Learning to Play a Satisfaction Equilibrium

Stéphane Ross & Brahim Chaib-draa
Computer Science Department
PLT Bdg, Laval University
Québec, PQ, Canada
{ross;chaib}@damas.ift.ulaval.ca

Abstract

In real life problems, agents are generally faced with situations where they only have partial or no knowledge about their environment and the other agents evolving in it. In this case all an agent can do is reasoning about its own payoffs and it cannot rely on the classical equilibria through deliberation. To palliate to this difficulty, we introduce the satisfaction principle from which an equilibrium can arise as the result of the agents individual learning experiences. We define such an equilibrium and then we present different algorithms that can be used to reach it. Finally, we present experimental results and theoretical proofs that show that using learning strategies based on this specific equilibrium, agents will generally coordinate themselves on a Pareto-optimal joint strategy, that is not always a Nash equilibrium, even though each agent is individually rational, in the sense that they try to maximize their own satisfaction.

1 Introduction

"Even if all members of a group would benefit if all cooperate, individual self-interest may not favor cooperation. The prisoner's dilemma codifies this problem and has been the subject of much research, both theoretical and experimental. Results from experimental economics show that humans often act more cooperatively than strict self-interest would seem to dictate. One reason for this may be that if the prisoner's dilemma situation is repeated, it allows non-cooperation to be punished more, and cooperation to be rewarded more, than the single-shot version of the problem would suggest..."; Quoted from [Wikepedia, 2006]. This paper takes this road by considering situations where agents have partial or no knowledge about their environment and the other agents evolving in it.

Game theory provides a general framework for decision making in multi-agent environments, though, general game models assume full knowledge and observability of the rewards and actions of the other agents. In real life problems, however, this is a strong assumption that does not hold in most cases.

A first game model proposed by [Harsanyi, 1967] considering incomplete information are Bayesian games. Theses games allow the modeling of unknown information as different agent types and a Nature's move that selects randomly each agent's type according to some probability distribution before each play. The agent must choose the action that maximizes its reward considering the probabilities it associates to each of the other agents' types and the probabilities it associates to the actions of the other agents when they are of a certain type. However, the concept of Nash equilibrium in these games can be troublesome if not all agents have the same common beliefs about the probability distribution of all agents' types. Furthermore, a Nash equilibrium requires that each agent knows the action-space and the corresponding set of strategies of the other agents, which is not always the case when an agent faces unknown agents.

Another recent approach based on Bayesian games is the theory of learning in games which relaxes the concept of equilibrium. Instead of considering an equilibrium as the result of a deliberation process, they consider an equilibrium to be the result of a learning process, over repeated play, and they define the concept of self-confirming equilibrium [Dekel *et al.*, 2004] as a state in which each agent plays optimally considering their beliefs and history of observations about the other agents' strategies and types. However, they showed that if an agent does not observe the other agents' actions, then the set of Nash equilibria and self-confirming equilibria may differ. While self-confirming equilibrium is a very interesting concept and worth consideration, we note that when an agent faces unknown agents and does not observe the other agents' actions, thinking rationally on possibly false beliefs may after all, not be optimal.

In order to address this problem, we will also consider that an equilibrium is the result of a learning process, over repeated play, but we will differ in the sense that we will pursue an equilibrium that arises as the result of a learning mechanism, instead of rational thinking on the agent's beliefs and observations. To make this equilibrium possible, we introduce the satisfaction principle, which stipulate that an agent that has been satisfied by its payoff will not change its strategy, while an unsatisfied agent may decide to change its strategy. Under this principle, an equilibrium will arise when all agents will be satisfied by their payoff, since no agent will have any reason to change their strategy. From now on, we will refer to this equilibrium as a satisfaction equilibrium.

We will show that if the agents have well defined satisfaction constraints, Pareto-optimal joint strategies that are not

Nash equilibria are satisfaction equilibria and that henceforth, cooperation and more optimal results can be achieved using this principle, instead of rational thinking.

In this article, we will first introduce the game model we will use to take into account the constrained observability of the other agents' actions and rewards and we will also present the different concepts we will need to analyze a game in terms of satisfaction. Afterward, we will present different algorithms that converge towards satisfaction equilibria with experimental results showing their strengths and drawbacks in some specific games. Finally, we will conclude with future directions that can be explored in order to achieve better results.

2 Related Work

The idea of investigating the equilibrium selection via learning is not new. Thus, Claus and Boutillier [2000] have addressed this problem through on-policy learning. The underlying principle is that each agent's individual utilities shift to reflect the frequency with which the agent has achieved a desirable reward, causing the agents to settle towards complementary policies in which both agents benefit. Unfortunately, agents using this technique do not always settle to an optimal equilibrium (Pareto-optimal Nash Equilibrium). In the same vein, Lauer [2000] have proposed a simple reinforcement learning technique based on a pre-arranged coordination mechanism, in which the agents retain as their optimal policy the first action that successfully maximized reward. While this approach always find an optimal policy in deterministic cooperative games, its greedy approach makes it unable to converge toward a more cooperative and Pareto efficient solution in the case of the prisoner's dilemma (PD) for instance. In the same context, kapetanaki and colleagues [2002] have used biased exploration technique to favor convergence to an optimal equilibrium in cooperatives games with stochastic rewards. To this end, they introduced a FQM (Frequency Maximum Q value) heuristic based on the maximum reward received for a given action and the frequency with which that reward has been observed. By doing so, this approach increases the likelihood of convergence to an optimal equilibrium in stochastic cooperative games, but does not guarantee it. As the previous approach of Kapetanaki et al. [2002] this approach also tries to converge toward a Nash equilibria and consequently it cannot find Pareto-efficient solutions in games as the PD.

The most directly related work to our approach is Fulda's approach [Fulda and Vantura, 2004] where authors present an equilibrium selection algorithm for reinforcement learning agents that incrementally adjusts the probability of executing each action based on the desirability of the outcome obtained in the last time step. In deterministic environments with one or more coordination equilibria, the approach learns to play a coordination equilibria[1]. The authors showed with empirical data, that this approach is also effective in stochastic environments. In the context of other games than coordinating games [Stimpson *et al.*, 2001] have adopted the algorithm and notation presented in [Karandikar and Mookherjee, 1998] to a satisfying and learning cooperation in the prisoner's dilemma (PD) and a n-player version of this problem called the multi-agent social dilemma (MASD). Those authors described a satisfycing learning strategy for the PD and MASD and present evidence that stable outcomes other than the Nash equilibria are possible. While this approach gave good performances on the PD and MASD problems, it is not guaranteed to converge and has not been extended and studied for other games. Here, we pursue in the same vein by extending the concept to address general games (not only the PD or the coordinating games) under imperfect private monitoring.

[1]A coordination equilibria maximizes payoff of all players, but does not always exists

2.1 Game Model

In this paper, the considered game model is a simplified repeated Bayesian game in which Nature (i) fixes, once and for all, the types of all agents at the beginning of the learning process, such that only the joint action of the agents affect their payoffs and; (ii) distributes the individual outcomes to each agent according to observation functions that take in parameter the joint action and returns the outcome observed by the agent. We also used a modified reward function that takes an outcome and returns its associated reward. This has been done in order to let the agents observe their own rewards but not the other agents' actions.

Formally, we define the game as a tuple $(n, A, \Omega, O_1, O_2, \ldots, O_n, R_1, R_2, \ldots, R_n)$ where n defines the number of agents, A defines the joint action space of all agents, e.g. $A = A_1 \times A_2 \times \ldots \times A_n$ where A_i represents the set of actions agent i can do, Ω is the joint outcome space,$\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_n$ where Ω_i is the set of possible outcomes for agent i, O_i the observation function $O_i : A \rightarrow \Omega_i$ which returns the observed outcome for agent i associated to the joint action of all agents and finally R_i the reward function $R_i : \Omega_i \rightarrow \mathbf{R}$ of agent i. Each agent participating in the game only knows its own action set A_i and reward function R_i. To compute its reward, agent i is given the outcome of the game $o \in \Omega_i$ corresponding to the joint action of the agents. However, since the agents do not know their observation function O_i, they do not know which joint action led to this outcome.

2.2 Satisfaction function and equilibrium

To introduce the satisfaction principle in the game model previously introduced, we add a satisfaction function $S_i : \mathbf{R} \rightarrow \{0, 1\}$ for each agent i, that returns 1 if the agent is satisfied and 0 if the agent is not satisfied. Generally, we can define this function as follows:

$$S_i(r_i) = \begin{cases} 0 & \text{if } r_i < \sigma_i \\ 1 & \text{if } r_i \geq \sigma_i \end{cases}$$

where σ_i is the satisfaction threshold of agent i representing the threshold at which the agent becomes satisfied, and r_i is a scalar that represents its reward.

Definition 1. *An outcome o is a satisfaction equilibrium if all agents are satisfied by their payoff under their satisfaction function and do not change their strategy when they are satisfied.*

$$\begin{array}{lll} (i) & S_i(R_i(o)) = 1 & \forall i \\ (ii) & s_i^{t+1} = s_i^t & \forall i, t : S_i(R_i(o_t)) = 1 \end{array}$$

s_i^{t+1} defines the strategy of agent i at time $t+1$, s_i^t its strategy at time t and o_t the outcome observed at time t. Condition (i) states that all agents must be satisfied by the outcome o, and condition (ii) states that the strategy of an agent i at time $t+1$ must not change if it was satisfied at time t. This is necessary in order to have an equilibrium.

We can now represent a satisfaction matrix by transforming a normal form game matrix with the satisfaction function of each agents. For example, the figure 1 shows the prisoner's dilemma game matrix with its transformed satisfaction matrix when both agents have a satisfaction threshold set to -1.

	C	D
C	-1,-1	-10,0
D	0,-10	-8,-8

$\Longrightarrow$

	C	D
C	1,1	0,1
D	1,0	0,0

$(\forall i | : \sigma_i = -1)$

Figure 1: Prisoner's dilemma game matrix (left) and satisfaction matrix (right).

We can easily see that in this example, the only satisfaction equilibrium is the joint strategy (C, C). While some might argue that in this example the equilibrium would be weak, since row agent can play D and still be satisfied if column agent continues to play C, we recall that if an agent is satisfied, it would never change its strategy (see definition 1), and therefore, a satisfaction equilibrium is always a strong equilibrium. If row agent would decide to change its strategy to D, then column agent would not be satisfied by the outcome (D, C), which would eventually result in a change of strategy to D by column agent. The end result is that the initial strategy change by row agent eventually result in an unsatisfying outcome (D, D), which would eventually make it go back to C. This supports the fact that when agents are satisfied, they should not change their strategy.

As a second remark, if we look back at figure 1, we can observe that the satisfaction equilibrium (C, C) is a Pareto-optimal strategy of the original game. This was the case in this example because we set both satisfaction thresholds to -1, which was the reward of the Pareto-optimal joint strategy of each agents. From this, we can conclude theorem 2.

Theorem 2. *In any game the outcome $O(s)$ such as s is the Pareto-optimal joint strategy; and its equivalent outcomes*[2] *are the only satisfaction equilibria if $\sigma_i = R_i(O(s))\ \forall i$.*

Proof. Let $G = (n, A, \Omega, O, R_1, \ldots, R_n)$ be a game such that it contains a Pareto-optimal joint strategy s. Now, let's define the satisfaction threshold of all agents such that : $\sigma_i = R_i(O(s))\ \forall i$. To prove that $O(s)$ and its equivalent outcomes are the only satisfaction equilibria in game G, we will show that $O(s)$ is a satisfaction equilibria and that for all joint strategies s' where $\neg(O(s') \equiv O(s))$, $O(s')$ is not a satisfaction equilibrium.

First let's show that $O(s)$ is a satisfaction equilibrium. We have that for all agents i, $R_i(O(s)) = \sigma_i$. By definition of the satisfaction function, we conclude that $S_i(R_i(O(s))) = 1\ \forall i$. Therefore, the outcome $O(s)$ is a satisfaction equilibrium. Similarly, all equivalent outcomes will be satisfaction equilibrium since if $R_i(O(s)) = R_i(O(s'))\ \forall i$, then $S_i(R_i(O(s'))) = 1\ \forall i$ and we have that $O(s')$ is a satisfaction equilibrium.

To show that all other outcomes $O(s')$ such that $\neg(O(s') \equiv O(s))$ are not satisfaction equilibrium, we will consider 2 cases: either strategy s Pareto-dominates s' or either strategy s' is not Pareto-comparable to s. These are the only 2 cases we must consider since $\neg(O(s') \equiv O(s))$ and s is Pareto-optimal.

Case 1 : Strategy s Pareto-dominates strategy s'.

Since s Pareto-dominates s', there exists an agent i such that $R_i(O(s')) < R_i(O(s))$. Therefore, since $\sigma_i = R_i(O(s))$, we conclude that $S_i(R_i(O(s'))) = 0$. Consequently, $O(s')$ is not a satisfaction equilibrium because there exists an agent i that is not satisfied by the joint strategy s'.

Case 2 : Strategy s' is not Pareto-comparable to strategy s.

Since s and s' are not Pareto-comparable, there exists an agent i such that $R_i(O(s')) > R_i(O(s))$ and an agent j such that $R_j(O(s')) < R_j(O(s))$. Therefore, since $\sigma_j = R_j(O(s))$, we conclude that $S_j(R_j(O(s'))) = 0$. Consequently, s' is not a satisfaction equilibrium because there exists an agent j that is not satisfied by the joint strategy s'.

Conclusion : If strategy s is Pareto-optimal, the outcome $O(s)$ and its equivalent outcomes are the only satisfaction equilibria if $\sigma_i = R_i(O(s))\ \forall i$. □

Therefore, we see that a major part of the problem of coordinating the agents on a Pareto-optimal joint strategy is to define correctly the satisfaction thresholds of each agent. While we have assumed so far that these thresholds were fixed at the beginning of the learning process, we will show algorithms in the last section that tries to maximize the satisfaction threshold of an agent such that it learns to play its optimal equilibrium under the other agents' strategies.

2.3 Satisfying strategies

Similarly to the concept of dominant strategies, we can define a satisfying strategy as a strategy s_i for agent i such that it is always satisfied when it plays this strategy. This is illustrated in figure 2.

	A	B
A	1,0	1,0
B	1,1	0,0

Figure 2: Games with satisfying strategies.

In this game we see that row agent has a satisfying strategy A. Therefore, if row agent starts playing strategy A, then column agent will be forced to accept an outcome corresponding to joint strategy (A, A) or (A, B). This is problematic since none of these outcomes are satisfaction equilibria. The effects of such strategies on the convergence of our algorithms will be showed with experimental results in the next sections.

[2] We consider that an outcome o' is equivalent to another outcome o if the rewards of all agents are the same in o and o' : $R_i(o) = R_i(o')\forall i$

2.4 Games with multiple Satisfaction Equilibria

In some games, more than one satisfaction equilibrium can exists depending on how the satisfaction thresholds are defined. For example, we can consider the battle of sexes, presented in figure 3 with satisfaction thresholds set to 1.

	B	F
B	2,1	0,0
F	0,0	2,1

$\Longrightarrow$

	B	F
B	1,1	0,0
F	0,0	1,1

$(\forall i | : \sigma_i = 1)$

Figure 3: Battle of sexes game matrix (left) and satisfaction matrix (right).

What will happen when more than one satisfaction equilibrium exists is that both agents will keep or change their strategy until they coordinate themselves on one of the satisfaction equilibrium. From there, they will keep playing the same action all the time.

2.5 Mixed Satisfaction Equilibrium

In some games, such as zero sum games with no tie strategy[3], it is impossible to find a satisfaction equilibrium in pure strategy, unless we set the satisfaction threshold to the minimum possible reward. However, higher expected rewards could be obtained by playing mixed strategies. This can be achieved by playing a mixed satisfaction equilibrium.

Definition 3. *A mixed satisfaction equilibrium is a joint mixed strategy p such that all agents are satisfied by their expected reward.*

$$S_i(E_i(p)) = 1 \qquad \forall i$$

$E_i(p)$ represents the expected reward of agent i under the joint mixed strategy p. While this works in theory, the only way an agent will have to compute its expected reward will be to compute the average of the past n rewards it obtained under its current strategy, since it does not know the strategy of the other agents.

3 Learning the Satisfaction Equilibrium

We now present an algorithm that can be used by agents to learn over time to play the satisfaction equilibrium of a game.

3.1 Pure Satisfaction Equilibrium with fixed thresholds

The most basic case we might want to consider is the case where an agent tries to find a pure strategy that will always satisfy its fixed satisfaction threshold.

Our algorithm 1 implements the satisfaction principle in the most basic way: if the agent is satisfied, it keeps its current action, else it chooses a random action in its set of actions to replace its current action.

Here, the threshold K defines the allowed number of repeated plays and the $ChooseAction$ function chooses a random action uniformly within the set of actions A_i of the agent. Under this learning strategy, once all agents are satisfied, no agent will change its strategy and therefore all agents reach an equilibrium. Once the agent has played K times, it returns its last chosen strategy. Evidently, in games where there exists no satisfaction equilibrium under the agents' satisfaction thresholds, they will never reach an equilibrium. Furthermore, if agent i has a satisfying strategy s_i, then we are not sure to reach a satisfaction equilibrium if s_i does not lead to an equilibrium (see figure 2 for an example).

[3]In zero sum games, a tie strategy is a pure joint strategy s such that the reward of all agents is 0.

Algorithm 1 PSEL: Pure Satisfaction Equilibrium Learning

```
Function PSEL(σ_i, K)
s_i ← ChooseAction()
for n = 1 to K do
    Play s_i and observe outcome o
    if R_i(o) < σ_i then
        s_i ← ChooseAction()
    end if
end for
return s_i
```

3.2 Using an exploration strategy

While we have considered in our previous algorithm 1 that the $ChooseAction$ function selects a random action within the set of actions of the agent, we can also try to implement a better exploration strategy such that actions that have not been explored often could have more chance to be chosen. To achieve this, the agent can compute a probability for each action, that corresponds to the inverse of the times it has chosen them, and then normalize the probabilities such that they sum to 1. Finally, it chooses its action according to the resulting probability distribution. The results presented in section 3.3 will confirm that using this exploration strategy, instead of a uniform random choice, offers a slight improvement in the average number of plays required to converge to a satisfaction equilibrium.

3.3 Empirical results with the PSEL Algorithm

We now present results obtained with the PSEL algorithm in different games. We have used 2 usual games, i.e. the prisoner's dilemma with satisfaction thresholds set to -1 for both agents (see figure 1 for the corresponding satisfaction matrix) and the battle of sexes with satisfaction thresholds set to 1 for both agents (see figure 3 for the corresponding satisfaction matrix). We also used a cooperative game presented in figure 4 with satisfaction thresholds set to 3.

	A	B	C
A	0,0	1,1	0,0
B	2,2	0,0	0,0
C	0,0	0,0	3,3

$\Longrightarrow$

	A	B	C
A	0,0	0,0	0,0
B	0,0	0,0	0,0
C	0,0	0,0	1,1

$(\forall i | : \sigma_i = 3)$

Figure 4: Cooperative game matrix (left) and satisfaction matrix (right).

In this cooperative game, we set the satisfaction thresholds to 3 for both agents such that the only satisfaction equilibrium is joint strategy (C, C). Finally, we also used a bigger

game to verify the performance of our algorithm when the joint strategy space is bigger. This game is presented in the following figure 5.

	A	B	C	D	E	F	G	H
A	0,0	0,0	-1,4	0,0	2,-2	0,0	3,0	0,0
B	1,2	0,0	0,0	3,0	0,0	0,0	0,3	1,1
C	0,0	3,3	0,0	1,1	0,0	0,0	2,2	0,0
D	4,4	0,0	5,1	0,2	2,2	1,4	0,0	0,0
E	0,1	0,0	0,0	5,5	0,0	0,0	2,1	0,0
F	0,4	2,2	0,2	0,0	0,0	3,3	0,0	4,4
G	0,0	5,3	3,0	0,0	-1,3	0,0	2,-1	0,0
H	0,0	2,4	1,1	0,0	0,0	-3,2	0,0	0,0

Figure 5: Big game matrix.

In this big game, the satisfaction thresholds were set to 5 for both agents and therefore, the only satisfaction equilibrium was joint strategy (E, D).

For each of these four games, we ran 5000 simulations, consisting of 5000 repeated plays per simulation, varying the random seeds of the agents each time. In figure 6, we present for each of these games the number of possible joint strategies, the number of SE, the convergence rate to a SE and a comparison of the average number of plays required to converge to such an equilibrium (with 95% confidence interval) with the random and exploration strategies presented.

In each of these games, the SE were corresponding to pareto-optimal joint strategies and the satisfaction thresholds were set according to theorem 2. In all cases, we always converged to a SE within the allowed 5000 repeated plays. Therefore, we see from these results that when the satisfaction thresholds are well defined, we seem to eventually converge toward a Pareto-optimal satisfaction equilibrium[4] (POSE).

3.4 Problematic games and convergence problems

The previous results were quite satisfying provided that we reach a POSE in all cases. However, we will see that some specific payoff structures can pose problem to the efficiency of this algorithm. For example, let's suppose we have a game as in figure 7.

	A	B	C
A	1,1	0,1	0,1
B	1,0	1,0	0,1
C	1,0	0,1	1,0

$\Longrightarrow$

	A	B	C
A	1,1	0,1	0,1
B	1,0	1,0	0,1
C	1,0	0,1	1,0

$(\forall i | : \sigma_i = 1)$

Figure 7: A problematic game matrix (left) and satisfaction matrix (right).

In this game, there exists a unique Pareto-optimal joint strategy (A, A). With the satisfaction thresholds set to 1 for both agents, the corresponding satisfaction matrix is the same as the original game matrix. But, what we can see in this example is that we will never reach the POSE (A, A) unless both agents starts with strategy A. Effectively, if one of the agent plays A but the other agent plays B or C, then the agent playing A will never be satisfied until it changes its strategy to B or C. This problem comes from the fact that an agent playing B or C will always be satisfied when the other agent plays A, and therefore, it will never change its strategy to A when the other agent plays A. Also, there is no point where both agents are unsatisfied that could allow a direct transition to joint strategy (A, A). From this, we conclude that if both agents does not start at the point of equilibrium (A, A) they will never reach an equilibrium since there exists no sequence of transitions that leads to this equilibrium.

Furthermore, in games where satisfying strategies exists (see figure 2), the results shown in figure 8 effectively show that the convergence rate of the PSEL algorithm is dramatically affected by such strategies.

Figure 8: Convergence rate to a SE in problematic games with the PSEL algorithm using the exploration strategy.

Game	$\|A\|$	conv. rate (%)
Problematic Game	9	10.88
Game with satisfying strategy	4	33.26

3.5 Convergence of the PSEL algorithm

While we have already showed that the PSEL algorithm does not work in all games, there is a specific class of games where we can easily define the convergence probability of the PSEL algorithm according to theorem 4.

Theorem 4. *In all games where all agents have the same satisfaction in all outcomes, i.e. ($S_i(R_i(o)) = S_j(R_j(o)) \forall i, j, o$), the PSEL algorithm, using a uniform random exploration, will converge to a SE within K plays with probability $1 - q^K$ where $q = 1 - n_{SE}/|A|$ and the expected number of plays required to converge is given by $|A|/n_{SE}$.*[5]

Proof. Let a game $G = (n, A, \Omega, O, R_1, \ldots, R_n)$ and the satisfaction thresholds of all agents be defined such that $S_i(R_i(o)) = S_j(R_j(o)) \forall i, j, o$. We have that, by definition of the satisfaction thresholds, in any outcome o, all agents are all satisfied or all unsatisfied by outcome o. In the case where all agents are satisfied, the PSEL algorithm has converged to a satisfaction equilibrium and the agents will keep playing the same strategy. We will therefore consider the case where all agents are unsatisfied. In this situation, every agent may decide to change its strategy or not and, since all agents use a random uniform distribution to choose their new action, all the joint strategies are equally probable to be chosen. Therefore, the probability p that the agents immediately converge to a SE is given by $p = n_{SE}/|A|$. In the case where the agents do not choose the right joint strategy such that they are still all unsatisfied, the same process will repeat again with the same probabilities, until they reach a SE and converge. Therefore, we seek to find the probability that a first successful transition to a SE happens within K play. This is given by

[4]We define a Pareto-optimal satisfaction equilibrium as a joint strategy that is a satisfaction equilibrium and also Pareto-optimal.

[5]$|A|$ represents the joint action space size and n_{SE} is the number of SE in the game.

Figure 6: Convergence rate and plays needed to converge to a SE in different games with the PSEL algorithm.

Game	$\|A\|$	n_{SE}	conv. rate	Random Avg. plays	Exploration Avg. plays	Improvement %
Prisoner's Dilemma	4	1	100	8.67 ± 0.23	6.72 ± 0.18	22.49
Battle of Sexes	4	2	100	1.97 ± 0.04	1.95 ± 0.04	1.02
Cooperative Game	9	1	100	8.92 ± 0.23	7.82 ± 0.19	12.33
Big Game	64	1	100	67.95 ± 1.89	61.51 ± 1.65	9.48

a geometric law where the probability of success is parameter $p = n_{SE}/|A|$. In a geometric law, the probability that it takes more than K trials to get a first success is given by $P(k > K) = q^K$ where $q = 1 - p$. Therefore the probability that we have a first success within K trials is given by $P(k \leq K) = 1 - P(k > K) = 1 - q^K$. Furthermore, the expected number of trials required to get a first success in a geometric law is given by $E = 1/p$. From these properties of the geometric law, we conclude that we will converge to a SE within K plays with probability $1 - q^K$ where $q = 1 - n_{SE}/|A|$ and that the expected number of plays required to converge is given by $|A|/n_{SE}$. □

This will be always true in cooperative games (i.e., a game where all agents have the same reward function) if we use the same satisfaction threshold for all agents. In this case, since all agents have the same rewards and satisfaction thresholds, they will always have the same satisfaction in all outcomes. From theorem 4, we can conclude that in such games, as $K \to \infty$, the convergence probability will tend toward 1.

4 Learning the Satisfaction Threshold

While the PSEL algorithm has showed interesting performance in some games, it has the disadvantage that the satisfaction threshold must correctly be set in order to achieve good results. To alleviate this problem, we present a new learning strategy that tries to maximize the satisfaction threshold while staying in a state of equilibrium.

4.1 Limited History Satisfaction Learning Algorithm

In order to achieve this, we present an algorithm that implements the strategy of increasing the satisfaction threshold when the agent is satisfied and decreasing the satisfaction threshold when it is unsatisfied. We also decreases the increment/decrement over time in order to converge to a certain fixed satisfaction threshold. This will be achieved by multiplying the increment by a certain factor within the interval $(0, 1)$ after each play. Moreover, we keep a limited history of the agent's experience in order to prevent it from overrating its satisfaction threshold, by checking whether it was unsatisfied by its current strategy in the past when its satisfaction threshold was higher than a certain threshold. We will see in the results, that this technique really helps the convergence rate of the algorithm compared to the case where we do not prevent this, as in the special case where the history size will be 0.

In this algorithm, the satisfaction threshold σ_i is initialized to the minimum reward of agent i and the variable δ_i is used to increment/decrement this satisfaction threshold. More precisely, δ_i is decremented over time, such that it tends toward 0, by multiplying it by the constant $\gamma_i \in (0, 1)$ after each play. The matrix S keeps an history of the last n states of satisfaction for each action and the matrix Σ keeps, for each action, an history of the last n satisfaction thresholds when the agent played these actions. This history is used to check before incrementing the satisfaction threshold, whether or not the agent was unsatisfied by its current strategy in the past when its satisfaction threshold was below its new satisfaction threshold. Finally, after each play, we update the history of the agent. We consider that the algorithm has converged to the optimal satisfaction threshold when δ_i is lower than a certain constant $\epsilon_i \simeq 0$. At this point, the algorithm returns the satisfaction threshold and the last strategy chosen by agent i. When all agents have converged, if they are all satisfied by their strategy, then we can consider that we have reach a satisfaction equilibrium since their satisfaction threshold will be almost stable. While we are not guaranteed to converge toward a POSE, we will see that in practice, this algorithm yields almost a convergence rate of 100% toward the POSE in any non problematic games.

Algorithm 2 LHSL : Limited History Satisfaction Learning

Function LHSL(δ_i, γ_i, n_i)
$\sigma_i \leftarrow min(r_i)$; $s_i \leftarrow ChooseAction()$
$S[0..|A_i| - 1, 0..n - 1] \leftarrow$ a matrix initialized with true values
$\Sigma[0..|A_i| - 1, 0..n - 1] \leftarrow$ a matrix initialized with $min(r_i)$ values
while $\delta_i > \epsilon_i$ **do**
  Play s_i and observe outcome o
  $lastStrategy \leftarrow s_i$; $satisfied \leftarrow (R_i(o) < \sigma_i)$; $tmp \leftarrow 0$
  if not $satisfied$ **then**
    $s_i \leftarrow ChooseAction()$; $tmp \leftarrow -\delta_i$
  else if not unsatisfied with s_i and $\sigma_i + \delta_i$ in history **then**
    $tmp \leftarrow \delta_i$
  end if
  If $n > 0$ add $satisfied$ and σ_i in history of $lastStrategy$ and remove oldest values
  $\sigma_i \leftarrow \sigma_i + tmp$; $\delta_i \leftarrow \delta_i \cdot \gamma_i$
end while
return (s_i, σ_i)

4.2 Empirical results with the LHSL Algorithm

To test the LHSL algorithm, we have used the same 6 games we have presented for the results with the PSEL algorithm and we now try to learn the POSE without giving a priori its value to set accordingly the satisfaction threshold. The results were obtained over 5000 simulations and we show the convergence rate to the POSE obtained with the best γ_i value

and history sizes we have tested[6]. We also compare these results to the special case where we do not use an history, i.e. $n = 0$. In all cases, δ_i was set to 1 and the convergence threshold ϵ_i was set to 10^{-20}.

In all cases, the best results, showed in figure 9, were obtained with the exploration strategy we have presented in section 3.2. In most games, except the problematic game (figure 7, we were able to get a convergence rate near 100%. We can also see that the use of an history offers a significant improvement over the results we obtain without an history. As a side note, the convergence rate of the LHSL algorithm seems to vary a lot depending on the history sizes and gamma values. This is illustrated in figure 10.

The first graphic in figure 10 compares the results with different history sizes and γ values. We can see that the bigger the history size, the closer to 1 γ must be in order to achieve better performances. Although, in general, the more slowly we decrement δ and the more bigger the history size is, the better are the results. In the second graphic, we compare the convergence rate of the 2 different exploration approaches under different γ values for the prisoner's dilemma, in the case where no history was used. This graphic confirms that the exploration strategy presented in section 3.2 improves slightly the convergence rate of the LHSL algorithm.

5 Discussions

Repeated games provides a formal framework to explore the long-term relationships between agents. There is now an extensive literature on this topic where the dominant tendency heavily relies on one crucial assumption, which does exclude a number of important applications [Kandori, 2002]. The key assumption in the existing literature is that agents (or players) share *common information* about each other's actions. In fact, one should distinguish between perfect and imperfect monitoring, and also between public and private monitoring. In perfect monitoring, agents commonly observe actual actions and rewards of all agents, and in the imperfect public monitoring, agents observe a common signal which is an indicator of the joint action played (e.g., the market price), but they do not observe the other agent's rewards. In this last case of imperfect private monitoring, the agents can only receive *private* information (e.g., one's own rewards). Basically, the imperfect private monitoring leads to two major difficulties [Kandori, 2002]: (1) the games lack the *recursive structure* in the sense of Abreu et al. [Abreu *et al.*, 1991] and consequently the set of equilibria does not possess a simple characterization and; (2) at each moment in time, agents must conduct *statistical inference* on what others are going to do, which can be quite complex.

Thus, when agents condition their own actions on commonly observed events, they play a Nash equilibria of the remaining game, which is identical to the original infinitely repeated game. In these conditions and after any history, the set of continuation payoffs is always equal to the equilibrium payoff set of the repeated game. This is the idea behind the recursive structure. Under private monitoring such a characterization is no longer available since at each moment t, an agent i determines its actions on its private history of its actions and private signals which are only known to him. As each agent i has its private history h_i^t, the *correlation device* formed by the agents' joint private histories becomes increasingly more complex over time, so the set of continuation payoffs changes over the time.

The second difficulty is that checking for a "better strategy" in each stage requires fairly complex statistical inference since all what they know, under private monitoring, is a private history for each agent. In other words, agent i should calculate conditional distribution $Pr(h_{-i}^t|h_i^t)$ by Bayes' rule in each stage t and this can become increasingly more complex as times passes by. Furthermore, if no prior knowledge about the game and other agents is available, this is generally impossible to compute.

Thus, the imperfect private monitoring is a non-trivial task and classical game theory concepts such as best response and Nash Equilibria cannot be computed by the agents. In order to address this problem, we must define an equilibrium that can arise as the result of a learning process, over repeated play where an agent i tries to determine $Pr(a_i^t|h_i^t)$, that does not require knowledge of the game matrix and observations of the other agent's actions and rewards. To this end, we have introduced in this paper the satisfaction principle, which stipulates that *an agent that has been satisfied by its payoff will not change its strategy, while an unsatisfied agent may decide to change its strategy*[7]. Under this principle, an equilibrium will arise when all agents will be satisfied by their payoff, since no agent will have any reason to change its strategy. The equilibrium sustaining this principle has been called a *satisfaction equilibrium*.

The satisfaction principle under imperfect private monitoring presented in this paper has shown great performance in practice, since except for some games with specific payoff structures that could pose problems, it converges to a satisfaction equilibrium, which is in general the Pareto optimal solution. By doing so, we have shown that a convention or a regularity (which is the satisfaction equilibria) amongst members of the game when they are facing to imperfect private monitoring, can emerged. Such a convention or regularity opens the door to "cooperations" in DP games and to more general results in other,; results which are more close to our "intuitions" than the usual rational thinking.

6 Conclusion and future works

While this article covered a lot of new concepts, it laid out only the basic theoretical foundations of the satisfaction equilibrium. The algorithms we have presented have shown great performance in practice, but we have seen some games with specific payoff structures that could pose problems or rend impossible the convergence to a satisfaction equilibrium. We have identified possible solutions, such as allowing mixed satisfaction equilibrium and trying to maximize the satisfaction threshold, that could sometimes palliate to these problems. Although, what we may realize is that in some games it might

[6] In these results, γ_i, δ_i, ϵ_i, σ_i and the history size were the same for all agents

[7] This principle takes its essence from the notion of aspiration-based reinforcement learning in repeated games where bounded rational agents try to satisfice (in the sense of H. Simon) rather than maximize payoffs [Bendor *et al.*, 2001].

Figure 9: Convergence rate to a POSE in different games with the LHSL algorithm

Game	$\|A\|$	With history conv. rate (%)	γ_i	n_i	Without history conv. rate (%)	γ_i
Prisoner's Dilemma	4	100	0.99	64	89.96	0.90
Battle of Sexes	4	100	0.90	16	97.60	0.80
Cooperative Game	9	99.66	0.995	128	97.62	0.95
Big Game	64	99.66	0.995	16	93.88	0.99
Problematic Game	9	9.86	0.95	128	7.88	0.50
Game with satisf. strat.	4	98.06	0.95	128	38.78	0.95

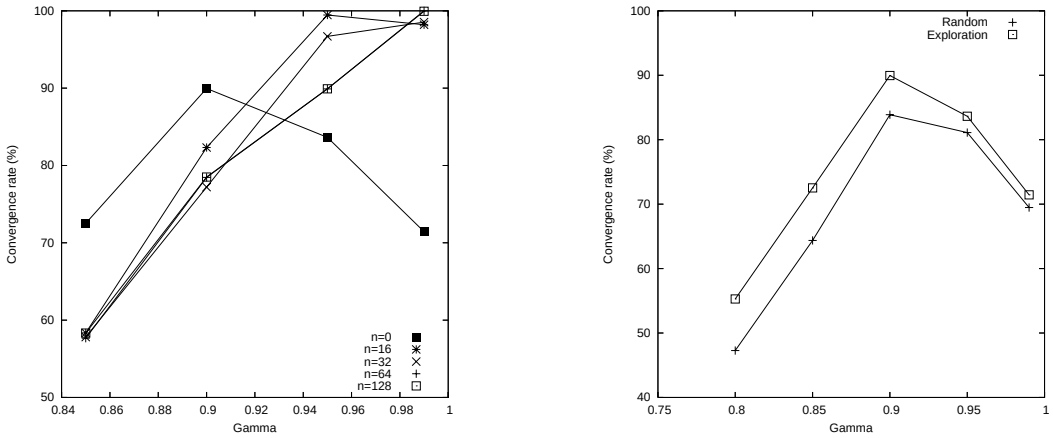

Figure 10: Convergence rate to a POSE in the prisoner's dilemma under different γ values, history sizes and exploration strategies

not always be possible to converge to a satisfaction equilibrium, or to a POSE. What we might want to do in these games is to converge toward a Nash equilibrium. If convergence to a Nash equilibrium is always possible, then we may try to find an algorithm that converges in the worst case to a Nash equilibrium, and in the best case, to a Pareto-optimal satisfaction equilibrium. In order to achieve this goal, the next step will be to develop an algorithm that can converge to a Pareto-optimal mixed satisfaction equilibrium. Also, a lot of theoretical work needs to be done to prove and/or bound the efficiency of the presented algorithms and identify clearly in which cases the algorithms will converge or not to a satisfaction equilibrium. Afterward, another long term goal is to apply the satisfaction equilibrium to stochastic games in order to allow agents to learn a Pareto-optimal joint strategy without knowing anything about the other agents in those games.

References

[Abreu *et al.*, 1991] D. Abreu, P. Milgron, and D. Pearce. Information and timing in repeated partnerships,. *Econometria*, 59:1713–1733, 1991.

[Bendor *et al.*, 2001] J. Bendor, D Mookherjee, and Ray D. Aspiration-based reinforcement learning in repetaed interaction games: an overview,. *International Game Theory Review*, 3(2&3):159–174, 2001.

[Claus and Boutillier, 2000] C. Claus and C. Boutillier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proc. of AAAI/IAAI*, pages pp. 746–752, 2000.

[Dekel *et al.*, 2004] Eddie Dekel, Drew Fudenberg, and David K. Levine. Learning to play bayesian games. *Games and Economic Behavior*, 46(2):282–303, 2004.

[Fulda and Vantura, 2004] N. Fulda and D. Vantura. Incremental policy learning: an equilibrium selection algorithm for reinforcement learning agents with common interests. In *Proc. of IJCNN'04*, pages pp. 1121–1126, 2004.

[Harsanyi, 1967] John Harsanyi. Games of incomplete information played by bayesian players. *Management Science*, 14:159–182, 320–334, and 486–502, 1967.

[Kandori, 2002] M. Kandori. Introduction to repeated games with private monitoring. *J. of Economic Theory*, 102:pp. 1–15, 2002.

[Kapetanakis and Kudenko, 2002] S. Kapetanakis and D. Kudenko. Improving on the reinforcement learning of coordination in cooperative multi-agent systems. In *Proc. of the* 2^{th} *AISB Sym. on Adaptative Agents and Multiagent Systems*, 2002.

[Karandikar and Mookherjee, 1998] R. Karandikar and D. Mookherjee. Evolving aspirations and cooperation. *J. of Economic Theory*, 80:pp. 292–331, 1998.

[Lauer and Riedmiller, 2000] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proc. of ICML'00*, pages pp. 535–542, SF, CA, 2000. Morgan kaufman.

[Stimpson *et al.*, 2001] J.L. Stimpson, M.A. Goodrich, and L.C. Walters. Satisficing and learning cooperation in the priosnner's dilemma. In *Proc. IJCAI'01*, 2001.

[Wikepedia, 2006] Wikepedia. http://en.wikipedia.org/wiki/cooperation. 2006.