# The importance of context-dependent learning in negotiation agents

Dan Kröhling, Federico Hernández, Omar Chiotti, Ernesto Martínez

INGAR (CONICET/UTN)
Avellaneda 3657
Santa Fe, Argentina

**Abstract.** Automated negotiation between artificial agents is essential to deploy Cognitive Computing and Internet of Things. The behavior of a negotiating agent depends significantly on the influence of environmental conditions or contextual variables, since they affect not only a given agent preferences and strategies, but also those of other agents. Despite this, the existing literature on automated negotiation is scarce about how to properly account for the effect of context-relevant variables in learning and evolving strategies. In this paper, a novel context-driven representation for automated negotiation is proposed. Also, a simple negotiating agent that queries available information from its environment, internally models contextual variables, and learns how to take advantage of this knowledge by playing against himself using reinforcement learning is proposed. Through a set of episodes against negotiating agents in the existing literature, it is shown that it makes no sense to negotiate without taking context-relevant variables into account. The context-aware negotiating agent has been implemented in the GENIUS negotiation environment, and results obtained are significant and revealing.

**Keywords:** Agents, automated negotiation, negotiation intelligence, Internet of Things, reinforcement learning.

## 1 Introduction

Artificial intelligence has definitely entered the mainstream of business innovation [1, 19]. Huge progresses in the existing technology [18], new theories of intelligence [15, 20, 25], and the increasingly refined comprehension of biological minds of humans and animals [12, 24], have lead to the development of new mathematical models that tackle the problem of creating the so-called intelligent agents in our daily life. Some examples of these are [8, 10, 22].

A topic that has gained attention among AI experts in recent years is the implementation of intelligent negotiating agents. The reason behind this is that people is usually reluctant to get involved in negotiations. As Fatima et. al. [9], taken from [3], put it: "When engaged in complex negotiations, people become tired, confused, and emotional, making naive, inconsistent, and rash decisions." This is a human condition: we could see it in our everyday life [11].

To realize the promise of novel technologies such as Internet of Things and Cognitive Computing, great efforts are being made to automatize negotiations between artificial agents, although some doubts remain about the design aspects of such artificial entities. A number of approaches to address this problem have been proposed [6, 7, 14, 27]. We consider the contributions of Fatima et. al. [9] and Baarslag [4] a great compendium of the state of the art in this research area.

In spite of the progresses made so far, there is an issue in automated negotiation that, from our point of view, has not been properly accounted for in the design of negotiating agents. This is the importance of the context in negotiations, or the existence of key external variables that could provide a competitive advantage when used to predict and model the opponent by associative learning, including its strategy and perceptions/assumptions from the context. As an example, an agent could be i) informed about issues in the environment, beyond the opponent himself, and ii) hypothesize about which information the opponent is actually using to make his predictions and learn. The importance of learning in automated negotiations has been previously recognized [28, 29], yet the context-awareness capability is not widely seen as a key issue [2, 17, 26]. Most of previous works circumscribe the agent learning to ad-hoc decision-making policies that may not capture appropriately the influence of the context on the outcome of a negotiation episode. To make our point clearer, let us discuss briefly some related work. In [2, 26], the context is represented through a fixed model, but any new variable that could change the course of the negotiation is discarded. Another example is given in [17]. Although a novel approach to model the utility functions of the agents is proposed, these functions are still prefixed and they do not take into account changes in relevant contextual variables. Finally, in the GENIUS negotiation environment [13], actually one of the most used negotiation simulators and the one we also choose to run our own computational experiments, the negotiation deadline is even of public (common) knowledge, when that is certainly a decision that agents should be able to make on their own, based on their strategies and the information available to them.

Based on the above considerations, the main hypothesis in this work is that negotiating agents that learn to use in their benefit relevant contextual variables to select and evolve their strategies will reap more benefits than those agents that concentrate only on learning their opponents strategies independently of the context. Accordingly, we aim to create a negotiation environment that includes both contextual variables and context-aware agents. We design a novel context-driven negotiation setting and insert therein a learning agent that takes this context into account. This agent will use the available information alongside with reinforcement learning [21, 23] and Self-Play to generate specific knowledge about the context and select the proper actions as negotiations proceed. We will then exploit this knowledge to interact with other negotiating agents defined in the existing literature, agents that do not take into account contextual variables and yet have won the ANAC (Automated Negotiating Agents Competition) in the last years. We use the GENIUS tool [13] to run the simulated negotiation episodes and obtain significant results.

This paper is organized as follows. Firstly, a conceptual representation of the negotiation setting for context-aware negotiating agents is discussed. Next, we present "Strawberry", our own context-aware negotiating agent. We define its internal design and its Self-Play learning strategy alongside the "Oracle", a conceptual entity that is going to answer the information queries made to the context by our agent. Later on, negotiation experiments are designed and run to generate results that can test our main hypothesis.

## 2 Negotiation setting

In this section, the main structure and components that made up our representation of the negotiation setting are presented. As in most of related works, a group of agents that agree to negotiate over certain issues are considered. To highlight what is important to us, in this work we concentrate our efforts in bilateral negotiations between two agents negotiating over one single issue with discrete values using a discrete time line. The alternating offers protocol which is, according to [9], the most influential protocol of all, is used throughout.

Formally, the context in which the negotiating agent is situated is divided up in two abstract spaces: the agent's private information and external context (see Fig. 1). The agent's private information is composed by all his internal or private variables, those that other agents can not see but could attempt to model observing the actions the agent performs. The external context is composed by all the other agents in the environment and the external or public variables, those that every agent would consider if relevant.

So, for a given agent, his private information is defined as follows:

$$PI = \{X_1,\ X_2,\ ...,\ X_l\} \tag{1}$$

where $X_i$ in equation 1 is the agent's $i$th private variable.

Next, we define the agent's external context as:

$$EC = \{Opp_1,\ Opp_2,\ ...,\ Opp_m\} \cup \{Y_1,\ Y_2,\ ...,\ Y_n\} \tag{2}$$

where $Opp_j$ in equation 2 is the $j$th opponent the agent can negotiate with, and $Y_k$ is the $k$th contextual variable that the agent can query.

Finally, the agent's negotiation environment will be defined by:

$$E = PI \cup EC \tag{3}$$

## 3 Strawberry: a context-aware negotiating agent

In this section, the proposed design for our agent Strawberry is presented. A component-based architecture proposed by Baarslag in [4], which receives the name of BOA (after Bidding strategy, Opponent model, and Acceptance strategy), is appropriated enough for implementing Strawberry. However, in order to

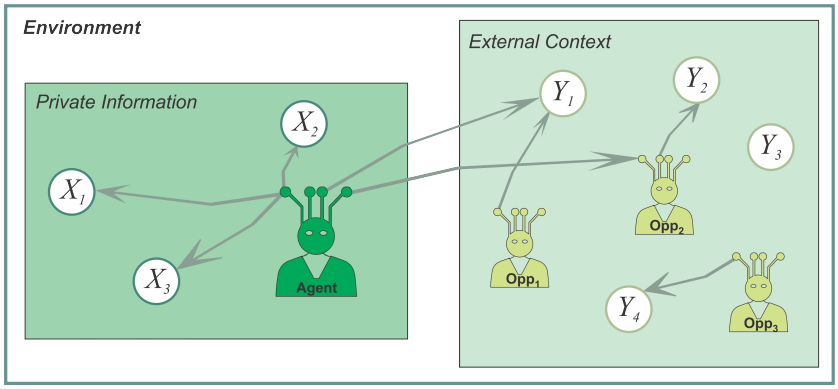


**Fig. 1.** Our negotiation environment from the perspective of a single agent.

test our main hypothesis, we incorporate to this architecture the possibility of querying the environment, and define the resulting architecture as *context-aware* BOA.

All these components and the resulting design are rather simple but will serve our purpose. As can be imagined, we could make things as complex as we want, but this view will suffice to prove our hypothesis about the role of contextual variables that are common knowledge. We profoundly believe that keeping things simple (as long as it is possible) is not only clearer, but also better.

On top of this architecture, we will use two techniques that are widely known, namely Self-Play [22] and Reinforcement Learning (or RL) [21, 23]. Strawberry will learn to better negotiate by simulating negotiation episodes against another instance of himself while using the well-known Q-Learning algorithm.

In Fig. 2[1], we present a graphical representation of our agent Strawberry and the different aspects that will be explained in the subsections below.

### 3.1 Environment model

We will begin with the description of our environment model. As we have shown in Section 2, we believe the environment can be modeled in a group of variables and agents (our opponents). All we need is to provide our Strawberry agent a way to query context relevant information as deemed necessary.

To this end, we introduce the Oracle, a conceptual entity that could get real-time information from the context variables and summarize it to our agent in two state variables, *necessity* ($\nu$) and *risk* ($\rho$), as follows:

$$\nu = max\{X_1,\ X_2,\ ...,\ X_l\} \quad ; \quad 0 \leq \nu \leq 1 \tag{4}$$

$$\rho = max\{Y_1,\ Y_2,\ ...,\ Y_n\} \quad ; \quad 0 \leq \rho \leq 1 \tag{5}$$

[1] Adapted from [5].

These variables represent the state for associative learning used by Strawberry[2].

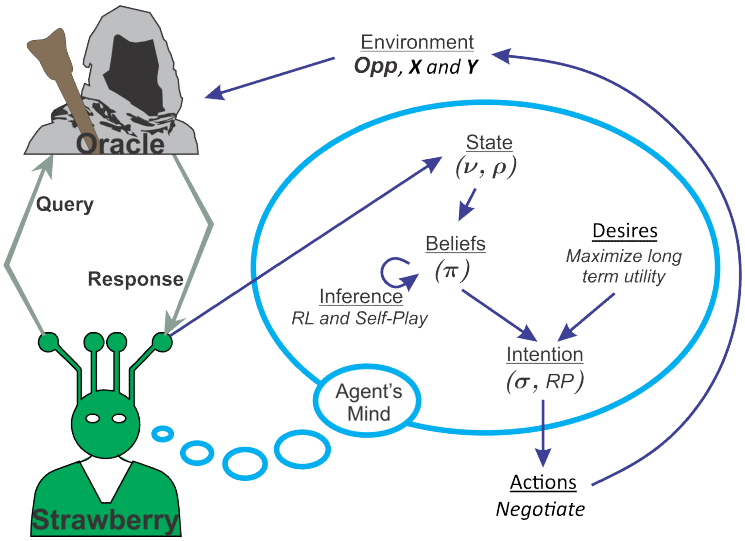


**Fig. 2.** Context-aware negotiating agent.

### 3.2 Bidding strategy

Taken from Fatima [9] and references therein, a heuristic concession strategy for Strawberry is defined as follows:

$$O_t = IP + (RP - IP) * \sigma(\beta, t, n) \tag{6}$$

where $O_t$ is the offer Strawberry will make at time $t$, $IP$ is the initial price, which is assumed to be the best deal the agent considers he can obtain from the negotiation, and $RP$ is the reserve price, which is the worst deal the agent can achieve at the end of a negotiation episode. The concession strategy $\sigma$ is based on:

$$\sigma = \left(\frac{t}{n}\right)^{1/\beta} \tag{7}$$

where $t$ is the time passed from the beginning of the negotiation, $n$ is the deadline, and $\beta$ defines the concession rate.

[2] We use a conservative approach to this end, considering that our necessities and risks are defined by the greatest necessity and the greatest risk the agent is subject to. This criterion will suffice to prove our hypothesis

### 3.3 Acceptance strategy

The acceptance strategy to be used is AC, taken from Baarslag [4], where its effectiveness was demonstrated. It could be summarized as follows: our agent will accept an offer from his opponent if and only if this offer supposes a higher utility for our agent than the utility he would obtain from his own next offer. In other words, Strawberry will accept the offer if:

$$u(O_t^{Opp}) \geq u(O_{t+1}^{Strawberry}) \tag{8}$$

where $u(O_t^{agent})$ is the long-term utility Strawberry will obtain from the offer $O$ made by the *agent* at time $t$.

### 3.4 Self-Play and Reinforcement Learning

Strawberry will use his self-play capacity to acquire some knowledge of his context through the model he makes out of it. To this end, he will play with another instance of himself, adapting his policy $\pi$ as he plays all the tricks he has under his sleeve in order to get better and better (or so we hope).

Strawberry's final desire is not only to maximize his next possible reward $r$ but also to maximize his long term utility $R$, as stated in Fig. 2. This learning strategy is implemented by the Q-Learning algorithm, which consists of a function that iterates over the expected cumulative rewards for future time steps in a negotiation episode (how many will depend on the tunning of the algorithm hyper-parameters) given the actual state $s_t$ of the environment providing that the agent takes a certain action $a$ from the set of possible actions $A$. The action to take is determined by a policy $\pi$ derived from the $Q$-values, which are the way this algorithm represents the immediate and long-term utility for every state-action pair. At the end of each episode, Strawberry observes the immediate reward $r$ and the next state $s_{t+1}$ that the environment returns, and obtains the Q-value from the best action the agent can take in that situation (indicated by $max_a\ Q(s_{t+1}, a)$ in equation 9). Then, the algorithm actualizes $Q$ according to:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\ \left[r_{t+1} + \gamma\ max_a\ Q(s_{t+1}, a) - Q(s_t, a_t)\right] \tag{9}$$

We choose each state for Strawberry to be defined by the tuple:

$$s = (\nu,\ \rho) \tag{10}$$

where $\nu$ and $\rho$ are the necessities and risks associated with the perceived state of the environment, as mentioned earlier, and each action by:

$$a = (RP,\ \beta) \tag{11}$$

where $RP$ and $\beta$ are the variables our agent will choose to vary his strategy.

A group of hyper-parameters need to be set in this algorithm to work; these are $\epsilon$, $\alpha$ and $\gamma$. $\epsilon$ defines the greediness of our agent, that is, the probability our agent takes an exploratory move (normally, a random move) rather than

exploiting his knowledge (following the actual policy). $\alpha$ is the learning rate parameter, which gives more weight to recent rewards than to past rewards. $\gamma$ is the discount rate, with which the agent will try to maximize the sum of the discounted rewards he receives in the future. We could say we have past, present and future in $\alpha$, $\epsilon$, and $\gamma$, respectively.

## 4 GENIUS

As has been said before, the GENIUS simulation software will be used to test our main hypothesis regarding the key role of accounting for contextual variables while learning to negotiate. GENIUS is a specialized non-commercial environment for simulating negotiations, where a given agent design can be implemented and then faced against a set of previously available agents. The initial intention of GENIUS was to prove that negotiating agents can be constructed using three basic components: Bidding strategy, Opponent model, and Acceptance strategy [4]. This component-based architecture receives the name of BOA.

GENIUS offers their users the possibility of creating negotiation domains, with a variety of issues of discrete or integer nature. Then, agents profiles are created, in which a set of preferences over the different issues is established, which the simulator uses to compute outcomes at the time of the negotiation.

Finally, one can run single negotiations or tournaments, choosing agents from the repository, creating new ones out of a group of available components, or codifying one from scratch in Java within the GENIUS framework. Negotiations are based on deadlines that are of common knowledge, and the GENIUS present the results in a table and a chart, where the Pareto frontier, the Nash equilibrium, and other social welfare measures could be seen.

The GENIUS simulator is a practical tool to try new agents in the field of automated negotiations. Nevertheless, from our point of view, the context and agent's profiles implementation are rather simple compared to a real-life negotiation setting. As we would like to prove that agents should take the context into account so as to make more rational decisions, we then implement the negotiation setting represented in Section 2 and include it in GENIUS. We will use this new concepts in the next sections to define and develop the experiments.

## 5 Experiments

In this section, we will describe the experimental setting, how negotiation simulations were run, and the results obtained after negotiation episodes.

In the first phase, we had to make some changes and adds to GENIUS. New features were developed in Java, using the IntelliJ IDEA environment, a package that gave us the possibility to adapt GENIUS to our needs. Finally, we developed our agent, Strawberry, with the capability to query contextual variables to the Oracle mentioned in section 3.

### 5.1 Setting

The first step to address our hypothesis was to create Strawberry's private and contextual variables. Without any loss of generality, only two variables in each space ($X_1$, $X_2$, $Y_1$, and $Y_2$), that could take random integer values between 0 and 3 are considered, and then normalized to 1. These variables will later be accessed by querying the Oracle, summarized in $\nu$ and $\rho$, and sent to Strawberry when required.

The second step was to select a domain in GENIUS that could gave us simple but revealing results, bearing in mind that the focus here are single-issue negotiations. The domain selected was the "pie domain", a problem usually addressed in game theory[16] and also available in GENIUS, in which two agents negotiate over a pie that is divided into a number of pieces (we choose this number to be a thousand). The aim in this domain is to get as many pieces as you can, taking into account that, if the deadline is reached without a deal, every agent get zero pieces. The utility is given by how many pieces an agent gets by the end of the negotiation episode divided by 1000.

The third step was to define the concrete aspects of the $Q$-Learning algorithm. We designed a reward function upon which the environment would give Strawberry a reward $r$ at the end of each negotiation episode, depending on the outcome of the negotiation and the environment state $s$ at which the negotiation takes place. This reward function is defined as follows:

– If the negotiation ended successfully:

$$r = u(O_{t=end}) \tag{12}$$

that is, the utility that the agent obtains from the last offer reported.

– If the negotiation ended unsuccessfully:

$$r = \begin{cases} -1 & \text{if } X_1 = 3 \\ -1/3 & \text{if } X_2 = 3 \\ -2/3 & \text{if } Y_1 = 3 \ \wedge \ Y_2 = 0 \\ 1/3 & \text{if } X_1 \leq 1 \\ 2/3 & \text{if } X_1 \leq 1 \ \wedge \ X_2 \leq 1 \end{cases} \tag{13}$$

The rationale behind this function is that the agent would be not only concerned by the result of the negotiation, but also by the perceived state of the context and how it affects him.

A number of hyper-parameters had to be set in order to make Strawberry capable of learning from reinforcements. As a common rule of thumb, $\alpha$ and $\epsilon$ are usually set to 0.1 [21], and $\gamma$ to 0.9, values that contribute to a fast learning by means of a reasonable exploration-exploitation trade-off. These are typical default values in the bibliography.

Finally, we set three alternative values for $\nu$ and $\rho$ corresponding to the intervals $[0; 0,33]$, $(0,33; 0,66]$, and $(0,66; 1]$. We also define the actions allowed

to Strawberry. We set three different values for $\beta$: 0.5, 1.0, and 2.0, and five possible values for $RP$: 0.0, 0.2, 0.4, 0.6, and 0.8, which provide Strawberry with 3 * 5 = 15 different concession rates. Summing up, we will have a maximum of 3 * 3 * 3 * 5 = 135 Q-values to learn.

### 5.2 Experimental design

The experiments were made in the pie domain, where a deadline of 180 rounds was used for simulating the negotiation episodes, a value that GENIUS uses as initial. The experiments were divide up in three phases: the learning phase, the negotiation phase, and the Self-Play improvement phase.

In the learning phase, the Strawberry agent is set to negotiate against himself using the tournament setting that GENIUS provides. Self-Play simulations were run for 10, 100, 500, 1000, 2000, 5000, and 10000 negotiation episodes in order to see how much learning may affect the subsequent negotiation phase.

In the negotiation phase, tournaments are played against some of the existent negotiating agents in GENIUS. We have chosen some simple ones and others really difficult to beat, winners of previous ANAC competitions. The chosen opponent types are:

- Random Party (RP)
- BoulwareNegotiationParty (B)
- ConcederNegotiationParty (C)
- CUHKAgent2015 (CUHK)
- AgentFSEGA (FSEGA)
- Agent_K (K)
- IAMcrazyHaggler (Haggler)
- AgentSmith (Smith)
- Gahboninho (G)
- BRAMAgent (BRAMA)

Simulations were run in two different settings: Strawberry against all other agents together, and Strawberry against each one of them, separately. Each tournament consisted of 100 negotiations were Strawberry used the knowledge previously gained through Self-Play, but did not learn whilst negotiating with the other opponents.

Finally, the Strawberry agent is put to negotiate against himself, but this time without doing any learning, in order to see if he achieves better agreements in Self-Play compared to the different learning episodes he has previously done.

### 5.3 Results and analysis

After the learning phase, the negotiation phase has given us some interesting results that are depicted in Fig. 3. At first glimpse, the cumulative utility of Strawberry against the average opponent starts below 300 and reaches 400 as he learns about the environment. This behavior was rather expected since the reinforcement learning method resorts to the association of the goodness of an action to the value of contextual variables. On the other hand, it is also possible to see that taking the environment into account could change the perspective of negotiation itself: Strawberry had received, in average, better outcomes than his negotiation counterparts when considering the context of the negotiation. This

reasoning tempts us to say that the environment should be taken into account so as to gain a competitive advantage.

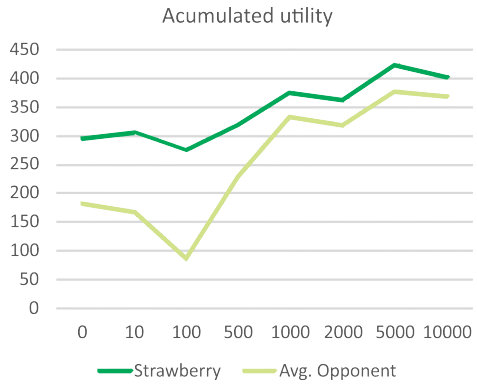


**Fig. 3.** Strawberry's accumulated utility over 100 tournament scenario against the average opponent, that is to say, the average utility the different opponents mentioned in Section 5.2 have obtained. The horizontal axis shows how many learning negotiations with Self Play has Strawberry made. The vertical axis shows the accumulated utility throughout the 100 tournament negotiation session.

Another important observation to make from Fig. 3. If some agent considers environmental variables, there will be an increase in the rest of the agents accumulated utilities, not only in his own. This astonishing result gives rise to two different theories. The first one is that it could be possible that if one of the agents takes some variables into account that the other does not, better agreements are reached, with a tendency to improve social welfare. The second theory, the one we think could explain better this phenomena, is that, as Strawberry learns, he makes more rational decisions and does not take so many actions at random. In this context, the other agents could build a better model out of him and predict better what his moves are going to be. In other words, they model the part of the environment they do not see through the model they made of Strawberry's behavior. We think this is one of the key aspects we have discovered through our research.

In Fig. 4, we can see the utilities obtained by Strawberry against each particular agent, and the utilities obtained by his opponents, which do not take environmental variables into account. Again, as expected, the utilities obtained by Strawberry are always better when the rewards of the environment are considered than when they are not (the rewards that GENIUS gives). Another thing we could see is that as agents get more complex (e.g., with opponent models, environment model and flexible strategies, etc.), they make it more difficult for Strawberry to get a good deal. Particularly, the CUHK agent seems to behave really tough, not letting much to Strawberry, but still getting a great deal of

accumulated utility. It is worth asking ourselves what would achieve an agent like this if it were resorting to model the environment as well.

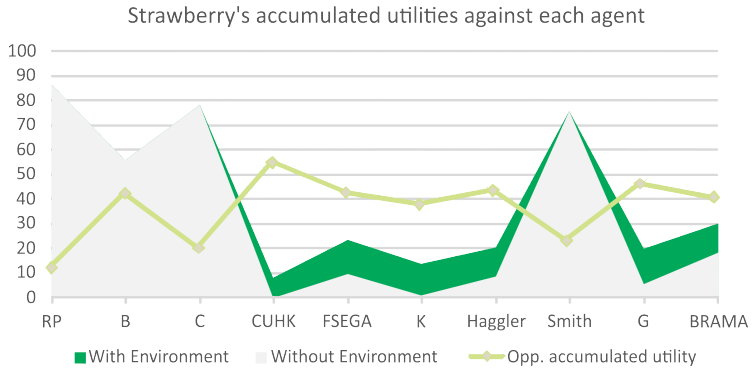


**Fig. 4.** Accumulated utility obtained by Strawberry after 10000 learning negotiation sessions in Self Play against each particular agent, when it is considered the reward from the environment and when it is not, and the utilities obtained by his opponents.

In the Self-Play improvement phase, we have reached other conclusions. As can be seen in Fig. 5, we see how Strawberry achieves better and better agreements as he learns associatively considering the context of the negotiation, thus increasingly maximizing the social welfare over negotiations. Also, in the graph on the right side, we see how Strawberry gets more successful negotiations as he learns. Fig. 4 and 5 vividly highlight the importance of contextual-learning in negotiations, as can be expected, following the previous results shown in Fig. 3.

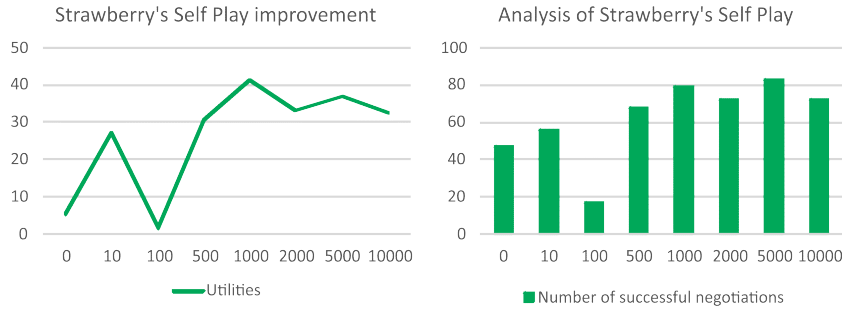


**Fig. 5.** The graphic on the right shows Strawberry's Self Play improvement in accumulated utility as he learns. In the left graphic, we see how many successful negotiations reaches Strawberry's when making 100 negotiation sessions against himself.

A final comment can be made about reaching an equilibrium. We have shown that Strawberry gets better through Self-Play and learning. However, he does not tend to reach the Nash equilibrium that GENIUS proposes, as can be seen in Fig. 6, when playing against himself. In fact, there are no great changes in the mean distance to the Nash point. It may be argued that this fact is because GENIUS does not consider the contextual variables to calculate the Nash equilibrium, but we do consider the competitive edge of a negotiating agent during the learning phase. If we see the mean distance to the Nash equilibrium as the learning advances, mainly after the 500 tournaments learning, there is a difference of approximately 0,3, which shows that the Nash equilibrium is not situated where the GENIUS locates it. In our favor, it could be stated that is not possible to find the real Nash equilibrium of a negotiation game unless the relevant contextual variables that affect all agents' utilities are taken into account.

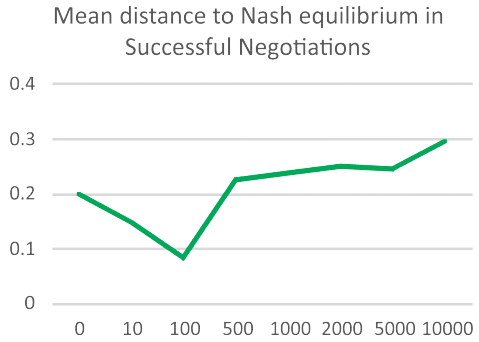


**Fig. 6.** This graphics presents in the vertical axis the mean distance to the Nash equilibrium proposed by GENIUS in 100 Self-Play episodes once learning has ended. The horizontal axis shows the number of learning episodes previously made by Strawberry.

## 6 Concluding remarks

The importance of key contextual variables have when two agents are negotiating over a certain issue has been addressed. A novel way of modeling the negotiation setting based on characterizing both the agent's private variables, which consists of the agent's strategies and preferences, and the external context, where, besides other negotiating agents, a group of external variables that influence the utilities and values of concerned agents, are used to learn negotiation strategies.

The proposed Strawberry agent is a situated agent that resorts to contextual variables to take some advantage. We have presented the way our agent account for contextual variables, based on which his bidding and acceptance strategies are built up. Then, we have explained how Strawberry would learn negotiation knowledge using the Q-Learning algorithm and Self-Play.

Computational experiments were designed to assess the validity of our central hypothesis: the advantage of including contextual variables in a negotiating agent. Results obtained confirm our earlier thoughts whereas other results are rather unexpected. Strawberry is quite competitive in a heterogeneous environment composed of a number of agents, even though he makes no explicit model of his opponents. We have proven that the utilities agents perceive are different whether we take or not the variables of the external context and the agent's private variables into account. These results sustain our main hypothesis, but more learning experiments are needed. We have seen, along with Strawberry's improvement, his opponent improvements as he learns. We theorize that the models other agents make out of Strawberry help them discover implicitly the external variables Strawberry takes into account, although they do not know of their existence. It can also be stated, from the results shown in Fig. 4, that the Strawberry agent reap higher utilities when he takes the external variables into account. The importance of Self-Play for cheap Learning is highlighted through results obtained. Hence, social welfare can be increased as agents learns collectively through inexpensive simulation-based Self-Play learning.

As a final word, it can be said that our hypothesis seems correct from the point of view of the Nash equilibrium. If we take the external variables into account, it makes no sense to find an equilibrium between the strategies of the negotiating agents considered in isolation. If contextual variables are not perceived by any of the agents, then they would attempt to reach an equilibrium that is nonexistent. Our agent Strawberry, simple as it is, when playing against himself shows us that the equilibrium is somewhere else, not just "in the middle". In other words, should we assume that, when two people claim for a piece of pie, the whole is to be always divided exactly in two? We thing we should not, and the results support our earlier thoughts that this division does not only depend on the agents and the pie itself, but also on external variables agents should not leave aside.

## References

1. A. Agrawal, J. Gans, and A. Goldfarb. *Prediction machines: The simple economics of artificial intelligence*. Harvard Business Review Press, Boston, Massachusetts, 2018.
2. B. Alrayes, Ö. Kafalı, and K. Stathis. Concurrent bilateral negotiation for open e-markets: The conan strategy. *Knowledge and Information Systems*, 2017.
3. D. Ariely. *Predictably irrational: The hidden forces that shape our decisions*. Harper Perennial, New York, revised and expanded ed. edition, 2010.
4. T. Baarslag. *Exploring the strategy space of negotiating agents: A framework for bidding, learning and accepting in automated negotiation*. Springer theses. Springer, Switzerland, 2016.
5. C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.
6. K. Cao, A. Lazaridou, M. Lanctot, J. Z. Leibo, K. Tuyls, and S. Clark. Emergent communication through negotiation. 2018.

7. N. Criado Pacheco, C. Carrascosa, N. Osman, and V. Julián Inglada. *Multi-Agent Systems and Agreement Technologies*, volume 10207. Springer International Publishing, Cham, 2017.
8. Daisuke Wakabayashi. Waymo's autonomous cars cut out human drivers in road tests, 2017.
9. S. Fatima, S. Kraus, and M. Wooldridge. *Principles of Automated Negotiation*. Cambridge University Press, 2014.
10. D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller. Watson: Beyond jeopardy! *Artificial Intelligence*, 199-200:93–105, 2013.
11. R. Fisher and W. Ury. *Sí, ¡de acuerdo! Como negociar sin ceder*. Libros universitarios y profesionales. Serie Norma de desarrollo gerencial. Norma, [Colombia], 1985.
12. P. W. Glimcher, editor. *Neuroeconomics: Decision making and the brain*. Academic Press, London and San Diego, CA, 1st ed. edition, 2009.
13. R. Lin, S. Kraus, T. Baarslag, D. Tykhonov, K. Hindriks, and C. M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1):48–70, 2014.
14. A. Monteserin and A. Amandi. Argumentation–based negotiation planning for autonomous agents. *Decision Support Systems*, 51(3):532–548, 2011.
15. Philip Ball. How life (and death) spring from disorder. *Quanta Magazine*, 2017.
16. A. D. Procaccia. Cake cutting: Not just child ' s play. *Communications of the ACM*, 56(7):78–87, 2013.
17. F. Ren and M. Zhang. A single issue negotiation model for agents bargaining in dynamic electronic markets. *Decision Support Systems*, 60(1):55–67, 2014.
18. S. Russell. Artificial intelligence: The future is superintelligent. *Nature*, 548(7669):520–521, 2017.
19. A. Sathi. *Cognitive (internet of) things: Collaboration to optimize action*. Nature America Inc, New York NY, 1st edition edition, 2016.
20. Shane Legg. *Machine Super Intelligence: PhD Thesis*. 2008.
21. R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Adaptive computation and machine learning. MIT Press, Cambridge Mass., 1998.
22. G. Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.
23. C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.
24. H. A. d. Weerd and R. Verbrugge. *If you know what I mean: Agent-based models for understanding the function of higher-order theory of mind*. University of Groningen and [Rijksuniversiteit Groningen] [host], [Groningen] and [Groningen], 2015.
25. A. D. Wissner-Gross and C. E. Freer. Causal entropic forces. *Physical Review Letters*, 110(16):1–5, 2013.
26. G. Yang, Y. Chen, and J. P. Huang. The highly intelligent virtual agents for modeling financial markets. *Physica A: Statistical Mechanics and its Applications*, 443:98–108, 2016.
27. F. Zafari and F. Nassiri-Mofakham. Popponent: Highly accurate, individually and socially efficient opponent preference model in bilateral multi issue negotiations. *IJCAI International Joint Conference on Artificial Intelligence*, (April):5100–5104, 2017.
28. D. D. Zeng. Benefits of learning in negotiation. (October 2014), 2001.
29. Y. Zou, W. Zhan, and Y. Shao. Evolution with reinforcement learning in negotiation. *PLoS ONE*, 9(7), 2014.