Optimally Solving Two-Agent Decentralized POMDPs Under One-Sided Information Sharing

Yuxuan Xie, Jilles Steeve Dibangoye, Olivier Buffet

Optimally Solving Two-Agent Decentralized POMDPs Under One-Sided Information Sharing

Yuxuan Xie [1] **Jilles S. Dibangoye** [1] **Olivier Buffet** [2]

Abstract

Optimally solving decentralized partially observable Markov decision processes (Dec-POMDPs) under either full or no information sharing received significant attention in recent years. However, little is known about how partial information sharing affects existing theory and algorithms. This paper addresses this question for a team of two agents, with one-sided information sharing, *i.e.* both agents have imperfect information about the state of the world, but only one has access to what the other sees and does. From the perspective of a central planner, we show that the original problem can be reformulated into an equivalent information-state Markov decision process and solved as such. Besides, we prove that the optimal value function exhibits a specific form of uniform continuity. We also present heuristic search algorithms utilizing this property and providing the first results for this family of problems.

1. Introduction

Over the last few years, Dec-POMDPs have been used as the underlying semantics for (optimal) planning and reinforcement learning in sequential decision making by a team of collaborative agents (Foerster et al., 2018; Rashid et al., 2018; Dibangoye & Buffet, 2018; Bard et al., 2020). In this setting, every agent acts simultaneously *but* can neither see the actual state of the world nor explicitly communicate its observations with each other, due to communication cost, latency, or noise (Bernstein et al., 2002). The critical problem with this assumption is that every decision variable, at a given point in time, directly influences any other one at the same point. The mutual influence of decision variables along with their double exponential growth with agents and time explain the worst-case complexity, *i.e.* infinite-horizon cases are undecidable, finite-horizon ones are NEXP-hard, and finding approximations remains hard (Bernstein et al., 2002).

Because in Dec-POMDPs, no agent can unilaterally choose its optimal policy, Dibangoye et al. (2014b) adopted the viewpoint of a central planner and recast the original problem into an equivalent non-observable MDP, namely Occupancy MDPs (oMDPs). This framework allows reasoning, all at once, about mutually dependent decision variables. In this approach, states are sufficient statistics to jointly find an optimal assignment to all mutually dependent decision variables (Szer et al., 2005; Oliehoek, 2013; Nayyar et al., 2013). In principle, theory and algorithms to optimally solving MDPs can apply. Unfortunately, given the double exponential growth of mutually dependent decision variables with agents and time, even a single update or *backup* of a state can be prohibitively expensive (Szer et al., 2005; Seuken & Zilberstein, 2008; Oliehoek et al., 2013; MacDermed & Isbell, 2013; Dibangoye et al., 2014c; Kumar et al., 2015).

In many cases, however, real-world multi-agent environments contain significant structure. Indeed, several forms of structure have been investigated in the past, ranging from loosely coupled dynamics (Becker et al., 2004) and rewards (Nair et al., 2005) to delayed and full information sharing (Nayyar et al., 2011). Algorithms that take advantage of this structure can optimally solve structured problems much faster than generic ones (Goldman & Zilberstein, 2004; Oliehoek & Spaan, 2012; Dibangoye et al., 2014a).

While Dec-POMDPs generally assume every agent acts *without* full knowledge of what others observe or plan to do, in many cases, one agent has access to what the others see and do. In hierarchical organizational structures, for example, each agent has virtually full knowledge of what its immediate subordinates see and do. After one agent takes a decision, its direct subordinates cannot change theirs. Though these characteristics are embodied in many cooperative real-world applications, including military systems, corporations, and governments, little can be said so far, the only exceptions being Hadfield-Menell et al. (2016); Malik

[1]Univ Lyon, INSA Lyon, INRIA, CITI, F-69621 Villeurbanne, France [2]Université de Lorraine, INRIA, CNRS, LORIA, F-54000 Nancy, France. Correspondence to: Jilles S. Dibangoye <jilles-steeve.dibangoye@insa-lyon.fr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

et al. (2018). They considered scenarios where only one agent has access to (i) the actual state of the world, and (ii) the action and observation of the other agent. Such a setting allows one to reason directly upon the belief state—*i.e.* a probability distribution over the states of the world. A similar line of research—beyond the scope of this paper—has been pursued in (mostly two-player zero-sum) partially observable stochastic games (Ghosh et al., 2004; Horák et al., 2017; Horák & Bošanskỳ, 2019).

In this paper, we investigate the problem of optimally solving two-agent Dec-POMDPs under one-sided information sharing (one-sidedness for short)—*i.e.* both agents have imperfect information about the state of the world, but only one agent has public actions and observations. From the central planner's perspective, public actions and observations allow us to move from non-observable *o*MDPs to partially observable ones, branching on possible public action-observation pairs. Doing so makes it possible to apply the theory and algorithms for *o*MDPs (Dibangoye et al., 2014b), albeit on (i) more concise state representations, *e.g.* probability distributions over belief states, and (ii) a piecewise-linear and convex (PWLC) value function of states. Perhaps the main result of this paper is the proof that, under one-sidedness, one can move from a PWLC value function to a linear one, though in a higher-dimensional space. Exploiting this property leads to improved scalability mainly because decision variables that previously influence one another can now be decoupled and independently processed. To support our findings, we provide three variants of the Heuristic Search Value Iteration (HSVI) algorithm (Smith, 2007) using either PWLC or linear value-function representations and compare them on standard problems from the literature.

2. Optimally Solving Dec-POMDPs

We begin with an overview of the multi-agent Dec-POMDP formalization. Then its single-agent reformulation, that allows single-agent theory and algorithms to apply.

2.1. Multi-agent Formulation

Definition 1 (Bernstein et al.)**.** *A 2-agent Dec-POMDP is given by $M \doteq \langle X, U, Z, r, p\rangle$, where X is a finite set of hidden states; U^i is a finite action set of agent i, where $U = U^1 \times U^2$ specifies the set of joint actions $u = (u^1, u^2)$; Z^i is an observation set of agent i, where $Z = Z^1 \times Z^2$ specifies the set of joint observations $z = (z^1, z^2)$; p describes a transition function with conditional probability distribution $p^{u,z}_{x,y}$ defining the probability of transitioning from state x to y after taking joint action u and seeing z; and r is a reward model with immediate reward $r(x, u)$.*

Throughout the paper, we make the following assumptions.

Assumption 1. *Agent 2 has public actions and observations,* i.e. $(u^2_\tau, z^2_{\tau+1}) \subseteq z^1_{\tau+1}$*, for every point in time τ.*

Assumption 2. *Rewards are two-side bounded,* i.e. *there exists some $c > 0$, such that $\|r(\cdot,\cdot)\|_\infty \leq c$.*

Assumption 3. *Planning horizon ℓ is finite, since ∞-horizon solutions are ϵ-close to ℓ-horizon optimal solutions, where $\ell = \lceil \log_\gamma (1-\gamma)\epsilon/c \rceil$, for discount factor $\gamma \in [0, 1)$ and some positive scalar ϵ.*

Optimally solving M aims at finding joint policy π, *i.e.* a n-tuple of sequences of private decision rules, one per agent,

$$\pi \doteq (a^1_{0:\ell}, a^2_{0:\ell}),$$

maximizing the expected γ-discounted cumulative rewards starting at initial state distribution b_0 onward, and given by

$$\upsilon_0(b_0; \pi) \doteq E\{\textstyle\sum_{\tau=0}^{\ell-1} \gamma^\tau r(x_\tau, u_\tau) \mid b_0, \pi\}.$$

For each agent i, private decision rule $a^i_\tau \colon o^i_\tau \mapsto u^i_\tau$ depends on τth histories $o^i_\tau \doteq (u^i_{0:\tau-1}, z^i_{1:\tau})$, with 0th private history being $o^i_0 \doteq \emptyset$. Unfortunately, optimally solving M in its multi-agent formulation is non-trivial, since it is not clear how to define a *right notion of state* (Hansen et al., 2004).

To better understand this, notice that every agent acts simultaneously, but can neither see the actual state of the world nor explicitly communicate its actions and observations with each other. As a consequence, what one agent sees and does directly affect what the others see and do, thus the mutual influence of all decision variables $a_\tau \doteq (a^1_\tau, a^2_\tau)$ at each time step τ. The motivation for a single-agent reformulation is twofold. The primary reason is that it allows us to reason simultaneously about all mutually dependent decision variables $a_\tau \colon o_\tau \mapsto u_\tau$, a set we shall refer to as a τth joint decision rule, *i.e.* a mapping from joint histories $o_\tau \doteq (o^1_\tau, o^2_\tau)$ to joint actions $u_\tau \doteq (u^1_\tau, u^2_\tau)$. Besides, it eases the transfer of theory and algorithms from single- to multi-agent systems.

2.2. Single-agent Reformulation

This equivalent reformulation aims at recasting M from the perspective of an offline central planner. Every point in time, this planner acts all at once on behalf of all agents; taking a joint decision rule, but receives no feedback. The history of selected joint decision rules, *i.e.* a joint policy, describes a non-observable MDP, namely occupancy MDP.

Definition 2 (Dibangoye et al.)**.** *An occupancy MDP w.r.t. M is given by a tuple $\hat{M} \doteq \langle \hat{S}, \hat{A}, \hat{T}, \hat{R}\rangle$ where S is the (occupancy-)state space, where (occupancy) states are conditional probability distribution over hidden states and joint histories given a joint policy followed so far; $\hat{A}$ is the space of actions describing joint decision rules; $\hat{T} \colon \hat{S} \times \hat{A} \mapsto \hat{S}$ is the transition rule, where $\hat{s}_{\tau+1} \doteq \hat{T}(\hat{s}_\tau, \hat{a}_\tau)$,*

$$\hat{s}_{\tau+1}(y, (o, u, z)) \doteq \textstyle\sum_{x \in X} \hat{s}_\tau(x, o) \cdot \delta^u_{\hat{a}_\tau(o)} \cdot p^{u,z}_{x,y}$$

$\hat{R}\colon \hat{S} \times \hat{A} \mapsto \mathbb{R}$ *describes the linear reward function,* i.e.

$$\hat{R}(\hat{s}_\tau, \hat{a}_\tau) \doteq \sum_{x \in X} \sum_{o \in O} \hat{s}_\tau(x, o) \cdot r(x, \hat{a}_\tau(o)).$$

Optimally solving $\hat{M}$ aims at finding value functions $(v^*_\tau)_\tau$, *i.e.* mappings states to reals, and solution of Bellman optimality equations: for every $\hat{s} \in \hat{S}$,

$$v^*_\tau(\hat{s}) = \max_{\hat{a} \in \hat{A}(\hat{s})} \left[\hat{R}(\hat{s}, \hat{a}) + \gamma v^*_{\tau+1}(\hat{T}(\hat{s}, \hat{a}))\right] \quad (1)$$

with boundary condition $v^*_\ell(\cdot) = 0$. Given the optimal value function, one can greedily select an optimal joint policy. Unfortunately, solving Bellman optimality equations (1) is not feasible since state space $\hat{S}$ describes a continuum. Instead, Dibangoye et al. (2014b) build on the fact that the optimal value function is a piecewise linear and convex function of the state space.

Lemma 1 (Dibangoye et al.)**.** *When optimized exactly an optimal value function of the Bellman optimality equation is always a piecewise linear and convex in the history-occupancy space,* i.e. *for every s_τ,*

$$v^*_\tau(\hat{s}_\tau) = \max_{\zeta_\tau \in \Gamma_\tau} \sum_{(x,o) \in \triangle(\hat{s}_\tau)} \hat{s}_\tau(x, o) \cdot \zeta_\tau(x, o)$$

where Γ_τ is a finite set of vectors ζ_τ in the probability space defined by sample space $X \times O_\tau$, the σ-algebra $X \times O_\tau$, and $\triangle(\hat{s}_\tau)$ is the set of state and joint history pairs with non-zero probability w.r.t. to $\hat{s}_\tau$.

Also, they introduced a backup operator that can circumvent the exhaustive enumeration of all joint decision rules using mixed-integer linear programs. To enhance value generalization from one state to another one, they provided equivalence relations among private histories. Altogether, these operations made it possible to use a couple of single-agent algorithms to solve multi-agent problems. But, the scalability remains the major issue.

2.3. Limitations w.r.t. One-Sidedness

The theory above also applies under one-sidedness, but the curse of dimensionality restricts its scalability in the face of domains of a practical scale. To better understand this, notice that its complexity depends on two operators: the *Bellman backup operator* necessary to improve the value function; and the *estimation operator* useful to maintain history-occupancy states. In either case, this theory is not geared to exploit the one-sidedness. Thus, it is typical to have to consider history-occupancy states and value functions over exponentially many variables, though multiple variables may have little influence on one another.

Performing the exact Bellman backup operator is infeasible because the history-occupancy space is a continuum. Instead, Dibangoye et al. (2014b) suggest employing point-based Bellman backup operator for each encountered history-occupancy state. Still, each application of this operator requires enumerating exponentially many joint decision rules, which limits its applicability. Mixed-integer linear programming formulations of the point-based Bellman backup operator exist, but the scalability remains limited. The reason is not only the time required to perform the backup but also the time necessary to encode the program. The estimation operator is also cumbersome. Indeed, maintaining history occupancy states, *i.e.* probability distributions over states and joint histories, requires enumerating exponentially many joint histories. Besides, this operator quickly becomes intractable with large planning horizons, let alone the infinite planning horizon. Finally, the generalization of values from one history occupancy state to another one applies whenever they share the same support, *e.g.* the *sawtooth approximation* (Hauskrecht, 2000; Smith, 2007; Dibangoye et al., 2014b).

In this paper, we investigate more concise sufficient statistics and a PWLC property about the optimal value function by fully exploiting the one-sidedness. In other words, we address the following question—*how can we improve the representation of history occupancy states and the PWLC value function and make backups more efficient to optimally solving M under one-sidedness?* Recently, Hadfield-Menell et al. (2016) and Malik et al. (2018) investigated the same question but for M with one-sided partial observability, *i.e.* agent 1 (w.l.o.g.) has access to the state of the world and agent 2's actions and observations are public. In such a setting, they demonstrated history-occupancy states could be restricted to belief states and point-based Bellman operator made more efficient. While their approach does not apply in the general case we target, we nonetheless establish strong connections. In particular, we show that under collective full observability, *i.e.* joint observations reveal the actual state of the world, our approach is equivalent to theirs.

3. From Non- to Partially Observable oMDPs

According to the centralized planning for decentralized execution theory, a central planner with no observations about what agents see and do at the online execution phase, can nonetheless plan on behalf of all agents at the offline planning phase. To fully exploit the one-sidedness assumption, we slightly relax this theory in the remainder.

In general, no agent can unilaterally choose its optimal policy because there is no explicit information sharing. This assumption explains Dibangoye et al. (2014b) adopted the viewpoint of a central planner and recast M into equivalent non-observable oMDP, where sufficient statistics are history occupancy states. Whatever actions or observations are made public to all agents allow, however, recasting M into a partially observable oMDP, thus branching out according to

possible public actions/observations histories, and reasoning on more concise sufficient statistics.

3.1. Optimality of Belief-Dependent Policies

The statistics summarizing the information available to the central planner about the process are called information states. A straightforward application of Dibangoye et al. (2014b) shows that the one-sidedness assumption allows (i) branching on histories of agent 2 and (ii) using, as an information state at time τ, the probability distribution over hidden states and agent 1's histories given history occupancy state and agent 2's history, *i.e.*

$$\iota_\tau(x_\tau, o^1_\tau) \doteq P(x_\tau, o^1_\tau \mid \hat{s}_\tau, o^2_\tau).$$

Also, one can prove that (i) this information state is sufficient to estimate immediate rewards, and (ii) next information state $\iota_{\tau+1} \doteq T(\iota_\tau, \hat{a}_\tau, z^2)$ depends on current one ι_τ, joint decision rule $\hat{a}_\tau \doteq (\hat{a}^1, u^2)$, and next observation z^2,

$$\iota_{\tau+1}(y, (o^1, u^1, z^1)) = \delta^{u^1}_{\hat{a}^1(o^1)} \sum_{x \in X} \iota_\tau(x, o^1) \cdot p^{u^1,u^2,z^1,z^2}_{x,y}.$$

Given the information state, the central planner acts on behalf of both agents by selecting an action for the second agent and a decision rule for the first one based upon their histories. However, the primary result of this paper, *cf.* Theorem 1, is the proof that one can equivalently condition the action selection upon either belief states or posterior probability distributions over belief states for agent 1 and 2, respectively. Before proceeding any further, we define two key concepts, namely belief occupancy states and belief-dependent joint decision rules.

Definition 3. *The belief occupancy state is a probability distribution*[1] *over belief states conditional on the information state,* i.e. $\forall b \in \triangle(X)$, $s_\tau \doteq P(b|\iota_\tau)$ *for any time* τ.

We call *augmented belief occupancy state* a belief occupancy state paired with agent 2's history o^2_τ (denoted $h(s_\tau)$). The latter will prove useful to extract the policy of agent 2 given the optimal value function.

Definition 4. *The belief-dependent joint decision rule* $a_\tau \doteq (a^1, u^2)$ *at any time* τ *maps belief states to probability distributions over joint actions,* i.e. $\forall b \in \triangle(X)$, $a_\tau(u|b) \doteq (a^1(u^1|b), u^2)$ *and* $a^1 \colon \triangle(X) \mapsto \triangle(U^1)$.

The primary result of this paper is the proof that, under one-sidedness, belief-dependent joint decision rules are as good as history-dependent ones.

[1]From Assumption 3, we know the number of joint histories that belief states summarize is finite, so is the number of reachable belief states. As a consequence, one can maintain distribution over reachable belief states.

Theorem 1 (Proof in App. A.1)**.** *In Dec-POMDPs with one-sidedness, optimal policies depend only upon belief states for agent* 1 *and belief occupancy states for agent* 2*.*

This theorem shows that agent 1 with access to joint histories can act optimally based on corresponding belief states. Instead, agent 2 has to reason on its belief about agent 1's belief states, *i.e.* the belief occupancy states. It is worth noticing that if agent 2 has private observations, then both agents' policies are history-dependent. However, Theorem 1 holds whether or not agent 2's actions are made public. That is because actions can be recovered from observation histories (Oliehoek, 2013). These policies can be made even more concise, assuming one-sided partial observability as in Hadfield-Menell et al. (2016), and Malik et al. (2018), *e.g.* (i) agent 2's actions and observations are public, and (ii) every joint observation reveals the actual state of the world. The collective full observability assumption turns all agent 1's belief states into states, and thus all agent 2's belief occupancy states into belief states.

3.2. Sufficiency of Belief Occupancy States

To demonstrate the sufficiency of both belief occupancy states and belief-dependent joint decision rules to optimally solve Dec-POMDPs with one-sidedness, it will prove useful to establish the following preliminary results.

Lemma 2 (Proof in App. A.2)**.** *The belief occupancy state* s *describes a process that is Markov,* i.e. *the next belief occupancy state,* $s' \doteq T(s, a, z^2)$*, depends on the current belief occupancy state* s*, belief-dependent joint decision rule* a *and observation* z^2*,*

$$s'(b') \propto \sum_{b \in \triangle(s)} s(b) \sum_{u^1} a^1(u^1|b) \sum_{z^1} \delta^{b'}_{b^{u,z}} \sum_{x,y} b(x) \cdot p^{u,z}_{x,y},$$

where $\triangle(s) \doteq \{b \in \triangle(X) \colon s(b) > 0\}$ *is a finite subset of the simplex* $\triangle(X)$ *and* $b^{u,z}(y) \propto \sum_{x \in X} b(x) \cdot p^{u,z}_{x,y}$*.*

Next, we show the belief occupancy states and belief-dependent joint decision rules are sufficient statistics for estimating immediate rewards.

Lemma 3 (Proof in App. A.2)**.** *For any arbitrary information state* ι *and belief-dependent joint decision rule* a*, the immediate reward,* i.e. $\hat{R}(\iota, a) \doteq E\{r(x, u) \mid \iota, a\}$*, depends only upon the corresponding belief occupancy state* s*:*

$$\hat{R}(\iota, a) = \sum_b s(b) \sum_u a(u|b) \sum_x b(x) \cdot r(x, u) \doteq R(s, a),$$

where $s(b) \doteq P(b|\iota)$ *for every* $b \in \triangle(s)$*.*

We also show that the belief occupancy state is a sufficient statistic for estimating the probability that an observation is made public given an information state and (belief-dependent) joint decision rule.

Lemma 4 (Proof in App. A.2)**.** *For any arbitrary information state ι, belief-dependent joint decision rule a, the probability that observation z^2 is made public,* i.e. $\hat{\Omega}(z^2|\iota, a) \doteq P(z^2 \mid \iota, a)$*, depends on ι only through the corresponding belief occupancy state s: $\hat{\Omega}(z^2|\iota, a) = \Omega(z^2|s, a)$, where*

$$\Omega(z^2|s,a) \doteq \sum_{b\in\triangle(s)} s(b) \sum_{x,u,y} a(u|b)\cdot b(x) \sum_{z^1} p^{u,z}_{x,y}.$$

Taking all lemmas together, we note that the process the belief occupancy state describes is a partially observable oMDP, namely the belief-occupancy MDP (boMDP).

Definition 5. *The boMDP is given by $\tilde{M} \doteq \langle S, A, R, P\rangle$ where: S defines the set of belief occupancy states, with initial belief occupancy state being $s_0 \doteq b_0$; A is the set of joint decision rules; $R\colon S\times A \mapsto \mathbb{R}$ defines the immediate reward after taking joint decision rule in a belief occupancy state; $P\colon S\times A\times S \mapsto [0,1]$ describes the probability $p^a_{s,s'} \doteq \sum_{z^2\in Z^2} \Omega(z^2|s,a)\delta^{s'}_{T(s,a,z^2)}$ of the next belief occupancy state s' after taking joint decision rule a in s.*

The optimal value function of $\tilde{M}$ is the solution of the Bellman optimality equation:

$$v^*(s) = \max_{a\in A} \{R(s,a) + \gamma \sum_{s'\in S} p^a_{s,s'} v^*(s')\}. \quad (2)$$

We are ready to prove another important result of the paper, which will make it possible to optimally solve M with one-sidedness by optimally solving $\tilde{M}$.

Theorem 2 (Proof in App. A.2)**.** *The belief occupancy state constitutes a sufficient statistic of the information state for optimally solving $\tilde{M}$. Besides, an optimal solution for $\tilde{M}$ is also an optimal solution for M with one-sidedness.*

Following the centralized planning for decentralized execution theory, it only remains to exhibit and exploit the properties of the optimal value function, before one can transfer algorithms from MDP theory and algorithms to Dec-POMDPs with one-sidedness.

4. Exploiting Value-Function Properties

This section presents perhaps the main results of the paper, including the proof that the optimal value function is a linear function of augmented belief occupancy states. Also, we investigate practical representations of lower and upper bounds of the optimal value function.

Lemma 5 (Proof in App. B.1)**.** *The optimal value function $(v^*_\tau)_{\tau\in\{0,\dots,\ell-1\}}$, solution of Equation (2), is a piecewise-linear and convex function of the belief occupancy states. Moreover, for any time τ, there exists a family (Λ_τ) of sets (β_τ) with $|X|$-dimensional vectors (called α-vectors) such that, for any belief occupancy state s_τ*

$$v^*_\tau(s_\tau) = \max_{\beta_\tau\in\Lambda_\tau} \sum_{b\in\triangle(s_\tau)} s_\tau(b) \max_{\alpha_\tau\in\beta_\tau} \sum_{x\in X} b(x)\alpha_\tau(x). \quad (3)$$

This property shows Lemma 1 also holds under one-sidedness but now over belief occupancy states, which are more concise than history occupancy states. Yet, both properties remain fundamentally different in their generalization capabilities. Here, the PWLC function over belief occupancy states is a family of PWLC functions over belief states. That suggests Lemma 5 has more ability to handle unseen states than Lemma 1. Yet, as we will show in Sections 4.1 and 4.2, storing and updating PWLC functions is non-trivial. Maintaining a value function over the entire belief occupancy space is cumbersome. Instead, one can alternatively only keep track of the value function induced by the current best joint policy, as in Dibangoye & Buffet (2018). That is, we only store and maintain values for belief occupancy states visited under that joint policy.

Lemma 6 (Proof in App. B.2)**.** *If we let π^* be an optimal joint policy of M with one-sidedness, then the optimal value function $(v^*_\tau)_{\tau\in\{0,\dots,\ell-1\}}$, solution of Equation (2), is a linear function of augmented belief occupancy states s_τ visited under $\pi^* \doteq (\pi^1, \pi^2)$,* i.e.

$$v^*_\tau(s_\tau) = \sum_{b\in\triangle(s)} s(b)\cdot V^1_\tau(h(s_\tau), b) = V^2_\tau(h(s_\tau)) \quad (4)$$

where $V^1_\tau\colon O^2_\tau\times\triangle(X)\mapsto\mathbb{R}$ and $V^2_\tau\colon O^2_\tau\mapsto\mathbb{R}$ denote τth value functions under policies π^1 and π^2.

A careful reader may wonder why do we need two value functions instead of a single one, as $V^1 \doteq (V^1_\tau)_{\tau\in\{0,\dots,\ell-1\}}$ seems to subsume $V^2 \doteq (V^2_\tau)_{\tau\in\{0,\dots,\ell-1\}}$. The knowledge of V^2 is critical when updating the value function. Intuitively, V^2 keeps track of the current best values under the current best policy of agent 2, which is not explicit in value function V^1. Thus, before agent 2 commits to another policy, one can check whether or not the latter improves the previous one, hence ensuring updates never commit to worst policies. Overall, Lemma 6 trades generalization capabilities of Lemma 5 for fast point-based backups.

Theorem 3 (Proof in App. B.2)**.** *The point-based backup of the value functions (V^1, V^2) at point s satisfies*

$$V^1_\tau(h(s),b) = \begin{cases} V^1_\tau(h(s),b), \text{ if } V^2_\tau(h(s)) \geq W^2_\tau(h(s)) \\ Q^1_\tau(h(s),b,a^s(b)), \text{ otherwise}, \end{cases}$$

$$V^2_\tau(h(s)) = \max\{V^2_\tau(h(s)), W^2_\tau(h(s))\}$$

where $\rho(b,u) \doteq \langle b, r(\cdot,u)\rangle$, and a^s, W^2_τ, Q^1_τ are given by:

$$W^2_\tau(h(s)) \doteq \sum_{b\in\triangle(s)} s(b) Q^1_\tau(h(s), b, a^s(b))$$

$$a^s \doteq \arg\max_{(u^2,a^1)} \sum_{b\in\triangle(s)} s(b)\cdot Q^1_\tau(h(s), b, a^1(b), u^2)$$

$$Q^1_\tau(h(s),b,u) \doteq \rho(b,u) + \gamma \mathbb{E}^b_u\{V^1_{\tau+1}((h(s),u^2,z^2), b^{u,z})\}$$

Intuitively, Theorem 3 establishes a policy improvement strategy of the joint policy embodied into the current linear value function. It makes possible to perform point-based backups in polynomial time, *i.e.* $\mathcal{O}(|\triangle(s)||U||Z|)$, *cf.*

Lemma 7 in App B.2, mainly because it performs local improvements of a single joint policy. This result generalizes to M with one-sidedness a recent efficient backup operator for M with one-sided partially observability, *cf.* Malik et al. (2018). Next, we provide different representations of the optimal value function using either piece-wise linearity and convexity or linearity properties, along with update rules.

4.1. Lower-Bound Representations

This section presents lower-bound representations of the optimal value function using either PWLC or linear properties from Lemmas 5 and 6, respectively.

The family of sets representation is not commonly used to update and store lower bounds on the optimal value function. In such a representation, the value at time τ of belief occupancy state s is the maximum projection of s onto a family Λ_τ of sets (β_τ) of α-vectors, *cf.* Figure 1 in App. B.1: $\underline{\upsilon}_\tau(s) \doteq \max_{\beta_\tau \in \Lambda_\tau} \sum_{b \in \triangle(s)} s(b) \max_{\alpha \in \beta_\tau} \langle b, \alpha \rangle$. Theorem 5 shows a family of sets can represent the optimal value function $(\upsilon^*_\tau)_{\tau \in \{0,\dots,\ell-1\}}$ exactly. Initially, family of sets Λ_τ contains a singleton β_τ, which includes α-vector α_τ given by: for all state $x \in X$, $\alpha_\tau(x) \doteq (\ell - \tau) \min_{u \in U} r(x,u)$. Updating $\underline{\upsilon}_\tau$ consists in adding a novel set of α-vectors into Λ_τ.

Corollary 1 (Proof in App. B.3)**.** *The backup operator of the lower bound $\underline{\upsilon}_\tau$, represented as a family of sets, generates a new set β^*_s of α-vectors for a given belief occupancy state s,* i.e. $\texttt{backup}(\underline{\upsilon}_\tau, s) = \Lambda_\tau \cup \{\beta^*_s\}$*, where*

$$\beta^*_s = \operatorname*{arg\,max}_{\beta^a_s \colon a \in A} \sum_{b \in \triangle(s)} s(b) \max_{\alpha \in \beta^a_s} \langle b, \alpha \rangle$$

$$\beta^a_s \doteq \{\alpha^a_b \mid \forall b \in \triangle(s)\}$$

$$\alpha^a_b \doteq \sum_{z^2 \in Z^2} \alpha^{a(b),z^2}_{b,\beta^a_{z^2}}$$

$$\beta^a_{z^2} \doteq \operatorname*{arg\,max}_{\beta \in \Lambda_{\tau+1}} \sum_{b \in \triangle(s)} s(b) \langle b, \alpha^{a(b),z^2}_{b,\beta} \rangle$$

$$\alpha^{u,z^2}_{b,\beta} \doteq \frac{R(u)}{|Z^2|} + \gamma \sum_{z^1 \in Z^1} \operatorname*{arg\,max}_{\alpha^{u,z} \colon \alpha \in \beta} \langle b, \alpha^{u,z} \rangle$$

where $\alpha^{u,z}(\cdot) \doteq \sum_{y \in X} \alpha(y) p^{u,z}_{\cdot,y}$, and $R(u) \doteq r(\cdot, u)$, for any $u \in U$ and $z \in Z$.

Corollary 1 describes an approach to update the family of sets by computing possibly a novel set of α-vectors for each belief occupancy state. However, this operation can be extremely expensive. Indeed the full complexity of a point-based backup of a family of sets is about $\mathcal{O}(|A||\triangle(s)||X|^2|Z^2| + |U||X|^2|Z||\triangle(s)||\beta^*||\Lambda|)$, where $|\beta^*| \geq \max_{\beta \in \Lambda} |\beta|$, *cf.* Lemma 8 in App B.3. One can slightly mitigate this drawback using mixed-integer linear programming, *cf.* Corollary 2 in App. B.3. But the scalability remains a major issue as the number of sets increases.

Finally, it is necessary to maintain a concise representation of the lower-bound value function through pruning. Here, we perform pruning incrementally. We first prune each set β of α-vectors to preserve only α-vectors in β that are non-dominated by other α-vectors in β using, for example, point-wise dominance criterion (Smith, 2007). Next, we prune sets $\beta \in \Lambda$ that are dominated by another set β' in Λ, *e.g.* all α-vectors in set β are point-wise dominated by some α-vector in set β'. More efficient pruning procedures exist, but their application often comes with prohibitive costs.

As for the lower-bound representation using the linearity property, one relies on tabular representations. Initially, $V^1_\tau(\cdot) = V^2_\tau(\cdot) = (\ell - \tau) \min_{u \in U} r(x,u)$. We update these value functions as discussed in Theorem 3. Notice that, for every updated augmented belief-occupancy state s, we need to update value function $V^1_\tau(h(s), \cdot)$ over the entire belief space. In the tabular representation, we backup all belief states in $\triangle(s)$ and assign lower bound $(\ell - \tau) \min_{u \in U} r(x,u)$ to any other belief state, yet a set of α-vectors could advantageously replace the tabular representation. To mitigate the number of histories stored in value functions V^1 and V^2, we retain only a portion of the entire history. Dibangoye et al. (2014b) show policies dependent on finite-memory can achieve performances as good as that of full-length histories. This approach may lead to approximate solutions.

4.2. Upper-Bound Value Functions

Here, we present upper-bound representations of the optimal value function using either PWLC or linear properties from Lemmas 5 and 6, respectively.

The state-value mappings, *i.e.* $\Psi \doteq \{(s^{(\kappa)} \mapsto v^{(\kappa)})\}$, are commonly used to represent upper-bound value functions $\bar{\upsilon}_\tau$ such that for any arbitrary belief occupancy state s at time τ, $\upsilon^*_\tau(s) \leq \bar{\upsilon}_\tau(s)$ where $\forall s, b, \kappa$

$$\bar{\upsilon}_\tau(s) \doteq \min\{\upsilon_{\texttt{MDP}}(s), \upsilon_{\texttt{SAWTOOTH}}(s,\kappa) \mid \kappa \in \Psi\}$$

$$\upsilon_{\texttt{SAWTOOTH}}(s,\kappa) \doteq \upsilon_{\texttt{MDP}}(s) + \max_{b \in \triangle(s^\kappa)} s(b) / \texttt{credit}(\kappa, b)$$

$$\texttt{credit}(\kappa, b) \doteq s^\kappa(b) / (v^\kappa - \upsilon_{\texttt{MDP}}(s^\kappa))$$

where $\upsilon_{\texttt{MDP}} \colon S \mapsto \mathbb{R}$ defines the value induced by the optimal policy of the underlying MDP; and $\upsilon_{\texttt{SAWTOOTH}} \colon S \times \Psi \mapsto \mathbb{R}$ describes the sawtooth approximation of the convex hull of points in Ψ (Smith, 2007), which is preferred to the convex-hull to make it possible to write the update of the upper-bound value function as a mixed-integer linear program. Updating the upper-bound value function consists in adding one point in the point set The update requires the exhaustive enumeration of all joint decision rules, which is extremely expensive for large domains. Similarly to the lower bound, one can slightly mitigate this drawback using mixed-integer linear programming, *cf.* Corollary 3 in

App. B.3. But the scalability remains a major issue as the number of sets increases. Finally, it is necessary to maintain a concise representation of the upper-bound value function through pruning. We prune points that are dominated if their values are higher or equal to that obtained using the other points (Smith, 2007).

Let us now turn our attention to the upper-bound representation using the linearity property. We proceed as in lower-bound value functions. Initially, $V_\tau^1(\cdot) = V_\tau^2(\cdot) = \upsilon_{\texttt{MDP}}(\cdot)$. Then, we backup these value functions as discussed in Theorem 3. For every encountered point s, we update all belief states in $\triangle(s)$ and assign upper bound $\upsilon_{\texttt{MDP}}(b)$ to any other belief state b. One can alternatively use the sawtooth approximation to generalize from one belief to another one.

4.3. Optimally Solving boMDPs

In principle any algorithm that applies in information-state MDPs should also apply in $\tilde{M}$ (resp. M with one-sidedness). Here, we adapt a state-of-the-art solver for information-state MDPs, namely heuristic search value iteration (HSVI) (Smith, 2007), *cf.* Algorithm 1. We choose HSVI because it is guaranteed to find an optimal solution in finite time, *cf.* (Smith, 2007; Dibangoye et al., 2014b), in contrast to other alternatives like PBVI (Shani et al., 2013) or POMCP (Silver & Veness, 2010).

Algorithm 1: The HSVI Algorithm for boMDP $\tilde{M}$.

function `HSVI` (s_0, ϵ) **begin**
  Initialize $\bar{\upsilon}$ and $\underline{\upsilon}$.
  while $\texttt{gap}(s_0) > \epsilon$ **do**
    `Explore` $(s_0, 0)$.

function `Explore` (s, τ) **begin**
  if $\texttt{gap}(s) > \epsilon\gamma^{-\tau}$ **then**
    $a^s \in \arg\max_a \{R(s,a) + \gamma \sum_{s' \in S} p^a_{s,s'} \bar{\upsilon}(s')\}$.
    $s^* \in \arg\max_{s'} p^{a^s}_{s,s'}(\bar{\upsilon}(s') - \underline{\upsilon}(s') - \epsilon/\gamma^{\tau+1})$.
    `Explore` $(s^*, \tau + 1)$.
    Update $\bar{\upsilon}$ and $\underline{\upsilon}$ at s.

All variants of HSVI proceed as follows. They generate trajectories of states greedily guided by upper bounds and iteratively update lower $\underline{\upsilon}$ and upper bounds $\bar{\upsilon}$ over states. Each trajectory starts at the initial state, and continues until either the planning horizon has been reached or the gap between bounds, *i.e.* $\texttt{gap}(s_\tau) \doteq \bar{\upsilon}_\tau(s_\tau) - \underline{\upsilon}_\tau(s_\tau)$, is zero. Once a trajectory terminates, it updates upper and lower bounds over states along the trajectory in the reversed order of visit. The algorithm stops whenever the gap between bounds at the initial belief occupancy state is zero. In all our variants, states are (augmented) belief occupancy states and actions are belief-dependent joint decision rules. They remain, however, fundamentally different. They differ in the way they represent and update lower and upper bounds.

We shall distinguish between three variants, *i.e.* $\text{HSVI}_1(m)$, HSVI_2 and HSVI_3. Algorithm $\text{HSVI}_1(m)$ uses linear representations and corresponding update rules. Its convergence may, however, be significantly affected by the exponential growth of histories of agent 2 with time. To study this issue, we will consider m-length histories. Algorithms HSVI_2 and HSVI_3 use PWLC representations, but employ exhaustive enumeration and MILP to greedily select actions, respectively.

5. Experiments

5.1. Setup

Algorithms. We ran our variants of HSVI algorithm on an Ubuntu machine with 3.0GHz Xeon E5 CPU and 32GB available RAM. We solved the MILPs using ILOG CPLEX Optimization Studio. While algorithms for general Dec-POMDPs exist, there is no reason they can compete against our variants since they are not geared to exploit the one-sidedness. Indeed, the state-of-the-art solver for general Dec-POMDPs, namely FB-HSVI (Dibangoye et al., 2014b), shares the same algorithmic schemes with our variants, yet our variants use more concise notions of states and thus more efficient update operators. For the sake of completeness, we report performances for solving underlying MDP and Dec-POMDP problems using value-iteration and FB-HSVI algorithms, respectively.

Tested domains. We evaluate our algorithms on multiple 2-agent benchmarks from the literature of Dec-POMDPs recast into Dec-POMDPs with one-sidedness. All used domains are available at masplan.org including `mabc`, `Recycling`, `Grid3x3corners`, `boxPushing`, `Mars`, and `tiger`. These are the largest and most challenging benchmarks from the Dec-POMDP literature, *cf.* Table 1 for their dimensions, *i.e.* number of states $|X|$, joint actions $|U|$, observations $|Z|$, and histories $|O| \doteq \sum_{\tau=0}^{\ell-1} |O_\tau|$, perhaps the most important feature to assess the complexity of a benchmark. For each of them we compare our variants of HSVI for planning horizon $\ell = 10$ and discount factor $\gamma = 1$ and report different statistics, *i.e.* time, memory, number of trials, value, and gap. We set the time limit at 5 hours.

5.2. Results and analysis

Results. In all tested benchmarks, HSVI_1 outperforms both HSVI_2 and HSVI_3, providing near-optimal (if not optimal) values at the initial state. For small domains, *e.g.* `mabc`, where all algorithms find an optimal solution, HSVI_1 is 5 and 200 times faster than HSVI_2 and HSVI_3, respectively. For large domains, *e.g.* `Mars`, only HSVI_1 can find an optimal solution, and it takes about 2 and 3 orders of mag-

nitude less time than both HSVI_2 and HSVI_3, respectively. Another important observation is that HSVI_1 performs extremely cheap trials compared to HSVI_2 and HSVI_3—*e.g.* for `tiger`, $\text{HSVI}_1(9)$ performs about 652 trials per second whereas HSVI_2 and HSVI_3 perform about 0.55 and 0.005 trials per second, respectively. However, HSVI_2 and HSVI_3 exhibit faster rates of convergence than HSVI_1, *e.g.* for `Recycling`, all variants find an optimal solution, yet with $\#it$ being 83, 50, and 50 for $\text{HSVI}_1(1)$, HSVI_2 and HSVI_3, respectively. In all tested benchmarks, the only exception being `Grid3x3corners`, HSVI_2 outperforms HSVI_3. For the domain with the largest $|O|$, *i.e.* `Grid3x3corners`, HSVI_2 runs out of memory and was killed by the system. We also studied affect of hyper-parameter m on the performances of $\text{HSVI}_1(m)$. Of course, the smaller m the more efficient is $\text{HSVI}_1(m)$. Non surprisingly, $\text{HSVI}_1(1)$ finds an optimal solution for all tested domains, the only exception being `tiger`, which required $\text{HSVI}_1(7)$. However, using small m may lead to erratic performances. For example on `tiger`, $\text{HSVI}_1(m)$ has lower bounds slightly higher than upper bounds for $m \in \{1, 2, 6\}$.

Analysis. Clearly, the experimental results support our theoretical findings. First, they show HSVI_1 outperforms both HSVI_2 and HSVI_3 mainly because HSVI_1 makes use of a polynomial-time backup operator while its competitors employ exponential ones. Next, HSVI_2 and HSVI_3 demonstrate faster rates of convergence w.r.t. HSVI_1 as they take full advantage of the generalization capabilities of the PWLC property of the optimal value function. Unfortunately, this advantage does not manifest in the overall running time because maintaining—*i.e.* updating and pruning—a PWLC representation is prohibitively expensive. Using the linear representation, HSVI_1 trades generalization capabilities for efficient backups. Also, HSVI_2 outperforms HSVI_3 in small- and medium-sized domains, as the enumeration procedures are cheaper than solving MILPs. Whenever we face large domains, the enumeration is no longer feasible, but we can still solve MILPs, up to a certain point, of course. Finally, we notice erratic behaviors for small hyper-parameters m than the ones necessary to preserve optimality. We are pursuing our investigations to understand better how to choose good hyper-parameters in $\text{HSVI}_1(m)$ for the problem at hand. To complete the analysis, we also provided experiments for domains with and without information sharing, *i.e.* MDPs and Dec-POMDPs. Non surprisingly, results show MDPs are much easier than Dec-POMDPs with one-sidedness, which in turn are much easier than Dec-POMDPs.

Algorithm	t	$(\#it)$	$\bar{v}_0(s_0)$	$\underline{v}_0(s_0)$	$gap(s_0)$	$size(\bar{v})$	$size(\underline{v})$
`mabc` $\lvert X\rvert = 4,\ \lvert U\rvert = 16,\ \lvert Z\rvert = 16,\ \lvert O\rvert \approx 2 \cdot 10^{173}$							
$\text{HSVI}_1(1)$	0.004	(10)	9.2901	9.2901	0.0	20	20
$\text{HSVI}_1(2)$	0.008	(18)	9.2901	9.2901	0.0	37	37
HSVI_2	0.02	(1)	9.2901	9.29	0.0001	10	11
HSVI_3	0.8	(1)	9.2901	9.29	0.0001	10	11
MDP	0.0	–	9.7856	9.7856	0.0	1	1
Dec-POMDP	0.78	–	9.30	9.29	0.1	–	–
`tiger` $\lvert X\rvert = 2,\ \lvert U\rvert = 9,\ \lvert Z\rvert = 4,\ \lvert O\rvert \approx 2.3 \cdot 10^{34}$							
$\text{HSVI}_1(1)$	0.04627	(34)	35.0464	35.185	−0.139	117	117
$\text{HSVI}_1(2)$	0.08167	(55)	36.7758	36.8574	−0.082	269	269
$\text{HSVI}_1(3)$	0.18503	(123)	38.1152	36.4025	1.7127	622	622
$\text{HSVI}_1(4)$	0.32220	(217)	38.1582	35.6555	2.5027	1159	1159
$\text{HSVI}_1(5)$	0.85365	(570)	37.6476	35.8861	1.7615	2684	2684
$\text{HSVI}_1(6)$	1.53857	(987)	37.4912	37.5	−0.0088	4909	4909
$\text{HSVI}_1(7)$	2.3	(1477)	37.5	37.5	0.0	8061	8061
$\text{HSVI}_1(8)$	3.12085	(2022)	37.5	37.5	0.0	11171	11171
$\text{HSVI}_1(9)$	3.57568	(2333)	37.5	37.5	0.0	12283	12283
HSVI_2	491.5	(274)	37.5	37.5	0.0	1090	653
HSVI_3	17916	(88)	39.42	37.5	1.9	432	354
MDP	0.0	–	200.0	200.0	0.0	1	1
Dec-POMDP	65.57	–	15.194	15.184	0.01	–	–
`Grid3x3corners` $\lvert X\rvert = 81,\ \lvert U\rvert = 25,\ \lvert Z\rvert = 81,\ \lvert O\rvert \approx 1.3 \cdot 10^{1019}$							
$\text{HSVI}_1(1)$	16.8	(71)	4.779	4.751	0.028	1769	1769
$\text{HSVI}_1(2)$	50.5533	(218)	4.78459	4.75261	0.03198	5593	5593
HSVI_2	*— out of memory —*						
HSVI_3	20806	(11)	4.831	4.588	0.243	75	99
MDP	0.02	–	4.8819	4.8819	0.0	1	1
Dec-POMDP	34.42	–	4.69	4.68	0.01	–	–
`Recycling` $\lvert X\rvert = 4,\ \lvert U\rvert = 9,\ \lvert Z\rvert = 4,\ \lvert O\rvert \approx 10^{14}$							
$\text{HSVI}_1(1)$	0.016	(14)	32.1953	32.0743	0.13	83	83
$\text{HSVI}_1(2)$	0.056	(36)	32.2636	31.9072	0.3564	273	273
HSVI_2	0.81	(49)	32.1893	32.1893	0.0	150	50
HSVI_3	212	(20)	32.1893	32.1893	0.0	89	50
MDP	0.0	–	33.8208	33.8208	0.0	1	1
Dec-POMDP	0.52	–	31.873	31.863	0.01	–	–
`boxPushing` $\lvert X\rvert = 100,\ \lvert U\rvert = 16,\ \lvert Z\rvert = 25,\ \lvert O\rvert \approx 10^{271}$							
$\text{HSVI}_1(1)$	1.06659	(48)	228.035	228.035	0.0	367	367
$\text{HSVI}_1(2)$	2.00303	(102)	228.032	228.031	0.001	677	677
$\text{HSVI}_1(3)$	4.67623	(226)	228.031	228.031	0.0	1269	1269
HSVI_2	4883.38	(29)	228.303	228.067	0.236	165	248
HSVI_3	18195.7	(7)	228.696	210.155	18.54	52	67
MDP	0.019	–	244.849	244.849	0.0	1	1
Dec-POMDP	293.7	–	223.75	223.74	0.01	–	–
`Mars` $\lvert X\rvert = 256,\ \lvert U\rvert = 36,\ \lvert Z\rvert = 64,\ \lvert O\rvert \approx 2.6 \cdot 10^{896}$							
$\text{HSVI}_1(1)$	34.0906	(64)	26.5991	26.5983	0.0008	557	557
$\text{HSVI}_1(2)$	27.8468	(62)	26.5482	26.5482	0.0	543	543
HSVI_2	13196.4	(34)	27.4218	26.5723	0.85	135	204
HSVI_3	19045.7	(30)	27.4295	26.5545	0.875	121	187
MDP	0.398	–	28.6133	28.6133	0.0	1	1
Dec-POMDP	62.7	–	26.32	26.31	0.01	–	–

Table 1. Running time t (in seconds), number of iterations $\#it$, upper and lower bound values at the initial state, the gap between bounds, and sizes of upper and lower bounds.

6. Discussion

This paper provides the first theory and algorithms to solve two-agent Dec-POMDPs with one-sided information sharing optimally. Findings include proofs that (i) a sufficient statistic for optimal decision making is the belief occupancy state, (ii) the optimal value function is PWLC in the belief-occupancy space and linear in a higher-dimensional space, and (iii) a polynomial-time point-based backup operator exists. We also present three HSVI variants utilizing these properties, hence providing the first experimental results on this family of problems. This paper specializes a previous work on Dec-POMDPs (Dibangoye et al., 2014b; Dibangoye & Buffet, 2018) by providing more concise sufficient statistics and efficient backup operators. Besides, it generalizes a recent work on two-agent Dec-POMDPs with one-sided partial observability (Hadfield-Menell et al., 2016; Malik et al., 2018)—here, however, we assume both agents have partial observations about the state of the world. We hope this work will serve as a theoretical building block for the growing field of cooperative multi-agent reinforcement learning (Foerster et al., 2018; Rashid et al., 2018; Dibangoye & Buffet,

2018; Bard et al., 2020).

At the core of most methods in this field is the planner-centric viewpoint, which allows a central planner to turn the original problem into an information-state MDP, where the information state is a partial joint policy (Szer et al., 2005). Another critical step was the proof that this reformulation allows (i) planning based on sufficient statistics of the partial joint policies, *i.e.* history-occupancy states, and (ii) exploiting the uniform-continuity property of the optimal value function (Dibangoye et al., 2014b). This paper shows that improved scalability can be achieved by (i) using the linearity property of the optimal value function as in (Dibangoye & Buffet, 2018), (ii) performing local improvements of this value function as in (Malik et al., 2018), and (iii) properly selecting the right history length of interest for the given problem at hand. We are currently investigating the question of how to take full advantage of the PWLC optimal value function while still preserving efficient backups. We shall extend our theory to hierarchical information-sharing structures—*i.e.* agent n knows all agent (n-1) knows, agent (n-1) knows all agent (n-2) knows, and so on—by reducing them into acyclic structures similarly to topological ordering approaches (Dibangoye et al., 2008; 2009). Several other questions are still open, including the ability to apply these insights to broader settings, *e.g.* in non-cooperative settings (Horák et al., 2017; Horák & Bošanskỳ, 2019), or under incomplete knowledge about the model.

Software and Data

The software and data we used to generate the experiments are available at https://gitlab.inria.fr/jdibango/osis-dec-pomdps.

Acknowledgements

This work was supported by ANR project PLASMA (http://perso.citi-lab.fr/jdibangoy/#/plasma) *i.e.* "Planning and Learning to Act in Systems of Multiple Agents", under Grant 19-CE23-0018-01.

References

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

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

Bellman, R. E. *Dynamic Programming*. Dover Publications, Incorporated, 1957.

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

Dibangoye, J. S. and Buffet, O. Learning to act in decentralized partially observable mdps. In *ICML*, 2018.

Dibangoye, J. S., Chaib-draa, B., and Mouaddib, A.-i. A novel prioritization technique for solving Markov decision processes. In *FLAIRS Conference*, pp. 537–542, 2008.

Dibangoye, J. S., Shani, G., Chaib-Draa, B., and Mouaddib, A.-I. Topological order planner for POMDPs. In *IJCAI*, 2009.

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

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

Dibangoye, J. S., Buffet, O., and Charpillet, F. Error-Bounded Approximations for Infinite-Horizon Discounted Decentralized POMDPs. In *ECML*, pp. 338–353, 2014c.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *AAAI*, 2018.

Ghosh, M., McDonald, D., and Sinha, S. Zero-sum stochastic games with partial information. *JOTA*, 121, 2004.

Goldman, C. V. and Zilberstein, S. Decentralized Control of Cooperative Systems: Categorization and Complexity Analysis. *JAIR*, 22, 2004.

Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. Cooperative inverse reinforcement learning. In *NIPS*, 2016.

Hansen, E. A., Bernstein, D. S., and Zilberstein, S. Dynamic Programming for Partially Observable Stochastic Games. In *AAAI*, 2004.

Hauskrecht, M. Value-Function Approximations for Partially Observable Markov Decision Processes. *JAIR*, 13: 33–94, 2000.

Horák, K. and Bošanskỳ, B. Solving partially observable stochastic games with public observations. In *AAAI*, 2019.

Horák, K., Bošanskỳ, B., and Pěchouček, M. Heuristic search value iteration for one-sided partially observable stochastic games. In *AAAI*, 2017.

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

MacDermed, L. C. and Isbell, C. Point Based Value Iteration with Optimal Belief Compression for Dec-POMDPs. In *NIPS*, 2013.

Malik, D., Palaniappan, M., Fisac, J., Hadfield-Menell, D., Russell, S., and Dragan, A. An efficient, generalized Bellman update for cooperative inverse reinforcement learning. In *ICML*, 2018.

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

Nayyar, A., Mahajan, A., and Teneketzis, D. Optimal Control Strategies in Delayed Sharing Information Structures. *TAC*, 56, 2011.

Nayyar, A., Mahajan, A., and Teneketzis, D. Decentralized stochastic control with partial history sharing: A common information approach. *TAC*, 58, 2013.

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

Oliehoek, F. A. and Spaan, M. T. J. Tree-Based Solution Methods for Multiagent POMDPs with Delayed Communication. In *AAAI*, 2012.

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

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*, 2018.

Seuken, S. and Zilberstein, S. Formal models and algorithms for decentralized decision making under uncertainty. *JAAMAS*, 17, 2008.

Shani, G., Pineau, J., and Kaplow, R. A survey of point-based POMDP solvers. *JAAMAS*, 27, 2013.

Silver, D. and Veness, J. Monte-Carlo Planning in Large POMDPs. In *NIPS*, 2010.

Smith, T. *Probabilistic Planning for Robotic Exploration*. PhD thesis, The Robotics Institute, Carnegie Mellon University, 2007.

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