Auton Agent Multi-Agent Syst (2010) 21:321–367
DOI 10.1007/s10458-009-9104-y

Coordinated learning in multiagent MDPs with infinite state-space

Francisco S. Melo · M. Isabel Ribeiro

Published online: 20 August 2009

Abstract In this paper we address the problem of simultaneous learning and coordination in multiagent Markov decision problems (MMDPs) with infinite state-spaces. We separate this problem in two distinct subproblems: learning and coordination. To tackle the problem of learning, we survey Q-learning with soft-state aggregation (Q-SSA), a well-known method from the reinforcement learning literature (Singh et al. in Advances in neural information processing systems. MIT Press, Cambridge, vol 7, pp 361–368, 1994). Q-SSA allows the agents in the game to approximate the optimal Q-function, from which the optimal policies can be computed. We establish the convergence of Q-SSA and introduce a new result describing the rate of convergence of this method. In tackling the problem of coordination, we start by pointing out that the knowledge of the optimal Q-function is not enough to ensure that all agents adopt a jointly optimal policy. We propose a novel coordination mechanism that, given the knowledge of the optimal Q-function for an MMDP, ensures that all agents converge to a jointly optimal policy in every relevant state of the game. This coordination mechanism, approximate biased adaptive play (ABAP), extends biased adaptive play (Wang and Sandholm in Advances in neural information processing systems. MIT Press, Cambridge, vol 15, pp 1571–1578, 2003) to MMDPs with infinite state-spaces. Finally, we combine Q-SSA with ABAP, this leading to a novel algorithm in which learning of the game and coordination take place simultaneously. We discuss several important properties of this new algorithm and establish its convergence with probability 1. We also provide simple illustrative examples of application.

Keywords Multiagent MDPs · Infinite state-spaces · Simultaneous learning and coordination · Q-learning with soft-state aggregation · Approximate biased adaptive play

F. S. Melo (✉)
School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave,
Pittsburgh, PA 15213, USA
e-mail: fmelo@cs.cmu.edu; fmelo@inesc-id.pt

M. I. Ribeiro
Institute for Systems and Robotics, Instituto Superior Técnico,
Av. Rovisco Pais, 1, 1049-001 Lisbon, Portugal
e-mail: mir@isr.ist.utl.pt

1 Introduction

Reinforcement learning (RL) addresses the problem of a decision-maker faced with a sequential decision problem and using evaluative feedback as a performance measure [42]. The general purpose of RL is to find a "good" mapping that assigns "perceptions" to "actions" and classically addresses situations in which a *single* decision-maker interacts with a *stationary* environment.

The powerful methods and impressive results of RL [12,46,48] have rendered this framework quite popular among the computer science and robotic communities, and recent years have witnessed increasing interest in extending RL methods to multiagent problems. Markov games (also known as stochastic games) and several variations or specializations thereof have been used to model multiagent RL problems [23]. Several researchers have applied single-agent RL methods (with adequate adaptations) to this multiagent framework.

Among the first, Littman [23] introduced the Minimax-Q algorithm as an extension of Q-learning to zero-sum Markov games. Hu and Wellman [19] later proposed Nash-Q as an elaboration of Minimax-Q that can be applied to general-sum Markov games. The authors established convergence of Nash-Q requiring, however, somewhat stringent conditions, as argued in [5,24]. This lead to the development of Friend-or-Foe Q-learning (FF-Q) [25]. FF-Q simplifies some of the computational burden involved in Nash-Q iterations, while retaining the convergence properties of the latter in most classes of games. In a somewhat related line of work, joint-action learners combine Q-learning with fictitious play in fully cooperative multiagent MDPs [11]. Fictitious play was also combined with prioritized sweeping to address planning in adversarial scenarios [52].

Gradient-based learning policies are analyzed in detail in [7,41]. In another work, Bowling and Veloso [9] propose a policy-based learning method that applies policy hill-climbing with a varying learning step, using the principle of "win or learn fast" (WoLF-PHC). Many other works on multiagent learning systems can be found in the literature—see, for example, the surveys works of [6,39].

However, most multiagent RL research focuses on problems in which the state-space is typically finite and not too large,[1] and only a few works on multiagent learning address problems with very large/infinite state-spaces. Some examples include the work of [7], where the WoLF (win-or-learn-fast) principle is combined with function approximation in Markov games with large or infinite state-spaces. The authors experimentally validate this combined method by applying it to the Goofspiel game, which has about 10^{11} states when using an ordinary card deck. Guestrin et al. [18] use coordination graphs to achieve coordination in multiagent problems with infinite state-spaces. This coordination mechanism uses structured communication and a variable elimination procedure to achieve coordination. In a related work [21], coordination graphs are also used to achieve coordination in continuous domains, this time with no communication assumed. Finally, Singh et al [41] also refer the interest of applying gradient ascent techniques to games with infinite state-spaces.

In this paper, we contribute another step in the research of solution methods for multiagent problems with infinite state-spaces. Concretely, we focus on fully cooperative multiagent decision problems, in which the state-space is a subset of $\mathbb{R}^p$, for some finite

[1] A curious fact that is worth mentioning is that Samuel's pioneer works back in the 1950–1960s, already address a gaming application with a huge state-space [37,38]. Also, the impressive results in generalization obtained by Tesauro's backgammon player also feature learning in a game with a huge state-space [46,47]. Both authors addressed learning in decision problems with large state-spaces and resort to approximation mechanisms to attain some level of generalization. However, both authors consider learning from a single-agent perspective.

p. In addressing this class of problems, we assume that the decision-makers have little prior knowledge on the scenario at hand. Therefore, they must *learn* a representation of the task for decision-making and then commit to a *common joint behavior* that is optimal in some sense. We assume that no explicit communication takes place, i.e., consensus must *emerge* from the mutual interaction among the different agents and with the environment.[2] Therefore, in this paper we feature cooperation as *coordination*: the multiple decision-makers must *coordinate* their individual decisions to yield an optimal joint behavior. It is worth noting that even in the simplified (fully cooperative) setting considered here, infinite state-spaces already pose several important challenges, not only in terms of learning but especially in terms of coordination.

In our approach we start by considering separately two distinct "subproblems", which can roughly be defined as:

- *Learning* the structure of the decision problem, where each decision-maker must learn a compact representation of the MMDP from which an optimal policy can be determined;
- *Coordination* where, in the presence of multiple optimal policies, all decision-makers agree upon one such policy *without any explicit communication*;

In the remainder of the paper, we generally refer to the first subproblem as the problem of learning and to the second as the problem of coordination. To address the first of the two, we propose the use of Q-learning with soft-state aggregation (Q-SSA). The combination of Q-learning and soft-state aggregation has been studied in the RL literature using different approaches [17,40,50]. In this paper, we describe the application of Q-SSA to multiagent MDPs with infinite state-spaces and introduce a modified version of the result in [40] on the convergence of Q-SSA. In particular, we identify the conditions required for Q-SSA to converge with probability 1 (w.p.1) in MMDPs with infinite state-spaces. We also establish a new result describing the rate of convergence of Q-SSA. This analysis is conducted in Sect. 3.

In addressing the problem of coordination, we introduce one of the main contributions of the paper, a novel coordination mechanism that we refer as *approximate biased adaptive play* (ABAP). This method is an extension of biased adaptive play (BAP) to infinite settings. BAP was first introduced in [53] and builds on a variation of fictitious play known as *adaptive play*.[3] We establish convergence of ABAP to an optimal policy w.p.1 when the game structure is known. The description of ABAP and corresponding analysis can be found in Sect. 4.

The ABAP method proposed in this paper differs from other coordination methods in several aspects. First of all, ABAP assumes that no communication takes place. On the other hand, ABAP is *rational* and *convergent in self-play*, in the sense of [8]. This is an important advantage of ABAP: in the presence of a heterogeneous group of decision-makers, ABAP is still able to converge to the best decision-rule possible if, for some reason, the other decision-makers are constrained to follow some particular sub-optimal individual policy.

In Sect. 5 we get to the final contribution of the paper, and combine ABAP with Q-SSA, this leading to the *coordinated approximate Q-learning* (CAQL) algorithm.[4] As will soon become apparent, some of the valuable aspects of this new method are its sound convergence

[2] The consideration of no explicit communication can be supported by several arguments (bandwidth constraints, cost of communication, possible added complexity to the problem, etc.). We do not pursue such arguments here and refer to several works that discuss these issues in greater detail [14,49].

[3] Fictitious play was first introduce in [10] and its convergence properties analyzed in several posterior works [22,36]. Adaptive play was introduced in [55].

[4] We must emphasize that our method, although baring a somewhat similar designation, has no relation whatsoever with the coordinated reinforcement learning algorithms proposed in [18].

properties and broad applicability. We analyze how ABAP can be interleaved with Q-SSA without affecting the convergence of either method, thus establishing convergence of CAQL. In Sect. 6 we also illustrate the performance of CAQL in simple illustrative scenarios and discuss several properties of the method. Finally, we conclude in Sect. 7 with some final remarks.

To minimize the disruption of the presentation, we collected some background material on Markov chains in Appendix A and all the proofs of all relevant results in Appendix B.

2 Background

Aiming at making the paper as self-contained as possible, we start by reviewing several fundamental concepts from game theory and Markov processes that constitute the background for all the material in the paper. We also introduce a couple of well-known results that are relevant in establishing the contributions in the paper.

2.1 Markov decision problems

Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^p$, for some finite p, and let $\{X(t)\}$ a $\mathcal{X}$-valued controlled Markov chain. In other words, $\{X(t)\}$ is a stochastic process in which the distribution of the random variable (r.v.) $X(t+1)$ depends on the values of the r.v.s $X(t)$ (the *state* at time t) and $A(t)$ (the *action* at time t) and is given by

$$\mathsf{P}_a(x, U) \triangleq \mathbb{P}\left[X(t+1) \in U \mid X(t) = x, A(t) = a\right],$$

for any measurable set $U \subset \mathcal{X}$. The $\mathcal{A}$-valued process $\{A(t)\}$ represents the control process: $A(t)$ is the control action at time instant t and $\mathcal{A}$ is the finite set of possible actions. A decision-maker must determine the control process $\{A(t)\}$ so as to maximize

$$V(x, \{A(t)\}) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X(t), A(t)) \mid X(0) = x\right], \tag{1}$$

where $0 \leq \gamma < 1$ is a discount-factor and $R(x, a)$ represents a random "reward" received for taking action $a \in \mathcal{A}$ in state $x \in \mathcal{X}$. We assume that the control process $\{A(t)\}$ is adapted to the σ-algebra generated by $\{X(t)\}$ and that there is a bounded real-valued function $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$, assigning a reward $r(x, a, y)$ every time a transition from x to y occurs after taking action a, such that

$$\mathbb{E}\left[R(x, a)\right] = \int_{\mathcal{X}} r(x, a, y)\mathsf{P}_a(x, \mathrm{d}y).$$

This simplifies the notation without introducing a great loss in generality. We refer to the 5-tuple $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ as a *Markov decision problem* (MDP).

Given an MDP $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$, the *optimal value function* V^* is defined for each state $x \in \mathcal{X}$ as

$$V^*(x) \triangleq \max_{\{A(t)\}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X(t), A(t)) \mid X(0) = x\right]$$

and verifies the recursive relation

$$V^*(x) = \max_{a\in\mathcal{A}} \int_{\mathcal{X}} \left[r(x,a,y) + \gamma V^*(y)\right] \mathsf{P}_a(x, \mathrm{d}y), \tag{2}$$

which is a form of the Bellman optimality equation. The optimal Q-values are defined for each state-action pair (x, a) as

$$Q^*(x,a) = \int_{\mathcal{X}} \left[r(x,a,y) + \gamma V^*(y)\right] \mathsf{P}_a(x, \mathrm{d}y). \tag{3}$$

For future reference, we define the Bellman operator $\mathbf{H}$ as

$$(\mathbf{H}Q)(x,a) = \int_{\mathcal{X}} \left[r(x,a,y) + \gamma \max_{b\in\mathcal{A}} Q(y,b)\right] \mathsf{P}_a(x, \mathrm{d}y), \tag{4}$$

where Q is a general function defined over $\mathcal{X}\times\mathcal{A}$ and taking values in $\mathbb{R}$. Notice that Q^* is the *fixed-point* of $\mathbf{H}$. From Q^* we can define the mapping π^* as

$$\pi^*(x) = \arg\max_{a\in\mathcal{A}} Q^*(x,a), \qquad \forall x \in \mathcal{X}.^{5}$$

The control process defined for all t by $A(t) = \pi^*(X(t))$ is optimal in the sense that $V(x, \{A(t)\}) = V^*(x)$, for all $x \in \mathcal{X}$. The mapping π^* is an *optimal policy* for the MDP $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$. It then follows that $V^*(x)$ accounts for the expected total discounted reward associated with the optimal policy when the initial state is x, while $Q^*(x,a)$ measures the expected total discounted reward associated with the optimal policy when the initial state is x and the first action is fixed to be a.

More generally, we define a (stationary) *policy* as any mapping π over $\mathcal{X}\times\mathcal{A}$ that generates a control process verifying, for all t,

$$\mathbb{P}\left[A(t) = a \mid X(t) = x\right] = \pi(x,a).$$

Since $\pi(x,\cdot)$ is a probability distribution over $\mathcal{A}$, it must satisfy

$$\sum_{a\in\mathcal{A}} \pi(x,a) = 1.$$

for all $x \in \mathcal{X}$. We write $V^\pi(x)$ instead of $V(x, \{A(t)\})$ if the control process $\{A(t)\}$ is generated by π. A *deterministic policy* is a policy assigning probability 1 to a single action in each state. We denote such policy as a mapping $\pi : \mathcal{X} \to \mathcal{A}$ that generates a control process $\{A(t)\}$ verifying $A(t) = \pi(X(t))$ for all t.

We emphasize that the optimal control process can be obtained from the optimal (deterministic) policy π^*, which can in turn be obtained from Q^*. Therefore, the Markov decision problem $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ is *solved* once the function Q^* is known for all pairs (x, a).

[5] On a side note, we should point out that, in general, $\arg\max_{a\in\mathcal{A}} Q^*(x,a)$ is a *set* and as such we should write $\pi^*(x) \in \arg\max_{a\in\mathcal{A}} Q^*(x,a)$. However, and to simplify the presentation, we adhere to the slight abuse of notation as in the expression above.

2.1.1 The Q-learning algorithm

The "classical" approach to reinforcement learning considers MDPs with a finite state-space $\mathcal{X}$. Under this finiteness assumption, Q^* (and hence any corresponding estimates) can be represented as an $|\mathcal{X}| \times |\mathcal{A}|$ matrix. In the remainder of this section, we focus on MDPs with finite state-space $\mathcal{X}$, postponing to the next section the treatment of problems where $\mathcal{X}$ is no longer finite.

In order to compute the Q-function for a given MDP without any previous knowledge of the transition kernel P and the reward function r, Watkins proposed in 1989 the *Q-learning algorithm* [54]. Q-learning is implemented as follows: given an MDP $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ and an infinite sample trajectory $\{x(t)\}$ of the underlying Markov chain obtained with some sampling policy π, let $\{a(t)\}$ and $\{r(t)\}$ denote the corresponding sample sequence of actions and rewards. Q-learning successively updates the estimate Q for Q^* using the rule

$$Q(x,a) \leftarrow (1-\alpha_t(x,a))Q(x,a) + \alpha_t(x,a)\left[r(t) + \gamma \max_{b\in\mathcal{A}} Q(x(t+1),b)\right], \qquad (5)$$

where $\{\alpha_t\}$ is a (x,a)-dependent step-size sequence. The estimate Q converges to Q^* w.p.1 as long as $\sum_t \alpha_t(x,a) = \infty$ and $\sum_t \alpha_t^2(x,a) < \infty$, where $\alpha(x,a) = 0$ if $(x,a) \neq (x(t), a(t))$. This condition is usually stated by saying that *Q-learning converges as long as every state-action pair is visited infinitely often*.

Q-learning uses a policy π (not necessarily optimal) to generate an infinite trajectory and convergence to Q^* is attained as t goes to infinity. The optimal control law, represented as the optimal policy π^*, can then be computed from Q^*. Q-learning converges to Q^* independently of the policy π chosen, as long as the requirement of infinite visits to all state-action pairs is met. For concreteness, we henceforth refer to the policy used during learning, i.e., the policy used to generate the sample trajectories $\{x(t)\}$, $\{a(t)\}$ and $\{r(t)\}$, as the *learning policy*.

2.1.2 Multiagent MDPs

A multiagent MDP (MMDP) is a tuple $(N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$, where N is the number of agents, $\mathcal{X}$ is the state-space, $\mathcal{A} = \times_{k=1}^N \mathcal{A}_k$ is the set of joint actions, P is the controlled transition kernel and r is the joint reward function. The differences between an MMDP and an MDP lie essentially on the fact that the action-space $\mathcal{A}$ of the former is the Cartesian product of the N individual action-spaces, $\mathcal{A}_k$, and thus corresponds to the *joint action-space*. This means that the transition probabilities in an MMDP depend on the actions of *all* agents. Therefore, the $\mathcal{A}$-valued process $\{A(t)\}$ represents the *joint control process*: at each time instant t each agent k independently chooses an action $A_k(t)$ from $\mathcal{A}_k$. The joint action $A(t)$ is obtained by combining all individual actions $A_1(t), \ldots, A_N(t)$, and is represented as a tuple $A(t) = (A_1(t), \ldots, A_N(t))$. We denote by $A_{-k}(t)$ a *reduced action*, obtained by removing the individual action $A_k(t)$ from $A(t)$.

As the chain moves from state $X(t)$ to state $X(t+1)$, *all* agents receive a reward $r(X(t), A(t), X(t+1))$. The purpose of *each* decision-maker is to determine the *individual* control process $\{A_k(t)\}$ so as to maximize the functional

$$V(x, \{A(t)\}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X(t), A(t)) \mid X(0) = x\right], \qquad (6)$$

where $0 \leq \gamma < 1$ is once again a discount-factor and $R(x,a)$ represents the random reward received by all agents when they take action $a \in \mathcal{A}$ in state $x \in \mathcal{X}$.

It is worth noting that an MMDP models a *fully cooperative* decision problem. In fact, since the reward function is common to all agents, whatever is "good" for one is good for all. In other words, all agents are trying to optimize the same objective function. On the other hand, since the state-evolution process and the reward received by each agent both depend on the actions of *all* agents, the agents must *coordinate* their action choice in order to ensure that their joint behavior is optimal. We postpone further discussions on the issue of coordination to Sect. 2.2.

An *individual policy* for agent k is denoted as π_k and defines the probability of agent k playing each action $a_k \in \mathcal{A}_k$ when the process is in state $x \in \mathcal{X}$. In other words, it is a mapping defined on $\mathcal{X} \times \mathcal{A}_k$ that generates a process $\{A_k(t)\}$ verifying, for all t,

$$\mathbb{P}\left[A_k(t) = a_k \mid X(t) = x\right] = \pi_k(x, a_k),$$

with $a_k \in \mathcal{A}_k$. A *joint policy* is a tuple $\pi = (\pi_1, \ldots, \pi_N)$ of individual policies, where $\pi(x, a)$ represents the probability of the joint action a being played in state x when all agents follow the policy π. We refer to π_{-k} as a *reduced policy*, obtained from π by removing the individual policy of agent k.

As mentioned above, in an MMDP all agents share the same joint goal (maximizing the total expected reward over all admissible joint control sequences). Therefore, apart from the way by which actions are chosen, there is no other significant difference between an MDP and its multiagent counterpart. This means that most properties from MDPs discussed in Sect. 2.1 carry without change to MMDPs. In particular, we write $V^{\pi}(x)$ instead of $V(x, \{A(t)\})$ if the joint control process $\{A(t)\}$ is generated by a joint policy π. Like MDPs, MMDPs always have at least one optimal joint policy and we can define the optimal value function and the optimal Q-function as in MDPs. The optimal value function also verifies (2). However, the fact remains that the decision process in MMDPs is distributed and requires coordination to be addressed explicitly [3].

2.2 Game theory

We now review several basic concepts on game theory used throughout the paper. Specifically, we discuss the fundamental class of matrix games and the biased adaptive play (BAP) coordination mechanism.

2.2.1 Matrix games

An *N-agent matrix game* is a tuple $(N, (\mathcal{A}_k), (r_k))$, where N is the number of agents in the game, $\mathcal{A} = \times_{k=1}^{N} \mathcal{A}_k$ is the finite set of all *joint actions* and r_k is a function assigning a payoff $r_k(a)$ to agent k, when the joint action $a \in \mathcal{A}$ is played.[6] A *joint action* $a \in \mathcal{A}$ is a tuple $a = (a_1, \ldots, a_N)$. We denote by a_{-k} a *reduced action*, obtained by removing the individual action a_k from a.

An individual policy for agent k is denoted as π_k and defines the probability of agent k playing each individual action $a_k \in \mathcal{A}_k$ in the game. A deterministic policy is known in the game-theoretic literature as a *pure policy*, and a *mixed policy* otherwise. As in MMDPs, a *joint policy* is a vector $\pi = (\pi_1, \ldots, \pi_N)$ of individual policies, where $\pi(a)$ represents the probability of the joint action a being played when all agents follow the policy π. We refer to π_{-k} as a *reduced policy*, obtained from π by removing the individual policy of agent k.

[6] For the sake of uniformity, we adopt the general designation of "agent" instead of the more common game-theoretic designation of "player".

The individual policy π_k^* of agent k is a *best response* to a reduced policy π_{-k} if agent k cannot improve its expected reward by using any other individual policy, i.e.,

$$\mathbb{E}_{(\pi_{-k},\pi_k^*)}\left[r_k(a)\right] \geq \mathbb{E}_{(\pi_{-k},\pi_k)}\left[r_k(a)\right]. \tag{7}$$

A *Nash equilibrium* is a joint policy $\pi^* = (\pi_1^*, \ldots, \pi_N^*)$ in which each individual policy π_k^* is a best response to the reduced policy π_{-k}^*. Every finite matrix game has at least one Nash equilibrium [32]. A Nash equilibrium π^* is *strict* if the inequality in (7) is strict for every individual policy π_k^*.

A game in which $r_1(a) = \ldots = r_N(a)$ for all $a \in \mathcal{A}$ is *fully cooperative*. In this class of games there is always (at least) one pure Nash equilibrium that yields maximum payoff for all agents (a *Pareto optimal* Nash equilibrium). In the remainder of this paper, we consider only fully cooperative games and thus refer to such an equilibrium as being an *optimal joint policy* for the game.

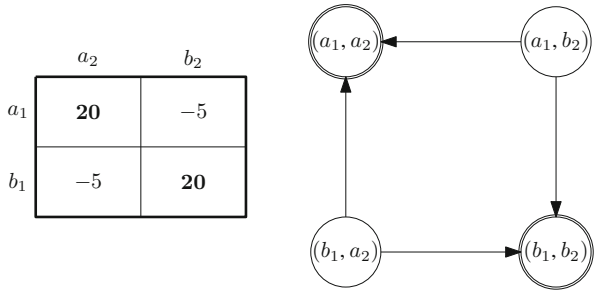
It is worth noting that that, in an MMDP, the function Q^* defines at each state $x \in \mathcal{X}$ a *fully cooperative matrix game* $\Gamma_x = \left(N, (\mathcal{A}^k), r^*\right)$ with $r^*(a) = Q^*(x, a)$. We refer to each such game as a *stage-game*. As shown in [4], if all agents coordinate in an optimal policy in each stage-game Γ_x, they will coordinate in an optimal policy for the MMDP. We use this result extensively in Sect. 5.

2.2.2 Biased adaptive play

The existence of at least one optimal policy does not imply its uniqueness. In fact, many games possess multiple optimal policies and this can lead to the so-called *coordination problem* [3], also known as the *equilibrium selection problem* in the game-theoretic literature. To better understand this problem consider the matrix game in Fig. 1a. It is a two-agent game in which each agent has two actions available. Agent 1 can choose any of the actions a_1 or b_1. Similarly, agent 2 can choose any of the actions a_2 or b_2.

In this game there are two optimal policies and there is no particular reason why one agent should prefer any of the two optimal policies to the other. Therefore, it may happen that agent 1 adheres to the policy (a_1, a_2), thus playing a_1, while agent 2 adheres to the policy (b_1, b_2), thus playing b_2. Alas, the joint action obtained, (a_1, b_2), is far from optimal.

This very simple example illustrates the coordination problem that often occurs in fully cooperative matrix games and, as will soon become apparent, in MMDPs. Even if all agents know the game, it is necessary to explicitly address the problem of coordination by means of a specific coordination mechanism. The purpose of this mechanism is to ensure that, in the presence of multiple optimal policies, all agents commit to the same one.



(a) Payoff matrix. **(b)** Best-response graph.

Fig. 1 Simple two-agent, two-action team matrix game. The optimal policies (i.e., the Pareto optimal equilibria) are marked in *bold* in the payoff matrix and with a *double line* in the best-response graph

It is worth mentioning that equilibrium selection problems also occur in more general games, in which different agents can have different rewards. In its most general form, it is a difficult problem and a topic of intense research in game theoretic literature.

In this paper, we are interested in an adaptive mechanism that ensures coordination to *emerge* as the agents *repeatedly* play the game.[7] One such mechanism was introduced in [53] and is known as *biased adaptive play* (BAP). BAP is a variant of fictitious play [10,36] and is designed to address coordination problems in fully cooperative repeated games. Both fictitious play and BAP rely on the fundamental observation that the past action choices of other agents can be used to estimate their policy. The distinctive features of BAP that set it apart from fictitious play are (1) an incomplete sampling mechanism that ensures that certain pathological behaviors observed with fictitious play do not occur [55]; (2) the construction of an auxiliary virtual game that ensures that, in self-play, all agents converge to an optimal joint policy.

To formalize BAP, we need some preliminary concepts.

Definition 1 (*Best response graph*) Let $\Gamma = (N, (\mathcal{A}_k), r)$ be a fully-cooperative repeated game with finite action-space $\mathcal{A} = \times_{k=1}^{N} \mathcal{A}_k$. The *best response graph* for Γ is a directed graph $G = (V, E)$, where $V = \mathcal{A}$ and, given any two vertices $a, b \in V$, $(a, b) \in E$ if and only if $a \neq b$, there is exactly one agent k for which b_k is a best response to a_{-k}, and $a_{-k} = b_{-k}$.[8]

The vertices in the best response graph thus correspond to all joint actions in the game and an edge between two vertices a and b implies that there is exactly one agent k that can improve its performance by changing its action from a_k to b_k, as long as no other agent changes its action. Clearly, a *sink* in the best response graph corresponds to a Nash equilibrium, since no agent can individually improve its performance by changing only its action. The best-response graph for the simple 2-agent, 2-action example in Fig. 1a is depicted in Fig. 1b.

Let now $\Gamma = (N, (\mathcal{A}_k), r)$ be a repeated game and $D \subset \mathcal{A}$ a set containing some of the pure Nash equilibria in Γ (and no other joint actions).

Definition 2 (*Weakly acyclic game*) A repeated game $\Gamma = (N, (\mathcal{A}_k), r)$ is *weakly acyclic* if, given any vertex a in its best response graph, there is a directed path to a vertex a^* from which there is no exiting edge. It is *weakly acyclic with respect to (w.r.t.) the bias set D* if, given any vertex a in the best response graph of Γ, there is a directed path to either a Nash equilibrium in D or a pure strict Nash equilibrium.

We are now in position to introduce BAP. Given a repeated game $\Gamma = (N, (\mathcal{A}_k), r)$, we construct a virtual game $VG = \left(N, (\mathcal{A}_k), \hat{r}\right)$, where $\hat{r}(a) = 1$ if a is an optimal joint policy for Γ and $\hat{r}(a) = 0$ otherwise. Every Nash equilibrium in this new game corresponds to an optimal policy in the original game. Therefore, if all agents are able to coordinate in a Nash equilibrium in VG, they will have coordinated in an optimal policy in the original game, as desired. If we now set $D = \left\{a \in \mathcal{A} \mid \hat{r}(a) = 1\right\}$, then VG is always weakly acyclic w.r.t. the bias set D [53].

[7] A game in which the agents repeatedly engage in the same matrix game is known as a *repeated game*. Repeated games allow for the possibility of having an agent change its policy depending on past plays of the other agents. This is not possible in standard matrix games, which are *one-shot games*.

[8] Note that since G is a direct graph, (a, b) represents a distinct edge from (b, a). In particular, the direction of the edges in G indicates the "best-response direction". Therefore, if there is an edge from a to b in G and the two joint actions differ on their kth component, then b_k is the best response to a_{-k} (or, equivalently, b_{-k}).

Let H_t denote the m most recent plays in Γ at time t, i.e.,

$$H_t = \{a(t-m+1), \ldots, a(t-1), a(t)\}.$$

For any $0 < K \leq m$, a *K-sample from H_t* is a set of K plays randomly drawn without replacement from H_t. We denote a K-sample from H_t as $\mathsf{K}(H_t)$.

Given the virtual game VG, at each time instant $t \geq m$ and *independently from all other agents*, each agent k draws a K-sample $\mathsf{K}_k(H_t)$ from the history of the m most recent plays and checks if:

1. There is a joint action $a^* \in D$ such that, for all played actions $a \in \mathsf{K}_k(H_t)$, $a_{-k} = a^*_{-k}$;
2. There is at least one played action $a \in \mathsf{K}_k(H_t)$ such that $a \in D$.

If these two conditions are verified, agent k concludes that all remaining agents have coordinated in an action in D. Therefore, agent k chooses its best response a^*_k so that

$$a^* = (a^*_{-k}, a^*_k) = \arg\max_{a(\tau) \in H_t} \{\tau \mid a(\tau) \in \mathsf{K}_k(H_t) \quad \text{and} \quad a(\tau) \in D\}.$$

If any of the above conditions is not verified, agent k uses the K-sample to estimate the expected payoff of each of its individual actions as

$$EP_t(a_k) = \sum_{a_{-k} \in \mathcal{A}_{-k}} \hat{r}(a_{-k}, a_k) \frac{n_K(a_{-k})}{K},$$

where $n_K(a_{-k})$ denotes the number of times that the reduced action a_{-k} appears in the K-sample $\mathsf{K}_k(H_t)$. Notice that $EP(a_k)$ estimates the value of the individual action a_k given the average policy of the other agents according to the K-sample. Agent k then chooses its action randomly from the best response set

$$BR_t = \left\{ a^*_k \mid EP_t(a^*_k) = \max_{a_k \in \mathcal{A}_k} EP_t(a_k) \right\}. \tag{8}$$

It is now worth detailing the fundamental intuitive principles behind BAP. The fundamental idea behind BAP is to track the actions chosen by the other agents, using these as an indication of their policy (in this sense, BAP is similar to fictitious play). However, unlike fictitious play, it does not use the complete history, but only the most recent m actions. In a sense, this "short-term memory" allows BAP to cope with other agents that are also adjusting their policy.

Another fundamental idea in BAP is to further sub-sample this short-term memory, by randomly choosing only k of the m actions. Sub-sampling ensures (probabilistically) that BAP does not enter undesirable cycles of suboptimal action choice. To illustrate this phenomenon, consider again the game in Fig. 1a, and suppose that each of the two agents keeps track of the past plays of the other agent to choose its own actions. However, in similar circumstances, agent 1 is biased toward choosing action a while agent 2 is biased toward action b. At the first time-step, therefore, agent 1 will choose a_1 and agent 2 b_2, leading to a suboptimal action. In the following time-step, agent 1 notices that agent 2 chose action b_2 and acts accordingly, choosing action b_1. The converse happens with agent 2, whom will choose action a_2, again leading to a suboptimal action. In the following time-step, each agent has played each action exactly once, so again agent 1 chooses action a_1 and agent 2 chooses b_2. It is easy to see that leads to a cycle of suboptimal actions in which the two agents have the worst possible performance. This adverse effect can sometimes be observed for example in fictitious play, but is avoided in BAP due to the sub-sampling mechanism.

Finally, conditions 1 and 2 ensure that, probabilistically, the agents "follow" the best response graph associated with the game. The fact that this game is weakly acyclic w.r.t. some bias set that contains only optimal policies ensures that all agents will eventually converge to an optimal policy, as intended.

The following theorem can be found in [53].

Theorem 1 *Let $\Gamma = (N, (\mathcal{A}_k), r)$ be a fully cooperative repeated game that is weakly acyclic w.r.t. some bias set D. If*

$$K \leq \frac{m}{L(\Gamma) + 2},$$

then biased adaptive play converges w.p.1 to either a strict Nash equilibrium or a Nash equilibrium in D.

The constant $L(\Gamma)$ is defined as $L(\Gamma) = \max_{a \in \mathcal{A}} L(a)$, where $L(a)$ is the shortest path in the best response graph of Γ going from vertex a to either a strict Nash equilibrium or a Nash equilibrium in D.

It is worth noting that the particular "path" followed by BAP depends on the K-samples drawn by the different agents along the process. Therefore, BAP does not necessarily converge to the equilibrium "closest" to the starting vertex.

The above result formalizes the convergence properties of BAP, establishing BAP as a sound coordination mechanism that, as will soon become apparent, can also be effectively used in the class of problems considered in this paper.

3 The problem of learning

In this section, we address the problem of learning/approximating the optimal Q-function in MMDPs with infinite state-spaces. To this purpose, we start by introducing *Q-learning with soft-state aggregation* (Q-SSA), an algorithm to approximate the optimal Q-function for MDPs with infinite state-spaces [40]. We establish convergence of this algorithm w.p.1 and produce a new result that describes the rate of convergence of Q-SSA. We then extend Q-SSA to MMDPs with infinite state-spaces, since the latter class of problems can be seen as generalized MDPs in which the decision-making process is distributed.[9]

3.1 Q-learning with soft-state aggregation

In the original Q-learning algorithm, the Q-values are updated according to (5) that we repeat here for commodity

$$Q(x, a) \leftarrow (1 - \alpha_t(x, a))Q(x, a) + \alpha_t(x, a)\left[r(t) + \gamma \max_{b \in \mathcal{A}} Q(x(t+1), b)\right].$$

This algorithm implicitly requires the function Q to be explicitly represented, which is clearly impossible if $\mathcal{X}$ is not a finite set. In this paper we are interested in those situations in which $\mathcal{X}$ is a compact subset of $\mathbb{R}^p$, for some finite p.[10] This means that, in general, we cannot apply the Q-learning algorithm in its original form.

[9] A similar analysis was conducted in [44] in a more restricted class of games.

[10] In the remainder of the paper, whenever we refer to the state-space $\mathcal{X}$ as a compact subset of $\mathbb{R}^p$, we implicitly assume p to be finite.

To address this difficulty, we consider a representation of Q^* as the linear combination of a fixed set of *basis functions*. Consider then a set of linearly independent functions $\{\phi_i, i = 1, \ldots, M\}$ defined over $\mathcal{X}$ and taking values in $\mathbb{R}$. We admit the functions $\phi_i, i = 1, \ldots, M$, to verify

$$\phi_i(x) \geq 0 \quad \sum_{i=1}^{M} \phi_i(x) = 1.$$

This means that the functions $\phi_i, i = 1, \ldots, M$ provide a *soft-partition* of the state-space $\mathcal{X}$ into M sets $U_1, \ldots, U_M$, each defined as

$$U_i \triangleq \text{supp}(\phi_i).$$

The value of $\phi_i(x)$ can be interpreted as a "probability" of x belonging to U_i. We now seek a good representation for Q^* among the parameterized family of functions $\mathcal{Q} = \{Q_\theta\}$, where each Q_θ is defined over $\mathcal{X} \times \mathcal{A}$ and takes values in $\mathbb{R}$, and can be written as

$$Q_\theta(x, a) = \sum_{i=1}^{M} \phi_i(x)\theta(i, a) = \phi^\top(x)\theta(a).$$

In the above expression, θ is a $M \times |\mathcal{A}|$ parameter matrix and we write $\theta(a)$ to denote the ath column of θ. We also denote by $\phi(x)$ the column vector with ith component $\phi_i(x)$ and write $^\top$ for the transpose operator.

In general, $Q^* \notin \mathcal{Q}$, but the elements of $\mathcal{Q}$ can be compactly represented by means of the corresponding parameter θ. Therefore, we can replace the algorithm for finding Q^* by a suitable "equivalent" algorithm to find a parameter θ^* such that Q_{θ^*} is the best approximation of Q^* in $\mathcal{Q}$. We thus move from a search in an infinite-dimensional function space to a search in a finite dimensional space, $\mathbb{R}^{M\times|\mathcal{A}|}$. This, however, has an immediate implication: unless if $Q^* \in \mathcal{Q}$, we will not be able to determine Q^* exactly.

Let $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ be an MDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and let π be any given (stationary) stochastic policy. Suppose that $\{x(t)\}$, $\{a(t)\}$ and $\{r(t)\}$ are sampled trajectories of states, actions and rewards from $\mathcal{M}$ using the policy π. The modified update rule for Q-learning with soft-state aggregation is

$$\theta(a) \leftarrow (1 - \alpha_t(a))\theta(a) + \alpha_t(a)\phi(x(t)) \left[r(t) + \gamma \max_{b\in\mathcal{A}} Q_\theta(x(t+1), b) \right] \tag{9}$$

where $\alpha_t(a) = 0$ if $a \neq a(t)$.

The next result formally establishes the convergence of Q-SSA when a stationary learning policy π is used and is a restatement of Corollary 1 in [40]. Given an MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$, we denote by $(\mathcal{X}, \mathsf{P}_\pi)$ the Markov chain induced in $\mathcal{M}$ by π and assume this chain to be geometrically ergodic, with invariant measure μ_π. Roughly speaking, this condition ensures that behavior of the algorithm along the sample trajectories of $(\mathcal{X}, \mathsf{P}_\pi)$ can be analyzed as if the samples were drawn according to μ_π. We refer to Appendix A for a formal definition. We also assume that $\pi(x, a) > 0$ for all $a \in \mathcal{A}$ and μ_π-almost all $x \in \mathcal{X}$.[11]

Theorem 2 *Let $\mathcal{M}$, π and $\{\phi_i, i = 1, \ldots, M\}$ be as defined above. For any initial condition $\theta_0 \in \mathbb{R}^{M\times|A|}$, it holds that:*

[11] We write that a given condition $\rho(x)$ holds for μ_π-almost all $x \in \mathcal{X}$ if the set of states $x \in \mathcal{X}$ for which $\rho(x)$ does not hold has null μ_π measure.

1. *Q-SSA converges w.p.1 as long as the step-size sequence verifies*

$$\sum_t \alpha_t(a) = \infty \quad \sum_t \alpha_t^2(a) < \infty$$

for all $a \in \mathcal{A}$*;*
2. *The limit point* θ^* *of Q-SSA verifies*

$$\theta^*(a) = \mathbb{E}_{\mu_\pi}\left[\phi(x)(\mathbf{H}Q_{\theta^*})(x,a)\right]. \tag{10}$$

Proof See Appendix B.

We emphasize that the above result guarantees convergence of Q-SSA under suitable conditions but provides no guarantees on the quality of the obtained approximation. Although we do not pursue such development here, it is possible to show that the Q-function obtained with Q-SSA is the "optimal Q-function" for an associated decision problem obtained from $\mathcal{M}$. Nevertheless, the quality of the obtained approximation will greatly depend on the representational power of the set of basis functions chosen. We refer to Sect. 7 and [40] for further discussion on these issues.

For the developments in Sect. 5 we also need some bound on the rate of convergence of Q-SSA. Previous work has addressed the rate of convergence of classical Q-learning [15,20,43], but to the extent of our knowledge, the following is the first result on the convergence rate of Q-SSA:

Theorem 3 *Under the conditions of Theorem 2, the sequence* $\{\theta_t\}$ *generated by Q-SSA verifies the following bound w.p.1:*

$$\limsup \frac{\|\theta_t - \theta^*\|_\infty}{\sqrt{\max_a \alpha_t(a) \log\left(\sum_{\tau=0}^t \alpha_\tau(a)\right)}} \leq K$$

for some $K > 0$*, as long as*

$$\alpha_t(a) = \begin{cases} \frac{1}{n_t(a)}, & \text{if } a = a(t) \\ 0, & \text{otherwise} \end{cases}$$

with $n_t(a)$ *denoting the number of times that action* a *has been tried up to time instant* t*.*

Proof See Appendix B. □

The previous result states that the convergence of θ to θ^* verifies a bound similar to the law of iterated logarithm.[12] Defining

$$\mathcal{E}_t \triangleq \sqrt{\max_a \alpha_t(a) \log\left(\sum_{\tau=0}^t \alpha_\tau(a)\right)}.$$

and using the standard notation for asymptotic complexity, the statement in Theorem 3 becomes, simply,

$$\left\|\theta_t - \theta^*\right\|_\infty \in \mathrm{O}(\mathcal{E}_t).$$

[12] Note, for example, that if $|\mathcal{A}|$=1, the step-size sequence becomes $\alpha_t = 1/t$ and the bound in Theorem 3 reduces to the law of iterated logarithm.

3.2 Multiagent MDPs with infinite state-spaces

We now extend Theorems 2 and 3 to MMDPs with infinite state spaces.

Let $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ be an MDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$. Consider also a set of N independent decision-makers (agents), each agent k having a repertoire $\mathcal{A}_k$ of individual actions such that $\mathcal{A} = \times_{k=1}^{N} \mathcal{A}_k$. In other words, suppose that $\mathcal{A}$ in the MDP $\mathcal{M}$ is the Cartesian product of the individual action-spaces $\mathcal{A}_k$. The MMDP $\Gamma = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ thus defined is equivalent to the MDP $\mathcal{M}$ and, as pointed out before, the only difference between $\mathcal{M}$ and Γ lies on the fact that, in the former, the decision process is inherently *centralized*. Taking advantage of this close relation between Γ and $\mathcal{M}$ we can extend the results concerning Q-SSA to multiagent scenarios in quite a straightforward manner.

Let then $\Gamma = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ be an MMDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$. As remarked above, the only difference between applying Q-SSA to an MDP or to an MMDP lies on the fact that, in the latter, the action sequence $\{A(t)\}$ is generated in a distributed fashion. This implies, in particular, that the convergence of the algorithm should not be affected and the sequence $\{\theta_t\}$ generated by Q-SSA will still converge w.p.1. Furthermore, if all agents follow the same algorithm, the sequence of estimates $\{\theta_t\}$ will be the same for all agents and all will converge to the same Q-function Q^*_θ.

We thus obtain the following immediate corollary of Theorem 2. Once again, let π be a stationary joint policy and $(\mathcal{X}, \mathsf{P}_\pi)$ the corresponding Markov chain with invariant probability measure μ_π.

Theorem 4 *Let Γ be an MMDP as defined above and π a stationary joint policy so that the induced chain $(\mathcal{X}, \mathsf{P}_\pi)$ verifies the conditions of Theorem 2. Further let $\{\phi_i, i = 1, \ldots, M\}$ be a set of basis functions verifying the conditions of Theorem 2 and $\{\alpha_t\}$ the step-size sequence defined as*

$$\alpha_t(a) = \begin{cases} \frac{1}{n_t(a)}, & \text{if } a = a(t) \\ 0, & \text{otherwise} \end{cases}$$

for all $a \in \mathcal{A}$. Then, the conclusions of Theorems 2 and 3 hold for the sequence $\{\theta_t\}$ generated by Q-SSA when applied to Γ.

The above result establishes the convergence of Q-SSA in MMDPs and provides an asymptotic bound on the corresponding rate of convergence. This constitutes the first step in building an algorithm for simultaneous learning and coordination in MMDPs with infinite state-spaces. We next proceed to the problem of coordination.

4 The problem of coordination

As Q-SSA handles the problem of learning, we now address the problem of coordination. In particular, we discuss *convergence in behavior* as we extend BAP to cope with MMDPs with infinite state-spaces.

4.1 Biased adaptive play in MMDPs with infinite state-spaces

Recall the BAP mechanism described in Sect. 2.2. This coordination mechanism uses samples from the history of past plays to estimate the average policies of the agents in the game. These estimates are then used to choose a best response policy, as long as the game is known.

When applying standard BAP to MMDPs with a finite state-space, coordination on each state requires that such state be visited a sufficient number of times to ensure that (1) adequate action sampling can take place; and (2) there is sufficient time to attain coordination, since the convergence guarantees in Theorem 1 are asymptotic. Successive visits to a state provide each agent with a sample of the other agents' policies *in that particular state* and hence the requirement that *every* state be visited infinitely often must be satisfied.

Formally, the condition of infinite visits amounts to requiring the underlying Markov chain to be irreducible (every state is "visitable") and recurrent (each "visitable" state is visited infinitely often). In the infinite state-space case, we instead require the underlying Markov chain to be ψ-irreducible (meaning that all but a negligible part of the state-space is "visitable") and Harris recurrent (meaning that every "visitable" region of the state-space is visited infinitely often). We will return to this point further ahead in this section.

Consider an MMDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$. Due to the infinite nature of the state-space, it is generally impossible to ensure that any particular state in the state-space is visited infinitely often. Therefore, we cannot apply standard BAP as described in Sect. 2.2 to the infinite state-space setting.

In adapting BAP to cope with infinite state-spaces, coordination at each state should rely not only in past visits to that particular state but should also use the information provided by plays in *nearby states*. The intuition behind this idea can be easily clarified. Each agent k can no longer use the past history at a particular state x to infer the other agents' policy in that state, since there is the possibility that it was never visited before. Instead, agent k will assume that *the policies of the other agents in the states sufficiently close to x do not change significantly*. If this assumption holds, agent k can use the past history at nearby states to estimate the policy of the other agents at state x.

To implement this idea, we rely on the distance between two states x and y in $\mathcal{X}$ as an indication on the "similarity" between the states.[13] As will soon become apparent, this approach suitably adapts BAP to MMDPs with infinite state-spaces while ensuring coordination in all but a negligible part of the state-space.

4.2 Approximate coordination

We now describe *approximate biased adaptive play* (ABAP) and establish its convergence w.p.1. To this purpose, we consider a simplified setting in which coordination takes place independently of the control of the underlying Markov process. This has several advantages. First of all, it allows us to disregard several technicalities concerning the underlying Markov chain. Secondly, it greatly facilitates the proof of our convergence result. Finally, as will become apparent in the next section, it will allow an easy combination of ABAP with Q-SSA. We postpone to the conclusions (Sect. 7) a thorough discussion on the implications of this simplified setting.

We begin by introducing two fundamental properties of Markov chains, namely ψ-irreducibility and Harris recurrence. We refer to Appendix A for a more detailed review on Markov chains.

[13] Other concepts of "similarity" are, of course, possible. We return to this discussion in Sect. 7 and also in Appendix B.

Definition 3 Let $\{X(t)\}$ be a Markov chain with transition kernel P. The chain is *ψ-irreducible* if there is a maximal measure ψ on $\mathcal{X}$ such that $\psi(U) > 0$ if and only if

$$\sum_{t=0}^{\infty} \mathsf{P}^t(x, U) > 0 \tag{11}$$

for any $x \in \mathcal{X}$ and any measurable set $U \subset \mathcal{X}$.[14]

Intuitively, the above definition simply means that all sets U with positive ψ measure (and only those) have a positive probability of being visited by the process $\{X(t)\}$ at some point in time, independently of the initial state of the chain.

Definition 4 Let $\{X(t)\}$ be a ψ-irreducible Markov chain, and let η_U denote the number of visits to a set U in an infinite trajectory of the chain. The chain is *Harris recurrent* if

$$\mathbb{P}\left[\eta_U = \infty \mid X(0) = x\right] = 1, \tag{12}$$

for any $x \in \mathcal{X}$ and any measurable set $U \subset \mathcal{X}$.

Intuitively, the above definition means that sets U with positive ψ measure are visited an infinite number of times.

Let $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ be an MMDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and finite joint action-space $\mathcal{A}$. Let Q^* be the optimal Q-function for $\mathcal{M}$ and define, for each $x \in \mathcal{X}$, the stage-game $\Gamma_x^* = (N, (\mathcal{A}_k), Q^*(x, \cdot))$. To introduce and analyze ABAP, we resort to an auxiliary process $\{X(t)\}$ evolving in $\mathcal{X}$. For the moment, we disregard the nature of this process that will become clear in Sect. 5.

We assume this process $\{X(t)\}$ to be a ψ-irreducible and Harris recurrent Markov chain, with a irreducibility measure absolutely continuous w.r.t. the Lebesgue measure in $\mathbb{R}^p$.[15]

At every time instant t, the N agents in $\mathcal{M}$ engage in the repeated game $\Gamma^*_{X(t)}$ where $X(t)$ is the state of the auxiliary process at time t. The sole purpose of the agents is to coordinate in an optimal policy in each stage-game Γ_x^*; the agents have no knowledge otherwise on the MMDP $\mathcal{M}$ or on the auxiliary process $\{X(t)\}$ and consider the payoffs $Q^*(x, \cdot)$ at different stage-games Γ_x^* to be independent. This technical artifice allows us to discard the effect of the joint actions of the agents on the state evolution of the MMDP. The agents merely visit the states in $\mathcal{X}$ along the trajectories of $\{X(t)\}$ and coordinate in each visited stage-game Γ_x^*.

Define the *history of the game* at time t as

$$\mathcal{H}_t = \{x(0), a(0), x(1), a(1), \ldots, x(t-1), a(t-1), x(t)\},$$

where the subsequence $\{x(t)\}$ is a sample trajectory of the process $\{X(t)\}$ and each joint action $a(\tau), \tau < t$ corresponds to that chosen by the agents in the stage-game $\Gamma_{x(\tau)}$. At each time instant t, each agent determines the distance between the current state $x(t)$ and each state $x(\tau)$, $\tau < t$, occurring in $\mathcal{H}_t$, given by $\|x(t) - x(\tau)\|$ for some norm $\|\cdot\|$. It then chooses m occurrences from this history so as to minimize the corresponding distance. The

[14] Maximal in this context means that, for any other measure φ that also verifies (11), it holds that $\varphi \preceq \psi$. As seen in [31], the existence of at least one measure φ verifying (11) immediately implies the existence of one such measure that is maximal.

[15] As discussed in Sect. 7 and Appendix B, the requirement of absolute continuity of ψ w.r.t. μ^{Leb} ensures that the topology of the state-space is well-adapted to the usual topology of $\mathbb{R}^p$. This requirement arises from the definition of "similar states" in terms of a distance of the form $d(x, y) = \|x - y\|$.

sample set thus obtained, denoted as $S_m(x(t), \mathcal{H}_t)$, contains the m elements in $\mathcal{H}_t$ closer to $x(t)$, i.e., those minimizing

$$\sum_{i=1}^{m} \|x(t) - x(t_i)\| .$$

We remark that a particular state $x \in \mathcal{X}$ may occur in $S_m(x(t), \mathcal{H}_t)$ more than once. On the other hand, if two occurrences $x(t_i)$ and $x(t_j)$ verify

$$\|x(t) - x(t_i)\| = \|x(t) - x(t_j)\|$$

and only one such occurrence must be chosen, then the most recent one should be picked. In the example above, if $t_j > t_i$ then $x(t_j)$ would be chosen. We also notice that, due to the ψ-irreducibility and Harris recurrence of $\{X(t)\}$, given any state $x \in \mathcal{X}$ and a corresponding neighborhood U with positive ψ-measure, there is a time T_0 such that, w.p.1, $S_m(x, H_t) \subset U$ for all $t > T_0$.

Once the set $S_m(x(t), \mathcal{H}_t)$ is determined, the corresponding m plays can now be used to draw a K-sample and proceed as in standard BAP.

The following theorem establishes the convergence of ABAP w.p.1 in all but a negligible set of states.

Theorem 5 *Let $\mathcal{M}$ and $\{X(t)\}$ be as defined above. In particular, assume that $\{X(t)\}$ is ψ-irreducible and Harris recurrent with irreducibility measure absolutely continuous w.r.t. the Lebesgue measure, μ^{Leb}. Let Q^* be the optimal Q-function for $\mathcal{M}$, continuous in $\mathcal{X}$ in all but a ψ-null set of states. If the N agents in $\mathcal{M}$ engage in the coordination games described above while following ABAP, they will coordinate in an optimal policy w.p.1 at ψ-almost every state in $\mathcal{X}$, as long as the conditions for convergence of BAP (Theorem 1) are met.*

Proof See Appendix B.

Notice that Theorem 5 is somewhat more restrictive than its finite counterpart, as it requires continuity of Q^* ψ-almost everywhere. However, this condition simply ensures that the function Q^* is "well-behaved", so that coordination at a given point x can be achieved by observing the past plays in points "sufficiently close" to x.

It is also worth discussing the requirement of absolute continuity of the irreducibility measure ψ w.r.t. μ^{Leb}. This requirement ensures that the topology of the state-space $\mathcal{X}$ is "well-adapted" to the usual topology of $\mathbb{R}^p$, and greatly facilitates the verification that the set of points in which ABAP may not converge has null ψ measure. However, this is a technical condition assumed for commodity of proof and we expect Theorem 5 to hold in more general settings. We refer to Sect. 7 and Appendix B for further discussion on this issue.

In the following section we combine the Q-SSA algorithm described in the previous section with approximate biased adaptive play described in this section, yielding a method for simultaneous learning and coordination. Our construction closely follows that in [53].

5 Coordinated approximate Q-learning

In this section, we contribute one other novel algorithm that we refer as *coordinated approximate Q-learning* (CAQL). As anticipated in previous sections, this algorithm combines Q-SSA and ABAP. With sufficient exploration, CAQL guarantees that the estimates Q_θ

converge to an approximation of Q^* and the agents' policies converge to an optimal policy w.r.t. this obtained approximation.

In CAQL, the agents are *simultaneously* estimating/approximating Q^* and learning how to coordinate, by interleaving the iterations of Q-SSA and ABAP. ABAP depends critically on the output of Q-SSA and, as will soon become apparent, this requires a modification of the basic ABAP algorithm that ensures that ABAP can accommodate with the crude estimates of Q^* produced by Q-SSA during earlier iterations.

5.1 Combining Q-SSA and ABAP

As pointed out above, there is one main difficulty in combining ABAP with Q-SSA since, at each time instant t, the agents do not know the optimal function Q^* as assumed in Sect. 4. Instead, the agents have access only to an estimate Q_{θ_t} thereof that they must use to choose their actions and coordinate. This means that, during learning, some sub-optimal actions may appear as optimal and vice-versa, potentially posing difficulties to ABAP.[16] Therefore, having simultaneous learning and coordination while retaining the convergence guarantees of both Q-SSA and ABAP cannot be ensured merely by running ABAP and Q-SSA in parallel.

The intuitive idea to overcome this difficulty is to built an *approximate virtual game* from Q_{θ_t}. This virtual game will include not only the optimal actions according to the current estimate Q_{θ_t} but *all actions* that "appear" to be close to optimal. Then, as $t \to \infty$, sub-optimal actions are gradually removed.

To formalize this idea, we start with the following definition.

Definition 5 Given a general function $F : \mathcal{A} \to \mathbb{R}$, an action $a^* \in \mathcal{A}$ is *ε-optimal w.r.t. F* if

$$F(a^*) \geq \max_{a \in \mathcal{A}} F(a) - \varepsilon.$$

The basic procedure of CAQL is as follows. Let $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ be an MMDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and let π be a stochastic stationary joint policy. We refer to the process $\{X(t)\}$ induced by π as the *learning process* and to π as the *learning policy*. Suppose that $\{x(t)\}$, $\{a(t)\}$ and $\{r(t)\}$ are sampled trajectories of states, actions and rewards from $\mathcal{M}$ obtained by following the joint policy π. As in Sect. 3, we consider a set of linearly independent functions $\{\phi_i, i = 1, \ldots, M\}$ defined over $\mathcal{X}$ and taking values in $\mathbb{R}$. We denote by Q_{θ_t} the function

$$Q_{\theta_t}(x, a) = \phi^\top(x)\theta_t(a),$$

where the sequence $\{\theta_t\}$ is generated recursively according to the update rule

$$\theta_{t+1}(a) = (1 - \alpha_t(a))\theta_t(a) + \alpha_t(a)\phi(x(t))\left[r(t) + \gamma \max_{b \in \mathcal{A}} Q_{\theta_t}(x(t+1), b)\right]. \quad (13)$$

The functions Q_{θ_t} thus obtained provide successive approximations for the optimal Q-function Q^*. This component of CAQL, which simply applies Q-SSA with a stationary learning policy π, addresses the problem of learning. We must now combine it with the coordination mechanism implemented using ABAP. We consider the learning process $\{X(t)\}$ as the

[16] We note that optimal actions can be "ruled out" by BAP at earlier stages of learning, when its associated Q-value is still inaccurately estimated, which may impact negatively the performance of the algorithm. We refer to [53] for further discussion on this issue. It is also worth noting that without the requirement for simultaneous learning and coordination, both algorithms can be easily combined, as seen in [28].

auxiliary process used in ABAP (see Sect. 4), and use the successive estimates $\{Q_{\theta_t}\}$ instead of the actual function Q^* (which is unknown) to define the successive stage-games used in ABAP.

Let $\mathbf{opt}_t^{\varepsilon}(x)$ denote the set of all ε-optimal actions w.r.t. Q_{θ_t} at state x, i.e.,

$$\mathbf{opt}_t^{\varepsilon}(x) = \left\{ a^* \in \mathcal{A} \mid Q_{\theta_t}(x, a^*) \geq \max_{a \in \mathcal{A}} Q_{\theta_t}(x, a) - \varepsilon \right\}.$$

and define the virtual game $\widehat{VG}_t^{\varepsilon} = (N, (\mathcal{A}_k), \hat{r}_t)$, where

$$\hat{r}_t(a) = \begin{cases} 1 & \text{if } a \in \mathbf{opt}_t^{\varepsilon}(x(t)) \\ 0 & \text{otherwise.} \end{cases}$$

As in Sect. 4, at each time-step t the N agents in $\mathcal{M}$ engage in the repeated game $\widehat{VG}_t^{\varepsilon}$. The sole purpose of the agents is, once again, to coordinate in an optimal policy in each game $\widehat{VG}_t^{\varepsilon}$, discarding any knowledge otherwise on the MMDP $\mathcal{M}$. We denote the sequence of joint actions thus obtained by $\{\hat{a}(t)\}$ and now define the *history of the game* at time t as

$$\mathcal{H}_t = \{x(0), \hat{a}(0), x(1), \hat{a}(1), \ldots, x(t-1), \hat{a}(t-1), x(t)\}.$$

Notice that the sequence of actions $\{a(t)\}$ obtained according to π is never considered in the process of coordination.

Given the history $\mathcal{H}_t$, each agent now proceeds as in ABAP (see Sect. 4). In particular, each agent computes the set $S_m^k(x(t), \mathcal{H}_t)$ by determining the m states in $\mathcal{H}_t$ closer to $x(t)$. Each agent then draws a K-sample h from $S_m(x(t), \mathcal{H}_t)$ to be used to determine the expected payoff of each action $a_k \in \mathcal{A}_k$ w.r.t. the virtual game $\widehat{VG}_t^{\varepsilon}$. As in BAP and ABAP, the K-samples are drawn individually by each agent and independently of the other agents. Each agent can now find the corresponding best response action a_k^*, unless the two BAP conditions described in Sect. 2.2 are met. For commodity, we repeat such conditions here:

1. There is a joint action $a \in \mathbf{opt}_t^{\varepsilon}(x(t))$ such that, for all played actions $\hat{a} \in h$, $\hat{a}_{-k} = a_{-k}$; and
2. There is at least one played action $\hat{a} \in h$ such that $\hat{a} \in \mathbf{opt}_t^{\varepsilon}(x(t))$.

Notice that the virtual games $\widehat{VG}_t^{\varepsilon}$ can not be stored in memory, since there may be infinitely many of them (due to the fact that $\mathcal{X}$ is infinite). This, however, poses no difficulty, as each such game can easily be determined from Q_{θ_t} as needed.

Given all individual actions at time t, $(\hat{a}_1(t), \ldots, \hat{a}_N(t)) = \hat{a}(t)$, each agent updates the history of the game, $\mathcal{H}_t$, and the game moves to a new state $x(t+1)$ according to the probabilities P_{π} defined by the learning policy π. All agents receive the corresponding reward $r(t)$ and use the observed transition $(x(t), a(t), r(t), x(t+1))$ to update each parameter vector θ_t using the update rule (13). Notice that the sequence of actions $\{\hat{a}(t)\}$ obtained from ABAP is never considered in the process of learning.

Finally, making $\varepsilon \to 0$ at an adequate rate, we expect that all suboptimal actions are eventually discarded from the virtual games $\widehat{VG}_t^{\varepsilon}$, hopefully ensuring convergence to an optimal policy with respect to Q_{θ^*}. The rate at which $\varepsilon \to 0$ must take into account the rate of convergence of the learning algorithm, i.e., the rate at which the sequence θ_t converges to the corresponding limit point θ^*. This issue is addressed in the continuation, as we establish the convergence of CAQL.

With this, we conclude the description of the CAQL algorithm. Figure 1 provides a pseudo-code description of CAQL.

5.2 Convergence of CAQL

To establish convergence of CAQL, we need the bound for the rate of convergence of Q-SSA from Theorem 3. Notice that in the CAQL algorithm the rate at which ε decays to zero is determined by the function B. To establish convergence of CAQL, we must ensure that such decay takes place at an adequate rate, given the convergence of Q-SSA. Using the bound in Theorem 3, we are in position to formalize all conditions for the convergence of CAQL.

Algorithm 1 The CAQL algorithm for one agent k. ABAP is implemented in instructions 9 through 22 and Q-SSA in instruction 25. The function $B : \mathbb{R} \to \mathbb{R}$ in instruction 28 controls the decay of ε to zero.

Initialization:

1: Initialize $t = 0$, $\mathcal{H}_0 = \{x(0)\}$ and $\varepsilon = \varepsilon_0$
2: **for all** $a \in \mathcal{A}$ **do**
3: Initialize $n_t(a) = 1$
4: Initialize $\theta_t(i, a) = 0$
5: **end for**

Coordination: Given current state $x(t)$

6: **if** $t \leq m$ **then**
7: Randomly select an action
8: **else**
9: Determine $\widehat{VG}_t^{\varepsilon} = (N, (\mathcal{A}_k), \hat{r}_t)$, with

$$\hat{r}_t(a) = \begin{cases} 1 & \text{if } a \in \mathbf{opt}_t^{\varepsilon}(x(t)) \\ 0 & \text{otherwise} \end{cases}$$

10: Define $D = \left\{a \in \mathcal{A} \mid \hat{r}_t(a) = 1\right\}$
11: Compute $H_t = S_m^k(x(t), \mathcal{H}_t)$
12: Compute $h = \mathsf{K}(H_t)$
13: **for all** $a_k \in \mathcal{A}_k$ **do**
14: Compute

$$EP_t(a_k) = \sum_{a_{-k} \in \mathcal{A}_{-k}} \hat{r}_t(a_{-k}, a_k) \frac{n_h(a_{-k})}{K}$$

15: **end for**
16: Compute $BR_t = \left\{ a_k \mid a_k = \arg\max_{b_k \in \mathcal{A}_k} EP_t(b_k) \right\}$
17: **if** Conditions 1 and 2 hold **then**
18: Choose the most recent joint action in $h \cap D$
19: **else**
20: Randomly choose an action in BR_t
21: **end if**
22: Update $\mathcal{H}_t \leftarrow \mathcal{H}_t \cup \left\{x(t), \hat{a}(t)\right\}$;
23: **end if**

Learning: Given current transition triplet $(x(t), a(t), x(t+1))$

24: $n_{t+1}(a(t)) \leftarrow n_t(a(t)) + 1$;
25: Update θ_t according to (13);
26: $t \leftarrow t + 1$;
27: **if** $\varepsilon \geq \varepsilon_0 B(n_t)$ **then**
28: $\varepsilon = \varepsilon_0 B(n_t)$;
29: **end if**

Theorem 6 *Let $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ be an MMDP with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and finite action-space $\mathcal{A}$. Let π be a stationary joint policy such that the Markov chain $(\mathcal{X}, \mathsf{P}_\pi)$ is geometrically ergodic with invariant probability measure μ_π absolutely continuous w.r.t. μ^{Leb}. Let $\{\phi_i, i = 1, \ldots, M\}$ be a set of basis functions verifying the conditions of Theorem 2 and continuous μ_π-almost everywhere. Further assume that*

1. *The function B decreases monotonically to zero and verifies*

$$\lim_{t \to \infty} \frac{\mathcal{E}_t}{B(t)} = 0. \tag{14}$$

2. *The step-size sequence $\{\alpha_t\}$ verifies the conditions of Theorem 3;*
3. *The cardinality m of the sets $S_m(x, \mathcal{H}_t)$ and the length K of the K-samples verify $m \geq K(N + 2)$.*

Then, the sequence $\{\theta_t\}$ generated by CAQL converges w.p.1 to θ^* as defined in (10). Furthermore, all agents converge w.p.1 to an optimal joint policy w.r.t. the obtained approximation Q_{θ^*} for μ_π-almost every $x \in \mathcal{X}$.

Proof See Appendix B.

Let us briefly go over the statement of the above theorem. In CAQL, ABAP must use the estimates for Q^* coming out of Q-SSA. At the earliest steps of learning, these estimates are typically very crude and it is possible that some sub-optimal action appear as optimal and vice-versa. If this is the case, ABAP in its original formulation would construct a virtual game in which the actual optimal actions are not valued but, instead, the suboptimal actions are. This can cause a bias in the ABAP estimation process and the convergence guarantees provided for CAQL would no longer hold.

The result on Theorem 3 provides an upper bound on the estimation error associated with any given action (optimal or not) at iteration t. Therefore, when running ABAP at iteration t, we require that

1. The virtual game values all actions that are within ε of the (apparently) optimal actions;
2. ε is larger than the bound in Theorem 3;
3. ε converges to zero.

If the above conditions are met, all optimal actions (and probably some suboptimal actions) are valued in the virtual game at all iterations. As $t \to \infty$, the bound in Theorem 3 goes to 0, translating the fact that Q-SSA converges. By ensuring that $\varepsilon \to 0$ without violating (14), we ensure that all sub-optimal actions are eventually ruled out of the virtual games used by ABAP. This, in turn, guarantees the latter to coordinate in an optimal action.

In practice, the algorithm does not depend critically on the value of ε as long as it verifies the conditions in the above theorem. A crude approximation of the bound in Theorem 3 can easily be found and, therefore, it is relatively simple to set the decay schedule for ε. We refer to [53], where a similar artifact is used to combine the original BAP mechanism with model-based learning in a finite setting.

We conclude by noting that the requirements of ψ-irreducibility and Harris recurrence in ABAP follow from the assumption of geometric ergodicity.

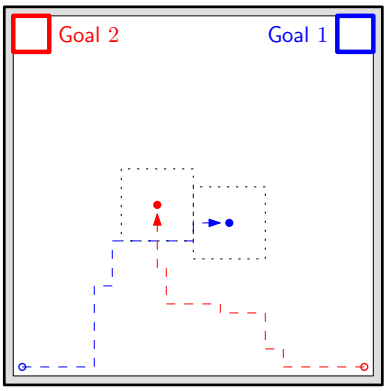


Fig. 2 Example of a continuous indoor environment. The *dotted black squares* represent the "crash area" around each robot

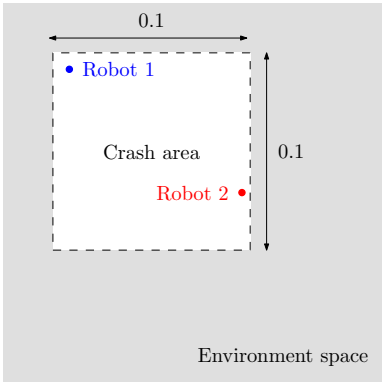


Fig. 3 Situation of possible crash

6 An illustrative example

We now analyze an example of application of CAQL to two simple multi-robot navigation tasks.

Consider the indoor environment in Fig. 2. Two mobile robots (1 and 2) must navigate to the corresponding goal regions, signaled with the bold colored lines. The environment is a 1×1 square, and the state of each robot k at each time instant is a pair $(\mathbf{x}_k, \mathbf{y}_k)$ of coordinates.[17] The coordinates of the corners in the goal regions are (1, 1) and (0, 1), respectively, and the corresponding goal regions are 0.1×0.1 squares, as depicted in Fig. 2. We denote the goal region for robot k by G_k and by G the Cartesian product of G_1 and G_2. In their trajectories, the robots must learn not to crash into each other by avoiding lying in the same 0.1×0.1 area simultaneously (see Fig. 3 for an illustration). We denote the state of robot k at time t by $X_k(t), k = 1, 2$. The state of the group is thus a pair $X(t) = (X_1(t), X_2(t))$ and can take any value in $([0; 1] \times [0; 1]) \times ([0; 1] \times [0; 1])$.

Each robot has four individual actions available, namely moving N, S, E and W. Each individual action moves the robot a random distance between 0 and 0.3 in the corresponding direction. We consider the movements of the robots to be independent of each other.

This problem can be modeled by the MMDP $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ where

[17] We use boldface symbols $\mathbf{x}$ and $\mathbf{y}$ to denote the physical coordinates of one robot to distinguish these from the symbols x and y used to denote generic elements of the state-space $\mathcal{X}$.

- $N = 2$;
- $\mathcal{X} = ([0; 1] \times [0; 1]) \times ([0; 1] \times [0; 1])$;
- $\mathcal{A}_k = \{N, S, E, W\}$ for $k = 1, 2$;
- The transition probabilities are defined by a transition kernel P given by

$$\mathsf{P}_a(x, U) = \mathsf{P}_{a_1}(x_1, U_1)\mathsf{P}_{a_2}(x_2, U_2)$$

  where $a = (a_1, a_2)$, $x = (x_1, x_2)$ and U is any measurable subset of $\mathcal{X}$ of the form $U = U_1 \times U_2$. $\mathsf{P}_{a_k}(x_k, U_k), k = 1, 2$, denotes the single-robot transition probabilities for robot k according to the description above, where U_k represents a measurable subset of $\mathcal{X}_k$;
- The reward function r is defined as

$$r(x, a, y) = \begin{cases} 20 & \text{if } y \in G; \\ -10 & \text{if } \|y_1 - y_2\|_\infty < 0.1; \\ 0 & \text{otherwise;} \end{cases}$$

- We consider $\gamma = 0.95$.

Note that, in the above MMDP, we consider that the *joint goal* of the robots is to reach their goal positions while avoiding crashing into each other.

We tested our algorithm by allowing the agents to interact with the environment and one another in two distinct phases. In a first phase, the *learning phase*, the agents were allowed to explore and learn during 10^4 time steps. During this learning stage, the robots applied CAQL with sampled trajectories from the game obtained using a uniform random policy. Because of the finite learning time, we used a non-vanishing step-size sequence (see Fig. 4). We also used a finite history of past plays, $\mathcal{H}_t$, with a maximum length of 1,000.

In a second phase, the *test phase*, the agents were again placed in the environment and allowed to execute the policy learned in the learning phase in episodes of 100 consecutive time-steps each, during which we evaluated the performance of the group. To evaluate the performance of the group, we collected four different statistics, namely

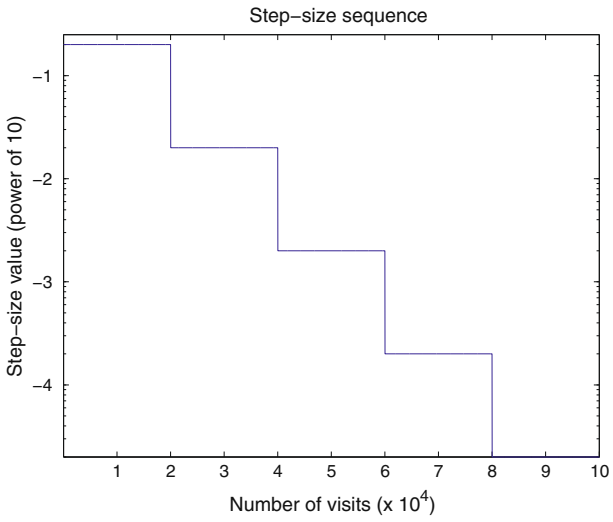


Fig. 4 Step-size sequence $\{\alpha_t(a)\}$ as a function of the number of visits to action a, $n_t(a)$. The labels in the vertical axis correspond to powers of 10

- *Total discounted reward* that should provide a good estimator on how good the learned policy is. In a sense, it also provides some information on the quality of the approximation Q_{θ^*} obtained with Q-SSA. The two next statistics provide further details on the quality of the obtained approximation.
- *Number of negative rewards* that details how well the robots learn to actually avoid the undesired crashes.
- *Percentage of successful runs* detailing the percentage of episodes in which the robots were able to reach the desired configuration.
- *Number of suboptimal actions* that details how well the ABAP algorithm was able to tackle the coordination problems for this game.

For the purpose of comparison we also provide the results obtained

1. Using the approximation obtained with Q-SSA but with a centralized controller—corresponding to the *centralized solution* in terms of coordination;
2. Using the approximation computed using Q-SSA but ignoring the ABAP coordination policy and just letting each robot choose its own action greedily—corresponding to the *uncoordinated solution*.

The comparison of the three different policies (dubbed *CAQL*, *Centr.*, and *Uncoord.*) provides a good evaluation of the impact of ABAP in ensuring coordination in this particular problem.

For comparison purposes, we ran two different experiments, each using a different partition of the state-space and each consisting in the aforementioned learning and test phases. In the first experiment, we used a hard partition consisting of 81 mutually exclusive sets. This partition can be implemented simply by using step-functions, as depicted in Fig. 5. We summarize the results from this first experiment in Table 1.

In a second experiment, we used a soft-partition in which the state-space was also divided into 81 "soft" sets, obtained as the support of Gaussian kernels. Figure 5 illustrates the relation between hard and the soft partitions for a 1-dimensional state-space. Since in both experiments the basis functions provide similar partitions of the state-space, we expect the results

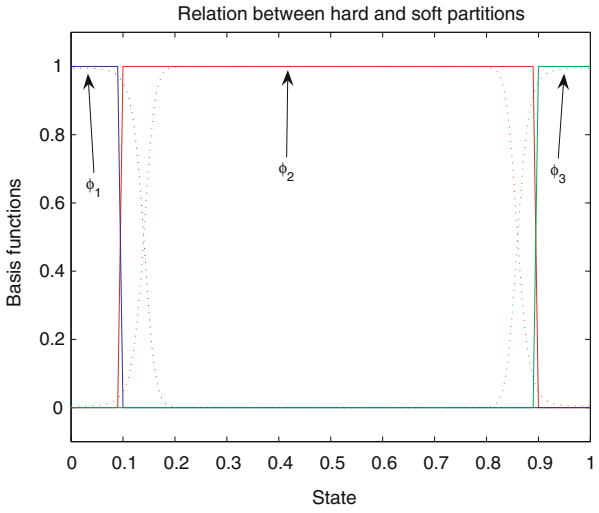


Fig. 5 Relation between hard and soft partitions in a 1-dimensional state-space. The *solid line* corresponds to the hard partition and the *dotted line* to the soft partition. The sets in the partition are defined as the *support* of the different basis functions

Table 1 Comparative results of CAQL against the optimal and uncoordinated policies in the navigation domain, using a hard state-space partition with 81 basis functions

	Tot. disc. reward	Miscoord.	Succ. runs (%)	Subopt. act.
CAQL	29.3 ± 5.17	1.0 ± 0.98	100	10.7 ± 3.80
Centr.	32.8 ± 3.85	1.6 ± 1.15	100	0.0 ± 0.00
Uncoord.	24.5 ± 5.95	3.5 ± 2.71	100	30.0 ± 4.07

The reported results were obtained *after* the learning period was complete. We present the average results obtained over 1, 000 Monte-Carlo runs

Table 2 Comparative results of CAQL against the optimal and uncoordinated policies in the navigation domain, using a soft state-space partition with 81 basis functions

	Tot. disc. reward	Miscoord.	Succ. runs (%)	Subopt. act.
CAQL	29.1 ± 4.02	0.1 ± 0.30	100	17.0 ± 3.78
Centr.	30.2 ± 7.49	0.0 ± 0.00	100	0.0 ± 0.00
Uncoord.	17.4 ± 7.26	3.5 ± 2.65	97.0	43.9 ± 4.89

The reported results were obtained *after* the learning period was complete. We present the average results obtained over 1, 000 Monte-Carlo runs

in both experiments to be similar. We summarize the results from the second experiment in Table 2.

Several remarks are in order. First of all, concerning the function Q_{θ^*} computed by the Q-SSA component of CAQL, the results in both experiments indicate that the algorithm was able to learn a "successful" approximation of the optimal Q-function. In fact, observing the performance of the centralized team ("Centr.") team both in terms of successful runs and miscoordinations, we conclude that the learned policy allows the robots to successfully fulfill the intended task, reaching the respective goals and avoiding crashes. We classify the obtained approximation as "successful" since we are unable to determine the true optimal Q-function and, therefore, we can only evaluate the quality of the approximation by the quality of the obtained policy.

A second remark is concerned with the ABAP component of CAQL. As seen in Tables 1 and 2, CAQL clearly outperforms the uncoordinated team in both experiments.[18] This, as observed before, is a clear indication that the ABAP coordination mechanism effectively addresses the problem of coordination in the presence of multiple equilibria in such MMDPs. This is particularly clear by observing the number of suboptimal actions performed by CAQL and the uncoordinated team.

It is also worth mentioning that the suboptimal actions observed when running CAQL may arise from two different factors. Considering that the convergence results for ABAP are asymptotic,

– The finite time and history may influence the ability of the algorithm to coordinate in some regions of the state-space that have been seldom visited.

[18] As a quick note, it is worth pointing out that, given the sample size (1, 000), the differences between the different methods in both Tables 1 and 2 are statistically very significant. Note, for example, that the 99.9% confidence interval for the average total discounted reward of CAQL in Table 1 is [29.26, 29.34] while for the uncoordinate team is [24.45, 24.55].

– Since the policy used during learning was uniform, it is possible that not all regions of the state-space were visited equally thoroughly.

Another interesting observation is that, even though using both sets of functions the results are very similar, the total discounted reward received is slightly larger in the experiment using a hard partition (Table 1) while the number of mis-coordinations is slightly smaller in the experiment using a soft partition (Table 2). This means that, while the hard partition is apparently more efficient for navigation toward the goal regions, the soft partition seems to be more efficient in preventing crashes between the robots. The observed difference also confirms our previous observation that the performance of the algorithm will greatly depend on the choice of basis functions.

Finally, we notice that, in the experiment with a soft-partition (Table 2), the suboptimal actions arising from the lack of a coordination mechanism in the "uncoordinated team" greatly impact the ability of the robots to reach their goal. This is evident both in terms of total discounted reward and in terms of successful runs.

We conclude this section with the results obtained in the environment of Fig. 6 that further illustrate some of the points discussed above. Once again, two mobile robots must navigate to the corresponding goal regions, signaled with the bold colored lines and the problem is, in all aspects, similar to the one considered above. The fundamental difference is that, in this environment, the robots have only available four narrow corridors that have the exact width of the "crash area". This means that the two robots can only cross the same hallway in opposite directions if they "stick" to opposite walls, as depicted in Fig. 7.

We again tested CAQL by allowing the agents to explore and learn for 10^4 time steps (the learning phase), after which the learned policy is evaluated in episodes of 100 consecutive time-steps each (the test phase). We again tested our method using both hard and soft partitions of the state-space, this time consisting of 64 sets. We summarize the results obtained with each partition in Tables 3 and 4.

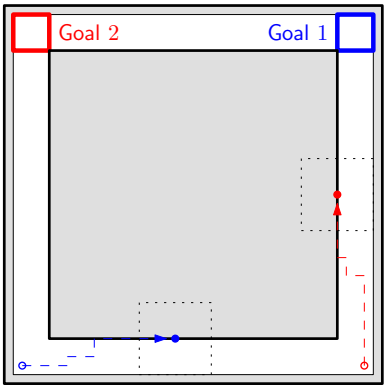


Fig. 6 Continuous indoor environment with narrow hallways. As in Fig. 2, the *dotted black squares* represent the "crash area" around each robot. Note that this particular environment is much more coordination critical, since the robots can hardly cross the same hallway in opposite directions without crashing

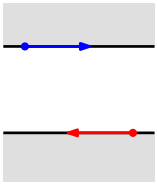


Fig. 7 The two robots can only cross the same hallway in opposite directions without crashing if they "stick" to opposite walls

Table 3 Comparative results of CAQL against the optimal and uncoordinated policies in the navigation domain of Fig. 6, using a hard state-space partition with 64 basis functions

	Tot. disc. reward	Miscoord.	Succ. runs (%)	Subopt. act.
CAQL	36.7 ± 4.50	1.8 ± 1.28	100	0.01 ± 0.03
Centr.	36.8 ± 4.25	1.8 ± 1.19	100	0.00 ± 0.00
Uncoord.	30.1 ± 7.77	2.5 ± 1.28	100	4.42 ± 1.59

The reported results were obtained *after* the learning period was complete. We present the average results obtained over 1, 000 Monte-Carlo runs

Table 4 Comparative results of CAQL against the optimal and uncoordinated policies in the navigation domain of Fig. 6, using a soft state-space partition with 64 basis functions

	Tot. disc. reward	Miscoord.	Succ. runs (%)	Subopt. act.
CAQL	10.5 ± 13.60	0.3 ± 0.45	54.7	0.00 ± 0.00
Centr.	10.8 ± 13.47	0.3 ± 0.43	55.7	0.00 ± 0.00
Uncoord.	5.1 ± 9.12	0.3 ± 0.82	41.1	14.54 ± 17.28

The reported results were obtained *after* the learning period was complete. We present the average results obtained over 1, 000 Monte-Carlo runs

In this scenario, CAQL exhibits a performance that is essentially similar to that of the centralized controller. This is visible with both sets of functions, but it is particularly flagrant in the results using a soft partition.

Another interesting aspect is that, in this scenario, the difference between the total discounted reward attained with a hard partition (Table 3) and that achieved using a soft partition (Table 4) is significantly more pronounced than in the previous scenario. This can be explained by noticing that the geometric nature of the corridors is more amenable to a hard partition of the state-space that closely replicates the geometry of the environment. Interestingly, once again the number of mis-coordinations is inferior in the experiment using a soft partition, reinforcing the idea that the hard partition is apparently more efficient for navigation toward the goal regions while the soft partition seems to be more efficient in preventing crashes between the robots.

7 Discussion

We now discuss several important issues referred along the text and postponed to these concluding remarks.

7.1 Quality of approximation

As mentioned in Sect. 3, Q-SSA computes an approximation to Q^* that verifies

$$Q_{\theta^*}(x,a) = \phi^\top(x)\mathbb{E}_{\mu_\pi}\left[\phi(z)(\mathbf{H}Q_{\theta^*})(z,a)\right].$$

Generally, such function will be different from Q^*. In fact, defining the operator

$$(\Phi q)(x,a) \triangleq \phi^\top(x)\mathbb{E}_{\mu_\pi}\left[\phi(z)q(z,a)\right]$$

we can rewrite the above fixed-point equation as

$$Q_{\theta^*}(x,a) = (\Phi \mathbf{H} Q_{\theta^*})(x,a). \tag{15}$$

Observing (15) it should be clear that $Q^* \neq Q_{\theta^*}$ unless if Q^* is also a fixed point of Φ.

In spite of this apparently discouraging fact, further insight can be obtained by exploring the relation between Q^* and Q_{θ^*}. First of all we recall that, by assumption,

$$\|\phi(x)\|_1 \triangleq \sum_i |\phi_i(x)| = 1,$$

which implies that $\|\Phi q\|_\infty \leq \|q\|_\infty$. Therefore,

$$\begin{aligned} \left\| Q_{\theta^*} - Q^* \right\|_\infty &\leq \left\| Q_{\theta^*} - \Phi Q^* \right\|_\infty + \left\| \Phi Q^* - Q^* \right\|_\infty \\ &= \left\| \Phi \mathbf{H} Q_{\theta^*} - \Phi \mathbf{H} Q^* \right\|_\infty + \left\| \Phi Q^* - Q^* \right\|_\infty \\ &\leq \gamma \left\| Q_{\theta^*} - Q^* \right\|_\infty + \left\| \Phi Q^* - Q^* \right\|_\infty . \end{aligned}$$

From here, we obtain

$$\left\| Q_{\theta^*} - Q^* \right\|_\infty \leq \frac{1}{1-\gamma} \left\| \Phi Q^* - Q^* \right\|_\infty .$$

The expression above provides an upper bound on the error in the obtained approximation in terms of the difference between ΦQ^* and Q^*. Interpreting ΦQ^* as a "pseudo-projection", $\|\Phi Q^* - Q^*\|_\infty$ can be seen as the "distance" between Q^* and the family of functions $\mathcal{Q}$ spanned by $\{\phi_i, i = 1, \ldots, M\}$.

Another interesting aspect about the obtained approximation concerns an associated decision problem, as seen in [40]. Let $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ be an MDP and $\{\phi_i, i = 1, \ldots, M\}$ a set of basis functions verifying the conditions in Sect. 3. Define the stochastic process $\{Z(t)\}$, where each random variable $Z(t)$ takes values in $\{1, \ldots, M\}$. This new process is governed by the probabilities

$$\mathbb{P}\left[Z(t) = i \mid \mathcal{F}_t\right] = \phi_i(X(t)),$$

where $\mathcal{F}_t$ is the σ-algebra generated by $\{X(0), \ldots, X(t), A(0), \ldots, A(t)\}$. Taking $Z(t)$ as an *indirect observation* of the state $X(t)$, the joint process $\{X(t), Z(t)\}$ is a *partially observable Markov decision process* with a finite observation space $\mathcal{O} = \{1, \ldots, M\}$ and compact state-space $\mathcal{X} \subset \mathbb{R}^p$.

Consider now the Markov chain induced by the learning policy, $(\mathcal{X}, \mathsf{P}_\pi)$, and let μ_π denote the corresponding invariant measure. From the stationary version of $(\mathcal{X}, \mathsf{P}_\pi)$ we can build a stationary version of $\{Z(t)\}$ where, for all t,

$$\mathbb{P}\left[Z(t) = i\right] = \mathbb{E}_{\mu_\pi}\left[\phi_i(x)\right]. \tag{16}$$

From this stationary process $Z(t)$, we define a general (non-Markov) decision process $\hat{M} = (\mathcal{O}, \mathcal{A}, \hat{\mathsf{P}}, \hat{r}, \gamma)$, where

- $\mathcal{O}$ is the observation space, $\{1, \ldots, M\}$;
- $\mathcal{A}$ is the action space of the original MDP $\mathcal{M}$;
- $\hat{\mathsf{P}}$ defines the "average" transition probabilities, given by

$$\begin{aligned} \hat{\mathsf{P}}_a(i,j) &= \mathbb{E}_{\mu_\pi}\left[\phi_i(x)\phi_j(y)\right] \\ &\triangleq \int_{\mathcal{X}} \phi_i(x) \int_{\mathcal{X}} \phi_j(y) \mathsf{P}_a(x, \mathrm{d}y) \mu_\pi(\mathrm{d}x). \end{aligned}$$

– $\hat{r}$ defines the "average" reward, given by

$$\hat{r}(i,a) = \mathbb{E}_{\mu_\pi}[\phi_i(x) r(x,a,y)] \triangleq \int_{\mathcal{X}} \phi_i(x) \int_{\mathcal{X}} r(x,a,y) \mathsf{P}_a(x, \mathrm{d}y) \mu_\pi(\mathrm{d}x).$$

– γ is the discount factor of the original MDP $\mathcal{M}$.

Let us discuss a little the meaning of the several elements above. Consider, for example, the probabilities $\hat{\mathsf{P}}_a(i,j)$. As can easily be checked,

$$\sum_j \hat{\mathsf{P}}_a(i,j) = \mathbb{E}_{\mu_\pi}[\phi_i(x)]$$

and, in general, the term on the right-hand side is different from 1. In fact, $\hat{\mathsf{P}}_a(i,j)$ represents not the probability of moving to state j given that the process started in state i and action a was taken, but the probability of observing a transition from i to j given that the action a was taken and the process is distributed according to (16).

Returning to Q-SSA, some algebraic manipulation leads to the conclusion that the parameter θ^* computed by the algorithm verifies the following recursion:

$$\theta^*(i,a) = \hat{r}(i,a) + \gamma \sum_{j=1}^{M} \hat{\mathsf{P}}_a(i,j) \max_b \theta^*(j,b)$$

and can be interpreted as the "optimal" Q-function for $\hat{\mathcal{M}}$.

We conclude with two relevant remarks. First of all, learning with function approximation in the cooperative scenarios considered in this paper is not excessively different from the same problem in single-agent scenarios. This is evidenced by the results in Sect. 3, in which the result for cooperative multiagent scenarios follows immediately from its counterpart in single-agent scenarios. Therefore, it is expectable that our results can be extended to other approximation architectures for which convergence results are available (e.g., point-based approximations [33,45]), although requiring the result in Theorem 3 to be extended to these methods.

Secondly, as argued above, the use of function approximation generally implies that the optimal Q-function, Q^*, cannot be recovered exactly. This impacts the quality of the obtained approximation. In cooperative multiagent settings, this means that the agents will coordinate in an "approximately optimal" policy. Guarantees on the quality of such approximately optimal policies follow from similar bounds in single-agent problems such as those found in [2]. However, to our knowledge, it remains an open question how the use of function approximation impacts the set of equilibria in general (non-cooperative) games.

7.2 "On-policy" coordination

When describing the ABAP algorithm, we considered an auxiliary process that "isolated" the transitions of the chain from the coordination process. In this sense, the coordination mechanism was "off-policy", in that the actions of the agents did not affect the dynamics of the underlying Markov chain. As argued before, the purpose of this mathematical device was to avoid any concerns about the dependence of the behavior of the underlying Markov chain on the action choice of the agents.

When combining ABAP with Q-SSA, we again resorted to such a process: we considered a learning policy driving the trajectories of the MMDP that was independent of the process of coordination. This separated the problem of learning from that of coordinating, ensuring that the same algorithm could tackle both problems simultaneously and in parallel.

In the case of a finite state-space, biased adaptive play can be combined with learning algorithms with no need for such artifice [26,53]. This is essentially due to two reasons:

1. Many classical algorithms can compute Q^* independently of the learning policy used (e.g., Q-learning);
2. BAP ensures coordination in the optimal policy even in the presence of exploratory moves [53].

Therefore, the combination of BAP with, for example, Q-learning, can be implemented taking only a few minor precautions to ensure proper exploration and adequate tuning of the ε parameter (as in CAQL).

In the case of infinite state-spaces, *off-policy* RL methods with function approximation exhibit unsound behavior when a changing learning policy is used [51]. This has an immediate implication: since in ABAP the agents necessarily adjust their policy toward an optimal policy, such methods can only be used as we did here, by separating learning and coordination. On the other hand, if convergence guarantees for *on-policy* RL methods can be established [30,35], it is unclear how such methods can be combined with ABAP. This difficulty arises even in the benign case of finite state-spaces, when combining BAP with on-policy learning methods. We refer to [26] for further discussion on this topic.

7.3 Storage of infinite histories

The ABAP algorithm, as formulated in Sect. 4, stores the *complete history* of the coordination process. This is fundamental in establishing the asymptotic convergence properties of the algorithm, since only the complete history of the process can guarantee that there are sufficient samples of "every" neighborhood in the state-space.

In practice, however, storing the potentially large trajectories of the algorithm is often impractical, both in terms of memory consumption and in terms of search. Therefore, actual implementations of CAQL as the one illustrated in Sect. 6 must usually "disregard" such requirement and use a fixed-size history. The latter should be chosen sufficiently large to ensure proper sampling of the state-space, but our experience showed that the impact of a fixed-size history is often negligible in many applications. The "optimal" history length will depend on the invariant measure induced by the learning policy, μ_π, and on the support of the Q-function used for coordination.

7.4 Absolute continuity of ψ with respect to μ^{Leb}

Another requirement in Theorems 5 and 6 is related with the absolute continuity of μ_π with respect to the Lebesgue measure, μ^{Leb}. As mentioned in Sect. 4, this requirement ensures that chain $(\mathcal{X}, \mathsf{P}_\pi)$ is well-adapted to the concept of "nearby states" used in ABAP.

To clarify this observation, we recall that the fundamental idea in ABAP is to use the policy of other agents in *nearby states* to coordinate in any given state x. This relies on two implicit assumptions, namely

1. The dynamics of the underlying Markov process are "similar" in such nearby states;
2. The optimal policy for the game is "similar" in such nearby states.

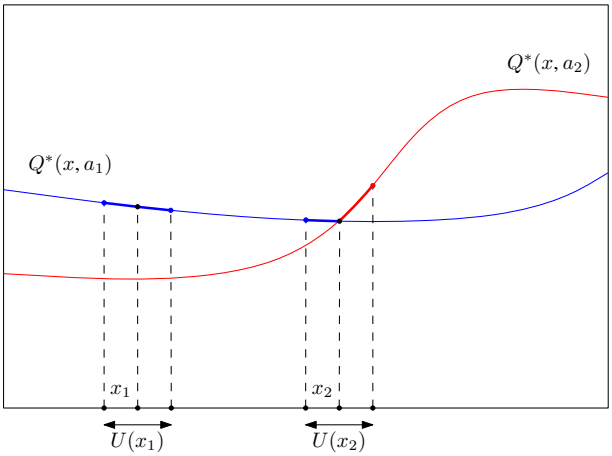


Fig. 8 "Pathological example", in which the existence of an atom $\{x_2\}$ may prevent coordination

Concerning 1, consider the chain $\{X(t)\}$ along which the agents coordinate in CAQL.[19] As seen in Sect. 5, this process is not driven by the actions of the agents and is purposely kept separated from the coordination mechanism. Coordination in CAQL is considered in terms of Q_{θ^*} and, as seen before, this approximation depends on the long-term behavior of $\{X(t)\}$, encapsulated in its invariant measure, μ_π. On the other hand, by using any of the usual metrics in $\mathbb{R}^p$, we are implicitly imposing the standard topology in $\mathbb{R}^p$, to which the Lebesgue measure is naturally adapted. Therefore, absolute continuity of μ_π w.r.t. μ^{Leb} immediately ensures that μ_π is well-adapted to the underlying topology imposed by the metric.

The above requirement also excludes chains that exhibit certain undesirable "pathological behaviors". Consider, for example, the function depicted in Fig. 8. There is an action a_1 that is optimal for all states in a neighborhood of x_1 and, therefore, samples from this neighborhood can be used to coordinate in x_1. This does not happen in x_2: any neighborhood of x_2 will include states in which only a_1 is optimal and states in which only a_2 is optimal. If μ_π is absolutely continuous w.r.t. μ^{Leb}, then the set $\{x_2\}$ will have 0 μ_π-measure. However, if this is not the case, it is possible to have an "atom" of probability, $\{x_2\}$, that may prevent coordination in this particular state.

Two final remarks are in order. First of all, CAQL can be modified to accommodate any general similarity criterion (e.g., using kernels). Assuming such criterion to induce a complete and separable topology $\mathcal{T}$ on $\mathcal{X}$, the condition of absolute continuity w.r.t. μ^{Leb} should then be replaced by the condition that μ_π be defined on the Borel σ-algebra associated with $\mathcal{T}$.

Secondly, it is our conviction that the requirement of absolute continuity of μ_π w.r.t. μ^{Leb} can be alleviated at a cost of a more evolved proof. The central idea would be to use the Lebesgue decomposition theorem to reduce the coordination problem to each atom and thus establish the convergence of ABAP in such atoms.

[19] We keep the discussion in terms of CAQL, but the same can be carried out using exactly the same arguments to just ABAP.

7.5 Rationality and convergence

Bowling and Veloso [8] introduce two properties that one would expect from any "intelligent" learning agent, namely *rationality* and *convergence*. An agent is *rational* if it is able to converge to an individual policy that is a best response to the policies of the other agents, as long as the other agents all converge to stationarity. This is important, for example, in situations where the other agents may have *limitations*. A rational agent should be able to take these limitations into consideration when choosing its own individual policy.

On the other hand, an agent is *convergent in self-play* if it converges to a stationary policy against other learning agents using the same learning algorithm. In other words, a convergent agent should be able to converge even if the other players in the game also adapt their policies.

By construction, CAQL is convergent in self-play. Furthermore, because of the simultaneous learning of Q, if the other agents all follow (or eventually reach) some stationary policy, this will reflect in the estimate of the Q-values associated with each action, and the best-response set defined in (8) will only include the actions that are actually best responses to the stationary policy of the other agents. This means that CAQL is also rational.

Nevertheless, an interesting question that is worth exploring in future work is concerned with the convergence of CAQL against other classes of learners. We have no theoretical results on this question at this point, but we dare to conjecture the following:

- Without any assumption on the particular coordination mechanism followed by the other agents, we believe that it may not be possible to establish convergence of ABAP in the presence of general learning agents. The rationale behind this conjecture is that it should be possible to design an "adversarial" mechanism that always chooses the "worst" action for ABAP. Even if such a mechanism would be senseless and of no practical usefulness, it would still provide a counter-example against which ABAP might not converge.
- To the extent of our knowledge, joint action learners (JAL) as proposed in [11] have only been applied in MMDPs with finite state-spaces. However, we note that JAL can be obtained from BAP by taking $m = K = t$ at each iteration t. Therefore, it should be possible to adapt JAL to infinite settings by again considering ABAP and taking, at each iteration t, $m = K = t$. In the presence of agents following this modified version of JAL, and since the underlying coordination mechanism in both methods is similar, we still expect CAQL to converge.

Another very interesting question that is also worth exploring is concerned with the application of CAQL/ABAP in cooperative settings in which not all agents share the same reward function. This is the case, for example, in scenarios with a similar reward structure as the well-known game of Battle-of-Sexes (BoS). Such scenarios are still "cooperative", in the sense that they require the agents to coordinate in one of the possible equilibria, but different agents prefer different equilibria.

One fundamental difference between the game of Battle-of-Sexes and the class of games considered in this paper is that there is no *coordination equilibrium* in the sense of Littman [24]. In the Battle-of-Sexes, although both equilibria are Pareto optimal, none of the two actually gives the maximum payoff to *both* agents. We note, however, that the best response graph for the game is still weakly acyclic, which means that BAP would still converge.

In this paper, we addressed learning and coordination/equilibrium selection in scenarios in which:

1. All agents share the reward function;
2. All agents have full access to the state of the game;

3. All agents have full access to the actions played by the other agents.

However, there are many important real-world problems in which one or more such assumptions do not hold. None of these two problems (learning and equilibrium selection) is trivial to deal with in real-world multiagent scenarios. As for 1, even in finite settings and focusing only on the learning aspect of the problem (i.e., ignoring the equilibrium selection aspect of the problem), few algorithms can actually tackle general sum games with any guarantees of convergence, and typically require somewhat restrictive conditions on the game. Nash-Q and FF-Q are among the few learning algorithms that address this very broad class of games with any formal guarantees of convergence, and there is certainly room for further work on this topic.

As for 2 and 3, there is a great body of work on problems with partial observability, and many models of different complexity have been proposed to address such problems (e.g., Dec-MDPs, Dec-POMDPs [1], I-POMDPs [16]). However, in such partially observable settings, the problem of decentralized decision-making is known to be NEXP-complete even in fully-cooperative two-agent games with finite state-spaces [1]. In other words, even if the model of the multiagent system is completely known and we focus on the problem of planning taking into account that, at execution-time, the agents have only local knowledge, the inherent complexity of the problem is prohibitive.

Nevertheless, the fact that CAQL accommodates for the use of function approximation already implies that it should be possible to extend our results to scenarios with a simplified form of partial observability—namely, centralized partial observability. In a recent work, we also started to explore learning in cooperative settings where each agent is allowed to have its own payoff function [29]. However, the extension of the results in this paper both to general-sum and to partially observable settings remains a very interesting avenue for future work.

Acknowledgments The authors would like to acknowledge the helpful discussions with Manuela Veloso, who pointed out that the use of observed actions at nearby points might be useful to extend BAP to infinite state-spaces. We also acknowledge the anonymous reviewers for their many comments and suggestions that contributed to improve the overall clarity of the paper, as well as for several interesting issues that contributed to enrich the discussions in the paper. This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia under the Carnegie Mellon-Portugal Program and the Information and Communications Technologies Institute (ICTI) (www.icti.cmu.edu) and also under Programa Operacional Sociedade do Conhecimento (POS_C) that includes FEDER funds. The views and conclusions contained in this document are those of the authors only. The first author acknowledges the PhD grant SFRH/BD/3074/2000.

Appendix

A Markov Chains

In this Appendix, we review some background material on Markov chains. In particular, we introduce several concepts that describe different types of stable behavior in this class of processes. In particular, we discuss

- The concept of ψ-*irreducibility*, that roughly translates the ability of a chain to reach all parts of the state-space;
- The concept of recurrence, that roughly translates the ability of a chain to revisit infinitely often all "relevant" parts of the state-space;
- The concept of positivity, that roughly translates the notion of equilibrium behavior in this class of processes;

– The concept of ergodicity, that roughly translates the notion of convergent behavior in this class of processes.

We start by defining a *homogeneous Markov chain* as a discrete-time stochastic process $\{X(t)\}$, where each r.v. $X(t)$, $t \geq 0$, takes values in $\mathcal{X}$. The r.v. $X(0)$ is distributed according to an *initial measurede* μ_0, i.e.,

$$\mathbb{P}[X(0) \in U] = \mu_0(U),$$

where $U \subset \mathcal{X}$ is any measurable set. The distribution of each r.v. $X(t)$, $t > 0$, depends on the value of $X(t-1)$ according to the transition probabilities

$$\mathbb{P}[X(t) \in U \mid X(t-1) = x] = \mathsf{P}(x, U), \tag{17}$$

where $U \subset \mathcal{X}$ is any measurable set. We refer to the kernel P as the *transition kernel* for the chain. We henceforth refer to a Markov chain either as a sequence $\{X(t)\}$ or as a pair $(\mathcal{X}, \mathsf{P})$ with the implicit assumption that some initial measure is understood. We also define the T-step transition kernel as

$$\mathsf{P}^T(x, U) \triangleq \mathbb{P}[X(t+T) \in U \mid X(t) = x]$$

and represents the probability of the chain entering the set U at time T, given that it departs from state x. The T-step transition kernel can be defined recursively from the transition kernel P as

$$\mathsf{P}^1(x, U) = \mathsf{P}(x, U) \quad \mathsf{P}^t(x, U) = \int_{\mathcal{X}} \mathsf{P}(y, U)\mathsf{P}^{t-1}(x, \mathrm{d}y).$$

Given an arbitrary measurable set $U \subset \mathcal{X}$, the *first return time to* U, τ_U, is defined as

$$\tau_U = \min\{t > 0 \mid X(t) \in U\},$$

and is a r.v. that corresponds to the first time-step in which the chain enters the set U. Note that, by definition, $\tau > 0$. Given a measure φ on $\mathcal{X}$, a Markov chain is *φ-irreducible* if

$$\mathbb{P}[\tau_U < \infty \mid X(0) = x] > 0,$$

for any $x \in \mathcal{X}$ and any measurable set $U \subset \mathcal{X}$ such that $\varphi(U) > 0$. This means that a chain is φ-irreducible if it is possible to define a measure φ that assigns positive measure only to those sets that can be reached from any initial state. If a Markov chain $(\mathcal{X}, \mathsf{P})$ is φ-irreducible, then there is a maximal irreducibility measure ψ for which $(\mathcal{X}, \mathsf{P})$ is ψ-irreducible.[20]

All maximal irreducibility measures are equivalent and hence a chain can be classified as being *ψ-irreducible* without specifying the maximal irreducibility measure ψ.

Let now η_U denote the r.v. corresponding to the number of visits to a measurable set $U \subset \mathcal{X}$ in an infinite trajectory of the chain. Then, a set U is said *Harris recurrent* if, for any $x \in \mathcal{X}$,

$$\mathbb{P}[\eta_U = \infty \mid X(0) = x] = 1.$$

Therefore, a Harris recurrent set is a set that is visited infinitely often from any initial condition. A ψ-irreducible Markov chain is *Harris recurrent* if any measurable set $U \subset \mathcal{X}$ such that $\psi(U) > 0$ is Harris recurrent. Therefore, in a Harris recurrent chain, any "parts" of the state-space that can be visited from every initial condition will be visited an infinite number of times.

[20] In this context, ψ is a maximal if $\psi \gg \varphi$ for any irreducibility measure φ.

Previous works that establish similar results to those portrayed in this paper but in finite state-space settings (see, for example, [27,53]) typically require that every state $x \in \mathcal{X}$ be visited infinitely often. In the more general state-space setting considered herein, that requirement translates in the above notions of ψ-irreducibility and Harris recurrence.

Given a Markov chain $(\mathcal{X}, \mathsf{P})$, a probability measure μ is called *invariant* if

$$\int_{\mathcal{X}} \mathsf{P}(x, U)\mu(\mathrm{d}x) = \mu(U). \tag{18}$$

for any measurable set $U \subset \mathcal{X}$. When it exists, the invariant probability measure for a chain is equivalent to the maximal ψ-irreducibility measure for that chain. The invariant probability measure for a chain describes, in a sense, its "steady-state" behavior. A ψ-irreducible Markov chain admitting an invariant probability measure is called a *positive chain*.

A Markov chain $(\mathcal{X}, \mathsf{P})$ is *geometrically ergodic* if there is a constant $r > 1$ such that, for any initial measure μ_0 on $\mathcal{X}$,

$$\sum_{t=0}^{\infty} r^t \left\| \mu_0 \mathsf{P}^t - \mu^* \right\| < \infty,$$

where $\left\| \mu_0 \mathsf{P}^t - \mu^* \right\|$ is the total variation distance between $\mu_0 \mathsf{P}^t$ and μ^*.[21] Intuitively, a Markov chain is geometrically ergodic if it converges exponentially fast to its invariant distribution. It is important to refer that geometric ergodicity is a stronger condition than ψ-irreducibility and Harris recurrence, and usually implies the latter two.

Finally, given a Markov chain $(\mathcal{X}, \mathsf{P})$ and a measurable function $f : \mathcal{X} \to \mathbb{R}$, the *Poisson equation* for the chain w.r.t. f is

$$v(x) - (\mathsf{P}v)(x) = f(x) - \mu f, \tag{19}$$

where we used the following compact notation

$$(\mathsf{P}v)(x) \triangleq \mathbb{E}\left[v(X(t+1)) \mid X(t) = x\right] = \int_{\mathcal{X}} v(y)\mathsf{P}(x, \mathrm{d}y);$$

$$\mu f \triangleq \mathbb{E}_{\mu}\left[f(x)\right] = \int_{\mathcal{X}} f(x)\mu(\mathrm{d}x).$$

If the chain $(\mathcal{X}, \mathsf{P})$ is positive Harris, the solution v to (19) is well defined for all $x \in \mathcal{X}$ and verifies

$$\mathrm{v}(x) = \sum_{t=0}^{\infty} \left[(\mathsf{P}^t f)(x) - \mu f\right]. \tag{20}$$

Roughly speaking, the Poisson equation quantifies the expected error incurred when approximating μf by its sample mean computed along a trajectory of the chain.

[21] Recall that the total variation distance between two probability measures μ and ν on a set $\mathcal{X}$ is defined as

$$\|\mu - \nu\| = \sup_{U} |\mu(U) - \nu(U)|,$$

with the supremum taken over all measurable subsets U of $\mathcal{X}$.

B Proofs

In this Appendix we present the proofs of the several theorems along the text. We resort to several classical results from stochastic approximation theory, introduced as needed. Given a general vector $x \in \mathbb{R}^p$, we denote its ith component interchangeably by $x(i)$ or x_i. Also, for a vector-valued function $f : \mathbb{R}^p \to \mathbb{R}^q$, we denote its ith component at x by $f_i(x)$. Finally, for a $p \times q$ matrix $\mathbf{A}$, we denote its (i, j) component interchangeably by $\mathbf{A}(i, j)$ or $\mathbf{A}_{i,j}$ and its jth column by either $\mathbf{A}(j)$ or $\mathbf{A}_j$. The (i, j) component of a matrix-valued function f at x is denoted by $f_{i,j}(x)$ and its jth column by $f_j(x)$. In the remainder of this section, $\|\cdot\|$ denotes the max-norm.

B.1 Proof of Theorem 2

To establish the convergence of Q-SSA we make use the following result from [2, Proposition 4.5].

Theorem 7 *Consider the general algorithm*

$$\theta_{t+1}(a) = (1 - \alpha_t(a))\theta_t(a) + \alpha_t(a)\left[h_a(\theta_t) + M_a(t+1) + \varepsilon_a(t)\right], \tag{21}$$

such that the following hold w.p.1:

1. *The step-size sequence $\{\alpha_t\}$ verifies, for all $a \in \mathcal{A}$*

$$\sum_t \alpha_t(a) = \infty \quad \sum_t \alpha_t^2(a) < \infty;$$

2. *The sequence $\{M(t)\}$ verifies*

$$\mathbb{E}\left[M(t+1) \mid \mathcal{F}_t\right] = 0 \quad \mathbb{E}\left[\|M(t+1)\|^2 \mid \mathcal{F}_t\right] \leq K\left(1 + \|\theta_t\|^2\right),$$

   where $\mathcal{F}_t$ is the σ-algebra generated by $\{(M(\tau), \varepsilon(\tau)), \tau = 1, \ldots, t\}$.
3. *The sequence $\{\varepsilon(t)\}$ is asymptotically negligible, i.e., there is a sequence $\{c_t\}$ going to 0 such that*

$$\|\varepsilon(t)\| \leq c_t(1 + \|\theta_t\|).$$

4. *The map $h : \mathbb{R}^M \to \mathbb{R}^M$ is a contraction, i.e.,*

$$\left\|h(\theta) - h(\theta')\right\| \leq \gamma \left\|\theta - \theta'\right\|$$

   with $0 \leq \gamma < 1$.

Then, for any θ_0, the sequence $\{\theta_t\}$ recursively defined by (21) converges w.p.1 to the fixed point of h.

To establish the convergence of Q-SSA, we verify that each of the conditions in the above result hold.

We start by writing (9) as (21). To that purpose, given the learning process $\{X(t)\}$, let us introduce the process $\{Z(t)\}$ defined as $Z(t) = (X(t-1), X(t))$. Then, for $z = (x, y)$, let the ath column of the matrix-valued function H be defined as

$$H_a(\theta, z) = \phi(x)\left(r(x, a, y) + \gamma \max_{b \in \mathcal{A}} Q_\theta(y, b)\right).$$

We now write (9) as

$$\theta_{t+1} = \theta_t(\mathbf{1} - \boldsymbol{\alpha}_t) + (h(\theta_t) + \delta(t+1))\,\boldsymbol{\alpha}_t, \tag{22}$$

with the ath columns of h given by

$$h_a(\theta_t) = \mathbb{E}_{\mu_\pi}\left[\phi(x)\left(r(x,a,y) + \gamma \max_{b\in\mathcal{A}} Q_{\theta_t}(y,b)\right)\right]$$

and

$$\delta_a(t+1) = H_a(\theta_t, Z(t+1)) - h_a(\theta_t).$$

In (22), we denoted by $\boldsymbol{\alpha}_t$ the diagonal matrix with component (a,a) given by $\alpha_t(a)$, with the implicit understanding that (22) denotes not a single equation but $|\mathcal{A}|$ distinct equations, one for each $a \in \mathcal{A}$. We will stick to this notation in the remainder of the proof.

The expectation in the definition of h is well-defined as a consequence of the assumption of geometric ergodicity of $(\mathcal{X}, \mathsf{P}_\pi)$. Going one step further, we let v denote the solution of the Poisson equation

$$v(\theta,z) - (\mathsf{P}_\pi v)(\theta,z) = H(\theta,z) - h(\theta).$$

We now use the Poisson equation above to further decompose $\delta(t)$, which will simplify the verification of the conditions of Theorem 7. Some explicit computations yield

$$\begin{aligned}
\delta(t+1) &= H(\theta_t, Z(t+1)) - h(\theta_t)\\
&= v(\theta_t, Z(t+1)) - (\mathsf{P}_\pi v)(\theta_t, Z(t+1))\\
&= v(\theta_t, Z(t+1)) - (\mathsf{P}_\pi v)(\theta_t, Z(t))\\
&\quad + (\mathsf{P}_\pi v)(\theta_t, Z(t))\boldsymbol{\alpha}_{t-1}\boldsymbol{\alpha}_t^{-1} - (\mathsf{P}_\pi v)(\theta_{t+1}, Z(t+1))\\
&\quad + (\mathsf{P}_\pi v)(\theta_{t+1}, Z(t+1)) - (\mathsf{P}_\pi v)(\theta_t, Z(t+1))\\
&\quad + (\mathsf{P}_\pi v)(\theta_t, Z(t))\left(\mathbf{1} - \boldsymbol{\alpha}_{t-1}\boldsymbol{\alpha}_t^{-1}\right).
\end{aligned}$$

Letting

$$\begin{aligned}
M(t+1) &= v(\theta_t, Z(t+1)) - (\mathsf{P}_\pi v)(\theta_t, Z(t)),\\
\eta_1(t+1) &= (\mathsf{P}_\pi v)(\theta_t, Z(t))\boldsymbol{\alpha}_{t-1}\boldsymbol{\alpha}_t^{-1} - (\mathsf{P}_\pi v)(\theta_{t+1}, Z(t+1)),\\
\eta_2(t+1) &= (\mathsf{P}_\pi v)(\theta_{t+1}, Z(t+1)) - (\mathsf{P}_\pi v)(\theta_t, Z(t+1)),\\
\eta_3(t+1) &= (\mathsf{P}_\pi v)(\theta_t, Z(t))\left(\mathbf{1} - \boldsymbol{\alpha}_{t-1}\boldsymbol{\alpha}_t^{-1}\right),
\end{aligned}$$

we get

$$\varepsilon(t+1) = \eta_1(t+1) + \eta_2(t+1) + \eta_3(t+1)$$

and (22) finally becomes

$$\theta_{t+1}(a) = (1-\alpha_t(a))\theta_t(a) + \alpha_t(a)\left(h_a(\theta_t) + M_a(t+1) + \varepsilon_a(t+1)\right).$$

We are now in position to verify the conditions of Theorem 7. Condition 1 holds by assumption. Concerning condition 2, it can be shown that $\{M(t)\}$ is a convergent martingale sequence (see, for example, [13]). In particular, it is immediate that $\mathbb{E}_{\mu_\pi}[M(t+1) \mid \mathcal{F}_t] = 0$. On the other hand, we notice that

$$\|H(\theta,z)\| \le K_0(1+\|\theta\|) \quad \|H(\theta_1,z) - H(\theta_2,z)\| \le K_1\|\theta_1 - \theta_2\| \tag{23}$$

for some positive constants K_0 and K_1. Combining the above facts with (20), we get

$$\|v(\theta, z)\| \leq K_0'(1 + \|\theta\|) \quad \|v(\theta_1, z) - v(\theta_2, z)\| \leq K_1' \|\theta_1 - \theta_2\| \tag{24}$$

for some positive constants K_0' and K_1' (for further details on the properties of v, see Chap. 17 of [31]). This implies that

$$\mathbb{E}\left[\|M(t+1)\|^2 \mid \mathcal{F}_t\right] \leq K_M\left(1 + \|\theta_t\|^2\right)$$

for some positive constant K_M.

Proceeding to condition 3 and using (23) and (24) we have

$$\begin{aligned}\|\eta_2(t+1)\| &\leq K\|\theta_{t+1} - \theta_t\| \leq K'\|\alpha_t\|(1 + \|\theta_t\|)\\ \|\eta_3(t+1)\| &\leq K''\left\|\mathbf{1} - \boldsymbol{\alpha}_{t-1}\boldsymbol{\alpha}_t^{-1}\right\|(1 + \|\theta_t\|),\end{aligned}$$

for some positive constants K, K' and K''. Furthermore, taking $\boldsymbol{\alpha}_{-1} = \mathbf{0}$, we have

$$\sum_{t=0}^{T} \eta_1(t+1)\boldsymbol{\alpha}_t = -(\mathsf{P}_\pi v)(\theta_{T+1}, z(T+1))\boldsymbol{\alpha}_T$$

and hence

$$\lim_{T\to\infty}\left\|\sum_{t=0}^{T}\eta_1(t+1)\boldsymbol{\alpha}_t\right\| = 0.$$

Since, by assumption, $\sum_t \alpha_t(a) = \infty$ for all $a \in \mathcal{A}$, we have that $\|\eta_1(t)\| \to 0$.

Finally, it remains to show that $h(\theta)$ is a contraction. To see this, and denoting by $h_{i,a}$ the (i, a) component of h,

$$\begin{aligned}\left|h_{i,a}(\theta_1) - h_{i,a}(\theta_2)\right| &= \left|\mathbb{E}_{\mu_\pi}\left[\phi_i(x)\left(\gamma\max_{b\in\mathcal{A}} Q_{\theta_1}(y,b) - \gamma\max_{b\in\mathcal{A}} Q_{\theta_2}(y,b)\right)\right]\right|\\ &\leq \gamma\mathbb{E}_{\mu_\pi}\left[|\phi_i(x)|\left|\max_{b\in\mathcal{A}} Q_{\theta_1}(y,b) - \max_{b\in\mathcal{A}} Q_{\theta_2}(y,b)\right|\right]\\ &\leq \gamma\mathbb{E}_{\mu_\pi}\left[\max_{b\in\mathcal{A}}\left|Q_{\theta_1}(y,b) - Q_{\theta_2}(y,b)\right|\right]\\ &= \gamma\mathbb{E}_{\mu_\pi}\left[\max_{b\in\mathcal{A}}\left|\phi^\top(y)\left(\theta_1(b) - \theta_2(b)\right)\right|\right]\\ &\leq \gamma\mathbb{E}_{\mu_\pi}\left[\max_{b\in\mathcal{A}}\left\|\phi^\top\right\|_1\|\theta_1(b) - \theta_2(b)\|_\infty\right]\\ &= \gamma\|\theta_1 - \theta_2\|.\end{aligned}$$

This establishes the first statement of Theorem 2. To establish the second statement, notice that the fixed point θ^* of h verifies

$$\begin{aligned}\theta^*_{i,a} &= h_{i,a}(\theta^*)\\ &= \mathbb{E}_{\mu_\pi}\left[\phi_i(x)\left(r(x,a,y) + \gamma\max_{b\in\mathcal{A}} Q_{\theta^*}(y,b)\right)\right]\\ &= \mathbb{E}_{\mu_\pi}\left[\phi_i(x)(\mathbf{H}Q_{\theta^*})(x,a)\right].\end{aligned}$$

This concludes the proof of Theorem 2. □

B.2 Proof of Theorem 3

In establishing the statement of the theorem, we consider a "synchronous" version of Q-SSA in which all a-components are updated simultaneously. This greatly simplifies the presentation, allowing us to use standard results from the stochastic approximation literature. The result for this synchronous version can, nevertheless, be extended to the general case. In fact, from Theorem 2, we know that $\theta_t \to \theta^*$ w.p.1 and, therefore, the rate of convergence of the algorithm can be studied component-wise. Considering the asymptotic rate of convergence of the *slowest* component of θ_t we obtain a lower bound for the rate of convergence of *all* components.

We resort to the following theorem from [34]. We denote by $\mathbb{I}_U$ the indicator function for U, defined as

$$\mathbb{I}_U(x) = \begin{cases} 1 & \text{if } x \in U \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 8 *Consider the general algorithm*

$$\theta_{t+1} = \theta_t + \alpha_t \left(\bar{h}(\theta_t) + M(t+1) + \varepsilon(t+1)\right),$$

such that the following hold for $a > 2$ and $b > 1$:

1. *The step-size sequence $\{\alpha_t\}$ is given by*

$$\alpha_t = \frac{1}{t}.$$

2. *There is a neighborhood U of θ^* in which*

$$\mathbb{E}\left[M(t+1) \mid \mathcal{F}_t\right] \mathbb{I}_U(\theta_t) = 0, \tag{25a}$$

$$\sup_t \mathbb{E}\left[\|M(t+1)\|^a \mid \mathcal{F}_t\right] \mathbb{I}_U(\theta_t) < \infty, \tag{25b}$$

$$\|\varepsilon(t+1)\| \le \mathrm{O}\left(\left\|\theta_t - \theta^*\right\|\right) + r(t+1), \tag{25c}$$

with $\|r(t+1)\| \mathbb{I}_U(\theta_t) \in \mathrm{O}(\sqrt{\alpha_t})$. Furthermore, there is a matrix $\mathbf{C}$ such that

$$\lim_{t\to\infty} \mathbb{E}\left[M(t+1)M^\top(t+1) \mid \mathcal{F}_t\right] = \mathbf{C}. \tag{26}$$

3. *There is a neighborhood U of θ^* such that*

$$\bar{h}(\theta) = \mathbf{A}(\theta - \theta^*) + \mathrm{O}\left(\left\|\theta - \theta^*\right\|^b\right),$$

where $\mathbf{A}$ is a stable matrix.

Then, denoting $S_t = \sum_{\tau=0}^{t} \alpha_\tau$, we have w.p.1

$$\limsup \frac{\|\theta - \theta^*\|}{\sqrt{\alpha_t \log(S_t)}} \le K \tag{27}$$

for some constant $K > 0$.

To prove the statement in Theorem 3, we start by writing

$$\bar{h}(\theta) = h(\theta) - \theta,$$

where h is the mapping defined in the proof of Theorem 2. Then, Theorem 3 holds as long as the conditions from Theorem 8 above hold. Since we are considering the synchronous version of Q-SSA, this implies that condition 1 is automatically verified.

Concerning condition 2, we note that the different conditions in (25) and (26) basically impose bounds on $\{M(t)\}$ in a close vicinity of the limit point θ^* and restrict the convergence behavior of both $\{M(t)\}$ and $\{\varepsilon(t)\}$ around that same point. We excuse from repeating a detailed derivation here and just point out that these easily follow from the bounds for $\{M(t)\}$ and $\{\varepsilon(t)\}$ derived in the proof of Theorem 2. In particular, (25a, 25b), and (26) follow trivially from Condition 2 in Theorem 7 and from the fact that $\{M(t)\}$ is a convergent martingale [13]. On the other hand, (25c) trivially follows from the proof of Theorem 2 where Condition 3 is verified.

Finally, consider the ordinary differential equation (ODE)

$$\dot{\theta}_t = \bar{h}(\theta_t) = h(\theta_t) - \theta_t.$$

Notice that this ODE has as unique equilibrium point θ^*. To establish condition 3, we study the stability properties of θ^* as an equilibrium point of the above ODE. Let $\tilde{\theta}_t = \theta_t - \theta^*$. We have, for $p \geq 1$,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left\|\tilde{\theta}_t\right\|_p &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{i,a}\tilde{\theta}_t^p(i,a)\right)^{1/p} \\
&= \frac{1}{p}\left(\sum_{i,a}\tilde{\theta}_t^p(i,a)\right)^{1/p-1}\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i,a}\tilde{\theta}_t^p(i,a) \\
&= \left\|\tilde{\theta}_t\right\|_p^{1-p}\sum_{i,a}\tilde{\theta}_t^{p-1}(i,a)\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\theta}_t(i,a) \\
&= \left\|\tilde{\theta}_t\right\|_p^{1-p}\sum_{i,a}\tilde{\theta}_t^{p-1}(i,a)\left(h_{i,a}(\theta_t) - \theta_t(i,a)\right).
\end{aligned}$$

Using the fact that $\bar{h}(\theta^*) = 0$,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left\|\tilde{\theta}_t\right\|_p &= \left\|\tilde{\theta}_t\right\|_p^{1-p}\sum_{i,a}\tilde{\theta}_t^{p-1}(i,a)\left(h_{i,a}(\theta_t) - h_{i,a}(\theta^*) - \theta_t(i,a) + \theta^*(i,a)\right) \\
&= \left\|\tilde{\theta}_t\right\|_p^{1-p}\sum_{i,a}\tilde{\theta}_t^{p-1}(i,a)\left(h_{i,a}(\theta_t) - h_{i,a}(\theta^*)\right) - \left\|\tilde{\theta}_t\right\|_p.
\end{aligned}$$

Using Hölder's inequality in the summation yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\|\tilde{\theta}_t\right\|_p = \left\|h(\theta_t) - h(\theta^*)\right\|_p - \left\|\tilde{\theta}_t\right\|_p.$$

Taking the limit as $p \to \infty$ and using the fact that h is a contraction, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\|\tilde{\theta}_t\right\| = (\gamma - 1)\left\|\tilde{\theta}_t\right\| < 0,$$

which implies that θ^* is a globally asymptotically stable equilibrium point. This, in turn, implies that $\bar{h}$ can be linearized around θ^* yielding condition 3. This means that the synchronous version of the algorithm verifies the bound in (27).

Specializing this bound to the "slowest component" of θ_t, we finally get that

$$\limsup \frac{\|\theta - \theta^*\|}{\sqrt{\max_a \alpha_t(a) \log\left(\sum_{\tau=0}^{t} \alpha_\tau(a)\right)}} \leq K$$

and the proof is complete. □

B.3 Proof of Theorem 5

In this proof we show that the set of states in which ABAP may not converge is contained in a set with null Lebesgue measure. We require several intermediate results that build up to the final statement of the theorem.

Let us first assume Q^* to be continuous. This means that the function $V^a(x) = Q^*(x, a)$ is continuous for each $a \in \mathcal{A}$. Take an arbitrary point $x \in \mathcal{X}$ and an arbitrary action $a_0 \in \mathcal{A}$. Then, one of two statements below holds:

1. $Q^*(x, a_0) < \max_{a \in \mathcal{A}} Q^*(x, a)$. If this is the case, due to the continuity of Q^* in x, the inequality above holds for some neighborhood U of x. In other words, there is a neighborhood U of x such that $Q^*(y, a_0) < \max_{a \in \mathcal{A}} Q^*(y, a)$, for all $y \in U$. This has an interesting implication: for every point $x \in \mathcal{X}$ there is a neighborhood U such that

$$\mathbf{opt}(y) \subset \mathbf{opt}(x), \tag{28}$$

   for all $y \in U$, where $\mathbf{opt}(x)$ is the set of optimal joint actions at state x.
2. $Q^*(x, a_0) = \max_{a \in \mathcal{A}} Q^*(x, a)$. If this is the case, two possible situations can occur:
   (a) There is a neighborhood U of x such that $a_0 \in \mathbf{opt}(y)$ for all $y \in U$;
   (b) Given any neighborhood U of x there is a point $y \in U$ such that $a_0 \notin \mathbf{opt}(y)$;

Denote by $D(a_0)$ the set of points $x \in \mathcal{X}$ verifying 2b and define the sets $D = \bigcup_{a \in \mathcal{A}} D(a)$ and $C = \mathcal{X} - D$. Before establishing several important properties of these two sets, C and D, we note that all states $x \in C$ have a neighborhood in which all states have the same optimal actions. Therefore, in terms of coordination, each such neighborhood can be treated as a single state, and this is indeed the idea behind ABAP. Therefore, to establish the result in the theorem, it remains essentially to show that D has zero Lebesgue measure.

We now show that

Lemma 1 *The set C is an open set and $D = \partial C$.*

Proof From 1 and 2a, we see that a point $x \in C$ has a neighborhood U such that $U \cap D = \emptyset$. This means that $C = \mathbf{int}(C)$ and thus C is open. On the other hand, since $D = \mathcal{X} - C$, $\partial C \subset D$. Since C and D are complementary and $C = \mathbf{int}(C)$, the second statement of the lemma follows. □

Since $D = \partial C$, it is immediate that D is closed and therefore measurable. In turn, C is open and thus also measurable. We proceed with the following result.

Lemma 2 *The set D defined above verifies $\mu^{\text{Leb}}(D) = 0$.*

Proof Recall that the function Q^* is continuous in x. Therefore, the function

$$V^*(x) = \max_{a \in \mathcal{A}} Q^*(x, a)$$

is also continuous. We define a new function G^a as $G^a(x) = V^a(x) - V^*(x)$. Clearly, G^a is continuous and $G^a(x) \leq 0$ for all $x \in \mathcal{X}$. We now show that $\Omega_{G^a} = \{x \in \mathcal{X} \mid G^a(x) < 0\}$ is an open set. Clearly, any point $x \in \Omega_{G^a}$ has a neighborhood $U \subset \Omega_{G^a}$, due to the continuity of G^a. This means that any point in Ω_{G^a} is an interior point and the set is, therefore, open. Since $\mathcal{X}$ is compact, Ω_{G^a} is a bounded open set and its boundary has null Lebesgue measure.

But then, by construction, we have that

$$\partial C \subset \bigcup_{a \in \mathcal{A}} \partial \Omega_{G^a},$$

and the conclusion follows. □

As discussed above, for each state $x \in C$ there is a neighborhood U such that $\mathbf{opt}(x) = \mathbf{opt}(y)$, for all $y \in U$. This can be seen by noticing that a point in C either verifies Condition 1 or Condition 2a above, for every action $a \in \mathcal{A}$. In other words, the set of optimal actions is "constant" in U and past plays at any states in U can be used for coordination. On the other hand, the set of optimal actions in any neighborhood of a state in D is "not constant" in the above sense. The above result ensures, however, that this set is "negligible" (in terms of Lebesgue measure).

Now given any state $x \in C$ and a corresponding neighborhood U, it is immediate that the virtual game obtained by setting to 1 all optimal actions and to 0 all non-optimal actions is the same in every point in U. This implies that, if $\psi(U) > 0$, there is a time T_0 such that, w.p.1, $S_m(x, H_t) \subset U$ for $t > T_0$ and ABAP reduces to BAP around x. Since, for all $t > T_0$ all K-samples are drawn from $S_m(x, H_t)$, convergence of standard BAP ensures that, for all points in C, ABAP coordinates in an optimal Nash equilibrium w.p.1. And, since ψ is absolutely continuous w.r.t. μ^{Leb}, Lemma 2 suffices to conclude that convergence to an optimal policy in all but a ψ-null set of points.

Several important observations are in order. First of all, coordination in a given state $x \in \mathcal{X}$ relies on past plays at "nearby" states. We could extend this concept of "nearby states" by considering a general similarity criterion between states in $\mathcal{X}$. Coordination in a state $x \in \mathcal{X}$ would now rely on past plays at similar states.

Secondly, we notice that a notion of "distance" in $\mathcal{X}$ naturally induces a topology on $\mathcal{X}$, from which concepts such as *open set, neighborhood* or *boundary* arise. When using a general similarity criterion between states, the topology for $\mathcal{X}$ must be built from this criterion, and all the above derivations hold.

Finally, a fundamental aspect of the proof above is that the set D has null ψ-measure. This fact, arising from Lemma 2 and the assumption of absolute continuity of ψ w.r.t. μ^{Leb}, binds the dynamic behavior of the chain (encapsulated in the ψ-measure) and the geometry induced by the similarity criterion (in our case, encapsulated in the Lebesgue measure). Using a general similarity criterion will induce some general topology on $\mathcal{X}$. In that case, the condition of absolute continuity of ψ w.r.t. μ^{Leb} should be modified to account for this fact. In particular, convergence of ABAP will require some condition ensuring that ψ is "well-adapted" to the geometry induced by the similarity criterion.

To conclude the proof of Theorem 5, suppose now that Q^* is continuous in all but a ψ-null set of points. Then, the previous proof holds for every point x in which Q^* is continuous, and the proof is complete. □

B.4 Proof of Theorem 6

As in Sect. 4, let $\Gamma^*_{x(t)}$ denote the virtual game built from $Q_{\theta^*}(x(t), \cdot)$. To establish our result we first show that the rate at which $\varepsilon_t \to 0$ guarantees that $\widehat{VG}^{\varepsilon}_t \to \Gamma^*_{x(t)}$. This is established in the following result.

Lemma 3 *Let $\Gamma = (N, \mathcal{X}, (\mathcal{A}_k), \mathsf{P}, r, \gamma)$ be a team Markov game and Λ_T the event that, for $t > T$, $\widehat{VG}^{\varepsilon}_t = \Gamma^*_{x(t)}$ for an agent following Q-SSA. If $B(t)$ decreases monotonically to zero and*

$$\lim_{t\to\infty} \frac{\mathcal{E}_t}{B(t)} = 0,$$

then $\lim_{T\to\infty} \mathbb{P}[\Lambda_T] = 1$.

Proof The proof closely follows the proof of Lemma 6 in [53].

Let $x \in \mathcal{X}$ be some fixed state and let λ_T be the event that, for all $t > T$,

$$\max_{a\in\mathcal{A}} \left| Q_{\theta_t}(x, a) - Q_{\theta^*}(x, a) \right| < \frac{K_0}{2} B(t).$$

Since, by assumption,

$$\lim_{t\to\infty} \frac{\mathcal{E}_t}{B(t)} = 0,$$

it holds that

$$\lim_{t\to\infty} \frac{K_1 \mathcal{E}_t}{K_0 B(t)} = 0,$$

for any positive constant K_1. Since, from Theorem 3,

$$\left\| Q(\theta_t) - Q(\theta^*) \right\| \leq \left\| \theta_t - \theta^* \right\| \leq K_0 \mathcal{E}_t$$

w.p.1, then given any $\rho > 0$ there is a time instant $T_0(\rho) > 0$ such that, for all $t > T_0$,

$$\mathbb{P}[\lambda_t] > 1 - \rho. \tag{29}$$

Take now two actions $a, b \in \mathcal{A}$ such that $a \in \mathbf{opt}(x)$ and b verifies

$$b = \arg\max_{u \notin \mathbf{opt}(x)} Q_{\theta^*}(x, u).$$

Define the quantity $\delta = |Q_{\theta^*}(x, a) - Q_{\theta^*}(x, b)|$. By assumption, $B(t) \to 0$ and, therefore, there is a time instant T_1 such that, for all $t > T_1$,

$$K_0 B(t) < \frac{\delta}{2}. \tag{30}$$

Let $T = \max\{T_0, T_1\}$. For all $t > T$ it holds with probability $p > 1 - \rho$ that, given any action $b \notin \mathbf{opt}(x)$,

$$\begin{aligned} Q_{\theta_t}(x, b) + K_0 B(t) &< Q_{\theta^*}(x, b) + K_0 B(t) + \frac{K_0}{2} B(t) \\ &< Q_{\theta^*}(x, b) + \frac{\delta}{2} + \frac{\delta}{4} \\ &\leq \max_{u\in\mathcal{A}} Q_{\theta^*}(x, u) - \frac{\delta}{4} \\ &< \max_{u\in\mathcal{A}} Q_{\theta_t}(x, u). \end{aligned} \tag{31}$$

The first inequality arises from (29); the second inequality arises from (30); the third inequality arises from the definition of δ and the last inequality arises from (29) once again. On the other hand, for all $t > T$ it holds with probability $p > 1-\rho$ that, given any action $a \in \mathbf{opt}(x)$,

$$Q_{\theta_t}(x,a) + K_0 B(t) > Q_{\theta^*}(x,a) + \frac{K_0}{2} B(t) > \max_{u \in \mathcal{A}} Q_{\theta_t} t(x,u). \tag{32}$$

The first inequality arises from (29) and the second inequality from (30).

By construction, we have that $\varepsilon_t \leq \varepsilon_0 B(t)$, and hence $\varepsilon_t \leq K_0 B(t)$ as long as $\varepsilon_0 \leq K_0$. Then, combining (31) and (32), it holds with probability $p > 1-\rho$ that, for all $t > T$,

$$\begin{aligned} Q_{\theta_t}(x,b) &< \max_{u \in \mathcal{A}} Q_{\theta_t}(x,u) - \varepsilon_t \\ Q_{\theta_t}(x,a) &> \max_{u \in \mathcal{A}} Q_{\theta_t}(x,u) - \varepsilon_t, \end{aligned}$$

for any actions $a \in \mathbf{opt}^{\varepsilon_t}(x)$ and $b \notin \mathbf{opt}^{\varepsilon_t}(x)$. The first expression implies that, for any $t > T$, no suboptimal action belongs to $\mathbf{opt}^{\varepsilon_t}(x)$; the second expression implies that all optimal actions do belong to $\mathbf{opt}^{\varepsilon_t}(x)$. This means that, for all $t > T$ such that $x(t) = x$, $\widehat{VG}_t^{\varepsilon} = \Gamma_x^*$ with probability $p > 1-\rho$ and, therefore, $\mathbb{P}[\Lambda_T] > 1-\rho$. The conclusion of the lemma follows. □

Convergence of θ_t to θ^* arises as an immediate consequence of Theorem 4. That same theorem also states that

$$\limsup \frac{\|\theta_t - \theta^*\|}{\mathcal{E}_t} \leq K_0.$$

Therefore, Lemma 3 guarantee that, w.p.1, the sequence of virtual games $\widehat{VG}_t^{\varepsilon}$ obtained from the sequence of estimates $\{Q_{\theta_t}\}$ converges to the virtual games Γ^* obtained from $\{Q_{\theta^*}\}$.

On the other hand, let C be the set defined in the proof of Theorem 5. Since the functions ϕ_i are continuous μ_π almost everywhere, so is Q_{θ^*}. The fact that the chain (X, P_π) is geometrically ergodic implies, in particular, that it is ψ-irreducible and Harris recurrent.

Take some state $x \in C$ and let Λ_T be the event that, for all $t > T$, $\widehat{VG}_t^{\varepsilon} = \Gamma_{x(t)}^*$. From Lemma 3, given any $\rho_1 > 0$ there is $T_1(\rho_1)$ such that

$$\mathbb{P}[\Lambda_t] > 1 - \rho_1$$

for all $t > T_1$. Furthermore, since $x \in C$, it holds that $\mathbf{opt}(y) \subset \mathbf{opt}(x)$ for all $y \in U$ (refining the neighborhood U, if necessary).

Suppose now that Λ_{T_1} occurs for some $T_1 > 0$. From Theorem 5, ABAP converges w.p.1 to an optimal policy in all states in U and, in particular, in x. In other words, if Λ_{T_1} occurs, ABAP converges w.p.1 to an optimal policy in x. Yet to put it differently, there is a time $T_2(\rho_2, T)$ such that, for any $t > T_2$,

$$\mathbb{P}[\lambda_t \mid \Lambda_T] > 1 - \rho_2,$$

where ρ_2 is an arbitrary positive constant and λ_T is the event that, for $t > T$, all agents play an optimal policy if they visit state x. But then there is a time instant $T_3(\rho_1, \rho_2)$ such that, for all $t > T_3$,

$$\mathbb{P}[\lambda_t] > \mathbb{P}[\lambda_t \mid \Lambda_t]\,\mathbb{P}[\Lambda_t] = (1-\rho_1)(1-\rho_2) > 1 - \rho_1 - \rho_2.$$

Since ρ_1 and ρ_2 are arbitrary, the conclusion of the theorem follows. □

References

1. Bernstein, D. S., Zilberstein, S., & Immerman, N. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research, 27*(4), 819–840.
2. Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming optimization and neural computation series*. Belmont, MA: Athena Scientific.
3. Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Proceedings of the 16th international joint conference on artificial intelligence (IJCAI'99)* (pp. 478–485).
4. Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on theoretical aspects of rationality and knowledge (TARK-96)* (pp. 195–210)
5. Bowling, M. (2000). Convergence problems of general-sum multiagent reinforcement learning. In *Proceedings of the 17th international conference on machine learning (ICML'00)* (pp 89–94). Morgan Kaufman.
6. Bowling, M., & Veloso, M. (2000a). *An analysis of stochastic game theory for multiagent reinforcement learning*. Technical Report CMU-CS-00-165, School of Computer Science, Carnegie Mellon University.
7. Bowling, M., & Veloso, M. (2000b). Scalable learning in stochastic games. In *Proceedings of the AAAI workshop on game theoretic and decision theoretic agents (GTDT'02)* (pp. 11–18). The AAAI Press, Published as AAAI Technical Report WS-02-06.
8. Bowling, M., & Veloso, M. (2001). Rational and convergent learning in stochastic games. In *Proceedings of the 17th international joint conference on artificial intelligence (IJCAI'01)* (pp. 1021–1026).
9. Bowling, M., & Veloso, M. (2002). Multi-agent learning using a variable learning rate. *Artificial Intelligence, 136*, 215–250.
10. Brown, G. W. (1949). *Some notes on computation of games solutions. Research Memoranda RM-125-PR*. Santa Monica: RAND Corporation.
11. Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the 15th national conference on artificial intelligence (AAAI'98)* (pp. 746–752).
12. Crites, R. H., & Barto, A. G. (1998). Elevator group control using multiple reinforcement learning agents. *Machine Learning, 33*(2–3), 235–262.
13. Duflo, M. (1997). Random iterartive Models. In *Applications of Mathematics* (Vol. 34). Springer.
14. Durfee, E. H., Lesser, V. R., & Corkill, D. D. (1987). Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers, 36*(11), 1275–1291.
15. Even-Dar, E., & Mansour, Y. (2003). Learning rates for Q-learning. *Journal of Machine Learning Research, 5*, 1–25.
16. Gmytrasiewicz, P., & Doshi, P. (2005). A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research, 24*, 49–79.
17. Gordon, G. J. (1995). *Stable function approximation in dynamic programming*. Technical Report CMU-CS-95-103, School of Computer Science, Carnegie Mellon University.
18. Guestrin, C., Lagoudakis, M. G., & Parr, R. (2002). Coordinated reinforcement learning. In *Proceedings of the 19th international conference on machine learning (ICML'02)* (pp, 227–234).
19. Hu, J., & Wellman, M. P. (2003). Nash Q-learning for general sum stochastic games. *Journal of Machine Learning Research, 4*, 1039–1069.
20. Kearns, M., & Singh, S. (1999). Finite-sample convergence rates for Q-learning and indirect algorithms. In M. J. Kearns, S. A. Solla, & D. A. Cohn, (Eds.), *Advances in neural information processing systems* (Vol. 11, pp. 996–1002). Cambridge, MA: MIT Press.
21. Kok J. R., Spaan, M. T. J., & Vlassis, N. (2002). An approach to noncommunicative multiagent coordination in continuous domains. In: M. Wiering, (Ed.), *Benelearn 2002: Proceedings of the 12th Belgian–Dutch conference on machine learning* (pp. 46–52). Utrecht, The Netherlands.
22. Leslie, D. S., & Collins, E. J. (2006). Generalised weakened fictitious play. *Games and Economic Behavior, 56*, 285–298.
23. Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In R. López de Mántaras, & D. Poole (Eds.), *Proceedings of the 11th international conference on machine learning (ICML'94)* (pp. 157–163). San Francisco, CA: Morgan Kaufmann.
24. Littman, M. L. (2001). Value-function reinforcement learning in Markov games. *Journal of Cognitive Systems Research, 2*(1), 55–66.
25. Littman, M. L. (2001b). Friend-or-foe Q-learning in general-sum games. In *Proceedings of the 18th international conference on machine learning (ICML'01)* (pp. 322–328). San Francisco, CA: Morgan Kaufmann.

26. Melo, F. S., & Ribeiro, M. I. (2007a). *Rational and convergent model-free adaptive learning for team Markov games*. Technical Report RT-601-07, Institute for Systems and Robotics, February.
27. Melo, F. S., & Ribeiro, M. I. (2007b). Learning to coordinate in topological navigation tasks. In *Proceedings of the 6th IFAC symposium on intelligent autonomous vehicles (IAV'07)* (to appear), September.
28. Melo, F. S., & Ribeiro, M. I. (2008). Emerging coordination in infinite team Markov games. In *Proceedings of the 7th international conference on autonomous agents and multiagent systems (AAMAS'08)* (pp. 355–362).
29. Melo, F. S., & Veloso, M. (2009). Learning of coordination: Exploiting sparse interactions in multiagent systems. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems (AAMAS'08)* (pp. 773–780).
30. Melo, F. S., Meyn, S. P., & Ribeiro, M. I. (2008). An analysis of reinforcement learning with function approximation. In *Proceedings of the 25th international conference on machine learning (ICML'08)* (pp. 664–671).
31. Meyn, S. P., & Tweedie, R. L. (1993). *Markov chains and stochastic stability*. Communicatons and Control Engineering Series. New York: Springer.
32. Nash, J. F. (1950). Equilibrium points in *n*-person games. *Proceedings of the National Academy of Sciences, 36*, 48–49.
33. Ormoneit, D., & Sen, Ś. (2002). Kernel-based reinforcement learning. *Machine Learning, 49*, 161–178.
34. Pelletier, M. (1998). On the almost sure asymptotic behaviour of stochastic algorithms. *Stochastic Processes and Their Applications, 78*, 217–244.
35. Perkins, T. J., & Precup, D. (2003). A convergent form of approximate policy iteration. In S. Thrun, S. Becker, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15, pp. 1595–1602). Cambridge, MA: MIT Press.
36. Robinson, J. (1951). An iterative method of solving a game. *Annals of Mathematics, 54*, 296–301.
37. Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development, 3*(3), 210–229. Reprinted in *IBM Journal of Research and Development, 44*(1/2), 206–226, 2000.
38. Samuel, A. L. (1967). Some studies in machine learning using the game of checkers II: Recent progress. *IBM Journal of Research and Development, 11*, 601–617.
39. Sen, S., & Weiß, G. (1999). *Learning in multiagent systems, chapter 6* (pp. 259–298). Cambridge, MA: MIT Press.
40. Singh, S. P., Jaakkola, T., & Jordan, M. I. (1994). Reinforcement learning with soft state aggregation. In *Advances in neural information processing systems* (Vol. 7, pp. 361–368). Cambridge, MA: MIT Press.
41. Singh, S. P., Kearns, M., & Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the 16th conference on uncertainty in artificial intelligence (UAI'00)* (pp. 541–548).
42. Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction. Adaptive computation and machine learning series* (3rd ed.). Cambridge, MA: MIT Press.
43. Szepesvári, C. (1997). The asymptotic convergence rates for *Q*-learning. *Proceedings of Neural Information Processing Systems (NIPS'97), 10*, 1064–1070.
44. Szepesvári, C., & Littman, M. L. (1999). A unified analysis of value-function-based reinforcement learning algorithms. *Neural Computation, 11*(8), 2017–2059.
45. Szepesvári, C., & Smart, W. D. (2004). Interpolation-based *Q*-learning. In *Proceedings of the 21st international conference on machine learning (ICML'04)* (pp. 100–107). New York, USA: ACM Press, July.
46. Tesauro, G. (1994). TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation, 6*(2), 215–219.
47. Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM, 38*(3), 58–68.
48. Tong, H., & Brown, T. X. (2000). Reinforcement learning for call admission control and routing under quality of service constraints in multimedia networks. *Machine Learning, 49*(2–3), 111–139.
49. Tsitsiklis, J. N., & Athans, M. (1985). On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control AC, 30*(5), 440–446.
50. Tsitsiklis, J. N., & Van Roy, B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning, 22*, 59–94.
51. Tsitsiklis, J. N., & Van Roy, B. (1996). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control, 42*(5), 674–690.

52. Uther, W., & Veloso, M. (2003). *Adversarial reinforcement learning*. Technical Report CMU-CS-03-107, School of Computer Science, Carnegie Mellon University, January.
53. Wang, X., & Sandholm, T. (2003). Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15, pp. 1571–1578). Cambridge, MA: MIT Press.
54. Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, King's College, University of Cambridge, May.
55. Young, H. P. (1993). The evolution of conventions. *Econometrica, 61*(1), 57–84.