Exploiting Separability in Multiagent Planning with Continuous-State MDPs

Jilles S. Dibangoye[a] Christopher Amato[b] Olivier Buffet[a] François Charpillet[a]

[a] INRIA – Université de Lorraine
Villers-lès-Nancy, France
firstname.lastname@inria.fr

[b]CSAIL / MIT
Cambridge, MA, USA
camato@csail.mit.edu

ABSTRACT

Recent years have seen significant advances in techniques for optimally solving multiagent problems represented as decentralized partially observable Markov decision processes (Dec-POMDPs). A new method achieves scalability gains by converting Dec-POMDPs into continuous state MDPs. This method relies on the assumption of a centralized planning phase that generates a set of decentralized policies for the agents to execute. However, scalability remains limited when the number of agents or problem variables becomes large. In this paper, we show that, under certain separability conditions of the optimal value function, the scalability of this approach can increase considerably. This separability is present when there is locality of interaction, which — as other approaches (such as those based on the ND-POMDP subclass) have already shown — can be exploited to improve performance. Unlike most previous methods, the novel continuous-state MDP algorithm retains optimality and convergence guarantees. Results show that the extension using separability can scale to a large number of agents and domain variables while maintaining optimality.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

Keywords

Planning under uncertainty, cooperative multiagent systems, decentralized POMDPs, ND-POMDPs

1. INTRODUCTION

There is a growing interest in research for solving multiagent problems represented as decentralized partially observable Markov decision processes (Dec-POMDPs) [5]. This formalism subsumes many multiagent models including multiagent Markov decision processes (MMDPs) [7, 18, 30], transition independent decentralized MDPs [3, 12, 13, 29], networked distributed partially observable MDPs (ND-POMDPs) [19, 21, 24, 34] and more recently transition decoupled decentralized MDPs [27, 35]. Unfortunately, the NEXP-hardness of the Dec-POMDP formalism has restricted its scalability; yet, many practical applications have a structure that should allow greater scalability while preserving optimality [14, 1, 2, 4, 6, 26]. Algorithms that exploit the domain structure when it is present are particularly successful. However, even these algorithms cannot scale to realistic domains since the number of agents, states, observations and actions will often be quite large.

Separability conditions can occur when optimal value functions are the sum of linear functions over factors associated with a small subset of problem variables. These value functions are known as additive weakly-separable and linear functions (AWSL), a property that is present in the optimal value functions of many practical multi-robot coordination applications [3, 12, 13, 29], broadcast channel protocols [5, 33] and target tracking by a team of sensors [19, 21, 24, 34]. The idea of exploiting the additive weak-separability and linearity in MDP-based models is not new. It can be traced back to Koller and Parr [18], who explored the use of this property as an approximation for accelerating dynamic programming in MDPs. Since then, numerous authors have refined the approach, exploiting value function approximation schemes [7, 16, 17, 22, 28, 15]; locality of interaction, in which agents have limited interactions with one another [19, 24, 34]; or the value factorization used in approximate inference based approaches [21]. In this paper, we target domains represented as ND-POMDPs [24], which are a subclass of Dec-POMDPs that exhibit locality of interaction.

A recent method has demonstrated a scalability increase on general Dec-POMDPs by recasting them as continuous-state and deterministic MDPs [11]. This centralized method is possible by using the common assumption that planning can be centralized while preserving decentralized execution. The states of this continuous-state and deterministic MDP, called occupancy states, are distributions over Dec-POMDP states and agent histories. The associated feature-based heuristic search value iteration (FB-HSVI) algorithm preserves the ability to converge to an optimal solution and illustrates significant scalability gains on a number of Dec-POMDP benchmarks. FB-HSVI's performance relies on its ability to represent and compute the value function in a compact form, generalizing values from a small subset of occupancy states to the entire set. Unfortunately, for domains with a large number of agents, states, observations and actions, this is typically not possible even when structure such as the locality of interaction exists.

This paper combines the benefits of transforming Dec-POMDPs into continuous-state MDPs and the locality of interaction found in ND-POMDPs. The primary contribution is a demonstration that, with the locality of interaction, optimal value functions are AWSL functions of occupancy states. Even more importantly, we prove that AWSL functions depend on occupancy states only through marginal probability distributions over factors. The AWSL property permits us to introduce new value function representations that can accelerate both action selection and information tracking steps in FB-HSVI, thus enhancing performance by several orders of mag-

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

nitude while still retaining accuracy and convergence guarantees. We demonstrate the scalability of the proposed approach on many ND-POMDP benchmark domains, showing the ability to optimally solve problems that include up to fifteen agents.

2. BACKGROUND

In this section, we briefly discuss Dec-POMDPs, ND-POMDPs and the conversion of Dec-POMDPs into continuous-state MDPs.

2.1 Dec-POMDPs and ND-POMDPs

A Dec-POMDP $\mathcal{P} \equiv (S, A, Z, p, r, \eta_0, T)$ with N agents is given by: a finite set of states S; a finite set of joint actions $A = A_1 \times A_2 \times \ldots \times A_N$; an action set A_i for agent i; a finite set $Z = Z_1 \times Z_2 \times \ldots \times Z_N$ of joint observations; an observation set Z_i for agent i ; a system dynamics model $p = \{p^{a,z} \colon a \in A, z \in Z\}$; state-to-state transition matrices $p^{a,z}$, where $p^{a,z}(s, s')$ describes the probability of transiting to state s', upon receiving joint observation z and after taking joint action a in state s; a reward model $r = \{r^a \colon a \in A\}$, where r^a is a reward vector and $r^a(s)$ is the immediate reward to be gained by executing joint action a in state s; an initial probability distribution η^0 over states; and finally, a planning horizon T.

In this model, each agent receives its own observations, but it receives neither observations nor actions of the other agents. As a result, it has to reason about what the other agents observed and plan to do in order to optimize a joint stream of rewards. This property is at the core of the high complexity of Dec-POMDPs. As such, the behavioral strategies of each agent — local policies — depend only upon local information of that agent. Hence, solving Dec-POMDPs requires determining N local policies, which jointly maximize the total expected stream of rewards starting in η_0..

In the following, we write $[1\colon N] = \{1, \ldots, N\}$, and for a given subset $u \subseteq [1\colon N]$ referred to as a factor (or neighborhood), we denote $-u$ the complement of u. That is $-u = [1\colon N] \backslash u$. We also define $|u|$, the cardinality of u.

Definition 1. *An ND-POMDP is a Dec-POMDP $\mathcal{P}$ that exhibits the following properties:*

1. *A **factored state space** $S = S_0 \times S_1 \times \ldots \times S_N$, where S_0 denotes local states that agents cannot affect, and S_i represents a local-state set of agent i; We denote $s_u = (s_0, s_i)_{i \in u}$, $a_u = (a_i)_{i \in u}$, and $z_u = (z_i)_{i \in u}$, the state, action and observation relative to factor u $\subseteq [1\colon N]$.*
2. *A **multiplicative weakly-separable dynamics model** p, that is, there exists dynamics models $p_0, p_1, \ldots, p_N$ such that:*
$$p_u^{a_u, z_u}(s_u, s'_u) \quad = \quad p_0(s_0, s'_0) \prod_{i \in u} p_i^{a_i, z_i}(s_i, s_0, s'_i)$$
for any factor $u \subseteq [1\colon N]$ and $s_u = (s_0, s_i)_{i \in u}$.
3. *An **additive weakly-separable reward model** r, that is, there exists reward models $r_{u_1}, r_{u_2}, \ldots, r_{u_M}$ such that:*
$$r(s, a) \quad = \quad \sum_{k=1}^{M} r_{u_k}(s_{u_k}, a_{u_k}),$$
where $u_k \subseteq [1\colon N]$, $s_{u_k} = (s_0, s_i)_{i \in u_k}$.
4. *A **multiplicative fully-separable distribution** η^0, that is, there exists independent distributions $\eta_0^0, \eta_1^0, \cdots, \eta_N^0$ such that:*
$$\eta^0(s) \quad = \quad \eta_0^0(s_0) \prod_{i=1}^{N} \eta_i^0(s_i).$$

Unlike general Dec-POMDPs, agents in ND-POMDPs interact only with a small subset of their neighbors, demonstrating locality of interaction. For a thorough introduction, motivating examples and a graphical notation of factors, the reader can refer to [24].

2.2 Policies and Value Functions

In this section, we discuss policies in Dec-POMDPs (and ND-POMDPs) as well as the objective criteria. In the following, we distinguish between local and joint policies.

A T-step local policy of agent i, denoted π_i, is a length-T sequence of local decision rules $\pi_i = (d_i^0, \ldots, d_i^{T-1})$. A local decision rule at step t, denoted d_i^t, is a mapping from t-step action and observation histories of agent i, denoted $\theta_i^t = (a_i^0, z_i^1, \ldots, a_i^{t-1}, z_i^t)$, to local actions of agent i. In many restricted settings, decision rules depend only upon state features rather than histories, this is mainly because agents can directly observe these state features [12]. In general, however, decision rules depend on action and observation histories.

A T-step joint policy, denoted π, is an N-tuple of T-step local policies $(\pi_1, \ldots, \pi_N)$, one for each agent. It is also a length-T sequence of joint decision rules $(d^0, \ldots, d^{T-1})$. A joint decision rule at step t, denoted d^t, is an N-tuple of local decision rules $(d_1^t, \ldots, d_N^t)$, one for each agent. It is also a mapping from t-step joint action and observation histories to joint actions. A t-step joint action and observation history, denoted θ^t, is an N-tuple of local action and observation histories $(\theta_1^t, \ldots, \theta_N^t)$, one for each agent.

We consider finite-horizon Dec-POMDPs, where the optimality criterion is to maximize the expected sum of rewards over finite steps T. Let π be a joint policy. The value function at step t, denoted υ_π^t, maps state and joint history pairs to reals:

$$\upsilon_\pi^t(s^t, \theta^t) \quad = \quad \mathbb{E}[\textstyle\sum_{\tau=t}^{T-1} r^{a^\tau}(s^\tau) \mid a^\tau = d^\tau(\theta^\tau), \pi],$$

for any step t state s^t and joint history θ^t. An optimal joint policy π^*, starting at η^0, satisfies equation: $\pi^* \in \arg\max_\pi \upsilon_\pi^0(\eta^0)$. Value functions $\upsilon_{\pi^*}^0, \ldots, \upsilon_{\pi^*}^{T-1}$ are optimal value functions with respect to η^0. At first glance, these value functions exhibit no structural restrictions on their shapes. A recent analysis, however, reveals that they are linear over some high-dimensional space.

2.3 Dec-POMDPs as Continuous-State MDPs

A common assumption in many Dec-POMDPs is that planning takes place in a centralized (offline) manner even though agents execute actions in a decentralized fashion (online). In such a planning paradigm, a centralized algorithm maintains, at each time step, the total available information it has about the process to be controlled. We call the information collected at the end of time step $t-1$, the step t *information state*.

A step t information state, is a sequence $(\eta^0, d^0, \ldots, d^{t-1})$ of past joint decision rules starting with the initial distribution η^0 and is denoted by ι^t. It further satisfies the following recursion: $\iota^0 = (\eta^0)$ and $\iota^{t+1} = (\iota^t, d^t)$ for $t \in [1\colon T-1]$. With the step t information state ι^t as a background, a centralized algorithm selects the step t joint decision rule d^t, transititions to the next-step information state $\iota^{t+1} = (\iota^t, d^t)$, and finally collects the immediate reward. If we repeat this process over T steps starting with information state ι^0, it describes a deterministic MDP that represents the original Dec-POMDP $\mathcal{P}$.

Definition 2. *Let $\mathcal{P}' \equiv (I, D, F, R, \iota^0)$ be the deterministic MDP with respect to $\mathcal{P}$ where: $I = \{I^t \colon t \in [0\colon T-1]\}$ is the information state set; I^t defines the step t information state set; $D = \{D^t \colon t \in [0\colon T-1]\}$ is the joint decision rule set, where D^t denotes the step t joint decision rule set; F specifies the next-step information state ι^{t+1} after taking joint decision rule d^t in information state ι^t: $F(\iota^t, d^t) = (\iota^t, d^t)$; R specifies the immediate expected reward to be gained by executing a joint decision rule d^t in information state ι^t: $R(\iota^t, d^t) = \sum_{s,\theta} P(s, \theta | \iota^t) \cdot r^{d^t(\theta)}(s)$; and ι^0 is the initial information state.*

It is worth noting that in constructing $\mathcal{P}'$, we use the transition, observation and reward models from $\mathcal{P}$. In particular, we need to compute the entire multivariate probability distribution $P(s, \theta|\iota^t)$ over all states and joint histories, in order to estimate the immediate rewards. This operation is often time-consuming because, in practice, it involves a large number of variables. As this operation occurs every time step, it is important to reduce the time required. To this end, Dibangoye et al. [12, 13, 11] and Oliehoek [25] introduced sufficient statistics with respect to information states. Such a statistic can retain problem features that are important for calculating rewards. Informally, a sufficient statistic with respect to information state ι and $\mathcal{P}'$ is a statistic that summarizes ι and preserves the ability to find an optimal solution of $\mathcal{P}'$. Given a sufficient statistic with respect to the current information state and the problem at hand, no additional data about the current information state would provide any further information about the problem. A formal definition follows.

THEOREM 1 ([11]). *A t-step sufficient statistic with respect to information state ι^t, which we call an* ***occupancy state*** *and denote η^t, is a probability distribution over all states and joint histories, $\eta^t(s, \theta) = P(s, \theta|\iota^t)$, for any state s and joint history θ.*

The next-step occupancy state $F(\eta^t, d^t) = \eta^{t+1}$ depends on the current occupancy state η^t and joint decision rule d^t:

$$\eta^{t+1}(s', (\theta, a, z)) \quad = \quad \mathbf{1}_{\{a\}}(d^t(\theta)) \textstyle\sum_{s \in S} \eta^t(s, \theta) \cdot p^{a,z}(s, s'),$$

where $\mathbf{1}_F$ is the indicator function, and for all states $s' \in S$, joint actions $a \in A$, joint observations $z \in Z$, and joint histories θ.

DEFINITION 3. *Let $\mathcal{P}'' \equiv (\triangle, D, F, R, \eta^0)$ be the MDP with respect to $\mathcal{P}'$, which we call the* ***occupancy Markov decision process****: where $\triangle = \{\triangle^t : t \in [0: T-1]\}$ is the set of occupancy states, $\triangle^t$ is the step t occupancy state set; and D, F, R, η^0 are identical to $\mathcal{P}'$ or eventually $\mathcal{P}$.*

Relative to $\mathcal{P}'$, the occupancy MDP $\mathcal{P}''$ is a deterministic and continuous-state MDP. An optimal joint policy for $\mathcal{P}''$, together with the correct estimation of the occupancy states, will give rise to an optimal behavior for $\mathcal{P}'$ and $\mathcal{P}$ [11]. One can solve either $\mathcal{P}'$ or $\mathcal{P}''$, and nevertheless provide an optimal solution for the original problem $\mathcal{P}$ [26, 11].

2.4 Solving Occupancy MDPs

POMDPs can be cast into continuous-state MDPs with piecewise-linearity and convexity structure of the optimal value functions [31]. As we discuss next, because the occupancy MDP represents a deterministic and continuous-state MDP with a piecewise-linear convex value function, POMDP theory and algorithms can be used.

2.4.1 Properties of Optimal Value Functions

In this section, we review the property of the optimal value functions in general Dec-POMDP settings. We start with the necessary condition for optimality in occupancy MDPs.

LEMMA 1. *The* ***optimality equation*** *for any occupancy state η^t is written as follows: for all $t \in [0: T-1]$,*

$$\upsilon_*^t(\eta^t) \quad = \quad \max_{d^t} \ \left(R(\eta^t, d^t) + \upsilon_*^{t+1}(F(\eta^t, d^t))\right).$$

For $t = T$, we add a boundary condition $\upsilon_*^T = 0$.

Dibangoye et al. [11] proved that value functions $\upsilon_*^0, \ldots, \upsilon_*^{T-1}$, which are solutions of the optimality equations (Lemma 1), are piecewise-linear and convex functions of the occupancy states. That is, there exist finite sets of linear functions $\Lambda^0, \ldots, \Lambda^{T-1}$ such that: $\upsilon_*^t(\eta^t) = \max_{\alpha^t \in \Lambda^t} \langle \alpha^t, \eta^t \rangle$ (where notation $\langle \cdot, \cdot \rangle$ is the inner-product), for any arbitrary t-step occupancy state η^t.

Algorithm 1: The FB-HSVI Algorithm.

```
function FB-HSVI()
    initialize $\underline{\upsilon}_t$ and $\bar{\upsilon}_t$ for all $t \in \{0, \cdots, T-1\}$.
    while $\neg$Stop$(\eta_0, 0)$ do Explore$(\eta_0, 0)$ ;

function Explore$(\eta_t, g_t)$
    $\tilde{\eta}_t \leftarrow$ Compact$(\eta_t)$.
    if $\neg$Stop$(\tilde{\eta}_t, g_t)$ then
        $d_t^* \in \arg\max_{d_t} R(\tilde{\eta}_t, d_t) + \bar{\upsilon}_{t+1}(F(\tilde{\eta}_t, d_t))$.
        Update $\bar{\upsilon}_t$.
        Explore$(F(\tilde{\eta}_t, d_t^*), R(\tilde{\eta}_t, d_t^*) + g_t)$.
        Update $\underline{\upsilon}_t$.
    return $g_t$

function Stop$(\eta_t, g_t)$
    if $\bar{\upsilon}_t(\eta_t) > \underline{\upsilon}_t(\eta_t)$ then return $g_t + \bar{\upsilon}_t(\eta_t) \leq \underline{\upsilon}_t(\eta_0)$ ;
    return true
```

2.4.2 The FB-HSVI Algorithm

The *heuristic search value iteration* (HSVI) algorithm is a leading POMDP solver which performs well on many POMDP domains, while preserving the ability to eventually find an optimal solution [32]. By recasting Dec-POMDPs as occupancy MDPs, the HSVI algorithm (as well as other POMDP algorithms) can be extended to solve Dec-POMDPs.

Dibangoye et al. [11] introduced feature-based HSVI (FB-HSVI), which is shown in Algorithm 1, to improve the efficiency of the HSVI algorithm in occupancy MDPs. It uses a trial-based best-first search and finds an optimal path from a given initial occupancy state to one T-step occupancy state. It traverses the search space by creating trajectories of occupancy states, each of which starts with the initial occupancy state. For each visited occupancy state, such trajectories always follow the best joint decision rule (ties are broken arbitrarily) specified by the upper bounds $(\bar{\upsilon}_t)_{t \in \{0,\ldots,T\}}$. As the algorithm traverses the search space, it updates the upper bounds of the occupancy states along the way. Once the trajectories are finished, it maintains lower bounds $(\underline{\upsilon}_t)_{t \in \{0,\ldots,T\}}$ of visited occupancy states in reverse order.

The FB-HSVI algorithm provably converges to optimal value functions with respect to the initial occupancy state. As it seeks the occupancy states where the upper bound is the largest, and maintains both upper and lower bounds, it reduces the gap between bounds over the initial occupancy state at each iteration. Once the gap is zero, the algorithm has converged. Moreover, the FB-HSVI algorithm guarantees termination after a finite number of iterations, although this number is (in the worst case) doubly exponential in the maximal length of a trajectory.

2.4.3 Key Limitations of FB-HSVI

The FB-HSVI algorithm demonstrated a significant improvement in performance on many domains, while preserving the ability to eventually find an optimal solution. Its scalability is nonetheless limited when the number of agents or problem variables is quite large. To better understand this, notice that the complexity of the FB-HSVI algorithm depends essentially on two operations: the *decision rule selection*; and the *information tracking*. In either case, the FB-HSVI algorithm is not geared to exploit the locality of interaction, and thus, it will typically have to consider decision rules and occupancy states over exponentially many variables, though multiple variables have little influence on one another.

In order to improve the scalability in the number of agents, there has been a growing interest in research for solving Dec-POMDPs that exhibit locality of interaction [19, 21, 34]. This property appears in many practical applications, including domains represented as ND-POMDPs [24]. Unfortunately, the expressiveness of this framework comes with a price, solving finite-horizon ND-POMDPs optimally is also NEXP-complete [5, 24]. This partially explains why the only optimal algorithm, namely *the global optimal algorithm* (GOA) [24], can often solve problems with a couple of agents, but cannot handle domains with larger number of agents. The other reason for this poor scaling behavior resides in the explicit enumeration of exponentially many policies, which though it ensures optimality, is often unnecessary or redundant.

Recently, approximate algorithms have been used to solve ND-POMDPs using ideas such as locally optimal heuristic search [23, 24, 34], and constraint-based dynamic programming [20, 21]. *To the best of our knowledge, none of these approaches can provide tight performance guarantees (error bounds or potential losses), features that are critical in a large range of real-world applications related to the military, environment, medical domains or social services.* In the remainder of this paper, we discuss an extension of FB-HSVI so it can exploit locality of interaction, enhancing its performance on domains with larger numbers of agents, while preserving optimality.

3. LEVERAGING SEPARABILITY

In this section, we discuss how locality of interaction through separability assumptions (Definitions 1) influences the structure of value functions and occupancy states.

3.1 Separable Value Functions

The primary contribution is a proof that the optimal value function is the sum of linear functions over factors, a property referred to as the additive weak separability and linearity. A formal definition of this property follows.

DEFINITION 4. *Value function g is* ***additively weakly separable and linear****, if there exist linear functions $g_{u_1}, g_{u_2}, \dots, g_{u_M}$ such that: $g(s,\theta) = \sum_{k=1}^{M} g_{u_k}(s_{u_k}, \theta_{u_k})$, $\quad u_1, \dots, u_M \subseteq [1\colon N]$. Value function g is said to be* ***additively fully separable and linear****, if $u_k \cap u_{k'} = \emptyset$ for all $k, k' \in [1\colon M]$.*

An optimization problem with an additively fully separable and linear objective function g can be reduced to M independent optimization problems with lower dimensionalities. If g is not fully separable, we often search the whole N-dimensional space all at once. However, algorithms that exploit the weak separability when it is present have been particularly successful, notable examples include weighted constraint satisfaction algorithms [8, 9, 10]. In the following, we present the proof that optimal value functions are AWSL functions of the occupancy states. Before proceeding any further, we introduce short-hand notation $g_{u_k|\theta_{u_k}}$ to represent a function over states s_{u_k} s.t.: $g_{u_k|\theta_{u_k}}(s_{u_k}) = g_{u_k}(s_{u_k}, \theta_{u_k})$.

THEOREM 2. *Value functions $(\upsilon^t_\pi)_{t\in[1\colon T-1]}$, for any joint policy π, are additively weakly separable and linear functions of occupancy states. That is, there exist vectors $(\alpha^t_{u_k|\theta_{u_k}})_{\theta,k\in[1\colon M]}$ s.t.*

$$\upsilon^t_\pi(\eta^t) \quad = \quad \textstyle\sum_u \sum_{s_u} \sum_{\theta_u} \eta^t_{u|\theta_u}(s_u) \cdot \alpha^t_{u_k|\theta_{u_k}}(s_u),$$

where $\eta^t_{u|\theta_u}(s_u) = \sum_{s_{-u},\theta_{-u}} \eta^t(s,\theta)$ for any η^t and $u \subseteq [1\colon N]$.

PROOF. The statement trivially holds for $t = T$, as there is no future rewards. Assume it holds for $t \geq \tau$, that is, for any arbitrary τ-step occupancy state η^τ, the following holds:

$$\upsilon^\tau_\pi(\eta^\tau) \quad = \quad \textstyle\sum_u \sum_{s_u} \sum_{\theta_u} \eta^\tau_{u|\theta_u}(s_u) \cdot \alpha^\tau_{u_k|\theta_{u_k}}(s_u).$$

Let $t = \tau - 1$. We first show that reward vectors in r are additively weakly separable and linear functions of occupancy states. Indeed, the following holds: $R(\eta^{\tau-1}, d^{\tau-1})$

$$\begin{aligned}
&= \textstyle\sum_\theta \sum_s \eta^{\tau-1}(s,\theta) \sum_u r_u^{d_u^{\tau-1}(\theta_u)}(s_u),\\
&= \textstyle\sum_u \sum_{s_u} \sum_{\theta_u} \left(\sum_{s_{-u}} \sum_{\theta_{-u}} \eta^{\tau-1}(s,\theta)\right) r_u^{d_u^{\tau-1}(\theta_u)}(s_u),\\
&= \textstyle\sum_u \sum_{s_u} \sum_{\theta_u} \eta^{\tau-1}_{u|\theta_u}(s_u) \cdot r_u^{d_u^{\tau-1}(\theta_u)}(s_u).
\end{aligned}$$

Next, we exploit the fact that the value function υ^τ_π is an AWSL function of occupancy states. We have $\upsilon^\tau_\pi(F(\eta^{\tau-1}, d^{\tau-1}))$

$$\begin{aligned}
&= \textstyle\sum_u \sum_{s'_u} \sum_{\theta_u} \alpha^\tau_{u|\theta_u}(s'_u) \cdot \eta^\tau_{u|\theta_u}(s'_u),\\
&= \textstyle\sum_u \sum_{s_u,\theta_u} \eta^{\tau-1}_{u|\theta_u}(s_u) \sum_{s'_u,z_u} \alpha^\tau_{u|\theta'_u}(s'_u) \cdot p_u^{d_u^{\tau-1}(\theta_u),z_u}(s_u, s'_u),\\
&= \textstyle\sum_u \sum_{s_u} \sum_{\theta_u} \eta^{\tau-1}_{u|\theta_u}(s_u) \cdot \alpha^{\tau-1}_{u|\theta_u}(s_u),
\end{aligned}$$

where $\alpha^{\tau-1}_{u|\theta_u}(s_u) = \sum_{s'_u,z_u} \alpha^\tau_{u|\theta'_u}(s'_u) \cdot p_u^{d_u^{\tau-1}(\theta_u),z_u}(s_u, s'_u)$ and $\theta'_u = (\theta_u, d_u^{\tau-1}(\theta_u), z_u)$. Finally, by combining immediate and future rewards we obtain: $\upsilon^{\tau-1}_\pi(\eta^{\tau-1})$

$$\begin{aligned}
&= R(\eta^{\tau-1}, d^{\tau-1}) + \upsilon^\tau_\pi(F(\eta^{\tau-1}, d^{\tau-1})),\\
&= \textstyle\sum_u \sum_{s_u} \sum_{\theta_u} \eta^{\tau-1}_{u|\theta_u}(s_u) \cdot \left(r_u^{d_u^{\tau-1}(\theta_u)}(s_u) + \alpha^{\tau-1}_{u|\theta_u}(s_u)\right),
\end{aligned}$$

which ends the proof. □

This theorem demonstrates that value functions can be represented using a finite set of low-dimensional vectors, one $|S_u|$-length vector $\alpha_{u|\theta_u}$ for each joint history θ_u. This result extends a previous separability property of the value function for ND-POMDPs [24], which stated that value functions of a specified joint policy can be decomposed into the sum of value functions over factors. Relative to the PWLC property of value function solutions of the optimality equations, the AWSL property provides a significant restrictive structure in the shape of value functions. It is nevertheless unclear how this property can improve efficiency of the FB-HSVI algorithm. In addition, this theorem yields interesting insights. It is worth noticing that this result holds even when there exists a unique factor $u = [1\colon N]$, that is, in general DecPOMDPs.

COROLLARY 1. *Value functions $(\upsilon^t_\pi)_{t\in[1\colon T-1]}$, for any joint policy π, are additively weakly separable and linear functions of occupancy states. That is, there exist vectors $(\alpha^t_{|\theta})_{\theta,t\in[0\colon T-1]}$ such that $\upsilon^t_\pi(\eta^t) = \sum_s \sum_\theta \eta^t_{|\theta}(s) \cdot \alpha^t_{|\theta}(s)$, where $\eta^t_{|\theta}(s) = \eta^t(s,\theta)$ for any arbitrary occupancy state η^t.*

PROOF. The proof holds directly from Theorem 2 with a single factor $u = [1\colon N]$. □

3.2 Separable Sufficient Statistics

Another important result from Theorem 2 is a proof that value functions depend on occupancy states only through marginal probability distributions over factors. This is a significant result as it allows us to maintain marginal probability distributions independently from one another, which saves non-negligible time and memory, while preserving optimality.

THEOREM 3. *For any ND-POMDP with factors $u_1, \dots, u_M$, marginal occupancy states $(\eta_{u_k|\theta_{u_k}})_{u_k,\theta_{u_k}}$ collectively constitute a sufficient statistic of occupancy state η. Marginal occupancy state*

$\eta_{u|\theta_u}$ can be updated at each step to incorporate the latest action a_u and observation z_u, where:

$$\eta_{u|\theta_u,a_u,z_u}(s'_u) \quad = \quad \sum_{s_u} \eta_{u|\theta_u}(s_u) \cdot p_u^{a_u,z_u}(s_u, s'_u).$$

PROOF. A careful look at Theorem 2 reveals that value functions depend on occupancy states only through marginal occupancy states. In addition, the multiplicative weak separability of dynamics model p allows us to maintain marginal occupancy states independently from one another. Initially, $\eta_u^0(s_u) = \eta_0^0(s_0) \prod_{i\in u} \eta_i^0(s_i)$; then $\eta = \eta_{u|\theta_u,a_u,z_u}$ satisfies the following recursive formula: $\eta(s'_u) = P(s'_u, \theta_u, a_u, z_u|\eta_u^0)$

$$\begin{aligned}
&= P((s'_0, s'_i, \theta_i, a_i, z_i)_{i\in u}|\eta_u^0),\\
&= \textstyle\sum_{s_u} P((z_i)_{i\in u}|(s_0, s_i, a_i, s'_0, s'_i)_{i\in u}) \cdot P((s_0, s_i, \theta_i)_{i\in u}|\eta_u^0),\\
&= \textstyle\sum_{s_u} \eta_{u|\theta_u}(s_u) \cdot p_0(s_0, s'_0) \prod_{i\in u} p_i^{a_i,z_i}(s_i, s_0, s'_i),
\end{aligned}$$

which ends the proof. □

This theorem permits us to circumvent unnecessary or redundant operations when maintaining the occupancy states. In particular, we can maintain marginal occupancy states independently from one another, and reuse pre-computed ones when it is possible. The following describes a novel representation of bounds in the FB-HSVI algorithm based on the AWSL property. To this end, the marginal occupancy states $(\eta_{u_k|\theta_{u_k}})_{u_k,\theta_{u_k}}$ are collectively referred to as a **separable occupancy state**.

4. AWSL BOUND REPRESENTATIONS

To address the key limitations in the FB-HSVI algorithm (see Section 2.4.3), we exploit the AWSL property. In particular, we introduce representations that can significantly reduce the memory required to maintain lower and upper bounds. We further show that these novel representations permit the FB-HSVI algorithm to scale up to ND-POMDPs of unprecedented size; enhancing the generalization of the bounds over unvisited regions of the search space; and speeding up the convergence to an optimal solution.

4.1 Lower-Bound Value Functions

The standard lower-bound representation uses sets $(\Lambda^t)_{t\in[0:\,T-1]}$ of linear functions, where each linear function $\alpha^t \in \Lambda^t$ maps from state and joint history pairs to reals [11]. In this form, lower bounds are updated as follows. Each trajectory of the FB-HSVI algorithm generates a joint policy π, which in turn produces linear functions $(\upsilon_\pi^t)_{t\in[0:\,T-1]}$. When a trajectory is finished, the algorithm adds linear functions $(\upsilon_\pi^t)_{t\in[0:\,T-1]}$ into the current representation. This update rule ensures a monotonic improvement of lower-bounds at the initial occupancy state over trials. It is nevertheless time and memory demanding to compute and maintain the standard representation.

To reduce the overwhelming time and memory requirements of the standard representation, we exploit the AWSL property. In particular, our lower-bound representation uses a finite set of length-$|S_u|$ vectors $\Lambda = \{\alpha_{u|\theta_u} : \forall u, \theta_u\}$ associated with a single joint policy π, such that, for t-step occupancy state η, we obtain $\underline{\upsilon}^t(\eta) = \upsilon_\pi^t(\eta) = \sum_u \sum_{\theta_u} \langle \alpha_{u|\theta_u}, \eta_{u|\theta_u}\rangle$. The update rule we use to maintain our lower-bound representation follows.

For any joint policy π, we compute vectors $\alpha_{u|\theta_u}$ using backward induction: for any state s_u,

$$\alpha_{u|\theta_u}(s_u) = r_u^{d_u(\theta_u)}(s_u) + \textstyle\sum_{z_u} p_u^{d_u(\theta_u),z_u}(s_u, s'_u)\alpha_{u|\theta'_u}(s'_u)$$

where $\theta'_u = (\theta_u, d_u(\theta_u), z_u)$. We set $\alpha_{u|\theta'_u} = \alpha_{u|t}$ for t-step histories θ'_u that are unreachable when following π. Vector $\alpha_{u|t}$ maps from states s_u to any trivial lower-bound, *e.g.*, $\alpha_{u|t}(s_u) = \min_{a_u}(T - t)r_u^{a_u}(s_u)$. If the value at the initial occupancy state $\upsilon_\pi^0(\eta^0)$ is greater than the current lower-bound $\underline{\upsilon}^0(\eta^0)$, then we replace $(\underline{\upsilon}^t)_{t\in[0:\,T-1]}$ by $(\upsilon_\pi^t)_{t\in[0:\,T-1]}$.

Like the standard representation, set Λ can accurately represent value functions of any optimal joint policy π. Our representation is nevertheless more compact. Where the standard representation keeps track of value functions associated with many different joint policies, our representation maintains only the value function of the current best joint policy. In addition, the associated update rule is more efficient, since it involves only states and histories in a single factor. It is worth noting that this representation comes with one drawback: it often yields lower-bound values that are weaker than those from the standard representation. This looseness may slow down the rate of convergence.

4.2 Upper-Bound Value Functions

The standard upper-bound representation is a mapping from visited occupancy states to upper-bounds. It distinguishes between *corner* and *non-corner* occupancy states. A corner occupancy state is a degenerate distribution, that is, the probability mass is localized at a single state and joint history pair, and zero otherwise. An occupancy state that is not a corner occupancy state, is a non-corner occupancy state. We use point set $\{(\eta^\ell \mapsto \beta^\ell) \colon \ell \in [1\colon L]\}$ to denote mapping from non-corner occupancy state to upper-bounds; and β^0 to represent the mapping from corner occupancy states to upper-bounds. Every time FB-HSVI encounters an occupancy state, it uses the point-set representation to estimate the upper-bound of the current occupancy state. Given t-step occupancy state η, the sawtooth interpolation [11, 32] yields an upper-bound value at η:

$$\bar{\upsilon}^t(\eta) \quad = \quad \min_\ell \left(\beta^0(\eta) + \delta(\eta, \eta^\ell) \cdot (\beta^\ell - \beta^0(\eta^\ell))\right),$$

where $\delta(\eta, \eta^\ell) = \min\{\eta(s, \theta)/\eta^\ell(s, \theta) \colon \eta^\ell(s, \theta) > 0\}$ is referred to as the interpolation coefficient. Notice that lower interpolation coefficients lead to weaker bounds. The update rule consists of adding a new point in the point set, using the greedy joint decision rule selection and the sawtooth interpolation. In this paper, we demonstrate that by using the AWSL property together with the sawtooth interpolation, one can produce tighter upper-bound values.

4.2.1 The Novel Representation

In this section, we extend the standard representation to exploit the AWSL property. In particular, we use separable occupancy states $\eta^\ell \equiv (\eta^\ell_{u|\theta_u})_{u,\theta_u}$ instead of full occupancy states; and replace upper bounds by point sets $\beta^\ell \equiv \{(\eta^\ell_{u|\theta_u} \mapsto \beta^\ell_{u|\theta_u}) \colon \forall u, \theta_u\}$. Like the standard representation, we distinguish between corner and non-corner separable occupancy states. A corner separable occupancy state corresponds to a corner occupancy state. We call any separable occupancy state that is not a corner separable occupancy state a non-corner separable occupancy state. Hence, we use β^0 to represent a mapping from corner separable occupancy states to upper bounds; unlike the standard representation, we use point set $\Gamma = \{(\eta^\ell \mapsto \beta^\ell) \colon \ell \in [1\colon L]\}$ to represent a mapping from separable occupancy states to point sets. These point sets $\beta^\ell \equiv \{(\eta^\ell_{u|\theta_u} \mapsto \beta^\ell_{u|\theta_u}) \colon \forall u, \theta_u\}$ represent mappings from marginal occupancy states to upper bounds, one point set for each separable occupancy state η^ℓ.

Initially, the point set Γ contains only corner points, that is, mappings from corner separable occupancy states to upper-bounds. To construct the initial point set, a general rule of thumb is to use the optimal value functions of a relaxation of the problem at hand. Here, we use the optimal value functions $(\upsilon^t_{\text{MDP}})_{t\in[0:\,T-1]}$ of the underlying MDP. Thus, the initial upper-bound values are given by: $\beta^0(s, \theta) = \upsilon^t_{\text{MDP}}(s)$, for any state s and joint history θ. Notice that mapping $\beta^0 = \sum_u \sum_{\theta_u} \beta^0_{u|\theta_u}$, that is, β^0 is AWSL. Next, we extend the sawtooth interpolation to exploit the AWSL property.

4.2.2 Enhancing Evaluations

This section extends the sawtooth interpolation using our upper-bound representation. In particular, we explore applying the sawtooth interpolation to marginal occupancy states instead of full occupancy states. That is, we demonstrate how to compute an upper-bound of a marginal occupancy state based on another one. To this end, we introduce the concept of *policy equivalence*. Two histories θ_u and $\bar{\theta}_u$ that are different, can nonetheless have the same future optimal policy. In this case, we say that θ_u and $\bar{\theta}_u$ are policy equivalent. For a thorough discussion on policy equivalence relations, the reader can refer to [11, 26]. Policy-equivalent histories can extrapolate their upper-bound values from one another [11]. Our extension of the sawtooth interpolation follows.

LEMMA 2. *Let $\alpha_{u|\theta_u}$ be the optimal linear function relative to factor u and history θ_u, $(\eta^\ell_{u|\theta_u}, \beta^\ell_{u|\theta_u})$ be a non-corner point, and $\eta_{u|\bar{\theta}_u}$ be a marginal occupancy state. For each marginal occupancy state $\eta_{u|\bar{\theta}_u}$, the following holds: $\upsilon_{\pi^*_u}(\eta_{u|\bar{\theta}_u}) \leq \beta_{u|\bar{\theta}_u}$, where for all factor u and history $\bar{\theta}_u$ that is policy equivalent to θ_u,*

$$\beta_{u|\bar{\theta}_u} = \beta^0_{u|\theta_u}(\eta_{u|\bar{\theta}_u}) + \delta(\eta_{u|\bar{\theta}_u}, \eta^\ell_{u|\theta_u})(\beta^\ell_{u|\theta_u} - \beta^0_{u|\theta_u}(\eta^\ell_{u|\theta_u})),$$

and $\delta(\eta_{u|\bar{\theta}_u}, \eta^\ell_{u|\theta_u}) = \min_{s_u}\{\frac{\eta_{u|\bar{\theta}_u}(s_u)}{\eta^\ell_{u|\theta_u}(s_u)} \colon \eta^\ell_{u|\theta_u}(s_u) > 0\}$.

PROOF. We first note that marginal occupancy state $\eta_{u|\bar{\theta}_u}$ can be written as a linear combination between two $|S_u|$-dimensional and positive vectors y and $\eta^\ell_{u|\theta_u}$, and some positive number δ. That is, $\eta_{u|\bar{\theta}_u} = y + \delta \cdot \eta^\ell_{u|\theta_u}$. We further note that inequalities $\alpha_{u|\theta_u}(\eta^\ell_{u|\theta_u}) \leq \beta^\ell_{u|\theta_u}$ and $\alpha_{u|\theta_u}(y) \leq \beta^0_{u|\theta_u}(y)$ hold. By linearity of $\alpha_{u|\theta_u}$, we know the following holds:

$$\alpha_{u|\theta_u}(\eta_{u|\bar{\theta}_u}) \quad \leq \quad \beta^0_{u|\theta_u}(y) + \delta \cdot \beta^\ell_{u|\theta_u}.$$

If we replace y by $(\eta_{u|\bar{\theta}_u} - \delta \cdot \eta_{u|\theta_u})$, and rearrange terms:

$$\begin{aligned} \alpha_{u|\theta_u}(\eta_{u|\bar{\theta}_u}) &\leq \beta^0_{u|\theta_u}(\eta_{u|\bar{\theta}_u} - \delta \cdot \eta^\ell_{u|\theta_u}) + \delta \cdot \beta^\ell_{u|\theta_u}, \\ &\leq \beta^0_{u|\theta_u}(\eta_{u|\bar{\theta}_u}) + \delta(\beta^\ell_{u|\theta_u} - \beta^0_{u|\theta_u}(\eta^\ell_{u|\theta_u})). \end{aligned}$$

To get the best upper-bound value, we wish to find the maximum value of δ, that is consistent with our assumptions. In particular, we need find δ consistent with: $y(s_u) \geq 0$ and $\eta^\ell_{u|\theta_u}(s_u) \geq 0$, for any state s_u. Since $y(s_u) = \eta_{u|\bar{\theta}_u}(s_u) - \delta \cdot \eta^\ell_{u|\theta_u}(s_u)$, we obtain expression: $\delta = \min_{s_u}\{\frac{\eta_{u|\bar{\theta}_u}(s_u)}{\eta^\ell_{u|\theta_u}(s_u)} \colon \eta^\ell_{u|\theta_u}(s_u) > 0\}$. □

This lemma presents a formula that assigns an upper bound to any specified marginal occupancy state. However, to preserve theoretical guarantees, it is crucial to perform the assignments of upper bounds to marginal occupancy states all at once. This is mainly because marginal occupancy states in separable occupancy states have values that depend on one another. The assignment rule for separable occupancy state $\eta \equiv (\eta_{u|\bar{\theta}_u})_{u,\bar{\theta}_u}$ given point set $\Lambda = \{(\eta^\ell \mapsto \beta^\ell) \colon \ell \in [1\colon L]\}$ follows:

$$(\beta_{u|\bar{\theta}_u})_{u,\bar{\theta}_u} \quad = \quad \operatorname*{arg\,min}_{(\beta^\ell_{u|\bar{\theta}_u})_{u,\bar{\theta}_u} \colon \ell \in [1\colon L]} \textstyle\sum_u \sum_{\bar{\theta}_u} \beta^\ell_{u|\bar{\theta}_u},$$

where $\beta^\ell_{u|\bar{\theta}_u}$ denotes the upper-bound value of $\eta_{u|\bar{\theta}_u}$ extrapolated based on point $(\eta^\ell \mapsto \beta^\ell)$. Thus, the upper-bound value at t-step separable occupancy state η is given by: $\bar{\upsilon}^t(\eta) = \sum_{u,\bar{\theta}_u} \beta_{u|\bar{\theta}_u}$. Notice that our upper bound is tighter or equal to that of the standard representation, as $\delta(\eta, \eta^\ell) = \min_u \delta(\eta_{u|\bar{\theta}_u}, \eta^\ell_{u|\theta_u})$.

4.2.3 Constraint-Based Decision Rule Selections

In this section, we extend the greedy joint decision rule selection to exploit our upper-bound representation. Similar to the standard FB-HSVI algorithm, we formulate and solve a weighted constraint satisfaction problem (WCSP).

A WCSP refers to a tuple $(V, \mathbf{X}, C)$ where: $V = \{V_1, \ldots, V_M\}$ is the set of M domains; $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$ is the set of M variables, taking values from their domains. C is the set of reward functions used to declare preferences among possible solutions. Each reward function $c \in C$ is defined over a subset of variables, $var(c) \subseteq \mathbf{X}$, called the scope. The objective function f is defined as the sum of all reward functions in C, that is, $f(\mathbf{X}) = \sum_{c \in C} c(\mathbf{X}_{var(c)})$. When variables are correctly assigned, finite rewards are received that express their degree of preference (higher value equates to higher preference) and when variables are not correctly assigned a reward $-\infty$ is received. The goal of this problem is to find a mapping from variables to values, which maximizes the objective function.

As we demonstrate later, to select the greedy joint decision rule, we consider L different WCSPs, each of which relies on a single non-corner point $(\eta^\ell \mapsto \beta^\ell)$ from our point-set representation $\Gamma = \{(\eta^\ell \mapsto \beta^\ell) \colon \ell \in [1\colon L]\}$. Each WCSP returns a joint decision rule, but the greedy decision rule is the one with the highest objective value (ties are broken arbitrarily). A formal definition of the WCSP relative to non-corner point $(\eta^\ell \mapsto \beta^\ell)$ follows.

DEFINITION 5. *Let $(\eta^\ell \mapsto \beta^\ell)$ be a non-corner point and η be a separable occupancy state. The ℓ-th WCSP $(\mathbf{X}, V, C^\ell)$ involves: $V = \{V_{u|\bar{\theta}_u} \colon u, \bar{\theta}_u\}$ consists of sets $V_{u|\bar{\theta}_u}$ of mappings from histories to actions $(\bar{\theta}_u \mapsto a_u)$; $\mathbf{X} = \{\mathbf{X}_{u|\bar{\theta}_u} \colon u, \bar{\theta}_u\}$ is the set of variables $\mathbf{X}_{u|\bar{\theta}_u}$, taking values from their domains $V_{u|\bar{\theta}_u}$; $C^\ell = \{nogood, c^\ell_{u|\bar{\theta}_u} \colon u, \bar{\theta}_u\}$ is a set of reward functions, for any arbitrary mapping $(\bar{\theta}_u \mapsto a_u)$, we have:*

$$c^\ell_{u|\bar{\theta}_u}(\bar{\theta}_u \mapsto a_u) \quad = \quad r^{a_u}_u(\eta_{u|\bar{\theta}_u}) + \textstyle\sum_{z_u} \beta^\ell_{u|\bar{\theta}_u, a_u, z_u};$$

The objective function f^ℓ is defined as the sum of all reward functions in C^ℓ, that is, $f^\ell(\mathbf{X}) = \sum_u \sum_{\bar{\theta}_u} c^\ell_{u|\bar{\theta}_u}(\mathbf{X}_{u|\bar{\theta}_u})$.

The following theorem states that the greedy decision rule corresponds to the solution of one of these WCSPs.

THEOREM 4. *A greedy joint decision rule for separable occupancy state $\eta \equiv (\eta_{u|\bar{\theta}_u})_{u,\bar{\theta}_u}$, is the solution with the maximum rewards among the solutions of WCSPs $(\mathbf{X}, V, C^\ell)_{\ell \in [1\colon L]}$.*

PROOF. We start with the standard joint decision rule selection for a specified occupancy state η^t:

$$d^t_* \quad = \quad \arg\max_{d^t} R(\eta^t, d^t) + \bar{\upsilon}^{t+1}(F(\eta^t, d^t)).$$

Next, we exploit the additive weak separability and linearity of the value functions. Using this property, we know that d^t_* is given by

$$\arg\max_{d^t} \min_\ell \textstyle\sum_{u,\bar{\theta}_u} r^{d^t_u(\bar{\theta}_u)}_u(\eta^t_{u|\bar{\theta}_u}) + \sum_{z_u} \beta^\ell_{u|\bar{\theta}_u, d^t_u(\bar{\theta}_u), z_u},$$

where $\beta^\ell_{u|\bar{\theta}_u, d^t_u(\bar{\theta}_u), z_u}$ corresponds to the upper-bound value of marginal occupancy state $\eta^t_{u|\bar{\theta}_u, d^t_u(\bar{\theta}_u), z_u}$ extrapolated based on the ℓ-th non-corner point $(\eta^\ell \mapsto \beta^\ell)$ in Γ. By Definition 5, we have that

$$\begin{aligned} d^t_* &= \arg\max_{d^t} \min_\ell \textstyle\sum_u \sum_{\bar{\theta}_u} c^\ell_{u|\bar{\theta}_u}(\bar{\theta}_u \mapsto d^t_u(\bar{\theta}_u)), \\ &= \max_\ell \arg\max_{d^t} f^\ell(d^t), \text{ s.t. } f^\ell(d^t) \leq f^l(d^t), \ \forall l \in [1\colon L] \backslash \{\ell\} \end{aligned}$$

Which ends the proof. □

5. EXPERIMENTS

We compare our extension of FB-HSVI for ND-POMDPs with the standard FB-HSVI algorithm [11], a state-of-the-art exact algorithm for solving general Dec-POMDPs. We call our extension, the separable feature-based heuristic search value iteration (SFB-HSVI) algorithm. We could not compare to the global optimal algorithm (GOA), as it quickly runs out of memory even for the smallest benchmarks. We nonetheless compare with the state-of-the-art approximate algorithms for solving ND-POMDPs, including constraint based dynamic programming (CBDP) [19], and FANS [23]. CBDP constructs joint policies based on a small selection of distributions over states. We set the number of distributions to 5 as advised in Kumar and Zilberstein [19]. FANS relies on various heuristics to build approximate joint policies. For each benchmark, we consider only the heuristic with the best performance.

T	Algorithms						
	CBDP		FANS		FB-HSVI		
	EV	CPU	EV	CPU	EV	CPU (ext.)	CPU (std.)
5-P domain — $\vert S\vert = 12$; $N = 5$, $\vert Z_i\vert = 2$, and $2 \leq \vert A_i\vert \leq 3$							
3	198.1	2	198.1	20	332.0	2.03	3.77
4	253.7	3	253.9	70	471.2	3.65	10.4
5	302.0	4	355.1	80	605.0	9.36	32.3
6	339.5	5	376.3	90	735.8	35.4	125
7	410.5	6	410.5	100	869.2	231.4	
10	558.6	9	569.4	400			
7-H domain — $\vert S\vert = 12$; $N = 7$, $\vert Z_i\vert = 2$, and $2 \leq \vert A_i\vert \leq 3$							
3	255.5	2	175.8	0.5	418.0	1.5	1.7
4	331.0	4	184.8	1.0	581.8	2.3	5.7
5	404.6	6	274.7	700	765.8	4.7	18.3
6	462.7	7	327.8	800	940.4	12.0	50.4
7	507.5	8	376.8	900	1082.8	40.4	162.6
8	561.4	9			1206.6	261	
10	658.1	10					
11-helix domain — $\vert S\vert = 49$; $N = 11$, $\vert Z_i\vert = 2$, and $2 \leq \vert A_i\vert \leq 4$							
3	328.8	20	255.0	135	554.4	3.1	
4	-	-			777.2	6.4	
5	-	-			1057.6	21.7	
6	-	-			1347.7	140.7	
7	-	-					
10	-	-					
15-3D domain — $\vert S\vert = 60$; $N = 15$, $\vert Z_i\vert = 2$, and $2 \leq \vert A_i\vert \leq 4$							
3	529.0	50	514.2	3000	814.0	4.6	
4	616.9	60			1167.0	7.9	
5	831.5	70			1587.1	22.4	
6	996.2	80			2008.0	78.3	
7	1124.7	90			2353.9	272.7	
10	1493.6	110					
15-Mod domain — $\vert S\vert = 16$; $N = 15$, $\vert Z_i\vert = 2$, and $2 \leq \vert A_i\vert \leq 4$							
3	515.9	60	367.6	200	814.0	2.0	
4	-	-			1142.5	3.5	
5	-	-			1553.2	8.6	
6	-	-			1971.2	26.6	
7	-	-			2336.5	103.8	

EV = $\upsilon^0_\pi(\eta^0)$ CPU (sec.) ‘ ’ = time (1000s) expired ‘-’ = no results available

Table 1: Performance of FB-HSVI (extended and standard versions), CBDP, and FANS. Blank spaces represent over the time or memory limits.

The experiments of FB-HSVI and SFB-HSVI were run on a Mac with a 2.2GHz Intel Core i7 CPU, 1GB of RAM available, and a time limit of one thousand seconds. We solved the WCSPs using toulbar2 [8]. The other experiments were conducted on a machine with 2.4GHz Intel dual core CPU and 1GB of RAM available. The main purpose of these experiments was to show the scalability of SFB-HSVI with respect to the number of agents. To do so, we conducted the experiments on the largest ND-POMDP benchmarks based on the sensor network domain [24, 23, 19], which range from five to fifteen agents. For a thorough discussion on the network sensor domain, the reader can refer to [24]. The other purpose of these experiments was to highlight the necessity of exact solvers in contrast to approximate methods. On each benchmark, we report value $\upsilon^0_\pi(\eta^0)$ relative to the best joint policy π each algorithm found. We also report running time in seconds for different planning horizons.

Results can be seen in Table 1. In all tested benchmarks, as depicted in column CPU (ext.), the SFB-HSVI algorithm can find an optimal joint policy for short planning horizons. In particular, it can optimally solve the largest benchmark (15-Mod) at planning horizon $T = 7$ in about one hundred seconds. The results show that the standard FB-HSVI algorithm can also find an optimal joint policy but only for medium-sized benchmarks. For instance, in 5-P and 7-H, both standard and extended FB-HSVI algorithms can find an optimal joint policy for $T \leq 6$. But SFB-HSVI is about three times faster than the standard FB-HSVI algorithm. Since the time required to compute an optimal joint policy increases with increasing planning horizons, the standard FB-HSVI algorithm always runs out of time before our extension, as illustrated in benchmark 5-P at $T = 6$, and benchmark 7-H at $T = 7$. In larger benchmarks 11-helix, 15-3D, and 15-Mod, which involve a dozen of agents, the standard FB-HSVI algorithm quickly runs out of memory, as it cannot exploit the locality of interaction.

We further compare SFB-HSVI with approximate ND-POMDP solvers CBDP and FANS. Experiments demonstrate that, although approximate methods can scale up with respect to planning horizon, they often produce poor solution quality. To illustrate this, consider benchmark 7-H at $T = 7$: CBDP takes 8 seconds and returns a joint policy with a return of 507.5; and FANS takes about 900 seconds and returns a joint policy with a return of 376.8; but, SFB-HSVI takes about 40 seconds to find an optimal joint policy with return 1082.6. Our extension provides solution quality three times higher than that of FANS, and two times higher than that of CBDP. It is worth noting that CBDP can improve solution quality by increasing the number of state distributions considered, but it cannot provide any guarantees since these distributions are not sufficient for optimal planning in ND-POMDPs.

To summarize, our experiments illustrate the scalability of SFB-HSVI with respect to the number of agents. Our algorithm optimally solves all ND-POMDP benchmarks with up to fifteen agents. These results also highlight the necessity of the exact algorithms, especially in critical domains where theoretical guarantees (error-bounds or potential losses) are required.

6. CONCLUSION

This paper has demonstrated that under a locality of interaction assumption, a property that is exploited in models such as ND-POMDPs, the optimal value functions are additively weakly separable and linear functions. This special structure can be utilized in the context of a recent method for transforming Dec-POMDPs into continuous-state MDPs, which has shown significant scalability gains over previous Dec-POMDP methods. This problem structure allows us to introduce a novel representation of lower and upper bounds of the optimal value functions. This representation has two properties: first, it preserves convergence to an optimal solution; but even more importantly, it significantly reduces the memory requirement of standard representations, thereby increasing scalability. With this representation as background, we extended the state-of-the-art algorithm for solving Dec-POMDPs as continuous-state MDPs to optimally solve ND-POMDPs. The resulting algorithm is the first exact algorithm for ND-POMDPs that can solve problems with up to fifteen agents. In the future, we plan to explore applying the additive weak separability and linearity property to general factored Dec-POMDPs. Furthermore, the scalability with respect to the number of agents of our algorithm is encouraging, and we will pursue additional improvements to also scale up with respect to the planning horizon.

7. ACKNOWLEDGEMENTS

We thank Akshat Kumar for providing his software. Research supported in part by AFOSR MURI project #FA9550-09-1-0538.

8. REFERENCES

[1] C. Amato, J. S. Dibangoye, and S. Zilberstein. Incremental policy generation for finite-horizon DEC-POMDPs. In *ICAPS*, 2009.

[2] R. Aras and A. Dutech. An investigation into mathematical programming for finite horizon decentralized POMDPs. *JAIR*, 37:329–396, 2010.

[3] R. Becker, S. Zilberstein, V. R. Lesser, and C. V. Goldman. Solving transition independent decentralized Markov decision processes. *JAIR*, 22:423–455, 2004.

[4] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein. Policy iteration for decentralized control of Markov decision processes. *JAIR*, 34:89–132, 2009.

[5] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Math. Oper. Res.*, 27(4), 2002.

[6] A. Boularias and B. Chaib-draa. Exact dynamic programming for decentralized POMDPs with lossless policy compression. In *ICAPS*, pages 20–27, 2008.

[7] C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121(1-2):49–107, 2000.

[8] S. de Givry, F. Heras, M. Zytnicki, and J. Larrosa. Existential arc consistency: Getting closer to full arc consistency in weighted CSPs. In *IJCAI*, pages 84–89, 2005.

[9] R. Dechter. Bucket elimination: a unifying framework for processing hard and soft constraints. *Constraints*, 2(1):51–55, 1997.

[10] R. Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1-2):41–85, 1999.

[11] J. S. Dibangoye, C. Amato, O. Buffet, and F. Charpillet. Optimally solving Dec-POMDPs as continuous-state MDPs. In *IJCAI*, 2013.

[12] J. S. Dibangoye, C. Amato, and A. Doniec. Scaling up decentralized MDPs through heuristic search. In *UAI*, pages 217–226, 2012.

[13] J. S. Dibangoye, C. Amato, A. Doniec, and F. Charpillet. Producing efficient error-bounded solutions for transition independent decentralized MDPs. In *AAMAS*, 2013.

[14] J. S. Dibangoye, A.-I. Mouaddib, and B. Chaib-draa. Point-based incremental pruning heuristic for solving finite-horizon DEC-POMDPs. In *AAMAS (1)*, pages 569–576, 2009.

[15] J. S. Dibangoye, G. Shani, B. Chaib-Draa, and A.-I. Mouaddib. Topological order planner for POMDPs. In *IJCAI*, pages 1684–1689, 2009.

[16] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In *NIPS*, pages 1523–1530, 2001.

[17] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *J. Artif. Intell. Res. (JAIR)*, 19:399–468, 2003.

[18] D. Koller and R. Parr. Computing factored value functions for policies in structured MDPs. In *IJCAI*, pages 1332–1339, 1999.

[19] A. Kumar and S. Zilberstein. Constraint-based dynamic programming for decentralized POMDPs with structured interactions. In *AAMAS*, pages 561–568, 2009.

[20] A. Kumar and S. Zilberstein. Point-based backup for decentralized POMDPs: complexity and new algorithms. In *AAMAS*, pages 1315–1322, 2010.

[21] A. Kumar, S. Zilberstein, and M. Toussaint. Scalable multiagent planning using probabilistic inference. In *IJCAI*, pages 2140–2146, 2011.

[22] B. Kveton, M. Hauskrecht, and C. Guestrin. Solving factored MDPs with hybrid state and action variables. *J. Artif. Intell. Res. (JAIR)*, 27:153–201, 2006.

[23] J. Marecki, T. Gupta, P. Varakantham, M. Tambe, and M. Yokoo. Not all agents are equal: scaling up distributed POMDPs for agent networks. In *AAMAS (1)*, pages 485–492, 2008.

[24] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *AAAI*, pages 133–139, 2005.

[25] F. A. Oliehoek. Sufficient plan-time statistics for decentralized POMDPs. In *IJCAI*, 2013.

[26] F. A. Oliehoek, M. T. J. Spaan, C. Amato, and S. Whiteson. Incremental clustering and expansion for faster optimal planning in Dec-POMDPs. *JAIR*, 46:449–509, 2013.

[27] F. A. Oliehoek, S. J. Witwicki, and L. P. Kaelbling. Influence-based abstraction for multiagent systems. In *AAAI*, 2012.

[28] R. Patrascu, P. Poupart, D. Schuurmans, C. Boutilier, and C. Guestrin. Greedy linear value-approximation for factored Markov decision processes. In *AAAI/IAAI*, pages 285–291, 2002.

[29] M. Petrik and S. Zilberstein. A bilinear programming approach for multiagent planning. *JAIR*, 35:235–274, 2009.

[30] M. L. Puterman. *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. Wiley-Interscience, Hoboken, New Jersey, 1994.

[31] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov decision processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[32] T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proc. of UAI*, pages 520–527, 2004.

[33] D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. In *UAI*, pages 568–576, 2005.

[34] P. Varakantham, J. Marecki, M. Tambe, and M. Yokoo. Letting loose a SPIDER on a network of POMDPs: Generating quality guaranteed policies. In *AAMAS*, 2007.

[35] S. J. Witwicki and E. H. Durfee. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *ICAPS*, pages 185–192, 2010.