Exploiting Sparse Dependencies for Communication Reduction in Multiagent Planning under Uncertainty

João V. Messias *
Institute for Systems and Robotics
Instituto Superior Técnico
jmessias@isr.ist.utl.pt

Matthijs T. J. Spaan
Institute for Systems and Robotics
Instituto Superior Técnico
mtjspaan@isr.ist.utl.pt

Pedro U. Lima
Institute for Systems and Robotics
Instituto Superior Técnico
pal@isr.ist.utl.pt

Abstract

Factored Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) form a powerful framework for multiagent planning under uncertainty, but optimal solutions require a rigid history-based policy representation. In this paper we allow inter-agent communication which turns the problem in a centralized Multiagent POMDP (MPOMDP). We map subsets of state factors to an agent's local actions through a projection of the factored joint MPOMDP policy. The key point is that when sparse dependencies between the agents' decisions exist, often the belief over its local state factors is sufficient for an agent to identify the optimal action, and communication can be avoided. We formalize these notions using the linear supports of the MPOMDP value function, and present experimental results illustrating the savings in communication that we can obtain.

1 Introduction

Intelligent decision making in real-world scenarios requires an agent to take into account its limitations in sensing and actuation. These limitations lead to uncertainty about the state of environment, as well as how the environment will respond to performing a certain action. When multiple agents interact and cooperate in the same environment, the optimal decision-making problem is particularly challenging. For an agent in isolation, planning under uncertainty has been studied using decision-theoretic models like Partially Observable Markov Decision Processes (POMDPs) [7]. Our focus is on multiagent techniques, building on the factored Multiagent POMDP model. In this paper, we propose a novel method that exploits sparse dependencies in such a model in order to reduce the amount of inter-agent communication.

The major source of intractability for optimal Dec-POMDP solvers is that they typically reason over all possible histories of observations other agents can receive, and all possible actions they might take. This allows for tightly-coupled optimal solutions, but is not very scalable. In this work, we consider factored Dec-POMDPs in which communication between agents is possible, which has already been explored for non-factored models [13, 14, 16] as well as factored Dec-MDPs [15]. When agents share their observations at each time step, the decentralized problem reduces to a centralized one, known as a Multiagent POMDP (MPOMDP) [13]. Such centralized solutions are of lower computational complexity, and they provide an upper bound on the performance achievable by a team of communicating agents [5].

Instead of the rigid history-based plans currently in use, in this work we take steps in developing a more flexible policy representation, departing from a MPOMDP solution. Value functions are a common way to represent plans in decision-theoretic planning, as they can be used to compute the relative benefit of taking an action in a particular situation. We map subsets of state factors to an agent's local actions through a projection of the factored MPOMDP solution. Individual policies map beliefs over these state factors to actions. While bounded approximations are possible for probabilistic inference [3], these results do not carry over directly to decision-making settings (but see [10]). Intuitively, even a small difference in belief can lead to a different action being taken.

However, when sparse dependencies between the agents' decisions exist, often the belief over its local state factors is sufficient for an agent to identify the action that it should take, and communication can be avoided. We formalize these notions using the linear supports of the MPOMDP value function, extracting those situations in which communication is superfluous. This is achieved by determining those regions in the local belief space which are covered by linear supports associated with a single action. We present experimental results showing the savings in communication that we can obtain, and the overall impact on decision quality.

The rest of the paper is organized as follows. First, Section 2 introduces a running example that illustrates our ideas, followed by the necessary background material in Section 3. Section 4 presents the formalization of projecting the MPOMDP value function to subsets of state factors. Next, Section 5 illustrates the concepts with experimental results, and Section 6 provides conclusions and discusses future work.

*This work was funded by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the PIDDAC Program funds. The work of J. Messias was supported by a PhD Student Scholarship, SFRH/BD/44661/2008, from the Portuguese FCT POCTI programme.

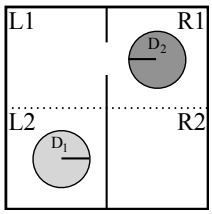


Figure 1: Layout of the *Relay* problem.

2 An Example Problem: Relay

Consider the following small-scale factored Dec-POMDP, called *Relay*, which will be used as a running example throughout the paper. In this environment, two agents operate inside a four-state world, see Figure 1, in which each agent is confined to a two-state area. One of the agents possesses a package which it must hand over to the other agent. The goal of these agents is then to "relay" this package between them through the opening between the rooms L1 and R1.

Each agent can either perform action *Shuffle*, *Exchange*, or *Sense*. A *Shuffle* action moves the agent randomly, and with equal probability, to either position in its area. The *Exchange* action attempts to perform the physical exchange of the package between the agents, and is only successful if both agents are in the correct position (L1 for the first agent, R1 for the second one) and if both agents perform this action at the same time. If it succeeds, the world is reset to a random state with uniform probability. The *Sense* action is an informative action, which allows the agent to sense whether it is in front of the opening or not, with probability of both false positives and false negatives. The feature of this small problem that we are interested in exploring is its sparse dependency between the decision processes of these agents. Evidently, the only cooperative action that the agents may perform is a joint *Exchange*. Since this action can only succeed in a particular joint state, it stands to reason that an agent which is sufficiently certain of not being in its correct, corresponding local state should always attempt to move there first (via *Shuffle*). In such a case, this decision can be taken regardless of the other agent's state, actions or observations (since the agents cannot observe each other).

The key idea in our paper is, that in some situations, the local information of these agents is enough for them to take locally optimal decisions. If, furthermore, the belief states over the local state factors are maintained independently, then the agents might not need to communicate at all between two decisions. The explicit need to communicate would only arise in situations where one agent's optimal action is dependent upon the other agent's information. In this example, this corresponds to the case where one agent is fairly certain of being in the correct place for the exchange. It then needs to reason over the other agent's local belief to make sure that an *Exchange* action is profitable in terms of expected reward.

3 Background

In this section we provide the necessary background on factored Dec-POMDPs and Multiagent POMDPs.

3.1 The Factored Dec-POMDP Model

A *factored* Dec-POMDP is defined as [11]

- $\mathcal{D} = \{1, ..., n\}$ is the set of agents. $\mathcal{D}_i$ will be used to refer to agent i;
- $\mathcal{S} = \times_i \mathcal{X}_i, i = 1, \ldots, n_f$ is the state space for the environment, decomposable into n_f factors $\mathcal{X}_i \in \{1, ..., m_i\}$ which lie inside a finite range of integer values. $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_{n_f}\}$ is the set of all state factors;
- $\mathcal{A} = \times_i \mathcal{A}_i, i = 1, ..., n$ is the joint action space. At each decision step, every agent i takes an individual action $a_i \in \mathcal{A}_i$, resulting in the *joint* action $\mathbf{a} = \langle a_1, ..., a_n \rangle \in \mathcal{A}$. Joint actions are not implicitly known by agents;
- $\mathcal{O} = \times_i \mathcal{O}_i, i = 1, ..., n$ is the space of joint observations $\mathbf{o} = \langle o_1, ..., o_n \rangle$, where $o_i \in \mathcal{O}_i$ is the observation that each agent receives after performing an action. An agent receives only its own observation in this manner;
- $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ specifies the transition probabilities $\Pr(s' | s, \mathbf{a})$;
- $O : \mathcal{O} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ specifies the joint observation probabilities $\Pr(\mathbf{o} | s', \mathbf{a})$;
- $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ specifies the reward that the team receives for performing action $\mathbf{a} \in \mathcal{A}$ in state $s \in \mathcal{S}$;
- $b_0 \in \mathcal{B}$ is a probability distribution over $\mathcal{S}$, representing the initial knowledge about the joint state. The set $\mathcal{B}$ is the space of all possible distributions over $\mathcal{S}$. We will refer to the probability of a given state being true as $b(s)$;
- h is the planning horizon, i.e. the total number of decisions that must be taken at each time step $t = 1, \ldots, h$.

The main advantage of factored (Dec-)POMDP models over their standard formulation lies in their more efficient representation, which helps counteract the naturally higher complexity associated with larger space states. In factored POMDP models, the transition and observation functions can be compactly represented through graphical representations, such as DBNs [2], which typically greatly reducing the size of the associated data structures.

Applying this notation to the *Relay* example, we can now further define the action and observation spaces of the agents as $\mathcal{A}_1 = \mathcal{A}_2 = \{\text{Shuffle}, \text{Exchange}, \text{Sense}\}$, $\mathcal{O}_1 = \mathcal{O}_2 = \{\text{Opening}, \text{Wall}, \text{Idle}\}$, and the trivial state space factorization which will be considered, as $\mathcal{X}_1 = \{\text{L1}, \text{L2}\}$ and $\mathcal{X}_2 = \{\text{R1}, \text{R2}\}$.

Existing methods for factored Dec-POMDPs can partition the decision problem across local subsets of agents, due to the possible (instantaneous) independence between their actions and observations [11]. Planning is then simplified by maximizing expected reward accrued additively between local interacting neighborhoods of agents. A natural state-space decomposition which is often possible in multi-agent teams, is to perform an *agent-wise* state space factorization, in which a state in the environment corresponds to a unique assignment over the states of individual agents. Note that this does not preclude the existence of state factors which are common to multiple agents.

3.2 From Dec-POMDPs to Multiagent POMDPs

Different assumptions over local and joint state observability further divide Dec-POMDPs into more restrictive subcategories [5]. In this work, we will consider the general case in which each factor may be partially observable.

The possibility of exchanging information between agents also greatly influences the overall complexity of solving a Dec-POMDP. In the non-communicative case, agents have to reason over the complete history of actions and observations of each other team member [1]. However, if agents are all able to communicate information (namely their observations) at each step, then it is possible to maintain a belief distribution over the joint state, which contains all necessary information through the Markov property. In such a case, the decentralized model can be reduced to a centralized one, the so-called *Multiagent POMDP* (MPOMDP) [13]. An MPOMDP is a regular single-agent POMDP but defined over the joint models of all agents. In a Dec-POMDP, at each t an agent i knows only a_i and o_i, while in an MPOMDP, it is assumed to know $\mathbf{a}$ and $\mathbf{o}$. In the latter case, inter-agent communication is necessary to share the local observations. Solving the MPOMDP is of a lower complexity class than solving the Dec-POMDP (PSPACE-Complete vs. NEXP-Complete) [1].

3.3 Linear Supports of POMDP Value Functions

It is well-known that, for a given decision step t, the value function V^t of a POMDP is a piecewise linear, convex function, which can be represented as [7]

$$V^t(b^t) = \max_{\alpha \in \Gamma^t} \alpha^{\mathrm{T}} \cdot b^t. \tag{1}$$

Where Γ^t is a set of vectors (traditionally referred to as α-vectors). It contains all information which is necessary to represent the value function at time t. Every $\alpha \in \Gamma^t$ has a particular joint action $\mathbf{a}$ associated to it, which we will denote as $\varphi(\alpha)$. Furthermore, every α-vector which is not extraneous defines a region of belief space over which it is a strict maximum. This region is a convex polytope with the constraints:

$$\begin{aligned} (\alpha - \alpha')^{\mathrm{T}} \cdot b^t \geq 0 \quad & \forall \alpha' \neq \alpha \in \Gamma^t \\ b^t(s) \geq 0 \quad & \forall s \in \mathcal{S} \\ \sum_{s \in \mathcal{S}} b^t(s) = 1 & \end{aligned} \tag{2}$$

We will refer to these regions as the *linear supports* of V, $\mathcal{L}(\alpha)$, as illustrated in Figure 2. For more details, see [4]. Furthermore, we shall make use of the *joint policy* $\pi(b)$, directly related to the value function as:

$$\pi^t(b^t) = \varphi\left(\arg\max_{\alpha \in \Gamma^t} \alpha^{\mathrm{T}} \cdot b^t\right) \tag{3}$$

In this work, our methods will assume that a value function in the form (1) is given, for its associated fully-communicative Multiagent POMDP. However, this value function need not be optimal, nor stationary. Our techniques will attempt to preserve the quality of the supplied value function, even if it is just an approximation. The only restriction, and mostly for theoretical purposes, is that a supplied value function is in its so-called *parsimonious* representation, which means that every $\alpha \in \Gamma$ has an associated non-empty linear support $\mathcal{L}(\alpha)$.

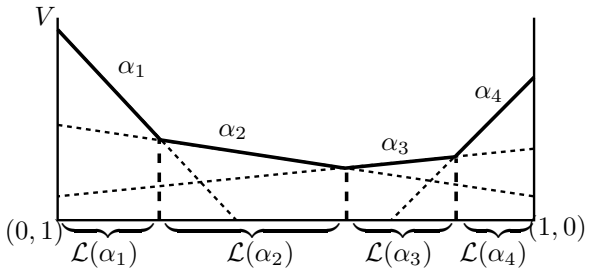


Figure 2: An example of the linear supports for a value function in a two-state POMDP.

3.4 Factored Belief States

A *joint* belief state is a probability distribution over the set of states $\mathcal{S}$, and encodes all of the information gathered by all agents in the Dec-POMDP up to a given time t:

$$\begin{aligned} b^t(s) &= \Pr(s^t | \mathbf{a}^{t-1}, \mathbf{o}^{t-1}, \mathbf{a}^{t-2}, \mathbf{o}^{t-2}, \ldots, \mathbf{a}^1, \mathbf{o}^1, b_0) \\ &= \Pr(\mathcal{X}_1^t, \ldots, \mathcal{X}_{n_f}^t | \cdot) \end{aligned} \tag{4}$$

A factored belief state is a representation of this very same joint belief as the product of n_b assumedly independent belief states over the state factors $\mathcal{X}_i$, which we will refer to as *belief factors*:

$$b^t = \times_{i=1}^{n_b} b_{\mathcal{G}_i}^t \tag{5}$$

Every factor $b_{\mathcal{G}_i}^t$ is defined over a subset $\mathcal{G}_i \subseteq \mathcal{X}$ of state factors, so that:

$$b^t(s) = \Pr(\mathcal{G}_1^t | \cdot) \Pr(\mathcal{G}_2^t | \cdot) \cdots \Pr(\mathcal{G}_{n_b}^t | \cdot) \tag{6}$$

With $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset \ , \forall i \neq j$. We will denote $\mathcal{B}_{\mathcal{G}}$ as the space of possible assignments to the belief factor defined over $\mathcal{G}$.

4 Exploiting Sparse Dependencies in Multiagent POMDPs

In the implementation of Multiagent POMDPs, an important practical issue is raised: since the joint policy arising from the value function maps joint beliefs to joint actions, all agents must maintain and update the joint belief equivalently for their decisions to remain consistent. The amount of communication required to make this possible can then become problematically large. In a direct implementation, agents would be required to communicate, at every step, their observations to all other agents.

Here, we will deal with a fully-communicative team of agents, but we will be interested in minimizing the necessary amount of communication. Even if agents can communicate with each other freely, they might not need to always do so in order to act locally, or even cooperatively. A similar idea has been used for Dec-MDPs [15], where factors can be directly observed. In that work, joint policies are broken down into individual factored policies, by reasoning over the possible

local alternative actions to a particular assignment of observable state features. The main difference to the Multiagent POMDP case lies in the presence of uncertainty over local features. This idea was approximated at runtime for Multiagent POMDPs [14], but with a reasonable loss of control quality due to the necessary heuristics, and required keeping track of a rapidly-growing number of joint belief samples.

The main assertion of this work is that, in an MPOMDP, the necessary information for efficient policy factorization is already contained within its value function, and can be obtained offline – it simply needs to be properly extracted. We will describe a method to map a belief factor (or several factors) directly to a local action.

4.1 Decision-making with factored beliefs

Note that, as fully described in [3], the factorization (6) typically results in an approximation of the true joint belief, since it is seldom possible to decouple the dynamics of a MDP into strictly independent subprocesses. An exception to this case is the so-called Network Distributed POMDP model, which assumes full independence between the action and observation models of the agents (keeping them coupled solely through the reward model) [8], which allows it to be scaled to a higher number of agents. In the general case, however, the dependencies between factors, induced by the transition and observation model of the joint process, quickly develop correlations when the horizon of the decision problem is increased, even if these dependencies are sparse. Still, it was proven in [3] that, if some of these dependencies are broken, the resulting error (measured as the KL-divergence) of the factored belief state, with respect to the true joint belief, is bounded.

Unfortunately, even a small error in the belief state can lead to different actions being selected, which may significantly affect the decision quality of the multiagent team in some settings [10, 12]. However, in rapidly-mixing processes (i.e., models with transition functions which quickly propagate uncertainty), the overall negative effect of using this approximation is minimized.

Each belief factor's dynamics can be described using a two-stage Dynamic Bayesian Network (DBN). For an agent to maintain, at each time step, a set of belief factors, it must have access to the state factors contained in a particular time slice of the respective DBNs. This can be accomplished either through direct observation, when possible, or by requesting this information from other agents. In the latter case, it may be necessary to perform additional communication episodes, besides those which are necessary solely for decision-making purposes. The amount of data to be communicated in this case, as well as its frequency, depends largely on the factorization scheme which is selected for a particular problem. We will not be here concerned with the problem of obtaining a suitable partition scheme of the joint belief onto its factors so that the accumulated error, or the implicit communication requirements, are minimized. Instead we will focus on the amount of communication which is necessary for the joint decision-making of the multi-agent team. We assume that such a partitioning is available, which is typically simple to identify for multi-agent teams which exhibit sparsity of interaction between subsets of agents.

4.2 Formal model

Here, the concepts related to the linear supports of a value function as introduced in Section 3.3 come into play. We will hereafter focus on the value function, and its associated quantities, at a given decision step t, and, for simplicity, we shall omit this dependency. However, we restate that the value function does not need to be stationary – for a finite-horizon problem, the following methods can simply be applied for every $t = 1, \dots, h$.

Recall that every α-vector has a joint action associated to it, $\varphi(\alpha)$, and a linear support over the joint belief state, $\mathcal{L}(\alpha)$. Let $\Gamma^{\mathbf{a}} = \{\alpha \in \Gamma : \varphi(\alpha) = \mathbf{a}\}$ represent the set of α-vectors which share the same joint action, $\mathbf{a}$. Then, we will define the *joint action support* of $\mathbf{a}$, $\Phi(\mathbf{a})$, as

$$\Phi(\mathbf{a}) := \bigcup_{\alpha \in \Gamma^{\mathbf{a}}} \mathcal{L}(\alpha). \tag{7}$$

Intuitively, $\Phi(\mathbf{a})$ represents the (possibly non-connected) regions of joint belief space over which $\mathbf{a}$ is the best action, as mapped by the team's joint policy, π. Note that, trivially, as a union of linear supports, action supports preserve the properties that $\cup_{\mathbf{a} \in \mathcal{A}} \Phi(\mathbf{a}) = \mathcal{B}$, and $\text{int}\left(\Phi(\mathbf{a})\right) \cap \text{int}\left(\Phi(\mathbf{a}')\right) = \emptyset$, $\forall \mathbf{a}, \mathbf{a}' \in \mathcal{A} : \mathbf{a} \neq \mathbf{a}'$, where $\text{int}(X)$ represents the topological interior of set X. Then, instead of defining π as (3), we could instead opt to represent it by mapping b through one of the existing action supports:

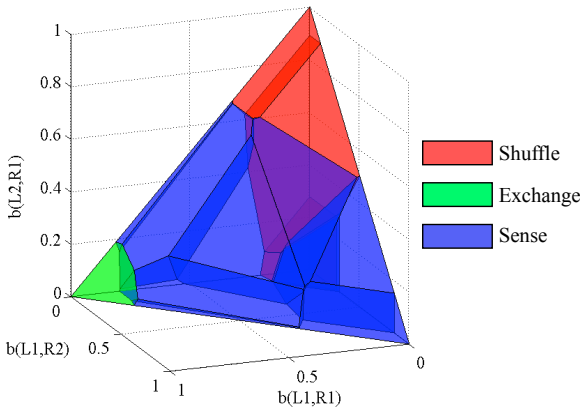
$$\pi(b) = \begin{cases} \mathbf{a}^1 & \text{if } b \in \Phi(\mathbf{a}^1) \\ \mathbf{a}^2 & \text{if } b \in \Phi(\mathbf{a}^2) \wedge b \notin \Phi(\mathbf{a}^1) \\ \cdot & \cdot \\ \cdot & \cdot \\ \mathbf{a}^{|\mathcal{A}|} & \text{if } b \in \Phi(\mathbf{a}^{|\mathcal{A}|}) \wedge b \notin \Phi(\mathbf{a}') \; \forall \mathbf{a}' \neq \mathbf{a}^{|\mathcal{A}|} \end{cases} \tag{8}$$

Note that the common boundaries of multiple linear supports are mapped to a single joint action. The same argument made here can be applied when considering only the actions of agent D_i. Let $\varphi_i(\alpha)$ denote the local action of agent D_i which corresponds to α, or, equivalently, the i-th component of $\varphi(\alpha)$. If $\Gamma_i^a = \{\alpha \in \Gamma : \varphi_i(\alpha) = a\}$, then

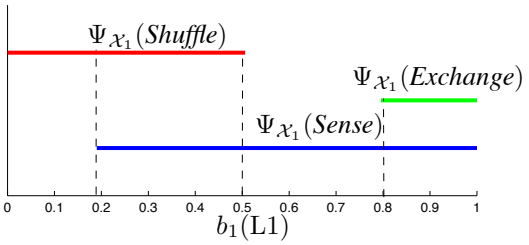
$$\Phi_i(a) = \bigcup_{\alpha \in \Gamma_i^a} \mathcal{L}(\alpha) \tag{9}$$

represents the action support of action a for agent D_i. Using these sets, we can describe $\pi_i(b)$, the local decision of agent D_i according to b (or the i-th component of $\pi(b)$) as in (8). This way, we can directly map a joint belief to a local action. A representation of these supports for agent D_1 in the *Relay* example is shown in Figure 3a. We are now interested in obtaining a representation of these supports in local belief space (i.e., over a belief factor). The marginalization of b onto $b_{\mathcal{G}}$, a belief factor defined over state factors $\mathcal{G} \subseteq \mathcal{X}$, is:.

$$\begin{aligned} b^t_{\mathcal{G}}(\mathcal{G}^t) &= \Pr\left(\mathcal{G}^t | \mathbf{a}^{1,\cdots,t-1}, \mathbf{o}^{1,\cdots,t-1}\right) \\ &= \sum_{\mathcal{X}^t \setminus \mathcal{G}^t} \Pr\left(\mathcal{X}^t_1, \mathcal{X}^t_2, \cdots, \mathcal{X}^t_{n_f} | \cdot\right) \\ &= \sum_{\mathcal{X}^t \setminus \mathcal{G}^t} b^t(s^t), \end{aligned} \tag{10}$$



(a) Action supports of agent $\mathcal{D}_1$.



(b) Local supports over belief factor $b_1(L1)$.

Figure 3: *Relay* example. (a) The $\Phi_1(\cdot)$ of agent $\mathcal{D}_1$ over the 4-dimensional joint belief space. The non-represented component is associated to a point's depth inside the tetrahedron. (b) The local supports $\Psi_{\mathcal{X}_1}$ over belief factor $b_1(L1)$. The dashed lines separate regions in which b_1 belongs to a single support.

which can be viewed as a projection of b onto the smaller subspace $\mathcal{B}_\mathcal{G}$:

$$b_\mathcal{G} = M_\mathcal{G}^\mathcal{X} b \tag{11}$$

where $M_\mathcal{G}^\mathcal{X}$ is a matrix such that $M_\mathcal{G}^\mathcal{X}(u,v) = 1$ if the assignments to the state factors in $\mathcal{G}$ are the same in $b_\mathcal{G}(u)$ and in $b(v)$, and 0 otherwise. It intuitively carries out the marginalization of points in $\mathcal{B}$ onto $\mathcal{B}_\mathcal{G}$. If we now marginalize over all points inside a given linear support $\mathcal{L}(\alpha)$, we will obtain a *local* linear support over belief factor $b_\mathcal{G}$. Let us define the projection operator of a general set X from space $\mathcal{B}$ onto subspace $\mathcal{B}_\mathcal{G}$:

$$P_\mathcal{G}^\mathcal{X}(X) := \left\{ b_\mathcal{G} \in \mathcal{B}_\mathcal{G} : b_\mathcal{G} = M_\mathcal{G}^\mathcal{X} b, \forall b \in X \subseteq \mathcal{B} \right\} \tag{12}$$

Since $\mathcal{L}(\alpha)$ is convex and we are performing the linear mapping (11), $P_\mathcal{G}^\mathcal{X}(\mathcal{L}(\alpha))$ is also convex. However, it is no longer true that $\text{int}\left\{P_\mathcal{G}^\mathcal{X}(\mathcal{L}(\alpha))\right\} \cap \text{int}\left\{P_\mathcal{G}^\mathcal{X}(\mathcal{L}(\alpha'))\right\} = \emptyset$, $\forall \alpha, \alpha' \in \Gamma : \alpha \neq \alpha'$, since different points in $\mathcal{B}$, which may lie inside different linear supports in joint space, can be mapped through (12) to the same point in $\mathcal{B}_\mathcal{G}$.

Despite this, we can still define (possibly overlapping) action supports over the space $\mathcal{B}_\mathcal{G}$, which we will refer to as *local* action supports:

$$\Psi_\mathcal{G}(a) = \bigcup_{\alpha \in \Gamma_i^a} P_\mathcal{G}^\mathcal{X}(\mathcal{L}(\alpha)) \tag{13}$$

4.3 Mapping Belief Factors to Local Actions

Returning to our *Relay* example, we can see in Figure 3b how the local action supports are spread over the belief factor b_1, which is defined over $\mathcal{G} = \{\mathcal{X}_1\}$. Here we can already see a remarkable result: some regions of $\mathcal{B}_{\mathcal{X}_1}$ are covered only by a single local action support. This implies that, if $b_1 \in \Psi_{\mathcal{X}_1}(a) \wedge b_1 \notin \Psi_{\mathcal{X}_1}(a')\ \forall a \neq a'$, there is no ambiguity as to what agent D_1 should do. Some of its expected behavior is here contained: if the agent has low probability of being in room L1, then the optimal action is to *Shuffle*; if it is not certain enough of being in L1, it should always *Sense*; and *Exchange* is always an ambiguous action, since it depends on factor $\mathcal{X}_2$. Note that, in such a case where b_1 belongs to more than one local action support simultaneously, the logical thing to do would be to request the belief b_2 from agent D_2, and then map its own action, unambiguously, from the reconstructed joint belief.

Using this insight, we can then redefine the policy of an agent D_i, in a general setting, over some belief factor $b_\mathcal{G}$, as follows:

$$\pi_i(b_\mathcal{G}) = \begin{cases} a_k \text{ if } b_\mathcal{G} \in \Psi_\mathcal{G}(a_k) \wedge b_\mathcal{G} \notin \Psi_\mathcal{G}(a') \\ \qquad\qquad \forall a' \neq a_k, k = 1, \ldots, |\mathcal{A}_i| \\ [\pi(b)]_i \qquad\qquad \text{otherwise} \end{cases} \tag{14}$$

Where $[x]_y$ represents the y-th component of x. Therefore, we can know through (14) the situations in which it is possible to map an agent's belief over a state factor directly to its own actions. While the exclusion of $b_\mathcal{G}$ from a local action support precludes the possibility of that action being the correct one, its inclusion in a support does not imply that the action is a possible choice in joint belief space. Looking back to the *Relay* example, some of the values of b_1 that map to both a_1 and a_3 through the local action supports, may in fact always map to a_1 through $\Phi_1(a_1)$ for all possible values of b_2. This is due to the assumption that it is possible to reconstruct $b = b_1 \times b_2$. In fact, the space of possible solutions to this constraint is itself a subset of $\mathcal{B}$. Some points in joint belief space can remain unreachable, although these points are also projected to the local belief space, and contribute to the formation of the local action supports.

A significant drawback in this formulation, in problems with a large number of factors, is that the full joint belief must be obtained whenever there is more than one possible action for a given belief factor. In reality, using (6) we could opt to map an agent's actions through a subset of belief factors, combining an agent's directly available belief factors with only a subset of the other agents. To this end, we will now extend our techniques to the general problem of MPOMDPs with any number of agents and state factors.

4.4 Generalization to Higher-Dimensional Problems

Let $\mathcal{H}$ be the set of all belief factors, such that $b = \times_{i \in \mathcal{H}} b_i$. Agent D_i will be associated with a subset $\mathcal{F} \subseteq \mathcal{H}$, which we

Algorithm 1 MapIntersections$\left(D_i, \mathcal{L}, \mathcal{F}, \bar{\mathcal{F}}\right)$

$j \leftarrow 0$
$\mathcal{M} \leftarrow \emptyset$
while $\bar{\mathcal{F}}$ is not empty **do**
  $b_{\bar{\mathcal{F}}} \leftarrow$ remove element from $\bar{\mathcal{F}}$
  $\Psi_{\mathcal{F}\cup\bar{\mathcal{F}}}(a) \leftarrow \bigcup_{\alpha\in\Gamma_i^a} P^{\mathcal{X}}_{\mathcal{F}\cup\bar{\mathcal{F}}}(\mathcal{L}(\alpha)) \quad \forall a \in \mathcal{A}_i$
  $\mathcal{I}_{\mathcal{F}\cup\bar{\mathcal{F}}} \leftarrow \bigcup_{a,a'\in\mathcal{A}_i} (\Psi_{\mathcal{F}\cup\bar{\mathcal{F}}}(a) \cap \Psi_{\mathcal{F}\cup\bar{\mathcal{F}}}(a'))$
  $\mathcal{I}_{\mathcal{F}} \leftarrow P^{\mathcal{F}\cup\bar{\mathcal{F}}}_{\mathcal{F}}(\mathcal{I}_{\mathcal{F}\cup\bar{\mathcal{F}}})$
  $\mathcal{M}_j \leftarrow \langle \mathcal{I}_{\mathcal{F}}, b_{\bar{\mathcal{F}}} \rangle$
  $j \leftarrow j + 1$
end while
return $\mathcal{M}$

will refer to as the agent's *local* factors. Intuitively, these are the factors that D_i must maintain and update locally at every step. Analogously, $\bar{\mathcal{F}} = \mathcal{H}/\mathcal{F}$ represents the set of *external* factors of the joint belief. We are interested in obtaining a description of the regions of $\mathcal{B}_{\mathcal{F}}$ over which agent D_i can independently map its local actions. Furthermore, and in addition to (14), in regions to which more than one action is associated, we will require an explicit enumeration of factors in $\bar{\mathcal{F}}$, which agent D_i must request so that this ambiguity is resolved. So, effectively, we require a mapping of points in $\mathcal{B}_{\mathcal{F}}$ to subsets of $\mathcal{H}$.

When marginalizing over multiple belief factors sequentially, every projection which is carried out may create new intersections between the action supports, and therefore increase the number of possible actions for a particular assignment of a $b_{\mathcal{F}}$. The following method overcomes the need to perform an exhaustive search over $\bar{\mathcal{F}}$ for every intersection, but, it is not guaranteed to return a minimal solution. The rationale is as follows: starting with a description of the linear supports $\mathcal{L}(\alpha)$ in joint belief space, we first select a single non-local factor in $\bar{\mathcal{F}}$, and marginalize each $\mathcal{L}(\alpha)$ over it. We then search for intersections amongst the action supports, in the resulting lower-dimensional belief space. If such intersections exist, we project them into $\mathcal{B}_{\mathcal{F}}$ where we know that the marginalized, non-local factor is needed (and so must be requested) in order to resolve those intersections. We proceed to select one of the remaining factors in $\bar{\mathcal{F}}$ to marginalize. The resulting map of intersections over $\mathcal{B}_{\mathcal{F}}$ then describes a set of non-local factors which the respective agent should request. If its current belief factor does not belong to any of the sets in that map, then it is guaranteed to be able to select its own action independently. This procedure is described in Algorithm 1.

4.5 Computational complexity

We discuss briefly the computational impact of our methods. As the linear supports $\mathcal{L}(\alpha)$ are convex polytopes, they admit a hyperplane-based representation (an *H*-representation), which can be used to efficiently check if a given belief state lies inside that set. Projecting convex sets onto lower-dimensional subspaces is an actively studied problem in the field of computational geometry [6]. In the average case, it exhibits polynomial complexity in the number of dimensions, and linear in the number of constraints and in the number of facets of the projected set. The focus of this paper is not on the efficient implementation of these operations, but rather on exploring the possibilities of its application on decision-making under uncertainty. Note that some long-standing POMDP solvers tackle with an equivalent problem, namely the vertex enumeration problem in the original linear support algorithm [4].

h	No Comm.	Full Comm.	Red. Comm.
6	8.56, 0%	14.55, 100%	14.14, 68.4%
10	-	38.16, 100%	37.61, 65.8%
∞	-	97.20, 100%	94.06, 72.9%

Table 1: *Relay* problem. For settings assuming no, full, and reduced communication, we show empirical control quality, communication usage.

5 Experiments

In this section, we present quantitative results regarding the application of these methods to the *Relay* setting. The joint value function was approximated, for different horizons, using the Perseus randomized point-based algorithm. The polyhedral manipulation involved in the creation and marginalization of the action supports was achieved through the use of the Multi-Parametric Toolbox for MATLAB [9].

In Table 1, the average accumulated reward is shown, for different horizons, in the fully-communicative MPOMDP case and in the reduced communication case which uses our method described in Section 4. These results show the mean of the average reward in the given horizon, over $10,000$ runs, as well as the percentage of time steps in which agents needed to communicate. The optimal value for the non-communicative (Dec-POMDP) case is also shown for $h = 6$, establishing a lower bound for performance. However, for larger horizons, no known methods are able to obtain the optimal non-communicative value due to computational complexity.

From these results, we see that there is a reduction of 35% in the number of communication events, when the actions of the agents are mapped through their respective belief factors. The effect on the decision quality of using an approximate, factored belief, is negligible (at most a 3% drop in average reward). We further note that the total savings in the amount of communication, achievable through the application of these methods, are directly related to the particular structure of the problem. On typical multiagent MDP benchmark problems, which exhibit a high level of interdependency between the agents, these methods can be applied as a way of analyzing where those dependencies lie.

In Figure 4, we show a trace of the Kullback-Leibler divergence between the approximate factored belief and a centralized belief, noting that this error is kept tightly bounded since the process is rapidly mixing ($\eta = \frac{1}{4}$, see [10]). This error only grows when the two belief factors become dependent through the application of a cooperative action. From this and due to the low loss in average value, it then follows that, for scenarios where there is sparse interaction between

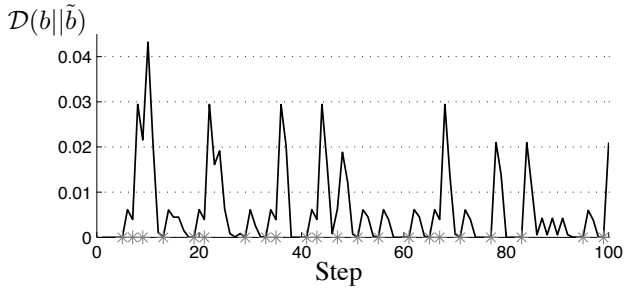


Figure 4: The Kullback-Leibler divergence between the factored belief state estimate and the "true" joint belief state, in an example run of the *Relay* problem. The marked timesteps correspond to the instances in which the agents performed a successful *Exchange*.

agents, maintaining a factored belief state is not only efficient communication-wise, but is also an acceptable approximation, in terms of its effect on the overall performance of the system.

6 Conclusions and Future Work

Traditional multiagent planning on partially observable environments mostly deals with fully-communicative or non-communicative situations. For a more realistic scenario where communication should be used only when necessary, state-of-the-art methods are only capable of approximating the optimal policy at run-time [14, 16]. Here, we have analyzed the properties of MPOMDP models which can be exploited directly from their formulation, in order to increase the efficiency of communication between agents. We have shown that these properties hold, for a simple illustrative scenario, and that the decision quality can be maintained while significantly reducing the amount of communication, as long as the dependencies within the model are sparse. We have also proposed an extension to higher-dimensional problems, which is, however, subject to an increase in computational complexity.

Future work will focus on developing new solutions to overcome the overall computational weight of these methods, which would allow their application to MPOMDP models of arbitrary size. Additionally, although one of the main features of these techniques is that they may be applied to any given MPOMDP value function in some situations this value function may be costly to obtain. We will investigate methods for obtaining MPOMDP value functions that are easy to partition using our techniques.

References

[1] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[2] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[3] Xavier Boyen and Daphne Koller. Tractable inference for complex stochastic processes. In *Proc. of Uncertainty in Artificial Intelligence*, 1998.

[4] H. Cheng. *Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, University of British Columbia, 1988.

[5] Claudia V. Goldman and Shlomo Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[6] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski. Equality set projection: A new algorithm for the projection of polytopes in halfspace representation. Technical report, Department of Engineering, University of Cambridge, March 2004. CUED/F-INFENG/TR.463.

[7] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[8] Akshat Kumar and Shlomo Zilberstein. Constraint-based dynamic programming for decentralized POMDPs with structured interactions. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*, 2009.

[9] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari. Multi-parametric toolbox (MPT). *Hybrid Systems: Computation and Control*, pages 121–124, 2004.

[10] David A. McAllester and Satinder Singh. Approximate planning for factored POMDPs using belief state simplification. In *Proc. of Uncertainty in Artificial Intelligence*, 1999.

[11] Frans A. Oliehoek, Matthijs T. J. Spaan, Shimon Whiteson, and Nikos Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*, 2008.

[12] P. Poupart and C. Boutilier. Value-directed belief state approximation for POMDPs. In *Proc. of Uncertainty in Artificial Intelligence*, volume 130, 2000.

[13] David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.

[14] M. Roth, R. Simmons, and M. Veloso. Decentralized communication strategies for coordinated multi-agent policies. In *Multi-Robot Systems: From Swarms to Intelligent Automata*, volume IV. 2005.

[15] Maayan Roth, Reid Simmons, and Manuela Veloso. Exploiting factored representations for decentralized execution in multi-agent teams. In *Proc. of Int. Conference on Autonomous Agents and Multi Agent Systems*, 2007.

[16] Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Multi-agent online planning with communication. In *Int. Conf. on Automated Planning and Scheduling*, 2009.