2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Reinforcement learning with function approximation for cooperative navigation tasks

Francisco S. Melo
Institute for Systems and Robotics,
Instituto Superior Técnico
Lisboa, Portugal
fmelo@isr.ist.utl.pt

M. Isabel Ribeiro
Institute for Systems and Robotics,
Instituto Superior Técnico
Lisboa, Portugal
mir@isr.ist.utl.pt

***Abstract*— In this paper, we propose a reinforcement learning approach to address multi-robot cooperative navigation tasks in infinite settings. We propose an algorithm to simultaneously address the problems of learning and coordination in multi-robot problems. The proposed algorithm extends those existing in the literature, allowing to address simultaneous learning and coordination in problems with an *infinite state-space*. We also present the results obtained in several test scenarios featuring multi-robot navigation situations with partial observability.**

I. INTRODUCTION

Autonomous navigation of mobile robots has been considered a key subject of investigation from the early days of robotic research. In fact, the ability of a robot to accomplish a certain task in a given environment depends, quite often, on the robotŠs ability to navigate in its environment. And, with the appearance of new and demanding robotic applications, there is a natural interest in developing more complex robotic systems, consisting of multiple independent robots. In such multi-robot applications, it is desirable that a each robot be able not only to navigate its environment but also to *adapt* and *coordinate* with the other robots. Classical reinforcement learning (RL) provides an appealing approach to address such adaptability issues [1] and the combination of RL with game theoretic ideas [2], [3] has led to interesting approaches that also address coordination in multi-agent problems [4]–[6].

The general purpose of RL is to find a "good" mapping that assigns "perceptions" to "actions" [1]. In theory, the formalism and methods of RL can be applied to address any optimal control task, yielding optimal solutions while requiring very little *a priori* information on the system itself. However, in practice, RL methods suffer from the *curse of dimensionality* [7] and exhibit limited applicability in complex control problems. Unfortunately, many actual control problems are inherently infinite, described in terms of *continuous state variables*. And, in the particular case of robotic applications, there is often some degree of uncertainty regarding the state of a system (due to noisy sensors, etc.), requiring a robot to decide upon a (real-valued) *belief* that describes some probability distribution. The attractiveness of the RL framework and this abundance of interesting but complex control problems emphasize the need to develop more powerful RL methods.

In this paper we address the problem of simultaneous learning and coordination in multi-agent problems with infinite state-spaces. We combine an approximate version of Q-learning [8] with an approximate coordination mechanism dubbed *approximate biased adaptive play* (ABAP) [9]. The main contribution of the paper is the combination of both algorithms to yield a unified algorithms that simultaneously learns and coordinates in infinite multi-agent settings. Our approach differs from other methods in the literature [6], [10] in several aspects: we assume no communication among the robots and do not require all robots to follow the same decision/coordination algorithm. This is an important advantage: in the presence of a heterogeneous group of robots , our algorithm is still able coordinate to the best decision-rule possible if, for some reason, the other robots act sub-optimally. Finally, we also remark that although in this paper we focus on multi-robot navigation tasks, we remark that the approach described can easily be extended to other application scenarios.

II. MARKOV MODELS FOR NAVIGATION

In this section we review the several models to be used throughout the paper.

A. Markov chains and Markov decision processes

A *homogeneous Markov chain* is a discrete-time stochastic process defined by a pair $(\mathcal{X}, \mathsf{P})$, where $\mathcal{X} \subset \mathbb{R}^p$ is the state-space and $\mathsf{P}(x, U)$ represents the time-independent transition probability from state x to set $U \subset \mathcal{X}$. Given a measurable set $U \subset \mathcal{X}$, the *first return time to* U is defined as

$$\tau_U = \min_{t \in \mathcal{T}} \{X_t \in U, \quad t \geq 1\}.$$

A Markov chain is ψ-*irreducible* if

$$\psi(U) > 0 \Rightarrow \mathbb{P}\left[\tau_U < \infty \mid X_0 = x\right] > 0 \qquad (1)$$

for any $x \in \mathcal{X}$ and ψ is maximal in the sense that if μ is some other measure verifying (1), then $\mu \ll \psi$. If η_U is the number of visits to a measurable set $U \subset \mathcal{X}$ in an infinite trajectory of the chain, the set U is said to be *Harris recurrent* if $\mathbb{P}\left[\eta_U = \infty \mid X_0 = x\right] = 1$ for all $x \in \mathcal{X}$. A ψ-irreducible

From January, 2008, F.S. Melo is with the School of Computer Science, Carnegie Mellon University, USA.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

Markov chain is *Harris recurrent* if any measurable set $U \subset \mathcal{X}$, $\psi(U) > 0$ is Harris recurrent [11].

A *Markov decision problem* (MDP) is a tuple $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ where $\mathcal{X}$ represents the state-space, $\mathcal{A}$ represents the action (or control) space, $\mathsf{P}_a(x, U)$ denotes the *action-dependent* transition probability from a state x to a set $U \subset \mathcal{X}$. The purpose of the decision-maker is to determine the $\mathcal{A}$-valued control process $\{A_t\}$ maximizing

$$V(\{A_t\}, x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, A_t, X_{t+1}) \mid X_0 = x\right], \quad (2)$$

where $0 \leq \gamma < 1$ is a discount-factor and $r(x, a, y)$ is a bounded numerical "reward" received for moving from state $x \in \mathcal{X}$ to state $y \in \mathcal{X}$ after taking action $a \in \mathcal{A}$. The *optimal value function* is defined as

$$V^*(x) = \max_{\{A_t\}} \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^t r(X_t, A_t, X_{t+1}) \mid X_0 = x\right] \quad (3)$$

and verifies Bellman optimality equation [12]. The *optimal Q-function* is defined for each state-action pair as

$$Q^*(x, a) = \int_{\mathcal{X}} \big[r(x, a, y) + \gamma V^*(y)\big] \mathsf{P}_a(x, dy). \quad (4)$$

Finally, the *optimal decision rule* can be obtained from Q^* as

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

The optimal control process is then given by $A_t = \pi^*(X_t)$ and π^* is the *optimal policy* for the MDP $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$.

B. Game theoretic approach to multi-agent systems

Markov games [13] are generalizations of MDPs to multiple decision-makers. Therefore, a Markov game is a tuple $\big(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, (r^k), \gamma\big)$, where N is the number of agents, $\mathcal{X}$ is the state-space, $\mathcal{A}^k$ is the set of *individual actions* for agent k and $\mathcal{A} = \times_{k=1}^{N} \mathcal{A}^k$ is the set of all *joint actions*; P is the controlled transition kernel and r^k is the reward function for agent k.

An *individual policy* for agent k is a *state and time-dependent* probability distribution π_t^k over the set $\mathcal{A}^k$. It defines the probability of agent k playing each action $a^k \in \mathcal{A}^k$ at each time instant and in each state. A *joint policy* is a vector $\pi_t = (\pi_t^1, \ldots, \pi_t^N)$ of individual joint policies and $\pi_t(x, a)$ represents the probability of the joint action a being played in state x at time t when all agents follow the policy π_t. We write $V^{\pi_t}(x)$ instead of $V(\{A_t\}, x)$ whenever the control sequence $\{A_t\}$ is generated by the joint policy π_t, and refer to V^{π_t} as being the *value function* associated with policy π_t.

In this paper, we are interested in *team Markov games*. In team Markov games, all agents share the same reward function, *i.e.*, $r^1 = \ldots = r^N$ and, as such, all have a common goal: to maximize the (common) total expected reward. This total expected reward is defined as in (2), where now $r(x, a, y)$ is the reward received by *all* agents for taking the joint action a in state x and moving to state y. It is immediate to define the *optimal value function* V^* for a team Markov game as in (3) (where now A_t stands for the joint action at time t) and the optimal Q-function, Q^*, as in (4).

If the definition of V^* and the existence of an optimal joint control policy arise immediately from the corresponding results for MDPs, the fact that the decision process in team Markov games is distributed implies that coordination must be addressed explicitly [4]. On the other hand, we note that the function Q^* defines, at each state $x \in \mathcal{X}$, a fully cooperative *matrix game* $\Gamma_x = \big(N, (\mathcal{A}^k), Q^*(x, \cdot)\big)$, that we refer as a *stage-game*. If the agents play an optimal Nash equilibrium in each stage-game Γ_x, the resulting policy is optimal for the team Markov game [14]. As in the single-agent situation, the optimal policy can be determined from Q^* and, in the next section, we discuss how Q^* can be estimated in general infinite problems.

III. LEARNING IN INFINITE PROBLEMS

This section addresses two fundamental issues arising in the class of problems considered in this paper. As seen in the previous section, the optimal control process can be obtained from Q^* both in the single and in the multi-agent situations. In this section we discuss how Q^* can be determined. A second distinct problem is related with the fact that the optimal control process needs not to be unique. In the multi-agent setting this leads to a problem known as *coordination problem* [4], the second issue discussed in this section.

A. Approximate Q-learning

We start by remarking that the optimal Q-function verifies the following recursive relation for every state-action pair

$$Q^*(x, a) = \int_{\mathcal{X}} \big[r(x, a, y) + \gamma \max_{b \in \mathcal{A}} Q^*(y, b)\big] \mathsf{P}_a(x, y).$$

The original Q-learning algorithm [15] implements a stochastic approximation of the recursion above to determine the optimal Q-values. The update-rule for Q-learning is

$$Q_{t+1}(x, a) = Q_t(x, a) + \alpha_t \Delta_t, \quad (5)$$

where Δ_t is the *temporal difference*

$$\Delta_t = R(x, a) + \gamma \max_{b \in \mathcal{A}} Q_t(X(x, a), b) - Q(x, a)$$

and $X(x, a)$ and $R(x, a)$ are $\mathcal{X}$-valued and $\mathbb{R}$-valued random variables obtained according to P_a and r. The sequence $\{\alpha_t\}$ is the *step-size sequence* verifying $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$. Notice that $R(x, a)$ and $X(x, a)$ can be obtained using some simulation/sampling device, not requiring the knowledge of either P or r. However, if $\mathcal{X}$ is an infinite set, it is not possible to straightforwardly apply (5), since it explicitly updates the Q-value for each individual state-action pair and there are infinitely many such pairs.

To circumvent such difficulty, we consider a linear family of functions $\mathcal{Q} = \{Q_\theta\}$ parameterized by a finite-dimensional parameter vector $\theta \in \mathbb{R}^M$. For a fixed set of M bounded, linearly independent functions $\phi_i \in \mathcal{Q}$, any function in $\mathcal{Q}$ can be written as

$$Q_\theta(x, a) = \sum_i \phi_i(x, a)\theta_i = \phi^\top(x, a)\theta,$$

where $^\top$ represents the transpose operator. We want to determine the point θ^* in parameter space such that Q_{θ^*} is the best approximation of Q^* in $\mathcal{Q}$, in some sense. By defining a suitable recursion for θ, we reduce the determination of the infinite-dimensional function Q^* to the determination of a finite-dimensional vector θ^*. In what follows, we assume the functions ϕ_i to verify $\sum_i |\phi_i(x,a)| \leq 1$ and $\|\phi_i\|_\infty = 1$. Notice that this immediately implies the functions to be bounded and linearly independent.

In the original Q-learning algorithm, the temporal difference Δ_t works as a 1-step "estimation error" with respect to (w.r.t.) and the update rule "moves" the estimates Q_t closer to Q^*, minimizing the expected value of Δ_t. Applying the same underlying idea, we resort to a smooth Dirac aproximation g_ε[1] to obtain the following update rule, that we henceforth refer as the *approximate Q-learning*,

$$\theta_{t+1}(i) = \theta_t(i) + \alpha_t g_\varepsilon(x_i, a_i, x_t, a_t)\Delta_t, \tag{6}$$

where the temporal difference Δ_t is

$$\Delta_t = r(x_t, a_t, x_{t+1}) + \gamma \max_{b\in\mathcal{A}} \phi^\top(x_{t+1}, b)\theta - \phi^\top(x_t, a_t)\theta.$$

In the previous update, $\{x_t\}$ and $\{a_t\}$ are state and action trajectories sampled from the Markov chain induced by some fixed learning policy π. The pairs $(x_i, a_i), i = 1, \ldots, M$ are such that $\phi_i(x_i, a_i) = 1$.

We are now in position to introduce our first result, a more detailed version of which can be found in [8]. Let π be a fixed learning policy and $(\mathcal{X}, \mathsf{P}_\pi)$ the corresponding Markov chain with invariant probability measure μ_X, absolutely continuous w.r.t. the Lebesgue measure μ^{Leb}, with a Radon-Nikodym derivative bounded away from zero [16].

Theorem 1: Let $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ be a Markov decision process with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and assume the Markov chain $(\mathcal{X}, \mathsf{P}_\pi)$ to be geometrically ergodic. Suppose that $\pi(x,a) > 0$ for all $a \in \mathcal{A}$ and μ_X-almost all $x \in \mathcal{X}$. Let $\phi_i, i = 1, \ldots, M$ be a set of bounded, linearly independent functions defined on $\mathcal{X} \times \mathcal{A}$ and taking values in R. In particular, admit that $\sum_i |\phi_i(x,a)| \leq 1$ for all pairs (x,a) and that $\|\phi_i\|_\infty = 1$. Then, for ε sufficiently small, the algorithm in 6 converges with probability 1 (w.p.1) as long as the step-size sequence α_t verifies

$$\sum_t \alpha_t = \infty \qquad \sum_t \alpha_t^2 < \infty.$$

Proof: See [8]. ∎

It is important to refer at this point that the quality of the obtained approximation greatly depends on the choice of basis functions, as discussed in [8]. The adequate choice of basis functions is a topic of intense current research.

[1] A smooth Dirac approximation is such that

$$\int_{\mathcal{X}\times\mathcal{A}} g_\varepsilon(x,a,y,u)\mu(dy,du) = 1$$

and

$$\lim_{\varepsilon\to 0} \int_{\mathcal{X}\times\mathcal{A}} g_\varepsilon(x,a,y,u) f(y,u)\mu(dy,du) = f(x,a),$$

where μ is some probability measure on $\mathcal{X} \times \mathcal{A}$.

B. Learning in multi-agent settings

As stated above, learning in multi-agent scenarios must consider two distinct problems: *learning the game* and *learning to coordinate*. Learning the game deals with determining Q^*. Once this function is known, the agents are able to determine the optimal policies and, if necessary, deal with the problem of coordination (i.e., learn to coordinate).

We start with the problem of learning the game. The only difference between applying approximate Q-learning to MDPs and to team Markov games lies on the fact that, in the latter, the action sequence $\{A_t\}$ is generated in a distributed fashion by the N agents in the game. This does not affect in any way the convergence of the algorithm and the sequence θ_t will converge w.p.1 to the same limit θ^* as it would in the single agent situation, for an adequate learning policy.

We now discuss the problem of coordination. Consider the scenario depicted in Fig. 1.

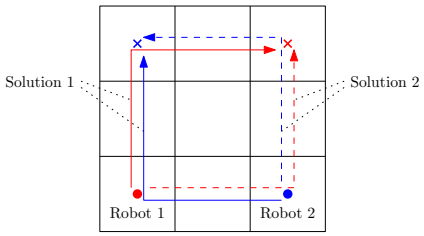


Fig. 1. Example with 2 robots in a 2×2 grid-world.

Two robots (1 and 2) must move from the corresponding cell in the bottom row to the opposite cell in the top row, without colliding with each other (*i.e.*, lying in the same cell). There are several optimal ways of doing this, two of which are depicted in Fig. 1. Suppose now that Robot 1 opts by choosing Solution 2 and Robot 2 opts by choosing Solution 1 (we assume no communication between the robots). This means that they will collide in the middle cell in the bottom row, which is an undesirable behavior.

This problem is known as a *coordination problem* [4]. Even if the robots know the model and the solutions, it is still necessary to devise some specific mechanism to ensure that, in the presence of multiple solutions, all robots commit to the same. This mechanism can rely on implicit assumptions on the way robots choose their actions [17], communication [18], social conventions [19] or coordination graphs [6].

In this paper we are interested in coordination *emerging* from the interaction among the robots, rather than "intrinsically implanted". We also consider that no *explicit communication* takes place. As such, we will make use of use *biased adaptive play* (BAP) [20], since it can easily be combined with Q-learning [21]. In order to address problems with infinite state-spaces, we will introduce a variation of BAP that can handle such problems.

The basic working of BAP is as follows. At each time step, each agent samples the history of past plays (at the corresponding stage-game) and uses these samples estimate the average policies of the other agents in that game (by

computing a simple average). Then, using standard game-theoretic reasoning, it is able to choose a best response to such policy, as long as the game is known.

As seen in [20], [21] (and similarly to standard Q-learning), BAP requires each state to be visited "infinitely often" for convergence to be ensured. However, in problems with infinite state-space such condition is generally impossible to ensure and BAP cannot be successfully applied. The reason behind this impossibility is easy to grasp: BAP relies on past plays of the stage-game at each state, and there is the possibility that a particular state has never been visited before, no matter for how long the agents wander in the environment.

In adapting BAP to infinite state-spaces, coordination should rely not only in past visits to one particular state but in the information provided by *nearby states*. As the agents can no longer use the past history at a given state to infer the other agents' policy at that state, they use the past history at states that are close to the desired state. If the game is "well-behaved", the optimal action in two states that are sufficiently close will be the same.

Let $\Gamma = \left(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, r, \gamma\right)$ be a team Markov game with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and finite action space $\mathcal{A}$. To simplify the argument, we assume the underlying chain to be ψ-irreducible and Harris recurrent, independently of the agents' choice of actions. We further assume that the optimal function Q^* is known and that the irreducibility measure ψ is absolutely continuous w.r.t. the Lebesgue measure on $\mathcal{X}$. These assumptions greatly simplify the argument but, as discussed ahead, can easily be alleviated without affecting the validity our result.

Let $\mathcal{H}_t = \{X_0, A_0, X_1, \ldots, X_{t-1}, A_{t-1}\}$ be the history of past plays up to time t. At each time instant t, each agent determines the distance between the current state X_t and each state X_i occurring in $\mathcal{H}_t$, given by $\|X_i - X_t\|$. It then chooses m such occurrences so as to minimize the corresponding distance. The set thus obtained, denoted as $S_m(X_t, \mathcal{H}_t)$, contains the m elements in $\mathcal{H}_t$ minimizing the total distance to X_t. We remark that a particular state $x \in \mathcal{X}$ may occur in $S_m(X_t, \mathcal{H}_t)$ more than once. Also, if two occurrences X_{t_i} and X_{t_j} verify $\|X_t - X_{t_i}\| = \left\|X_t - X_{t_j}\right\|$ and one must be chosen, then the most recent one should be picked (*e.g.*, if $t_j > t_i$ in the previous situation, X_{t_j} would be chosen). Due to the ψ-irreducibility and Harris recurrence of the Markov chain, 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, $S_m(x, \mathcal{H}_t) \subset U$ for $t > T_0$. Once the set $S_m(X_t, \mathcal{H}_t)$ is determined, the corresponding m plays can now be used to proceed as in standard BAP (see [20], [21] for details on BAP). We refer to this coordination mechanism as *approximate BAP* (ABAP).

The next theorem establishes the convergence of ABAP.

Theorem 2: Let $\{X_t\}$ be the ψ-irreducible and Harris recurrent Markov chain obtained from the team Markov game, as defined above. Further assume that $\psi \ll \mu^{\text{Leb}}$ and that r is continuous ψ-almost everywhere (ψ-a.e.).[2] Then, the agents following ABAP coordinate in an optimal Nash equilibrium w.p.1 in ψ-almost every Γ_x.

Proof: See [9]. ∎

IV. The CAQL algorithm

In this section we introduce the main contribution the paper, namely the *coordinated approximate Q-learning* (CAQL) algorithm. As anticipated, this algorithm combines approximate Q-learning and ABAP. With sufficient exploration, CAQL guarantees that the estimates Q_t converge to a suitable approximation of Q^* and the agents' policies converge to an optimal policy w.r.t. this approximation.

The basic procedure of CAQL is as follows. At each time instant t, each agent k determines the set $S_m(X_t, \mathcal{H}_t)$ using the similarity function ϕ. From this set, the agent uses BAP to estimate the expected payoff of each action $a^k \in \mathcal{A}^k$ w.r.t. a *virtual game* VG_t obtained from Q_t and chooses its individual action as perscribed by standard BAP [20]. This virtual game is a matrix game $VG_t = (N, (A^k), r_t)$ where $r_t(a)$ takes the value 1 if, at state X_t, the joint action a is δ_t-*optimal w.r.t.* Q_t, *i.e.*, if $Q_t(X_t, a) \geq \max_{b \in \mathcal{A}} Q_t(X_t, b) - \delta_t$. Once all individual actions A_t^k are chosen, yielding the joint action A_t, the game moves to a new state X_{t+1} according to the probabilities in P and all agents receive the corresponding reward $r(X_t, A_t, X_{t+1})$. All agents now use the observed transition $(X_t, A_t, r(X_t, A_t, X_{t+1}), X_{t+1})$ to update the parameter vector θ_t

Several remarks are now in order. First of all, Theorem 1 requires a *fixed* learning policy that induces a geometrically ergodic Markov chain; Theorem 2 requires the Markov chain to be ψ-irreducible and Harris recurrent. It is easy to verify [11] that the former (*i.e.*, geometric ergodicity) implies the latter (*i.e.*, ψ-irreducibility and Harris recurrence).

Secondly, the use of ABAP necessarily means that the learning policy *is not fixed*. Therefore, to ensure that both algorithms are compatible, we need to guarantee that the learning policy *changes slowly* (so that it "seems" fixed in terms of the learning algorithm) and ensures sufficient exploration (to meet the requirement $\pi(x, a) > 0$). The use of GLIE policies (greedy in the limit with infinite exploration [22]) readily solves this problem. Also, we emphasize that the exploration of sub-optimal actions does not affect the convergence of ABAP, as long as the probability of choosing an "exploratory action" eventually decays to zero [20], as is the case in a GLIE policy. The rate at which this exploration probability must decay essentially depends on the step-size sequence, as discussed in [22].

Thirdly, the δ_t parameter used to build the virtual game VG_t accounts for the fact that the agents are deciding upon *an estimate* Q_{θ_t}, instead of the limit function Q_{θ^*}. However, as $t \to \infty$, the δ_t parameter should decay to zero. This needs to be done at an appropriate rate, to ensure that no optimal actions (w.r.t. Q_{θ^*}) are ruled out too soon. That rate depends

[2] We denoted by μ^{Leb} the Lebesgue measure in $\mathbb{R}^p$.

on the rate of convergence of the algorithm. From [20], [23] it can be shown that, as long as

$$\lim_{t\to\infty} \frac{\sqrt{\frac{\log\log(t)}{t}}}{\delta_t} = 0, \tag{7}$$

no optimal action is ruled out too soon.

All these considerations are formalized and summarized in the following final result.

Theorem 3: Let $(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, r, \gamma)$ be a team Markov game with compact state-space $\mathcal{X} \subset \mathbb{R}^p$ and finite action-space $\mathcal{A} = \times_{k=1}^{N} \mathcal{A}^k$. Let π_{θ_t} be the θ_t-dependent policy obtained from CAQL and assume that such policy verifies, for all pairs (x, a),

$$|\pi_\theta - \pi_{\theta'}| \leq C \, \|\theta - \theta'\|$$

for some constant $C > 0$ independent of x and a. Assume that the Markov chain induced by the policy π_θ, denoted as $(\mathcal{X}, \mathsf{P}_\theta)$, is geometrically ergodic for each θ with invariant probability measure $\mu_X^\theta \ll \mu^{\mathrm{Leb}}$, with a Radon-Nikodym derivative bounded away from zero. Further assume that

- For each θ, the reward function r is continuous μ_X^θ-a.e.;
- The parameter δ_t decreases monotonically to zero and verifies (7);
- The conditions of convergence for BAP are met [20];

Then, the sequence θ_t generated by CAQL converges w.p.1 to a parameter vector θ^* and all agents converge in behavior w.p.1 to a common, optimal policy w.r.t. Q_{θ^*}.

The proof of this theorem essentially formalizes the ideas presented above and can be found in [24]

V. SOME ILLUSTRATIVE RESULTS

In this section we describe the results obtained in several benchmark problems from the literature, featuring multi-robot navigation tasks. We consider scenarios with a finite number of states but include partial state observability arising from noisy sensor measurements. As is well known [25], it is possible to redefine the decision process in terms of *beliefs*, computed using standard Markov localization [26]. This reduces the decision problem to a Markov game with infinite state-space.

A. Test scenarios

In Figure 2 we describe some test scenarios for CAQL, most of which are standard benchmarks used in the POMDP literature [27]. The dark cells correspond to rooms and the light ones correspond to hallways. Each scenario is partitioned into discrete cells and, in each cell, each robot can lie in 4 possible orientations. Each pair of color-matching cells represents a starting/goal pair of cells for one particular robot in the team.

In our formulation of the problem, all possible *joint positions* for the group of robots must be considered. We use a team Markov game $\big(N, \mathcal{X}, (\mathcal{A}^k), \mathsf{P}, r, \gamma\big)$ to model the corresponding navigation problem. We also account for *partial observability* by considering a *centralized sensor* that provides *all robots* with similar sensorial information on the whole team. In particular, we consider that surveillance cameras keep the environment monitored. The images from the cameras are processed in a central processor and the processed data is then sent to all the robots. This processed data contains the state of each robot as perceived by the cameras. We consider that each robot has a distinctive feature that allows the cameras to distinguish the different robots while perceiving the state (position and orientation) of each robot. Due to the fact that all robots receive the same observation, it is possible to define a *common belief* over the state-space $\mathcal{X}$, as in Section II. Therefore, we can immediately cast the team Markov game with partial observability as an equivalent team Markov game with infinite state-space, defined in terms of beliefs: all robots maintain the same belief on the position of the team in the environment and decide upon this belief. Notice that, for each scenario, the corresponding belief will be a $|\mathcal{X}|$-dimensional probability vector, with $|\mathcal{X}|$ the cardinality of $\mathcal{X}$.

The team receives a reward of $+20$ for reaching the goal configuration. Also whenever two robots choose an action that can lead both robots to end up in the same cell, the team receives a "penalty" of -10 and the movement does not succeed. A more detailed description of the test scenarios and robot models can be found in [24].

B. Results

We used a simulator to generate state transitions, observations and immediate rewards in the various scenarios described. The initial belief state for each robot corresponds to a uniform distribution over all non-goal states. Every time the team reaches the goal configuration, it is reset to the initial configuration. We allow the algorithm to learn for a period of 10^6 time steps, to ensure sufficient exploration of the state-action space. We then ran a series of trials on each learnt policy to evaluate its performance. A single trial consisted of a truncated trajectory of 250 simulated steps starting from the initial state. The immediate rewards were appropriately discounted and then added to yield a sample of the total discounted reward. This was repeated for $2,000$ independent trials. The discount factor was 0.95 for all experiments. We also recorded for each trial whether the team was able to successfully reach the goal configuration within the 250 time steps. We determined the percentage of successful trials and used this percentage as a second performance measure.

We applied CAQL to each scenario, using the natural basis functions arising from the beliefs b_t. In particular, we used the basis functions $\phi_{i,a}$, $i = 1, \ldots, |\mathcal{X}|$, $a = 1, \ldots, |\mathcal{A}|$, with each $\phi_{i,a}$ given by $\phi_{i,a}(b, u) = b(i)\mathbb{I}_a(u)$, for all beliefs b and actions $u \in \mathcal{A}$. We denoted by $\mathbb{I}_a$ the indicator function for the set $\{a\}$. Using this approximation, the learnt parameter vector θ^* has the same dimension as the Q-functions for the fully observable game. We used Boltzmann exploration to ensure a suitable exploration/exploitation tradeoff.

The total reward obtained during the learning period, for each of the 6 scenarios considered, is summarized in Figure 3. The slope of the curve provides a rough indicator

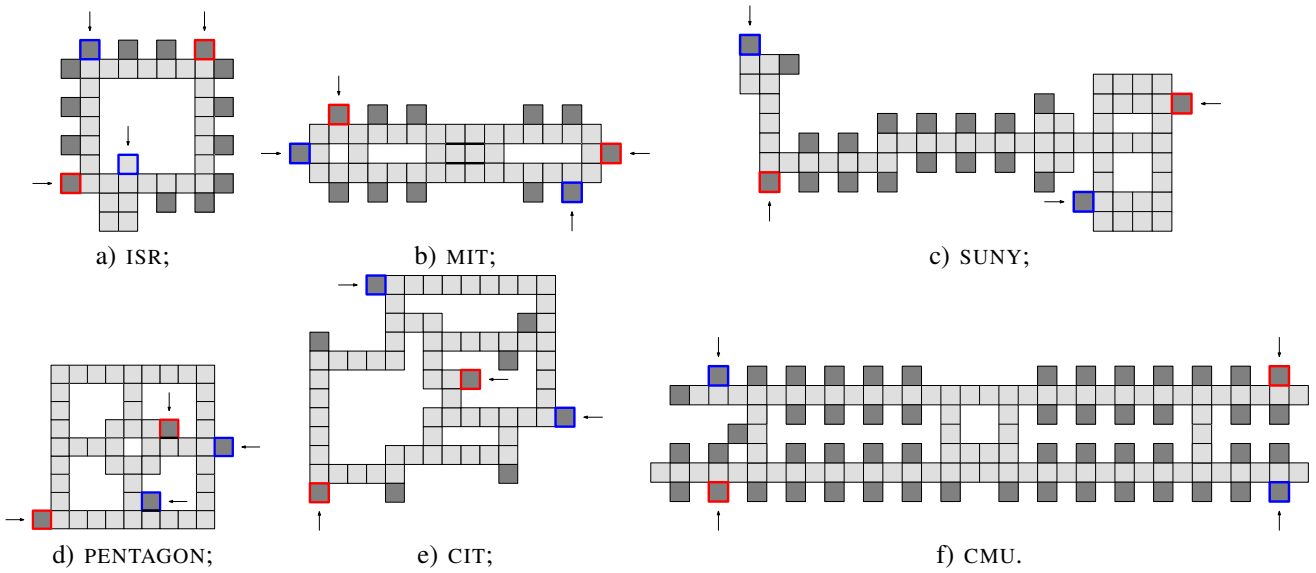


Fig. 2. Scenarios used for the topological navigation experiments.

of the performance of the team. Notice that, as $t \rightarrow \infty$, the exploration decays and it is evident from the figures the instant when coordination was attained.

The total discounted reward in the test period (after learning was terminated) is summarized in Table I. The results presented correspond to the average over $2,000$ independent Monte-Carlo trials. For the purpose of comparison, we also present the results for a group of robots using only approximate Q-learning (without coordination).

TABLE I

TOTAL DISCOUNTED REWARD AND PERCENTAGE OF SUCCESSFUL MISSIONS IN THE DIFFERENT EXPERIMENTS USING CAQL AND APPROXIMATE Q-LEARNING WITHOUT COORDINATION (BOTH AFTER THE LEARNING PERIOD IS COMPLETE). WE PRESENT THE AVERAGE TOTAL DISCOUNTED REWARD AND STANDARD DEVIATION OBTAINED OVER 2000 MONTE-CARLO RUNS.

Env.	CAQL	No-Coord
ISR	9.916 (100%)	5.976 (100.00%)
MIT	6.963 (100%)	3.992 (100.00%)
PENTAGON	9.275 (100%)	5.036 (100.00%)
CIT	5.900 (100%)	3.666 (99.90%)
SUNY	1.800 (100%)	0.090 (94.85%)
CMU	1.628 (100%)	0.438 (99.10%)

When comparing the results of the "uncoordinated" team with those of the coordinated team, it is evident that the use of the coordination mechanism greatly improves the performance of the team. Notice that both teams learn the same Q-values, as they both use the same algorithm to learn the game. This makes even more striking the difference in performance observed in the two tests.

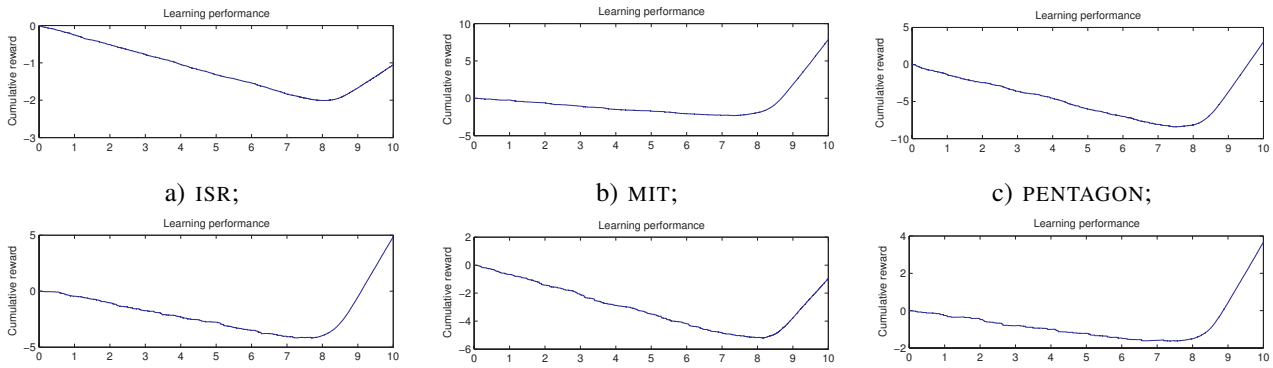
Also notice that the absence of coordination in terms of the success rate of the team is much more relevant in the larger environments. This fact can easily be interpreted. First of all, the success rate measures the number of trials that the team was able to reach the final configuration. In the smaller environments, the final configuration can be reached very rapidly, as long as the robots are able to minimally coordinate. Therefore, mis-coordinations do affect the total discounted reward received, but will hardly prevent the team from reaching the goal. In the larger scenarios, because of the size of the environments, reaching the goal takes a significant amount of time and even one mis-coordination may translate in a delay that the team cannot afford. This indicates that coordination mechanisms do have a decisive influence in the team's ability to complete complex missions.

VI. CONCLUDING REMARKS

In this paper we introduced the CAQL algorithm to address multi-agent RL problems with infinite state-spaces. We explored the applicability of this methodology in cooperative navigation tasks by applying CAQL to several large benchmark problems from the literature. In these test scenarios, a group of mobile robots with centralized sensors must navigate from an initial configuration to a target configuration. We applied CAQL to this set of problems and verified that, in all situations, the team is able to coordinate and reach the target configuration, exhibiting a nearly perfect performance.

We note that CAQL has a broader applicability than robotic navigation tasks, even if in the paper the method was introduced envisioning this specific application. In fact, CAQL can be used to address any general multi-agent decision problem where coordination is fundamental. On the other hand, robotic navigation tasks are particularly "well-behaved", since they exhibit some locality in the transitions (a robot cannot "jump" between arbitrary states). This locality helps to decrease uncertainty and makes robotic tasks particularly amenable to a reinforcement learning approach relying in belief-states.

Two final comments on ABAP mechanism. Since we are considering CAQL algorithm to run along an infinite trajectory, storing the complete history (as required for ABAP) is infeasible. In a practical implementation, we can rely on a



a) ISR; b) MIT; c) PENTAGON;

d) CIT; e) SUNY; f) CMU.

Fig. 3. Cumulative reward during the learning period. The scale in both axis should be multiplied by 10^5.

fixed-size history, sufficiently large to properly sample the state-space in a representative way. The exact length of the history to be chosen will depend on the irreducibility measure associated with the sampled chain and with the support of Q^*. The second remark is concerned with the implementation of the learning period. Allowing the learning period to be conducted with the actual robot is often time consuming and in some situations even lead to damage of the robot. Therefore, it is customary to develop simplified simulation models to run the learning process. Such simulation models often capture the fundamental situations where decision-making is required from the robot and allow for a much faster and hazard-safe learning period. However, simulation models can only provide approximate representations of the actual situations and it is convenient that, once the learnt policy is implemented in the real robot, the learning process is allowed to continue (with no exploration) as the robot interacts with the actual environment.

ACKNOWLEDGEMENTS

Work partially supported by POS_C that includes FEDER funds. Francisco S. Melo acknowledges the PhD grant SFRH/BD/3074/2000.

REFERENCES

[1] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
[2] M. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Machine Learning*, 1994, pp. 157–163.
[3] J. Hu and M. Wellman, "Nash Q-learning for general sum stochastic games," *J. Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.
[4] C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *Proc. 16th Int. Joint Conf. Artificial Intelligence*, 1999.
[5] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proc. 15th Nat. Conf. Artificial Intelligence (AAAI'98)*, 1998, pp. 746–752.
[6] C. Guestrin, M. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," in *Proc. 19th Int. Conf. Machine Learning*, 2002.
[7] R. Bellman, *Dynamic Programming*. Dover Publications, Inc., 2003.
[8] F. Melo and I. Ribeiro, "Q-learning with linear function approximation," in *Proc. 20th Annual Conf. Learning Theory*, 2007, pp. 308–322.
[9] F. Melo and M. I. Ribeiro, "Emerging coordination in infinite team Markov games," in *Proc. 7th Int. Conf. Autonomous Agents and Multiagent Systems*, 2008 (to appear).
[10] J. Kok, M. Spaan, and N. Vlassis, "An approach to noncommunicative multiagent coordination in continuous domains," in *Proc. 12th Belgian-Dutch Conf. Machine Learning*, 2002, pp. 46–52.
[11] S. Meyn and R. Tweedie, *Markov Chains and Stochastic Stability*. Springer-Verlag, 1993.
[12] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.
[13] L. Shapley, "Stochastig games," *Proc. National Academy of Sciences*, vol. 39, pp. 1095–1100, 1953.
[14] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Proc. 6th Conf. Theoretical Aspects of Rationality and Knowledge*, 1996, pp. 195–210.
[15] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, University of Cambridge, May 1989.
[16] P. Halmos, *Measure Theory*. Springer, 1974.
[17] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *Proc. 17th Int. Conf. Machine Learning*, 2000, pp. 535–542.
[18] F. Fischer, M. Rovatsos, and G. Weiss, "Hierarchical reinforcement learning in communication-mediated multiagent coordination," in *Proc. 3rd Int. Joint Conf. Autonomous Agents and Multiagent Systems*, 2004, pp. 1334–1335.
[19] N. Findler and R. Malyankar, "Social structures and the problem of coordination in intelligent agent societies," Agent-Based Simulation, Planning and Control Session, IMACS World Congress, 2000.
[20] X. Wang and T. Sandholm, "Reinforcement learning to play an optimal Nash equilibrium in team Markov games," in *Advances in Neural Information Processing Systems*, 2003, vol. 15, pp. 1571–1578.
[21] F. Melo and M. I. Ribeiro, "Learning to coordinate in topological navigation tasks," in *Proc. 6th IFAC Symp. Intelligent Autonomous Vehicles*, 2007.
[22] S. Singh, T. Jaakkola, M. Littman, and C. Szepesvari, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–310, 2000.
[23] M. Pelletier, "On the almost sure asymptotic behaviour of stochastic algorithms," *Stochastic Processes and their Applications*, vol. 78, pp. 217–244, 1998.
[24] F. Melo, "Reinforcement learning in coordinated navigation tasks," Ph.D. dissertation, Instituto Superior Técnico, July 2007 (submitted).
[25] A. Cassandra, L. Kaelbling, and M. Littman, "Acting optimally in partially observable stochastic domains," in *Proc. 12th Nat. Conf. Artificial Intelligence*, 1994, pp. 1023–1028.
[26] D. Fox, "Markov localization: A probabilistic framework for mobile robot localization and navigation," Ph.D. dissertation, University of Bonn, Germany, 1998.
[27] A. Cassandra, "Exact and approximate algorithms for partially observable Markov decision processes," Ph.D. dissertation, Brown University, May 1998.