A Deep Reinforcement Learning Approach to Concurrent Bilateral Negotiation

Pallavi Bagga[1*], **Nicola Paoletti**[1], **Bedour Alrayes**[2] and **Kostas Stathis**[1]
[1]Royal Holloway, University of London, UK
[2]King Saud University, Saudi Arabia
{pallavi.bagga, nicola.paoletti}@rhul.ac.uk, balrayes@ksu.edu.sa, kostas.stathis@rhul.ac.uk

Abstract

We present a novel negotiation model that allows an agent to learn how to negotiate during concurrent bilateral negotiations in unknown and dynamic e-markets. The agent uses an actor-critic architecture with model-free reinforcement learning to learn a strategy expressed as a deep neural network. We pre-train the strategy by supervision from synthetic market data, thereby decreasing the exploration time required for learning during negotiation. As a result, we can build automated agents for concurrent negotiations that can adapt to different e-market settings without the need to be pre-programmed. Our experimental evaluation shows that our deep reinforcement learning based agents outperform two existing well-known negotiation strategies in one-to-many concurrent bilateral negotiations for a range of e-market settings.

1 Introduction

We are concerned with the problem of learning a strategy for a buyer agent to engage in concurrent bilateral negotiations with unknown seller agents in open and dynamic e-markets such as E-bay[1]. Previous work in this context has mainly focused on heuristic strategies [Nguyen and Jennings, 2004; Mansour and Kowalczyk, 2014; An *et al.*, 2006], some of which adapt to changes in the environment [Williams *et al.*, 2012]. Different bilateral negotiations are managed in such strategies either through a coordinator agent [Rahwan *et al.*, 2002] or by coordinating multiple dialogues internally [Alrayes and Stathis, 2013], but do not support learning which is our main focus. Other approaches use learning based on Genetic Algorithms (GA) [Oliver, 1996; Zou *et al.*, 2014], but they require a huge number of trials for obtaining a good strategy, which makes them infeasible for online settings. Reinforcement Learning (RL)-based negotiation typically employ Q-learning [Papangelis and Georgila, 2015; Bakker *et al.*, 2019] which does not support continuous actions. This is an important limitation in our setting because we want to learn how much to concede e.g. on the price of an item for sale, which naturally leads to a continuous action space. Consequently, the design of autonomous agents capable of learning a strategy from concurrent negotiations with other agents is still an important open problem.

We propose, to the best of our knowledge, the first Deep Reinforcement Learning (DRL) approach for one-to-many concurrent bilateral negotiations in open, dynamic and unknown e-market settings. Our DRL-inspired model *ANEGMA* (*A*daptive *NEG*otiation model for e-*MA*rkets) allows the buyer to develop an adaptive strategy to effectively use against its opponents. Such opponents use fixed but unknown to the agent strategies during negotiations, giving rise to an environment with incomplete information. We choose deep neural networks as they provide a rich class of strategy functions to capture the complex decisions-making required.

Since RL approaches need a long time to find an optimal policy from scratch we pre-train our deep negotiation strategies using supervised learning (SL) from a set of examples. To overcome the lack of real-world data for the initial training, we generate synthetic datasets using the simulation environment in [Alrayes *et al.*, 2016] and two well-known strategies for concurrent bilateral negotiation described in [Alrayes *et al.*, 2018] and [Williams *et al.*, 2012] respectively.

With this work, we empirically demonstrate three important benefits of our deep learning framework for automated negotiations: 1) existing negotiation strategies can be accurately approximated using neural networks; 2) evolving a pre-trained strategy using DRL with additional negotiation experience yields strategies that even outperform the teachers, i.e., the strategies used for supervision; 3) buyer strategies trained assuming a particular seller strategy quickly adapt via DRL to different (and unknown) sellers' behaviours.

In summary, our contribution is threefold: we propose a novel agent model for one-to-many concurrent bilateral negotiations based on DRL and SL; we extend the simulation environment [Alrayes *et al.*, 2016] to generate data and perform experiments that support agent learning for negotiation; and we run extensive experiments showing that our approach outperforms the existing strategies and produces adaptable agents that can transfer to a range of e-market settings.

2 Related Work

RL has been proposed as a learning mechanism for negotiation environments with incomplete information. A number of approaches use Q-learning, e.g. in contract negotiation [Rodriguez-Fernandez *et al.*, 2019], however the state/action space in that work is not continuous as in ours. The work of [Bakker *et al.*, 2019] uses tabular Q-learning to learn the bidding strategy by discretizing the continuous state/action space (not optimal for large state/action spaces as it may lead to the curse of dimensionality and loss of rele-

*Contact Author
[1]https://www.ebay.com/

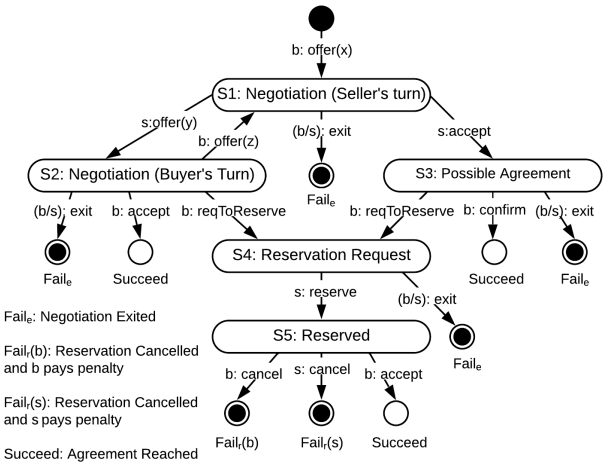

Figure 1: The CONAN Negotiation Protocol [Alrayes *et al.*, 2018]

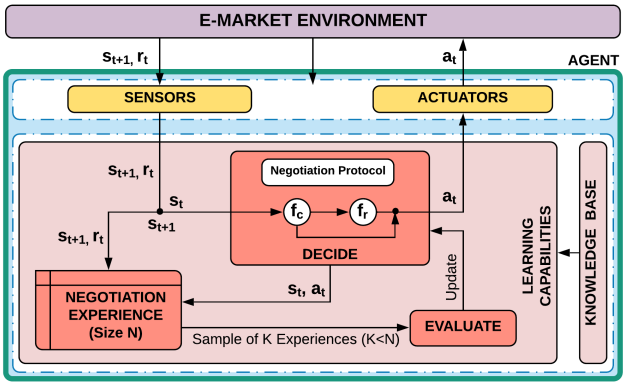

Figure 2: The Architecture of *ANEGMA*

vant information about the state/action domain structure). We avoid these issues by using a model-free actor-critic RL approach [Lillicrap *et al.*, 2016].

The work of [Lewis *et al.*, 2017] combines SL (Recurrent Neural Network (RNN)) and RL (REINFORCE [Williams, 1992]) to learn a strategy and linguistic skills by being trained on human negotiation dialogues. We also combine SL and RL but with the main focus to train the agent learn a bidding strategy from agent interactions governed by a negotiation protocol. In addition, we use Artificial Neural Network (ANN) for SL and the actor-critic model called DDPG [Lillicrap *et al.*, 2016] for RL.

Independently of the approach, numerous works in the domain of bilateral negotiation rely on the Alternating Offers protocol [Rubinstein, 1982] as the negotiation mechanism, which, despite its simplicity does not capture many practical bargaining scenarios. We will be adopting the CONAN negotiation protocol shown in Fig. 1, that can support a wider range of practical negotiation applications.

3 The *ANEGMA* Model

3.1 Negotiation Environment

We consider e-marketplaces like E-bay where the competition is visible, i.e. a buyer can observe the number of competitors that are dealing with the same resource from the same seller. We assume the environment to be a single e-market m with P agents, with a non-empty set of buyers B_m and a non-empty set of sellers S_m – these sets need not be mutually exclusive. For a buyer $b \in B_m$ and resource r, we denote with $S^t_{b,r} \subseteq S_m$ the set of sellers from market m which, at time point t, negotiate with b for a resource r (over a range of issues I). The buyer b uses $|S^t_{b,r}|$ negotiation threads, in order to negotiate concurrently with each seller $\in S^t_{b,r}$. We assume that no agent can be both buyer and seller for the same resource at the same time, that is, $\forall b, r, t.\ s \in S^t_{b,r} \implies S^t_{s,r} = \emptyset$. $C^t_{b,r} = \{b' \neq b \in B_m \mid S^t_{b',r} \neq \emptyset\}$ is the set of competitors of b, i.e. those agents negotiating with the same sellers and for the same resource r as that of b. However, it is possible that a seller, whom a buyer b is negotiating with, to accept a deal from a competitor buyer $b' \in C^t_{b,r}$.

The class of concurrent bilateral negotiations that we study is governed by the negotiation protocol of Fig. 1. This protocol assumes an open e-market where agents can enter or leave the negotiation at their own will. A buyer b always starts the negotiation by making an offer whose start time is t_{start}. Any negotiation is for a resource r, since we index the negotiation thread with the seller's name s and the resource r, and can last for up to time t_b, the maximum time b can negotiate for. The deadline for b is, thus, $t_{end} = t_{start} + t_b$, which for simplicity we assume for all resources being negotiated. Information about the deadline t_b, Initial Price IP_b and Reservation Price RP_b is private to each $b \in B_m$. Each seller s also has its own Initial Price IP_s, Reservation Price RP_s and maximum negotiation duration parameter t_s (which are not visible by other agents). The protocol is turn-based and allows agents to take actions from a pool $Actions$ at each negotiation state (S1 to S5 – see Fig. 1), where $Actions = \{offer(x), reqToReserve, reserve, cancel, confirm, accept, exit\}$. In S1, s can accept current b's offer and move to S3, or make a counter-offer and move to S2. In S2, b can accept s's offer (success), make a counter-offer and move to S1, or request to reserve and move to S4. In S3, b can confirm its offer (success), or request to reserve and move to S4. In S4, s can reserve and move to S5. In S5, b can accept (success). Either agent can exit the negotiation at any state.

3.2 *ANEGMA* Components

Our proposed agent negotiation model supports learning during concurrent bilateral negotiations with unknown opponents in dynamic and complex e-marketplaces. In this model, we use a centralized approach in which the coordination is done internally to the agent via multi-threading synchronization. This approach minimizes the agent communication overhead and thus, improve the run-time performance. The different components of the proposed model are shown in Fig. 2 and explained below.

Physical Capabilities

The *sensors* of the agent enable it to access an e-marketplace. They allow a buyer b to perceive the current (external) state of the environment s_t and represent that state locally in the

form of internal attributes as shown in Table 1. Some of these attributes (NS_r, NC_r) are perceived by the agent using its sensors, some of them (IP_b, RP_b, t_{end}) are stored locally in its knowledge base and some of the them (S_{neg}, X_{best}, T_{left}) are obtained while interacting with other seller agents during a negotiation. At time t, the internal agent representation of the environment is s_t, which is used by the agent to decide what action a_t to execute using its *actuators*. Action execution then changes the state of the environment to s_{t+1}.

Learning Capabilities

The foundation of our model is a component providing learning capabilities similar to those in the Actor-Critic architecture as in [Lillicrap *et al.*, 2016]. It consists of three subcomponents: *Negotiation Experience*, *Decide* and *Evaluate*.

Negotiation Experience stores historical information about previous *negotiation experiences* which involve the interactions of an agent with other agents in the market. Experience elements are of the form $\langle s_t, a_t, r_t, s_{t+1} \rangle$, where s_t is the state of the e-market environment, a_t is action performed by b at s_t, r_t is scalar reward or feedback received from the environment and s_{t+1} is new e-market state after executing a_t.

Decide refers to a negotiation strategy which helps b to choose an optimal action a_t among a set of actions ($Actions$) at a particular state s_t. In particular, it consists of two different functions f_c and f_r. f_c takes state s_t as an input and returns a discrete action among *counter-offer, accept, confirm, reqToReserve* and *exit*, see (1). When f_c decides to perform a *counter-offer* action, f_r is used to compute, given an input state s_t, the value of the counter-offer, see (2). From a machine learning perspective, deriving f_c corresponds to a classification problem, deriving f_r to a regression problem.

$$f_c(s_t) = a_t, a_t \in Actions \tag{1}$$

$$f_r(s_t) = x, x \in [IP_b, RP_b] \tag{2}$$

Evaluate refers to a critic which helps b learn and evolve the strategy for unknown and dynamic environments. It is a function of K ($K < N$) past negotiation experiences randomly selected. The learning process of b is *retrospective* since it depends on the feedback (or scalar rewards) obtained during classification (i.e. r_t using (3)) and regression (i.e. r'_t using (4)) from the e-market environment by performing action a_t at state s_t These rewards evaluate the discrete and continuous action of *Decide* respectively at time t . Our design of reward functions accelerate agent learning by allowing b to receive rewards after every action it performs in the environment instead of at the end of the negotiation.

$$r_t \text{ (during classification)} = \begin{cases} U_b(x,t), & \text{if } t \leq t_{end}, \text{ Agreement} \\ -1, & \text{if } t \leq t_{end}, \text{ No Deal} \\ r'_t & \text{if } a_t = \text{Counter-offer} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$r'_t \text{ (during regression)} = \begin{cases} U_b(x,t), & \text{if } t \leq t_{end}, x \leq \forall i \in O_t \\ -1, & \text{if } t \leq t_{end}, x > \forall i \in O_t \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In (3) and (4), $U_b(x,t)$ refers to the utility of offer x (generated using (2)) at time t and calculated using Initial Price (IP_b), Reservation Price (RP_b), agreement offer (x) and temporal discount factor ($d_t \in [0,1]$) [Williams *et al.*, 2012] as defined in (5), assists b to negotiate without delay. The reward function r'_t in (4) helps b learn that it should not offer more than what active sellers have already offered it. O_t is a list of preferred offers received from sellers $s \in S^t_{b,r}$ at time t, which b maintains during negotiation. In (3), "No Deal" means that the agent chooses to quit the negotiation.

Attribute	Description
NS_r	Number of sellers that b is concurrently dealing for resource r at time t ($\|S^t_{b,r}\|$).
NC_r	Number of buyer agents competing with b for resource r at time t ($\|C^t_{b,r}\|$).
S_{neg}	Current state of the negotiation protocol (S1 to S5 [Alrayes *et al.*, 2018])).
X_{best}	Best offer made by either b or s in S_{neg}.
T_{left}	Time left for b to reach t_{end} after the last action of s.
IP_b	Minimum price which b can offer at the start of the negotiation.
RP_b	Maximum price which b can offer to s.

Table 1: Agent's State Attributes

$$U_b(x,t) = \left(\frac{RP_b - x}{RP_b - IP_b}\right) \cdot \left(\frac{t}{t_{end}}\right)^{d_t} \tag{5}$$

In our experiments, the value of d_t is set to 0.6. Higher the d_t value, higher is the penalty due to delay.

4 Materials and Methods

4.1 Data Set Collection

In order to collect the dataset to train the *ANEGMA* agent using an SL model, we have used a simulation environment [Alrayes *et al.*, 2016] that supports concurrent negotiations between buyers and sellers. The buyers use the strategies presented in [Alrayes *et al.*, 2018] and [Williams *et al.*, 2012], whereas the sellers use the strategies described in [Faratin *et al.*, 1998]. We could have also collected training data using other buyer strategies for concurrent negotiation in the same setting as ours, or any real-world market data; however, to the best of our knowledge none of these were readily available. We have selected the input features of our dataset manually, and this set of features correspond to the agent's state attributes in Table 1. To avoid choosing overlapping features, we have applied the *Pearson Correlation coefficient* [Lee Rodgers and Nicewander, 1988] and ensured no correlation (with all correlation coefficients between -0.16 and 0.16; most are closer to 0) between the selected features.

4.2 Performance Evaluation Measures

To successfully evaluate the performance of *ANEGMA* and compare it with other negotiation approaches, it is necessary to identify the appropriate performance metrics. For our experiments, we have used the following widely adopted metrics [Williams *et al.*, 2012; Faratin *et al.*, 1998; Nguyen and

Jennings, 2004; Alrayes *et al.*, 2018]: *Average utility rate (U_{avg})*, *Average negotiation time (T_{avg})* and *Percentage of successful negotiations ($S_{\%}$)*, which are described below:

- U_{avg}: Sum of all the utilities of the buyer averaged over the successful negotiations. (Ideal value: High(1.0))
- T_{avg}: Total time taken by the buyer (in milliseconds) averaged over all successful negotiations to reach the agreement. (Ideal value: Low($\approx$ 1000ms))
- $S_{\%}$: Proportion of total negotiations in which the buyer reaches an agreement successfully with one of the concurrent sellers. (Ideal value: High(100%))

Our main motive behind calculating the U_{avg} is to calculate the agent profit over only successful negotiations, hence we exclude the unsuccessful ones in this metric. We capture the (un)successful negotiations in a separate metric called $S_{\%}$.

4.3 Methodology

During our experiments, sellers and competitor buyers use fixed strategies that are initially unknown to the buyer. As these strategies are fixed, they will be learned by *ANEGMA* later, after a number of simulation runs. Thus after a number of negotiation simulation runs our environment can be considered *fully-observable*. Given our *dynamic* (i.e. agents leave and enter the market at any time) and *episodic* (i.e. the negotiation terminates at some point) environment, we use a *model-free*, *off-policy* RL approach which generates a *deterministic policy* based on the *policy gradient* method to support continuous control. Specifically, we use the *Deep Deterministic Policy Gradient algorithm (DDPG)*, which is an actor-critic RL approach and generates a deterministic action selection policy for the buyer (see [Lillicrap *et al.*, 2016] for more details). We consider a *model-free RL* approach because our buyer is more concerned with determining which action to take for a particular state rather than predicting a new state. This is because the strategies of sellers and competitor buyers are unknown. On the other hand, we consider the *off-policy* approach for efficient and independent exploration of continuous action spaces. Furthermore, instead of initializing the RL policy randomly, we use a policy generated by an ANN [Goodfellow *et al.*, 2016] due to its compatibility with DRL in order to speed up and reduce the cost of the RL process. To reduce over-fitting and generalization errors, we apply regularization techniques (dropout) during the training of ANN.

5 Experimental Setup and Results

Our experiments are based on the following hypotheses.

Hypothesis A: The *Market Density (MD)*, the *Market ratio* or *Demand/Supply Ratio* (MR), the *Zone of Agreement* (ZoA) and the *Buyer's Deadline (t_{end})* have a considerable effect on the success of negotiations. Here,

- MD is the total agents in the e-market at any given time dealing with the same resource as that of our buyer.
- MR is the ratio of the total number of buyers over the sellers in the e-market.
- ZoA refers to the intersection between the price ranges of buyers and sellers for them to agree.

	Values
IP_b	$[300-350]$
RP_b	$[500-550]$
IP_s	$100\%[500-550]$, $60\%[580-630]$, $10\%[680-730]$
RP_s	$100\%[300-350]$, $60\%[380-430]$, $10\%[480-530]$
MD	H$\{30,40,50\}$, A$\{18,23,28\}$, L$\{8,10,12\}$
MR	H$\{$10:1, 1:1, 1:10$\}$, A$\{$5:1, 1:1, 1:5$\}$, L$\{$2:1, 1:1, 1:2$\}$
t_{end}	Lg[151s –210s], A[91s –150s], Sh[30s –90s]
ZoA	H(100%), A(60%), L(10%)

Table 2: Simulation Parameter Values

In practice, buyers have no control over these parameters except the deadline, which can be decided by the user or constrained by a higher-level goal the buyer is trying to achieve.

Hypothesis B: The *ANEGMA* buyer outperforms SL, CONAN, and Williams' negotiation strategies in terms of U_{avg}, T_{avg} and $S_{\%}$ in a range of e-market settings.

Hypothesis C: An *ANEGMA* buyer if trained against a specific seller strategy, still performs well against other unknown seller strategies. This shows that the *ANEGMA* agent behaviour is *adaptive* in that the agent transfers knowledge from previous experience to unknown e-market settings.

5.1 Design of the Experiments

To carry out our experiments, we have extended the simulation environment RECON [Alrayes *et al.*, 2016] with a new online learning component for *ANEGMA*.

Seller Strategies

For the purpose of training our SL model and conducting large-scale quantitative evaluations, we have used two groups of fixed seller strategies developed by Faratin *et al.* [1998]: Time-Dependent (*Linear*, *Conceder* and *Boulware*) and Behaviour-Dependent (*Relative tit-for-tat*, *Random Absolute tit-for-tat* and *Averaged tit-for-tat*). During experimentation, the same private deadlines were used for both sellers and buyer. Other parameters such as IP_s and RP_s are determined by the ZoA parameter, as shown in Table 2.

Simulation Parameters

We assume that the buyer negotiates with multiple sellers concurrently to buy a second-hand laptop ($r = Laptop$) based only on a single issue *Price* ($I = \{Price\}$). We stress that the single-issue assumption is not unrealistic for e-markets like e-Bay, where sellers advertise a product with a fixed set of issues (e.g. Lenovo, 16GB RAM, 250GB HDD, i7 processor) and the only issue being negotiated is price. The simulated market allows the agents to enter and leave the market at their own will. The maximum number of agents allowed in the market, the demand/supply ratio, the buyer's deadline and the ZoAs are simulation-dependent.

As in [Alrayes *et al.*, 2018], three qualitative values are considered for each parameter during simulations, e.g., High (H), Average (A) and Low (L) for MD or Long (Lg), Average (A) and Short (Sh) for t_{end}. Parameters are reported in

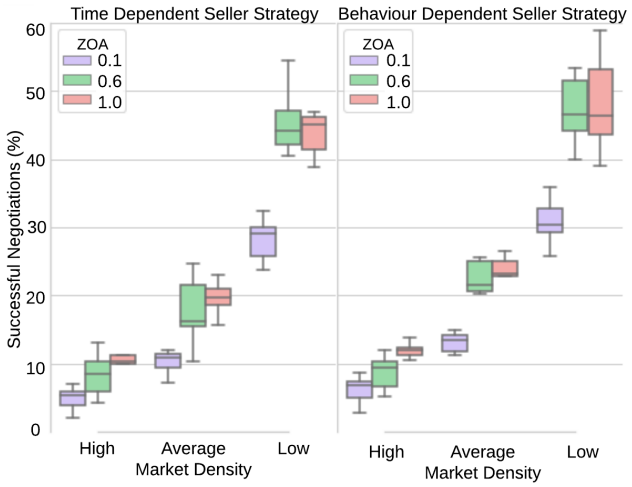

Figure 3: Effect of Market Density and Zone of Agreement on Proportion of Successful Negotiations using time-dependent (left) and behaviour-dependent (right) strategies.

Table 2. The user can select one of such qualitative values for each parameter. Each qualitative value corresponds to a set of three quantitative values, of which only one is chosen at random for each simulation (e.g., setting H for parameter MD corresponds to choosing at random among 30, 40, and 50). The only exception is parameter ZoA, which maps to a range of uniformly distributed quantitative values for the seller's initial price IP_s and reservation price RP_s (e.g., selecting A for ZoA leads to a value of IP_s uniformly sampled in the interval $[580, 630]$). Therefore, the total number of simulation settings is 81, as we consider 3 possible settings for each of MD, MR, t_{end}, and ZoA (see Table 2).

5.2 Empirical Evaluation

Hypothesis A ***(MD, MR, ZoA and t_{end} Have Significant Impact on Negotiations)***

We experimented with 81 different e-market settings over 500 simulations using the CONAN buyer strategy. Both time-dependent and behaviour-dependent seller strategies were considered for each setting. These experiments suggest that MD and ZoA have a considerable effect on $S_{\%}$ (Fig. 3). We observe that the agents reach more negotiation agreements when MD is low. Also, there is not much difference in the agreement rate for 60% and 100% ZoA when MD is low. The small number of successful negotiations for 10% ZoA is not unexpected since only a minority of agents is willing to concede more in such a small ZoA. On the other hand, MR and t_{end} have, according to our experiments, a comparably minor impact on the negotiation success (only some effect of MR on $S_{\%}$ is observed under low MD against behaviour-dependent strategies)[2]. Moreover, we performed significance tests (i.e. Z-tests for independent proportions) for all the relevant pairwise comparisons. All the differences in the proportions of successful runs were found significant at $p < 2.12E-13$.[3]

[2]See [Bagga *et al.*, 2020] for detailed results on MR.

[3]For each ZoA=H,A,L, we tested (MD=H vs MD=A) and (MD=A vs MD=L). For each MD=H,A,L, we tested (ZoA=L vs ZoA=A) and (ZoA=L vs ZoA=H).

Metric	*CONAN*	*WILLIAMS*
	Conceder Time Dependent Seller Strategy	
U_{avg}	**0.27 ± 0.03**	0.18 ± 0.08
T_{avg}	**172942.78 ± 15177.77**	177091.09 ± 15304.90
$S_{\%}$	**80.80**	78.20
	Relative Tit For Tat Behaviour Seller Strategy	
U_{avg}	**0.25 ± 0.03**	0.22 ± 0.05
T_{avg}	**176018.69 ± 14380.28**	176334.65 ± 14683.03
$S_{\%}$	**81.80**	73.00

Table 3: Performance comparison of CONAN and Williams' model. Best results are in bold.

Hence, these results support our hypothesis.

Hypothesis B ***(ANEGMA Outperforms SL and CONAN)***

We performed simulations for our *ANEGMA* agent in low MD, 60% ZoA[4], high MR and a long t_{end} because these settings yielded the best performance in terms of $S_{\%}$ in our experiments for Hypothesis A. We used these settings against *Conceder Time Dependent* and *Relative Tit for Tat Behaviour Dependent* seller strategies.

Firstly, we collected training data for ANN using two distinct strategies for supervision, viz. CONAN [Alrayes *et al.*, 2018] and Williams [Williams *et al.*, 2012]. Both were run for 500 simulations and with the same settings. Table 3 compares the performances of CONAN's and Williams' models. CONAN outperforms Williams' strategy in these settings.

Then, the resulting trained ANN models – called ANN-C and ANN-W respectively – were used as the initial strategies in our DRL approach (based on DDPG), where strategies evolved using negotiation experience from additional 500 simulations. In the remainder, we will abbreviate this model by *ANEGMA(SL+RL)*.

Finally, we used test data from 101 simulations to compare the performance of such derived *ANEGMA(SL+RL)* buyers against CONAN, Williams' model, ANN-C, ANN-W, and the so-called *ANEGMA(RL)* model, which used DDPG but initialized with a random strategy.

According to our results shown in Table 4, the performance of ANN-C is comparable to that of CONAN (see Table 3). We observe the same for ANN-W and the William's strategy. So, we conclude that our approach can successfully produce ANN strategies which are able to imitate the behaviour and performance of the CONAN and Williams' models (the training accuracies were in the range between 93.0% and 98.0%).

Even more importantly, the results demonstrate that *ANEGMA(SL+RL)-C* (i.e. DDPG initialized with ANN-C) and *ANEGMA(SL+RL)-W* (i.e. DDPG initialized with ANN-W) improve on their respective initial ANN strategies obtained by SL, and outperform *ANEGMA(RL)* initialized at random, see Table 4. This proves that both the evolution of the strategies via DRL and the initial supervision are beneficial. Furthermore, *ANEGMA(SL+RL)-C* and *ANEGMA(SL+RL)-W* also outperform the existing "teacher

[4]See [Bagga *et al.*, 2020] for more results with 100% ZOA.

Metric	*ANN*		*ANEGMA(SL+RL)*		*ANEGMA(RL)*
			Trained and Tested on Conceder Time Dependent Seller Strategy		
	ANN-C	*ANN-W*	*ANEGMA(SL+RL)-C*	*ANEGMA(SL+RL)-W*	
U_{avg}	0.27 ± 0.04	0.21 ± 0.08	**0.29 ± 0.04**	0.21 ± 0.04	-0.38 ± 0.14
T_{avg}	173529.47 ± 14651.15	171096.09 ± 14584.90	67750.62± 37628.57	132477.71 ± 26601.48	**768.55 ± 373.65**
$S_{\%}$	81.18	80.19	**87.12**	81.19	64.36
			Trained and Tested on Relative Tit for Tat Behaviour Dependent Seller Strategy		
	ANN-C	*ANN-W*	*ANEGMA(SL+RL)-C*	*ANEGMA(SL+RL)-W*	
U_{avg}	0.26 ± 0.03	0.23 ± 0.05	**0.29 ± 0.03**	0.23 ± 0.14	-0.19 ± 0.42
T_{avg}	167183.62 ± 13388.30	169334.65 ± 12389.03	36331.34 ± 70247.33	41225.17 ± 72938.79	**755.74 ± 292.29**
$S_{\%}$	82.18	75.24	**85.15**	74.26	61.38

Table 4: Performance comparison of ANN VS *ANEGMA(SL+RL)* VS *ANEGMA(RL)*. Best results are in bold.

Metric	*ANN*		*ANEGMA(SL+RL)*		*ANEGMA(RL)*
			Trained on Relative Tit for Tat Behaviour Dependent and Tested on Conceder Time Dependent Seller Strategy		
	ANN-C	*ANN-W*	*ANEGMA(SL+RL)-C*	*ANEGMA(SL+RL)-W*	
U_{avg}	0.16 ± 0.05	0.17 ± 0.04	**0.26 ± 0.06**	0.23 ± 0.07	-0.36 ± 0.12
T_{avg}	174139.30 ± 14655.42	174035.91 ± 14627.59	38402.78± 64367.45	108051.11 ± 57755.84	**738.55 ± 279.65**
$S_{\%}$	70.29	69.30	**86.13**	81.19	54.45
			Trained on Conceder Time Dependent and Tested on Relative Tit for Tat Behaviour Dependent Seller Strategy		
	ANN-C	*ANN-W*	*ANEGMA(SL+RL)-C*	*ANEGMA(SL+RL)-W*	
U_{avg}	0.25 ± 0.05	0.21 ± 0.04	**0.28 ± 0.01**	0.21 ± 0.08	-0.28 ± 0.51
T_{avg}	176048.05 ± 14423.36	175170.19 ± 14623.53	19295.84 ± 53767.54	114510.00 ± 64667.79	**806.83 ± 375.51**
$S_{\%}$	79.21	76.23	**84.16**	71.28	51.48

Table 5: Performance comparison for the adaptive behaviour of ANN VS *ANEGMA(SL+RL)* VS *ANEGMA(RL)*. Best results are in bold.

strategies" (CONAN and Williams) used for the initial supervision and hence can improve on them, see Table 3.

Hypothesis C *(ANEGMA is Adaptable)*

In this final test, we evaluate how well our *ANEGMA* agents can adapt to environments different from those used at training-time. Specifically, we deploy strategies trained using *Conceder Time Dependent* opponents into an environment with *Relative Tit for Tat Behaviour Dependent* opponents, and vice-versa. The *ANEGMA* agents use experience from 500 simulations to adapt to the new environment. Results are presented in Table 5 and show clear superiority of the *ANEGMA* agents over the ANN-C and ANN-W strategies which, without online retraining, cannot maintain their performance in the new environment. This confirms our hypothesis that *ANEGMA* agents can learn to adapt at run-time to different unknown seller strategies.

Further Discussion

Pondering over the negative average utility of *ANEGMA(RL)* (Table 4), recall that we define utility as in Equation (5) but without the discount factor. Therefore, if an agent concedes a lot to make a deal, it will collect negative utility. This is precisely what happens to the initial random (and inefficient) strategy used in *ANEGMA(RL)*. The combination of SL and DRL prevents this problem as it uses an initial pre-trained strategy which is much less likely to incur negative utility. For the same reason, we observe a consistently shorter T_{avg} for *ANEGMA(RL)* caused by a buyer that concedes more to reach the agreement without negotiating for a long time with the seller. Hence, a shorter T_{avg} alone does not generally imply a better negotiation performance. An additional advantage of our approach is that it alleviates the common limitation of RL, namely, that an RL agent needs a non-trivial amount of experience before reaching satisfactory performance.

6 Conclusions and Future Work

We have proposed *ANEGMA*, a novel agent negotiation model supporting agent learning and adaptation during concurrent bilateral negotiations for a class of e-markets. An *ANEGMA* agent derives an initial neural network strategy via supervision from well-known negotiation models, and evolves the strategy via DRL. We have empirically evaluated the performance of an *ANEGMA* buyer agent against fixed but unknown to the agent seller strategies in different e-market settings. We have shown that *ANEGMA* outperforms well-known "teacher strategies", the strategies trained with SL only and those trained with DRL only. Crucially, our model has also exhibited adaptive behaviour in that it can transfer to environments with unknown sellers, viz., sellers that use different strategies from those used during training.

In the future, we will study complex domains with bilateral negotiations on multiple issues against adaptive opponents.

Acknowledgements

We would like to thank Emanuele Uliana, Benedict Wilkins, Joel Clarke, and the anonymous reviewers for their useful suggestions on a previous version of this paper.

References

[Alrayes and Stathis, 2013] Bedour Alrayes and Kostas Stathis. An agent architecture for concurrent bilateral negotiations. In *Decision Support Systems III-Impact of Decision Support Systems for Global Environments*, pages 79–89. Springer, 2013.

[Alrayes *et al.*, 2016] Bedour Alrayes, Özgür Kafalı, and Kostas Stathis. RECON: a robust multi-agent environment for simulating concurrent negotiations. In *Recent advances in agent-based complex automated negotiation*, pages 157–174. Springer, 2016.

[Alrayes *et al.*, 2018] Bedour Alrayes, Özgür Kafalı, and Kostas Stathis. Concurrent bilateral negotiation for open e-markets: the CONAN strategy. *Knowledge and Information Systems*, 56(2):463–501, 2018.

[An *et al.*, 2006] Bo An, Kwang Mong Sim, Liang Gui Tang, Shuang Qing Li, and Dai Jie Cheng. Continuous-time negotiation mechanism for software agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(6):1261–1272, 2006.

[Bagga *et al.*, 2020] Pallavi Bagga, Nicola Paoletti, Bedour Alrayes, and Kostas Stathis. A deep reinforcement learning approach to concurrent bilateral negotiation. *arXiv preprint arXiv:2001.11785*, 2020.

[Bakker *et al.*, 2019] Jasper Bakker, Aron Hammond, Daan Bloembergen, and Tim Baarslag. RLBOA: A modular reinforcement learning framework for autonomous negotiating agents. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 260–268. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[Faratin *et al.*, 1998] Peyman Faratin, Carles Sierra, and Nick R Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[Lee Rodgers and Nicewander, 1988] Joseph Lee Rodgers and W Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[Lewis *et al.*, 2017] Mike Lewis, Denis Yarats, Yann N Dauphin, Devi Parikh, and Dhruv Batra. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*, 2017.

[Lillicrap *et al.*, 2016] Timothy Paul Lillicrap, Jonathan James Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*, 2016.

[Mansour and Kowalczyk, 2014] Khalid Mansour and Ryszard Kowalczyk. Coordinating the bidding strategy in multiissue multiobject negotiation with single and multiple providers. *IEEE transactions on cybernetics*, 45(10):2261–2272, 2014.

[Nguyen and Jennings, 2004] Thuc Duong Nguyen and Nicholas R Jennings. Coordinating multiple concurrent negotiations. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1064–1071. IEEE Computer Society, 2004.

[Oliver, 1996] Jim R Oliver. A machine-learning approach to automated negotiation and prospects for electronic commerce. *Journal of management information systems*, 13(3):83–112, 1996.

[Papangelis and Georgila, 2015] Alexandros Papangelis and Kallirroi Georgila. Reinforcement learning of multi-issue negotiation dialogue policies. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 154–158, 2015.

[Rahwan *et al.*, 2002] Iyad Rahwan, Ryszard Kowalczyk, and Ha Hai Pham. Intelligent agents for automated one-to-many e-commerce negotiation. In *Australian Computer Science Communications*, volume 24, pages 197–204. Australian Computer Society, Inc., 2002.

[Rodriguez-Fernandez *et al.*, 2019] J Rodriguez-Fernandez, T Pinto, F Silva, I Praça, Z Vale, and JM Corchado. Context aware Q-learning-based model for decision support in the negotiation of energy contracts. *International Journal of Electrical Power & Energy Systems*, 104:489–501, 2019.

[Rubinstein, 1982] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109, 1982.

[Williams *et al.*, 2012] Colin R Williams, Valentin Robu, Enrico H Gerding, and Nicholas R Jennings. Negotiating concurrently with unknown opponents in complex, real-time domains. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 834–839, 2012.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[Zou *et al.*, 2014] Yi Zou, Wenjie Zhan, and Yuan Shao. Evolution with reinforcement learning in negotiation. *PLOS one*, 9(7):e102840, 2014.