Enhancing Multi-agent Bargaining with the TD-based Reinforcement Learning Approach

Shiu-li Huang  Fu-ren Lin
Department of Information Management
National Sun Yat-sen University
Kaohsiung, Taiwan
Email: slh@mis.nsysu.edu.tw, frlin@cc.nsysu.edu.tw

Abstract

This study proposes a negotiation mechanism that applies TD-based reinforcement learning to deal with on-line bargaining between two parties both with incomplete information. The agent embedded with the TD-based reinforcement learning capability can learn dynamic strategy incrementally by itself with the past bargaining experiences. This study investigates the scenarios that a TD-based seller agent bargains with buyers who have different risk-attitudes, and both seller and buyer agents gifted with the TD-based reinforcement learning capability to negotiate with each other. Bargaining experiments are conducted on JADE, a software agent framework based on the FIPA specifications, to evaluate the bargaining performance in average payoff and settlement rate. Results show that the negotiation mechanism handles the multi-agent bargaining process effectively. This study can be further applied to electronic commerce environment for on-line automated bargaining.

Keywords: Markov decision process, reinforcement learning, temporal difference learning, risk-attitude, automated bargaining, multi-agent bargaining process

1. Introduction

Bargaining or negotiation is generally used interchangeably, which describes the negotiation between two or more parties attempting to agree on a mutually acceptable outcome from their negatively correlated preferences [1]. The application of multi-agent technologies to automated bargaining emerges with the growing interests in applying agent technologies to electronic commerce.

In the game theory field, games can be viewed by various dimensions, such as static vs. dynamic, and complete information vs. incomplete information. In one dimension, a game is static when the players choose their actions simultaneously or each player cannot observe the opponents' actions. A dynamic game is a game where players move are explicit and each player knows the other players' moves while making each of his or her decisions. In the other dimension, a game is said to have complete information when the players know the payoffs of the others; otherwise, it is an incomplete information game. A bargaining can be viewed as a sequential, dynamic game with incomplete information. Bargainers observe one another's actions to decide what actions should be taken to react without knowing the others' payoffs. Since early 1980s, some researchers have developed models of sequential bargaining with incomplete information, but these models are only suitable for simplified settings. How an agent to perform automated bargaining process has not been fully realized [2,3,4,5].

Some efforts have been spent on designing agents for automated negotiation, such as Kasbah [6], AuctionBot[7], Tete-a-Tete [8], and MARI [9]. However, these agents are not gifted with the learning ability, and only adopt static bargaining strategies. To grant more dynamic strategies to software agents, some researchers propose the rule-based approach. For example, Sadri, Toni and Torroni [10] modeled dialogues for the one-to-one agent negotiation via the logic-based approach. Kraus, Sycara, and Evenchik [11] presented the logical model of an agent’s mental states in its beliefs, desires, intentions, and goals. Huang, Yuan, and Lin [12] propose an automated bargaining system with the persuasion capability that presents persuasion arguments using fuzzy rules reasoning while offering prices. Some researchers adopt the techniques of machine learning, such as Bayesian probability [13], genetic algorithm [14,15], genetic programming [16,17], and sequential pattern [18], to enable learning ability in the bargaining process.

Among those machine learning techniques, the temporal difference (TD) based reinforcement learning exhibits its suitability for an agent to strengthen its strategies while interacting with its environments. A bargaining process is a Markov decision process in which the agent can perceive a set of distinct states of environment and has a set of actions that the agent can perform. At each discrete time step, the agent can sense the current state and choose an action to perform. After that, the environment responds to the agent...
with reward (positive reinforcement) or punishment (negative reinforcement) which composes the succeeding state. The task for an agent is to learn optimal policy to maximize the total rewards. The TD-based reinforcement learning best fits to this process while the numbers of states and actions are finite [19,20]. The TD-based reinforcement learning has been applied to the balance-control problem, such as cart-pole balancing task [21,22,23] and ball-beam system [24,25]. This method also demonstrates impressive performance to deal with mobile robot control [26,27], forecasting problem [28,29], and scheduling problem [30,31].

In the game-playing area, the TD-Gammon program for playing the Backgammon game achieves the human master level [32]. It also shows that the TD learning approach outperforms supervised learning approach, which relies on human expertise. Unlike supervised learning which uses the error between each prediction and actual outcome to learn, the TD method learns from the difference of two successive predictions. The TD method also performs well to play Go game [33,34,35]. Unlike Backgammon and Go games in which players make positional judgment with complete information, in a bargaining game with incomplete information, bargainers do not know the opponent’s bottom line. Bearing for enhancing the automated bargaining process in electronic markets, in this study, we adopt the TD-based reinforcement learning to strengthen the automated bargaining performance. The agents embedded with the TD-based reinforcement learning capability can learn how to offer and counteroffer on their own through the reinforcement signals and tackle delayed-reward prediction problem.

In this paper, we will describe the TD-based reinforcement learning approach in details and the advantages of using it to tackle the on-line bargaining problem in the next section. Section 3 explains the architecture of the TD-based bargaining agent, and Section 4 designs bargaining experiments for evaluating the performance of multi-agent automated bargaining with the TD-based reinforcement learning capability. The experimental results are analyzed in Section 5. We conclude this research in Section 6.

2. TD-based Reinforcement Learning

In the reinforcement learning paradigm, a learning agent continually receives sensory inputs from the environment, selects and performs actions to react to the environment. After each action, the agent receives a scalar signal called reinforcement from the environment. The reinforcement can be positive (reward), negative (punishment), or zero. The objective of learning is to construct an optimal action selection policy that maximizes the agent’s gain. A natural measure of the gain is the discounted cumulative reinforcement [21].

Sutton [19] addresses the temporal credit assignment problem using temporal difference (TD) learning. Conventional prediction-learning methods are trained by comparing the differences between predicted and actual outcomes, while TD learning methods are trained by comparing the differences between temporally successive predictions, and the learning carries on whenever there is a change between predictions over time. For example, a company forecasts its annual financial performance at the end of each season. The company may not know the actual annual revenue at the time of making the prediction until the end of the year. The conventional prediction-learning approach compares each prediction with the actual annual performance at the end of the year, but the TD approach is to compare each prediction with its following prediction.

The Q-learning is a type of model-free reinforcement learning [20]. The action selection policy is formulated as

$$\pi^*(s) = \arg\max_a Q(s,a)$$

(1)

$Q(s,a)$ is the evaluation function which is the maximum discounted cumulative reward that can be achieved starting from state $s$ and executing action $a$. The agent learns the optimal policy $\pi^*$ that chooses the action $a$ in state $s$ to maximize the $Q(s,a)$. For learning actual $Q$ function, the agent repeatedly observes the current state $s$, execute action $a$, receives immediate rewards $r$, and then observes the new state $s'$. The agent updates the estimate of $Q$ function $\hat{Q}(s,a)$ during the nth iteration in a nondeterministic environment according to

$$\hat{Q}_n(s,a) \leftarrow (1-\alpha_n)\hat{Q}_{n-1}(s,a) + \alpha_n[r + \gamma \max_{a'}\hat{Q}_{n-1}(s',a')]$$

(2)

In Equation (2), $\gamma (0 \leq \gamma < 1)$ is a discount factor and $\alpha_n (0 \leq \alpha_n < 1)$ is a learning factor. The value of $\alpha_n$ decreases as $n$ increases, and as $n \to \infty$, $\hat{Q}_n(s,a)$ will converge to $Q(s,a)$. The agent’s over-commitment to actions with high $\hat{Q}$-values that are found during early training, and failure to explore other
actions that have even higher values, can be avoided by applying the probabilistic approach to select actions; i.e.,
\[
P(a_t|s_t) = e^{\hat{Q}(s_t,a_t) / T} / \sum_a e^{\hat{Q}(s_t,a) / T}
\]
(3)

Every action is assigned a nonzero probability, and actions with higher \(\hat{Q}\) values have higher probabilities to be chosen. \(T\) is a temperature factor to adjust the randomness.

Lin [36] applies the backpropagation neural network to construct the \(Q\) function. The error is defined as the difference between before and after updating \(Q\)-values, and the learning occurs by adjusting the weights of the neural network. The weight in time step \(t\) is updated by adding \(\Delta w_t\), which is defined as
\[
\Delta w_t = \eta [Q_{t+1} - Q_t] \nabla Q
\]
(4)

With the conventional \(Q\)-learning described above, the \(\hat{Q}\) values are only updated by one-step lookahead. The training rule of Equation (2) reduces the difference between the estimated \(\hat{Q}\) values of a state and its subsequent state. To deal with the temporal credit assignment problem and construct TD-based reinforcement learning mechanism, we should design a training rule that reduces discrepancies between this state and more distant states. Sutton introduces the parameter \(\lambda (0 \leq \lambda \leq 1)\) to combine the predictions obtained from various lookahead steps in the following fashion [19]:
\[
Q^\pi(s_t,a_t) = r_t + \gamma (1 - \lambda) \max_a \hat{Q}(s_{t+1},a_t) + \lambda Q^\pi(s_{t+1},a_t)
\]
(5)

Introducing the TD(\(\lambda\)) method to the back-propagation neural network, \(\Delta w_t\) becomes
\[
\Delta w_t = \eta [Q_{t+1} - Q_t] \sum \lambda^{t-t'} \nabla Q
\]
(6)

The quantity \(\lambda\) controls the temporal credit assignment by determining how an error detected at a given time step feeds back to correct previous predictions. When \(\lambda = 1\), the extreme case, all predictions are altered to an equal extent and the error feeds back without decay with the time. When \(\lambda = 0\), the case of conventional \(Q\)-learning, no feedback occurs beyond the current time step. In the case of \(0 < \lambda < 1\), greater alterations tend to occur in more recent predications.

In this research, we use software agents gifted with the TD-based reinforcement learning capability to perform on-line bargaining transaction with the following reasons.

1. The bargaining process is a temporal credit assignment problem. In a bargaining game, like playing chess, the agent receives the actual cumulative reward (actual payoff) at the end of the game. If the deal is made, the actual payoff is the difference between the settled price and the reservation price. If no deals are made, the payoff is zero or negative value induced from some costs. The TD-based reinforcement learning is suitable for this temporal credit assignment problem.

2. The reinforcement learning is a semi-supervised learning which can incrementally enhance the bargaining strategies without domain experts’ teaching. It is hard to externalize human bargaining knowledge to formalize rules to guide software agents to execute the bargaining process. An agent using the TD-based reinforcement learning does not need a supervisor to guide its actions. In the bargaining process, an agent can learn by itself how to react to the counterpart’s offer through the rewards fed back from the counterpart’s offers. The TD-based reinforcement learning utilizes outcomes from each interaction to incrementally update its policy instead of waiting until the end of the game. In the electronic commerce environment, software agents can learn from their interaction in the automated bargaining process.

3. Agents can autonomously react to other agents’ or users’ responses without human intrusions. When an agent observes the state of its environment, it can quickly decide what action should be performed based on the \(Q\)-values and the probabilistic action-select function (e.g., Equation (3)). In the electronic commerce environment, autonomous agents enable the online automated bargaining process.

3. TD-based Bargaining Agent

For determining an effective bargaining strategy in terms of returning a price to reach a good bargaining position which leads to the end of bargaining, an agent should calculate its own utility and perceive its opponent’s bargaining power along the bargaining process. An agent’s utility can be calculated by its reservation price and seller’s current price offer (seller’s price position) and buyer’s current price offer (buyer’s price position) [37]. The perceived opponent’s bargaining power can be estimated by
analyzing opponent’s concession behavior [38] (e.g., Analyzing opponent’s average concession range that is the difference between opponent’s initial price offer and current price offer divided by current time step). At the beginning of a bargaining process, an agent knows its reservation price, its initial price and the opponent’s initial price. Therefore, a bargaining agent should obtain the information of the seller’s and buyer’s price positions and current time step in each state. We use the back-propagation neural network to construct the $Q(s, a)$ function for an agent to learn bargaining strategies. The inputs of $Q(s, a)$ function include a state $s$ represented by a seller’s price position, buyer’s price position, and current time step; an action $a$ in terms of the price offered by the agent. The output of $Q(s, a)$ function is the $Q$-value that indicates the prediction of agent’s actual payoff at the end of the bargaining.

The automated bargaining agent gifted with TD-based reinforcement learning capability is called the TD-based bargaining agent. In each interaction, the bargaining agent uses the $Q$-function network to calculate the $Q$-values of each candidate actions in the current state. By the probabilistic action-select function as shown in Equation (3), actions with higher $Q$-values have higher probabilities to be chosen to respond to the counterpart’s agent. The agent adopts Equation (6) to update the weights in the $Q$-function network and receives delayed reward (actual payoff) at the end of the bargaining. The architecture of the TD-based Bargaining agents is depicted in Figure 1. In evaluating the proposed architecture of the TD-based bargaining agent, we conduct two sets of bargaining experiments: agent-to-human and agent-to-agent. The experimental designs are described in Section 4.

4. Multi-agent Bargaining with the TD-based Learning Approach

In the electronic commerce, a web site hosted by a seller may provide automated agents for numerous human users to directly bargain with. We would like to understand how the TD-based bargaining agent acts as the seller’s agent to bargain with buyers who have different risk attitudes. A seller agent formulates its price position as $(sp_t - srp) / (sp_1 - srp)$, where $srp$ is the seller’s reservation price, $sp_t$ is the seller agent’s offer in time step $t$, and $sp_1$ is seller’s initial offer. A seller agent views a buyer’s price position as $(bp_t - bp_1) / (bp_1 - bp_2)$, where $bp_t$ is the buyer’s counteroffer in time step $t$, and $bp_1$ is the buyer’s initial counteroffer. Since a seller agent does not know the buyer’s reservation price during the bargaining process, it treats the buyer’s reservation prices as the initial offer $sp_1$. The current time step $t$ is a positive integer and the input of time step is encoded by $1 / t$. The actual payoff of the seller agent is encoded by $(settled_price - srp) / (sp_1 - srp)$ if the deal is made, otherwise is 0.

A bargaining lesson is defined as a bargaining which starts when either party offers the initial price and ends when a deal is made or either party quits. In the beginning of a bargaining lesson, the seller agent offers the initial price $(sp_1)$ and then receives the buyer’s initial counteroffer $(bp_1)$. The seller agent’s candidate actions in time step 2 are (1) persisting in its previous offer $sp_1$, (2) making a concession, where we define the concession unit as $\delta$, so that the concession offer could be one of $sp_1, sp_1 - \delta, sp_1 - 2\delta, \ldots, srp$, (3) informing the buyer that it persists its previous offer, and if the buyer does not accept this offer, it will quit this bargaining game, and (4) accepting buyer’s counteroffer. In every time step, the seller agent chooses an action from candidate actions, abiding by the following three constraints. First, a seller agent cannot regret for its offers, i.e., $sp_t \geq srp$. Second, its offers cannot be lower than the seller’s reservation price, i.e., $sp_t \geq srp$. If the seller agent has no way to make concession, it will quit or accept the counteroffer.

Figure 1. The architecture of TD-based bargain agents
Third, the seller agent’s offer cannot be lower and equal to the buyer’s previous counteroffer; that is, \( sp_1 > bp_{-1} \), otherwise the seller must accept the \( bp_{-1} \).

### 4.1 TD-based bargaining seller agent vs. buyers

We build a human-like buyer agent to simulate a human buyer. The buyer agent is not gifted with the TD-Bargain mechanism but with human-like risk attitudes. Generally, if people would trade a gamble for a sure payoff that is less than the expected monetary value won from the gamble, they are risk-averse and the shapes of their utility functions are concave (the curve opens downward). People are risk-seeking might be eager to get into gambling and their utility functions are convex (the curve opens upward). People also can be risk-neutral, whose utility functions are drawn in straight lines. For a risk-neutral person, the maximal expected monetary value is the same as maximal expected utility.

We conduct Game 1 to simulate and evaluate agent-to-human bargaining processes. We set the seller agent’s reservation price \( sp_1 \) $60, the initial offer \( sp_{-1} \) $100, and the concession unit $5 (\delta).

**Game 1: agent vs. buyer with three risk attitudes and various price ranges**

A buyer’s reservation price \( brp \) belongs to the uniform distribution over interval \([80, 100]\), and his/her initial counteroffer \( bp_{-1} \) belongs to the uniform distribution over interval \([50, 70]\). A buyer’s risk attitude is randomly selected from risk-neutral, risk-seeking, or risk-averse.

A risk-neutral buyer makes the fixed and intermediate range of price in each concession. With the feasible range \( bp_{1} \sqcup sp \sqcup brp \), the probability distribution for a buyer to accept \( sp \), follows the decline slop, \( \frac{prob_1}{prob_2} = \frac{brp - sp}{brp - bp_1} \), where \( prob_1 \) is set to 1 and \( prob_2 \) is set to 0.5. In the bargaining process, when the seller agent responds that it is about to quit the game if the buyer does not accept the offer, \( prob_2 \) is increased from 0.5 to 0.7.

A risk-seeking buyer makes small range concessions in the early stages followed by larger concessions toward the later stages of the bargaining process. With the feasible range \( bp_{1} \sqcup sp \sqcup brp \), the probability distribution for a buyer to accept \( sp \) follows the decline slop, \( \frac{prob_1}{prob_2} = \frac{brp - sp}{brp - bp_1} \), where \( prob_1 \) is set to 1 and \( prob_2 \) is set to 0.2. When the seller agent responds that it is about to quit the game if the buyer does not accept the offer, \( prob_2 \) is increased from 0.2 to 0.4.

A risk-averse buyer makes large range concessions in the early stages followed by smaller concessions toward the later stages of the bargaining process. A risk-averse buyer makes large range concessions in the early stages followed by smaller concessions toward the later stages of the bargaining process. With the feasible range \( bp_{1} \sqcup sp \sqcup brp \), the probability distribution for a buyer to accept \( sp \) follows the decline slop, \( \frac{prob_1}{prob_2} = \frac{brp - sp}{brp - bp_1} \), where \( prob_1 \) is set to 1 and \( prob_2 \) is set to 0.8. When the seller agent responds that it is about to quit the game if the buyer does not accept the offer, \( prob_2 \) is increased from 0.8 to 1.

In this game, the buyer cannot regret for his/her counteroffers either, \( i.e., bp_{-1} \sqcup bp_1 \), and his/her counteroffers cannot be higher than his/her reservation price, \( i.e., bp_1 \sqcup brp \). If the buyer has no way to make concession, he or she can quit the bargaining or accept the offer.

### 4.2 TD-based bargaining seller agent vs. TD-based bargaining buyer agent

We further investigate the scenario that both seller and buyer utilize TD-based bargaining agents to negotiate with each other. The buyer agent encodes the seller agent’s price position as \( (sp_1 - bp_1) \sqcup (sp_1 - bp_1) \), and the buyer agent’s price position as \( (bp_1 - bp_1) \sqcup (brp - bp_1) \). Since a buyer agent does not know the seller’s reservation price, either, it treats the seller’s reservation price as the initial counteroffer \( bp_1 \). The actual payoff of the buyer agent is encoded by \( (brp - settled\_price) \sqcup (brp - bp_1) \) if the deal is made, otherwise is 0. The bargaining game is designed as follows.

**Game 2: agent vs. agent**

The seller agent’s behavior has been described in the beginning of Section 4, and we set \( sp_1 \) $60, \( sp_{-1} \) $100, and \( \delta \) $5. The buyer agent’s behavior is described as follows. In this experimental game, we assume buyer’s reservation price \( brp \) is $90, initial counteroffer \( bp_1 \) is $50, and concession unit \( \delta \) is $5. The buyer agent gives the initial counteroffer and then receives the seller agent’s reaction. The buyer agent’s second move can be (1) staying at its previous counteroffer \( bp_1 \), (2) making a concession, the concession counteroffer could be one of \( bp_1 + \epsilon, bp_1 + 2\epsilon, \ldots, brp \), (3) informing the seller agent that it persists its previous counteroffer, and if the seller agent does not accept this, it will quit this bargaining.
lesson, or (4) accepting seller agent’s offer. In every time step, the buyer agent chooses an action from candidate actions, abiding by the following constraints. First, it cannot regret for its counteroffers, i.e., \( bp_{t+1} \geq bp_t \). Second, its counteroffers cannot be higher than the buyer’s reservation price, i.e., \( bp_t \leq brp \). If the buyer agent cannot make concession, it will quit or accept the offer. Third, the buyer agent’s counteroffer cannot be higher and equal to the seller agent’s previous offer, i.e., \( bp_t < sp_{t-1} \), otherwise the buyer agent must accept the \( sp_{t-1} \).

5. Experimental Design and Results

The seller and buyer agents are built on the JADE (Java Agent DEvelopment framework; http://jade.cselt.it) platform that complies with the FIPA (Foundation for Intelligent Physical Agents; http://www.fipa.org) specifications. In the TD-Bargain mechanism, we fix the discount factor \( \gamma = 1 \), which assumes that the agent is patient enough and the interaction intervals are very short. The parameters of the neural network are set as learning rate 0.25, momentum 0.05, and the initial weights are generated randomly. The temperature factor of probabilistic action-selection function is 0.05.

To demonstrate the incremental improvement of the TD-based reinforcement learning, we conduct several experiments to play bargaining games. An experimental game consists of a sequence of periods, where one period contains 10,000 consecutive bargaining lessons. Experimental results are evaluated by setting \( \lambda \) at 1, 0.5, and 0 respectively in these games. The performance of bargaining games is evaluated by the average payoff and settlement rate per period. Average payoff at each period is calculated by summing up all payoffs from all bargaining lessons in a period, and then dividing the sum by the number of lessons in that period. The settlement rate per period is calculated by dividing the number of deals made by the number of lessons in a period. To facilitate the termination decision of the incremental learning process, we define one time-window as 10 periods. When the increase of the settlement rate is lower than 1% in a time-window, the experimental game stops. We also evaluate the effects of the number of hidden nodes in the Q-function network on the bargaining performance. These results are shown and discussed as follows.

**Results from Game 1: agent vs. buyer with three risk attitudes and various price ranges**

Game 1 is designed to meet realistic business situations, where the different buyers have different attitudes, reservation prices, and initial counteroffers. In Figure 2, we can see that the seller agent with \( \lambda = 1 \) has the highest performance. In the 15th period, the seller agent with \( \lambda = 1 \) can reach the highest average payoff $19.26 (the average surplus = average \( brp - srp \) = (100+80)/2 – 60 = $30). In the 2nd period, the seller agent has earned more than half of the average surplus per bargaining lesson. In the 17th period, the seller agent with \( \lambda = 1 \) can reach the highest settlement rate 96.09%. This result indicates that the TD-based seller agent can keep its strength in bargaining with buyers with dynamic negotiation behaviors in terms of the price range, initial counteroffer, and risk attitudes.

![Figure 2. Bargaining performance of TD-Based seller agent in Game 1](image)

**Results from Game 2: agent vs. agent**

To measure the effect of different \( \lambda \) values of the seller agent, we fix the buyer agent’s \( \lambda \) value at 0.5 and adjust the seller agent’s \( \lambda \) values at 0, 0.5 and 1 respectively. Figure 3 reveals that the seller agent with \( \lambda = 1 \) can reach the highest average payoff $10.27 and settlement rate 77.34% in the 68th period. In the other experiment, we also fix the \( \lambda \) value of seller agent at 0.5 and adjust the buyer’s \( \lambda \) values at 0, 0.5 and 1 respectively. Figure 4 depicts the buyer agent’s bargaining performance. The buyer agent with \( \lambda = 1 \) can reach the highest average payoff $11.77 and settlement rate 71.47% in the 69th period.
From these experiments, we found that both of the seller and buyer agents gradually increase their payoffs as they learn along with the periods. It shows that the TD-based reinforcement learning capability enables autonomous agents to perform automated bargaining to improve their bargaining strategies incrementally, especially when $\lambda = 1$.

The effects of the number of hidden nodes of Q-function network

In the neural network, the hidden nodes in the hidden layer perform complex nonlinear mapping between input and output nodes. Fewer hidden nodes usually have better generalization ability and less over-fitting problem. However, too few hidden nodes may not effectively model this non-linearity. The most common way to determine the number of hidden nodes is through sensitivity analysis by testing various numbers. Tesauro’s research [32] using neural network with temporal difference learning for the game of Backgammon suggests that adding more hidden nodes would continue to yield further improvements in playing ability. In those experiments conducted aforementioned, we design the architecture of the Q-function network with three hidden nodes. We increase the number of hidden nodes from three to six and nine and replay the Game 1 with $\lambda = 1$. The performance tests of the seller agent are depicted in Figure 5. We can find that the neural network with three hidden nodes is the most compact architecture and capable of playing Game 1.

We also test the effects of the number of hidden nodes on the bargaining performance in Game 2. In this experiment, we fix the $\lambda$ values of both seller and buyer agents at 1, and adjust the numbers of their hidden nodes respectively. The bargaining performance of seller agent and buyer agent are depicted in Figure 6. The performance in terms of average payoff and settlement rate of these three $\lambda$ values has no
significant differences. Therefore, the neural network with three hidden nodes is also capable of playing
this agent-to-agent bargaining game.

![Graphs showing results from different numbers of hidden nodes in Game 2](image)

**Figure 6. Results from different numbers of hidden nodes in Game 2**

6. Conclusions

This study proposes the TD-based bargaining agent architecture gifted with the TD-based
reinforcement learning mechanism to perform price negotiation game between two parties both with
incomplete information. We use the back-propagation neural network to implement the $Q$-function
network to execute the TD-based reinforcement learning. The $\lambda$ parameter controls the temporal credit
assignment by determining how an error detected at a given time step feeds back to correct previous
predictions. Two sets of bargaining games are designed to measure the agent’s bargaining ability with the
TD-based reinforcement learning mechanism. The experimental results reveal that the effectiveness of the
TD-based bargaining agent in terms of the average payoff and the deal settlement rate.

Specifically, we discover that introducing the TD($\lambda$) method to perform reinforcement learning in
dealing with the bargaining problem can improve the learning ability, especially with $\lambda=1$. The TD-based
seller agent can achieve the comparative performance in bargaining with buyers with various risk attitudes.
When two agents both with the TD-based learning capability act as seller and buyer agents respectively,
they both can achieve the improved performance while learning along a sequence of bargaining periods.
The neural network with three hidden nodes is the most compact and effective architecture for these
bargaining games concluded from experiments.

The experimental results from Game 1 reveal that the TD-based seller agent is able to bargain with
buyers to obtain good payoffs no matter of buyers’ risk-attitudes. On the other side, buyers can also invoke
TD-based buyer agents to bargain with seller agents in light of payoff maximization policy. According to
the experimental results of Game 1 and Game 2, we can find that a seller agent obtains less surplus in the
agent-to-agent case than the agent-to-human case. The drawback of the TD-based reinforcement approach
is that it needs thousands of training lessons to optimize the $Q$-function network. Some speedup methods
proposed by Lin’s research [36] can tackle this problem.

Although we only consider price issue in bargaining games in this study, the TD-based learning
mechanism can be extended to deal with the multi-issue and time constraint cases. In future research, we
can consider the multi-issue by adding more input nodes of the $Q$-function network to encode states and
actions. To deal with the time constraint in the bargaining process, for example, a bargaining lesson should
be terminated in the time step $t$, one way is to set the agent to complete a bargaining lesson within time $t$.
The other way is to discount the reward after time $t$. Future research can also focus on the implementation
of one-to-many bargaining cases, where one agent has to bargain with many agents simultaneously, and
introducing argumentation to the bargaining process for receiving information and feedbacking
immediately from negotiators.

References

1969.


