Distributed Planning in Stochastic Games with Communication

Andriy Burkov and **Brahim Chaib-draa**
DAMAS Laboratory
Laval University
G1K 7P4, Quebec, Canada
{burkov,chaib}@damas.ift.ulaval.ca

Abstract

This paper treats the problem of distributed planning in general-sum stochastic games with communication when the model is known. Our main contribution is a novel, game theoretic approach to the problem of distributed equilibrium computation and selection. We show theoretically and via experimentations that our approach to multiagent planning, when adopted by all agents, facilitates an efficient distributed equilibrium computation and leads to a unique equilibrium selection in general-sum stochastic games with communication.

Introduction

Stochastic games is a natural generalization of MDPs to a multiagent decision problem setting. In this context, the reward the agent obtains is determined by the state of the environment and the joint-action of all agents. Solving a stochastic game (SG) consists of finding a joint-policy that prescribes to each agent an action to do given the environment state and (possibly) some other information available. A solution of a (stochastic) game is usually called an *equilibrium*. In this sense, an equilibrium is a joint-policy, composed of policies of all players. This joint-policy is such that no player is interested in deviating from its own policy given that all other players stick to theirs.

When the model of the environment is known to all agents, planning is an approach to solving an SG. In contrast to the learning, a model-free trial-error based approach, there has been a relatively small number of algorithms proposed in the planning context (Shapley 1953; Vrieze 1987; Kearns, Mansour, & Singh 2000). The first algorithm created for planning in SGs (Shapley 1953) deals with SGs having a particular structure: two-player strictly competitive (zero-sum) games. So, this algorithm cannot be easily applicable to any game. The second algorithm (Vrieze 1987) is essentially Fictitious play adapted to the SG context. Since it inherits convergence properties of Fictitious play, it can only converge in zero-sum games and in the games that are solvable by iterated strict dominance. Furthermore, Fictitious play is not guaranteed to converge to an equilibrium: often it converges only in estimates of the opponents' strategies (Fudenberg & Levine 1999).

For us, the most interesting planning approach is the one proposed by Kearns, Mansour, & Singh (2000) and called FINITEVI. It is an algorithm of finite value iteration in SGs. While two other algorithms of planning in SGs, which we have mentioned above, have limitations relative to the reward structure of the game that can be solved by them, FINITEVI has been proven to yield an equilibrium in any SG. This algorithm, however, requires a *high level of centralization*. Indeed, it requires the presence of an *oracle* that is always available during the planning process (a) to *select an equilibrium* (if they are numerous) in each state at each iteration and (b) to *assign it* to all agents thereafter.

Communication is considered by many researchers as a natural way to avoid centralization in decentralized systems. In this paper, we propose an original way of avoiding centralization in planning algorithms and facilitating distributed equilibrium computation by means of communication between agents. Our main contribution consists in a novel, game theoretic approach to the problem of distributed equilibrium *computation and selection* in SGs with multiple equilibria. We show theoretically and via experimentations that our decentralized planning approach, called FINITEVI-COM, when adopted by all agents, leads to a unique Nash equilibrium in any general-sum SG without need (in principle) that the particular algorithms adopted by the agents be the same. To our knowledge, currently there are no algorithms capable of having such properties without implying additional strong restrictions on the environment. These restrictions can be, for example, a need for a centralization in the equilibrium selection task (Hu & Wellman 2003; Kearns, Mansour, & Singh 2000), or a requirement of a particular reward structure, such as zero-sum games, team games and others (Littman 1994; Wang & Sandholm 2002).

Stochastic games

As we have already mentioned above, SGs could be viewed as a generalization of MDPs to multiagent systems. In SGs, there are n agents. We will refer to j to denote some agent chosen among n, and to $-j$ to denote the set of all other agents $1 \dots n$ except the agent j. Each agent j has a set $\mathcal{A}^j$ of available *actions* (or pure strategies) denoted as a^j. When agents simultaneously execute their actions, we say that a *joint-action* has been executed. A joint-action is a vector $\mathbf{a} = (a^1, a^2, \dots, a^n)$ containing one simple action per agent. The set $\mathbf{A} \subseteq \mathcal{A}^1 \times \mathcal{A}^2 \times \dots \times \mathcal{A}^n$ is called *joint-action space*, with $\mathbf{a} \in \mathbf{A}$. The environment has a finite set of states $\mathbf{S}$, with a vector $\mathbf{s} = (s^1, s^2, \dots, s^n) \in \mathbf{S}$ called

joint-state. This vector is composed of respective personal states of agents, and there is a special state $\mathbf{s}_0$ called *start state*. It is assumed that the game always starts from $\mathbf{s}_0$. There is a transition function $T : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \mapsto [0, 1]$ which defines a probability of transition from one state to another. This function has the following property: $\sum_{\mathbf{s}'} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') = 1 \ \forall \mathbf{s} \in \mathbf{S}, \forall \mathbf{a} \in \mathbf{A}$. Similarly to MDPs, once an action is executed, the agents receive a real-valued *reward* (also called immediate utility) from the environment. This reward is defined by the reward functions R^j, one for each agent, where $R^j : \mathbf{S} \times \mathbf{A} \mapsto \mathbb{R}$ is a reward function of agent j.

One can consider the states of an SG as being matrix (or normal-form) games (Fudenberg & Levine 1999). Indeed, each state $\mathbf{s}$ of the environment has its own reward function $R^j(\mathbf{s}, \cdot)$ for each agent j and the value that this agent obtains in each state depends on the actions played simultaneously by all agents in this state.

Solving an SG consists of finding a *joint-policy* π that assigns a strategy to execute to each agent. Since π is a joint-policy, therefore it is a vector $\pi = (\pi^1, \pi^2, \ldots, \pi^n)$ containing *individual policies* (or just "policies") for each agent. A policy π^j, in turn, is a rule assigning to an agent j a *strategy*, $\pi^j(\mathbf{s})$, to execute in each state $\mathbf{s}$. These strategies can be pure or mixed. A pure strategy, as we already noted, is a simple (i.e., non-joint) action. A mixed strategy is a probability distribution over simple actions.

Each joint-policy π has a set $\{U^j(\pi) : j = 1 \ldots n\}$ of real-valued utilities associated with it. According to these utilities agents can prefer one joint-policy to another. A *Nash equilibrium* is a joint-policy $\hat{\pi}$, in which every agent among n has no interest in unilaterally changing its policy given the policies of other agents are unchanged. This means that for each agent j, the utility $U^j(\hat{\pi})$ is not lower than the utility of any other joint-policy in which player j plays some other policy whereas the other agents play according to $\hat{\pi}$. More formally, a policy $\hat{\pi} = (\hat{\pi}^j, \hat{\pi}^{-j})$ is a Nash equilibrium if and only if $\forall j$ and $\forall \pi^j \neq \hat{\pi}^j$: $U^j(\hat{\pi}) \geq U^j(\pi^j, \hat{\pi}^{-j})$.

A Nash equilibrium strategy $\hat{\pi}^*$ is said to be non-Pareto dominated by no other Nash equilibrium of the game if and only if $\forall \pi \neq \hat{\pi}^* \ \exists j$ such that $U^j(\pi) < U^j(\hat{\pi}^*)$. Any stochastic game has at least one Nash equilibrium, while there can be more than one such equilibrium in a game. Such a multiplicity of equilibria is a matter of difficulties of all planning algorithms for both matrix and stochastic games. This is because different agents can prefer different equilibria. Indeed, this is an important open problem in both multiagent and game theory communities and is referred to as an "equilibrium selection" problem (Myerson 1991). As we will show below, our approach permits overcoming this difficulty in SGs with communication (Com-SGs).

Planning in stochastic games

As in MDPs, there can be two types of planning in SGs: planning with a finite horizon and planning with an infinite horizon[1]. A horizon of planning is a number of time steps (or transitions of the environment between states) starting from which the agent becomes indifferent to the rewards it obtains from the environment. When the horizon is *finite*, the utility for a given agent, say j, of a sequence $seq = [\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \mathbf{s}_2, \mathbf{a}_2, \ldots]$ of state transitions (possibly infinite) is given as follows:

$$U^j_H(seq) = R^j(\mathbf{s}_0, \mathbf{a}_0) + R^j(\mathbf{s}_1, \mathbf{a}_1) + \ldots + R^j(\mathbf{s}_H, \mathbf{a}_H)$$

where H is the length of horizon. Note that in this case, $U^j_H(seq) = U^j_{H+k}(seq), \ \forall k > 0$.

An agent in an SG with an *infinite* horizon always has a non-zero interest to all future rewards. To maintain preferences between different sequences of possible state transitions, the agent uses a discount factor permitting it to calculate a utility of any sequence of joint-state–joint-action pairs so as not to obtain infinite values.

In short, the FINITEVI algorithm by Kearns, Mansour, & Singh (2000) for planning in SGs with *finite* horizon works as follows. In every joint-state $\mathbf{s} \in \mathbf{S}$, a list of Q-values is maintained. At each iteration, these Q-values are updated using a form of Bellman update:

$$Q^j(\mathbf{s}, \mathbf{a}, t) \leftarrow R^j(\mathbf{s}, \mathbf{a}) + \sum_{\mathbf{s}'} T(\mathbf{s}, \mathbf{a}, \mathbf{s}') u^j_f(\mathbf{s}', t-1)$$

where $u^j_f(\mathbf{s}, t-1)$ is the utility of an equilibrium of the matrix game composed of Q-values calculated on the previous iteration in the state $\mathbf{s}$. This value is returned by a certain function f, called *Nash selection function*. This function constructs a matrix game from the Q-values of all agents in the state $\mathbf{s}$ and then solves this game so as to find a unique equilibrium. If the game has several equilibria, the function f must choose one of them and communicate it to all agents. As one can see, this is an obvious centralization point of the given algorithm which can be viewed as an oracle. A need of an oracle is generally seen as a drawback to the application of this algorithm in distributed systems.

In their paper, Kearns, Mansour, & Singh claim that an algorithm of value iteration in SGs with *infinite horizon* that would converge in arbitrary general-sum games cannot exist. (This claim has been recently justified by Zinkevich, Greenwald, & Littman (2005).) On the other hand, they have shown that a *finite horizon* value iteration can converge to a Nash equilibrium in general-sum SGs (their FINITEVI always converges provided an arbitrary Nash selection function). We will rely on this result by proposing our algorithm of planning in Com-SGs with finite horizon.

Planning with communication

In this section, we present our approach to planning in Com-SGs. As we already mentioned above, the main disadvantage of FINITEVI is its strong centralization originating from using the function f, which should find a unique equilibrium and communicate it to all agents. Such a centralization is often undesirable for the following reasons. First of all, as we have already noted, this function can be viewed as an oracle, which, informally speaking, "knows better" what is "good" for all agents. This property is often difficult to assert, especially when different agents can have

[1] Sometimes throughout the paper, we will also say "SG with finite or infinite horizon" in the same sense.

different preferences about what is "good" and what is "not so good" for them. One can imagine a situation when each agent has its own function f^j, which would permit avoiding the need of an oracle. In this case, however, the problem still persists, since now all agents, in order to select the same equilibrium, are required to have the same function f^j, i.e., $f^1 = f^2 = \ldots = f^n$, which in general cannot be assured in distributed systems. Besides, even if such function (the same for all agents) can exist, it must be deterministic. This means that if $f^j(\mathbf{s}, t)$ returns to some agent j a strategy $\hat{\pi}^j(\mathbf{s}, t)$ pertinent to some Nash equilibrium $\hat{\pi}(\mathbf{s}, t)$, then all other agents must receive, from their respective Nash selection functions, strategies belonging to the same equilibrium $\hat{\pi}(\mathbf{s}, t)$. Obviously, in general (not explicitly cooperative) case such property is also not easy to guarantee.

It is required to note that a similar problem is observed in some other algorithms for SGs. For example, in Hu & Wellman's Nash-Q learning algorithm (Hu & Wellman 2003) the agents are required to always choose the first computed equilibrium, or the second, and so on. I.e., the agents not only need to always make the same decisions but also to compute always the same sets of equilibria. This makes impossible to the agents to use different algorithms of equilibrium computation or certain efficient non-deterministic methods.

In this paper, to avoid such a centralization and add some other desirable properties which a decentralized system could have (such as a distributed solution computation) we propose a communication based game theoretic equilibrium selection approach for value iteration in SGs. However, instead of using a unique and centralized function f computing and selecting a unique equilibrium in a state for all agents, we divide the equilibrium selection process into two phases. The first phase is an "equilibrium computation" phase. During this phase, each agent computes a (not necessarily complete) set of equilibria for a joint-state–time pair by using any known equilibrium computation technique (not necessarily the same for all agents). The second phase is a "communication" phase. During that phase, the agents communicate between them in order to (possibly) share their computed equilibria and to select a unique equilibrium among those calculated. According to our approach, the communication phase is held in a form of a matrix game. This (new) game, which we call "communication game", is dynamically constructed from the equilibria computed by the agents during the equilibrium computation phase. In the next section, we will explain in detail how this new game is constructed and played. Then, we will show that once a game playing process in communication game has converged to a (pure) equilibrium, this equilibrium corresponds to a unique (possibly, mixed) equilibrium of the original, stochastic game.

Communication games

In this subsection, we present our new FINITEVICOM algorithm of distributed planning in Com-SGs with finite horizon, we define the notion of communication games and give an example of such a game.

Equilibrium selection as a game The FINITEVICOM algorithm of finite horizon value iteration in Com-SGs is presented in Algorithm 1.

```
1:  function FiniteViCom(H, C, P)
2:      returns: a joint-policy.
3:      inputs: H, a horizon; C, a vector of equilibrium computation algorithms; P, a vector of game playing algorithms.
4:      t ← 0
5:      while t ≤ H do
6:          for all s ∈ S do
7:              for all j = 1 ... n do
8:                  for all a ∈ A do
9:                      if t = 0 then
10:                         Q^j(s, a, t) ← R^j(s, a)                    ▷ Initialization
11:                     else
12:                         Q^j(s, a, t) ← Σ_{s'} T(s, a, s') u^j_E(s', t−1) + R^j(s, a)    ▷ Q-value update
13:             E^j ← C^j(s, t)
14:         E = (E^1, E^2, ..., E^n)
15:         π^j(s, t) ← Play(P, s, t, E)
16:     t ← t + 1                                                       ▷ Next iteration
17:     return π = (π^1, π^2, ..., π^n)
```

Algorithm 1: Finite horizon value iteration algorithm to compute a Nash equilibrium in Com-SGs.

The algorithm uses three input parameters: H, the horizon of planning, and two others, $\mathbf{C}$ and $\mathbf{P}$, that need to be described in more detail. $\mathbf{C} = (C^1, C^2, \ldots, C^n)$ is a vector containing algorithms of equilibrium computation, C^j, one for each agent. In practice, C^j may be any algorithm that is able to compute all or just a subset of equilibria of a normal-form game, given the game matrix (McKelvey & McLennan 1996). $\mathbf{P} = (P^1, P^2, \ldots, P^n)$ is a vector of game playing algorithms for matrix games. This vector contains one algorithm P^j for each agent. Similarly to vector $\mathbf{C}$, the algorithms P^j of vector $\mathbf{P}$ can be different for different agents. A particular algorithm P^j may be any known algorithm of game playing in matrix games. For example, such algorithm can be Fictitious Play, Adaptive play, Joint-Action Learner, PHC (Fudenberg & Levine 1999; Young 1993; Bowling & Veloso 2002) or others.

During the value iteration, the set of Q-values is updated using a Bellman equation (line 12) in which $u^j_{\mathcal{E}}(\mathbf{s}, t-1)$ is the value of the equilibrium selected in the state $\mathbf{s}$ at the previous iteration. In each state $\mathbf{s}$ and at each iteration t of FINITEVICOM, a unique equilibrium selection is held as follows. First, each agent uses its algorithm C^j to compute a set $\mathcal{E}^j$ of equilibria of the matrix game, which is given by the Q-values, $Q^j(\mathbf{s}, \mathbf{a}, t)$, of the state $\mathbf{s}$ at iteration t (line 13). Then, during the communication phase (the function $Play$, line 15 of Algorithm 1), the agents use their game playing algorithms P^j to play a *communication game* $CG(\mathbf{s}, t)$ against all other agents. In this communication game, the actions available for agents are the equilibria from their respective sets $\mathcal{E}^j$. Notice that throughout this paper, we will call these actions "communication actions" to distinguish them from the actions available to the agents in the Com-SG.

On a game turn, each agent communicates an equilibrium from its set $\mathcal{E}^j$ (or, we can also say that it "executes

a communication action") to all other agents and observes the communication actions played by others. If all players have played the same equilibrium as the one played by j, the reward the agent j obtains after this play is its respective utility in the equilibrium corresponding to the communication action played. In all other cases, all players obtain a zero-reward. (For simplicity of presentation, we assume that all equilibria of the original SG are non-negative in any state. The approach can be easily extended to the games with negative equilibria. To do that, the reward associated in $CG(\mathbf{s}, t)$ with a joint-communication action in which not all players play the same equilibrium of the original game should be set for player j to the lower bound on the utility of this player in the original SG.) The game $CG(\mathbf{s}, t)$ is played by the agents repeatedly until convergence to a pure strategy equilibrium in $CG(\mathbf{s}, t)$ (see next section for convergence results). This equilibrium, unique, will then be used to do value iteration in the original SG.

To demonstrate how an equilibrium is being chosen during the communication phase, let us show an example. For simplicity, consider a two-player case and suppose that in a state $\mathbf{s}$ at iteration t the sets of equilibria the agents have calculated using their algorithms C^1 and C^2 are as follows: $\mathcal{E}^1 = \mathcal{E}^2 = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. In that case, the game $CG(\mathbf{s}, t)$ will constitute a set of two matrices, one per agent. Each matrix will have the following form:

$$\begin{pmatrix} u^j(\mathbf{e}_1) & 0 & 0 \\ 0 & u^j(\mathbf{e}_2) & 0 \\ 0 & 0 & u^j(\mathbf{e}_3) \end{pmatrix}$$

Assuming that player 1 plays by selecting rows of the matrix and player 2 selects its columns, $u^j(\mathbf{e}_k)$ is the reward the agent j obtains in the communication game when both players play a communication action corresponding to the same equilibrium $\mathbf{e}_k$ of the original SG. This reward is simply the utility of equilibrium $\mathbf{e}_k$ for player j according to the Q-values in state $\mathbf{s}$ at iteration t.

Distributed equilibrium computation During the game playing process, the agents are interchanging their precomputed equilibria in order to eventually select a unique equilibrium. If all the agents used the same deterministic algorithm of equilibrium computation, it would be easy to assure that $\mathcal{E}^1 = \mathcal{E}^2 = \ldots = \mathcal{E}^n$ as in the above example. In that case, the game matrix $CG(\mathbf{s}, t)$ would be guaranteed to contain at least one joint-communication action that all agents would prefer. In practice, however, each agent can use its own equilibrium computation method that can be able to compute only a subset of equilibria (McKelvey & McLennan 1996) and can be non-deterministic, i.e., to compute different subsets of equilibria after each run. Such methods have recently been observed to be very fast in practice (Pavlidis, Parsopoulos, & Vrahatis 2005).

In the above example, if in some state at some iteration $\mathcal{E}^1 \cap \mathcal{E}^2 = \varnothing$, the players could never select an equilibrium. To avoid this and to profit from the distributed character of the problem, the agents must be able to put "unknown" equilibria, communicated by the other agents during communication game playing, into their equilibrium sets. We say that an equilibrium $\mathbf{e}^k$, $\mathbf{e}^k \notin \mathcal{E}^j$, communicated by some agent $k = 1 \ldots j-1, j+1 \ldots n$ can be *safely* put by the agent j into its own set $\mathcal{E}^j$ of equilibria if the following two conditions hold: (1) $\mathbf{e}^k$ can be verified by j to be a true equilibrium of $CG(\mathbf{s}, t)$ in a reasonable time (for example, polynomial in the game size) and (2) $\mathbf{e}^k$ is non-Pareto-dominated by no other equilibrium from $\mathcal{E}^j$.

The second condition is obvious: no agent is interested in a convergence to an equilibrium Pareto-dominated by some other known equilibrium. This property is easily verifiable[2] by comparing the utility of $\mathbf{e}^k$ with the utilities of equilibria of the set $\mathcal{E}^j$. The following theorem satisfies the first condition as well.

Theorem 1. *Let MG be a matrix game where n is the number of players, $\mathcal{A}^j$ is the action set of player j, $j = 1 \ldots n$, and $\mathbf{A}^{-j}$ is the joint-action set of all players except j. In any game MG, a pure and a mixed Nash equilibrium can be verified in $\mathcal{O}(n|\mathcal{A}^j||\mathbf{A}^{-j}|)$ $\forall j$.*

Proof. We will prove this theorem by providing a polynomial time algorithm to verify a given joint-strategy to be an equilibrium of a matrix game (Algorithm 2).

```
1:  function VERIFYEQUILIBRIUM(e, GM)
2:      returns: true or false.
3:      inputs: e, a pure or a mixed joint-strategy; GM, a
        game matrix.
4:      for all j = 1 ... n do
5:          Save in tmp the utility of e for player j according
            to GM.
6:          for all b^j ∈ A^j do
7:              if e is pure then
8:                  Let e = a = (a^j, a^{-j}) ∈ A.
9:                  Set U^j(b^j) ← u^j(b^j, a^{-j}).
10:             else
11:                 Compute U^j(b^j) using Equation (1).
12:             if U^j(b^j) > tmp then
13:                 return false.
14:     return true.
```

Algorithm 2: The algorithm to verify a Nash equilibrium.

In the above algorithm, $U^j(b^j)$ is the utility for player j of playing a pure action $b^j \in \mathcal{A}^j$. There can be two cases: $\mathbf{e}$, the joint-strategy to verify (i.e., an "unknown" equilibrium communicated by certain opponent agent) can be pure or mixed. If $\mathbf{e}$ is pure (i.e., $\mathbf{e}$ is a certain joint-action $\mathbf{a} = (a^j, \mathbf{a}^{-j}) \in \mathbf{A}$) then $U^j(b^j)$ is simply the utility for the player j of some joint-action in $\mathbf{A}$ according to the game matrix GM (this utility is denoted in Algorithm 2 as $u^j(b^j, \mathbf{a}^{-j})$ $\forall b^j \in \mathcal{A}^j$). In the mixed strategy case, $U^j(b^j)$ is the *expected* utility of playing the pure strategy b^j by the player j given that the other players play according to the mixed strategy equilibrium $\mathbf{e}$. This expected utility is given by,

$$U^j(b^j) = \sum_{\mathbf{b}^{-j} \in \mathbf{A}^{-j}} u^j(b^j, \mathbf{b}^{-j}) \Pr(\mathbf{b}^{-j}|\mathbf{e}) \qquad (1)$$

[2] More precisely, the verification time is in $\mathcal{O}(|\mathcal{E}^j|^2)$.

where $\Pr(\mathbf{b}^{-j}|\mathbf{e})$ is the probability that a certain joint-action $\mathbf{b}^{-j}$ will be played by the other players according to the equilibrium $\mathbf{e}$.

Thus, due to two nested "for" loops and one nested summation over $\mathbf{A}^{-j}$, we can conclude that the time required to verify a Nash equilibrium (pure or mixed) is in $\mathcal{O}(n|\mathcal{A}^j||\mathbf{A}^{-j}|)\ \forall j$. □

Convergence results

In this section, we present the main theoretical results concerning our approach to finite horizon planning in Com-SGs.

Convergence in stochastic games

Kearns, Mansour, & Singh have shown that FINITEVI is guaranteed to converge to a unique Nash equilibrium in any SG with finite horizon by proving the following theorem.

Theorem 2 (Kearns, Mansour, & Singh (2000))**.** *Let SG be a two-player stochastic game (the extension to n-player games with $n > 2$ is straightforward), let f be any Nash selection function, and let H be a horizon. Then the joint-policy $\pi = (\pi^1, \pi^2)$ output by the* FINITEVI*(H, f) algorithm is a Nash equilibrium for SG.*

From the above result of Kearns, Mansour, & Singh, the following Theorem can be formulated for the FINITEVICOM algorithm.

Theorem 3. *Let SG be a two-player stochastic game (the extension to n-player games with $n > 2$ is straightforward), and let H be a horizon. If the algorithms in vector $\mathbf{P}$ have a property of mutual convergence to a pure strategy equilibrium in any communication game, then the joint-policy $\pi = (\pi^1, \pi^2)$ output by the* FINITEVICOM*$(H, \mathbf{C}, \mathbf{P})$ algorithm is a Nash equilibrium for SG.*

Proof. (Sketch) As one can observe, the FINITEVICOM algorithm inherits the convergence properties of FINITEVI, given the convergence of the former to an equilibrium in all communication games played during the planning process. This is true, since in that case the communication game of FINITEVICOM plays a part of the Nash selection function f used in FINITEVI. As soon as, according to the Theorem 2, FINITEVI converges to a Nash equilibrium, FINITEVICOM will also do so. □

Convergence in communication games

As noted above, in order to assure fulfilment of the conditions of Theorem 3, the process of equilibrium selection during the communication phase (the function $Play$ at the line 15 of Algorithm 1) must yield a unique pure equilibrium, which then will be used as a part of agents' policies. Besides, the value of the equilibrium selected on the previous iteration will also be used in the Q-value update (line 12).

As one can observe, in communication games there can be pure and mixed equilibria. All pure equilibria lay on the diagonal of the communication game matrix. The mixed equilibria can be formed out of an arbitrary number of pure equilibria by playing the communication actions corresponding to those equilibria according to some nontrivial probability distribution. The following theorem shows that all such mixed equilibria are Pareto-dominated by the pure equilibria used to construct them. Therefore, using game playing algorithms converging to mixed equilibria in communication games is not only impractical for our purposes (we want agents to choose a unique equilibrium in any state–iteration of the original stochastic game, and not a mixture of equilibria) but also it is not rational from a game theoretical viewpoint. Such algorithms hence can be excluded from consideration in relation to using them in communication games.

Theorem 4. *Let $CG(\mathbf{s}, t)$ be a two-player communication game (the extension to n-player games with $n > 2$ is straightforward). Any mixed strategy equilibrium in $CG(\mathbf{s}, t)$ is Pareto-dominated by each pure strategy equilibrium from its support.*

Proof. Let $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ denote at once pure equilibria, used to construct a certain mixed equilibrium, and corresponding communication actions of players. This set of communication actions is called a *support* of the mixed equilibrium. Obviously, each action in the support is played with a non-zero probability, because if not (i.e., if there was an action $\mathbf{e}_l$, $1 \leq l \leq k$, not played with a non-zero probability) that action could be removed from the support without changing the mixed equilibrium. Let for simplicity denote players i and j, where $(i, j) \in \{(1, 2), (2, 1)\}$.

Proof by contradiction. Let $\mathbf{e}^l$, $1 \leq l \leq k$, be a pure equilibrium whose the value for both players is lower than the value of the mixed equilibrium. For each player, in the mixed equilibrium there is a non-zero probability $p^i(\mathbf{e}_l) < 1$ associated with the action $\mathbf{e}^l$. According to the definition of a communication game, the value $v^i(\mathbf{e}_l)$ of the pure action $\mathbf{e}_l$ played by player i is given as:

$$\begin{aligned} v^i(\mathbf{e}_l) &= p^j(\mathbf{e}_l)u^i(\mathbf{e}_l) + \sum_{1 \leq l' \leq k \wedge l' \neq l} p^j(\mathbf{e}_{l'}) \cdot 0 \\ &= p^j(\mathbf{e}_l)u^i(\mathbf{e}_l) \\ &< u^i(\mathbf{e}_l) \end{aligned}$$

Recall a property of a mixed equilibrium: for each player, every pure action in the support of this mixed equilibrium has the same value as the value of the equilibrium. Thus, the above inequality states that the mixed equilibrium having the action $\mathbf{e}_l$ in its support has the value lower than the value of the pure equilibrium $\mathbf{e}_l$. Since we did not precise the way according to which the action $\mathbf{e}_l$ has been chosen, it is clearly a contradiction to the original assumption. □

There is a number of game playing algorithms possessing a property of convergence to an equilibrium in matrix games. While the performance of different game playing algorithms against each other has recently been studied in some cases, the formal proofs of convergence to an equilibrium are typically given in the literature for the two-player and/or two-action case.

In FINITEVICOM, such well-known algorithms for matrix games as Fictitious play (Fudenberg & Levine 1999), IGA (Singh, Kearns, & Mansour 1994), GIGA (Zinkevich 2003), Adaptive play (Young 1993), ReDVaLeR (Banerjee & Peng 2004), AWESOME (Conitzer & Sandholm 2007),

Joint-Action Learner (Claus & Boutilier 1998) and even a single-agent Q-learning could be used as a game playing technique for communication games. To present our main theoretical results, we have opted for Adaptive play (AP) algorithm (Young 1993) as a game playing technique for communication games. Our choice is dictated by the following considerations. Some algorithms (such as Joint-Action Learner (Claus & Boutilier 1998) or a single agent Q-learning) although tested in a wide range of different practical situations still do not have formal convergence proofs. The theoretical guarantees that some other algorithms mentioned above possess (e.g., Fictitious play, IGA and GIGA) are based on the assumptions that are too restrictive (two-player or two-action case, for example). ReDVaLeR and AWESOME, could indeed be taken as a game playing technique for communication games. However, they also require some important preconditions to be satisfied. Furthermore, their capabilities and structural complexity surpass the minimal requirements with which such technique should comply in order to be an appropriate game playing technique for communication games. Finally, the question of convergence of all these algorithms to a pure strategy Nash equilibrium in communication games is itself an important theoretical and practical issue to explore. In contrast, AP is structurally simple, it requires neither the game matrix to be known in advance (nor even to be able to be explicitly constructed) nor the observability of the opponent's actual strategy. And, as we will show below, it does converge to a pure Nash equilibrium in any communication game.

Adaptive play in communication games

Adaptive play works as follows. Let $\mathbf{a}_l \in \mathbf{A}$ be a joint-action played at iteration l by all players. Fix integers k and m such that $1 \leq k \leq m$. While $l \leq m$, adaptive player j randomly chooses its actions and plays them.

Let $\mathcal{H}_l = \mathbf{a}_{l-m}^{-j}, \mathbf{a}_{l-m+1}^{-j}, \ldots, \mathbf{a}_{l-1}^{-j}$ denote the m most recent joint-actions played by the opponent agents so far. Starting from $l = m+1$, player j randomly and without replacement draws k samples from $\mathcal{H}_l$ and saves them in the set $\hat{\mathcal{H}}_l$ ($\hat{\mathcal{H}}_l \subseteq \mathcal{H}_l$). Let $C(\mathbf{a}^{-j}|\hat{\mathcal{H}}_l)$ be the number of times a certain opponents' joint-action $\mathbf{a}^{-j}$ appears in the set $\hat{\mathcal{H}}_l$. Let $u^j(\mathbf{a})$ be the reward agent j obtains when the joint-action $\mathbf{a} \in \mathbf{A}$ is played. The expected utility $U^j(a^j)$ of playing a simple action $a^j \in \mathcal{A}^j$ is computed by player j as follows:

$$U^j(a^j) = \sum_{\mathbf{a}^{-j} \in \mathbf{A}^{-j}} u^j(a^j, \mathbf{a}^{-j}) \frac{C(\mathbf{a}^{-j}|\hat{\mathcal{H}}_l)}{k}$$

The set of best responses, denoted as $\mathcal{BR}_l^j$, is then formed as $\mathcal{BR}_l^j = \{a^j : a^j = \operatorname{argmax}_{b^j \in \mathcal{A}^j} U^j(b^j)\}$. At each iteration l, the adaptive player j plays an action randomly drawn from $\mathcal{BR}_l^j$ by giving a non-zero probability to any action $a^j \in \mathcal{BR}_l^j$ to be drawn. As we already mentioned, the convergence of AP to an equilibrium in self-play has been proven for a class of games called "weakly acyclic" (Young 1993). Let us now define this notion.

Definition 1 (Weakly acyclic game (Young 1993))**.** *Let MG be a n-player matrix game. Let $\mathcal{BR}^j(\mathbf{a}^{-j})$ denote the set of best responses of agent j to an opponents' joint-action $\mathbf{a}^{-j}$. The best-response graph constructed on MG has $\mathbf{A}$ as its set of vertices and there is a directed edge between vertices $\mathbf{a} = (a^j, \mathbf{a}^{-j})$ and $\mathbf{a}' = (a'^j, \mathbf{a}'^{-j})$ if and only if (i) $\mathbf{a} \neq \mathbf{a}'$ and (ii) $\exists! j = 1 \ldots n : a'^j \in \mathcal{BR}^j(\mathbf{a}^{-j}) \wedge \mathbf{a}'^{-j} = \mathbf{a}^{-j}$. The game MG is said to be weakly acyclic if, in its best-response graph from any initial vertex $\mathbf{a}$, there exists a directed path to some vertex $\mathbf{a}^*$ from which there is no outgoing edge.*

In the above definition, the vertex $\mathbf{a}^*$ is essentially a Nash equilibrium in pure strategies.

Theorem 5. *Any communication game constructed as described in the previous section is weakly acyclic.*

Proof. Let us first limit ourselves to a two-player case. We will show next that the result can be extended to the n-player case as well. Recall the example of the communication game matrix of agent j in state $\mathbf{s}$:

$$\begin{pmatrix} u^j(\mathbf{e}_1) & 0 & 0 \\ 0 & u^j(\mathbf{e}_2) & 0 \\ 0 & 0 & u^j(\mathbf{e}_3) \end{pmatrix}$$

In the above game, we assume that both players, j and $-j$ have the same sets of equilibria, i.e., $\mathcal{E}^j = \mathcal{E}^{-j} = \mathcal{E}$. Indeed, if these sets were different, i.e., if there existed a non-empty set $\mathcal{E}' = (\mathcal{E}^j \cup \mathcal{E}^{-j}) \setminus (\mathcal{E}^j \cap \mathcal{E}^{-j})$, the communication actions corresponding to the equilibria of the set $\mathcal{E}'$ would never be played simultaneously by both players and, hence, both players would always obtain zero after having played them. In the terms of the best response graph, this situation corresponds to a vertex $\mathbf{a}_l$ of the best response graph from which there is always a path to some other vertex resulting from the communication actions of the set $\mathcal{E}^j \cap \mathcal{E}^{-j}$. Therefore, in our proof we can limit ourselves to a case $\mathcal{E}^j = \mathcal{E}^{-j} = \mathcal{E}$.

By observing the structure of a communication game, $CG(\mathbf{s}, t)$, note that in such a game, there are always equilibria (of this, communication game) in pure strategies and all these equilibria lie on the diagonal of the game matrix. This is true because if one player has an intention to play a certain communication action (some equilibrium e from the set $\mathcal{E}$), another player cannot do better than to play its communication action corresponding to the same equilibrium e. Therefore, the resulting joint-communication action will be a Nash equilibrium of the communication game and will necessarily lie on the diagonal of the game matrix.

Now, to show that $CG(\mathbf{s}, t)$ is weakly acyclic, we must consider two cases: 1) current vertex of the best-response graph, $\mathbf{a}_l \in \mathcal{E} \times \mathcal{E}$, corresponds to a diagonal element of matrix $CG(\mathbf{s}, t)$ and 2) $\mathbf{a}_l$ does not correspond to a diagonal element of $CG(\mathbf{s}, t)$. In the case 1), $\mathbf{a}_l$ corresponds to a vertex $\mathbf{a}^*$ of the best response graph since there cannot be outgoing edge from $\mathbf{a}_l$ (it is already an equilibrium). In the case 2), agent j has only one communication action $a'^j \in \mathcal{E}$ in its best-response set $\mathcal{BR}^j(\mathbf{a}_l^{-j})$. This communication action a'^j is such that $(a'^j, \mathbf{a}'^{-j})$ is a diagonal element $\mathbf{a}'$ of matrix $CG(\mathbf{s}, t)$ and the following two conditions hold (i) $(a_l^j, \mathbf{a}_l^{-j}) \neq (a'^j, \mathbf{a}'^{-j})$ and (ii) $\mathbf{a}'^{-j} = \mathbf{a}_l^{-j}$. Hence, in both

cases there exists a directed path from the initial vertex to a vertex $\mathbf{a}^*$ from which there is no outgoing edge. Therefore, by definition any two-player communication game $CG(\mathbf{s}, t)$ is weakly acyclic.

Now, consider an n-player case. The difference between this situation and the two-player case considered above lies in the case 2). We must show that if the current vertex $\mathbf{a}_l$ of the best-response graph does not correspond to a diagonal element of $CG(\mathbf{s}, t)$, then there is a directed path from $\mathbf{a}_l$ to some $\mathbf{a}^*$ from which there is no outgoing edge. Let's denote by $k,\ k = 1 \ldots |\mathcal{E}|$, the k-th communication action available to a player, and by $\mathbf{k} \in \times_j \mathcal{E}$ the joint-communication action in which $\forall a^j \in \mathcal{E}, a^1 = a^2 = \ldots = a^n = k$. In this context, $\mathbf{k}$ is a diagonal element of the communication game matrix and hence it corresponds to a vertex $\mathbf{a}^*$ of the best response graph from which there is no outgoing edge. Take note that if the current vertex $\mathbf{a}_l$ of the best response graph is not $\mathbf{k}$ for a certain k then there exists a player $j = 1 \ldots n$ such that $a_l^j \neq k$. Let's denote by $\mathcal{D}(\mathbf{a}_l, k)$ the set of all such players with respect to k. We will say that a joint-communication action $\mathbf{a}$ is closer to some $\mathbf{k}$ than some other action $\mathbf{a}' \in \times_j \mathcal{E}$ if and only if $|\mathcal{D}(\mathbf{a}, k)| < |\mathcal{D}(\mathbf{a}', k)|$. Remark now that if there is an edge in the best-response graph from $\mathbf{a}_l$ to some vertex $\mathbf{a}'$ then, necessarily, $|\mathcal{D}(\mathbf{a}', k)| = |\mathcal{D}(\mathbf{a}_l, k)| - 1$ for some k. If $|\mathcal{D}(\mathbf{a}_l, k)| \neq 1\ \forall k$, then for each player $j = 1 \ldots n$ the set of best responses to $\mathbf{a}_l^{-j}$ contains all available actions since the utility of each best response is 0. Hence, there will necessarily be an edge in the best response graph from $\mathbf{a}_l$ to some $\mathbf{a}'$ such that $|\mathcal{D}(\mathbf{a}', k)| = |\mathcal{D}(\mathbf{a}_l, k)| - 1$ for some k. In the other words, if $|\mathcal{D}(\mathbf{a}_l, k)| > 1$ there is always an edge from $\mathbf{a}_l$ to some vertex, which is closer to an equilibrium. If the current vertex $\mathbf{a}_l$ is such that for some k, $|\mathcal{D}(\mathbf{a}_l, k)| = 1$ (i.e., there is exactly one player j such that $a_l^j \neq k$) then there will be exactly one edge from $\mathbf{a}_l$ to $\mathbf{k}$ in the best response graph. As we have noted above, since $\mathbf{k}$ is an equilibrium, there is no outgoing edge from it and, hence, it corresponds to the vertex $\mathbf{a}^*$ of the best response graph. The latter observation permits us to conclude that the result of the Theorem is extended to the n-player case. □

Young (1993) has shown that Adaptive play converges to a Nash equilibrium in any weakly acyclic game by proving the following theorem.

Theorem 6 (Young (1993))**.** *Let MG be a weakly acyclic n-player normal-form game. If the parameters of AP, k and m, are such that $k \leq m/(L_G + 2)$, then AP converges to an equilibrium w.p. 1.*

In the above theorem, L_G is the length of the shortest directed path in the best response graph from a vertex to a Nash equilibrium. The maximal such path over all starting vertices is taken. In our case, in each communication game, L_G should simply be set to $\max_{\mathbf{a} \in \mathcal{E} \times \mathcal{E}} \min_{k=1 \ldots |\mathcal{E}|} |\mathcal{D}(\mathbf{a}, k)|$.

Corollary 1. *If, during the planning process, all agents use AP in communication games (i.e., the vector $\mathbf{P}$ is such that its components $P^j = AP,\ \forall j$) the algorithm* FINITEVICOM *will converge to a Nash equilibrium in any Com-SG.*

The above corollary is a direct implication of Theorems 3, 5 and 6. It shows that there exists at least one vector $\mathbf{P}$ for which the algorithm FINITEVICOM converges to a Nash equilibrium in any Com-SG. This vector is the one in which all players use AP.

It is interesting to note here that even if AP can only converge to a pure equilibrium in any communication game the resulting equilibrium of the original, stochastic game can still be mixed! Indeed, this fact only depends on the SG itself: AP is only a way to "choose" between the equilibria computed by the agents.

In the next section, we present the results of experiments justifying the theoretical results stated above.

Experimental results

We tested our FINITEVICOM algorithm on a sort of a multi-robot grid-world problem created by Hu & Wellman (2003) to test their Nash-Q algorithm. Briefly, the problem consists of two robots on a square grid. The initial positions of robots are respectively bottom-left and bottom-right corners of the grid. The robots have their respective goal cells in the opposite corners of the grid. The actions they have in their disposal are *L* (go left), *R* (go right), *U* (go up) and *NoOp* (do nothing). Both robots have the following reward function. The reward of 100 is obtained if a robot makes an action in its goal cell; the reward of -1 is obtained if there was a collision (both robots tried to simultaneously transit into the same cell) and the reward of 0 is obtained in all other cases. This sort of grid-world game possesses all the key elements of SGs: location- or state-specific actions, inter-state transitions, and immediate and long-term rewards.

It is easy to see that when the transition function is deterministic this game has ten equilibria (joint-trajectories) in the 3×3 grid for the horizon $H = 4$, as shown in Figure 1.

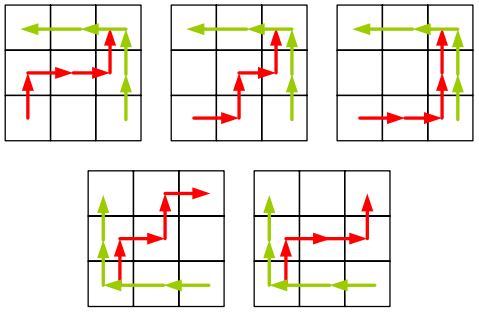

Figure 1: Equilibria of deterministic 3×3 grid-world game. Other five are obtained by symmetry.

When the planning horizon is $H \geq 5 = 4 + k$, $k = 1, 2, 3, \ldots$, the above trajectories do not change. However, the equilibria in these cases will contain k additional actions *NoOp* to execute in the goal cell. On the other hand, when the horizon is $H \leq 3$, multiple optimal solutions of this game exist, all bringing the utility of 0 to both agents regardless of the actions of the opponent player. One of them could be to always execute the *NoOp* action in the start cell.

$t=4$	**U , L** 16.62%	**R , U** 16.73%	**U , L** 11.08%	**U , L** 11.12%	**U , L** 5.69%	**U , U** 5.44%	**R , U** 11.1%	**R , U** 11.05%	**R , U** 5.55%	**U , U** 5.63%
$t=3$	**R , L** 16.62%	**R , L** 16.73%	**R , L** 11.08%	**U , L** 11.12%	**U , U** 5.69%	**U , L** 5.44%	**R , L** 11.1%	**R , U** 11.05%	**U , U** 5.55%	**R , U** 5.63%
$t=2$	**R , U** 16.62%	**U , L** 16.73%	**U , U** 11.08%	**R , U** 11.12%	**R , L** 11.12%		**U , U** 11.1%	**U , L** 11.05%	**R , L** 11.18%	
$t=1$	**U , U** 33.35%		**R , U** 33.32%				**U , L** 33.33%			

Table 1: Observed distribution over equilibrium actions in the 3×3 grid with horizon 4 for each time step.

The tests have been done for the case when both agents use AP as a game playing algorithm. Like Hu & Wellman (2003), we used the Lemke-Howson algorithm (McKelvey & McLennan 1996) to compute equilibria. Since the latter algorithm is deterministic, the sets of equilibria calculated by each agent were always the same. To simulate the case when these sets are different we stochastically withdrew some equilibria from the sets of both agents.

We observed the convergence to a Nash equilibrium in all tests in both cases, i.e., when all equilibria were available and when some of them were stochastically withdrawn. The distribution over the found equilibria of this SG is presented in Table 1. For example, at the time step 1 there are three different equilibrium joint-actions to which FINITEVICOM might converge: (*U*,*U*), (*R*,*U*) and (*U*,*L*). (Here, $(\cdot, \cdot)$ stands for (a^1, a^2) where $a^1 \in \mathcal{A}^1$ and $a^2 \in \mathcal{A}^2$.) One can note that the distribution over these joint-actions is close to being uniform. This is not surprising since AP guarantees a convergence to an equilibrium given that all actions are selected from the set of best responses with a non-zero probability. In our tests, we used a uniform distribution to choose between best response actions, that is why the distribution over equilibria themselves is also uniform.

Conclusion and future work

In this paper, we presented a novel approach to distributed equilibrium computation and selection in finite horizon planning problems in stochastic games. We proposed an algorithm using this approach and showed its validity, both theoretically and via experimentations.

In our work, we did not consider such important features which a particular communication game can have as communication cost and reliability. Indeed, we assumed that the communication is *always available*, *reliable* and *free*. In reality, however, this is often not the case. In our future work, we plan to explore in detail these questions.

One more important question is the scalability of the proposed approach. The influence of such parameters as the number of stochastic game states, the number of equilibria in each state (and also the fraction of this number computed by the agents during the equilibrium computation phase) on the algorithm's running time need to be explored and detail.

Another interesting research direction is an analysis of the convergence properties of different game playing algorithms in combined play, first experimentally and then theoretically. Indeed, in the context of the analysis of applicability of FINITEVICOM in either situation, one of the principal question is to know what the vectors $\mathbf{P}$ are in which all algorithms P^j, $\forall j = 1 \ldots n$, converge against each other to a pure Nash equilibrium in communication games.

References

Banerjee, B., and Peng, J. 2004. Performance bounded reinforcement learning in strategic interactions. *Proceedings of AAAI-04*.

Bowling, M., and Veloso, M. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136(2):215–250.

Claus, C., and Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of AAAI'98*.

Conitzer, V., and Sandholm, T. 2007. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning* 67(1):23–43.

Fudenberg, D., and Levine, D., eds. 1999. *The Theory of Learning in Games*. MIT Press, Massachusetts.

Hu, J., and Wellman, M. 2003. Nash Q-learning for general-sum stochastic games. *Journal of ML Research* 4:1039–1069.

Kearns, M.; Mansour, Y.; and Singh, S. 2000. Fast planning in stochastic games. In *Proceedings of UAI'2000*.

Littman, M. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the ICML'94*.

McKelvey, R., and McLennan, A. 1996. Computation of equilibria in finite games. *Handbook of Computational Economics* 1:87–142.

Myerson, R. 1991. Game Theory: Analysis of Conflict. *Boston, USA*.

Pavlidis, N.; Parsopoulos, K.; and Vrahatis, M. 2005. Computing Nash equilibria through computational intelligence methods. *J. of Computational and Applied Mathematics* 175(1):113–136.

Shapley, L. 1953. Stochastic games. *Proceedings of the National Academy of Sciences* 39:1095–1100.

Singh, S.; Kearns, M.; and Mansour, Y. 1994. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of UAI'94*.

Vrieze, O. 1987. *Stochastic games with finite state and action spaces*. Centrum voor wiskunde en informatica.

Wang, X., and Sandholm, T. 2002. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. *Proceedings of NIPS'02*.

Young, H. 1993. The evolution of conventions. *Econometrica* 61(1):57–84.

Zinkevich, M.; Greenwald, A.; and Littman, M. 2005. Cyclic Equilibria in Markov Games. *Proceedings of NIPS'2005*.

Zinkevich, M. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. *School of CS, CMU*.