Multiagent Planning and Learning As MILP

Jilles Dibangoye, Olivier Buffet, Akshat Kumar

Multiagent Planning and Learning As MILP

Jilles S. Dibangoye[1] Olivier Buffet[2] Akshat Kumar[3]

[1] Univ Lyon, INSA Lyon, INRIA, CITI, F-69621 Villeurbanne, France
[2] INRIA / Université de Lorraine, Nancy, France
[3] Singapore Management University, Singapore

jilles-steeve.dibangoye@insa-lyon.fr

Résumé

Les processus décisionnels de Markov décentralisés et partiellement observables (Dec-POMDPs) offrent un cadre unifié pour la prise de décisions séquentielles par de plusieurs agents collaboratifs—mais ils restent difficiles à résoudre. Les reformulations en programmes linéaires mixtes (PLMs) se sont avérées utiles pour les processus décisionnels de Markov partiellement observables. Malheureusement, les applications existantes se limitent uniquement aux domaines mobilisant un ou deux agents. Dans cet article, nous exploitons une propriété de linéarisation qui nous permet de reformuler les contraintes non linéaires, omniprésentes dans les systèmes multi-agents, pour en faire des contraintes linéaires. Nous présentons en outre des approches de planification et d'apprentissage s'appuyant sur de nouvelles reformulations en PLMs des Dec-POMDPs, dans le cas général ainsi que quelques cas spécifiques. Les expérimentations sur des bancs de test standards à deux et plus de deux agents fournissent un solide soutien à cette méthodologie.

SMAs, Planification, Apprentissage, PLM

Abstract

The decentralized partially observable Markov decision process offers a unified framework for sequential decision-making by multiple collaborating agents but remains intractable. Mixed-integer linear formulations proved useful for partially observable domains, unfortunately existing applications restrict to domains with one or two agents. In this paper, we exploit a linearization property that allows us to reformulate nonlinear constraints from n-agent settings into linear ones. We further present planning and learning approaches relying on MILP formulations for general and special cases, including network-distributed and transition-independent problems. Experiments on standard 2-agent benchmarks as well as domains with a large number of agents provide strong empirical support to the methodology.

MAS, Planning, Learning, MILP

1 Introduction

The decentralized partially observable Markov decision process offers a unified framework to solving cooperative, decentralized stochastic control problems [Bernstein et al., 2002]. This model encompasses a large range of real-world problems in which multiple agents collaborate to optimize a common objective. Central to this setting is the assumption that agents can neither see the actual state of the world nor explicitly communicate their observations with each other due to communication cost, latency or noise. This assumption partially explains the worst-case complexity: finite-horizon cases are in NEXP; and infinite-horizon cases are undecidable [Bernstein et al., 2002]. A general methodology to solving decentralized stochastic control builds upon the concept of occupancy states, *i.e. sufficient statistics* for evaluating and selecting decentralized policies [Dibangoye et al., 2013, Oliehoek, 2013, Dibangoye et al., 2014b]. The occupancy-state space is a probability simplex of points in the Cartesian product over the state and history spaces. For every occupancy state, dynamic programming and reinforcement learning approaches compute and store tables consisting of one value per state-history pair. Unfortunately, the state space grows exponentially with the number of state variables, and the history space expands doubly exponentially with time. Known as the *curse of dimensionality*, these phenomena render existing approaches intractable in the face of decentralized decision-making problems of practical scale.

Methods that can overcome the curse of dimensionality previously arose in the literature of decentralized control, but restricting to 2-agent settings. Examples include memory-bounded dynamic programming [Seuken and Zilberstein, 2008, Kumar et al., 2015] and linear and nonlinear programming using finite-state controllers [Amato et al., 2010, Kumar et al., 2016]. These successes shed light on approximate dynamic and linear programming as potentially powerful tools for large-scale decentralized partially observable Markov decision processes. One approach to dealing with the curse of dimensionality is to rely

on parametrized occupancy measures [Dibangoye et al., 2014b]. However, choosing a parametrization that can closely mimic the desired occupancy measures requires human expertise or theoretical analysis. Though crucial, the design of an approximation architecture goes beyond the scope of this paper. Instead this paper focusses on exact planning and learning approaches relying on MILP for computing good decentralized policies given parametrized occupancy measures.

To this end, we first exploit a linearization property that allows us to reformulate into linear ones all nonlinear constraints that arose from multiple collaborating agents. We then present a general MILP formulation for n-agent settings, restricting attention to deterministic finite-memory decentralized policies[1]. In addition, we introduce a two-phase approach that produces a sequence of decentralized policies and dynamics through iteration. At the first phase called the model estimation, we maintain statistics about the dynamics. At the second phase namely the policy improvement, we rely on our MILP formulation to calculate a new decentralized policy based on the current dynamics. Under the discounted reward criterion, the sequence of decentralized policies converges to some optimal deterministic and finite-memory decentralized policy. We further demonstrate how to use this planning and learning scheme to exploit two properties present in many decentralized stochastic control problems, namely joint-full observability and weak separability [Becker et al., 2004, Nair et al., 2005]. Experiments on standard 2-agent benchmarks as well as domains with a large number of agents provide strong empirical support to the methodology for planning and learning good decentralized policies.

2 Related Work

Mixed-integer linear programming was used previously for decentralized decision-making, but always with a focus different from ours. Much of the effort has been directed toward exact formulations for restricted classes of either problems or policies. Witwicki and Durfee [2007] and later on Mostafa and Lesser [2011] presented formulations for 2-agent transition independent decentralized Markov decision processes [Becker et al., 2004]. Aras and Dutech [2010] introduced an exact formulation for 2-agent finite-horizon decentralized decision-making, which inevitably scales poorly with the number of state variables and planning horizon. More recently, Kumar et al. [2016] proposed yet another 2-agent formulation but with a focus on finite-state controllers. Unfortunately, there are a number of factors that may affect the performance of solvers while optimizing finite-state controllers. First, the numbers of variables and contraints grow linearly with the number of agents, states, actions, and observations; even more importantly they grow quadratically with the number of nodes per controller. Consequently, MILP formulations of interest for practical problems involve prohibitively large numbers of variables and constraints. The second limitation is somewhat imperceptible and has to do with the semantic of nodes in finite-state controllers. Each node aims at representing a partition of the history space as well as prescribing an action to be taken in that node. Interestingly, these separate decisions are interconnected. The actions to be taken in all nodes may affect the histories to be subsumed in each specific node and vice-versa. Taken all together, these barriers make optimizing finite-state controllers particularly hard tasks, which explains the impetus for the development of a novel approximation approach. Kumar et al. [2016] suggest a heuristic search method that can incrementally build small 2-agent finite-state controllers. However, to the best of our knowledge, no existing MILP formulation for decentralized decision-making can cope with problems with more than two agents. The primary contribution of this paper is to provide the first attempt to handle this issue.

To address the adaptive case, *i.e.* when the dynamics model is unknown, a standard approach is reinforcement learning. Existing reinforcement learning methods for decentralized decision-making extend adaptive dynamic programming and policy search approaches from single to multiagent settings. Currently, no multiagent reinforcement learning methods is based on MILP. This is somewhat surprising since, there are single-agent reinforcement learning methods based on linear programming. One such approach, namely `probing`, consists of two phases: (i) the estimation phase, where the transition probabilities are updated; and (ii) the control phase, where a new policy is calculated based on the current transition probabilities [Altman, 1999]. The algorithm iterates these two phases forever or until the training budget is exhausted. In this paper, we extend this approach to decentralized decision-making, thus providing the first multiagent model-based reinforcement learning method based on MILP.

3 Backgrounds

The paper makes use of the following notation. $\delta_y(x)$ is the Kronecker delta function. For any arbitrary finite set B, $|B|$ denotes the cardinality of B, $\mathbb{N}_{\leq |B|} = \{0, 1, \ldots, |B|\}$, $\mathbb{N}^+_{\leq |B|} = \{1, \ldots, |B|\}$, and $\triangle(B)$ is the $(|B| - 1)$-dimensional real simplex. Also, we use shorthand notations $b_{1:n} = (b_1, b_2, \ldots, b_n)$ and $b^\top$ to denote the transpose of b. Finally, we shall use short-hand notation $P_\zeta(\cdot)$ to denote probability distribution $\mathbb{P}(\cdot|\zeta)$ conditional on ζ.

3.1 Problem Formulation

A decentralized partially observable Markov decision process is given by a tuple $M_n \doteq (X, U, Z, p, q, r, \nu)$ made of: a finite set of n agents; a finite state space X; a finite action space $U = U_1 \times U_2 \times \cdots \times U_n$; a finite observation

[1]Though randomized finite-memory decentralized policies should achieve better performances than deterministic ones, the corresponding optimization problem is non-convex, often leading to poorer or equivalent random solutions in comparison to deterministic ones [Amato et al., 2010].

space $Z = Z_1 \times Z_2 \times \cdots \times Z_n$; state transition probabilities $p^u(x, y)$ representing the probability that next state will be y given that the current state is x and the current action is u; observation probabilities $q^u(y, z)$ representing the probability that after taking action u next state and observation will be y and z, respectively; rewards $r(x, u)$ representing the reward incurred when taking action u in current state x; and ν is the initial state-history distribution.

Solving decentralized stochastic control problems aims at finding a (decentralized) policy a, *i.e.,* n independent policies $(a_1, a_2, \ldots, a_n)$, one individual policy for each agent. Each policy a prescribes actions conditional on (action-observation) histories $o = (o_1, o_2, \ldots, o_n)$, initially $o \doteq (\emptyset, \emptyset, \ldots, \emptyset)$, such that

$$a(o, u) = \prod_{i=1}^{n} a_i(o_i, u_i), \qquad \forall o \in O, u \in U, \qquad (1)$$

where $O = O_1 \times \cdots \times O_n$ is a finite set of histories, ranging from 0- to ℓ-steps histories. We shall restrict attention to ℓ-order Markov policies. These policies map 0- to ℓ-steps histories to actions, in particular 1-order Markov policies are called Markov policies. For any arbitrary ϵ, one can choose $\ell = \log_\alpha (1-\alpha)\epsilon / \|r\|_\infty$ so that there always exists at least one ℓ-order policy within ϵ of an optimal one. For each policy a, we define a transition matrix P^a, where each entry $P^a(x, o, x', o')$ denotes the probability of transiting from state-history (x, o) to state-history (x', o').

Policies of interest are those that achieve the highest performance. In this paper, we consider the infinite-horizon normalized discounted reward criterion, which ranks policies a according to the initial state-history distribution ν and a discount factor $\alpha \in (0, 1)$ as follows:

$$J_\alpha(a; \nu) \doteq (1-\alpha) \sum_{\tau=0}^{\infty} \alpha^\tau \mathbb{E}^a_\nu \{r(x_\tau, u_\tau)\}, \qquad (2)$$

where $\mathbb{E}^a_\nu\{\cdot\}$ denotes the expectation with respect to state-action-history distributions $P^a_\nu(\tau)$ at each time step τ conditional on distribution ν and policy a, also known as a τ-th *occupancy state*. $P^a_\nu(\tau; x_\tau, o_\tau)$ denotes the probability of being in state x_τ after experiencing history o_τ at decision epoch τ when agents follow policy a starting in ν. An optimal policy $a^* \in \arg\max_a \ J_\alpha(a; \nu)$ is one that achieves the unique optimal value $J_\alpha(\nu) = J_\alpha(a^*; \nu)$.

3.2 Extended Occupancy Measures

This section presents the notion of *extended occupancy measures* which describe the variables when solving M_n as MILP. Extended occupancy measures subsume two critical quantities: (i) the target policy; and (ii) the state-history-action frequency called hereafter *occupancy measure*. Next, we provide intuitions behind the concept of occupancy measures as well as key properties.

To overcome the fact that agents can neither see the state of the world nor explicitly communicate with one another, Szer et al. [2005] suggest formalizing decentralized stochastic control problems from the perspective of an offline central planner (respectively learner). A central planner selects a policy to be executed by the agents. In general, resulting policies are non-stationary, *i.e.* agents may act differently from one decision epoch to another one. For the sake of conciseness, we restrict attention to stationary policies. This choice gives rise to statistics, namely *occupancy measures* $s_\alpha(\nu, a)$, that summarizes all occupancy states $\{P^a_\nu(\tau)\}_{\tau \in \mathbb{N}}$ encountered under policy a starting at state distribution ν.

Definition 1. *The occupancy measure under policy a starting at initial distribution ν is given by:*

$$s_\alpha(\nu, a) \doteq (1-\alpha) \textstyle\sum_{\tau=0}^{\infty} \alpha^\tau P^a_\nu(\tau). \qquad (3)$$

Interestingly, the occupancy measure comes with many important properties.

Lemma 1. *$s_\alpha(\nu, a)$ is a probability distribution.*

If α were seen as a survival probability at each time step, then $s_\alpha(\nu, a)$ gives, for state-history pair (x, o), the probability to be in that situation just before dying. Combining (2) and (3), it appears that $J_\alpha(a; \nu)$ is a linear function of occupancy measure $s_\alpha(\nu, a)$:

Lemma 2. *$s_\alpha(\nu, a)$ is a sufficient statistic for estimating infinite-horizon normalized discounted reward:*

$$J_\alpha(a; \nu) = \mathbb{E}^a_{s_\alpha(\nu, a)} \{r(x, u)\}. \qquad (4)$$

Lemma 2 proves occupancy measures $s_\alpha(\nu, a)$ also preserves ability to estimate α-discount reward $J_\alpha(a; \nu)$. Finally, occupancy measure $s_\alpha(\nu, a)$ satisfies a linear characterization, which shall prove critical to solve M_n as MILP.

Lemma 3. *Occupancy measure $s_\alpha(\nu, a)$ is the solution of the following linear equation w.r.t. $s_\alpha(\nu, a)$:*

$$s_\alpha(\nu, a)^\top (I - \alpha P^a) = (1-\alpha)\nu^\top. \qquad (5)$$

To solve M_n, it will prove useful to search both a policy a and the corresponding occupancy measure $s_\alpha(\nu, a)$. We are ready to define extended occupancy measures.

Definition 2. *Extended occupancy measure $\zeta_\alpha(\nu, a) \doteq \{\zeta_\alpha(\nu, a; x, o, u)\}$ over state-history-action triplets, associated with each policy a, initial distribution ν, and discount factor α, is given by*

$$\zeta_\alpha(\nu, a; x, o, u) \doteq s_\alpha(\nu, a; x, o) \cdot a(o, u), \quad \forall x, o, u. \quad (6)$$

The extended occupancy measure captures the frequency of visits of each state-history-action triplet when the system runs under policy a, conditioned on initial distribution ν. Interestingly, because it subsumes an occupancy measure, it also inherited occupancy measures' properties, including: (i) it is a probability distribution; (ii) it can accurately estimate infinite-horizon normalized discounted reward; and (iii) it satisfies a linear characterization.

4 MILP Reformulations

To motivate the role of extended occupancy measures (6), let us start with a mathematical program to finding a^*. Consider problem $(\mathcal{P}_1)$ given by:

$$\text{Maximize}_{a,a_{1:n},\zeta}\ \mathbb{E}_s\{r(x,u)\} \text{ subject to: (1) and (5)}$$

where a_i is agent i's policy, a defines the decentralized policy, and ζ denotes the extended occupancy measure. It can be shown, using (5), that any feasible ζ of $(\mathcal{P}_1)$ is an extended occupancy measure $\zeta_\alpha(\nu, a)$ under policy a. It follows that, for any $\zeta = \zeta_\alpha(\nu, a)$ solution of program $(\mathcal{P}_1)$, policy a is optimal for any selected class of finite-memory policies. Unfortunately, $(\mathcal{P}_1)$ is a nonlinear optimization problem, with many local optima. Earlier attempts to solving $(\mathcal{P}_1)$—for one or two agents only—make use of nonlinear solvers, often leading to local optima [Amato et al., 2010]. The remainder of this section presents an exact mixed-integer linear program for M_n, restricting attention to deterministic and stationary ℓ-order Markov policies, as they have been shown to achieve ϵ-optimal performance.

4.1 ℓ-order Markov Policies

Notice that (5) is a nonlinear constraint, so that $(\mathcal{P}_1)$ is not a MILP. Therefore, finding an ϵ-optimal policy by directly solving $(\mathcal{P}_1)$ is hopeless in general, though from case to case nonlinear programming may achieve good results [Amato et al., 2010]. However, it is possible to reformulate the constraints to transform the problem into a MILP. Previous linearization of nonlinear programs to solving decentralized stochastic control problems have been limited to two-agent cases, with either specific problem assumptions, *e.g.,* transition-independent settings [Wu and Durfee, 2006], or restricted classes of policies, *e.g.,* sequence-form policies [Aras and Dutech, 2010]; finite-state controllers [Kumar et al., 2016].

Next, we introduce a mixed-integer linear programming approach for general discrete-time decentralized stochastic control problems. Before proceeding any further let us provide preliminary properties that will be useful to establish the main results of the paper. In particular, we present a linearization property that allows us to formulate nonlinear constraints in $(\mathcal{P}_1)$ as linear constraints. We start with the linearization of the product between Boolean and continuous variables [Berthold et al., 2009].

Lemma 4 ([Berthold et al., 2009])**.** *If we let v_1, v_2, and w be Boolean, random, and non-negative variables, respectively; and*

$$(C_1)\left|\begin{array}{ll} w - v_k & \leq 0 \quad \forall k \in \{1,2\} \\ v_1 + v_2 - w & \leq 1 \end{array}\right.$$

then solutions of polyhedron (C_1) satisfy $w = v_1 \cdot v_2$.

The next property shows for the first time how to exploit Lemma 4 to reformulate nonlinear constraints from n-agent cases into linear ones.

Proposition 1. *If we let $\zeta(x, o_{1:n}, u_{1:n})$ be a joint distribution and $\{a_i(o_i, u_i)\}_{i\in\mathbb{N}^+_{\leq n}}$ be Boolean variables; and*

$$(C_\zeta(o_i,u_i))\left|\begin{array}{ll} P_\zeta(o_i,u_i) - a_i(o_i,u_i) & \leq 0 \\ P_\zeta(o_i) + a_i(o_i,u_i) - P_\zeta(o_i,u_i) & \leq 1 \end{array}\right.$$

then solutions of polyhedron $\{C_\zeta(o_i, u_i)\}_{i\in\mathbb{N}^+_{\leq n}}$ satisfy

$$\begin{array}{ll} \zeta(x, o_{1:n}, u_{1:n}) & = P_\zeta(x, o_{1:n})\prod_{i=1}^n P_\zeta(u_i|o_i) \\ a_i(o_i, u_i) & = P_\zeta(u_i|o_i). \end{array}$$

Proof. The extended occupancy state $\zeta(x, o, u)$ can be rewritten equivalently as follows:

$$\begin{aligned} \zeta(x,o,u) &\doteq P_\zeta(x,o_{1:n})\textstyle\prod_{j=1}^n P_\zeta(u_j|o_j) \\ &= P_\zeta(x,o_{-i},u_{-i}|o_i,u_i)P_\zeta(o_i)P_\zeta(u_i|o_i) \quad (7) \\ &= P_\zeta(x,o_{-i},u_{-i}|o_i,u_i)P_\zeta(o_i,u_i) \quad (8) \end{aligned}$$

Since LHS of both (7) and (8) are equal, we have equivalently

$$P_\zeta(o_i,u_i) = P_\zeta(o_i)P_\zeta(u_i|o_i), \qquad \forall i, o_i, u_i. \quad (9)$$

Using Lemma 4 along with the fact that $\{P_\zeta(u_i|o_i)\}$ are Boolean variables (and the obvious result that $P_\zeta(o_i,u_i) - P_\zeta(o_i) \leq 0$), we know that solutions of $\{C_\zeta(o_i,u_i)\}$ also satisfy (9). Equality $a_i(o_i,u_i) = P_\zeta(u_i|o_i)$ follows directly from Lemma 4. Indeed, if $P_\zeta(o_i) = 0$, the first inequality implies $a_i(o_i,u_i) = 0 (= P_\zeta(u_i|o_i))$; otherwise $P_\zeta(o_i) = 0$, and Lemma 4 gives us $P_\zeta(o_i,u_i) = a_i(o_i,u_i)\cdot P_\zeta(o_i) \neq 0$, so that $a_i(o_i,u_i) = P_\zeta(o_i,u_i)/P_\zeta(o_i) = P_\zeta(u_i|o_i)$ using Bayes rule. □

We are now poised to present a MILP to solving general decentralized stochastic control problems.

Theorem 1. *If we let $\{a_i(o_i,u_i)\}$ and $\{\zeta(x,o,u)\}$ be Boolean and non-negative variables, respectively, then a solution of mixed-integer linear program $(\mathcal{P}_2)$:*

$$\max_{a_{1:n},\zeta}\ \mathbb{E}_\zeta\{r(x,u)\} \ \textit{s.t. (5) and } \{C_\zeta(o_i,u_i)\}_{i,o_i\in O_i,u_i\in U_i}$$

is an ϵ-optimal solution of $(\mathcal{P}_1)$, where a_i is agent i's policy and ζ denotes the extended occupancy measure.

Proof. From Oliehoek et al. [2008], we know there always exists a deterministic history-dependent decentralized policy that is as good as any randomized history-dependent decentralized policy. Moreover, as previously discussed, by restricting attention to ℓ-order Markov decentralized policies, the best possible performance in this subclass is within $\|r\|_\infty \cdot \alpha^\ell/(1-\alpha)$ of the optimal performance, *i.e.,* the regret of taking arbitrary decisions from time step ℓ onward. Hence, by searching in the space of deterministic and ℓ-order Markov decentralized policies, *i.e.,* one individual ℓ-order Markov policy a_i for each agent $i \in \mathbb{N}^+_{\leq n}$, we preserve ability to find an ϵ-optimal solution of the

original problem M_n under the discounted-reward criterion, where $\epsilon \leq \|r\|_\infty \cdot \alpha^\ell/(1-\alpha)$. Since a_i is agent i's policy and ζ denotes the extended occupancy measure, we know that $\{a_i(o_i,u_i)\}$ and $\{\zeta(x,o_{1:n},u_{1:n})\}$ are Boolean and non-negative variables, respectively. Thus, from Proposition 1, we have that solutions of polyhedron $\{C_\zeta(o_i,u_i)\}_{i,o_i\in O_i,u_i\in U_i}$ satisfy $\zeta(x,o_{1:n},u_{1:n}) = s(x,o_{1:n})\prod_{i=1}^n a_i(o_i,u_i)$ and $s(x,o_{1:n})$ are marginal probabilities $P_\zeta(x,o_{1:n})$ and $a_i(o_i,u_i)$ are conditional probabilities $P_\zeta(u_i|o_i)$ for $i\in\mathbb{N}^+_{\leq n}$. □

This theorem establishes a general MILP to finding an ϵ-optimal policy in M_n under the discounted-reward criterion. We will refer to this problem as the exact MILP. Unfortunately, the state, action, history spaces for practical problems are enormous due to the curse of dimensionality. Consequently, the MILP of interest involves prohibitively large numbers of variables and constraints. $(\mathcal{P}_2)$ considers less constraints than its nonlinear counterpart $(\mathcal{P}_1)$, but the same number of variables. Variables in $(\mathcal{P}_1)$ are all free, whereas variables $\{a_i(o_i,u_i)\}_{i\in\mathbb{N}^+_{\leq n},o_i\in O_i,u_i\in U_i}$ in $(\mathcal{P}_2)$ are Boolean and remainders $\{\zeta(x,o,u)\}_{x\in X,o\in O,u\in U}$ are free.

5 Tractable Subclasses

In this section, we present two examples involving the mixed-integer linear formulations for subclasses of decentralized partially observable Markov decision processes. The intention is to illustrate more concretely how the formulation might be achieved and how reasonable choices lead to near-optimal policies. We shall consider *joint-observability* and *weak-separability* assumptions.

5.1 Joint observability assumption

We first consider a setting where agents collectively observe the true state of the world. This assumption, known as joint observability, arises in many decentralized Markov decision processes [Bernstein et al., 2002], *e.g.,* transition-independent decentralized Markov decision processes [Becker et al., 2004]. More formally, we say that a system is jointly observable if and only if there exists a surjective function $\varphi\colon Z\mapsto X$ which prescribes the true state of the world given the current joint observation.

Corollary 1. *Under joint observability assumption, if we let $a_i(z_i,u_i)_{i\in\mathbb{N}^+_{\leq n}}$ and $\zeta(z,u)$ be Boolean and non-negative variables, respectively, then:*
(i) the transition probability from observation z to observation z' upon taking action u is

$$p^u_\varphi(z,z') \doteq \textstyle\sum_{x\in X}\delta_x(\varphi(z))\sum_{y\in X}p^u(x,y)\cdot q^u(y,z'),$$

where the rewards over observations is given by $r_\varphi(z,u)\doteq\sum_{x\in X}\delta_x(\varphi(z))\cdot r(x,u)$; and the initial distribution over observations is given by $\nu_\varphi(z)\doteq\sum_{x\in X}\delta_x(\varphi(z))\nu(x)$;
(ii) the occupancy measures over observations satisfy

$$s^\top(I-\alpha P^a)\doteq(1-\alpha)\nu^\top_\varphi; \tag{10}$$

(iii) a solution of mixed-integer linear program $(\mathcal{P}_3)$

$$\max_{a_{1:n},\zeta}\mathbb{E}_\zeta\{r_\varphi(z,u)\}\ \textit{s.t. (10) and}\ \{C_\zeta(z_i,u_i)\}_{i,z_i\in Z_i,u_i\in U_i}$$

is also solution of $(\mathcal{P}_2)$, where a_i is agent i's policy and ζ denotes the extended occupancy measure.

This corollary presents an approximate mixed-integer linear program that can find Markov decentralized policies under joint observability. Markov policies, a.k.a. 1-order Markov policies, act depending only upon the current observation. This formulation depends on states and histories only through the current observations, which results in a significant reduction in the number of variables and constraints, *i.e.,* from $\mathbf{O}(|X||O||U|)$ in $(\mathcal{P}_2)$ to $\mathbf{O}(|Z||U|)$ in $(\mathcal{P}_3)$. Interestingly, this formulation finds optimal policies in transition-independent decentralized Markov decision processes, as deterministic Markov policies were proven to be optimal in such a setting [Goldman and Zilberstein, 2004].

5.2 Weak separability assumption

Next, we consider the weak separability assumption, which arises in network-distributed partially observable Markov decision processes [Nair et al., 2005]. The assumption allows us to decouple variables involved in the approximate mixed-integer linear programs into factors, *i.e.,* subsets of variables, which make it possible to scale up to large number of agents. The intuition behind this assumption is that not all agents interact with one another; often an agent interacts only with a small subset of its neighbors, hence its decisions may not affect the remainder of its teammates. To take into account the locality of interaction, we make the following assumptions.

Definition 3. *Let E be a set of subsets e of agents. A decentralized partially observable Markov decision process (n,X,Z,U,q,p,r,ν) is said to be weakly separable if the following holds: n denotes the number of agents; $X\doteq X_0\times X_1\times\cdots\times X_n$; ν is multiplicatively fully separable,* i.e., *there exists $(\nu_i)_{i\in\mathbb{N}_{\leq n}}$ such that $\nu(x)=\prod_{i\in\mathbb{N}_{\leq n}}\nu_i(x_i)$, where $x=(x_i)_{i\in\mathbb{N}_{\leq n}}$; p is multiplicatively weakly separable,* i.e., *there exists $(p_i)_{i\in\mathbb{N}_{\leq n}}$ such that $p^{u_e}_e(x_e,y_e)=p_0(x_0,y_0)\prod_{i\in e}p^{u_i}_i(x_i,y_i)$, where $x_e=(x_0,(x_i)_{i\in e})$ and $u_e=(u_i)_{i\in e}$; q is multiplicatively weakly separable,* i.e., *there exists $(q_i)_{i\in\mathbb{N}^+_{\leq n}}$ such that $q^{u_e}_e(z_e,y_e)=\prod_{i\in e}q^{u_i}_i(z_i,y_i)$, where $y_e=(y_i)_{i\in e}$, $z_e=(z_i)_{i\in e}$ and $u_e=(u_i)_{i\in e}$; r is additively weakly separable,* i.e., *there exists $(r_e)_{e\in E}$ such that $r(x,u)=\sum_{e\in E}r_e(x_e,u_e)$, for all state and action x,u.*

This assumption suggests two agents, i and j, can only affect one another if they share the same subset e, *i.e.,* $i,j\in e$; otherwise they can choose what to do with no knowledge about what the other sees or plans to do. As a consequence, the value function in this setting is proven to be additively weakly separable [Dibangoye et al., 2014a],

i.e., $\mathbb{E}_\zeta\{r(x,u)\} = \sum_{e\in E}\mathbb{E}_{\zeta_e}\{r_e(x,u)\}$, where

$$s_e^\top (I - \alpha P^{a_e}) \doteq (1-\alpha)\nu_e^\top, \tag{11}$$

describes the recursion definition of the occupancy measure s_e extract from the extended occupancy measure ζ_e. The following exploits this property to define an exact mixed-integer linear program that decouples variables according to E, resulting in significant dimensionality reduction.

Corollary 2. *Let M_n be weakly separable. If we let $\{a_i(o_i,u_i)\}_{i\in\mathbb{N}^+_{\leq n}}$, and $\{\zeta_e(x_e,o_e,u_e)\}$ be Boolean and non-negative variables, respectively; then a solution of mixed-integer linear program $(\mathcal{P}_4)$*

$$\max_{a_{1:n},\zeta} \textstyle\sum_{e\in E}\mathbb{E}_{\zeta_e}\{r_e(x_e,u_e)\} \text{ s.t. (11) and } \{C_{\zeta_e}(o_i,u_i)\}$$

is also a network-distributed solution of $(\mathcal{P}_2)$, where where a_i is agent i's policy and ζ_e denotes the extended occupancy measure for all e.

This mixed-integer linear program exploits the so-called weak separability assumption that arises under locality of interaction. It is worth mentioning that $(\mathcal{P}_4)$ can find an optimal policy for network-distributed partially observable Markov decision processes, assuming reasonable choice of histories O_e for each subset of agents e [Dibangoye et al., 2014a].

6 Adaptive Decentralized Control

In this section, we extend to decentralized partially observable Markov decision processes the probing algorithm originally introduced for Markov decision processes. Similarly to the original algorithm, ours alternates between model estimation and policy improvement phases forever or until the training budget is exhausted. The estimators of both dynamics $\{P_\tau\}$ and exploration policies $\{\pi_\tau\}$ shall involve counting the number of times the algorithm visits state-action-state-observation quadruplets (x,u,y,z), state-action pairs (x,u), and states x after τ interactions between the agent and the environment—by an abuse of notation we shall use $w_\tau(x,u,y,z)$, $w_\tau(x,u)$, $w_\tau(x)$ to store these numbers, respectively. It is worth noticing that since the model does not depend on histories, maintaining history-dependent policies $\{a_\tau\}$ is useless, instead it suffices to maintain state-dependent policies $\{\pi_\tau\}$ corresponding to extended occupancy measures $\{\zeta_\tau\}$.

Under standard ergodicity conditions, the model estimation phase ensures each state-action pairs is visited infinitely often, making $P_\tau = \{p_\tau^{u,z}(x,y)\}$ a consistent estimator of dynamics after τ interactions: $p_\tau^{u,z}(x,y) = w_\tau(x,u,y,z)/w_\tau(x,u)$, if $w_\tau(x,u) > 0$ and chosen arbitrary otherwise. In other terms, if each state-action pair is visited infinitely often, then by the strong law of large number, $\lim_{\tau\to\infty} P_\tau = P$. To do so, we make use of an exploration strategy, namely `probe`. To better understand the probing exploration policy, let $U(x) = \{1,\ldots,|U(x)|\}$ be the set of available actions in state $x\in X$, and $\sigma(x)$ be the number of actions to be experienced in state $x\in X$. Before a new estimation phase starts, we set $\sigma(x) = |U(x)|$ for every state $x\in X$. At each time step of the model estimation phase, if $\sigma(x) > 0$, the centralized coordinator executes action $\sigma(x)$ in state x and decrements $\sigma(x)$; otherwise, he or she selects the action which minimizes the difference between the estimated and the optimized exploration policies (updated at the improvement phase), denoted $\hat{\pi}_\tau$ and π_τ, respectively: for any arbitrary $x\in X$ and vector of counts σ,

$$\texttt{probe}(x,\sigma) \doteq \begin{cases} \sigma(x), \text{ if } \sigma(x) > 0 \\ \arg\min_{u\in U(x)}\{\hat{\pi}_\tau(u|x) - \pi_\tau(u|x)\}, \text{ otherwise.} \end{cases}$$

If the state space forms a single positive recurrent class under any stationary policy, `probe` ensures every state-action pair gets visited at least once at each model estimation phase, in which case the estimation phase terminates. Otherwise, we shall impose a training budget $\tau_{\max}$ during each model estimation phase. Once the budget is exhausted the estimation phase stops—in that case, there is no guarantee of visiting every state-action pair. Next, the algorithm proceeds to the policy-improvement phase.

Each policy-improvement phase starts by computing an extended occupancy measure ζ_τ for the current estimate dynamics P_τ using our MILP formulations. Then, it calculates the state-dependent exploration policy as follows:

$$\pi_\tau(u|x) = \textstyle\sum_o \zeta_\tau(x,o,u) / \sum_{x,o}\zeta_\tau(x,o,u). \tag{12}$$

Next, it ensures $\hat{\pi}_\tau \doteq \{\hat{\pi}_\tau(u|x)\}$ is a consistent estimator of π_τ, where $\hat{\pi}_\tau(u|x) \doteq w_\tau(x,u)/w_\tau(x)$, if $w_\tau(x) > 0$; and 0 otherwise. To this end, it explores the state space by selecting the action which minimizes the difference between the estimated and the optimized exploration policies, until $\|\hat{\pi}_\tau - \pi_\tau\|_\infty$ goes below a certain threshold. It is worth noticing that that this algorithm requires no hyper-parameter tuning. We present the pseudocode of our `probing` algorithm in the supplementary material.

7 Experiments

This section empirically demonstrates and validates the scalability of the proposed planning and learning approach w.r.t. the number of agents for $\alpha = 0.9$. We show that our planning and learning approach applies to n-agent Dec-POMDPs where no other MILP formulation does. We run our experiments on Intel(R) Xeon(R) CPU E5-2623 v3 3.00GHz.

7.1 ND-POMDPs

Setup. We conduct experiments on well-established benchmarks for evaluating n-agent Dec-POMDPs, *i.e.* network-distributed domains based on the sensor network applications [Nair et al., 2005], which range from four to fifteen agents. The reader interested in the description of

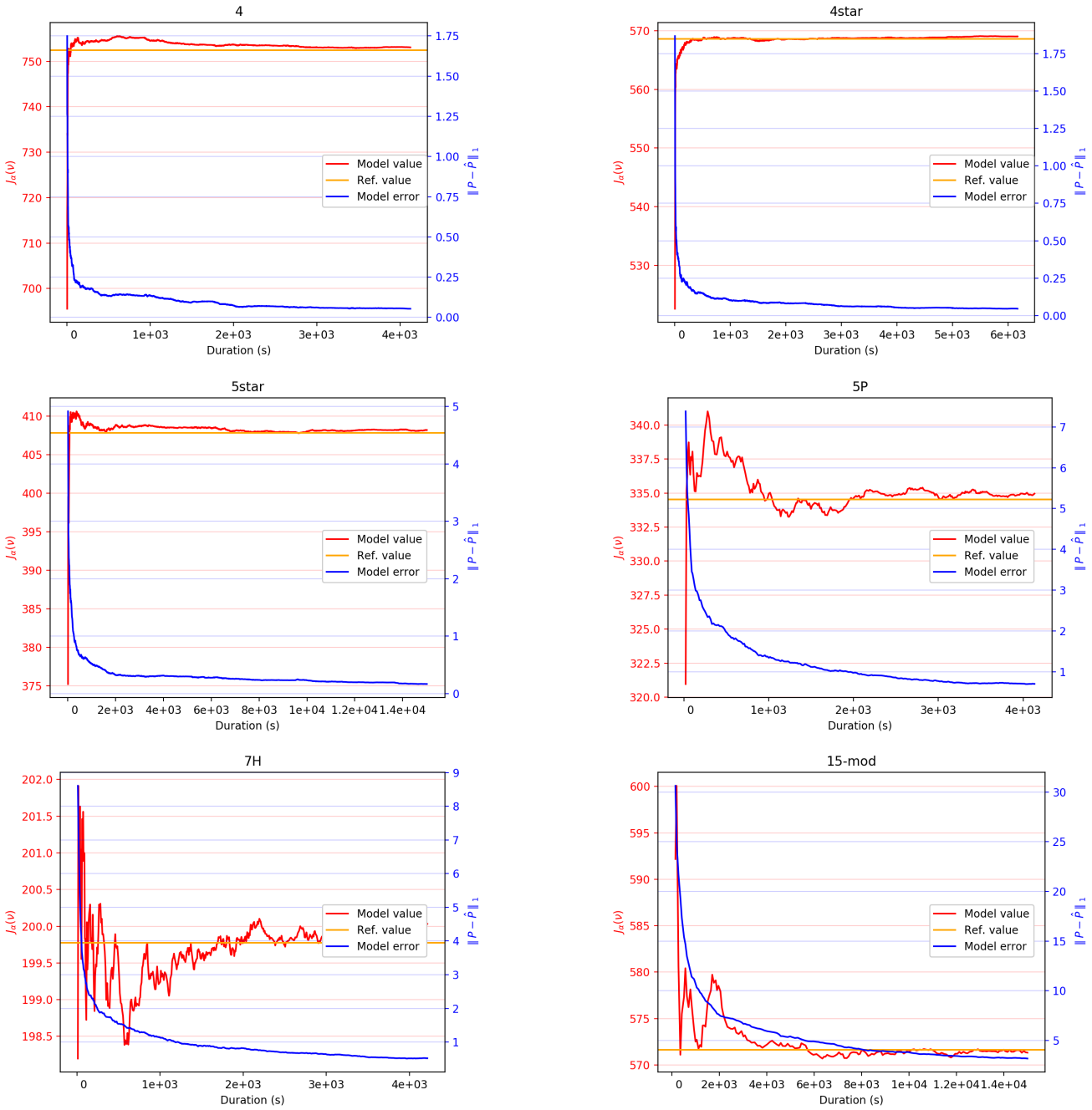

Figure 1: `Probing` results for n-agent ∞-horizon ND-POMDPs with $\alpha = 0.9$.

Algorithm	$\|a\|$	Time	$J_\alpha(a;\nu)$
4 domain — $\|X\| = 12$, $n = 4$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 3$			
$\mathcal{P}_4(\ell = 1)$	3×3	0.37s	752.492
$\mathcal{P}_4(\ell = 2)$	5×5	0.39s	752.492
4-star domain — $\|X\| = 12$, $n = 4$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 3$			
$\mathcal{P}_4(\ell = 1)$	3×3	0.279s	568.642
$\mathcal{P}_4(\ell = 2)$	5×5	0.298s	568.642
5-star domain — $\|X\| = 12$, $n = 5$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 3$			
$\mathcal{P}_4(\ell = 1)$	3×3	0.932s	407.821
$\mathcal{P}_4(\ell = 2)$	5×5	1.241s	407.821
5-P domain — $\|X\| = 12$, $n = 5$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 3$			
$\mathcal{P}_4(\ell = 1)$	3×3	7.12s	334.536
$\mathcal{P}_4(\ell = 2)$	5×5	10.01s	334.536
7-H domain — $\|X\| = 12$; $n = 7$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 3$			
$\mathcal{P}_4(\ell = 1)$	3×3	5.10s	199.773
$\mathcal{P}_4(\ell = 2)$	5×5	7.63s	199.773
15-3D domain — $\|X\| = 60$; $n = 15$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 4$			
$\mathcal{P}_4(\ell = 1)$	3×3	1328.48s	409.069
$\mathcal{P}_4(\ell = 2)$	5×5	3002.19s	409.069
15-Mod domain — $\|X\| = 16$; $n = 15$, $\|Z_i\| = 2$, *and* $2 \leq \|U_i\| \leq 4$			
$\mathcal{P}_4(\ell = 1)$	3×3	27.1845s	571.642
$\mathcal{P}_4(\ell = 2)$	5×5	59.238s	571.642

Table 1: MILP results for n-agent ∞-horizon ND-POMDPs. Higher $J_\alpha(a;\nu)$ is better.

the benchmarks can refer to `http://teamcore.usc.edu/projects/dpomdp/`. To the best of our knowledge, no other MILP formulation can solve these domains, *e.g.,* [Kumar et al., 2016] is inapplicable. Alternative approaches include: an extension of FB-HSVI for network-distributed domains [Dibangoye et al., 2014a], unfortunately the only available formulation is dedicated for finite-horizon settings; and local search methods, *e.g.,* [Kumar et al., 2011], which (i) can only provide local optima; (ii) trade theoretical guarantees for scalability w.r.t. the number of states and (iii) go beyond the scope of this paper. Instead, we target scalability w.r.t. the number of agents while preserving theoretical guarantees over a selected class of decentralized and deterministic ℓ-order Markov policies. Table 1 reports results on all tested n-agent domains. To validate the potential of this learning approach, we demonstrate one can learn the exact model and a corresponding policy in Figure 1.

Analysis. Experiments show the ability for our MILP formulations to quickly find finite-memory decentralized policies that may serve as good approximations of the optimal decentralized policy. They also demonstrate the scalability with respect to the number of agents. Our MILP formulations optimally solve all network-distributed domains with up to fifteen agents. It is worth noticing that both planning and learning processes, the main limitation is the lack of scalability w.r.t. the number of states, actions, observations, and hence histories. We argue that for domains where the double-exponential number of histories affect the scalability far more strongly than the number of states and actions, our approach is particularly useful. In future work, we plan to learn a low-dimensional representation of the model of world making it possible to apply our approach even when facing domains with larger state, observation and action spaces.

7.2 Two-agent Dec-POMDPs

Algorithm	$\|a\|$	Time	$J_\alpha(a;\nu)$
Broadcast ($\|X\| = 4$, $\|U_i\| = 2$, $\|Z_i\| = 2$)			
$\mathcal{P}_2(\ell = 1)$	3×3	0.01s	9.19
$\mathcal{P}_2(\ell = 2)$	5×5	0.01s	9.2629
FB-HSVI	102	19.8s	9.271
FB-HSVI($\delta = 0.01$)	435	7.8s	9.269
Kumar et al. [2016]	3×3	0.05s	9.1
Dec-tiger ($\|X\| = 2$, $\|U_i\| = 3$, $\|Z_i\| = 2$)			
FB-HSVI($\delta = 0.01$)	52	6s	13.448
FB-HSVI	25	157.3s	13.448
Kumar et al. [2016]	7×7	4.2s	13.4
MPBVI	231	< 18000s	13.448
EM	6	142s	-16.3
$\mathcal{P}_2(\ell = 2)$	5×5	0.01s	-20
Recycling robots ($\|X\| = 4$, $\|U_i\| = 3$, $\|Z_i\| = 2$)			
$\mathcal{P}_3(\ell = 1)$	3×3	0.01s	31.9291
FB-HSVI	109	2.6s	31.929
FB-HSVI($\delta = 0.01$)	108	0s	31.928
Kumar et al. [2016]	3×3	1.1s	31.9
EM	2	13s	31.50
Meeting in a 3x3 grid ($\|X\| = 81$, $\|U_i\| = 5$, $\|Z_i\| = 9$)			
$\mathcal{P}_3(\ell = 1)$	10×10	0.19s	5.81987
FB-HSVI	108	67s	5.802
FB-HSVI($\delta = 0.01$)	88	45s	5.794
Kumar et al. [2016]	10×10	4.4s	5.8
Box-pushing ($\|X\| = 100$, $\|U_i\| = 4$, $\|Z_i\| = 5$)			
FB-HSVI($\delta = 0.01$)	331	1715.1s	224.43
FB-HSVI($\delta = 0.05$)	288	1405.7s	224.26
Kumar et al. [2016]	7×8	6.2s	181.2
$\mathcal{P}_2(\ell = 1)$	6×6	0.06s	181.985
$\mathcal{P}_2(\ell = 2)$	26×26	1.86s	197.607
Mars rover ($\|X\| = 256$, $\|U_i\| = 6$, $\|Z_i\| = 8$)			
FB-HSVI($\delta = 0.01$)	136	74.31s	26.94
FB-HSVI($\delta = 0.2$)	149	85.72s	26.92
Kumar et al. [2016]	9×9	20.2s	23.8
$\mathcal{P}_2(\ell = 1)$	9×9	2.48s	23.8302
EM	3	5096s	17.75

Table 2: Results for infinite-horizon domains. Higher $J_\alpha(a;\nu)$ is better. δ and ℓ denote the regret in Bellman's backup and the class of policies, respectively.

Setup. For the sake of completeness we also present our performances for 2-agent Dec-POMDPs, where we provide competitive results w.r.t. state-of-the-art methods. Experiments were conducted on standard benchmarks, all available at `masplan.org`. We use two of our 2-agent MILP formulations, *i.e.,* $\mathcal{P}_2(\ell)$ and $\mathcal{P}_3(\ell = 1)$, for $\ell \in \{1, 2\}$. Though our MILP formulations are guaranteed to find an optimal solution in the target class of policies, we do not expect them to do always better than the state-of-the-art solver FB-HSVI, since the latter achieves provably a near-optimal performance on these benchmarks. Instead, these domains serve for the sanity check, assessing the quality of our solutions w.r.t. near-optimal ones. We also report performances from other Dec-POMDP solvers, *e.g.,* [Kumar et al., 2016] and EM.

Analysis. Results for EM [Kumar et al., 2016] were likely computed on different platforms, and, therefore, time comparisons may be approximate at best. Results for 2-agent domains can be seen in Table 2. In many tested 2-agent domains, low-order Markov policies achieve good performances. Hence our MILP formulations are competitive to state-of-the-art algorithms. In decentralized MDPs, *e.g.,* recycling robots and meeting in a 3x3 grid, decentralized and deterministic Markov policies are optimal, non-

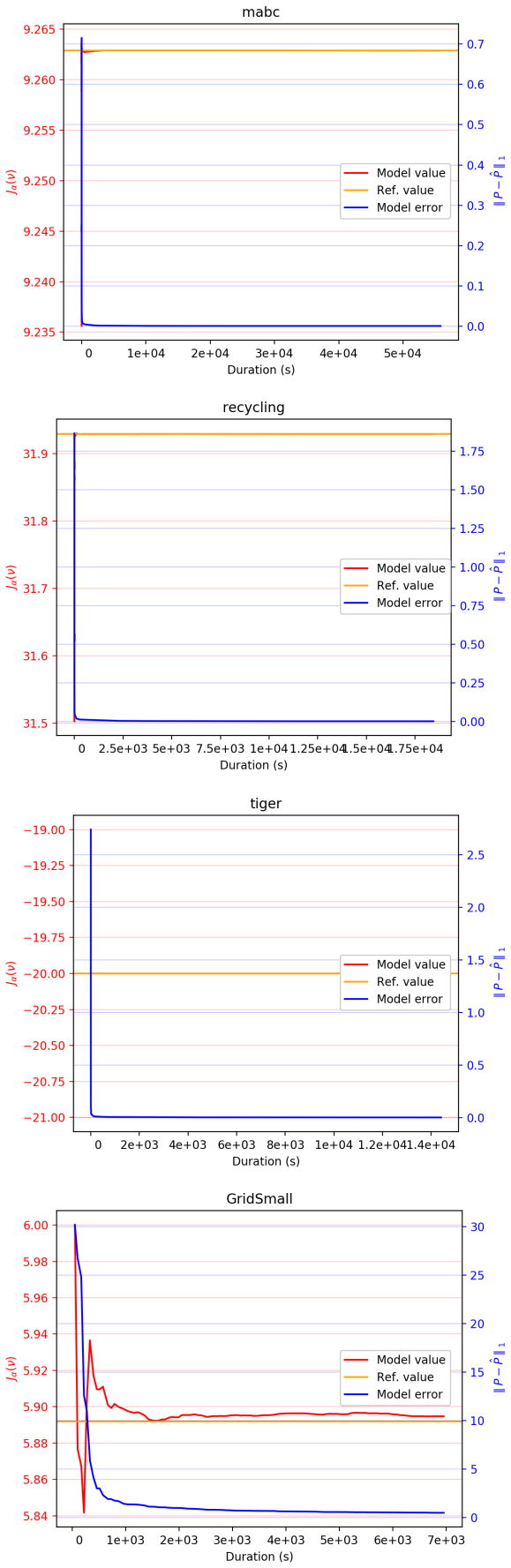

Figure 2: `Probing` results for n-agent ∞-horizon 2-agent Dec-POMDPs.

surprisingly $\mathcal{P}_3$ can find optimal solutions. However, in domains requiring more memory, *i.e.,* higher-order Markov policies, our MILP formulations may achieve poor performances, see for example Dec-tiger. The difficulty comes from the exponentially many joint histories to consider as ℓ increases. Overall, MILP formulations provide a simple yet efficient alternative to solving 2-agent domains, especially when (i) finite-memory policies can achieve good performances; and (ii) states, actions and observations are small. Finally, we successfully run the adaptive decentralized control approach on two small domains, *i.e.* recycling robots and broadcast see Figure 2, providing a strong theoretical support for this promising approach.

8 Conclusion

In this paper, we investigated MILP formulations, which proved useful for partially observable domains, but existing applications restrict to benchmarks with one or two agents. To overcome this limitation, we introduced a novel linearization property that allows us to reformulate nonlinear constraints from general decentralized partially observable Markov decision processes into linear ones. We further presented MILP formulations for general and special cases, including network-distributed and transition-independent problems. Our experiments on both standard 2-agent benchmarks as well as domains with a large number of agents illustrate the ability for our planning and learning approaches based on MILP formulations to find good approximate solutions often and sometimes optimal ones. Yet, the scalability, w.r.t. states, actions and observations, remains a major limitation. In future work, we shall generate an approximation of the extended occupancy measures within a parameterized class of functions to deal with the intractability of the exact MILP formulation, in a spirit similar to that of statistical regression. We shall draw inspiration from the literature of deep generative models, and more specifically (discrete) variational autoencoders [Kingma and Welling, 2014, Ha and Schmidhuber, 2018]. Besides, we plan to apply standard decomposition methods in the literature of discrete optimization.

Acknowledgements

This work was supported by ANR project PLASMA, "Planning and Learning to Act in Systems of Multiple Agents", under Grant 19-CE23-0018-01.

References

E. Altman. *Constrained Markov Decision Processes*. Stochastic Modeling Series. Taylor & Francis, 1999. ISBN 9780849303821.

C. Amato, D. S. Bernstein, and S. Zilberstein. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *JAAMAS*, 21(3):293–320, 2010.

R. Aras and A. Dutech. An Investigation into Mathe-

matical Programming for Finite Horizon Decentralized POMDPs. *JAIR*, 37:329–396, 2010.

R. Becker, S. Zilberstein, V. R. Lesser, and C. V. Goldman. Solving Transition Independent Decentralized Markov Decision Processes. *JAIR*, 22:423–455, 2004.

D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The Complexity of Decentralized Control of Markov Decision Processes. *Mathematics of Operations Research*, 27(4), 2002.

T. Berthold, S. Heinz, and M. E. Pfetsch. Nonlinear pseudo-boolean optimization: relaxation or propagation? Technical Report 09-11, ZIB, Takustr. 7, 14195 Berlin, 2009.

J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Optimally Solving Dec-POMDPs As Continuous-state MDPs. In *IJCAI*, pages 90–96, 2013.

J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Exploiting Separability in Multi-Agent Planning with Continuous-State MDPs. In *AAMAS*, pages 1281–1288, 2014a.

J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Optimally solving Dec-POMDPs as Continuous-State MDPs: Theory and Algorithms. Research Report RR-8517, INRIA, 2014b.

C. V. Goldman and S. Zilberstein. Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis. *JAIR*, 22(1):143–174, 2004. ISSN 1076-9757.

D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In *NeurIPS*, pages 2450–2462, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014.

A. Kumar, S. Zilberstein, and M. Toussaint. Scalable Multiagent Planning Using Probabilistic Inference. In *AAAI*, pages 2140–2146, 2011.

A. Kumar, S. Zilberstein, and M. Toussaint. Probabilistic Inference Techniques for Scalable Multiagent Decision Making. *JAIR*, 53:223–270, 2015.

A. Kumar, H. Mostafa, and S. Zilberstein. Dual formulations for optimizing Dec-POMDP controllers. In *ICAPS*, pages 202–210, 2016.

H. Mostafa and V. Lesser. Compact mathematical programs for dec-mdps with structured agent interactions. In *UAI*, pages 523–530, 2011.

R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked Distributed POMDPs: A Synthesis of Distributed Constraint Optimization and POMDPs. In *AAAI*, pages 133–139, 2005.

F. A. Oliehoek. Sufficient Plan-Time Statistics for Decentralized POMDPs. In *IJCAI*, 2013.

F. A. Oliehoek, M. T. J. Spaan, and N. A. Vlassis. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *JAIR*, 32:289–353, 2008.

S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *JAAMAS*, 17(2):190–250, 2008.

D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A Heuristic Search Algorithm for Solving Decentralized POMDPs. In *UAI*, 2005.

S. Witwicki and E. Durfee. Commitment-driven distributed joint policy search. In *AAMAS*, pages 75:1–75:8, 2007.

J. Wu and E. H. Durfee. Mixed-integer linear programming for transition-independent decentralized MDPs. In *AAMAS*, pages 1058–1060, 2006.

A Proofs

Lemma 1. *$s_\alpha(\nu, a)$ is a probability distribution.*

Proof. Let $\boldsymbol{e}$ be a vector of all ones. Then we have

$$\begin{aligned}
s_\alpha(\nu,a)^\top e &\doteq \left((1-\alpha)\nu^\top \sum_{\tau=0}^{\infty} \alpha^\tau (P^a)^\tau\right)\boldsymbol{e} \\
&= (1-\alpha)\nu^\top \sum_{\tau=0}^{\infty} \alpha^\tau \boldsymbol{e} \\
&= (1-\alpha)\nu^\top (1-\alpha)^{-1}\boldsymbol{e} && \text{given } \sum_{\tau=0}^{\infty}\alpha^\tau = (1-\alpha)^{-1} \\
&= \nu^\top \boldsymbol{e} \\
&= 1 && \nu \text{ being a probability distribution}
\end{aligned}$$

and the claim follows. □

Lemma 2. *$s_\alpha(\nu, a)$ is a sufficient statistic for estimating infinite-horizon normalized discounted reward:*

$$J_\alpha(a;\nu) = \mathbb{E}^a_{s_\alpha(\nu,a)}\{r(x,u)\}. \tag{4}$$

Proof. Let measure $\nu^\top (P^a)^\tau$ be the probability distribution over state-history pairs conditional on initial state distribution ν and policy a, after τ decision epochs. Then we have

$$\begin{aligned}
J_\alpha(a;\nu) &\doteq (1-\alpha)\sum_{\tau=0}^{\infty}\alpha^\tau E^a_\nu\{r(x_\tau,u_\tau)\} \\
&= (1-\alpha)\sum_{\tau=0}^{\infty}\alpha^\tau E\{r(x_\tau,u_\tau) \mid \{(x_\tau,o_\tau)\sim\nu^\top(P^a)^\tau, u_\tau\sim a(o_\tau,\cdot)\} \\
&= (1-\alpha)\sum_{\tau=0}^{\infty}\alpha^\tau \sum_{x\in X}\sum_{o\in O}(\nu^\top(P^a)^\tau)(x_\tau=x,o_\tau=o)\sum_{u\in U}a(o,u)\cdot r(x,u) \\
&= \sum_{x\in X}\sum_{u\in U} r(x,u)\sum_{o\in O}a(o,u)\left((1-\alpha)\nu^\top\sum_{\tau=0}^{\infty}\alpha^\tau(P^a)^\tau\right)(x_\tau=x,o_\tau=o) \\
&= \sum_{x\in X}\sum_{u\in U} r(x,u)\sum_{o\in O}a(o,u)\cdot s_\alpha(\nu,a;x,o) \\
&\doteq E^a_{s_\alpha(\nu,a)}\{r(x,u)\}
\end{aligned}$$

and the claim follows. □

Lemma 3. *Occupancy measure $s_\alpha(\nu, a)$ is the solution of the following linear equation w.r.t. $s_\alpha(\nu, a)$:*

$$s_\alpha(\nu,a)^\top(I-\alpha P^a) = (1-\alpha)\nu^\top. \tag{5}$$

Proof. If we let I be the identity matrix, the following holds:

$$\sum_{\tau=0}^{\infty}\alpha^\tau P^a_\nu(\tau) = \nu^\top\sum_{\tau=0}^{\infty}\alpha^\tau(P^a)^\tau \ = \nu^\top(I-\alpha P^a)^{-1}. \tag{13}$$

Injecting (13) into (3), and re-arranging terms lead to a linear characterization of occupancy measures:

$$s_\alpha(\nu,a)^\top(I-\alpha P^a) = (1-\alpha)\nu^\top.$$

Which ends the proof. □

Lemma 4. *If we let v_1, v_2, and w be Boolean, random, and non-negative variables, respectively; and*

$$(C_1) \left| \begin{array}{ll} w - v_k & \leq 0 \quad \forall k \in \{1, 2\} \\ v_1 + v_2 - w & \leq 1 \end{array} \right.$$

then solutions of polyhedron (C_1) satisfy $w = v_1 \cdot v_2$.

Proof. Building upon [Berthold et al., 2009], the proof proceeds by considering all possible values for v_1.

1. If $v_1 = 0$, then from $w - v_k \leq 0$ and $w \in [0, 1]$, we know that $w = 0$ no matter v_2, which satisfies $w = v_1 \cdot v_2$.
2. On the other hand, if $v_1 = 1$, then from $v_1 + v_2 - w \leq 1$ and $w \in [0, 1]$, we know that $w \leq 1$ for $v_1 = v_2$, which is further tightened when considering $v_1 = v_2$, *i.e.,* $w \leq v_2$. The last inequality $v_1 + v_2 - w \leq 1$ shows that $w \geq v_2$. As a consequence $w = v_2$, which satisfies $w = v_1 \cdot v_2$.

Which ends the proof. □