Reaching Correlated Equilibria Through Multi-agent Learning

Ludek Cigler
Ecole Polytechnique Fédérale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
ludek.cigler@epfl.ch

Boi Faltings
Ecole Polytechnique Fédérale de Lausanne
Artificial Intelligence Laboratory
CH-1015 Lausanne, Switzerland
boi.faltings@epfl.ch

ABSTRACT

Many games have undesirable Nash equilibria. For example consider a resource allocation game in which two players compete for an exclusive access to a single resource. It has three Nash equilibria. The two pure-strategy NE are efficient, but not fair. The one mixed-strategy NE is fair, but not efficient. Aumann's notion of correlated equilibrium fixes this problem: It assumes a correlation device which suggests each agent an action to take.

However, such a "smart" coordination device might not be available. We propose using a randomly chosen, "stupid" integer coordination signal. "Smart" agents learn which action they should use for each value of the coordination signal.

We present a multi-agent learning algorithm which converges in polynomial number of steps to a correlated equilibrium of a wireless channel allocation game, a variant of the resource allocation game. We show that the agents learn to play for each coordination signal value a randomly chosen pure-strategy Nash equilibrium of the game. Therefore, the outcome is an efficient correlated equilibrium. This CE becomes more fair as the number of the available coordination signal values increases.

We believe that a similar approach can be used to reach efficient and fair correlated equilibria in a wider set of games, such as potential games.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Economics

Keywords

Multiagent Learning, Coordination, Game Theory

1. INTRODUCTION

The concept of Nash equilibrium forms the basis of game theory. It allows us to predict the outcome of an interaction between rational agents playing a given game.

Cite as: Reaching Correlated Equilibria Through Multi-agent Learning, L. Cigler and B. Faltings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX-XXX.

However, many games have undesirable equilibrium structure. Consider the following resource allocation game: Two agents are trying to access a single resource. Agents can choose between two actions: *yielding* (Y) or *accessing* (A). The resource may be accessed only by one agent at a time. If an agent accesses the resource alone, she receives a positive payoff. If an agent does not access the channel, her payoff is 0. If *both* agents try to access the channel at the same time, their attempts fail and they incur a cost c.

The payoff matrix of the game looks as follows:

	Y	A
Y	0, 0	0, 1
A	1, 0	$-c$, $-c$

Such a game has two pure-strategy Nash equilibria (NE), in which one player yields and the other one goes straight. It has also one mixed-strategy NE, where each player yields with probability $\frac{1}{c+1}$. The two pure-strategy NE are efficient, in that they maximize the social welfare, but they are not fair: Only one player gets the full payoff, even though the game is symmetric. The mixed-strategy NE is fair, but not efficient: The expected payoff of both players is 0.

In his seminal paper, Aumann ([1]) proposed the notion of *correlated equilibrium* which fixes this problem. A correlated equilibrium (CE) is a probability distribution over the joint strategy profiles in the game. A correlation device samples this distribution and recommends an action for each agent to play. The probability distribution is a CE if agents do not have an incentive to deviate from the recommended action.

In the simple game described above, there exists a CE which is both fair and socially efficient: just play the two pure-strategy NE with probability $\frac{1}{2}$. This corresponds to an authority which tells each player whether to yield or access the resource.

Correlated equilibria have several nice properties: They are easier to find (for a succinct representation of a game, in polynomial time, [11]) and every Nash equilibrium is a correlated equilibrium. Also, any convex combination of two correlated equilibria is a correlated equilibrium. However, a "smart" correlation device which randomizes over joint strategy profiles might not always be available.

It is possible to achieve a correlated equilibrium without the actual correlation device. Assume that the game is played repeatedly, and that agents can observe the history of actions taken by their opponents. They can learn to predict the future action (or a distribution of future actions) of the opponents. These predictions need to be *calibrated*, that is, the predicted probability that an agent i will play a cer-

tain action a_j should converge to the actual frequency with which agent i plays action a_j. Agents always play an action which is the best response to their predictions of opponents' actions. Forster and Vohra in [5] showed that in such a case, the play converges to a set of correlated equilibria.

However, in their paper, Foster and Vohra did not provide a specific learning rule to achieve a certain CE. Furthermore, their approach requires that every agent were able to observe actions of every other opponent. If this requirement is not met, convergence to a correlated equilibrium is not guaranteed anymore.

In this paper, we focus on a variant of the resource allocation game, a game of wireless channel allocation. In this game, there are N agents who always have some data to transmit, and there are C channels over which they can transmit. We assume that $N \geq C$. Access to a channel is slotted, that is, all agents are synchronized so that they start transmissions at the same time. Also, all transmissions must have the same length. If more than one agent attempts to transmit over a single channel, a collision occurs and none of the transmissions are successful. An unsuccessful transmission has a cost for the agent, since it has to consume some of its (possibly constrained) power for no benefit. Not transmitting does not cost anything.

We assume that agents only receive binary feedback. If they transmitted some data, they find out whether their transmission was successful. If they did not transmit, they can choose some channel to observe. They receive information whether the observed channel was free or not.

The game has several efficient (but unfair) pure-strategy Nash equilibria, in which a group of C agents gets assigned all the channels. The remaining $N - C$ agents get stranded. It has also a fair but inefficient mixed-strategy NE, in which agents choose the transmission channels at random. As in the resource allocation game, there exists a correlated equilibrium which is efficient and fair.

In this scenario, a global coordination device that would tell each agent which channel to transmit on is not available. Imagine that the agents are wireless devices belonging to different organizations. Setting up such a coordination device would require additional communication before the transmissions. Moreover, agents cannot observe all the actions of their opponents, since the feedback they receive is very limited. Therefore, they cannot learn the fair and efficient correlated equilibrium from the history of the play.

We propose a different approach to achieve an efficient and fair correlated equilibrium in such a game. We do not want to rely on a complex correlation device which needs to know everything about the game. Also, we do not want to rely on the history which may not be observable. Instead, we assume that agents can observe, before each round of the game, a randomly chosen integer from a set $\{0, 1, \ldots, K-1\}$. For each possible signal value, agents learn which action to take.

Our correlation signal does not need to know anything about the game. It does not have to tell agents which action to take. For example, the agents may just observe noise on some frequency. This is the principal difference from using the "smart" coordination device, which is assumed in the original definition of correlated equilibrium.

The main contributions of this work are the following:

- We propose a learning strategy for agents in the wireless channel allocation game which, using minimal information, converges in polynomial time to a randomly chosen efficient pure-strategy Nash equilibrium of the game.

- We show that when the agents observe a common integer correlation signal, they learn to play such an efficient pure-strategy NE for each signal value. The result is a correlated equilibrium which is increasingly fair as the number of available signals K increases.

The rest of the paper is organized as follows: In Section 2, we present the algorithm agents use to learn an action for each possible correlation signal value. In Section 3 we prove that such an algorithm converges to an efficient correlated equilibrium in polynomial time in the number of agents and channels. We show that the fairness of the resulting equilibria increases as the number of signals K increases in Section 4. Section 5 highlights experiments which show the actual convergence rate and fairness. In Section 6 we present some related work from game theory and cognitive radio literature, and Section 7 concludes.

2. LEARNING ALGORITHM

In this section, we describe the algorithm which the agents will use to learn a correlated equilibrium of the wireless channel allocation game.

Let us denote the space of available correlation signals $\mathcal{K} := \{0, 1, \ldots, K-1\}$, and the space of available channels $\mathcal{C} := \{1, 2, \ldots, C\}$. Assume that $C \leq N$, that is there are more agents than channels (the opposite case is easier). An agent i has a strategy $f_i : \mathcal{K} \to \{0\} \cap \mathcal{C}$ which it uses to decide which channel it will access at time t when it receives a correlation signal k_t. When $f_i(k_t) = 0$, the agent does not transmit at all for signal k_t. The agent stores its strategy simply as a table.

It adapts the strategy as follows:

1. In the beginning, for each $s \in \mathcal{K}$, $f_i(s)$ is initialized uniformly at random from $\mathcal{C}$.

2. At time t, if $f_i(k_t) > 0$, the agent tries to transmit over channel $f_i(k_t)$. If otherwise $f_i(k_t) = 0$, the agent chooses a random channel $m_i(t) \in \mathcal{C}$ which it will monitor for activity.

3. Subsequently, the agent observes the outcome of its choice: if the agent transmitted over some channel, she observes whether the transmission was successful. If it was, the agent will keep her strategy unchanged. If a collision occurred, the agent sets $f_i(k_t) := 0$ with probability p.

4. If the agent did not transmit, it observes whether there was a transmission on the channel $m_i(t)$ it monitored. If that channel was free, the agent sets $f_i(k_t) := m_i(t)$.

3. CONVERGENCE

An important property of the learning algorithm is if, and how fast it can converge to a pure-strategy Nash equilibrium of the channel allocation game for every signal value. The algorithm is randomized. Therefore, instead of analyzing its worst-case behavior (which may be arbitrarily bad), we will analyze its expected number of steps before convergence.

3.1 Convergence for $C = 1, K = 1$

We prove the following theorem:

THEOREM 1. *For N agents and $C = 1, K = 1, 0 < p < 1$, the expected number of steps before the allocation algorithm converges to a pure-strategy Nash equilibrium of the channel allocation game is $O\left(\frac{1}{p(1-p)} \log N\right)$.*

To prove the convergence of the algorithm, it is useful to describe its execution as a Markov chain.

When N agents compete for a single signal value (a "slot"), a state of the Markov chain is a vector from $\{0,1\}^N$ which denotes which agents are attempting to transmit. For the purpose of the convergence proof, it is only important how *many* agents are trying to transmit, not which agents. This is because the probability with which the agents back-off is the same for everyone. Therefore, we can describe the algorithm execution using the following chain:

Definition 1. A Markov chain describing the execution of the allocation algorithm for $C = 1, K = 1, 0 < p < 1$ is a chain whose state at time t is $X_t \in \{0, 1, \ldots, N\}$, where $X_t = j$ means that j agents are trying to transmit at time t.

The transition probabilities of this chain look as follows:

$$P(X_{t+1} = N | X_t = 0) = 1 \qquad \text{(restart)}$$

$$P(X_{t+1} = 1 | X_t = 1) = 1 \qquad \text{(absorbing)}$$

$$P(X_{t+1} = j | X_t = i) = \binom{i}{j} p^{i-j}(1-p)^j \qquad i > 1, j \leq i$$

All the other transition probabilities are 0.

We are interested in the number of steps it will take this Markov chain to first arrive at state $X_t = 1$ given that it started in state $X_0 = N$. This would mean that the agents converged to a setting where only one of them is transmitting, and the others are not. This quantity is known as the *hitting time*.

Definition 2. [10] Let $(X_t)_{t \geq 0}$ be a Markov chain with state space I. The *hitting time* of a subset $A \subset I$ is a random variable $H^A : \Omega \to \{0, 1, \ldots\} \cup \{\infty\}$ given by

$$H^A(\omega) = \inf\{t \geq 0 : X_t(\omega) \in A\}$$

Specifically, we are interested in the *expected* hitting time of a set of states A, given that the Markov chain starts in an initial state $X_0 = i$. We will denote this quantity

$$k_i^A = \mathbb{E}_i(H^A).$$

In general, the expected hitting time of a set of states A can be found by solving a system of linear equations. Solving them analytically for our Markov chain is however difficult. Fortunately, when the Markov chain has only one absorbing state $i = 0$, and it can only move from state i to j if $i \geq j$, we can use the following theorem to derive an upper bound on the hitting time (proved in [12]):

THEOREM 2. *Let $A = \{0\}$. If*

$$\forall i \geq 1 : E(X_{t+1} | X_t = i) < \frac{i}{\beta}$$

for some $\beta > 1$, then

$$k_i^A < \lceil \log_\beta i \rceil + \frac{\beta}{\beta - 1}$$

The Markov chain of our algorithm does not have the property required by this theorem. The problem is that the absorbing state is state 1, and from state 0 the chain goes back to N.

Nevertheless, we can use Theorem 2 to prove the following lemma:

LEMMA 1. *Let $A = \{0, 1\}$. The expected hitting time of the set of states A in the Markov chain described in Definition 1 is $O\left(\frac{1}{p} \log N\right)$.*

PROOF. We will first prove that the expected hitting time of a set $A' = \{0\}$ in a slightly modified Markov chain is $O\left(\frac{1}{p} \log N\right)$.

Let us define a new Markov chain $(Y_t)_{t \geq 0}$ with the following transition probabilities:

$$P(Y_{t+1} = 0 | Y_t = 0) = 1 \qquad \text{(absorbing)}$$

$$P(Y_{t+1} = j | Y_t = i) = \binom{i}{j} p^{i-j}(1-p)^j \qquad j \geq 0, i \geq 1$$

Note that the transition probabilities are the same as in the chain $(X_t)_{t \geq 0}$, except for states 0 and 1. From state 1 there is a positive probability of going into state 0, and state 0 is now absorbing. Clearly, the expected hitting time of the set $A' = \{0\}$ in the new chain is an upper bound on the expected hitting time of set $A = \{0, 1\}$ in the old chain. This is because any path that leads into state 0 in the new chain either does not go through state 1 (so it happened with the same probability in the old chain), or goes through state 1, so in the old chain it would stop in state 1 (but it would be one step shorter).

If the chain is in state $Y_t = i$, the next state Y_{t+1} is drawn from a binomial distribution with parameters $(i, 1-p)$. The expected next state is therefore

$$E(Y_{t+1} | Y_t = i) = i(1-p)$$

We can therefore use the Theorem 2 with $\beta := \frac{1}{1-p}$ to derive that for $A' = \{0\}$, the hitting time is:

$$k_i^{A'} < \left\lceil \log_{\frac{1}{1-p}} i \right\rceil + \frac{1}{p} \approx O(\frac{1}{p} \log i)$$

which is also an upper bound on k_i^A for $A = \{0, 1\}$ in the old chain. □

LEMMA 2. *The probability h_i that the Markov chain defined in Definition 1 enters state 1 before entering state 0, when started in any state $i > 1$, is greater than $1 - p$.*

PROOF. Calculating the probability that the chain X enters state 1 before state 0 is equal to calculating the *hitting probability*, i.e. the probability that the chain ever enters a given state, for a modified Markov chain where the probability of staying in state 0 is $P(X_{t+1} = 0 | X_t = 0) = 1$. For a set of states A, let us denote h_i^A the probability that the Markov chain starting in state i ever enters some state in A. To calculate this probability, we can use the following theorem (proved in [10]):

THEOREM 3. *Let A be a set of states. The vector of hitting probabilities $h^A = (h_i^A : i \in \{0, 1, \ldots, N\})$ is the minimal non-negative solution to the system of linear equations*

$$h_i^A = \begin{cases} 1 & \text{for } i \in A \\ \sum_{j \in \{0,1,\ldots,N\}} p_{ij} h_j^A & \text{for } i \notin A \end{cases}$$

For the modified Markov chain which cannot leave neither state 0 nor state 1, computing h_i^A for $A = 1$ is easy, since the matrix of the system of linear equations is lower triangular.

We'll show that $h_i \geq \gamma = 1 - p$ for $i > 1$ using induction. The first step is calculating h_i for $i \in \{0, 1, 2\}$.

$$\begin{aligned} h_0 &= 0 \\ h_1 &= 1 \\ h_2 &= (1-p)^2 h_2 + 2p(1-p)h_1 + p^2 h_0 \\ &= \frac{2p(1-p)}{1-(1-p)^2} = \frac{2(1-p)}{2-p} \geq 1-p. \end{aligned}$$

Now, in the induction step, derive a bound on h_i by assuming $h_j \geq \gamma = 1 - p$ for all $j < i, j \geq 2$.

$$\begin{aligned} h_i &= \sum_{j=0}^{i} \binom{i}{j} p^{i-j}(1-p)^j h_j \\ &\geq \sum_{j=0}^{i} \binom{i}{j} p^{i-j}(1-p)^j \gamma - i p^{i-1}(1-p)(\gamma - h_1) - p^i h_0 \\ &= \gamma - i p^{i-1}(1-p)(\gamma - 1) \geq \gamma = 1 - p. \end{aligned}$$

This means that no matter which state $i \geq 2$ the Markov chain starts in, it will enter into state 1 earlier than into state 0 with probability at least $1 - p$. □

From Lemma 2, we derive that in the original Markov chain (where stepping into state 0 meant going into state N), the chain takes on average $\frac{1}{1-p}$ passes through all its states before it converges into state 1. We know from Lemma 1 that one pass takes in expectation $O\left(\frac{1}{p}\log N\right)$ steps, so the expected number of steps before reaching state 1 is $O\left(\frac{1}{p(1-p)}\log N\right)$. This concludes the proof of Theorem 1.

3.2 Convergence for $C \geq 1, K = 1$

THEOREM 4. *For N agents and $C \geq 1, K = 1$, the expected number of steps before the learning algorithm converges to a pure-strategy Nash equilibrium of the channel allocation game is $O\left(C\frac{1}{1-p}\left[\frac{1}{p}\log N + C\right]\right)$.*

PROOF. In the beginning, in at least one channel, there can be at most N agents who want to transmit. It will take on average $O\left(\frac{1}{p}\log N\right)$ steps to get to a state when either 1 or 0 agents transmit (Lemma 1). We will call this period a *round*.

If all the agents backed off, it will take them on average at most C steps before some of them find an empty channel. We call this period a *break*.

The channels might oscillate between the "round" and "break" periods in parallel, but in the worst case, the whole system will oscillate between these two periods.

For a single channel, it takes on average $O\left(\frac{1}{1-p}\right)$ oscillations between these two periods before there is only one agent who transmits in that channel. For $C \geq 1$, it takes on average $O\left(C\frac{1}{1-p}\right)$ steps between "round" and "break" before all channels have only one agent transmitting. Therefore, it will take on average $O\left(C\frac{1}{1-p}\left[\frac{1}{p}\log N + C\right]\right)$ steps before the system converges. □

3.3 Convergence for $C \geq 1, K \geq 1$

To show what is the convergence time when $K > 1$, we will use a more general problem. Imagine that there are K identical instances of the same Markov chain. We know that the original Markov chain converges from any initial state to an absorbing state in expected time T. Now imagine a more complex Markov chain: In every step, it selects uniformly at random one of the K instances of the original Markov chain, and executes one step of that instance. What is the time T_{all} before all K instances converge to their absorbing states?

This is an extension of the well-known *Coupon collector's problem* ([4]). We will prove the following rough upper bound:

LEMMA 3. *Let there be K instances of the same Markov chain which is known to converge to an absorbing state in expectation in T steps. If we select randomly one Markov chain instance at a time and allow it to perform one step of the chain, it will take on average $E[T_{all}] = O(K^2 T)$ steps before all K instances converge to their absorbing states.*

PROOF. Let R_i be the number of steps of the joint Markov chain after which the instance i converges (by joint Markov chain we mean the chain that selects randomly an instance to perform one step). We are interested in

$$E[T_{all}] = E\left[\max_{i\in\{1,\ldots,K\}} R_i\right]$$

For this, it holds that

$$E\left[\max_{i\in\{1,\ldots,K\}} R_i\right] \leq E\left[\sum_{i=1}^{K} R_i\right] = \sum_{i=1}^{K} E[R_i]$$

For $\forall i$, $E[R_i] = KT$, because an instance i is selected in every step with probability $\frac{1}{K}$, and it takes it in expectation T steps to converge. Therefore, $E[T_{all}] \leq K^2 T$. □

For arbitrary $C \geq 1, K \geq 1$, the following theorem follows from Theorem 4 and Lemma 3:

THEOREM 5. *For N agents and $C \geq 1, K \geq 1, 0 < p < 1$, the expected number of steps before the learning algorithm converges to a pure-strategy Nash equilibrium of the channel allocation game for every $k \in \mathcal{K}$ is*

$$O\left(K^2 C \frac{1}{1-p}\left[C + \frac{1}{p}\log N\right]\right).$$

From [1] we know that any Nash equilibrium is a correlated equilibrium, and any convex combination of correlated equilibria is a correlated equilibrium. We also know that all the pure-strategy Nash equilibria that the algorithm converges to are efficient: there are no collisions, and in every channel for every signal value, some agent transmits. Therefore, we conclude the following:

THEOREM 6. *The learning algorithm defined in Section 2 converges in expected polynomial time (with respect to K, C, $\frac{1}{p}$, $\frac{1}{1-p}$ and $\log N$) to an efficient correlated equilibrium of the wireless channel allocation game.*

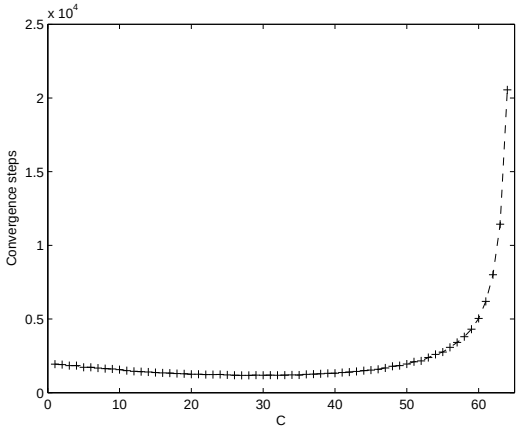


Figure 1: Average number of steps to convergence for $N = 64$, $K = N$ and $C \in \{1, 2, \ldots, N\}$.

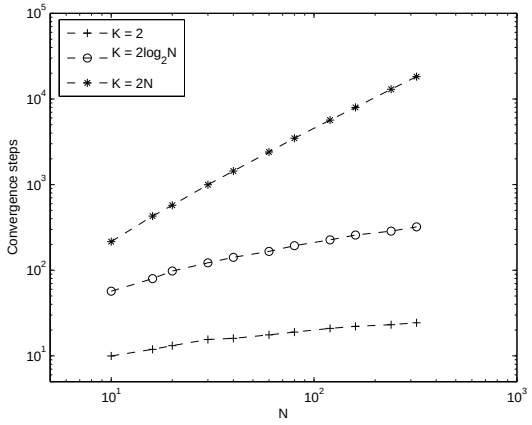


Figure 2: Average number of steps to convergence for $C = \frac{N}{2}$ and varying K.

4. FAIRNESS

Agents decide independently for each value of the coordination signal (a "slot"). Therefore, every agent has an equal chance that the game converges to an equilibrium which is favorable to her. If the agent can transmit in the resulting equilibrium for a given signal value, we say that the agent *wins* the slot. For C available channels and N agents, an agent wins a given slot with probability $\frac{C}{N}$ (since no agent can transmit in two channels at the same time).

We can describe the number of slots won by an agent i as a random variable X_i. This variable is distributed according to a binomial distribution with parameters $\left(K, \frac{C}{N}\right)$.

As a measure of fairness, we use the *Jain index* ([7]). For a random variable X, the Jain index is the following:

$$J(X) = \frac{(E[X])^2}{E[X^2]}$$

When X is distributed according to a binomial distribution with parameters $(K, \frac{C}{N})$, its first and second moments are

$$E[X] = K \cdot \frac{C}{N}$$
$$E\left[X^2\right] = \left(K \cdot \frac{C}{N}\right)^2 + K \cdot \frac{C}{N} \cdot \frac{N-C}{N},$$

so the Jain index is

$$J(X) = \frac{C \cdot K}{C \cdot K + (N - C)}.$$

For the Jain index it holds that $0 < J(X) \leq 1$. An allocation is considered fair if $J(X) = 1$.

THEOREM 7. *For any C, if $K = \omega\left(\frac{N}{C}\right)$, that is the limit $\lim_{N\to\infty} \frac{N}{C \cdot K} = 0$, then*

$$\lim_{N\to\infty} J(X) = 1,$$

so the allocation becomes fair as N goes to ∞.

PROOF. The theorem follows from the fact that

$$\lim_{N\to\infty} J(X) = \lim_{N\to\infty} \frac{C \cdot K}{C \cdot K + (N - C)}$$

For this limit to be equal to 1, we need

$$\lim_{N\to\infty} \frac{N - C}{C \cdot K} = 0$$

which holds exactly when $K = \omega\left(\frac{N}{C}\right)$ (note that we assume that $C \leq N$). □

5. EXPERIMENTAL RESULTS

5.1 Convergence

First, we are interested in the convergence of our allocation algorithm. From Section 3 we know that it is polynomial. How many steps does the algorithm need to converge in practice?

Figure 1 presents the average number of convergence steps for $N = 64$, $S = N$ and increasing number of available channels $C \in \{1, 2, \ldots, N\}$. Interestingly, the convergence takes the longest time when $C = N$. The lowest convergence time is for $C = \frac{N}{2}$, and for $C = 1$ it increases again.

What happens when we change the size of the signal space K? Figure 2 shows the number of convergence steps in that case, for increasing number of agents in the system. Note that this graph uses a double logarithmic scale, so a straight line denotes polynomial, rather than linear dependence of the number of convergence steps on N.

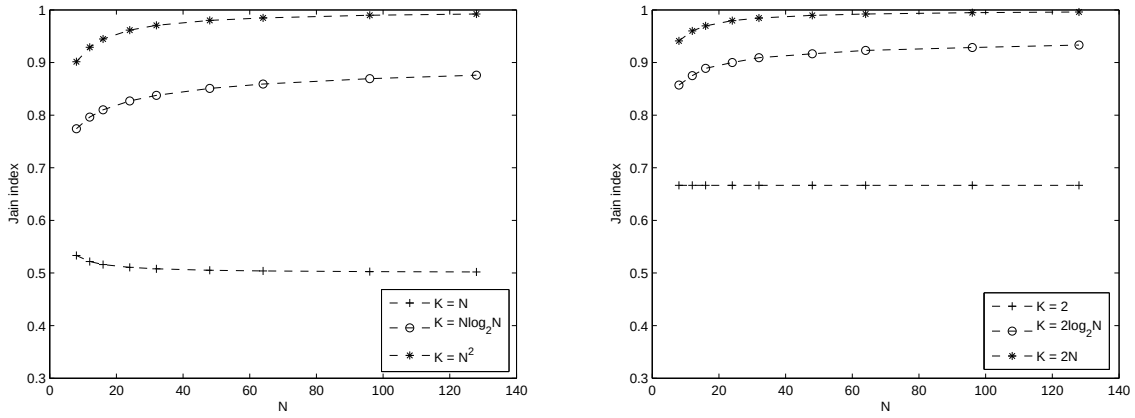
5.2 Fairness

From Section 4, we know that when $K = \omega\left(\frac{N}{C}\right)$, the Jain fairness index converges to 1 as N goes to infinity. But how fast is this convergence? How big do we need to choose K, depending on N and C, to achieve a reasonable bound on fairness?

Figure 3 shows the Jain index as N increases, for $C = 1$ and $C = \frac{N}{2}$ respectively, for various settings of K. Even though every time when $K = \omega\left(\frac{N}{C}\right)$ the Jain index increases, there is a marked difference between the various settings of K.

5.3 Optimizing Fairness

We saw how fair the outcome of the allocation algorithm is when agents consider the game for each slot independently.



(a) $C = 1$ (b) $C = \frac{N}{2}$

Figure 3: Jain fairness index for different settings of C and K, for increasing N.

However, is it the best we can do? Can we further improve the fairness, when each agent correlates her decisions for different signal values?

In a perfectly fair solution, every agent wins (and consequently can transmit) for the same number of slots. However, we assume that agents do not know how many other agents there are in the system. Therefore, the agents do not know what is their fair share of slots to transmit in. Nevertheless, they can still use the information in how many slots they already transmitted to decide whether they should back-off and stop transmitting when a collision occurs.

Definition 3. For a strategy f_i of an agent i, we define its *cardinality* as the number of signals for which this strategy tells the agent to transmit:

$$|f_i| = |\{k \in \mathcal{K} | f_i(k) > 0\}|$$

Intuitively, agents whose strategies have higher cardinality should back-off more often than those with a strategy with low cardinality.

We compare the following variations of the channel allocation scheme, which differ from the original one only in the probability with which agents back off on collisions:

Constant Our scheme; Every agent backs off with the same constant probability p.

Linear The back-off probability is $p = \frac{|f_i|}{K}$.

Exponential The back-off probability is $p = \gamma^{\left(1-\frac{|f_i|}{K}\right)}$ for some parameter $0 < \gamma < 1$.

Worst-agent-last In case of a collision, the agent who has the *lowest* $|f_i|$ does not back off. The others who collided, do back off. This is a greedy algorithm which requires more information than what we assume that the agents have.

To compare the fairness of the allocations in experiments, we need to define the Jain index of an actual allocation. For an allocation $\mathbb{X} = (X_1, X_2, \ldots, X_N)$, its Jain index is:

$$J(\mathbb{X}) = \frac{\left(\sum_{i=1}^{N} X_i\right)^2}{N \cdot \sum_{i=1}^{N} X_i^2}$$

Figure 4 shows the average Jain fairness index of an allocation for the back-off probability variations. The fairness is approaching 1 for the *worst-agent-last* algorithm. It is the worst if everyone is using the same back-off probability. As the ratio between the back-off probability of the lowest-cardinality agent and the highest-cardinality agent decreases, the fairness increases.

This shows that we can improve fairness by using different back-off probabilities. Nevertheless, the shape of the fairness curve is the same for all of them. Furthermore, the exponential back off probabilities lead to much longer convergence, as shown on Figure 5.

6. RELATED WORK

Broadly speaking, in this paper we are interested in games where the payoff an agent receives from a certain action is inversely proportional to the number of other agents who chose the same action. How can we achieve efficient and fair outcome in such games? Variants of this problem have been studied in several previous works.

The simplest such variant is the *Minority game* ([3]). In this game, N agents have to simultaneously choose between two actions. Agents who chose an action which was chosen by a minority of agents receive a payoff of 1, whereas agents whose action choice was in majority receive a payoff of 0. This game has many pure-strategy Nash equilibria, in which some group of $\left\lfloor \frac{N-1}{2} \right\rfloor$ agents chooses one action and the rest choose the other action. Such equilibria are efficient, since the largest possible number of agents achieve the maximum payoff. However, they are not fair: the payoff to the losing group of agents is always 0. This game has also one mixed-strategy NE which is fair: every agent chooses its action randomly. This equilibrium, on the other hand, is not efficient: the expected size of the minority group is lower than $\left\lfloor \frac{N-1}{2} \right\rfloor$ due to variance of the action selection.

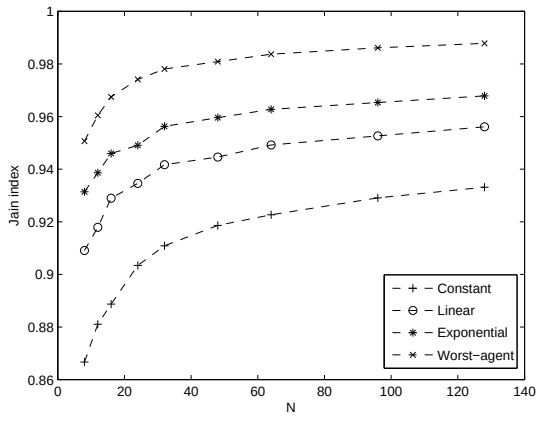


Figure 4: Jain fairness index of the channel allocation scheme for various back-off probabilities, $C = \frac{N}{2}$, $K = 2\log_2 N$

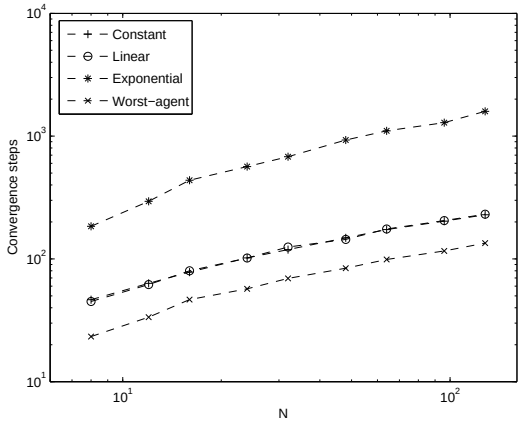


Figure 5: Convergence steps for various back-off probabilities.

Savit *et al.* ([13]) show that if the agents receive feedback on which action was in the minority, they can learn to coordinate better to achieve a more efficient outcome in a repeated minority game. They do this by basing the agents' decisions on the history of past iterations. Cavagna [2] shows that the same result can be achieved when agents base their decisions on the value of some random coordination signal instead of using the history. This is a direct inspiration for our work.

The ideas from the literature on Minority games have recently found their way into the cognitive radio literature. Mahonen and Petrova [8] present a channel allocation problem much like ours. The agents learn which channel they should use using a strategy similar to the strategies for minority games. The difference is that instead of preferring the action chosen by the minority, in the channel allocation problem, an agent prefers channels which were not chosen by anyone else. Using this approach, Mahonen and Petrova are able to achieve a stable throughput of about 50% even when the number of agents who try to transmit over a channel increases. However, each agent is essentially choosing one out of a fixed set of strategies, which they cannot adapt. Therefore, it is very difficult to achieve a perfectly efficient channel allocation.

Another, more general variant of our problem, called *dispersion game* was described by Grenager *et al.* in [6]. In a dispersion game, agents can choose from several actions, and they prefer the one which was chosen by the smallest number of agents. The authors define a *maximal dispersion outcome* as an outcome where no agent can move to an action with fewer agents. The set of maximal dispersion outcomes corresponds to the set of pure-strategy Nash equilibria of the game. They propose various strategies to converge to a maximal dispersion outcome, with different assumptions on the information available to the agents. On the contrary with our work, the individual agents in the dispersion games do not have any particular preference for the actions chosen or the equilibria which are achieved. Therefore, there are no issues with achieving a fair outcome.

Verbeeck *et al.* [14] use reinforcement learning, namely *linear reward-inaction automata*, to learn Nash equilibria in common and conflicting interest games. For the class of conflicting interest games (to which our wireless channel allocation game belongs), they propose an algorithm that allows the agents to circulate between various pure-strategy Nash equilibria, so that the outcome of the game is fair. In contrast with our work, their solution requires more communication between agents, and it requires the agents to *know* when the strategies converged. In addition, linear reward-inaction automata are not guaranteed to converge to a PSNE in conflicting interest games; they may only converge to pure strategies.

All the games discussed above, including the wireless channel allocation game, form part of the family of *potential games* introduced by Monderer and Shapley ([9]). A game is called a potential game if it admits a *potential function*. A potential function is defined for every strategy profile, and quantifies the difference in payoffs when an agent unilaterally deviates from a given strategy profile. There are different kinds of potential functions: exact (where the difference in payoffs to the deviating agent corresponds directly to the difference in potential function), ordinal (where just the sign of the potential difference is the same as the sign of the payoff difference) etc.

Potential games have several nice properties. The most important is that any pure-strategy Nash equilibrium is just a local maximum of the potential function. For finite potential games, players can reach these equilibria by unilaterally playing the best-response, no matter what initial strategy profile they start from.

The existence of a natural learning algorithm to reach Nash equilibria makes potential games an interesting candidate for our future research. We would like to see to which kind of correlated equilibria can the agents converge there, if they can use a simple correlation signal to coordinate.

7. CONCLUSIONS

In this paper, we proposed a new approach to reach desirable correlated equilibria in games. Instead of using a "smart" coordination device, as the original definition of CE assumes, we use "stupid" signal, a random integer k taken from a set $\mathcal{K} = \{0, 1, \ldots, K-1\}$, which has no a priori relation to the game. Agents then are "smart": they learn,

for each value of the coordination signal, which action they should take.

We showed a learning strategy which, for a variant of a wireless channel allocation game, converges in expected polynomial number of steps to an efficient correlated equilibrium. We also proved that this equilibrium becomes increasingly fair as K, the number of available synchronization signals, increases. We have confirmed both the fast convergence as well as increasing fairness with increasing K experimentally.

In the future work, we would like to see whether this approach ("stupid" coordination signal and "smart" learning agents) can help to reach desirable correlated equilibria of other games, such as potential games.

8. REFERENCES

[1] R. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, March 1974.

[2] A. Cavagna. Irrelevance of memory in the minority game. *Physical Review E*, 59(4):R3783–R3786, April 1999.

[3] D. Challet, M. Marsili, and Y.-C. Zhang. *Minority Games: Interacting Agents in Financial Markets (Oxford Finance)*. Oxford University Press, New York, NY, USA, January 2005.

[4] W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley, 3 edition, January 1968.

[5] D. P. Foster and R. V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2):40–55, October 1997.

[6] T. Grenager, R. Powers, and Y. Shoham. Dispersion games: general definitions and some specific learning results. In *Proceedings of the Eighteenth national conference on Artificial intelligence (AAAI-02)*, pages 398–403, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[7] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical report, Digital Equipment Corporation, September 1984.

[8] P. Mahonen and M. Petrova. Minority game for cognitive radios: Cooperating without cooperation. *Physical Communication*, 1(2):94–102, June 2008.

[9] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, pages 124–143, May 1996.

[10] J. R. Norris. *Markov Chains (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, July 1998.

[11] C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):1–29, July 2008.

[12] V. Rego. Naive asymptotics for hitting time bounds in markov chains. *Acta Informatica*, 29(6):579–594, June 1992.

[13] R. Savit, R. Manuca, and R. Riolo. Adaptive competition, market efficiency, and phase transitions. *Physical Review Letters*, 82(10):2203–2206, March 1999.

[14] K. Verbeeck, A. Nowé, J. Parent, and K. Tuyls. Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Autonomous Agents and Multi-Agent Systems*, 14(3):239–269, June 2007.