Influence-Based Policy Abstraction for Weakly-Coupled Dec-POMDPs

Stefan J. Witwicki and Edmund H. Durfee
Computer Science and Engineering
University of Michigan
{witwicki,durfee}@umich.edu

Abstract

Decentralized POMDPs are powerful theoretical models for coordinating agents' decisions in uncertain environments, but the generally-intractable complexity of optimal joint policy construction presents a significant obstacle in applying Dec-POMDPs to problems where many agents face many policy choices. Here, we argue that when most agent choices are independent of other agents' choices, much of this complexity can be avoided: instead of coordinating full policies, agents need only coordinate policy abstractions that explicitly convey the essential interaction influences. To this end, we develop a novel framework for influence-based policy abstraction for weakly-coupled transition-dependent Dec-POMDP problems that subsumes several existing approaches. In addition to formally characterizing the space of transition-dependent influences, we provide a method for computing optimal and approximately-optimal joint policies. We present an initial empirical analysis, over problems with commonly-studied flavors of transition-dependent influences, that demonstrates the potential computational benefits of influence-based abstraction over state-of-the-art optimal policy search methods.

Introduction

Agent team coordination in partially-observable, uncertain environments is a problem of increasing interest to the research community. The decentralized partially-observable Markov decision process (Dec-POMDP) provides an elegant theoretical model for representing a rich space of agent behaviors, observability restrictions, interaction capabilities, and team objectives. Unfortunately, its applicability and effectiveness in solving problems of significant size has been substantially limited by its generally-intractable complexity (Goldman and Zilberstein 2004). This is largely due to the policy space explosion that comes with each agent having to consider the possible observations and actions of its peers, on top of its own observations and actions.

To combat this complexity, researchers have sought tractable Dec-POMDP subclasses wherein agents are limited in their interactions. For instance, there has been significant effort in developing efficient, scalable solution methods for Transition-Independent DEC-MDPs (Becker, Zilberstein, and Lesser 2004) and Network-Distributed POMDPs (Nair et al. 2005; Varakantham et al. 2007), where agents can only influence one another through their reward functions. Because the agents' transitions and observations remain independent, the complexity of this subclass is immune to the growth of the general Dec-POMDP class. Some work has been done in exploring subclasses where agents can influence each others' transitions. However, these either introduce additional restrictions on individual agent behavior (Beynier and Mouaddib 2005; Marecki and Tambe 2009), yield no guarantees of optimality (Varakantham et al. 2009), or have only been shown effective for teams of two or three agents executing for a handful of time steps (Becker, Zilberstein, and Lesser 2004; Oliehoek et al. 2008).

This paper presents an alternative approach to planning for teams of agents with transition-dependent influences. To address the issue of policy space complexity head-on, we contribute a formal framework for policy abstraction that subsumes two existing approaches (Becker, Zilberstein, and Lesser 2004; Witwicki and Durfee 2009). The primary intuition of our work is that by planning joint behavior using abstractions of policies rather than the policies themselves, weakly-coupled agents can form a compact *influence space* over which to reason more efficiently.

We begin by framing the problem as a class of Transition-Decoupled POMDPs (TD-POMDPs) with an expressive, yet natural, representation of agents with rich behaviors whose interactions are limited. Moreover, TD-POMDPs lead us to a systematic analysis of the influences agents can exert on one another, culminating in a succinct model that accommodates both exact and approximate representations of interagent influence. To take advantage of these beneficial traits, we contribute a general-purpose influence-space search algorithm that, based on initial empirical evidence, demonstrably advances the state-of-the-art in exploiting weakly-coupled structure and scaling transition-dependent problems to larger teams of agents without sacrificing optimality.

Coordination of Weakly-coupled Agents

We focus on the problem of planning for agents who are nearly independent of one another, but whose limited, structured dependencies require coordination to maximize their collective rewards. Domains for which such systems have been proposed include the coordination of military field units (Witwicki and Durfee 2007), disaster response systems (Varakantham et al. 2009), and Mars rover exploration (Mostafa and Lesser 2009). Here, we introduce a class of Dec-POMDPs called Transition-Decoupled POMDPs (TD-POMDPs) that, while still remaining quite general, provides a natural representation of the weakly-coupled structure present in these kinds of domains.

Autonomous Planetary Exploration

As a concrete example, consider the team of agents pictured in Figure 1A whose purpose is to explore the surface of a distant planet. There are rovers that move on the ground collecting and analyzing soil samples, and orbiting satellites that (through the use of cameras and specialized hardware) perform various imaging, topography, and atmospheric analysis activities. In representing agents' activities, we borrow from the TAEMS language specification (Decker 1996), assigning to each abstract task a *window* of feasible execution times and a set of possible *outcomes*, each with an associated *duration* and *quality* value. For example, the satellite agent in Figure 1A has a path-planning task that may take 2 hours and succeed with probability 0.8 or may fail (achieving quality 0) with probability 0.2 (such as when its images are too blurry to plan a rover path). Surface conditions limit the rover's visit to site A to occur between the hours of 2 and 8. Additional constraints and dependencies exist among each individual agent's tasks (denoted by lines and arrows).

Although each agent has a different view of the environment and different capabilities (as indicated by their local model bubbles), it is through their limited, structured interactions that they are able to successfully explore the planet. For instance, the outcome of the satellite's path-planning task influences the probabilistic outcome of the rover's site-visiting task. Navigating on its own, the rover's trip will take 6 hours, but with the help of the satellite agent, its trip will take only 3 hours (with 0.9 probability). In order to maximize productivity (quantified as the sum of outcome qualities achieved over the course of execution), agents should carefully plan (in advance) policies that coordinate their execution of interdependent activities. Though simplistic, this example gives a flavor of the sorts of planning problems that fit into our TD-POMDP framework.

Transition-Decoupled POMDP Model

The problem from Figure 1 can be modeled using the finite-horizon Dec-POMDP, which we now briefly review. Formally, this decision-theoretic model is described by the tuple $\langle S, A, P, R, \Omega, O\rangle$, where S is a finite set of world states (which model all features relevant to all agents' decisions), with distinguished initial state s^0. $A = \times_{1\leq i\leq n} A_i$ is the joint action space, each component of which refers to the set of actions of an agent in the system. The transition function $P(s'|s,a)$ specifies the probability distribution over next states given that joint action $a = \langle a_1, a_2, \ldots, a_n\rangle \in A$ is taken in state $s \in S$. The reward function $R(s,a,s')$ expresses the immediate value of taking joint action $a \in A$ in state $s \in S$ and arriving in state $s' \in S$; the aim is to maximize the expected cumulative reward from time steps 1 to T (the horizon). The observation function $O(o|a,s')$ maps joint actions and resulting states to probabilities of joint observations, drawn from finite set $\Omega = \times_{1\leq i\leq n}\Omega_i$. We denote the observation history for agent i as $\vec{o}_i^{\,t} = \langle o_i^1, \ldots, o_i^t\rangle \in \Omega_i^t$, the set of observations i experienced from time step 1 to $t \leq T$. A solution to the Dec-POMDP comes in the form of a joint policy $\bar{\pi} = \langle \pi_1, \ldots, \pi_n\rangle$, where each component π_i (agent i's local policy) maps agent i's observation history $\vec{o}_i^{\,t}$ to an action a_i, thereby providing a decision rule for any sequence of observations that each agent might encounter.

Though the general class of Dec-POMDPs accommodates arbitrary interactions between agents through the transition and reward functions, our example problem contains structure that translates to the following useful properties. First, the world state is *factored* into state features $s = \langle a, b, c, d, \ldots\rangle$, each of which represent a different aspect of the environment. In particular, different features are relevant to different agents. Whereas a rover agent may be concerned with the composition of the soil sample it has just collected, this is not relevant to the satellite agent. As with other related models (e.g. those discussed at the end of this section), we assume a particular grouping of world state features into local features that make up an agent's *local state* s_i. We introduce a further decomposition of local state (that is unique to the TD-POMDP class) whereby agent i's local state s_i is comprised of three disjoint feature sets: $s_i = \langle \bar{u}_i, \bar{l}_i, \bar{n}_i\rangle$, whose components are as follows.

- *uncontrollable features* $\bar{u}_i = \langle u_{i1}, u_{i2}, ...\rangle$ are those features that are not controllable (Goldman and Zilberstein 2004) by any agent, but may be observable by multiple agents. Examples include *time-of-day* or *temperature*.
- *locally-controlled features* $\bar{l}_i = \langle l_{i1}, l_{i2}, ...\rangle$ are those features whose values may be altered through the actions of agent i, but are not (directly) altered through the actions of any other agent; a rover's position, for instance.
- *nonlocal*(*ly-controlled*) *features* $\bar{n}_i = \langle n_{i1}, ...\rangle$ are those features that are each controlled by some other agent but whose values directly impact i's local transitions (Eq. 1).

With this factoring, division of world state features into agents' local states is not strict. *Uncontrollable features* may be part of more than one agent's state. And each *nonlocal feature* in agent i's local state appears as a *locally-controlled feature* in the local state of exactly one other agent. In the example (Figure 1), the rover models whether or not the satellite agent has planned a path for it, so *path-A-planned* would be a nonlocal feature in the rover's local state.

The reward function R is decomposed into into local reward functions, each dependent on local state and local action: $R(s,a,s') = F(R_1(s_1, a_1, s_1'), ..., R_n(s_n, a_n, s_n'))$. The joint reward composition function $F()$ has the property that increases in component values do not correspond to decreases in joint value: $r_i > r_i' \rightarrow F(r_1, ..., r_{i-1}, r_i, r_{i+1}, ..., r_n) \geq F(r_1, ..., r_{i-1}, r_i', r_{i+1}, ..., r_n)\ \forall r_1, ..., r_{i-1}, r_{i+1}, ..., r_n$. In the example problem, local rewards are the qualities attained from the tasks that the agents execute, which combine by summation to yield the joint reward by which joint policies are evaluated.

The observation function is similarly factored $O(o|a,s') = \prod_{1\leq i\leq n} O_i(o_i|a_i, s_i')$, allowing agents direct (partial) ob-

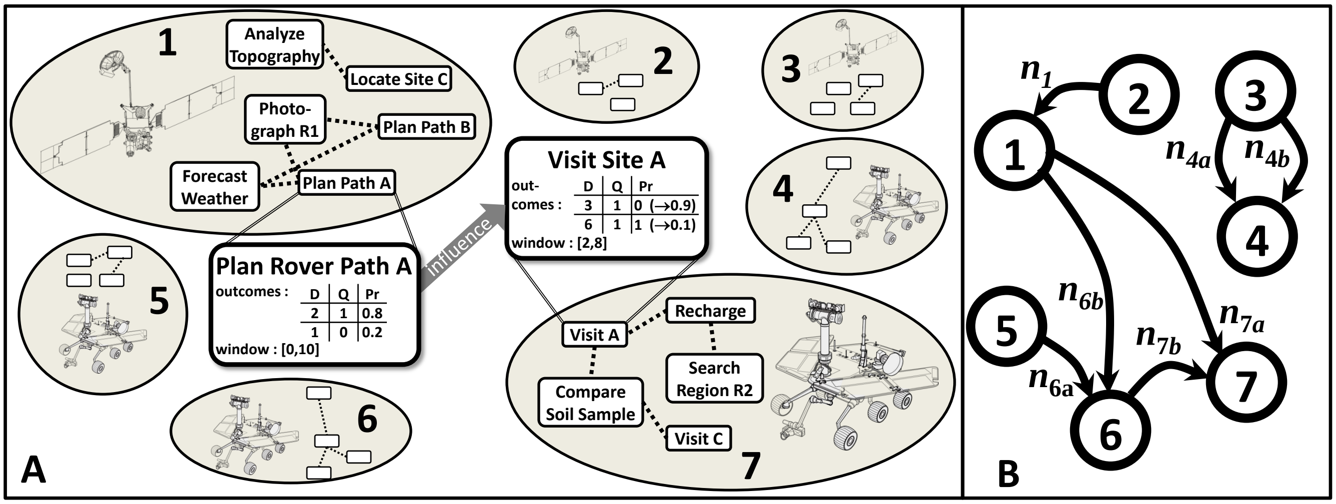


Figure 1: Autonomous Exploration Example.

servations of their local state features but not of features outside their local states. Note, however, that this does not imply observation independence (whereas other models (Becker et al. 2004; Nair et al. 2005) do) because of the shared uncontrollable and nonlocal state features. Likewise, this model is *transition-dependent*, as the values of nonlocal features controlled by agent j may influence the probabilistic outcomes of agent i's actions. Formally, agent i's local transition function, describing probability of next local state $s_i^{t+1} = \langle \bar{u}_i^{t+1}, \bar{l}_i^{t+1}, \bar{n}_i^{t+1} \rangle$ given that joint action a is taken in world state s^t, is the product of three independent terms:

$$Pr\left(s_i^{t+1}|s^t,a\right) = Pr\left(\bar{u}_i^{t+1}|\bar{u}_i^t\right) \cdot Pr\left(\bar{l}_i^{t+1}|\bar{l}_i^t, \bar{n}_i^t, \bar{u}_i^t, a_i\right) \cdot Pr\left(\bar{n}_i^{t+1}|s^t - \bar{l}_i^t, a_{\neq i}\right) \quad (1)$$

The result of this factorization is a structured transition dependence whereby agents alter the effects of each others' actions sequentially but not concurrently. Agent i may set the value of one of agent j's nonlocal state features and agent j's subsequent transitions are influenced by the new value.

The structure we have identified is significant because it decouples the Dec-POMDP model into a set of weakly-coupled local POMDP models that are tied to one another by their transition influences. Without the existence of *nonlocal features*, an agent cannot influence another's observations, transitions, or rewards, and the agents' POMDPs become completely independent decision problems. With an increasing presence of nonlocal features, the agents subproblems become more and more strongly coupled. We can concisely describe the coupling and locality of interaction in a TD-POMDP problem with an *interaction digraph* (Figure 1B), which represents each instance of a nonlocal feature with an arc between agent nodes. As pictured, the interaction digraph for our example problem contains an arc from agent 1 (the satellite) to agent 7 (the rover) labeled n_{7a} that refers to the nonlocal feature *path-A-planned*.

Although the TD-POMDP is less general than the Dec-POMDP (and the factored Dec-POMDP (Oliehoek et al. 2008)), it is more general than prior transition-dependent Dec-POMDP subclasses (Becker, Zilberstein, and Lesser 2004; Beynier and Mouaddib 2005). Beynier's (2005) OC-DEC-MDP assumes fixed execution ordering over agent tasks and dependencies in the form of task precedence relationships. Becker's (2004) Event-Driven DEC-MDP is more closely related, but it assumes local full observability, and restricts transition dependencies to take the form of mutually exclusive events which could trivially be mapped to nonlocal features in the TD-POMDP model. The TD-POMDP is also more general than the DPCL (Varakantham et al. 2009) in its representation of observation (since local observations can depend on other agents' actions), but less general in its representation of interaction (since agents cannot affect each others' local transitions concurrently). Generality aside, we contend that the structure that we have defined provides a very natural representation of interaction, making it straightforward to map problems into TD-POMDPs. Further, as we shall see, TD-POMDP structure leads us to a broad characterization of transition-dependent influences and a systematic methodology for abstracting those influences.

Decoupled Solution Methodology

To take advantage of the TD-POMDP's weakly-coupled interaction structure, we build upon a general solution methodology that decouples the joint policy formulation. Central to this approach is the use of local models, whereby each agent can separately compute its individual policy. As derived by Nair (2003), any Dec-POMDP can be transformed into a single-agent POMDP for agent i assuming that the policies of its peers have been fixed. This *best-response* model is prohibitive to solve in the general case (given that the agent must reason about the possible observations of the other agents), but in various restricted contexts, iterative best-response algorithms have been devised which provide substantial computational leverage (Becker et al. 2004; Nair et al. 2005). As we describe later on, the TD-POMDP (which is composed of weakly-coupled local POMDPs) can be decoupled into fully-independent POMDPs that have been augmented with compact models of influence.

Given this decoupling scheme, planning the joint policy becomes a search through the space of combinations of optimal local policies (each found by solving a local best-response model). This approach is taken in much of the literature to solve transition-independent reward-dependent models (e.g TI-DEC-MDPs (Becker et al. 2004), ND-POMDPs (Nair et al. 2005; Varakantham et al. 2007)). And while some approaches have solved transition-dependent models in this way, the results have been either limited to just two agents (Becker, Zilberstein, and Lesser 2004), or to approximately-optimal solutions without formal guarantees (Varakantham et al. 2009; Witwicki and Durfee 2007). In the remainder of this paper, we present and evaluate a formal framework that subsumes previous transition-dependent methods and produces provably optimal solutions, focusing on abstraction to make the search tractable and scalable.

Influence-Based Policy Abstraction

The Dec-POMDP joint policy space (which is exponential in the number of observations and doubly exponential in the number of agents and the time horizon) grows intractably large very quickly. The primary intuition behind how our approach confronts this intractability is that, by abstracting weakly-coupled interaction influences from local policies, an influence space emerges that is more efficient to explore than the joint policy space. We begin by discussing policy abstraction in the context of a simple, concrete example with some very restrictive assumptions. Over the course of this section, we gradually build up a less restrictive language through which agents can convey their abstract influences, culminating in a formal characterization of the general space of interaction influences for the class of TD-POMDPs.

Figure 2 portrays an interaction wherein one rover (R5) must prepare a site before another rover (R6) can benefit from visiting the site. Assume that apart from this interaction, the two agents' problems are completely independent. Neither of them interact with any other agents, nor do they share any observations except for the occurrence of site C's preparation and the current time. In a TD-POMDP, this simple interaction corresponds to the assignment of a single boolean nonlocal feature *site-C-prepared* that is locally-controlled by R5, but that influences (and is nonlocal to) R6. Thus, in planning its own actions, R6 needs to be able to make predictions about *site-C-prepared*'s value (influenced by R5) over the course of execution.

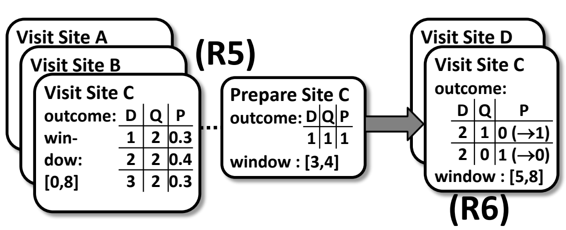


Figure 2: Example of highly-constrained influence.

Definition 1. *For a TD-POMDP interaction x represented by agent j as nonlocal feature n_{jx}, which is controllable by agent i and affects the transitions of agent j, we define the **influence** of i's policy π_i on n_{jx}, denoted $\Gamma_{\pi_i}(n_{jx}) = Pr\,(n_{jx}|\ldots)$, to be a sufficient summary of π_i for agent j to model expected changes to n_{jx} and to plan optimal decisions given that agent i adopts π_i.*

By representing the influence of R5's policy with a distribution $Pr($*site-C-prepared*$|...)$ as in Definition 1, R6 can construct a transition model for nonlocal feature *site-C-prepared*. The last multiplicand of Equation 1 suggests that this construction requires computing a transition probability for every value of the (nonlocal subsection of) world state ($s^t - \bar{l}_i^t$), nonlocal action ($a_{\neq i}$), and next nonlocal feature value. However, in this particular problem, R6 does not need a complete distribution that is conditioned on all features. In fact, the only features that R6 can use to predict the value of *site-C-prepared* are $time$ and *site-C-prepared* itself. Although *site-C-prepared* is dependent on other features from R5's local state, R6 cannot observe any evidence of these features except through its (perhaps partial) observations of *site-C-prepared* and $time$. Thus, all other features can be marginalized out of the distribution $Pr($*site-C-prepared*$|...)$.

In this particular example, the only influence information that is relevant to R6 is the probability with which *site-C-prepared* will become $true$ conditioned on $time = 4$. At the start of execution, *site-C-prepared* will take on value $false$ and remain $false$ until R5 completes its "Prepare Site C" task (constrained to finish only at time 4, if at all, given the task window in Figure 2). After the site is prepared, the feature will remain $true$ thereafter until the end of execution. With these constraints, there is no uncertainty about when *site-C-prepared* will become $true$, but only if it will become $true$. Hence, the influence of R5's policy can be summarized with just a single probability value, $Pr($*site-C-prepared* $= true|time = 4)$, from which R6 can infer all transition probabilities of *site-C-prepared*.

Reasoning about concise influence distributions instead of full policies can be advantageous in the search for optimal joint policies. The **influence space** is the domain of possible assignments of the influence distribution, each of which is achieved by some feasible policy. In our simple example, this corresponds to the feasible values of $Pr($*site-C-prepared* $= true|time = 4)$. As shown in Figure 2, R5 has several sites it can visit, each with uncertain durations. In general, different policies that it adopts may achieve different interaction probabilities. However, due to the constraints in Figure 2, many of R5's policies will map to the same influence value. For instance, any two policies that differ only in the decisions made after time 3 will yield the same value for $Pr($*site-C-prepared* $= true|time = 4)$. For this example, the influence space is strictly smaller than the policy space. Thus, by considering only the feasible influence values, agents avoid joint reasoning about the multitude of local policies with equivalent influences.

A Categorization of Influences

The influence in the example from Figure 2 has a very simple structure due to the highly-constrained transitions of the nonlocal feature. By removing constraints, we can more generally categorize the influence between R5 and R6. Let the window of execution of "Prepare Site C" be unconstrained: $[0, 8]$. With this change, there is the possibility of R5 preparing site C at any time during execution. The consequence is that a single probability is no longer sufficient to characterize R5's influence. Instead of representing a single probability value, R6 needs to represent a probability for each time *site-C-prepared* could be set to true. In other words, this influence is dependent on a feature of the agents' state: $time$.

Definition 2. *An influence $\Gamma_{\pi_i}(n_{jx})$ is **state-dependent** w.r.t. feature f if its summarizing distribution must be conditioned on the value of f: $Pr\left(n_{jx}|f, ...\right)$.*

As we have seen in prior work (Witwicki and Durfee 2009), the set of probabilities $Pr(\textit{site-C-prepared}|time)$ is an abstraction of R5's policy that accommodates temporal uncertainty of the interaction.

Generalizing further, the probability of an interaction may differ based on both present and past values of state features. For instance, in the example from Figure 1A, if it is cloudy in the morning, this might prohibit the satellite from taking pictures, and consequently lower the probability that it plans a path for the rover in the afternoon. So by monitoring the history of the weather, the rover could anticipate the lower likelihood of help from the satellite, and might change some decisions accordingly. Becker employs this sort of abstraction in his Event-driven DEC-MDP solution algorithm, where he relates probabilities of events to dependency histories (Becker, Zilberstein, and Lesser 2004).

Definition 3. *Influence $\Gamma_{\pi_i}(n_{jx})$ is **history-dependent** w.r.t. feature f if its summarizing distribution must be conditioned on the history of f: $Pr\left(n_{jx}^{t+1}|\vec{f}^{\,t}, ...\right)$.*

Moreover, there may also be dependence between influences. For instance, agent 4 has two arcs coming in from agent 3 (in Figure 1B), indicating that agent 3 is exerting two influences, such as if agent 3 could plan two different paths for agent 4. In the case that agent 3's time spent planning one path leaves too little time to plan the other path, the nonlocal features n_{4a} and n_{4b} are highly correlated, requiring that their joint distribution be represented.

Definition 4. *Influence $\Gamma_{\pi_i}(n_x)$ and influence $\Gamma_{\pi_i}(n_y)$ are **influence-dependent** (on each other) if their summarizing distributions are correlated, requiring $Pr\left(n_x, n_y|...\right)$.*

A Comprehensive Influence Model

With the preceding terminology, we have systematically though informally introduced an increasingly comprehensive characterization of transition influences. A given TD-POMDP influence might be state-dependent and history-dependent on multiple features, or even dependent on the history of another influence. Furthermore, there may be chains of influence-dependent influences. In Figure 1B, for example, agent 7 models two nonlocal features, one (n_{7a}) influenced by agent 1 and the other (n_{7b}) influenced by agent 6. The additional arc between agents 1 and 6 forms an undirected cycle that implies a possible dependence between n_{7a} and n_{7b} by way of n_{6b}. The only way to ensure a complete influence model is to incorporate all three influences into a joint distribution.

In general, for any team of TD-POMDP agents, their influences altogether constitute a Dynamic Bayesian Network (DBN) whose variables consist of the nonlocal features as well as their respective dependent state features and dependent history features with links corresponding to the dependence relationships. This *influence DBN* encodes the probability distributions of all of the outside influences affecting each agent. Once all of an agent's incoming influences (exerted by its peers) have been decided, the agent can incorporate this probability information into a local POMDP model with which to compute optimal decisions. The agent constructs the local POMDP by combining the TD-POMDP local transition function (terms 1 and 2 of Equation 1) with the probabilities of nonlocally-controlled feature transitions $Pr\left(\bar{n}_j^{t+1}|...\right)$ encoded (as conditional probabilities) by the influence DBN. Rewards and observations for this local POMDP are dictated by the TD-POMDP local reward function R_i and local observation function O_i, respectively.

As agents' interactions become more complicated, more variables are needed to encode their effects. However, due to TD-POMDP structure, the DBN need contain only those critical variables that link the agents' POMDPs together.

Proposition 1. *For any given TD-POMDP, the influence $\Gamma_{\pi_i}(n_{jx})$ of agent i's fixed policy π_i on agent j's nonlocal feature n_{jx} need only be conditioned on histories (denoted $\vec{m}_j$) of mutually-modeled features $\bar{m}_j = \bigcup_{k \neq j} (s_j \cap s_k)$.*

Proof Sketch. The proof of this proposition emerges from the derivation of a *belief-state* (see Nair et al. 2003) representation for TD-POMDP agent j's best-response POMDP.

$$b_j^t = \left\langle Pr\left(s_j^t, \vec{m}_j^{t-1}|\vec{a}_j^{\,t-1}, \vec{o}_j^{\,t}\right), \forall s_j^t, \vec{m}_j^{t-1}\right\rangle \quad (2)$$

We can derive an equation for the components (each indexed by one value of $\left\langle s_j^{t+1}, \vec{m}_j^t\right\rangle$) of j's belief-state at time $t+1$ by applications of Bayes' rule, conditional probability, and the factored TD-POMDP local observation function $O_i()$:

$$b_j^{t+1}\left(s_j^{t+1}, \vec{m}_j^t\right) = Pr\left(s_j^{t+1}, \vec{m}_j^t|\vec{a}_j^{\,t}, \vec{o}_j^{\,t+1}\right)$$
$$= \frac{O_j\left(o_j^{t+1}|a_j^t, s_j^{t+1}\right)\sum_{s_j^t} Pr\left(s_j^{t+1}|s_j^t, \vec{m}_j^{t-1}, \vec{a}_j^{\,t}, \vec{o}_j^{\,t}\right) b_j^t\left(s_j^t, \vec{m}_j^{t-1}\right)}{Pr\left(o_j^{t+1}|\vec{a}_j^{\,t-1}, \vec{o}_j^{\,t}, a_j^t\right) \text{ : a normalizing constant}} \quad (3)$$

Next, from conditional independence relationships implied by the factored transitions (Equation 1) of the TD-POMDP:

$$= \frac{O_j(\ldots)\sum_{s_j^t - \bar{m}_j^t} Pr\left(\bar{l}_j^{t+1}|s_j^t, a_j^t\right) Pr\left(\bar{u}_j^{t+1}|s_j^t\right) Pr\left(\bar{n}_j^{t+1}|\vec{m}_j^t\right) b_j^t(\ldots)}{Pr\left(o_j^{t+1}|\vec{a}_j^{\,t-1}, \vec{o}_j^{\,t}, a_j^t\right) \text{ : a normalizing constant}} \quad (4)$$

Equation 4 has three important consequences. First, agent j can compute its next belief state using only its peers' policies, the TD-POMDP model, its previous belief state, and its latest action-observation pair (without having to remember the entire history of observations). Second, the denominator of Equation 4 (which is simply a summation of the numerators across all belief-state components) allows the agent to compute the probability of its next observation (given its current action) using only its peers' policies, the TD-POMDP model, and its previous belief state. These two consequences by themselves prove sufficiency of the belief state representation for optimal decision-making. Third, the only term in the numerator of Equation 4 that depends upon peers' fixed policies is $Pr\left(\bar{n}_j^{t+1}|\vec{m}_j^t\right)$, and hence this distribution is a sufficient summary of all peers' policies. □

Corollary 1. *The influence DBN grows with the number of shared state features irrespective of the number of local state features and irrespective of the number of agents.*

The implication of Proposition 1 is that the local POMDP can be compactly augmented with histories of only those state features that are shared among agents. Moreover, the complexity with which an agent models a peer is controlled by its tightness of coupling and *not* by the complexity of the peer's behavior. Efficiency and compactness of local models is significant because they will be solved repeatedly over the course of a distributed policy-space search.

Another way to interpret this result is to relate it to the relative complexity of the influence space, which is the number of possible influence DBNs. Each DBN is effectively a HMM whose state is made up of shared features (and histories of shared features) of the TD-POMDP world state. Given Proposition 1's restrictions on feature inclusion, the space of DBNs should scale more gracefully than the joint policy space with the number of world features and number of agents (under the assumption that agents remain weakly-coupled), a claim that is supported by our empirical results.

Searching the Influence Space

Given the compact representations of influence that we have developed, agents can generate the optimal joint policy by searching through the space of influences and computing optimal local policies with respect to each. Drawing inspiration from Nair's (2005) GOA method for searching through the policy space, here we describe a general algorithm for searching the (TD-POMDP) influence space.

Algorithm 1 outlines the skeleton of a depth-first search that enumerates all feasible values, one influence at a time, as it descends from root to leaf. At the root of the search tree, influences are considered that are independent of all of other influences. And at lower depths, feasible influence values are determined by incorporating any higher-up influence values on which they depend. This property is ensured given any total ordering of agents (denoted $ordering$ in Algorithm 1) that maintains the partial order of the acyclic interaction digraph.[1] At each node of the depth-first search, procedure OIS() is called on agent i, who invokes the next agent's OIS() execution and later returns its result to the previous agent.

[1] In the event of a cyclic interaction digraph, we can still ensure this property, but with modifications to Algorithm 1. Note that although the digraph may contain cycles, the influence DBN itself cannot contain cycles (due to the non-concurrency of agent influences described following equation 1). We can therefore separate an agent's time-indexed influence variables into those {*dependent upon*, *independent of*} another influence, and reason about those sets at separate levels of the search tree. If we separate influence variables sufficiently, cyclic dependence can be avoided as we progress down the search tree.

Algorithm 1 Optimal Influence-Space Search

OIS($i, ordering, DBN, vals$)
1: $POMDP_i \leftarrow$ BUILDBESTRESPONSEMODEL(DBN)
2: **if** $i =$ LASTAGENT($ordering$) **then**
3: $\quad\langle vals[i], \pi_i\rangle \leftarrow$ EVALUATE($POMDP_i$)
4: $\quad$**return** $\langle vals, DBN\rangle$
5: **end if**
6: $j \leftarrow$ NEXTAGENT($i, ordering$)
7: $I \leftarrow$ GENERATEFEASIBLEINFLUENCES($POMDP_i$)
8: $bestVal \leftarrow -\infty$
9: $bestDBN \leftarrow nil$
10: **for each** $influence_i \in I$ **do**
11: $\quad thisVals \leftarrow$ COPY($vals$)
12: $\quad\langle thisVals[i], \pi_i\rangle \leftarrow$ EVALUATE($POMDP_i, influence_i$)
13: $\quad DBN_i \leftarrow$ COMBINE($DBN, influence_i$)
14: $\quad\langle thisVals, DBN_{child}\rangle \leftarrow$ OIS($j, ordering, DBN_i, thisVals$)
15: $\quad jointVal \leftarrow$ COMPOSEJOINTREWARD($thisVals$)
16: $\quad$**if** $jointVal > bestVal$ **then**
17: $\quad\quad vals \leftarrow thisVal$
18: $\quad\quad bestDBN \leftarrow DBN_{child}$
19: $\quad$**end if**
20: **end for**
21: **return** $\langle vals, bestDBN\rangle$

The algorithm is decentralized, but is initiated by a root agent whose influence does not depend on its peers.

The search begins with the call OIS($root, ordering, \emptyset, \infty$), prompting the first agent to build its (independent) local POMDP (line 1) and to generate all of the feasible combinations of its outgoing influence values (line 7), each in the form of a DBN (as described in the previous section). A naïve implementation of GENERATEFEASIBLEINFLUENCES() would simply enumerate all local policies, and for each, compute the requisite conditional probabilities that the policy implies and incorporate them into a DBN model. At the end of this section, we suggest a more sophisticated generation scheme. The root creates a branch for each feasible influence DBN, passing down the influence along with the value of the best local policy that achieves the DBN's influences (computed using EVALUATE()). Each such call to OIS() prompts the next agent to construct a local POMDP in response to the root's influence, compute its feasible influences and values, and pass those on to the next agent.

At the root of the tree, the DBN starts out as empty and gradually grows as it travels down the tree, each iteration accumulating another agent's fixed influences. The agent at the leaf level of the tree does not influence others, so simply computes a best response to all of the fixed influences and passes up its policy value (lines 1-4). Local utility values get passed down and washed back up so that intermediate agents can evaluate them via COMBINE() (which composes expected local utilities into expected joint utilities). In this manner, the best outgoing influence values get chosen at each level of the tree and returned to the root. When the search completes, the result is an optimal influence-space point: a DBN that encodes the feasible influence settings that optimally coordinate the team of agents. As a post-processing step, the optimal joint policy is formed by computing all agents' best-response policies (via BUILDBESTRESPONSEMODEL()

and EVALUATE()) in response to the optimal influence point returned by the search.

Approximation Techniques. An attractive trait of this framework is the natural accommodation of approximation methods that comes with representing influences as probability distributions. One straightforward technique is to discretize the DBN space, grouping probability values that are within ϵ of each other so as to guarantee that a distribution is found whose influences are close to that of the optimal influence. A second technique is to approximate the structure of the influence DBN. For a given influence, feature selection methods could be used to remove all but the most useful influence dependencies, thereby sacrificing completeness of the abstraction for a reduction in search space.

Efficient Generation and Evaluation of Feasible State-Dependent Influences. A commonly-studied subclass of TD-POMDPs involves state-dependent, history-independent influences whereby (in particular) agents coordinate the timings of interdependent task executions (Becker, Zilberstein, and Lesser 2004; Beynier and Mouaddib 2005; Witwicki and Durfee 2009). To reason about these influences, we can utilize a constrained policy formulation technique that is based upon the dual-form Linear Program (LP) for solving Markov decision processes (Witwicki and Durfee 2007). Under the assumption of a non-recurrent state space, the dual form represents the probabilities of reaching states as *occupancy measure* variables, which are exactly what we need to represent the influences that an agent exerts. For instance, $Pr\left(\textit{site-C-prepared}|time = 4\right)$ corresponds to the probability of R5 (in Figure 2) entering any state for which $\textit{site-C-prepared} = true$ and $time = 4$, which is the summation of LP occupancy measures associated with these states (or belief-states, for POMDPs). The agent can use this LP method to (1) calculate its outgoing influence given any policy, (2) determine whether a given influence is feasible, and if so (3) compute the optimal local policy that is constrained to exert that influence.

We devise a useful extension: an LP that *finds* relevant influence points. For a given influence parameter p, we would like the agent to find all feasible values for that parameter (achievable by any deterministic policy). This can be accomplished by solving a series of (MI)LPs, each of which looks for a (pure) policy that constrains the parameter value to lie within some interval: $p_{\min} < p < p_{\max}$ (starting with interval $[0, 1]$). If the LP returns a solution, the agent has simultaneously found a new influence ($p = p_0$) and computed a policy that exerts that influence, subsequently uncovering two new intervals $\{(p_{\min}, p_0), (p_0, p_{\max})\}$ to explore. If the LP returns "no solution" for a particular interval, there is no feasible influence within that range. By divide and conquer, the agent can find all influences or stop the search once a desired resolution has been reached (by discarding intervals smaller than ϵ). In general, this method allows agents to generate all of their feasible influences without exhaustively enumerating and evaluating all of their policies.

Empirical Results

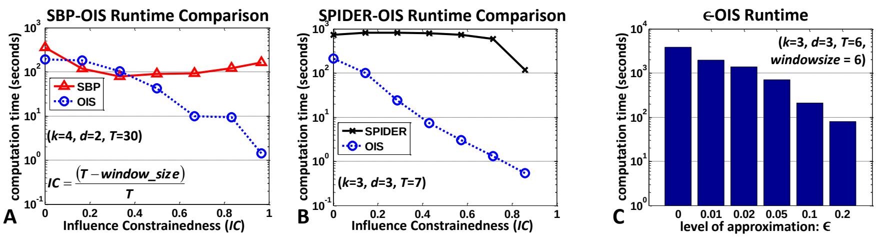
We present an initial empirical study analyzing the computational efficiency of our framework. Results marked "OIS" correspond to our implementation of Algorithm 1 that follows the LP-based influence generation approach (discussed previously). We compare influence-space search to two state-of-the-art optimal policy search methods: (1) a Separable Bilinear Programming ("SBP") algorithm (Mostafa and Lesser 2009) for problems of the same nature as ED-DEC-MDPs (Becker, Zilberstein, and Lesser 2004) and (2) an implementation of "SPIDER" (Varakantham et al. 2007) designed to find optimal policies for two-agent problems with transition dependencies (Marecki and Tambe 2009). Both implementations were graciously supplied by their respective authors to improve the fairness of comparison.

Plots 3A and 3B evaluate the claim that influence-space search can exploit weak coupling to find optimal solutions more efficiently than policy-space search. These two plots compare OIS with SBP and SPIDER, respectively, on sets of 25 randomly-generated 2-agent problems from the planetary exploration domain, each of which contains a single interaction whereby a task of a satellite agent influences the outcome of a task of a rover agent.[2] For each problem, the *influence constrainedness* was varied by systematically decreasing the window size (from T to 1) of the influencing task. While the computation time[3] (plotted on a logarithmic scale) taken by SBP and SPIDER to generate optimal solutions remains relatively flat, OIS becomes significantly faster as influences are increasingly constrained. This result, although preliminary, demonstrates that influence-based abstraction can take great advantage of weak agent coupling but might prove less valuable in tightly-coupled problems.

The third experiment (shown in Figure 3C-D) evaluates OIS on a set of 10 larger problems (where SBP and SPIDER were infeasible), each with 4 agents connected by a chain of influences. One of agent 1's tasks (chosen at random) influences one of agent 2's tasks, and one of agent 2's tasks influences one of agent 3's tasks, etc. We compare optimal OIS with "ϵ-OIS", which discretizes probabilities with a step size of ϵ in the probability space. The quality and runtime figures indicate that, for this space of problems, influence-space approximation can achieve substantial computational savings at the expense of very little solution quality. Additionally, this result is notable for demonstrating tractability of *optimal* joint policy formulation on a size of problems (4 agents, 6 time units) that has been beyond the reach of the prior approaches to solving transition-dependent problems (with relatively unrestricted local POMDP structure).

[2] As denoted in Figure 3, the agents each have k tasks, each with d randomly-selected durations (with duration probabilities generated uniformly at random) and randomly-selected outcome qualities executed for a horizon of T time units. Because the implementations of SBP and SPIDER were tailored to specific domains, we could not run them on the same problems. For instance, the SBP implementation assumes that agents are not able to wait between task executions. Both domains assume partial observability such that agents can directly observe all of their individually-controlled tasks, but not the outcomes of the tasks that influence them.

[3] All computation was performed on a single shared CPU.



D

ϵ-OIS Solution Quality / Runtime

epsilon	0	0.01	0.02	0.05	0.1	0.2
normalized joint utility value V	1.000	1.000	0.998	0.991	0.990	0.922
stddev(V)	0.000	0.000	0.005	0.020	0.020	0.117
improvement over uncoordinated local optimization	22.7 %	22.7 %	22.5 %	21.7 %	21.6 %	14.6 %
runtime	3832	1985	1369	702.2	208.9	79.80
stddev(runtime)	4440	2502	1995	1013	207.7	60.40

Figure 3: Empirical Evaluation of Influence-space Search.

Conclusions

This paper contributes a formal framework that characterizes a broad array of weakly-coupled agents' influences and abstracts them from the agents' policies. Although previous methods have abstracted specialized flavors of transition influence (Becker, Zilberstein, and Lesser 2004; Witwicki and Durfee 2009) or used abstraction to guide heuristic search (Witwicki and Durfee 2007), the comprehensive model we have devised places these conceptually-related approaches into a unified perspective. As a foundation for our framework, we have introduced a TD-POMDP class whose factored transition structure engenders a decoupling of agents' subproblems and a compact model of nonlocal influence. Inspired by the successful scaling of the (transition-independent, reward-dependent) ND-POMDP model (Nair et al. 2005; Varakantham et al. 2007) to teams of many agents, we have cast the TD-POMDP joint policy formulation problem as one of local best-response search.

Prior to this work, there have been few results shown in scaling transition-dependent problems to teams of several agents whilst maintaining optimality. Our compactness result suggests that, for weakly-coupled transition-dependent problems, agents can gain traction by reasoning in an abstract influence space instead of a joint policy space. We give evidence supporting this claim in our initial empirical results, where we have demonstrated superior efficiency of optimal joint policy generation through an influence-space search method on random instances of a class of commonly-studied weakly-coupled problems. But more importantly, our general influence-based framework offers the building blocks for more advanced algorithms, and a promising direction for researchers seeking to apply Dec-POMDPs to teams of many weakly-coupled transition-dependent agents. Future work includes a more comprehensive investigation into problem characteristics (e.g. digraph topology and influence type) that impact the performance of influence-space search, and further development and comparison of approximate flavors of OIS with other approximate approaches (such as TREMOR (Varakantham et al. 2009)).

Acknowledgements

This material is based upon work supported, in part, by AFOSR under Contract No. FA9550-07-1-0262. We thank Hala Mostafa and Janusz Marecki for supplying us with the SBP and SPIDER implementations, respectively, and the anonymous reviewers for their thoughtful comments.

References

Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. 2004. Solving transition independent decentralized Markov Decision Processes. *JAIR* 22:423–455.

Becker, R.; Zilberstein, S.; and Lesser, V. 2004. Decentralized Markov decision processes with event-driven interactions. In *AAMAS-04*, 302–309.

Beynier, A., and Mouaddib, A. 2005. A polynomial algorithm for decentralized Markov decision processes with temporal constraints. In *AAMAS-05*, 963–969.

Decker, K. 1996. TAEMS: A framework for environment centered analysis & design of coordination mechanisms. In *Foundations of Distr. Artif. Intelligence, Ch. 16*, 429–448.

Goldman, C., and Zilberstein, S. 2004. Decentralized control of cooperative systems: Categorization and complexity analysis. *JAIR* 22:143–174.

Marecki, J., and Tambe, M. 2009. Planning with continuous resources for agent teams. In *AAMAS-09*, 1089–1096.

Mostafa, H., and Lesser, V. 2009. Offline planning for communication by exploiting structured interactions in decentralized MDPs. In *Proceedings of IAT-09*, 193–200.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D. V.; and Marsella, S. 2003. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI-03*, 705–711.

Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed POMDPs: A synthesis of distributed cnstrnt. optim. and POMDPs. *AAAI-05* 133–139.

Oliehoek, F.; Spaan, M.; Whiteson, S.; and Vlassis, N. 2008. Exploiting locality of interaction in factored dec-pomdps. In *AAMAS-08*, 517–524.

Varakantham, P.; Marecki, J.; Yabu, Y.; Tambe, M.; and Yokoo, M. 2007. Letting loose a spider on a network of POMDPs: generating quality guaranteed policies. In *AAMAS-07*, 817–824.

Varakantham, P.; Kwak, J.; Taylor, M.; Marecki, J.; Scerri, P.; and Tambe, M. 2009. Exploiting coordination locales in distributed POMDPs via social model shaping. *ICAPS-09*.

Witwicki, S., and Durfee, E. 2007. Commitment-driven distributed joint policy search. In *AAMAS-07*, 480–487.

Witwicki, S., and Durfee, E. 2009. Flexible approximation of structured interactions in decentralized Markov decision processes. In *AAMAS-09*, 1251–1252.