function C at round t. Since buyers and sellers can enter and leave the market at any time, the values m and n may constantly change with ongoing negotiation.

III. Decide when to make a proposal

A. Learn trading parties' response times

Rather than assuming agents have complete information about trading parties' response times, this work utilizes Bayesian learning to learn agents' response times.

In classical statistics, Bayesian theorem of continuous random variable has the following form:

$$\pi(\theta|x) = \frac{p(x|\theta)\pi(\theta)}{\int_\theta p(x|\theta)\pi(\theta)d\theta}$$

where $\pi(\theta)$ is the prior distribution density function. $\pi(x|\theta)$ is the conditional density of X when the random variable θ is given. $\pi(\theta|x)$ is the conditional distribution density function of θ when samples $X = (X_1, X_2, \ldots, X_n)$ are given, i.e., posterior distribution of θ.

In a dynamic e-commerce market, there are many reasons resulting in different trading parties having different response times. For ease of analysis, this paper assumes that each agent's response time follows a normal distribution.[1] Assume a buyer agent is trying to learn the response time of a seller agent. Assume the real response time of the seller follows a normal distribution $N(\theta, \sigma^2)$, where σ^2 is known and θ is unknown. That is, the buyer intends to learn the value of θ. Through interaction during negotiation, the buyer receives a series of response times $X_1, X_2, \ldots, X_n$ (samples of normal distribution $N(\theta, \sigma^2)$) from the seller agent. Let another normal distribution $N(\mu, \tau^2)$ be the buyer's prior belief of the distribution of θ. Assume that ${\sigma_0}^2 = \sigma^2/n$, $\overline{x} = \sum_{i=1}^n x_i/n$. The posterior distribution of θ, i.e., $\pi(\theta|x)$, calculated using Bayesian theorem is $N(\mu', {\sigma'}^2)$, where

[1]Bayesian learning can also be applied to learning parameters of other distributions [2].

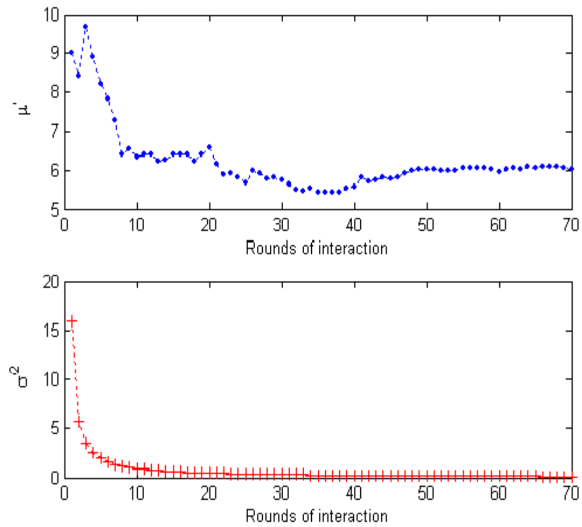

Fig. 1. Learning agents' response times.

$$\mu' = \frac{(\overline{x}{\sigma_0}^{-2} + \mu\tau^{-2})}{({\sigma_0}^{-2} + \tau^{-2})}$$

$${\sigma'}^2 = ({\sigma_0}^{-2} + \tau^{-2})^{-1}$$

${\sigma_0}^2$ is the variance of the mean of samples $\overline{x}$, and ${\sigma_0}^{-2}$ is the precision of $\overline{x}$. τ^2 is the variance of the prior distribution $N(\mu, \tau^2)$, and τ^{-2} is the precision of μ. Thus, the posterior mean μ' averages the prior mean μ and the mean $\overline{x}$ of the samples weighted according to their precision. The smaller τ^2 is, the larger the proportion of the prior mean to the posterior mean is. On the other hand, the larger the number n of samples is, the smaller ${\sigma_0}^2/n$ is, and the larger the proportion of $\overline{x}$ to the posterior mean is. During learning, the impact of samples becomes increasingly important.

Example 1: Assume a buyer b intends to learn the response time $N(\theta, \sigma^2)$ of a seller s. Let s's response time follow $N(\theta, \sigma^2) = N(6, 3^2)$ and b's prior belief of s's response time be $N(\mu, \tau^2) = N(9, 4^2)$. The learned posterior distribution of seller s's response time $N(\mu', {\sigma'}^2)$ is shown in Fig. 1. It can be observed that 1) with ongoing interaction, buyer b's updated belief of s's response time is becoming closer to the real value. In Fig. 1, the value of μ is close to 6 after 40 rounds of learning; and 2) with ongoing learning, b becomes more sure of its learned value. For example, in Fig. 1, the value of τ^2 decreases with ongoing learning.

B. Proposing time strategy

This section presents agents' proposing time strategies based on synchronization situations of negotiation, which are determined by trading parties' response times. A negotiation's synchronization situation S is given by:

$$S = \frac{\sqrt{D(C)}}{E(C)}$$

where C is the set of response times of all the trading parties, $C = (C_1, C_2, \ldots, C_n)$, where C_i is the learned response time of trading party i (i.e., μ' in Section 3.1).[2] $E(C)$ is the expectation of C, and $D(C)$ is the variance of C. S increases with the increase of the variance $D(C)$.

While making the decision of when to make a proposal at round $t+1$, an agent may confront the following dilemma: 1) Not propose until having received most or all counter-proposals from its trading parties. The agent will get more information after receiving more counter-proposals, but it will take more time to complete a round of negotiation. Consequently, the agent may be not able to complete several rounds of negotiation before the deadline approaches. 2) Propose as soon as possible. Although this approach reduces the time spent in a round of negotiation, the proposal for a trading party may not be as good as the proposal generated after receiving more counter-proposals. As the synchronization situation of a negotiation shows the difference of the set of response times of trading parties, it is intuitive to take the synchronization situations into account while making the decision of when to make a proposal.

The following strategy is used to decide when to make a proposal: at each round, after a sub-agent first receives a proposal from its trading party, it waits until the number of the sub-agents having received proposals reaches a ratio R_{wait}, and then all the sub-agents having received proposals propose to their trading parties respectively. The waiting ratio R_{wait} is given by:

$$R_{wait} = \min\Big(\lambda \frac{S_{\max} - S}{S_{\max}}, 1\Big)$$

where $\lambda > 0$, $S_{\max} = \max(C)/E(C)$ and $\max(C)$ is the maximum response time in C. Since

$$\frac{d\, R_{wait}(S)}{d\, S} = -\lambda \frac{1}{S_{\max}}$$

and $S_{\max}$ is nonnegative, the slope $d\, R_{wait}(S)/d\, S$ is always negative. Therefore, an agent will wait for a longer time with the increase of the synchronization level S.

When $\lambda = 1$, the proposing time strategy here is equivalent to the static proposing time strategy in [1] as $R_{wait} = \min(1(S_{\max} - S)/S_{\max}, 1) = (S_{\max} - S)/S_{\max}$. Thus, the proposing time strategy here can represent more proposing time strategies than the static strategy. In response to different negotiation environments, an agent should adopt different proposing time strategies. By introducing parameter λ, a negotiation agent can use different proposing time strategies in different negotiation environments. This work tries to find an approximate value of λ that can let an agent make a good agreement in different negotiation environments. Finding a solution to this optimization problem is difficult as there exist different criteria for negotiation results and there are infinitely many proposing time strategies for an agent. A possible approach for addressing this problem is to use a multi-objective GA to search the population of strategies for a best-response proposing time strategy.

[2] The value of τ is ignored here as 1) μ' is more important than the standard variance τ, and 2) with learning's ongoing, the value of τ is close to 0 (as in Fig. 1, τ decreases quickly at the beginning of learning and is close to 0 after 10 rounds of learning).

IV. Multi-objective Genetic Algorithm

This section introduces a multi-objective genetic algorithm (*MOGA*) to evolve the best-response proposing time strategies for different negotiation situations. A sequence of ever improving populations is generated as the outcome of a search method modeled using three operations: selection, crossover and mutation. The individuals of a population are negotiation agents, and their genetic materials are the parameters of the proposing time strategies. Both the buyer and seller populations are simultaneously evolving and the fitness of an individual in one population is based on direct competition with individuals from the other population. Each individual of a population is represented as a fixed-length string, and more specifically, the bits of a string (a gene) have the following structure: 1) λ, which determines the agent's proposing time strategy, 2) T_{max} (deadline), 3) IP (initial proposal), and 4) RP (reserve proposal).

Similar to basic evolutionary ideas, fitness determines an agent's chance of surviving to the next population generation. The fitness value of a gene (corresponding to a proposing time strategy) is determined by the negotiation results as the gene negotiates with other genes. Since the score of a proposing time strategy is determined by three objectives: utility (U), success rate (S), and negotiation speed (T), for this multi-objective problem, every negotiation result has three fitness values. For example, for a negotiation result $X = (X_U, X_S, X_T)$ where X_U, X_S, and X_T are the negotiation results of all the three objectives of X respectively, the function $f_I(X_I)$ is the performance measure I of the result X_I, in which $I \in \{U, S, T\}$. A performance measure I of the result X_I, i.e., $f_I(X_I)$, is defined as:

$$f_U(X_I) = \frac{X_I - min(X_I)}{max(X_I) - min(X_I)}$$

where $max(X_I)$ and $min(X_I)$ are, respectively, the best and worst value of the performance measure I.

Hence, the fitness value $f(X)$ for the multi-objective optimization problem in this work can be expressed as follows:

$$max_X\ f(X) = f\Big(f_U(X_U), f_S(X_S), f_T(X_T)\Big)$$

In defining $f(X)$, two issues were considered: 1) agents' constraints on multiple objectives. For an objective $I \in \{U, S, T\}$, each constraint can be described as $c_I \in [0, 1]$, which acts as a threshold for calculating the fitness value of a negotiation result. A larger value of c_I indicates that an agent has rigorous requirements on objective I; and 2) agents' preferences for placing the importance of the three objectives, which is represented as the priorities. The priority of an objective $I \in \{U, S, T\}$ is represented as $w_I \in [0, 1]$. Considering agents' constraints and preferences for the three objectives, $f(X)$ is given as follows:

$$f(X) = \frac{\sum_{I \in \{U,S,T\}} \Big(w_I \diamond \mu_I \Big(f_I(X_I) \Big) \Big)}{3}$$

where $\mu_I\Big(f_I(X_I)\Big)$, $I \in \{U, S, T\}$, is defined as:

$$\mu_I\Big(f_I(X_I)\Big) = \begin{cases} f_I(X_I) & \text{if } f_I(X_I) > c_I \\ 0 & \text{otherwise} \end{cases}$$

and operator $\diamond : [0,1] \times [0,1] \rightarrow [0,1]$, called a *priority operator*, satisfies [4, p.50]: 1) $\forall a_1, a_2, a_2' \in [0,1], a_2 \leq a_2' \implies a_1 \diamond a_2 \leq a_1 \diamond a_2'$, 2) $\forall a_1, a_1', a_2 \in [0,1], a_1 \leq a_1' \implies a_1 \diamond a_2 \geq a_1' \diamond a_2$, 3) $\forall a \in [0,1], 1 \diamond a = a$, and 4) $\forall a \in [0,1], 0 \diamond a = 0$.

The operator $\diamond : [0,1] \times [0,1] \rightarrow [0,1]$ is given by:

$$a_1 \diamond a_2 = (a_2 - 1) \times a_1 + 1$$

Example: Let the priority set be $w_U = 0.9$, $w_S = 0.3$ and $w_t = 0.3$. Assume that $\mu_U(f_U(X_U)) = 0.8$, $\mu_S(f_S(X_S)) = 0.4$ and $\mu_T(f_T(X_T)) = 0.7$. The fitness value $f(X)$ is given by: $f(X) = \big((0.9 \diamond 0.8) + (0.3 \diamond 0.4) + (0.3 \diamond 0.7)\big)/3 = \big((0.8-1) \times 0.9 + 1 + (0.4-1) \times 0.3 + 1 + (0.7-1) \times 0.3 + 1)\big)/3 = 2.56$.

The value of the constraint $c_I \in [0,1]$ of objective I acts as a threshold for calculating the fitness value of a negotiation outcome. A larger value of c_I indicates that an agent has rigorous requirements on objective I. For example, a combination $C = \big\langle c_U = 0, c_S = 0, c_T = 0 \big\rangle$ models the situation that an agent has no constraint on the multiple objectives. In contrast, a combination $C = \big\langle c_U = 0.8, c_S = 0.8, c_T = 0.8 \big\rangle$ indicates that an agent has rigorous requirements on the three objectives.

As agent designers may have different preferences for different objectives, it seems intuitive to place "weights" on the individual fitness values taking into account the constraints of the multiple objectives using the priority operator $\diamond$, and aggregating individual fitness using the weights to produce a single fitness value for every negotiation result. The priority operator $\diamond$ is used to express agents' different preferences on the three objectives. For instance, if an agent has sufficient negotiation time and higher chance to reach an agreement, then it may place more emphasis on optimizing its utility, and less emphasis on negotiation speed and success rate. In contrast, when an agent has shorter deadline for negotiation and lower chance to reach an agreement, it may place more emphasis on negotiation time and success rate.

The whole flow of the *MOGA* is explained as follows:

1) **[Initialization]**: In this step, the initial population P_0 is created by randomly generating n genes representing (both buyer and seller) agents with different values of λ.

2) **[Fitness calculation]**: When a new population is generated, each gene first negotiates with other genes, and the negotiation result obtained by each gene is used to determine its fitness using $f(X)$.

3) **[Select the gene with the highest fitness]**: The gene with the highest fitness $Best_{P_t}$ in the current population P_t is selected and included as part of the new population P_{t+1} in round $t+1$. Selection, crossover and mutation are applied to the other $n-1$ individuals (let them be $P_{t+1}^{'}$) in P_t.

4) **[Tournament selection]**: Tournament selection is used in the *MOGA* for selecting individuals from P_t for inclusion in the mating pool MP. Through tournament selection, k individuals are randomly selected from P_t. The individual with the highest fitness among the selected k individuals is placed in the MP, in which k is called the tournament size (in this work, k is 2). This process is repeated $n-1$ times and generates $n-1$ new genes (which forms $P_{t+1}^{'}$) for applying the crossover and mutation operations.

5) **[Crossover]**: Two individuals from $P_{t+1}^{'}$ are randomly selected. The crossover operation is only performed on the bits representing the value of λ. Crossover points are randomly selected and sorted in ascending order. The genes between successive crossover points are alternately exchanged between the individuals, with a probability Pc. Through experimental tuning, a value of $Pc = 0.6$ is adopted.

6) **[Mutation]**: Mutation is also only performed on the bits representing the value of λ. Some of the genes in $P_{t+1}^{'}$ are randomly selected for mutation. For the selected genes, a random value chosen from the domain of the gene is used to replace the value of selected bits representing the value of λ. The remaining genes have a probability $Pm(t)$ of undergoing mutation where $Pm(t) = 0.01$ is adopted.

After applying selection, crossover and mutation, the population P_{t+1} in round $t+1$ is composed of $Best_{P_t}$ in round t together with $n-1$ newly created individuals $P_{t+1}^{'}$.

7) **[Terminating condition]**: Steps 2)$\sim$6) are repeated until the following stopping criterion is met: 95% of the individuals have the same fitness or the number of iterations reaches a predetermined maximum (e.g., 100).

The complexity of each round of learning is polynomial with the number of possible strategies. The termination condition specifies the maximum number of learning rounds. Therefore, the complexity of the *MOGA* is still polynomial and thus the *MOGA* is appropriate to be used in real negotiation support systems.

V. Empirical Results

A series of experiments was carried out to compare the performance of *SSAs* and *BRSAs*.In the experiments, agents of both types were subjected to different market densities, different market types, and other factors (e.g., deadline, eagerness, optimism, greed) (Table I). All the six input parameters in Table I are generated randomly following a uniform distribution. Both market density and market type depend on the probability of generating an agent in each round and the probability of the agent being a buyer (or a seller). From a buyer agent's perspective, for a favorable (respectively, an unfavorable) market, an agent enters a market with lower (respectively, higher) probability of being a buyer agent and higher (respectively, lower) probability of being a seller. The lifespan of an agent, i.e., its deadline, is randomly selected from $[150s, 600s]$. The range of $[150s, 600s]$ for deadline is adopted based on experimental tuning and agents' behaviors. In the current experimental

TABLE I

INPUT DATA SOURCES

Input Data	**Possible Values**		
Market Type	*Favorable*	*Balanced*	*unfavorable*
P_{party}	< 0.5	0.5	> 0.5
P_{party}: *Probability* of an agent *being a trading party*			
Competitor/party	{1:2, 1:4, 1:10}	1:1	{10:1, 4:1, 2:1}
Market Density	*Sparse*	*Moderate*	*Dense*
P_{gen}	0.25	0.5	1
P_{gen}: *Probability* of *generating an agent* per round			
Deadline	*Short*	*Moderate*	*Long*
T_{max}	$150 - 250s$	$250 - 400s$	$500 - 600s$
Eagerness	*Lower range*	*Mid-range*	*Upper range*
β	5	1	0.2
Optimism	*Lower range*	*Mid-range*	*Upper range*
η	0.98	0.93	0.8
Greed	*Lower range*	*Mid-range*	*Upper range*
σ	0.02	0.05	0.1

TABLE II

LEARNED BEST-RESPONSE STRATEGIES (λ)

Market Density	Market Type	Deadline		
		Long	Moderate	Short
Sparse	Favorable	1.41	0.83	0.31
	Balanced	3.12	0.97	0.46
	Unfavorable	6.18	1.73	0.55
Moderate	Favorable	1.11	0.71	0.23
	Balanced	2.16	0.79	0.31
	Unfavorable	4.38	1.34	0.46
Dense	Favorable	0.64	0.43	0.08
	Balanced	1.41	0.67	0.19
	Unfavorable	2.54	0.81	0.34

setting, the response time of every agent is about 5s following a uniform distribution for convenience and it was found that: 1) for very short deadline ($< 150s$), very few agents could complete deals, and 2) for deadlines $> 600s$, there was little or no difference in performance among agents. Hence, for the purpose of experimentation, a deadline between the range of $150 - 250s$ (respectively, $250 - 400s$ and $450 - 600s$) is considered as short (respectively, moderate and long). The value of eagerness is randomly generated from $[0.1, 10]$ and the range values for eagerness follow the definition in Section II. It was found that when $\beta > 10$ (respectively, $\beta < 0.1$), agents made little (respectively, large) concession at the beginning of negotiation and there was little or no difference in performance among agents. Hence, representative values of eagerness from 1) the lower range (e.g., 5), 2) the upper range (e.g., 0.2), and 3) mid-range (e.g., 1) were used. The value of optimism is randomly generated from $[0.7, 1]$. Through experimentation, it was found that when the optimism value $\eta > 0.99$ or $\eta < 0.7$, there was little or no difference in performance among negotiation agents. Thus, an optimism value of 0.8 (respectively, 0.98 and 0.93) is considered as upper range (respectively, lower range and mid-range). The value of greed is selected from the range $[0.01, 0.15]$ based on experimental tuning. It was found that 1) when the greed value $\sigma > 0.15$, most agents failed to reach agreements, and 2)when the greed value $\sigma < 0.01$, there was little or no difference in performance among agents. Therefore, a greed value of 0.1 (respectively, 0.02 and 0.05) is considered as upper range (respectively, lower range and mid-range). The search space of λ is $[0.001, 10]$ because through experimental tuning, it was found that when $\lambda > 10$ (respectively, $\lambda < 0.001$), agents wait for most (respectively, few) counter-proposals during negotiation and there was almost no difference in performance among agents.

Performance measure: 1) average utility: $U_{average}$ is the average utility of agents that reached consensus, 2) success rate:$R_{success}$=$N_{success}/N_{total}$, where $N_{success}$ is the number of agents that reached consensus and N_{total} is the total number of agents, and 3) average negotiation time: R_{time}= $(\sum_{i=1}^{N_{total}} T_{end}^i / T_{max}^i)/N_{total}$, where T_{end}^i is the time spent in negotiation by agent i and T_{max}^i is its deadline.

Due to space limitation, only some empirical results are presented in this section. For the empirical results here, the market is of moderate density, the eagerness value is 0.8, the optimism value is 0.8, the greed value is 0.1, the combination of constraints is C=$\langle c_U = 0, c_S = 0, c_T = 0\rangle$, and the priority combination is $W = \langle w_U = 0.9, w_S = 0.3, w_T = 0.3\rangle$ (i.e., an agent places very high emphasis on optimizing its utility, but hopes to complete negotiation in a reasonably short time and maintains a certain level of success rate).

Evolving best-response strategies: Table II shows the best-response values of λ and it can be observed that:

1) Given the same market density and market type, but with shorter deadlines, agents wait for less counter-proposals by adopting a smaller value of λ. With shorter deadlines, an agent faces higher risk of not reaching an agreement if it spends too much time in waiting for proposals. Failing to reach an agreement will also lower the agent's success rate. Consequently, agents react to more stringent time constraints by spending less time in waiting.

2) Given the same market density and deadline, but in a favorable market, an agent is inclined to wait for less counter-proposals by adopting a smaller value of λ. In a favorable market, an agent has possibly higher chance of reaching an agreement. Thus, agents speed up negotiation and seek for the best proposal from its trading parties.

3) Given the same market type and deadline, but with higher market density, an agent is inclined to wait for less counter-proposals. With higher market densities, agents have higher chance of reaching an agreement. An agent may speed up negotiation by adopting a smaller value of λ.

It is noted that each value of λ in Table II has two decimal places, and hence, the maximum error of representing the values of λ is lower than 0.01. Similarly, other values of λ can also be obtained for other different negotiation situations and fitness functions using the proposed *MOGA*. In the series of experiments, *BRSAs* were programmed to adopt the values of λ in Table II, and *SSAs* adopt the value of $\lambda = 1$ as it is shown in Section III-B that when $\lambda = 1$, *BRSAs* are

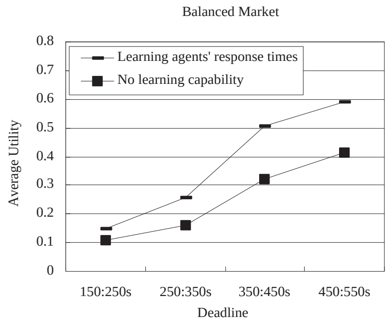

Fig. 2. Average utility and deadline.

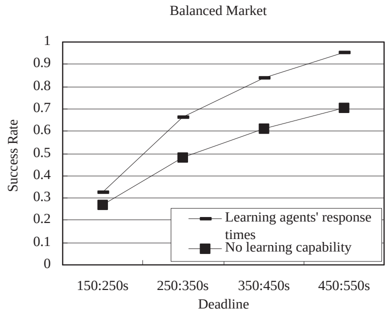

Fig. 3. Success rate and deadline.

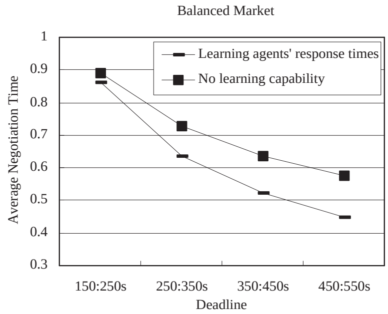

Fig. 4. Negotiation time and deadline.

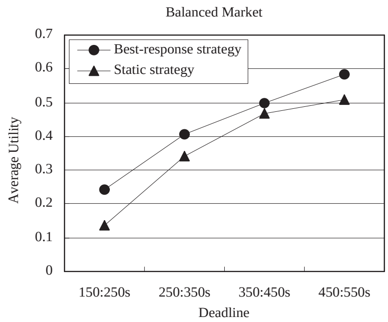

Fig. 5. Average utility and deadline.

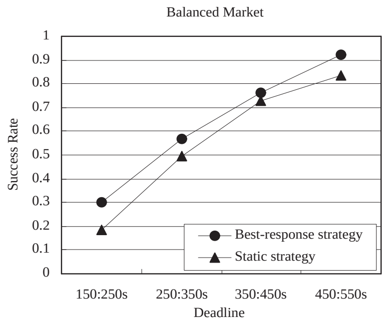

Fig. 6. Success rate and deadline.

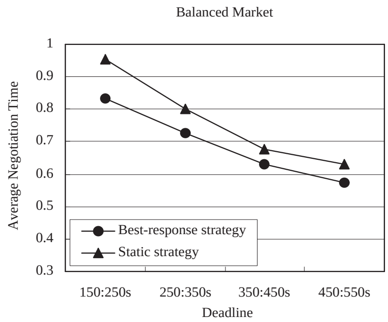

Fig. 7. Negotiation time and deadline.

equivalent to *SSAs*.

Observation 1: When subjected to different deadlines, *BRSAs* equipped with the capability of learning trading parties' response times achieved higher $U_{average}$, higher $R_{success}$ and lower R_{time} than *BRSAs* without the learning capability. For example, in Fig. 2, when the deadline is between 250s and 350s, the average utilities are 0.26 for *BRSAs* with learning mechanism, and 0.16 for agents without learning capability, respectively.

Additionally, it can also be observed from Figs. 2, 3, and 4 that, when *BRSAs* have shorter deadlines, the difference between utilities of *BRSAs* with learning capability and those without learning capability tapers. It corresponds to the intuition that when negotiation becomes tougher, the potential of improving agents' negotiation performance decreases.

Observation 2: When subjected to different deadlines, *BRSAs* achieved higher $U_{average}$, higher $R_{success}$ and lower R_{time} than *SSAs*. For example, in Fig. 5, when the deadline is between 250s and 350s, the average utilities are 0.41 for *BRSAs*, and 0.34 for *SSAs*, respectively.

Additionally, let λ^* be the value of λ adopted by *BRSAs* and $\varphi = |\lambda^* - 1|$ be the distance from 1 (the value of λ adopted by *SSAs*) to λ^*. It can be observed from Figs. 5-7 that the difference of utility, success rate, and negotiation time between *BRSAs* and *SSAs* is mainly determined by the distance φ from 1 to λ^*: the larger the value of φ, *BRSAs* outperform *SSAs* more. For example, in Fig. 5, with the short deadline 150s-250s, $\varphi = |0.31-1| = 0.69$, and the difference in the average utility between *BRSAs* and *SSAs* is $0.25 - 0.14 = 0.11$; with long deadline 350s-450s, $\varphi = |0.79-1| = 0.21$, and the difference in the average utility between *BRSAs* and *SSAs* is $0.50 - 0.48 = 0.02$.

VI. RELATED WORK

The literature of automated negotiation and negotiation agents (e.g., [5], [10]) forms a very large collection and space limitations preclude introducing all of them here. For a survey on negotiation agents for e-commerce, see [7], and [12], respectively.

In terms of the number of agents participating in negotiation, agent based automated negotiation can be divided into three cases [8]: *one-to-one negotiation* (*bilateral negotiation*), *many-to-many negotiation* and *one-to-many negotiation*. Compared with auction mechanisms [14], one-to-many interactive negotiation is more flexible. For example, agents can adopt different negotiation strategies with different trading parties (alternatives), negotiation can be taken under different negotiation environments and protocols [15]. There are some negotiation systems and negotiation strategies supporting one-to-many negotiation (see [1] for a more detailed review). ITA (Intelligent Trading Agency)[15], [9] is a framework for one-to-many negotiation by means of conducting a number of concurrent coordinated one-to-one negotiation. In the framework, a number of agents, all working on behalf of one party, negotiate individually with other parties. After each negotiation cycle, these agents report back to a coordinating agent that evaluates how well each agent has done, and issues new instructions accordingly. Each individual agent conducts reasoning by using constraint based techniques. Nguyen and Jennings [13] have developed and evaluated a heuristic model that enables an agent to participate in multiple, concurrent bilateral encounters in competitive situations in which there is information uncertainty and deadlines. Li et al. [11] present a model for bilateral contract negotiation that considers the uncertain and dynamic outside options. Outside options affect the negotiation strategies via their impact on the reserve price. The model is composed of three modules: single-threaded negotiation, synchronized multi-threaded negotiation, and dynamic multi-threaded negotiation.

To overcome the limitations of the discrete time negotiation, a continuous time one-to-many negotiation mechanism was proposed in [1] in which an agent can decide when to propose according to synchronization situations of negotiation. Two strategies (fixed waiting time based strategy and fixed waiting ratio based strategy) for the decision making of when to make a proposal were proposed. The simulation results show that the waiting time (or waiting ratio) given by the two strategies leads to the most favorable outcomes. Furthermore, the flexible mechanism can be easily transformed to the discrete time negotiation mechanism. Although the continuous time negotiation mechanism brings much flexility to negotiation agents which concurrently negotiate with a set of trading parties, negotiation agents have no ability to adaptive to different negotiation environments and constraints. To meet this end, this paper proposes a multi-objective genetic algorithm to evolve the best-response proposing time strategy for different negotiation situations as it was noted in [6] that GA is an alternative to the standard game-theoretic models for generating optimal solutions in a bargaining problem, particularly in practical situations involving a large agent populations. In addition, Bayesian learning mechanism is used to learn trading parties' response times.

VII. Conclusion

The main contributions of this research are twofold: Firstly, a multi-objective GA for evolving the best-response proposing time strategies for different market situations was designed and implemented. Agents are designed with the flexibility of take different proposing time strategies for different environments. Experimental results show that *BRSAs* generally achieved higher $U_{average}$, higher $R_{success}$ and lower R_{time} than *SSAs* in many situations. Furthermore, in the fitness function of the *MOGA* in this work, both the constraints and preferences of negotiation agents on the three objectives to be optimized are modeled using a priority operation, and by placing weights on the constraints. Secondly, by adopting Bayesian learning, negotiation agents are enhanced to be capable of learning the response times of trading parties. Experimental results show that negotiation agents with learning ability outperform agents without learning capability.

This work provides a more plausible representation of negotiators in real-life and are designed with more flexibility in negotiation. Whereas the *MOGA* in its present form is only used to evolve the best-response proposing time strategies, perhaps, it may also be used to evolve sub-negotiators' concession strategies represented by eagerness (β), optimism (η), and greed (σ). Another possible future direction for this work is designing multi-objective GAs that can converge faster by pruning the initial search space in some situations. In addition, in its present form, this work assumes that both *SSAs* and *BRSAs* can randomly enter or leave an e-market, the strategies learned may not necessarily converge and negotiation agents may be trapped in a local minimum. Analyzing the convergence of learning algorithms in dynamic environments is a complex problem, and it is among the list of agendas for future enhancements of this work.

References

[1] B. An, K. M. Sim, L. Tang, S. Li, and D. Cheng, "Continuous time negotiation mechanism for software agents," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 36, no. 6, pp. 1261–1272, Dec. 2006.

[2] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*. New York: John Wiley, 1994.

[3] F. M. T. Brazier, F. Cornelissen, R. Gustavsson, C. M. Jonker, O. Lindeberg, B. Polak, and J. Treur, "A multi-agent system performing one-to-many negotiation for load balancing of electricity use," *Electronic Commerce Research and Applications*, vol. 1, no. 2, pp. 208–224, 2002.

[4] D. Dubois and H. Prade, "Qualitative possibility theory and its applications to constraint satisfaction and decision under uncertainty," *International Journal of Intelligent Systems*, vol. 14, no. 1, pp. 45–61, 1999.

[5] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Int. Journal of Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 159–182, 1998.

[6] S. Fatima, M. Wooldridge, and N. R. Jennings, "A comparative study of game theoretic and evolutionary models for software agents," *Artificial Intelligence Review*, vol. 23, no. 2, pp. 187–205, 2005.

[7] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge, "Automated negotiation: prospects, methods and challenges," *Int. Journal of Group Decision and Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.

[8] J. B. Kim and A. Segev, "A framework for dynamic ebusiness negotiation processes," in *Proc. of IEEE Int. Conf. on E-Commerce*, Jan. 2003, pp. 24–27.

[9] R. Kowalczyk and V. Bui, "On constraint-based reasoning in e-negotiation agents," *Agent-Mediated E-Commerce III, Lecture Notes in Computer Science*, vol. 2003, pp. 31–46, 2003.

[10] S. Kraus, "Automated negotiation and decision making in multi-agent environments," *Multi-Agents systems and applications*, no. 2, 2001.

[11] C. Li, J. A. Giampapa, and K. Sycara, "Bilateral negotiation decisions with uncertain dynamic outside options," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Special issue on Game-theoretic Analysis and Stochastic Simulation of Negotiation Agents*, vol. 36, no. 1, Jan. 2006.

[12] A. R. Lomuscio, M. Wooldridge, and N. R. Jennings, "A classification scheme for negotiation in electronic commerce," *Int. Journal Group Decision Negotiation*, vol. 12, no. 1, pp. 31–56, Jul./Aug. 2003.

[13] T. D. Nguyen and N. R. Jennings, "Concurrent bi-lateral negotiation in agent systems," in *Proc. of the fourth DEXA Workshop on e-Negotiations*, 2003, pp. 839–844.

[14] D. C. Parkes, L. H. Ungar, and D. Foster, "Accounting for cognitive costs in on-line auction design," *Agent Mediated Electronic Commerce. Lecture Notes in Artificial Intelligence*, vol. 1571, pp. 25–40, 1999.

[15] I. Rahwan, R. Kowalczyk, and H. H. Pham, "Intelligent agents for automated one-to-many e-commerce negotiation," *Australian Computer Science Communications*, vol. 24, no. 1, pp. 197–204, Jan./Feb. 2002.

[16] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109.

[17] K. M. Sim, "A market-driven model for designing negotiation agents," *Computational Intelligence, Special issue in Agent Technology for E-commerce*, vol. 18, no. 4, pp. 618–637, 2002.

[18] ——, "Equilibria, prudent compromises, and the "waiting" game," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 33, no. 3, pp. 712– 724, Jun. 2005.

[19] K. M. Sim and C. Y. Choi, "Agents that react to changing market situations," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 33, no. 2, pp. 188–201, Apr. 2003.

[20] M. Vetter and S. Pitsch, "Towards a flexible trading process over the internet," in *Agentlink*, ser. Lecture Notes in Computer Science, vol. 1991. Springer, 2001, pp. 148–162.