2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Adaptive Play Q-Learning with Initial Heuristic Approximation

Andriy Burkov
Département d'informatique et de génie logiciel
Université Laval
Sainte-Foy, QC, Canada, G1K 7P4
Email: burkov@damas.ift.ulaval.ca

Brahim Chaib-draa
Département d'informatique et de génie logiciel
Université Laval
Sainte-Foy, QC, Canada, G1K 7P4
Email: chaib@damas.ift.ulaval.ca

Abstract— **The problem of an effective coordination of multiple autonomous robots is one of the most important tasks of the modern robotics. In turn, it is well known that the learning to coordinate multiple autonomous agents in a multiagent system is one of the most complex challenges of the state-of-the-art intelligent system design. Principally, this is because of the exponential growth of the environment's dimensionality with the number of learning agents. This challenge is known as "curse of dimensionality", and relates to the fact that the dimensionality of the multiagent coordination problem is exponential in the number of learning agents, because each state of the system is a joint state of all agents and each action is a joint action composed of actions of each agent. In this paper, we address this problem for the restricted class of environments known as goal-directed stochastic games with action-penalty representation. We use a single-agent problem solution as a heuristic approximation of the agents' initial preferences and, by so doing, we restrict to a great extent the space of multiagent learning. We show theoretically the correctness of such an initialization, and the results of experiments in a well-known two-robot grid world problem show that there is a significant reduction of complexity of the learning process.**

I. Introduction

The problem of an effective coordination of multiple autonomous robots is one of the most important tasks of the modern robotics. In turn, it is well known that the learning to coordinate multiple autonomous agents in a multiagent system is one of the most complex challenges of the state-of-the-art intelligent system design. Principally, this is because of the exponential growth of the environment's dimensionality with the number of learning agents. This challenge is known as "curse of dimensionality", and relates to the fact that the dimensionality of the multiagent coordination problem is exponential in the number of learning agents since each state of the system is a joint state of all agents and each action is a joint action composed of actions of each agent.

To reduce the state space of a single-agent system a variety of methods have been proposed. One of them is the so called heuristic search. Essentially, heuristic search is a set of methods based on the knowledge of a heuristic function that can estimate the real utility of any visited state. Generally, if that heuristic function is sufficiently informative and satisfies certain conditions, then the algorithm using it does not need to visit the entire state space to find the solution. Unfortunately, in multiagent systems, in most cases an explicit search in the state space is practically impossible, since the search supposes that the properties of the environment in each state are known to the agent. This is not the case when there are several, possibly adversarial, agents affecting the environment, and their policies and rationality principles are not known to the learning agent. Thus, since the centralized planning in that context is not always possible the agents are usually faced with the learning or adaptation tasks, which, as it was noted, have an exponential dimensionality.

In this paper, we address the problem of multiagent learning complexity reduction in a specific context, namely, in the goal-directed stochastic games with action-penalty representation. In such context, all agents have their respective goals and the rewards of making an action are negative in any state except the goal state. The idea is to use a single-agent problem solution as a heuristic approximation of the agents' initial preferences. Recently, it was shown that this approach permits to restrict to a great extent the space of multiagent learning [1]. We show theoretically the correctness of such an initialization. To do that, we provide the proofs of admissibility and monotonicity (consistence) of the proposed heuristic function.

As a basis for our approach, we use the Adaptive Play Q-learning (APQ) algorithm. The theoretical base of the Adaptive Play was founded by Young [2] and then this technique was extended to the multiagent learning context with good empirical results [3]. Further in this paper, this learning algorithm will be described in more detail.

In the sections that follow we give a detailed description of our framework and, more precisely, the assumptions we made about the structure of the environment and the agents' initial knowledge. We then demonstrate the correctness of the approximation of multiagent solution by an optimal single-agent one in the goal-directed stochastic games with action-penalty representation. Further, we evaluate the behavior of our algorithm on a simple test bench: a two-robot grid world problem. We conclude with a survey of some previous work related to our approach and present an overview of our future work.

II. Notation and Concepts

First, we consider Markov Decision Process (MDP). An MDP is an environment which has a Markovian inter-state transition model, additive rewards, and the state where the agent finds itself at each moment of time is fully observable.

1-4244-0602-1/07/$20.00 ©2007 IEEE.

More formally, an MDP is defined as a tuple, $(\mathcal{S}, \mathcal{A}, T, R)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, T is the transition function, $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$, R is the reward function, $\mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, and $s_0 \in \mathcal{S}$ is the initial state. A solution in MDPs is called policy. A policy π assigns to each state, which is possible to be visited if the agent follows this policy, an action to execute. If we let s_t be the state the agent is in after executing π during t steps, a utility $U(\pi(s))$ of a policy π in a state s is

$$U(\pi(s)) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) | s_0 = s\right] \quad (1)$$

A learning task assumes that the agents do not have preliminary knowledge about the environment in which they act. A learning agent should calculate an optimal policy $\overset{\star}{\pi}$ by making a number of trials, i.e., by interacting with the environment. Q-learning [4] is a dynamic programming method that consists in calculating the utility of an action in a state by interacting with the environment. The goal of Q-learning is to create a function $Q : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ assigning to each state-action pair a Q-value, $Q(s, a)$, that corresponds to the agent's expected reward of executing an action a in a state s and following infinitely an optimal policy starting from the next state s':

$$Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_a Q(a, s')$$

where $\gamma \in (0, 1]$ is the discount factor used to avoid infinite summation.

Since the transition function, T, is not known for the learning agent, Q-learning consists in estimating the real value $\overset{\star}{Q}(s, a)$ by executing action a in state s of the environment, observing the reward $R(s, a)$ obtained and the system's next state s', using following update rule:

$$\hat{Q}(s, a) \leftarrow (1-\alpha)\hat{Q}(s, a) + \alpha[R(s, a) + \gamma \max_a \hat{Q}(s', a)] \quad (2)$$

where $\hat{Q}(s, a)$ is an estimated value of $\overset{\star}{Q}(s, a)$ and $\alpha \in [0, 1]$ is the learning rate. All along the learning process the agent selects actions to execute in each state by maximizing the Q-value in that state with some stochastic exploration which decreases over time. The convergence of the estimated Q-values, $\hat{Q}(s, a)$, to their optimal values, $\overset{\star}{Q}(s, a)$, was proven in [4] under the conditions that each state-action pair is updated infinitely often, rewards are bounded and α tends asymptotically to 0. It was then shown in [5] that Q-learning in general case may have an exponential computational complexity. But such a complexity may be substantially reduced (to some small polynomial function in the size of the state space) if an appropriate reward structure is chosen and if Q-values are initialized with some "good" values. An appropriate reward structure is the so-called *action-penalty representation* where the agent is penalized for every executed action in every state except the goal states. In fact, the action-penalty representation is the most frequent reward structure in MDPs and the problems that are solved in such MDPs are called *stochastic shortest path* problems. In turn, although the Q-learning technique has existed for decades, the initialization of Q-values has not been explored much in the literature, substantially because a good heuristic approximation cannot be easily found for the problems where environment is not known, as well as the location of the goal states and the reward function.

In the following sections, we will show that in the multiagent case there may exist a good heuristic function to initialize the Q-values. Before that, we need to introduce some important game theoretic concepts and notation.

A. Game Theoretic Concepts

A (normal form) stage game is a tuple $(n, \mathcal{A}^{1\ldots n}, R^{1\ldots n})$, where n is the number of players, $\mathcal{A}^j$ is the strategy space of player $j, j = 1 \ldots n$, and the value function $R^j : \times\mathcal{A}^j \mapsto \mathbb{R}$ defines the utility for player j of a joint action $\mathbf{a} \in \mathbf{A} = \times\mathcal{A}^j$.

A *mixed* strategy for player j is a distribution π^j, where $\pi^j_{a^j}$ is the probability for player j to select some action a^j. A strategy is *pure* if $\pi^j_{a^j} = 1$ for some a^j. A *strategy profile* is a collection $\Pi = \{\pi^j | j = 1 \ldots n\}$ of all players' strategies. A *reduced profile for player* j, $\Pi^{-j} = \Pi \backslash \{\pi^j\}$, is a strategy profile containing strategies of all players except j, and $\Pi^{-j}_{\mathbf{a}^{-j}}$ is the probability for players $k \neq j$ to play a joint action $\mathbf{a}^{-j} \in \mathbf{A}^{-j} = \times\mathcal{A}^{-j}$ where $\mathbf{a}^{-j}$ is $\langle a^k | k = 1 \ldots n, k \neq j \rangle$. Given a player j and a reduced profile Π^{-j}, a strategy $\hat{\pi}^j$ is a *best reply* (BR) to Π^{-j} if the expected utility of the strategy profile $\Pi^{-j} \cup \{\hat{\pi}^j\}$ is maximal for player j. Since a best reply may not to be unique, there is a set of best replies of player j to a reduced profile Π^{-j} which is denoted as $BR^j(\Pi^{-j})$. More formally, the expected utility of a strategy profile Π for a player j is given by:

$$U^j(\Pi) = \sum_{a^j \in \mathcal{A}^j} \pi^j_{a^j} \sum_{a^{-j} \in \mathbf{A}^{-j}} R(\langle a^j, \mathbf{a}^{-j} \rangle) \Pi^{-j}_{\mathbf{a}^{-j}}$$

where Π is $\Pi^{-j} \cup \{\pi^j\}$ and $R(\langle a^j, \mathbf{a}^{-j} \rangle)$ is the value that player j receives if the joint action $\mathbf{a} = \langle a^j, \mathbf{a}^{-j} \rangle$ is played by all players. In this case, a best reply of player j to the reduced profile Π^{-j} is a strategy $\hat{\pi}^j$ such that:

$$U^j(\Pi^{-j} \cup \{\hat{\pi}^j\}) \geq U^j(\Pi^{-j} \cup \{\pi^j\}) \quad \forall \pi^j \neq \hat{\pi}^j$$

Solution in the game theoretic framework is called *equilibrium*. A strategy profile Π forms a *Nash equilibrium* if a *unilateral* deviation of each player j from Π does not increase its own expected utility, or, in other words, Π is a Nash equilibrium if and only if for each player j its strategy $\hat{\pi}^j \in \Pi$ is a best reply to the reduced profile Π^{-j}, that is:

$$\hat{\pi}^j \in BR^j(\Pi^{-j}) \quad \forall j$$

For simplicity of presentation, in the rest of this paper we use the term "equilibrium" to denote the Nash equilibrium.

B. Stochastic Games

Stochastic games (SGs) combine MDPs and stage games. An SG is a tuple $(n, \mathbf{S}, \mathcal{A}^{1\ldots n}, T, R^{1\ldots n})$, where n is the number of agents, $\mathbf{S}$ is the set of states $\mathbf{s} \in \mathbf{S}$ now represented as vectors, $\mathcal{A}^j$ is the set of actions $a^j \in \mathcal{A}^j$ available to agent

j, $\mathbf{A}$ is the joint action space $\mathcal{A}^1 \times \ldots \times \mathcal{A}^n$, T is the transition function: $\mathbf{S} \times \mathbf{A} \times \mathbf{S} \mapsto [0, 1]$, R^j is the reward function for agent j: $\mathbf{S} \times \mathbf{A} \mapsto \mathbb{R}$ and $\mathbf{s}_0 \in \mathbf{S}$ is the initial state.

Since there are multiple agents selecting actions, the agent's next state and rewards depend on the joint actions of all players[1]. It's easy to see that if in an SG there is only one player then this SG becomes MDP. The goal of each agent in an SG is to maximize its expected utility of being in this game. In the stochastic games framework the "expected utility" is a combination of two expectations in the sense that the agents in an SG aim to maximize their expected utilities over other players' joint strategy in each stage game (state), and their temporal utility over all future games. Formally, for an agent j, the discounted utility U^j of a state $\mathbf{s}$ of an SG is defined as follows:

$$\begin{aligned} U^j(\Pi(\mathbf{s})) &= E\left[\sum_{t=0}^{\infty} \gamma^t u^j(\Pi(\mathbf{s}_t)) | \mathbf{s}_0 = \mathbf{s}\right] \\ &= u^j(\Pi(\mathbf{s})) + \gamma \sum_{\mathbf{s}' \in \mathbf{S}} T(\mathbf{s}, \Pi(\mathbf{s}), \mathbf{s}') U^j(\Pi(\mathbf{s}')) \end{aligned} \quad (3)$$

where u^j is the "immediate" expected utility of a stage game $\mathbf{s}_t$ for the agent j, Π is the policy of joint strategies of players, which defines a strategy profile $\Pi(\mathbf{s})$ for each state $\mathbf{s} \in \mathbf{S}$. In SGs a policy Π is a Nash equilibrium if and only if in each state $\mathbf{s} \in \mathbf{S}$ the strategy profile, $\Pi(\mathbf{s})$, forms this kind of equilibrium.

The algorithm we use as a basis for our approach is called Adaptive Play Q-learning (APQ) [3]. This algorithm is based on Q-learning combined with the Adaptive Play [2] to calculate for each player a policy of best reply to other players' strategies. Note that APQ was chosen solely to demonstrate the reduction of learning complexity, since we needed an algorithm which operates with Q-values and converges in stochastic games (or at least in a subclass of SGs). This choice, however, is not critical for our approach and any other multiagent learning algorithms having these properties may be suitable as well.

III. Adaptive Play Q-learning

Formally, each player j playing the Adaptive Play saves in memory a history $H_t^j = \{\mathbf{a}_{t-p}^{-j}, \ldots, \mathbf{a}_t^{-j}\}$ of the last p joint actions played by the other players. To select a strategy to play at time $t+1$ each player randomly and irrevocably samples from H_t^j a set of examples of length l, $\hat{H}_t^j = \{\mathbf{a}_{k_1}^{-j}, \ldots, \mathbf{a}_{k_l}^{-j}\}$, and calculates the empiric distribution $\hat{\Pi}^{-j}$ as an approximation of the real reduced profile of strategies played by the other players, using the following:

$$\hat{\Pi}_{\mathbf{a}^{-j}}^{-j} = \frac{C(\mathbf{a}^{-j}, \hat{H}_t^j)}{l}$$

where $C(\mathbf{a}^{-j}, \hat{H}_t^j)$ is the number of times that the joint action $\mathbf{a}^{-j}$ was played by the other players according to the set $\hat{H}_t^j$. Given the probability distribution over the other players' actions, $\hat{\Pi}^{-j}$, the player j plays its best reply, $BR^j(\hat{\Pi}^{-j})$, to this distribution with some exploration. If there are several equivalent best replies, the player j randomly chooses one of them. Young [2] proved the convergence of the Adaptive Play to an equilibrium when played in self-play for a big class of games such as the coordination and common interest games. APQ [3] is a simple extension of Young's Adaptive Play to the multi-state SG context. To do that, the usual single-agent Q-learning update rule (2) was modified to consider multiple agents as follows:

$$\begin{aligned} \hat{Q}^j(\mathbf{s}, \mathbf{a}) \leftarrow\ & (1-\alpha)\hat{Q}^j(\mathbf{s}, \mathbf{a}) + \alpha[R^j(\mathbf{s}, \mathbf{a}) \\ & + \gamma \max_{a^j \in \pi^j(\mathbf{s}')} U^j(\hat{\Pi}(\mathbf{s}') \cup \{\pi^j(\mathbf{s}')\})] \end{aligned}$$

where j is an agent, $\mathbf{a}$ is a joint action played by the agents in state $\mathbf{s} \in \mathbf{S}$, $\hat{Q}^j(\mathbf{s}, \mathbf{a})$ is the current value for player j of playing the joint action $\mathbf{a}$ in state $\mathbf{s}$, $R^j(\mathbf{s}, \mathbf{a})$ is the immediate reward the player j receives if the joint action $\mathbf{a}$ is played in the state $\mathbf{s}$ and $\pi^j(\mathbf{s}')$ are all possible *pure* strategies that are available for player j.

Having this notation in mind we are now ready to present our approach to the complexity reduction of Q-learning in the stochastic games context.

IV. Q-values Initialization

In our approach we made several important assumptions about the model of the environment. The first assumption is that SGs where agents are intended to act are goal directed with action-penalty representation, i.e. all agents are penalized for any executed action in any state except the goal states. We also assume that multiagent environment applies additional restrictions on the reward and transition functions of the underlying MDP. That is the multiagent penalties for all state-action pairs may be only higher than the corresponding single-agent values and the multiagent transitions in the direction of optimal single-agent actions may be only more uncertain. More formally, we assume that

$$\begin{aligned} R^j(\mathbf{s}, \mathbf{a}) &\leq R^j(s^j, a^j) \\ \forall \mathbf{s} = \langle s^j, \mathbf{s}^{-j} \rangle,\ \mathbf{a} &= \langle a^j, \mathbf{a}^{-j} \rangle \end{aligned} \quad (4)$$

where $\mathbf{s}$ is a multiagent state, $\mathbf{a}$ is a joint action, s^j and a^j correspond to j's position in $\mathbf{s}$ and action in $\mathbf{a}$, $R^j(\mathbf{s}, \mathbf{a})$ is the reward of j when $\mathbf{a}$ is played in $\mathbf{s}$ and $R^j(s^j, a^j)$ is the corresponding single-agent reward. In turn, given the same rewards in multiagent and single-agent cases the multiagent transition function is related to the single-agent one by affecting the utilities as follows:

$$U^j(\Pi(\mathbf{s})) \leq U^j(\overset{\star}{\pi}{}^j(s^j)) \qquad \forall \Pi, \forall \mathbf{s} \quad (5)$$

where $\mathbf{s} = \langle s^j, \mathbf{s}^{-j} \rangle$, $U^j(\Pi(\mathbf{s}))$ is defined using equation (3) and $U^j(\overset{\star}{\pi}{}^j(s^j))$ is defined using equation (1). As is easy to see, the multiagent solution in that case is an appropriate *relaxation* of the multiagent learning problem, by speaking the language of the heuristic search terminology.

As mentioned above, in MDPs it is not evident how to find an informative heuristic function to initialize Q-values with the purpose of reducing the time of the learning. But in many cases in SGs there is such a function: a single-agent solution of the underlying MDP. An MDP may be solved with a

[1] In the SG framework the terms *agent* and *player* mean the same.

variety of techniques (value iteration, reinforcement learning, heuristic search, etc). All these techniques are well known and we leave their description outside this paper. Besides, as soon as the single-agent environment model is much simpler than the multiagent one, we suppose that all agents are able to calculate an optimal single-agent policy before starting to learn in multiagent context.

In order to ensure the tractability of the Q-learning algorithm the Q-values of all state-action pairs must be initialized with some monotonic and admissible function [5]. Let's now define admissibility and monotonicity of Q-values for goal-directed learning.

Definition 1 (monotonicity): Let $\mathcal{G}$ denote the set of the goal states, $\mathcal{G} \subseteq \mathbf{S}$. Q-value $\hat{Q}(s,a)$ is said to be monotonic for the goal directed Q-learning with action-penalty representation if and only if $\forall s, a$

$$\left.\begin{array}{r} 0 \\ R(s,a) + E_{s'}\left[U(\overset{\star}{\pi}(s'))\right] \end{array}\right\} \leq \hat{Q}(s,a) \leq 0 \left\{\begin{array}{l} \text{if } s \in \mathcal{G} \\ \text{if } s \notin \mathcal{G} \end{array}\right.$$

Obviously, the monotonicity property of Q-values corresponds to the consistence of the heuristic function in the heuristic search terminology and means that the triangle inequality holds.

Definition 2 (admissibility): Q-value $\hat{Q}(s,a)$ is said to be admissible for the goal directed Q-learning with action-penalty representation if and only if

$$\left.\begin{array}{r} 0 \\ \overset{\star}{Q}(s,a) \end{array}\right\} \leq \hat{Q}(s,a) \leq 0 \left\{\begin{array}{l} \text{if } s \in \mathcal{G} \\ \text{if } s \notin \mathcal{G} \end{array}\right.$$

where $\overset{\star}{Q}(s,a)$ is the real Q-value.
In turn, admissibility means that for all state-action pairs $-\hat{Q}(s,a)$ never overestimates $-\overset{\star}{Q}(s,a)$. It may be easily verified that uniformly initialized (e.g. zero-initialized) Q-values are monotonic and admissible.

According to our approach multiagent Q-values are initialized by using precalculated single-agent state utilities and single-agent transition function as follows:

$$\hat{Q}^j(\mathbf{s}, \langle a^j, \mathbf{a}^{-j}\rangle) \leftarrow \overset{\star}{Q}^j(s^j, a^j) \;\; \forall \mathbf{a}^{-j} \tag{6}$$

where $\mathbf{s}$ is a multiagent state, s^j is the j's component of the vector $\mathbf{s}$ (in other words, s^j is the agent j's state in the corresponding single-agent world) and $\overset{\star}{Q}^j(s^j, a^j)$ is an optimal single-agent Q-value that is calculated from the single-agent solution and the model as follows:

$$\overset{\star}{Q}^j(s^j, a^j) = R(s^j, a^j) + \gamma \sum_{s'^j} T^j_{s,a,s'} U(\overset{\star}{\pi}^j(s'^j)) \tag{7}$$

where $T^j_{s,a,s'}$ denotes $T(s^j, a^j, s'^j)$, the single-agent transition function, and $U(\overset{\star}{\pi}^j(s^j))$ is the utility of the single-agent state s^j according to the optimal policy $\overset{\star}{\pi}^j$.

Let's now show that in the goal directed stochastic games with action-penalty representation, Q-values initialized with a single-agent solution are admissible and monotonic. To do that, let's prove the following theorem.

Theorem 1: If in a goal directed stochastic game with action-penalty representation, Q-values $\hat{Q}(\mathbf{s}, \mathbf{a})$ are initialized using the utilities of the corresponding single-agent state-action pairs according to equation (6), then these Q-values are admissible and monotonic.
The proof of the above Theorem results from the following two Lemmas.

Lemma 1: If in a goal directed stochastic game with action-penalty representation, Q-values of agent j, $\hat{Q}^j(\mathbf{s}, \mathbf{a})$, are initialized according to equation (6), then these Q-values are monotonic.

Proof: Let $\mathbf{G}$ be the set of multiagent goal states. Obviously, if $\mathbf{s} \in \mathbf{G}$ hence $\hat{Q}^j(\mathbf{s}, \mathbf{a}) = 0 \;\forall \mathbf{a}$. Therefore, we must show that if $\mathbf{s} \notin \mathbf{G}$ then $0 \geq \hat{Q}^j(\mathbf{s}, \mathbf{a}) \geq R^j(\mathbf{s}, \mathbf{a}) + E_{\mathbf{s}'}\left[U^j(\Pi(\mathbf{s}))\right]$. For that let's show that $0 \geq \overset{\star}{Q}^j(s^j, a^j) \geq R^j(\mathbf{s}, \mathbf{a}) + E_{\mathbf{s}'}\left[U^j(\Pi(\mathbf{s}))\right]$. In fact, since the rewards are negative in all states except the goal states, therefore $0 \geq \overset{\star}{Q}^j(s^j, a^j)$. As soon as, according to equation (7), Q-values $\overset{\star}{Q}^j(s^j, a^j)$ are defined as $R^j(s^j, a^j) + E_{s'^j}\left[U^j(\overset{\star}{\pi}^j(s'^j))\right]$ and since inequalities (4) and (5) hold, hence $0 \geq \overset{\star}{Q}^j(s^j, a^j) \geq R^j(\mathbf{s}, \mathbf{a}) + E_{\mathbf{s}'}\left[U^j(\Pi(\mathbf{s}))\right]$ ■

Lemma 2: If in a goal directed stochastic game with action-penalty representation, Q-values of agent j, $\hat{Q}^j(\mathbf{s}, \mathbf{a})$, are initialized according to equation (6), then these Q-values are admissible.

Proof: Let $\mathbf{G}$ be the set of multiagent goal states. Evidently, if $\mathbf{s} \in \mathbf{G}$ therefore $\hat{Q}^j(\mathbf{s}, \mathbf{a}) = 0 \;\forall \mathbf{a}$. Hence, we must demonstrate that if $\mathbf{s} \notin \mathbf{G}$ then $0 \geq \hat{Q}(\mathbf{s}, \mathbf{a}) \geq \overset{\star}{Q}^j(\mathbf{s}, \mathbf{a})$. To do that let's demonstrate that $0 \geq \overset{\star}{Q}^j(s^j, a^j) \geq \overset{\star}{Q}^j(\mathbf{s}, \mathbf{a})$. Since the rewards are negative in all states except the goal states, therefore $0 \geq \overset{\star}{Q}^j(s^j, a^j)$. Since, (i) according to equation (7), Q-values $\overset{\star}{Q}^j(s^j, a^j)$ are defined as $R^j(s^j, a^j) + E_{s'^j}\left[U^j(\overset{\star}{\pi}^j(s'^j))\right]$ and as long as (ii) by definition $\overset{\star}{Q}^j(\mathbf{s}, \mathbf{a}) = R^j(\mathbf{s}, \mathbf{a}) + E_{\mathbf{s}'}\left[U^j(\Pi(\mathbf{s}'))\right]$ and since (iii) inequalities (4) and (5) hold, hence, it follows that $\overset{\star}{Q}^j(s^j, a^j) \geq \overset{\star}{Q}^j(\mathbf{s}, \mathbf{a})$ ■

From the Theorem 1 and by being based on the theoretical results by Koenig and Simmons [5] one can expect that in the multiagent case, if a Q-learning based learning algorithm is used and if it is initialized using an approximative function which has both admissibility and monotonicity properties, the complexity of the learning process will be reduced as compared to zero-initialized (uninformed) case. In the next section, we provide the results of the experiments, produced on several examples of the two-robot grid world problem, which justify this statement. The experiments show also to what extent the learning complexity may be reduced.

V. Experiments

To do our experiments, we programmed a two-robot grid world environment. It may be depicted as presented in Fig. 1. There are two robots on the grid. Each robot j has a set of four available actions, $\mathcal{A}^j = \{N, S, W, E\}$. These actions have stochastic effect. If an action taken is successful, robot changes its position on the grid to the intended cell, otherwise its position remains unchanged. Each action has a negative reward, or penalty, associated with it. In our example we use the reward of -0.04 for any action in any cell except the goal cell where the rewards of all actions are 0. In the case of

collision, no transition is made and both robots obtain the value of -0.1. Thus, robots are interested in attainment of their respective goals by making a minimal number of actions and avoiding collisions. It is easy to see that there are six optimal single-agent trajectories for each robot. In the multiagent case, however, some of these trajectories when used simultaneously can provoke a collision. Hence, a solution of this stochastic game is an equilibrium of two optimal single-agent trajectories when all robots reach their respective goals without collision. Obviously, such an equilibrium is not unique. Thus, this is a goal directed coordination stochastic game with action-penalty representation and, hence, in self-play, APQ initialized with monotonic and admissible Q-values is expected to converge to an equilibrium.

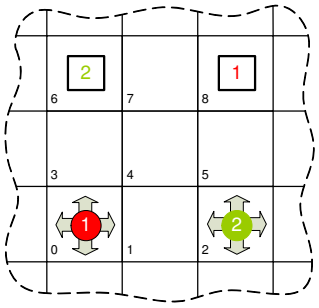

Fig. 1. A fragment of the two-robot grid world environment containing the start and goal positions of agents. The total number of cells in the grid may be arbitrarily big.

We tested our algorithm on this example in a zero-initialized (called "uninformed") case and in a case (called "informed") when Q-values were initialized using single-agent solution, calculated via a simple value iteration. The tests were conducted on a machine with two processors of 2.6 GHz each and 4 GB of RAM. The grids we considered contained 5×5 and 23×23 cells, the both with the same start and goal positions.

Our experiments showed that in the 5×5 environment, uninformed agents ($Q_0^i(\mathbf{s}, \mathbf{a}) = 0,\ \forall \mathbf{a}, \mathbf{s}$) explored all possible 600 states and converged to an equilibrium solution after $450,000$ trials, while informed agents explored about of 570 states and converged to an equilibrium as early as after $250,000$ trials. The convergence curves of the informed and uninformed learning processes in the 5×5 environment are presented in Fig. 3(a).

However more impressive were the results obtained for the bigger environment, 23×23 cells: after $400,000$ trials, uninformed agents explored almost all possible states ($\sim 250,000$) and did not find any equilibrium solution, while informed agents were able to converge to an equilibrium solution after this number of learning trials and explored merely about of $10,000$ states ($\sim 4\%$).

It is significant that in order to observe the behavior of our algorithm with the different initial values of the learning rate, α_0, and to find an optimal one, if such exists, we ran the algorithm with α_0 varying within the range $[0.1, 0.9]$. Interestingly that the number of states explored during the learning process grown uniformly with the growth of α_0, but the number of trials, required to discover an optimal solution, decreased to some minimum value up to $\alpha_0 \approx 0.6$, and started to grow rapidly thereupon (Fig. 3(b)). Therefore, there is an optimal value of α_0, with which the number of trials required to explore a solution is minimal.

Finally, to show that the multiagent solution may differ substantially from the initial heuristic (i.e., from the pair of optimal single-agent trajectories calculated via value iteration by each agent), we investigated the following interesting case. We set the reward of -0.15 for both agents if they reached their respective goals non-synchronously. (Notice, that in this case, a pair of two single-agent solutions cannot in general be an appropriate multiagent solution because of the action failures and the mistiming they can provoke). We observed that in the case of occasional mistiming, if there was no wall near the goal cell (Fig. 2 (a)), the agents did not attempt to synchronize, because it would require one agent to stop and "wait" another one, but there was no action "wait" in the agents' set of actions. However, if there was a wall nearby (Fig. 2 (b)), the agent that was ahead decided to hit the wall once to stay put and "wait" another agent for one step.

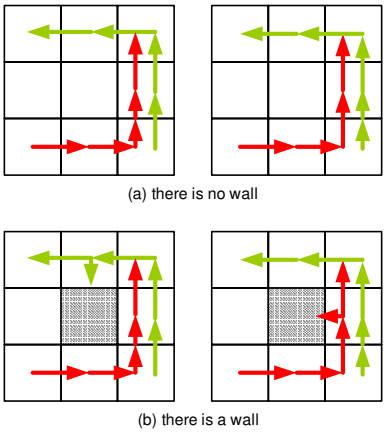

Fig. 2. Mistiming case equilibria. The left and right images are a cases where the agents 1 and 2 respectively are retarded.

VI. Related Work

To our knowledge, the question of initial approximation of Q-values in the multiagent learning context was not widely explored in the literature. In the single-agent case there is a remarkable example of study of the complexity of single-agent Q-learning with a comparison of heuristically initialized and zero-initialized cases by Koenig and Simmons [5]. As regards the learning component, the extensive studies have been made. Relatively to our approach, Sen et al. [6] and Tan [7] studied an application of single-agent Q-learning to multiagent tasks without taking into account the opponents' strategies. They showed that if the other agents' policies are stationary then the learning agent will converge to some stationary policy as well. Claus and Boutilier [8] studied the case of coordination repeated

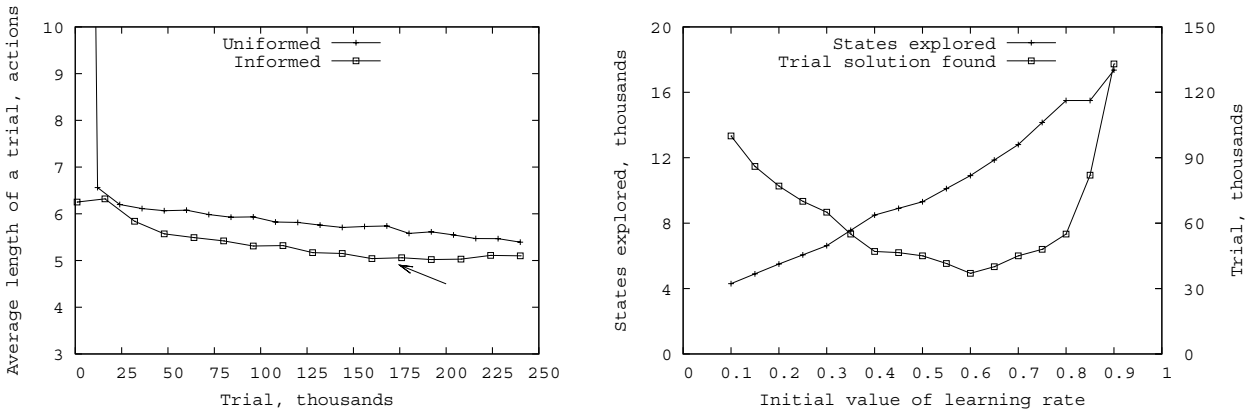

Fig. 3. (a) reflects the dynamics of learning in the grid 5×5. The curves show the average length of a learning trial as a function of the number of trials. The arrow points to a trial, starting with which the informed agents found a solution, but the uninformed ones did not. (b) represents the number of states explored and the trial, in which an equilibrium was found, as a functions of the initial value of α.

games with the opponent modeling via fictitious play. They showed empirically the convergence of Q-learning in that case. Gies and Chaib-draa [3] studied an application of the Adaptive Play to the uninformed multiagent Q-learning in coordination stochastic game context and showed empirically that the agents' policies converged to an equilibrium in pure strategies. Hu and Wellman [9], as well as Littman [10], proposed their approaches, which, in each state, explicitly calculate a Nash equilibrium of the matrix games composed of the Q-values in these states. However, all these methods suffer the low scalability, as well as other multiagent learning methods that restrict neither state space nor joint-action space of the problem during the learning process.

VII. Conclusions and Future Work

In this paper, we proposed an approach to multiagent learning, which uses an initial heuristic approximation of agents' preferences. In our approach, an optimal single-agent solution was used as such a heuristic. We showed that the initialization of multiagent Q-values using a precalculated single-agent solution permits significantly reducing the complexity of the learning process. We also showed that such an initialization is admissible and monotonic for the problems that can be modeled as a goal-directed stochastic game with action-penalty representation. By producing a set of empirical tests on the multiagent coordination problem we showed that the uninformed multiagent learning quickly becomes intractable, while the informed, heuristically initialized, algorithm remains tractable with growth of state space while being weakly sensible to that growth due to the strict focusing on the relevant states only.

It is important to note that the real life robotic tasks are concerned with a set of particularities, such as noisy sensors, continuous state space and inter-state transitions, limited observability of the other robots' actions, and so on. The stochastic game framework, just as it is, is not able to represent these particularities. While understanding a limited applicability of the learning algorithms created for the SGs, the principles proposed in this paper are, in our opinion, of a great interest for the robotics and, we believe, can be adopted to the more complex environments.

In our future work we intend to extend the applicability of our approach to the general form stochastic games. For that, a suitable *relaxation* should be derived from the general multiagent model in such a way that a solution of such a relaxed model was an admissible and monotonic approximation of the original model and was easier to be calculated as compared to the solution of the original problem. The advantages and limitations of practical applications of this approach to real life problems should also to be investigated.

References

[1] A. Burkov and B. Chaib-draa, "Reducing the complexity of multiagent learning," in *In proceedings of the 2007 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'07)*, Honolulu, Hawai'i, 2007, poster.

[2] H. Young, "The evolution of conventions," *Econometrica*, vol. 61(1), pp. 57–84, 1993.

[3] O. Gies and B. Chaib-draa, "Apprentissage de la coordination multi-agent : une méthode basée sur le Q-learning par jeu adaptatif," *Revue d'Intelligence Artificielle*, vol. 20(2-3), pp. 385–412, 2006.

[4] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8(3), pp. 279–292, 1992.

[5] S. Koenig and R. G. Simmons, "The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms," *Machine Learning*, vol. 22, pp. 227–250, 1996.

[6] S. Sen, M. Sekaran, and J. Hale, "Learning to coordinate without sharing information," in *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94)*, Seattle, Washington, 1994, pp. 426–431.

[7] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the Tenth International Conference on Machine Learning (ICML'93)*, Amherst, MA, 1993, pp. 330–337.

[8] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*. Menlo Park, CA: AAAI Press, 1998.

[9] J. Hu and P. Wellman, "Multiagent reinforcement learning: Theoretical framework and an algorithm." in *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*. San Francisco, CA: Morgan Kaufmann, 1998, pp. 242–250.

[10] M. Littman, "Friend-or-foe Q-learning in general-sum games," in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01)*, San Francisco, CA, 2001, p. Morgan Kaufman.