Learning to Act in Continuous Dec-POMDPs

Jilles S. Dibangoye[1] Olivier Buffet[2]

[1] INSA de Lyon, laboratoire CITI, INRIA
[2] LORIA, INRIA

prénom.nom@inria.fr

Résumé

Nous nous attaquons au problème d'apprentissage par renforcement dans le cadre des processus décisionnels de Markov partiellement observables et décentralisés. Les tentatives précédentes ont conduit à différentes variantes de la méthode généralisée d'itération de politiques, qui dans le meilleur des cas abouties à des optima locaux. Dans ce papier, nous nous restreindrons au plans, qui sont des formes plus simples que des politiques. Nous dériverons, sous certaines conditions, le premier algorithm algorithme optimal d'apprentissage par renforcement coopératif. Afin d'accroître le passage à l'échelle de cet algorithme, nous remplacerons l'opérateur glouton traditionnel par un programme linéaire en nombre entier. Les résultats expérimentaux montrent que notre méthode est capable d'apprendre de façon optimale dans plusieurs bancs de test de la litérature.

Mots Clef

Processes décisionnels de de Markov partiellement observables et décentralisés, Apprentissage par Renforcement.

Abstract

We address a long-standing open problem of reinforcement learning in continuous decentralized partially observable Markov decision processes. Previous attempts focussed on different forms of generalized policy iteration, which at best led to local optima. In this paper, we restrict attention to plans, which are simpler to store and update than policies. We derive, under mild conditions, the first optimal cooperative multi-agent reinforcement learning algorithm. To achieve significant scalability gains, we replace the greedy maximization by mixed-integer linear programming. Experiments show our approach can learn to act optimally in many finite domains from the literature.

Keywords

Decentralized Markov Decision Partially Observable Processes, Reinforcement Learning.

1 Introduction

Decentralized partially observable Markov decision processes (Dec-POMDPs) emerged as the standard framework for sequential decision making by a team of collaborative agents Bernstein et al. [2000]. A key assumption of Dec-POMDPs is that agents can neither see the actual state of the system nor explicitly communicate their noisy observations with each other due to communication cost, latency or noise, hence providing a partial explanation of the double exponential growth at every control interval of the required memory in optimal algorithms Hansen et al. [2004], Szer et al. [2005], Oliehoek et al. [2008, 2013], Dibangoye et al. [2016]. While planning methods for finite Dec-POMDPs made substantial progress in recent years, the formal treatment of the corresponding reinforcement learning problems received little attention so far, let alone its continuous counterpart. The literature of multi-agent reinforcement learning (MARL) can be divided into two main categories: *concurrent* and *team approaches* Tan [1998], Panait and Luke [2005].

Perhaps the dominant paradigm in MARL is the concurrent approach, which involves multiple simultaneous learners: typically, each agent has its learning process. Self-interested learners, for example, determine their best-response behaviors considering their opponents are part of the environment, often resulting in local optima Brown [1951], Hu and Wellman [1998], Littman [1994]. While concurrent learning can apply in Dec-POMDPs, a local optimum may lead to severely suboptimal performances Peshkin et al. [2000], Zhang and Lesser [2011], Kraemer and Banerjee [2016]. Also, methods of this family face two conceptual issues that limit their applicability. The primary concern is that of the co-adaptation dilemma, which arises when each attempt to modify an agent behavior can ruin learned behaviors of its teammates. Another major problem is that of the multi-agent credit assignment, that is, how to split the collective reward signal among independent learners.

Alternatively, the team approach involves a single learner acting on behalf of all agents to discover a collective solution Salustowicz et al. [1998], Miconi [2003]. Interestingly, this approach circumvents the difficulties arising from both the co-adaptation and the multi-agent credit assignment.

Coordinated agents, for example, simultaneously learn their control choices and the other agent strategies assuming instantaneous and free explicit communications Guestrin et al. [2002], Kok and Vlassis [2004]. While methods of this family inherit from standard single-agent techniques, they need to circumvent two significant drawbacks: the explosion in the state space size; and the centralization of all learning resources in a single place. Recently, team algorithms ranging from Q-learning to policy-search have been introduced for finite Dec-POMDPs, but with no guaranteed global optimality Kraemer and Banerjee [2016], Wu et al. [2013], Liu et al. [2015, 2016]. So, it seems one can either compute local optima with arbitrary bad performances or calculate optimal solutions but assuming noise-free, instantaneous and explicit communications.

A recent approach to optimally solving finite Dec-POMDPs suggests recasting them into occupancy-state MDPs (oMDPs) and then applying (PO)MDP solution methods Dibangoye et al. [2016]. In these oMDPs, the states called occupancy states are distributions over hidden states and joint histories of the original problem, and actions called decision rules are mappings from joint histories to controls Nayyar et al. [2011], Oliehoek [2013], Dibangoye et al. [2016]. This approach achieves scalability gains by exploiting the piece-wise linearity and convexity of the optimal value function. Since this methodology successfully applied for planning in finite Dec-POMDPs, it is natural to wonder which benefits it could bring to the corresponding MARL problem. Unfortunately, a straightforward application of standard RL methods to oMDPs will face three severe limitations. First, occupancy states are unknown, and hence must be estimated. Second, they lie in a continuum making tabular RL methods inapplicable. Finally, the greedy maximization is computationally demanding in decentralized stochastic control problems Radner [1962], Kumar and Zilberstein [2009].

This paper extends the methodology of Dibangoye et al. to MARL, focussing on the three major issues that limit its applicability. Our primary result is the proof that, by restricting attention to plans instead of policies, a linear function over occupancy states and decision rules, which is simple to store and update, can capture the optimal performance for Dec-POMDPs. We further use plans instead of policies in a policy iteration algorithm, with the plan always being improved with respect to a linear function and a linear function always being driven toward the linear function for the plan. Under accurate estimation of the occupancy states, the resulting algorithm, called occupancy-state SARSA (oSARSA) Rummery, G. A. and Niranjan [1994], is guaranteed to converge with probability one to an optimal plan for any finite Dec-POMDP. To extend its applicability to higher-dimensional domains, oSARSA replaces the greedy (or soft) maximization by a mixed-integer linear program for finite settings, and a gradient approach for continuous ones. Altogether, we obtain a MARL algorithm that can apply to continuous Dec-POMDPs. Experiments show our approach can learn to act optimally in many finite domains from the literature.

We organize the remainder of this paper as follows. Section 2 extends a recent planning theory from finite to continuous settings, starting with a formal definition of finite and continuous Dec-POMDPs. We proceed with the introduction of a framework for centralized MARL in Dec-POMDPs in Section 3. Also, we discuss our solutions to the three limitations mentioned above. We present the resulting algorithm oSARSA along with convergence guarantees in Section 4. Finally, we conduct experiments in Section 5, demonstrating our approach can learn to act optimally in many finite domains from the literature.

2 Planning in Dec-POMDPs as oMDPs

2.1 Continuous Dec-POMDPs

A continuous Dec-POMDP is a tuple $M \doteq (n, X, \{U^i\}, \{Z^i\}, p, r, \ell, \gamma, b_0)$, where n denotes the number of agents involved in the decentralized stochastic control process; X is a set of hidden world states, denoted x or y; U^i is a private control set of agent $i \in [\![1; n]\!]$, where $U = U^1 \times \cdots \times U^n$ specifies the set of controls $u = (u^1, \ldots, u^n)$; Z^i is a private observation set of agent i, where $Z = Z^1 \times \cdots \times Z^n$ specifies the set of observations $z = (z^1, \ldots, z^n)$; p describes a transition probability kernel with conditional density $p^{u,z}(x, y)$; r is a reward model with immediate reward $r(x, u)$, we assume rewards are two-side bounded, i.e., for some $c \in \mathbb{R}^+$, $\forall x \in X, u \in U \colon |r(x, u)| \leqslant c$; ℓ is the planning horizon; $\gamma \in [0, 1]$ denotes the discount factor; and b_0 is the initial belief state with density $b_0(x_0)$. We shall restrict attention to finite planning horizon $\ell < \infty$ since an infinite planning horizon solution is within a small scalar $\epsilon > 0$ of a finite horizon solution where $\ell = \lceil \log_\gamma((1-\gamma)\epsilon/c) \rceil$. A Dec-POMDP is usually either *finite*, i.e., sets X, U^i and Z^i for all $i \in [\![1; n]\!]$ are finite, or *continuous*, i.e., the same sets are continuous. The remainder of this section extends concepts and standard results from finite to continuous settings.

Because we are interested in MARL, we assume an incomplete knowledge about M, *i.e.,* p and r are either unavailable or only through a generative model. Hence, the goal of solving M is to find a joint plan, *i.e.,* a tuple of individual decision rules, one for each agent and time step: $\rho \doteq (a^1_{0:\ell}, \ldots, a^n_{0:\ell})$. A tth individual decision rule $a^i_t \colon O^i_t \mapsto \mathcal{P}(U^i)$ of agent i prescribes private controls based on the whole information available to the agent up to the tth time step, *i.e.,* history of controls and observations $o^i_t = (u^i_{0:t-1}, z^i_{1:t})$, where $o^i_0 = \varnothing$ and $o^i_t \in O^i_t$. A tth joint decision rule, denoted $a_t \colon O_t \mapsto \mathcal{P}(U)$, can be specified as $a_t(u|o) \doteq \prod_{i=1}^n a^i_t(u^i|o^i)$, where $O_t \doteq O^1_t \times \cdots \times O^n_t$, $o^i \in O^i_t$ and $o \doteq (o^1, \ldots, o^n) \in O_t$. From control interval t onward, agents collectively receive discounted cumulative rewards, denoted by random variable $R_t \doteq \gamma_1 r_t + \cdots + \gamma_\ell r_\ell$, where α_t denotes the time-step dependent weighting factors, often set to $\gamma_t = \gamma^t$ for discounted problems or $\gamma_t = \frac{1}{\ell}$ for the average reward case. For any control interval t, joint plans $a_{0:t}$ of interest are those that achieve the highest performance measure $J(a_{0:t}) \doteq \mathbb{E}^{a_{0:t}}\left\{R_0 \mid b_0\right\}$ starting at b_0, where $\mathbb{E}^{a_{0:t}}\{\cdot\}$ denotes the expectation with respect to the probability distribution over state-action pairs joint plan $a_{0:t}$ induces, in particular $J(\rho) \doteq J(a_{0:\ell-1})$ for $\rho \doteq a_{0:\ell-1}$. One can show that, in Dec-POMDPs, there always exists a deterministic plan that is as good as any stochastic plan [see Puterman, 1994, Lemma 4.3.1]. Unfortunately, there is no direct way to apply the theory developed for Markov decision processes Bellman [1957], Puterman [1994] to Dec-POMDPs, including: the Bellman optimality equation; or the policy improvement theorem. To overcome these limitations, we rely on a recent theory by Dibangoye et al. that recasts M into an MDP, thereby allowing knowledge transfer from the MDP setting to Dec-POMDPs.

2.2 Occupancy-State MDPs

To overcome the fact that agents can neither see the actual state of the system nor explicitly communicate their noisy observations with each other, Szer et al. (2005) and later on Dibangoye et al. (2016) suggest formalizing M from the perspective of a centralized algorithm. A centralized algorithm acts on behalf of the agents by selecting a joint decision rule to be executed at each control interval based on all data available about the system, namely the information state. The information state at the end of control interval t, denoted $\iota_{t+1} \doteq (b_0, a_{0:t})$, is a sequence of joint decision rules the centralized algorithm selected starting at the initial belief state. Hence, the information state satisfies the following recursion: $\iota_0 \doteq (b_0)$ and $\iota_{t+1} \doteq (\iota_t, a_t)$ for all control interval t, resulting in an ever-growing sequence. To generalize the value from one information state to another one, Dibangoye et al. introduced the concept of occupancy states. The occupancy state at control interval t, denoted $s_t \doteq \mathbb{P}(x_t, o_t | \iota_t)$, is a distribution over hidden states and joint histories conditional on information state ι_t at control interval t. Interestingly, the occupancy state has many important properties. First, it is a sufficient statistic of the information state when estimating the (current and future) reward to be gained by executing a joint decision rule:

$$R(s_t, a_t) \doteq \textstyle\int_X \int_O s_t(x, o) \int_U a_t(u|o) \cdot r(x, u) \mathrm{d}u \mathrm{d}o \mathrm{d}x.$$

In addition, it describes a deterministic and fully observable Markov decision process, where the next occupancy state depends only on the current occupancy state and next joint decision rule, for all $y \in X, o \in O, u \in U, z \in Z$:

$$T(s_t, a_t) \doteq s_{t+1}$$
$$s_{t+1}(y, (o, u, z)) \doteq a_t(u|o) \textstyle\int_X s_t(x, o) \cdot p^{u,z}(x, y) \mathrm{d}x.$$

The process the occupancy states describe is known as the occupancy-state Markov decision process (oMDP), and denoted $M' \doteq (S, A, R, T, \ell, \gamma, s_0)$. This is an ℓ-steps deterministic and continuous MDP with respect to M, where $S \doteq \cup_{t \in [\![0;\ell-1]\!]} S_t$ is the set of occupancy states up to control interval $\ell - 1$; $A \doteq \cup_{t \in [\![0;\ell-1]\!]} A_t$ is the set of joint decision rules up to control interval $\ell - 1$; R is the reward model; and T is the transition rule; s_0 is the initial occupancy state, which is essentially the initial belief in M; γ and ℓ are as in M. It is worth noticing that there is no need to construct explicitly M'; instead we use M (when available) as a generative model for the occupancy states $T(s_t, a_t)$ and rewards $R(s_t, a_t)$, for all control intervals t.
Note that, using control terminology, planning (or learning) in M' can be made either open-loop or closed-loop. M is called an open-loop planning problem because the class of considered solutions (*i.e.,* plans) are only function of time (and not of the underlying occupancy states). It is called a closed-loop planning problem since it is a Markov decision process, whose solution is known to be a policy (mapping from occupancy states to joint decision rules). Policies $\pi \colon S \mapsto A$, mappings from occupancy states to decision rules, generalizes plans ρ, by prescribing a joint decision rule depending on the current time and occupancy state. Plans are in general sub-optimal compared to policies, as they prescribes a decision rule depending only upon the current time. However, here, both open- and closed-loop approaches can lead to an optimal solution. Below, we review a closed-loop approach based on the dynamic programming theory Bellman [1957].
Due to the deterministic nature of the dynamics in oMDP M', planning (or learning) in M' can be seen, without loss of optimality, not only as closed-loop—a solution being a policy $\pi \colon S \mapsto A$, mapping (encountered/reachable) occupancy states to joint decision rules—, but also as open-loop—a solution being a(n inconditional) plan ρ, that is, a sequence of joint decision rules. Below, we review a closed-loop approach based on the dynamic programming theory Bellman [1957].
For any finite M, the Bellman equation is written as follows: for all occupancy state $s_t \in S_t$, and some fixed policy π,

$$V_t^\pi(s_t) \doteq R(s_t, \pi(s_t)) + \gamma_1 V_{t+1}^\pi(T(s_t, \pi(s_t))) \quad (1)$$

with boundary condition $V_\ell^\pi(\cdot) \doteq 0$, describes the return of a particular occupancy state s_t when taking decision rule $a_t = \pi(s_t)$ prescribed by π. The equation for an optimal policy π^* is referred to as the Bellman optimality equation: for any control interval t, and occupancy state s_t,

$$V_t^*(s_t) \doteq \max_{a_t \in A} R(s_t, a_t) + \gamma_1 V_{t+1}^*(T(s_t, a_t)) \quad (2)$$

with boundary condition $V_\ell^*(\cdot) \doteq 0$. Unfortunately, occupancy states lie in a continuum, which makes exact dynamic programming methods infeasible. When optimized exactly, the value function solution of (2) along with the boundary condition is always piece-wise linear and convex in the occupancy-state space Dibangoye et al. [2016]. Similarly to continuous POMDPs Porta et al. [2006], we generalize this property to continuous Dec-POMDPs.

Lemma 1. *For any arbitrary M', the solution $V^*_{0:\ell}$ of (2) is convex in the occupancy-state space. If we restrict attention to deterministic policies and finite M (and corresponding M'), the solution of (2) is piece-wise linear and convex in the occupancy-state space. Hence, the optimal value at any occupancy state s_t is as follows:*

$$V^*_t(s_t) \doteq \max_{\alpha_t \in \Gamma_t} \langle s_t, \alpha_t \rangle, \tag{3}$$

where $\langle s_t, \alpha_t \rangle$ is used to express the expectation of a linear function α_t (also called α-function[1]) in the probability space defined by sample space $X \times O$, the σ-algebra $X \times O$ and the probability distribution s_t; and Γ_t is the set of all tth α-functions.

Proof. Consider the α-function induced when agents follow plan $a_{t:\ell-1}$ from control interval t onwards, denoted $\alpha^{a_{t:\ell-1}}$ and given by $\alpha^{a_{t:\ell-1}}(x, o) \doteq \mathbb{E}\{R_t | x_t = x, o_t = o, a_{t:\ell-1}\}$. Hence, the optimal value starting at any occupancy state s_t is given by taking the maximum over values of all possible plans from control interval t onwards: $V^*_t(s_t) = \max_{a_{t:\ell-1}} \langle s_t, \alpha^{a_{t:\ell-1}} \rangle$. In addition, the linearity of the expectation also implies that $\alpha^{a_{t:\ell-1}}$ is linear in the occupancy-state space. The proof directly follows from [Rockafellar, 1970, Theorem 5.5]. □

Lemma 1 shows that for any arbitrary M and corresponding M', the solution of (2), represented by sets $\Gamma_{0:\ell}$, is convex in the occupancy-state space. Each α-function defines the value function over a bounded region of the occupancy-state space. In addition, it is associated with a plan, defining the optimal plan for a bounded region of the occupancy-state space. Sets $\Gamma_{0:\ell}$ are iteratively improved by adding a new α-function that dominates current ones over certain regions of the occupancy-state space. The α-function to be added is computed using point-based Bellman backup operator $\mathbb{H}$:

$$[\mathbb{H}\Gamma_{t+1}](s_t) = \operatorname*{arg\,max}_{\alpha^a_t \colon a \in A_t, \alpha_{t+1} \in \Gamma_{t+1}} \langle s_t, \alpha^a_t \rangle,$$

where $\alpha^a_t(x, o) \doteq \mathbb{E}\{r(x, u) + \gamma_1 \alpha_{t+1}(y, (o, u, z)) | a\}$, for each hidden state $x \in X$, and joint history $o \in O$. Interestingly, $\mathbb{H}$ remains both a contracting and isotonic operator in continuous settings Porta et al. [2006]. To keep the number of α-functions manageable, one can prune those that are dominated over the entire occupancy-state space. All in all, the oMDP reformulation permits us to solve finite M by means of M' using near-optimal planning methods leveraging on the special structure of the optimal value function Shani et al. [2013]. This methodology results in the current state-of-the-art algorithm to optimally solving finite Dec-POMDPs Dibangoye et al. [2016]. So it seems natural to wonder if the same methodology can also succeed when applied to the corresponding reinforcement-learning problem. In other words, how can a centralized algorithm learn to coordinate a team of agents with possibly contradicting perceptual information?

[1] In continuous M, α-functions represent linear functions (including parametric ones); whereas in finite M, α-functions become α-vectors, that is finite-dimensional vectors.

3 Learning in Dec-POMDPs as *o*MDPs

Using the oMDP reformulation, a natural approach to achieve centralized RL for decentralized stochastic control suggests applying exact RL methods. In the Q-learning algorithm Watkins and Dayan [1992], for example, one would learn directly the Q-value function when following a fixed policy π: for any control interval $t \in [\![0; \ell-1]\!]$,

$$Q^\pi_t(s_t, a_t) \doteq R(s_t, a_t) + \gamma_1 V^\pi_{t+1}(T(s_t, a_t)) \tag{4}$$

with boundary condition $Q^\pi_\ell(\cdot, \cdot) = 0$. The policy improvement theorem provides a procedure to change a sub-optimal policy π into an improved one $\bar{\pi}$ Howard [1960]: for any control interval $t \in [\![0; \ell-1]\!]$,

$$\bar{\pi}(s_t) \doteq \arg\max_{a_t \in A_t} Q^\pi_t(s_t, a_t). \tag{5}$$

Unfortunately, this approach has three severe limitations. First, the occupancy states are unknown and must be estimated. Second, even if we assume a complete knowledge of the occupancy states, they lie in a continuum, which precludes exact RL methods to accurately predict α-functions even in the limit of infinite time and data. Finally, the greedy maximization required to improve the value function proved to be NP-hard in finite settings and problematic in continuous ones Radner [1962], Kumar and Zilberstein [2009].

3.1 Addressing Estimation Issues

Although mappings T and R in M' are unknown to either agents or a centralized algorithm, one can instead estimate on the fly both $T(s_0, a_{0:t-1})$ and $R(T(s_0, a_{0:t-1}), a_t)$ for some fixed plan $\rho \doteq a_{0:\ell-1}$ through successive interactions of agents with the environment. To this end, we shall distinguish between two settings. The first one assumes a generative model is available during the centralized learning phase, e.g. a black box simulator; and the second does not. In both cases, we build on the concept of replay pool Mnih et al. [2015], except that we extend it from stationary single-agent domains to non-stationary multi-agent domains.

If a generative model is available during the learning phase, then a Monte Carlo method can approximate $T(s_0, a_{0:t-1})$ and $R(T(s_0, a_{0:t-1}), a_t)$ arbitrary closely. To this end, the generative model allows the agents to sample experiences generated from M. A ℓ-steps experience is a 4-tuple $\xi \doteq (x_{0:\ell-1}, u_{0:\ell-1}, r_{0:\ell-1}, z_{1:\ell})$, where $x_{0:\ell-1}$ are sampled hidden states, $u_{0:\ell-1}$ are controls made, $r_{0:\ell-1}$ are reward signals drawn from the reward model, and $z_{1:\ell}$ are the resulting observations, drawn from the dynamics model. If we let $\mathcal{D}^\rho \doteq \{\xi^{[i]}\}_{i \in [\![1:K]\!]}$ be the replay pool of K i.i.d random samples created through successive interactions with the generative model, then empirical occupancy state $\hat{s}_t \approx T(s_0, a_{0:t-1})$ and reward $\hat{R}_t \approx R(T(s_0, a_{0:t-1}), a_t)$ corresponding to the current $\mathcal{D}^\rho$ are given by: for any control interval $t \in [\![0 : \ell-1]\!]$,

$$\hat{s}_t(x, o) \doteq \tfrac{1}{K} \textstyle\sum_{i=1}^K \delta_x(x^{[i]}_t) \cdot \delta_o(u^{[i]}_{0:t}, z^{[i]}_{1:t}) \tag{6}$$

$$\text{and} \quad \hat{R}_t \doteq \tfrac{1}{K} \textstyle\sum_{i=1}^K r^{[i]}_t, \tag{7}$$

where $\delta_x(\cdot)$ and $\delta_o(\cdot)$ denote the delta-Dirac mass located in hidden state and joint history pair, respectively. By the law of large numbers the sequence of averages of these estimates converges to their expected values, and the standard-deviation of its error falls as $1/\sqrt{K}$ [Sutton and Barto, 1998, chapter 5]. The error introduced by Monte Carlo when estimating $T(\hat{s}_{t-1}, a_{t-1})$ instead of $T(s_0, a_{0:t-1})$ is upper bounded by $2\ell/\sqrt{K}$. The proof follows from the performance guarantee of the policy-search algorithm by Bagnell et al. [2004]. Hence, to ensure the learned value function is within $\epsilon > 0$ of the optimal one, one should set the replay-pool size to $K = O(4\frac{\ell^2}{\epsilon^2})$.

When no generative model is available, the best we can do is to store samples agents collected during the learning phase into replay pools $\mathcal{D}^\rho$, one experience for each episode within the limit size of K. We maintain only the K recent experiences, and may discard[2] hidden states since they are unnecessary for the updates of future replay pools and the performance measure. The rationale behind this approach is that it achieves the same performances as a Monte Carlo method for the task of approximating $T(s_0, a_{0:t-1})$ and $R(T(s_0, a_{0:t-1}), a_t)$ given a fixed plan $\rho \doteq a_{0:\ell-1}$. In fact, if we let $\mathcal{D}^\rho$ be a replay pool of K i.i.d. samples generated according to ρ, the empirical occupancy state $\hat{s}_t \approx T(s_0, a_{0:t-1})$ and reward $\hat{R}_t \approx R(T(s_0, a_{0:t-1}), a_t)$ corresponding to $\mathcal{D}^\rho$ are given by (6) and (7), respectively. One can further show this approach preserves performance guarantees similar to those obtained when using a generative model.

3.2 Addressing Prediction Issues

The key issue with large (possibly continuous) spaces of occupancy states and decision rules is that of generalization, that is, how experiences with a limited subset of occupancy states and decision rules can produce a good approximation over a much larger space. Fortunately, a fundamental property of *o*MDPs is the convexity of the optimal value function over the occupancy-state space, see Lemma 1. Building on this property, we demonstrate a simple yet important preliminary result before stating the main result of this section.

Lemma 2. *For any arbitrary M' (resp. M), the optimal Q-value function is the upper enveloppe of sets $\Omega^*_{0:\ell}$ of α-functions over occupancy states and joint decision rules: for any control interval t, $Q^*_t(s_t, a_t) = \max_{q_t \in \Omega^*_t} \langle s_\tau \odot a_\tau, q_t \rangle$, where $q_t \in \Omega^*_t$ are appropriate α-functions, and $s_\tau \odot a_\tau$ denotes the Hadamard product*[3].

Proof. We proceed by induction to prove this property. In the following we assume that all operations (e.g. integrals) are well-defined in the corresponding spaces. For control interval $t = \ell - 1$, we only have to take into account the immediate reward and, thus, we have that $Q^*_{\ell-1}(s_{\ell-1}, a_{\ell-1}) = R(s_{\ell-1}, a_{\ell-1})$. Therefore, if we define the set $\Omega^*_{\ell-1} = \{q_{\ell-1}\}$, where $q_{\ell-1}(x, o, u) \doteq r(x, u)$, the property holds at control interval $t = \ell - 1$. We now assume the property holds for control interval $\tau + 1$ and we show that it also holds for control interval τ. Using (2) and (4), we have that, $Q^*_\tau(s_\tau, a_\tau) = R(s_\tau, a_\tau) + \gamma_1 \max_{a_{\tau+1}} Q^*_{\tau+1}(T(s_\tau, a_\tau), a_{\tau+1})$, and by the induction hypothesis, let $s_{\tau+1} \doteq T(s_\tau, a_\tau)$:

$$Q^*_{\tau+1}(s_{\tau+1}, a_{\tau+1}) = \max_{q \in \Omega^*_{\tau+1}} \textstyle\int_{X,O,U} s_\tau(x, o) a_\tau(u|o) \int_{X,Z,U} p^{u,z}(x, y) a_{\tau+1}(u'|o, u, z) \, q(y, (o, u, z), u') \mathrm{d}u \mathrm{d}o \mathrm{d}x \mathrm{d}y \mathrm{d}z \mathrm{d}u'.$$

With the above,

$$Q^*_\tau(s_\tau, a_\tau) = \max_{a \in A_{\tau+1}, q \in \Omega^*_{\tau+1}} \textstyle\int_{X,O,U} s_\tau(x, o) a_\tau(u|o) [r(x, u) + \gamma_1 \int_{Y,Z,U} p^{u,z}(x, y) a(u'|o, u, z) \, q(y, (o, u, z), u') \mathrm{d}y \mathrm{d}z \mathrm{d}u'] \mathrm{d}u \mathrm{d}o \mathrm{d}x.$$

At this point, we can define the bracketed quantity as

$$q^{a_{\tau+1}}(x, o, u) \doteq r(x, u) + \gamma_1 \textstyle\int_{X,Z,U} p^{u,z}(x, y) \, a_{\tau+1}(u'|o, u, z) q(y, (o, u, z), u') \mathrm{d}y \mathrm{d}z \mathrm{d}u'.$$

Note that α-function $q^{a_{\tau+1}}$ is independent of occupancy state s_τ and decision rule a_τ for which we are computing Q^*_τ. With this, we have that $Q^*_\tau(s_\tau, a_\tau) = \max_{q^a \colon a \in A_{\tau+1}, q \in \Omega^*_{\tau+1}} \langle s_\tau \odot a_\tau, q^a \rangle$ and, thus the lemma holds. □

Lemma 2 generalizes the convexity property demonstrated in Lemma 1 from optimal value functions over occupancy states to optimal value functions over occupancy states and decision rules. As a consequence, finite sets $\Omega^*_{0:\ell-1}$ of α-functions can produce solutions arbitrarily close to the optimal Q-value function $Q^*_{0:\ell-1}$. Though Q-value function $Q^*_{0:\ell-1}$ generalizes from a pair of occupancy state and decision rule to another one, storing and updating a convex hull is non trivial. Instead of learning the optimal Q-value function over all occupancy states and decision rules, we explore a simpler yet tractable alternative, which will prove sufficient to preserve ability to eventually find an optimal plan starting at initial occupancy state s_0.

Theorem 1. *For any arbitrary M' (resp. M), the Q-value function $Q^{\rho^*}_{0:\ell-1}$ under an optimal plan $\rho^* \doteq a^*_{0:\ell-1}$ starting at initial occupancy state s_0 is linear in occupancy states and decision rules: $Q^{\rho^*}_t(s_t, a_t) = \langle s_t \odot a_t, q^{\rho^*}_t \rangle$ where $q^{\rho^*}_t \doteq \arg\max_{q_t \in \Omega^*_t} \langle T(s_0, a^*_{0:t-1}) \odot a^*_t, q_t \rangle$.*

Proof. The proof derives directly from Lemma 2. First, notice that any arbitrary non-dominated joint plan ρ induces a sequence of α-functions $q^\rho_{0:\ell-1}$ stored in $\Omega^*_{0:\ell-1}$, which proves the Q-value function under a fixed plan is linear over occupancy states and joint decision rules. In addition, each α-function $q^\rho_t \in \Omega^*_t$ describes the expected

[2] Note that one should keep hidden states when available since they often speed up the convergence.

[3] $\forall (x, o, u) \colon [s_\tau \odot a_\tau](x, o, u) \doteq s_\tau(x, o) \cdot a_\tau(u|o)$.

returns from $t \in [\![0;\ell-1]\!]$ onward, when agents follow non-dominated joint plan ρ. If we let ρ^* be a greedy joint plan with respect to $Q^*_{0:\ell-1}$, then $q^{\rho^*}_{0:\ell-1}$ is maximal along $\{T(s_0, a^*_{0:t-1})\}_{t\in[\![0;\ell-1]\!]}$. $\square$

Theorem 1 proves that the Q-function for a given optimal joint plan achieves performance at the initial occupancy state s_0 as good as the Q-value function for an optimal joint policy. Standard policy iteration algorithms search for an optimal joint policy, which requires a finite set of α-functions to approximate V^*/Q^*, hence the resulting PWLC approximator is tight almost everywhere. Building upon Theorem 1, we search for an optimal ρ, which requires only a single α-function to approximate V^ρ/Q^ρ, thus the resulting linear approximator is loose everywhere except in the neighborhood of a few points. The former approach may require less iterations before convergence to an optimal joint policy, but the computational cost of each iteration shall increase with the number of α-functions maintained. The latter approach may require much more iterations, but all iteration shares the same computational cost.

3.3 Addressing Plan Improvement Issues

This section introduces a procedure to improve a plan starting with a sub-optimal one.

Suppose we have determined the value function $V^\rho_{0:\ell-1}$ for any arbitrary $\rho \doteq a_{0:\ell-1}$. For some control interval $t \in [\![0;\ell-1]\!]$, we would like to know whether or not we should change decision rules $a_{0:t}$ to choose $\bar{a}_{0:t} \neq a_{0:t}$. We know how good it is to follow the current plan from control interval t onward—that is V^ρ_t—but would it be better or worse to change to the new plan? One way to answer this question is to consider selecting $\bar{a}_{0:t}$ at control interval t and thereafter following decision rules $a_{t+1:\ell-1}$ of the existing ρ. The value of the resulting joint plan is given by $J(\bar{a}_{0:t-1}) + \gamma_1 V^\rho_{t+1}(T(s_0, \bar{a}_{0:t-1}))$. The key criterion is whether this quantity is greater or less than $J(\rho)$. Next, we state the plan improvement theorem for occupancy-state Markov decision processes.

Theorem 2. *Let $\rho \doteq a_{0:\ell-1}$ and $\bar{\rho} \doteq \bar{a}_{0:\ell-1}$ be any pair of plans and $J_{0:\ell}$ be a sequence of α-functions such that, for all t, $J_t(x_t, o_t) \doteq \mathbb{E}\{\alpha_0 r_0 + \ldots + \alpha_t r_t | b_0, x_t, o_t, a_{0:t-1}\}$. Let $\bar{s}_t \doteq T(s_0, \bar{a}_{0:t-1})$ and $s_t \doteq T(s_0, a_{0:t-1})$ be occupancy states at any control interval $t \in [\![0;\ell-1]\!]$ under $\bar{\rho}$ and ρ, respectively. Then, $\langle \bar{a}_{0:t*-1}, a_{t*:\ell-1}\rangle$ such that $t^* = \arg\max_{t\in[\![0;\ell-1]\!]} \langle \bar{s}_t - s_t, J_t - \gamma_1 V^\rho_t\rangle$ is as good as, or better than, ρ.*

Proof. The proof follows from the difference between the performance measure of $\rho \doteq a_{0:\ell-1}$ and $\bar{\rho} \doteq \bar{a}_{0:\ell-1}$. Let $\varsigma_t(\bar{\rho}, \rho)$ be the advantage of taking plan $\langle \bar{a}_{0:t-1}, a_{t:\ell-1}\rangle$ instead of ρ: for any control interval $t \in [\![0;\ell-1]\!]$,

$$\begin{aligned}\varsigma_t(\bar{\rho}, \rho) &= J(\bar{a}_{0:t-1}) + \gamma_1 V^\rho_t(T(s_0, \bar{a}_{0:t-1})) - J(\rho)\\ &= J(\bar{a}_{0:t-1}) - J(a_{0:t-1}) + \gamma_1(V^\rho_t(\bar{s}_t) - V^\rho_t(s_t))\\ &= \langle \bar{s}_t - s_t, J_t - \gamma_1 V^\rho_t\rangle.\end{aligned}$$

If we let $t^* \doteq \arg\max_{t=0,1,\ldots,\ell-1} \varsigma_t(\bar{\rho}, \rho)$, then plan $\langle \bar{a}_{0:t*-1}, a_{t*:\ell-1}\rangle$ achieves the highest advantage among plan set $\{\langle \bar{a}_{0:t-1}, a_{t:\ell-1}\rangle\}_{t\in[\![0;\ell-1]\!]}$ constructed based on $\bar{\rho}$. If $t^* = 0$, then $\langle \bar{a}_{0:t*-1}, a_{t*:\ell-1}\rangle = \rho$, and no improved plans were found from plan set generated from $\bar{\rho}$. Otherwise, new $\langle \bar{a}_{0:t*-1}, a_{t*:\ell-1}\rangle$ must be better than ρ. $\square$

The plan improvement theorem shows how, given $\rho \doteq a_{0:\ell-1}$ and α-function $q^\rho_{0:\ell-1}$, we can easily evaluate a change in ρ at any control interval to a particular (possibly improved) plan. To ease exploration towards promising plans, we investigate the ϵ-greedy maximization (or soft-maximization). At each control interval t and occupancy state s_t, it randomly selects $\hat{a}_t$ with probability ϵ; otherwise, it greedily selects $\hat{a}_t$ w.r.t. the current Q-value function Q^ρ_t:

$$\hat{a}_t \doteq \arg\max_{a_t:\, a^1_t\in A^1_t,\ldots,a^n_t\in A^n_t} Q^\rho_t(s_t, a_t),$$

where $\hat{\rho} \doteq \hat{a}_{0:\ell-1}$. Unfortunately, this operation proved to be NP-hard for finite M and problematic in continuous M Radner [1962], Kumar and Zilberstein [2009]. We present a mixed-integer linear programming method, which successfully performs the greedy maximization for finite M. Mixed-Integer Linear Program 1 builds on MacDermed and Isbell [2013], which introduced an integer program for the greedy maximization in finite M. We also exploit the occupancy state estimation, in which $\hat{s}_t$ replaces s_t, and the current α-function q^ρ_t.

Mixed-Integer Linear Program 1 (For finite M)**.**

$$\underset{a_t, a^1_t,\ldots,a^n_t}{\textit{Maximize}}\ \textstyle\sum_x \sum_o \hat{s}_t(x, o) \displaystyle\sum_u a_t(u|o)\cdot q^\rho_t(o, u) \quad (8)$$

$$\textit{s.t.:}\ \textstyle\sum_{u^j} a_t(u^j, u^i|o) = a^i_t(u^i|o^i),\ \forall i, u^i, o \quad (9)$$

$$\textstyle\sum_u a_t(u|o) = 1,\ \forall o \quad (10)$$

where $\{a_t(u|o)\}$ and $\{a^i_t(u^i|o^i)\}$ are positive and boolean variables, respectively.

Mixed-Integer Linear program 1 optimizes positive variables $\{a_t(u|o)\}_{u\in U, o\in O_t}$, one positive variable for each control-history pair. More precisely, each variable represents the probability $a_t(u|o)$ of control u being taken given that agents experienced joint history o. Constraints must be imposed on these variables to ensure they form proper probability distributions (10), and that they result from the product of independent probability distributions (9), one independent probability distribution for each agent. In order to make the description of the conditional independence,

$$a_t(u|o) = a^1_t(u^1|o^1) \times \cdots \times a^n_t(u^n|o^n), \quad (11)$$

we use additional variables $\{a^i_t(u^i|o^i)\}_{i\in[\![1;n]\!], u^i\in U^i, o^i\in O^i_t}$. Marginalizing out both sides of (11) over all control-history pairs of all agents except agent i, denoted $-i$, leads to (9). That is not sufficient to ensure conditional independence in general. If we further constrain $\{a^i_t(u^i|o^i)\}$ to be boolean, then system of equations (9) implies (11).

Given (9) and (10), agent variables $\{a_t^i(u^i|o^i)\}_{u^i\in U^i, o^i\in O_t^i}$ describe a proper probability distribution, so we omit corresponding constraints. Our greedy maximization approach is fundamentally different from previous ones, including the integer program by MacDermed and Isbell [2013] and the constraint optimization program by Kumar and Zilberstein [2009], Dibangoye et al. [2016]. First, while previous approaches made use of boolean variables, we use both positive and boolean variables instead. Next, prior approaches optimize a value function represented as a convex hull; we optimize an α-function instead.

4 The oSARSA Algorithm

This section presents the oSARSA algorithm with tabular representations and function approximations (using either linear functions or deep neural networks) along with convergence guarantees. oSARSA algorithms are specializations of Policy Iteration, except that we use plans instead of policies. For the sake of conciseness, we describe a generic algorithm, which can fit to either tabular or approximate representations.

In Dec-POMDPs, the goal of oSARSA is to learn $q^*_{0:\ell-1}$, a sequence of α-vectors of an optimal plan ρ^*. In particular, we must estimate $q_t(x,o,u)$ for the current plan ρ and for all reachable state x, joint history o, control u, and any control interval t. At the same time, the algorithm changes ρ towards improved plans according to the plan improvement theorem. The improved plans are constructed by exploring the occupancy-state space according to ϵ-greedy plans (see Section 3.3). To provide good estimations, we store all experiences in data set $\mathcal{D}^\rho$, from which we estimate the occupancy states and returns under ρ for any control interval (see Section 3.1). Upon estimating occupancy state $\hat{s}$ and selecting joint decision rule a, we update parametrized α-function q_t with parameter θ_t using q_{t+1}, $\mathcal{D}^\rho$ and a_{t+1} by means of temporal difference learning: for all (x,o,u),

$$\theta_t^{[\tau+1]} \doteq \theta_t^{[\tau]} + \beta_\tau \mathbb{E}_{\hat{s},a,\mathcal{D},a_{t+1}}\{\delta_t \nabla q_t^{[\tau]}(x,o,u)\} \quad (12)$$
$$\delta_t = r + \gamma_1 q_{t+1}^{[\tau]}(y,o',u') - q_t^{[\tau]}(x,o,u),$$

where β_τ is a step size, and quantity $\nabla q_t(x,o,u)$ denotes the gradient of q_t at (x,o,u) w.r.t. some parameter θ_t. Using tabular representations (e.g., finite/small M), $\theta_t = q_t$ and thus $\nabla q_t(x,o,u)$ is a unit vector $e_{x,o,u}$ whose value at (x,o,u) is one and zero otherwise. Using linear function approximations (e.g., continuous/large M), $q_t(x,o,u) \doteq \phi_t(x,o,u)^\top \theta_t$, where $\nabla q_t(x,o,u) = \phi_t(x,o,u)$ is the feature vector at (x,o,u). Algorithm 1 shows the pseudocode of oSARSA.

To establish the convergence of oSARSA, we introduce the following assumptions.

Theorem 3. *Consider assumptions: (1) The stepsizes $\{\beta_\tau\}_{\tau=1,2,\ldots}$ satisfy Robbins and Monro's conditions; (2) The occupancy states $\hat{s}_{0:\ell-1}$ and immediate returns $\hat{R}_{0:\ell-1}$ are accurately estimated; and (3) Every pair of reachable occupancy state and joint decision rule is visited infinitely often. Under these assumptions, the sequence $q^{[\tau]}_{0:\ell-1}$ generated by oSARSA converges with probability 1 to $q^*_{0:\ell-1}$.*

Algorithm 1 The oSARSA Algorithm

```
Initialize ḡ = −∞, ρ̄ and q_{0:ℓ−1} arbitrary, and D^ρ̄.
while q_{0:ℓ−1} has not converged do
    Select ε-greedily ρ w.r.t. q_{0:ℓ−1} and D^ρ̄.
    Compose D^ρ with N trajectories {ξ^[τ]}_{τ=1}^N.
    Estimate (g, ς) from [Σ_{t=0}^{ℓ−1} R̂_t | D^ρ, ŝ_0 = s_0].
    If g − ς ⩾ ḡ then (ρ̄, ḡ, D^ρ̄) = (ρ, g + ς, D^ρ).
    Update α-functions q_{0:ℓ−1} as described in (12).
end while
```

Proof. Under these assumptions, we define $\mathbb{H}^\rho$ that maps a sequence of α-vectors $q_{0:\ell-1}$ to a new sequence of α-vectors $\mathbb{H}^\rho q_{0:\ell-1}$ according to the formula: for all hidden state x, joint history o and control u, at control interval t,

$$(\mathbb{H}^\rho q_{0:\ell-1})(x,o,u) = r(x,u) + \gamma_1 \mathbb{E}\{v_{t+1}(y, o\oplus(u,z))\},$$

where $v_t(x,o) \doteq q_t(x,o,\rho(o))$ and $\rho(o)$ is the control prescribed by ρ at joint history o. Then, the plan evaluation step of the oSARSA algorithm is of the form

$$q_t^{[\tau+1]}(x,o,u) = (1-\beta_t) q_t^{[\tau]}(x,o,u) + \beta_t \kappa_t^{[\tau]}(x,o,u),$$
$$\kappa_t^{[\tau]}(x,o,u) = (\mathbb{H}^\rho q^{[\tau]}_{0:\ell-1})(x,o,u) + w_t(x,o,u),$$

where $w_t(x,o,u) = r(x,u) + \gamma_1 v^{[\tau]}_{t+1}(y,o\oplus(u,z)) - (\mathbb{H}^\rho q^{[\tau]}_{0:\ell-1})(x,o,u)$ is a zero mean noise term. Using this temporal-difference update-rule, see (12), we converge with probability 1 to $q^\rho_{0:\ell-1}$. It now remains to be verified that the plan improvement step of the oSARSA algorithm changes the current plan for an improved one. Initially, $\bar{g}$ is arbitrarily bad, so any new plan is an improved one. Then, $\bar{g} = J(\rho)$ for the current best plan ρ since occupancy state and return are accurately estimated. Hence, when ever $g \geqslant \bar{g}$, we know that the new plan $\bar{\rho}$ yields a performance measure $J(\bar{\rho})$ superior to $J(\rho)$, thus $\bar{\rho}$ improves ρ. We conclude the proof noticing that in finite M, the number of deterministic plans is finite. As a consequence, by visiting infinitely often every pair of occupancy state and decision rule we are guaranteed to visit all deterministic plans, hence an optimal one. □

It is now important to observe that we meet assumption (2) in Theorem 3 only when M is available. Otherwise, we rely on confidence bounds $[g-\varsigma, g+\varsigma]$, e.g. *Hoeffding's inequality*, on estimate $g \approx J(\rho)$. In particular, we use lower-bounds $g - \varsigma$ on sample means instead of the sample means g themselves, to limit situations where g is overestimated. Small data sets often lead to suboptimal solutions, but as the number of experiences in data set $\mathcal{D}^\rho$ increases, sample means and corresponding lower bounds get close to the mean, i.e., ς tends to 0. It is worth noticing that the memory complexity of the oSARSA algorithms is linear with the size of an α-function, *i.e.,* $\mathcal{O}(|\mathcal{D}^{\rho}|)$; and its time complexity is linear with the episodes.

5 Experiments

We ran the oSARSA algorithm on a Mac OSX machine with 3.8GHz Core i5 and 8GB of available RAM. We solved the MILPs using ILOG CPLEX Optimization Studio. We define features to use sequences of K last joint observations instead of joint histories, hence the dimension of the parameter vector θ is $|X|(|U||Z|)^K$ for finite M.
We evaluate our algorithm on multiple benchmarks from the literature all available at `masplan.org`: Mabc, Recycling, Gridsmall, Grid3x3corners, Boxpushing, and Tiger. These are the largest and the most challenging benchmarks from the Dec-POMDP literature. For each of them, we compare our algorithm to the state-of-the-art algorithms based on either a complete or a generative model: FB-HSVI Dibangoye et al. [2016], RLaR Kraemer and Banerjee [2016], and MCEM Wu et al. [2013]. We also reported results of the state-of-the-art *model-free* solver: (distributed) REINFORCE Peshkin et al. [2000]. For REINFORCE and oSARSA, we used hyper-parameters ϵ and β ranging from 1 to 10^{-3} with a decaying factor of 10^4, sample size $|\mathcal{D}| = 10^4$. We use maximum episodes and time limit 10^5 and 5 hours, respectively, as our stopping criteria.
Surprisingly, REINFORCE performs very well on domains that consist of weakly coupled agents, *see* Figure 1. However, for domains with strongly coupled agents, *e.g.,* Tiger or BoxPushing, it often gets stuck at some local optima. In contrast, oSARSA converges to near-optimal solutions when enough ressources are available over all domains, *see* Figure 1 and Table 1. Regarding the most challenging benchmarks, which require more resources, oSARSA stops before the convergence to a near-optimal solution; yet, it often outperforms the other RL algorithms. RLaR can achieve near-optimal result for small domains and short planning horizon ($\ell \leqslant 5$), assuming there exists a unique optimal plan. As for MCEM, it can solve infinite horizon problems, but similarly to REINFORCE may get stuck in local optima; this is essentially as they both use a form of gradient descent in a parametrized policy space.

T	RLaR	MCEM	REINFORCE	**oSARSA**	FB-HSVI
	Tiger ($\lvert X\rvert = 2, \lvert Z\rvert = 4, \lvert U\rvert = 9, K = 3$)				
3	5.19	N.A.	5.0	5.19	5.19
4	4.46	N.A.	4.6	4.80	4.80
5	6.65	N.A.	2.2	6.99	7.02
6	–	N.A.	0.3	2.34	10.38
7	–	N.A.	-1.7	2.25	9.99
∞	N.A.	-10	-19.9	-0.2	13.44
	Grid3x3corners ($\lvert X\rvert = 81, \lvert Z\rvert = 81, \lvert U\rvert = 25, K = 1$)				
6	–	N.A.	1.46	1.49	1.49
7	–	N.A.	2.17	2.19	2.19
8	–	N.A.	2.96	2.95	2.96
9	–	N.A.	3.80	3.80	3.80
10	–	N.A.	4.66	4.69	4.68
	Boxpushing ($\lvert X\rvert = 100, \lvert Z\rvert = 16, \lvert U\rvert = 25, K = 1$)				
3	66.08	N.A.	17.6	65.27	66.08
4	98.59	N.A.	18.1	98.16	98.59
5	–	N.A.	35.2	107.64	107.72
6	–	N.A.	36.4	120.26	120.67
7	–	N.A.	36.4	155.21	156.42
8	–	N.A.	52.9	186.04	191.22
9	–	N.A.	54.5	206.75	210.27
10	–	N.A.	54.7	218.39	223.74
∞	N.A.	59.1	58.9	144.57	224.43

Table 1: Comparing $V^{\rho}(s_0)$ of all solvers when available, where the "–" sign mean "out of memory" and/or "out of time".

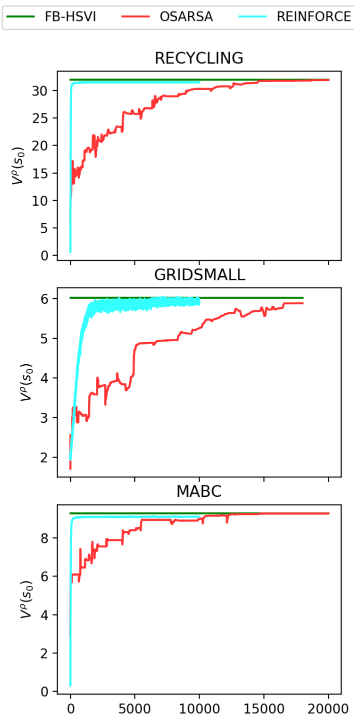

Figure 1: Comparing $V^{\rho}(s_0)$ of solvers with $\ell = \infty$ and $\gamma = 0.9$.

6 Discussion

This paper extends a recent but growing (deep) MARL paradigm Szer et al. [2005], Dibangoye et al. [2016], Kraemer and Banerjee [2016], Mordatch and Abbeel [2017], Foerster et al. [2017], namely RL for decentralized control, from model-based to model-free settings. This paradigm allows a centralized algorithm to learn on behalf of all agents how to select an optimal joint decision rule to be executed at each control interval based on all data available about the system during a learning phase, while still preserving ability for each agent to act based solely on its private histories at the execution phase. In particular, we introduced tabular and approximate oSARSA algorithms, which demonstrated promising results often outperforming state-of-the-art MARL approaches for Dec-POMDPs. To do so, oSARSA learns a value function that maps pairs of occupancy state and joint decision rule to reals. To ease the generalization in such high-dimensional continuous spaces, we restrict attention to plans rather than policies, which in turn restricts value functions of interest to linear functions. To speed up the greedy maximization, we used a MILP for finite settings—we shall use a gradient approach instead of a MILP for continuous settings in future works. Finally, we present a proof of optimality for a MARL algorithm when the estimation error is neglected. We shall investigate an approach to relax this somewhat restrictive assumption, perhaps within the probably approximately correct learning

framework.
The RL for decentralized control paradigm is significantly different from the standard RL paradigm, in which agents have the same amount of information during both the learning and the execution phases. Another major difference lies in the fact that learned value functions in standard (deep) RL algorithms are mapping from histories (or states) to reals. In contrast, oSARSA learns a value function that maps occupancy-state/decision-rule pairs to reals—spaces of occupancy states and joint decision rules are multiple orders of magnitude larger than history or state spaces. As a consequence, standard (MA)RL methods, *e.g.* REINFORCE and MCEM, may converge towards a local optimum faster than oSARSA, but the latter often converges towards a near-optimal solution. oSARSA uses occupancy states instead of joint histories mainly because occupancy states are (so far minimal) sufficient statistics for optimal decision-making in Dec-POMDPs—using joint histories instead of occupancy states may lead to suboptimal solutions except in quite restrictive settings. For example, RLaR learns value functions mapping history/action pairs to reals, but convergence towards an optimal solution is guaranteed only for domains that admit a unique optimal joint plan—which essentially restricts to POMDPs Kraemer and Banerjee [2016].

References

J. A. Bagnell, S. M. Kakade, J. G. Schneider, and A. Y. Ng. Policy Search by Dynamic Programming. In *Advances in Neural Information Processing Systems 16*. 2004.

R. E. Bellman. *Dynamic Programming*. 1957.

D. S. Bernstein, S. Zilberstein, and N. Immerman. The Complexity of Decentralized Control of Markov Decision Processes. In *Proc. of the Sixteenth Conf. on Uncertainty in AI*, 2000.

G. W. Brown. Iterative Solutions of Games by Fictitious Play. In *Activity Analysis of Production and Allocation*. 1951.

J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Optimally Solving Dec-POMDPs as Continuous-State MDPs. *Journal of AI Research*, 55, 2016.

J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual Multi-Agent Policy Gradients. *CoRR*, 2017.

C. Guestrin, M. G. Lagoudakis, and R. Parr. Coordinated Reinforcement Learning. In *Proc. of the Eighteenth Int. Conf. on ML*, San Francisco, CA, USA, 2002.

E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic Programming for Partially Observable Stochastic Games. In *Proc. of the Nineteenth National Conf. on AI*, 2004.

R. A. Howard. *Dynamic Programming and Markov Processes*. 1960.

J. Hu and M. P. Wellman. Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm. In *Proc. of the Fifteenth Int. Conf. on ML*, San Francisco, CA, USA, 1998.

J. R. Kok and N. Vlassis. Sparse Cooperative Q-learning. In *Proc. of the Twentieth Int. Conf. on ML*, New York, NY, USA, 2004.

L. Kraemer and B. Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

A. Kumar and S. Zilberstein. Constraint-based dynamic programming for decentralized POMDPs with structured interactions. In *Proc. of the Eighth Int. Conf. on Autonomous Agents and Multiagent Systems*, 2009.

M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. of the Eleventh Int. Conf. on ML*, 1994.

M. Liu, C. Amato, X. Liao, L. Carin, and J. P. How. Stick-breaking policy learning in Dec-POMDPs. In *Int. Joint Conf. on AI (IJCAI) 2015*. AAAI, 2015.

M. Liu, C. Amato, E. P. Anesta, J. D. Griffith, and J. P. How. Learning for Decentralized Control of Multiagent Systems in Large, Partially-Observable Stochastic Environments. In *AAAI*, 2016.

L. C. MacDermed and C. Isbell. Point Based Value Iteration with Optimal Belief Compression for Dec-POMDPs. In *Advances in Neural Information Processing Systems 26*, 2013.

T. Miconi. When Evolving Populations is Better Than Coevolving Individuals: The Blind Mice Problem. In *Proc. of the 18th Int. Joint Conf. on AI*, IJCAI'03, 2003.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540), feb 2015. ISSN 0028-0836.

I. Mordatch and P. Abbeel. Emergence of Grounded Compositional Language in Multi-Agent Populations. *CoRR*, abs/1703.0, 2017.

A. Nayyar, A. Mahajan, and D. Teneketzis. Optimal Control Strategies in Delayed Sharing Information Structures. *Automatic Control, IEEE Transactions on*, 56(7), 2011.

F. A. Oliehoek. Sufficient Plan-Time Statistics for Decentralized POMDPs. In *Proc. of the Twenty-Fourth Int. Joint Conf. on AI*, 2013.

F. A. Oliehoek, M. T. J. Spaan, and N. A. Vlassis. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *Journal of AI Research*, 32, 2008.

F. A. Oliehoek, M. T. J. Spaan, C. Amato, and S. Whiteson. Incremental Clustering and Expansion for Faster Optimal Planning in Dec-POMDPs. *Journal of AI Research*, 46, 2013.

L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3), 2005.

L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling. Learning to Cooperate via Policy Search. In *Sixteenth Conf. on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000.

J. M. Porta, N. Vlassis, M. T. J. Spaan, and P. Poupart. Point-Based Value Iteration for Continuous POMDPs. *J. Mach. Learn. Res.*, 7, 2006. ISSN 1532-4435.

M. L. Puterman. *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. Hoboken, New Jersey, 1994.

R. Radner. Team Decision Problems. *Ann. Math. Statist.*, 33(3), 1962.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, 22(3), 1951. ISSN 0003-4851.

R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton, N. J., 1970.

M. Rummery, G. A. and Niranjan. On-line Q-learning using connectionist systems. Technical report, Cambridge University Engineering Department, 1994.

R. Salustowicz, M. Wiering, and J. Schmidhuber. Learning Team Strategies: Soccer Case Studies. *ML*, 33(2-3), 1998.

G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *Journal of Autonomous Agents and Multi-Agent Systems*, 27(1), 2013.

R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A Heuristic Search Algorithm for Solving Decentralized POMDPs. In *Proc. of the Twenty-First Conf. on Uncertainty in AI*, 2005.

M. Tan. Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents. In *Readings in Agents*. San Francisco, CA, USA, 1998.

C. J. C. H. Watkins and P. Dayan. Q-Learning. *ML*, 8(3), 1992.

F. Wu, S. Zilberstein, and N. R. Jennings. Monte-Carlo Expectation Maximization for Decentralized POMDPs. In *Proc. of the Twenty-Fourth Int. Joint Conf. on AI*, 2013.

C. Zhang and V. Lesser. Coordinated Multi-Agent Reinforcement Learning in Networked Distributed POMDPs. In *Proc. of the Twenty-Fifth AAAI Conf. on AI*, San Francisco, California, USA, 2011.