Heuristic Search of Multiagent Influence Space

Frans A. Oliehoek[1], Stefan Witwicki[2], and Leslie P. Kaelbling[1]

[1] CSAIL, MIT, {fao,lpk}@csail.mit.edu
[2] INESC-ID and Instituto Superior Técnico, UTL, Portugal, witwicki@inesc-id.pt

Abstract. Two techniques have substantially advanced efficiency and scalability of multiagent planning. First, heuristic search gains traction by pruning large portions of the joint policy space. Second, influence-based abstraction reformulates the search space of joint policies into a smaller space of influences, which represent the probabilistic effects that agents' policies may exert on one another. These techniques have been used independently, but never together, to solve solve larger problems (for Dec-POMDPs and subclasses) than was previously possible. In this paper, we combine multiagent A* search and influence-based abstraction into a single algorithm. This enables an initial investigation into whether the two techniques bring complementary gains. Our results indicate that A* can provide significant computational savings on top of those already afforded by influence-space search, thereby bringing a significant contribution to the field of multiagent planning under uncertainty.

1 Introduction

Computing good policies for agents that are part of a team is an important topic in multiagent systems. This task, planning, is especially challenging under uncertainty, e.g., when actions may have unintended effects and each agent in the team may have a different view of the global state of the environment due to its private observations. In recent years, researchers have proposed to gain grip on the problem by abstracting away from *policies* of other agents and instead reasoning about the effects, or *influences*, of those policies [2, 1, 22, 26, 24]. However, no methods have been proposed to effectively search the space of influences other than enumeration. In this paper, we fill this void by showing how it is possible to perform heuristic search of the influence space, thereby significantly speeding up influence-based planning.

The problem of multiagent planning under uncertainty can be formalized as a decentralized partially observable Markov decision process (Dec-POMDP) [3]. However, its solution is provably intractable (NEXP-complete). As such, many methods either focus on finding approximate solutions without quality guarantees [11, 19, 14], or providing optimal solutions for restricted sub-classes [12, 23, 22, 24]. In particular, more efficient procedures have been developed for models that exhibit *transition and observation independence* [2, 12] or *reward independence* [1]. Unfortunately, these sub-classes are too restrictive for many interesting tasks, such as agents collaborating in the search for a target.

Recently, the *transition-decoupled* POMDP (TD-POMDP) [26] has been introduced as a model that allows for transition, observation, and reward dependence, while still allowing for more efficient solutions than the general Dec-POMDP model.[3] The core idea is to exploit independence between agents by formalizing the *influence* they can exert on each other. This allows us to search the space of joint influences, rather than the space of joint policies, which can lead to significant savings since the former can be much smaller for many problems (cf. [25] chapter 4).

A difficulty in this approach, however, is that it is not clear how to efficiently search this space; although the influence space is often much smaller the policy space, it may still grow exponentially with the problem size. Previous work has performed exhaustive search of the joint influence space. On the other hand, in general Dec-POMDPs, A* search guided by heuristics, i.e., multiagent A* (MAA*), has been shown to be an extremely powerful method for finding optimal solutions [21, 15, 20]. The main contribution of this paper, is to show how heuristics can be defined to more efficiently search the influence space.

To accomplish this, we make the following auxiliary contributions: we show how one can define heuristics in influence space, we prove the admissibility of such heuristics—thus guaranteeing optimality of A* search—and we provide the results of an empirical evaluation that shows that our proposed methods can yield significant performance increases, especially on problems that are hard for exhaustive influence search. Additionally, we demonstrate how TD-POMDPs can be used for an important class of problems: locating objects or targets with a team of agents, which also leads us to the first application of influence search on problems that have cyclic dependencies between the agents.

2 Influence-Based Abstraction

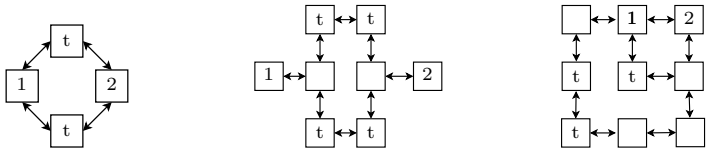
Here we provide background on the TD-POMDP model. We also review the concept of influence-based policy abstraction, and explain how this abstraction can be exploited to find optimal solutions via optimal influence space search. We begin with a motivating application.

2.1 Motivating Domain: Locating Targets

Although the TD-POMDP model and the methods presented in this paper extend to other settings, in this paper we focus on their application to problems where a team of agents has to locate a target. We assume that a prior probability distribution over its location is available. Also, the target is assumed to be either stationary or to move in a manner that does not depend on the strategy used by the searching agents.

More concretely, we consider a problem domain called HouseSearch in which a team of robots must find a target in a house with multiple rooms.

[3] Some other recent models that allow for limited amounts of both types of dependence are treated in Section 5.



(a) *Diamond.* (b) *Rectangle.* (c) *Squares.*

Fig. 1: HOUSESEARCH environments. '1' and '2' mark the start positions of the search robots. 't' marks possible target locations.

Such an environment can be represented by a graph, as illustrated in Fig. 1. At every time-step an agent i can stay in the current node n or move to a neighboring node n'. The location of an agent i is denoted l_i and that of the target is denoted l_{target}. The movements, or actions a_i, of each agent i have a specific cost $c_i(l_i,a_i)$ (e.g., the energy consumed by navigating to a next room) and can fail; we allow for stochastic transitions $p(l'_i|l_i,a_i)$. Also, each robot receives a penalty c_{time} for every time step that the target is not caught yet. When a robot is in the same node n as the target, there is a probability of detecting the target $p(detect_i|l_{target},l_i)$, which will be modeled by a state variable 'target found by agent i'. When the target is detected, the agents receive a reward r_{detect}. Additionally, i automatically communicates to all other agents that it found the target. Given the prior distribution and model of target behavior, the goal is to optimize the sum of rewards, thus trading off movement cost and probability of detecting the target as soon as possible.

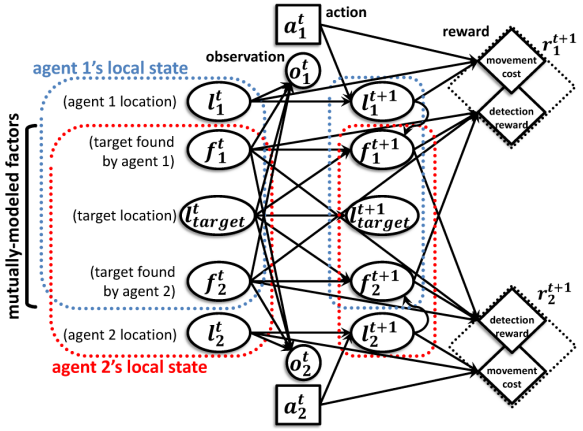
2.2 TD-POMDP Model

Here we formalize the planning task for scenarios such as the HOUSESEARCH task described above. First, we introduce the (single-agent) factored POMDP, a common model for single-agent planning under uncertainty. Then we describe how a TD-POMDP extends this model to multiple agents.

A *factored partially observable Markov decision process* for a single agent i is a tuple $\langle \mathcal{S}_i, \mathcal{A}_i, T_i, R_i, \mathcal{O}_i, O_i \rangle$, where $\mathcal{S}_i = X_1 \times \cdots \times X_k$ is the set of states s_i induced by a set of k state variables or *factors*, $\mathcal{A}_i$ is the set of actions that the agent can take, T_i is the transition model that specifies $\Pr(s'_i|s_i,a_i)$, $R_i(s_i,a_i,s'_i)$ is the reward function, $\mathcal{O}_i$ is the set of observations o_i, and O_i is the observation function that specifies $\Pr(o_i|a_i,s'_i)$. Because the state space is factored, it is usually possible to specify T_i, R_i and O_i in a compact manner using a Bayesian network called a two-stage temporal Bayesian network (2TBN) [4]. Given this model, the planning task for a POMDP is to find an optimal policy π_i that maximizes the expected sum of rewards over h time steps or stages, where h specifies the horizon. Such a policy maps from *beliefs*, probability distributions over states, to actions. While solving a POMDP is an intractable problem, in the last two decades many exact and approximate solution methods have been proposed (see, e.g., [8]).

affected	NMF	MMF
locally	x_i^l	m_i^l
nonlocally	N/A	m_i^n
unaffectable	x_i^u	m_i^u

(a) Different types of state factors in the state representation for agent i.



(b) A TD-POMDP for HouseSearch.

Fig. 2: TD-POMDP State components (left) and illustration (right).

Intuitively, a TD-POMDP is a *set* of factored POMDPs, one for each agent, where there is overlap in the state factors of each agent. Of course, for such a setting to be well-defined means that different transition models T_i need to be consistent since the local transition probabilities can depend on other agents too. (Alternatively one can think of there being a single transition model T for all agents.) In a TD-POMDP, the set of state factors can be divided into factors that occur only in one agent's local state space ('non-mutual' factors, or NMFs) and factors that are 'mutually modeled' by more than one agent (MMFs). The joint reward function for the TD-POMDP is the sum of the individual rewards of the agents: $R(s,a,s') = \sum_i R_i(s_i,a_i,s_i')$. Additionally, we assume that there is a known initial state distribution $\boldsymbol{b}^0$. A TD-POMDP imposes the restriction that each state factor can be *directly* affected by the action of at most one agent. That is, each factor can have an incoming edge from only 1 action variable, thus no two agents can directly affect the same factor. This does not mean that state factors depend on just one agent, since factors can be indirectly (i.e., via a directed path consisting of multiple edges) influenced by many agents. This factorization leads to different parts of an agent's local state, as summarized in Fig. 2a. Using the notation defined in this table, we will write the local state of an agent i as $s_i = \langle x_i^l, x_i^u, m_i^l, m_i^n, m_i^u \rangle = \langle x_{i,}, m_i \rangle$. For a more formal introduction of the TD-POMDP framework, see the second author's dissertation [25].

As in a factored POMDP, the TD-POMDP transition, observation and reward model can be represented by a 2TBN. For instance, Fig. 2b illustrates the 2TBN for HouseSearch. It clearly shows that the local states of both agents are overlapping: both agents model the target location and the factors 'target found by agent 1/2'. Note that the mutually modeled state factors can only be characterized as (non)locally affected from the perspective of a particular agent. E.g., f_1 is locally affected for agent 1, but non-locally affected for agent 2. The figure also shows that, for this particular problem, each agent's reward function

R_i is factored as the sum of two components R_{detect} and R_{move}. The former models the rewards for detection, as well as the time cost (c_{time}) of not detecting. This component depends on f_1^{t+1}, f_2^{t+1} as well as on f_1^t, f_2^t: only when (at least) one of the f_i variables switches from false to true the agents receive the reward; when all four factors are false the agents get the time penalty and otherwise the rewards are 0 (but the movement costs remain). The movement reward component only depends on the agents' non-mutual locations and local action.

The TD-POMDP is a non-trival subclass of the factored Dec-POMDP [16], for which the NEXP-completeness result still holds [25]. This also means that single-agent POMDP solution methods do not directly apply. A first problem is that, in a multiagent context, we are now searching for a joint policy $\pi = \langle \pi_1, \ldots, \pi_n \rangle$. Moreover, when there are multiple agents interacting, the agents can no longer base their policy over simple beliefs over states[4], instead the policies are mappings from *histories* of observations $\vec{o}_i$ to actions.

2.3 Influences and Local Models

A well-known solution method for Dec-POMDPs, JESP [11], searches for a locally optimal joint policy as follows: it starts with a random joint policy and then selects one agent to improve its policy while keeping the other policies fixed. The improvement of the selected agent is done by computing a best response. From the perspective of a single agent i, by fixing π_{-i} (the policies of the other agents) the problem can be re-cast as an augmented POMDP, where the augmented state is a tuple $\langle s, \vec{o}_{-i} \rangle$ of a nominal state and the observation histories of the other agents.

JESP directly applies to TD-POMDPs. However, because of the special structure a TD-POMDP imposes, we can compute the best response in more efficient way: rather than maintaining a JESP belief $b_i(s, \vec{o}_{-i})$, agent i can maintain a condensed belief $b_i(x_i^t, \vec{m}_i^t)$ over just its own private factors and the history of mutually modeled factors [25, 26]. Intuitively, this is possible, because all information about $\vec{o}_{-i}$ and the state factors that are not in agent i's local state (i.e., x_j for $j \neq i$) is captured by $\vec{m}_i^t$. For instance, Fig. 2b illustrates that all information agent 2 has about l_1^t is inferred from the history of f_1^t and l_{target}^t.

A second important observation is that an agent i is only influenced by other agents via its nonlocal mutually modeled factors m_i^n. E.g., in Fig. 2b agent 1 only influences agent 2 through changes to factor f_1. Therefore, if, during planning, the value of this factor at all stages is known, agent 2 can completely forget about agent 1 and just solve its local POMDP (and similar for agent 1). This line of reasoning holds even if agent 2 does not know the exact values of f_1 ahead of time, but instead knows the probability that f_1 turns to true for each stage. This insight lies at the basis of *influence-based policy abstraction*: all policy profiles π_{-i} that lead to the same distributions over non-local MMFs $m_i^{n,0}, \ldots, m_i^{n,h-1}$

[4] In order to predict the action of a teammate as well as possible, an agent has to maintain a more complex belief over states and future policies of agents [6].

can be clustered together, since they will lead to the same best response of agent i.

This idea is formalized using the notion of incoming influence. An *incoming influence point* of agent i, denoted $I_{\rightarrow i}$, specifies a collection of conditional probability tables (CPTs): one for each nonlocally affected MMF, for each stage $t = 1, \ldots, h-1$.[5] We denote a CPT for f_1^t (from our example) as $p_{f_1^t}$, which specifies probabilities $p_{f_1^t}(v|\cdot)$ for values $v \in \{0,1\}$ of f_1^t given its parents $(\cdot)$. In this example, $I_{\rightarrow 2} = \{p_{f_1^1}, p_{f_1^2}, \ldots, p_{f_1^{h-1}}\}$. To specify these CPTs, it is necessary to only use $\vec{m}_i$, the history of mutual features, as the parents [26]. I.e., the CPTs are specified as $p_{m_i^{n,t+1}}(\cdot|\vec{m}_i^t)$. With some abuse of notation, we also write $\Pr(m_i^{n,t+1}|\vec{m}_i^t, I_{\rightarrow i})$ for the probability of (some value of) a non-local factor $m_i^{n,t+1}$ according to $I_{\rightarrow i}$. Because the CPTs can only depend on $\vec{m}_i$, an incoming influence point $I_{\rightarrow i}$ enables the computation of a best response π_i independent of the other agents.

Of course, in general the actions of agent i can also influence other agents, so in order to find optimal solutions, we will also need to reason about this influence. We denote by $I_{i\rightarrow}$ the *outgoing influence point* of agent i, which specifies a collection of CPTs: one for each of its locally affected MMFs. Again, these CPTs can depend on only (the history of) MMFs $\vec{m}_i$. An incoming and outgoing influence point together form a (complete) *influence point* $I_i = \langle I_{\rightarrow i}, I_{i\rightarrow} \rangle$. A *joint influence point* $I = \langle I_{1\rightarrow}, \ldots, I_{n\rightarrow} \rangle$ specifies an outgoing influence point for each agent. Note that I also specifies the incoming influences, since every incoming influence point is specified by the outgoing influence points of the other agents. Fig. 3a illustrates the dependencies of an influence point in a so-called *influence DBN*. For instance, the possible CPTs $p_{f_1^{t+1}}$ are conditioned on $\vec{l}_{target}^t$, the history of the target location, as well f_1^t, the value of 'target found by agent 1' at the previous stage.

Given I_i, agent i has an augmented local POMDP with local states, rewards and transitions. In this local model, a state is a pair $\langle x_i^t, \vec{m}_i^t \rangle$ such that, as discussed above, a belief is of the form $b_i(x_i^t, \vec{m}_i^t)$. Given an incoming influence point that dictates the transition probabilities of its nonlocally-affected MMFs, this local POMDP is independent of the other agents, but subject to constraints: the solution should adhere to the specified outgoing influence point. We call such a restricted model together with the influence point an *influence augmented local model (IALM)*. Solving the IALM is non-trivial since standard POMDP solvers will not respect the additional constraints. The problem can be solved by reformulating it as a mixed integer linear program (MILP) [25].

2.4 Optimal Influence Search

The key property of influences is that each influence point can compactly represent potentially many policies. Moreover, the number of influence points is finite, since we need to consider at most one for each deterministic joint policy.[6] There-

[5] For $t = 0$ the distribution is specified by the initial state distribution b^0.

[6] A deterministic solution is guaranteed to exist.

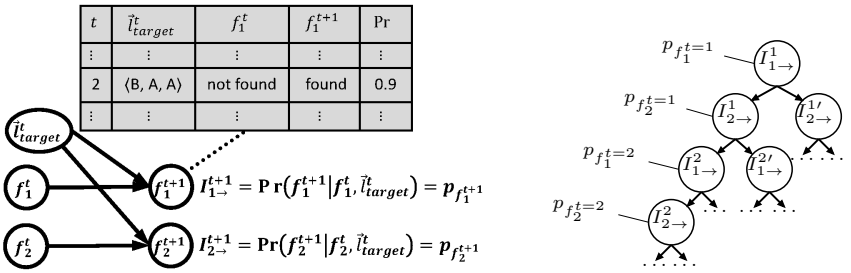


Fig. 3: The influence DBN and search tree for the example of Figure 2b.

fore, rather than searching in the larger space of joint policies, we can search in the space of joint influence points and evaluate each of them by computing the agents' best responses. In particular, the value of a fully specified joint influence point is

$$V(I) = \sum_{i=1}^{n} V_i(I) \tag{1}$$

where $V_i(I) = V_i(\langle I_{\rightarrow i}, I_{i\rightarrow}\rangle)$ is the value of agent i's best response to $I_{\rightarrow i}$ subject to the constraints of satisfying $I_{i\rightarrow}$, i.e., the value that results from solving its IALM.

Given that we can compute the value of a joint influence point I, we can optimally solve a TD-POMDP by enumerating all I. Optimal Influence Search (OIS) [26] does this by constructing a tree, as illustrated in Fig. 3b. An outgoing influence *slice* $I^t_{i\rightarrow}$ is that part of agent i's outgoing influence point corresponding to a particular stage t. The search tree contains the outgoing influence slices for all agents for stage $t = 1$ on the first n levels, it contains the slices for $t = 2$ on the next n levels, etc. An influence point is defined by a complete path from root to leaf. OIS performs an exhaustive depth-first search to find the optimal joint influence point from which the optimal joint policy can be reconstructed.

Although an apparently simple search strategy, OIS's influence abstraction has led to impressive gains in efficiency. It has established itself as the state of the art in computing optimal solutions for weakly-coupled agents agents, demonstrating improvements over several other approaches that also exploit interaction structure [26].

3 Heuristic Influence Search

OIS can greatly improve over other methods by searching in the space of joint influences, which can be much smaller than the space of joint policies. However, its weakness is that it needs to search this space exhaustively. In contrast, for general Dec-POMDPs, heuristic search methods (in particular A*, see, e.g., [18]) have shown to be very effective [20]. The main idea here, therefore, is to extend heuristic search to be able to search over the joint influence space.

In the subsections that follow, we develop the mechanics necessary to compute admissible heuristic values for nodes of the influence search tree. As we describe, this is a non-trivial extension, due to the fact that an influence summarizes a set of possible policies.

3.1 Computing Heuristic Values

In order guarantee that heuristic search finds the optimal solution we need an *admissible* heuristic; i.e, a function F mapping nodes to heuristic values that are guaranteed to be an over-estimation of the value of the best path from root to leaf that passes through that node. In our setting this means that the heuristic $F(\check{I})$ for a partially specified joint influence point $\check{I}$ (corresponding to a path from the root of the tree to a non-leaf node) should satisfy

$$F(\check{I}) \geq \max_{I \text{ consistent with } \check{I}} V(I). \tag{2}$$

We will also write $I^{*|\check{I}}$ for the maximizing argument of the r.h.s. of (2).

In Dec-POMDPs, it is possible to perform A* search over partially specified joint policies [15]. For a 'past joint policy' $\boldsymbol{\varphi} = (\pi^0, \ldots, \pi^{t-1})$ that specifies the joint policy for the first t stages, it is possible to define $F(\boldsymbol{\varphi}) = G(\boldsymbol{\varphi}) + H(\boldsymbol{\varphi})$, where G gives the actual expected reward over the first t stages $0, \ldots, (t-1)$ and where H is a heuristic of the value achievable for the remaining stages. There are multiple ways to define H. For instance, one general form [21] is:

$$H(\boldsymbol{\varphi}) = \sum_s \Pr(s|\boldsymbol{b}^0, \boldsymbol{\varphi}) H^t(s), \tag{3}$$

where $H^t(s)$ is a guaranteed overestimation of the expected value starting from s in stage t. Such an overestimation can be obtained, for instance, by solving the underlying MDP (called Q_{MDP}) or POMDP [5, 15] .

Unfortunately, it is not possible to adapt the above approach to searching influence space in a straightforward fashion. Given an $\check{I}$, the past joint policy is not fixed, because $\pi^{*|\check{I}}$ the best joint policy for $I^{*|\check{I}}$ is unknown. Therefore, we take a somewhat different approach, as detailed next.

3.2 Restricted Scope Restricted Horizon Heuristic

We exploit the fact that $V(I)$ in (1) can be additively decomposed. That is we upper bound (1) by

$$F(\check{I}) = \sum_{i=1}^{n} F_i(\check{I}) \tag{4}$$

Clearly, when $F_i(\check{I}) \geq V_i(I^{*|\check{I}})$ for all agents i, then $F(\check{I}) \geq V(I^{*|\check{I}})$ and $F(\check{I})$ is admissible.

The problem of computing a heuristic value $F_i(\check{I})$ is illustrated in Figure 4. For now, we assume that $\check{I}$ specifies all the influences for the first $\overline{h}$ stages

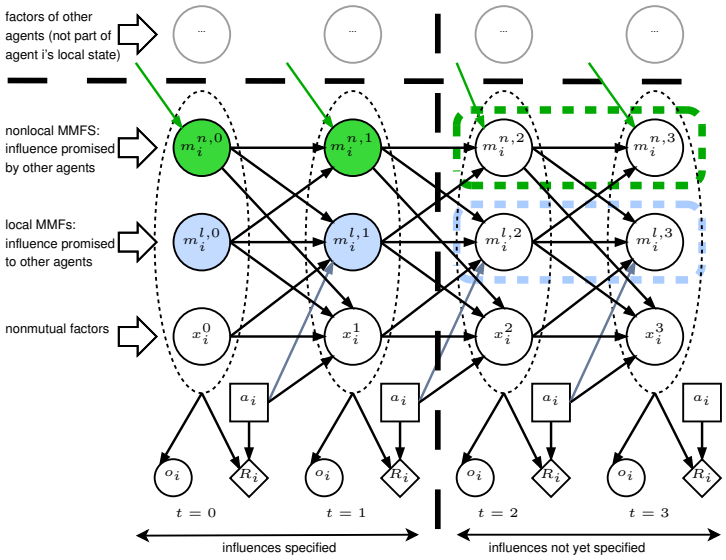


Fig. 4: A partially specified joint influence point $\check{I}$ from the perspective of agent i. Dashed black elipses denote the agent's local state. The figure does not include unaffectable factors. Influences are specified for stage 0,1. Green (dark) nodes are specified incoming influences, blue (light) nodes are specified outgoing influences. The dashed boxes denote the unspecified incoming (dark green) and outgoing (light blue) influences for stages 2,3.

(i.e., up to but *not* including stage $\overline{h}$). The figure, in which $\overline{h} = 2$, shows that computation of $F_i(\check{I})$ depends on only a subset of state factor (i.e., a restricted scope). In order to actually compute the $F_i(\check{I})$, we suggest a 2-step approach: 1) compute an admissible heuristic for the stages for which the influence is not yet specified, and 2) subsequently use these heuristic values to solve a constrained POMDP over horizon $\overline{h}$. We will refer to heuristics of this form as *restricted scope restricted horizon (RSRH)* heuristics.

Step 1: The Unspecified-Influence Stages. The goal here is to compute $H_i^{\overline{h}}$ similar to the term used in (3). This means computing an optimistic estimate of the value over the remaining (unspecified-influence) stages. In particular, we use an approach similar to QMDP: we compute the value of the underlying MDP but restricted to local states of the agent. In order to do so, we make optimistic assumptions on the unspecified incoming influences. Intuitively, this amounts to assuming that an agent i's peers will adopt policies that will exert the most beneficial effect on agent i's local state.

Remember that an IALM state $\langle s_i^t, \vec{m}_i^{t-1} \rangle = \langle x_i^t, \vec{m}_i^t \rangle$, and that we write $x_i = \langle x_i^l, x_i^u \rangle$ and $m_i = \langle m_i^l, m_i^n, m_i^u \rangle$. Now the overestimation we use is[7]

$$H_i^t(x_i, \vec{m}_i) \triangleq \max_{a_i} \Big[R(s_i,a_i) + \sum_{x_i', m_i^{l\prime}, m_i^{u\prime}} \Pr(x_i', m_i^{l\prime}, m_i^{u\prime} | s_i, a_i) \max_{m_i^{n\prime}} H_i^{t+1}(x_i', \vec{m}_i') \Big] \tag{5}$$

It is clear that

$$H_i^t(x_i, \vec{m}_i) \geq \max_{a_i} \Big[R(s_i,a_i) + \sum_{x_i', m_i^{l\prime}, m_i^{u\prime}, m_i^{n\prime}} \Pr(x_i', m_i^{l\prime}, m_i^{u\prime}, m_i^{n\prime} | x_i, \vec{m}_i a_i, I_{\to i}) \\ V_{i,MDP}^{I_{\to i}}(x_i', \vec{m}_i') \Big] = V_{i,MDP}^{I_{\to i}}(x_i, \vec{m}_i), \quad \forall I_{\to i}. \tag{6}$$

Here, $V_{i,MDP}^{I_{\to i}}$ is the value of the underlying restricted-scope MDP given *any* fully specified incoming influence point $I_{\to i}$. Also, it is important to note that $\Pr(x_i', m_i^{l\prime}, m_i^{u\prime} | s_i, a_i)$ in (5) can be directly computed due to the structure imposed by the TD-POMDP. As such, our 'optimistic estimate of QMDP' H_i^t can be computed via dynamic programming starting at the last stage $h-1$ and working back to stage $\overline{h}$.

Step 2: The Specified-Influence Stages. Here we use $H_i^{\overline{h}}$ found in step 1 to construct a restricted-horizon constrained POMDP, i.e., the IALM for agent i for only the first $\overline{h}$ stages, which we will denote by $\overline{M}$ (we denote all quantities of $\overline{M}$ with bars). For this IALM, we change the immediate rewards for the 'last' stage, stage $\overline{h}-1$, to include the heuristic $H_i^{\overline{h}}$ for the remaining stages:

$$\overline{R}^{\overline{h}-1}(x_i, \vec{m}_i, a_i) \triangleq R(s_i,a_i) + \sum_{x_i', m_i^{l\prime}, m_i^{u\prime}} \Pr(x_i', m_i^{l\prime}, m_i^{u\prime} | s_i, a_i) \max_{m_i^{n\prime}} H_i^{\overline{h}}(x_i', \vec{m}_i') \tag{7}$$

That is, we apply the same optimistic estimate, effectively transforming the immediate rewards of stage $\overline{h}-1$ into optimistic heuristic 'action-value' estimates. The result is a completely specified, restricted-horizon, IALM for agent i that can be solved in exactly the same way as the full-horizon IALM. The value it achieves is $F_i(\check{I}) \triangleq \overline{V}_i(\overline{I})$.

Partially Specified Joint Influence Slices. So far we assumed that the (outgoing) influences, for all agents, up to and including stage $\overline{h}-1$ were specified. However, for many nodes in the influence tree in Figure 3b the influences are only specified for a subset of agents at stage $\overline{h}-1$. However, we can easily overcome this problem by adapting the computation of $F_i(\check{I})$ in the following fashion.

If an outgoing influence at stage $\overline{h}-1$ is not specified we just omit the constraint in the MILP. If an incoming influence at stage $\overline{h}-1$ is not specified

[7] In the remainder of the paper we assume rewards of the form $R(s_i,a_i)$ for simplicity, but the extension to $R(s_i,a_i,s_i')$ is straightforward.

we transform the transition probability for the last transition in the restricted-horizon IALM (i.e., the transition from stage $\overline{h}-2$ to $\overline{h}-1$) such that for all $\langle x_i^{l,t}, x_i^{u,t}, m_i^{l,t}, m_i^{u,t} \rangle$ the local state will always transition to the fully specified local state $\langle x_i^{l,t}, x_i^{u,t}, m_i^{l,t}, m_i^{n,t}, m_i^{u,t} \rangle$ with the highest heuristic value.

Theorem 1. *$F_i(\check{I})$ is admissible.*

Proof. See appendix.

The implication is that our heuristic can be used to prune those influence assignments that are guaranteed to be sub-optimal. As such, we will be able to expand potentially far fewer nodes of the influence-space search tree and still guarantee optimality.

4 Experiments

We now present an empirical evaluation of our heuristic influence-space search method. Our primary hypothesis is that exhaustive influence-space search (OIS), which has established itself as the state of the art for computing optimal solutions for weakly-coupled transition-dependent agents, can gain even more traction if combined with heuristic search methods. Although it would be interesting to additionally compare with optimal Dec-POMDP solution methods that employ heuristic search but not influence abstraction (e.g., [20]), we expect that these problems are too large, especially in the number of individual observations ($4 \times 2 \times 2 = 16$ for *Diamond*, 32 for *Rectangle*, and 36 for *Squares*), which are way beyond what optimal Dec-POMDP solvers can handle (the largest of those problems have 5 individual observations). In order to test our hypothesis, we performed experiments both on the HouseSearch configurations shown in Fig. 1 as well as on SatelliteRover, a TD-POMDP test set involving two agents that interact through task dependencies [26].

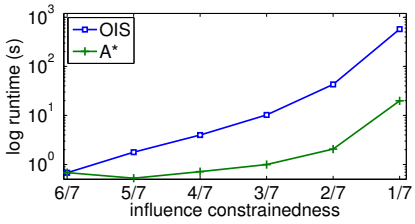
For HouseSearch, we experimented with different degrees of stochasticity. I.e., we considered problems ranging from deterministic actions and deterministic observations ("d.a.d.o.") to stochastic actions (where the probability that a move action will fail is 0.1) and stochastic observations (where the probability of observing no target when in the same room as the target is 0.25) ("labeled s.a.s.o"). For all problems, the parameters were set to $c_{time} = -5$, $c_i = -1$ for each movement action, and $r_{detect} = 0$. Table 1 compares the runtimes of OIS with those of A* using our restricted scope restricted horizon heuristic. As shown, using this simple heuristic can lead to significant speedups over depth-first search, especially on the *Diamond* configuration where we see as much as two orders of magnitude improvement (e.g., at horizon 3 of *Diamond* d.o.s.a) not to mention scaling up of influence-based planning to larger time horizons than was previously possible. For *Rectangle* and *Squares*, however, the benefit of A* over exhaustive OIS are less pronounced. (Given space restrictions, we omit the d.o.d.a., d.o.s.a, and s.o.s.a. variations of these problems, whose trends were the same as in s.o.d.a.)

	Diamond (d.o.d.a)		(s.o.d.a)		(d.o.s.a)		(s.o.s.a)		*Rectangle* (s.o.d.a)		*Squares* (s.o.d.a)	
h	OIS	A*	OIS	A*	OIS	A*	OIS	A*	OIS	A*	OIS	A*
1	0.16	0.15	0.21	0.26	0.20	0.30	0.34	0.37	0.11	0.18	0.21	0.27
2	0.95	0.57	3.65	1.20	6.38	1.92	69.65	5.05	1.04	0.95	1.56	1.65
3	8.65	1.55	423.5	19.39	15,042	96.02		11,124	36.52	29.92	75.59	80.53
4	108.0	7.54		881.0					5,629	3,346		13,769
5	1,403	45.29										

Table 1: Runtime results for different variations of HouseSearch problems.

We also tested A* on SatelliteRover, in which the lengths of task execution windows were systematically varied to affect the level of *influence constrainedness* [26]. The less constrained the agents' interactions, the larger the influence space, as demonstrated by the exponentially increasing runtimes plotted on a logarithmic scale in Fig. 5a. Evidently, it is on these less-constrained problems, which are hardest for OIS, where we get the most speedup from A*. Here, A* search leads to significant savings of well over an order of magnitude (573s vs. 19.9s for IC=1/7), thereby complementing the savings achieved by influence-based abstraction.

The differences between the impact of A* in *Diamond*, *Rectangle*, and *Squares* warrant a more detailed analysis. In the latter two variations, the tighter heuristic appears too loose to effectively guide heuristic search except on problems with longer time horizons. Upon closer inspection, we discovered an inherent bias in the application of our heuristic to HouseSearch problems; it encourages the 'stay' action. This is because the heuristic evaluation of each agent makes the optimistic assumption that the other agent will find the target, in which case the agent need not look itself and incur the associated movement cost. To verify this hypothesis we performed some additional experiments where movements do not have a cost associated with them. The results are shown in Fig. 5b and clearly confirm the hypothesis. A* now shows substantial speedups on *Rectangle* and on *Squares*, and even more impressive speedups on *Diamond*.



(a)

	Diamond		*Rectangle*		*Squares*	
h	OIS	A*	OIS	A*	OIS	A*
1	0.22	0.25	0.13	0.16	0.22	0.26
2	3.49	0.59	0.79	1.24	2.24	2.24
3	349.6	3.52	27.34	25.48	68.70	45.10
4		40.68	5,434	275.8		656.3
5		2,163				

(b)

Fig. 5: Results for SatelliteRover (a), and Runtimes for the s.o.d.a. variations of HouseSearch without movement costs (b).

5 Related Work

Recently, some other methods and models that allow for both transition and reward dependence have been proposed. As mentioned, MAA* for Dec-POMDPs has seen various improvements [15, 20, 21]. The big contrast with that work, is that here we perform search in the more compact space of influences.

A related model is the EDI-CR [10] that makes explicit a set of joint transition and reward dependencies. The authors propose an MILP-based solution method that is conceptually related to influence abstraction; it clusters action-observation histories that have equivalent probabilistic effects so as to reduce the number of joint histories considered. A significant difference is that, unlike the algorithms we develop here, instead, it entails solving a single joint model framed as an MILP. In contrast, our methodology employs more compact local models augmented with influence information and decoupled from the joint model.

Another related model, the DPCL [22, 24], uses 'coordination locales' (CLs) to facilitate the computation of response policies using local models that incorporate the effect of other agents' policies. Here, coordination locales isolate the structured dependencies among agents analogously to the way that TD-POMDP mutually-modeled features enable compact specification of influences. However, in contrast to our work, the DPCL has only ever been solved approximately, with no guarantees on solution quality.

Finally, there have also been a number of papers on computing approximate solutions for factored Dec-POMDPs, which are more general than TD-POMDPs in that they do not impose restrictions on the number of agents that can directly affect a state factor. Also, they allow for an arbitrary (e.g., not agent-wise) decomposition of the joint reward function. For a finite horizon, one can try to exploit the factorization using collaborative graphical Bayesian games, which can in turn be approximated using approximate inference techniques [13]. Similarly, for the infinite-horizon, EM has been proposed to exploit factorization while optimizing policies represented as finite state controllers [9, 17]. Again, the biggest difference with the work described here is that these methods search in policy space, rather than the space of influences.

6 Conclusions & Future Work

We have introduced heuristic A* search of the influence space for the optimal solution of multiagent planning problems formalized as TD-POMDPs. As previous work has shown, the space of influences can be much smaller than the space of joint policies and therefore searching the former can lead to significant improvements in performance. We illustrated the efficacy of our approach on sets of problems from two different domains including HouseSearch, wherein we showed the first application of optimal influence search on TD-POMDPs with cyclic dependencies between agents. Our empirical evaluation shows that A* search of the influence space can lead to significant improvements in performance over plain depth-first OIS. In particular, the results indicate that in

problems that are harder (i.e., where there is a high number of possible influences) A* leads to the most improvements. In other words, influence abstraction and heuristic search can provide complementary gains. This suggests that A* search of influence space can be an important tool in scaling up a large class of multiagent planning problems under uncertainty.

There are a number of directions for future research. First, the results indicate that in some cases only a moderate amount of pruning is realized. As such, an important direction of future work is to find tighter heuristics that can still be computed efficiently. Because of the connection this paper establishes between searching influence space and MAA* for Dec-POMDPs, it is natural to try and extend recent improvements in the latter to the former. One question is whether it is possible to incrementally expand the nodes in the search tree. Such incremental expansion has yielded significant increases in performance for Dec-POMDPs [20]. Another interesting question is whether it is possible to cluster influence points. That is, it may be possible to characterize when different joint influence points correspond to best responses that are guaranteed to be the identical.

7 Acknowledgements

This research was supported by AFOSR MURI project #FA9550-09-1-0538, the Portuguese Fundação para a Ciência e Tecnologia, and the the Carnegie Mellon Portugal Program under project CMU-PT/SIA/0023/2009.

Appendix: Proof of Theorem 1

Proof. We need to show that

$$\forall_{I_i} \quad F_i(\check{I}) = \overline{V}_i(\overline{I}) \geq V_i(I^{*|\check{I}}) \tag{8}$$

We assume an arbitrary $I_{\rightarrow i}$,$I_{i\rightarrow}$ consistent with $\check{I}$. Since the first $\overline{h}-1$ stages are identical, (8) clearly holds if

$$\forall_{b_i} \qquad \overline{Q}_i^{\overline{h}-1}(b_i,a_i) \geq Q_i^{\overline{h}-1,I_{\rightarrow i},I_{i\rightarrow}}(b_i,a_i). \tag{9}$$

We choose an arbitrary b_i. Expanding both sides, we need to show that

$$\overline{R}^{\overline{h}-1}(b_i,a_i) \geq R^{\overline{h}-1}(b_i,a_i) + \sum_{b'} P(b'|b_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(b_i'). \tag{10}$$

Expanding the expectations over IALM states:

$$\sum_{x_i,\vec{m}_i} b_i(x_i,\vec{m}_i)\overline{R}^{\overline{h}-1}(x_i,\vec{m}_i,a_i) \geq \sum_{x_i,\vec{m}_i} b_i(x_i,\vec{m}_i) \Big[R(s_i,a_i) + \sum_{s_i'}\sum_{o_i} \Pr(s_i',o_i|s_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x_i',\vec{m}_i',b_i').\Big]$$

Substituting the definition of $\overline{R}$:

$$\sum_{x_i,\vec{m}_i} b_i(x_i,\vec{m}_i)\Big[R(s_i,a_i) + \sum_{x_i',m_i^{l\prime},m_i^{u\prime}} \Pr(x_i',m_i^{l\prime},m_i^{u\prime}|s_i,a_i) \max_{m_i^{n\prime}} H_i^{\overline{h}}(x_i',\vec{m}_i')\Big]$$
$$\geq \sum_{x_i,\vec{m}_i} b_i(x_i,\vec{m}_i)\Big[R(s_i,a_i) + \sum_{s_i'}\sum_{o_i} \Pr(s_i',o_i|s_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x_i,\vec{m}_i,b_i')\Big].$$

This is proven if we can show that

$$\forall_{x_i,\vec{m}_i} \sum_{x_i',m_i^{l\prime},m_i^{u\prime}} \Pr(x_i',m_i^{l\prime},m_i^{u\prime}|s_i,a_i) \max_{m_i^{n\prime}} H_i^{\overline{h}}(x_i',\vec{m}_i')$$
$$\geq \sum_{s_i'}\sum_{o_i} \Pr(s_i',o_i|s_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x_i',\vec{m}_i',b_i') \quad (11)$$

We assume arbitrary $x_i, \vec{m}_i$ and now continue with the right hand side. Since it is well-known that the MDP value function is an upper bound to the POMDP value function [7], we have

$$\sum_{s_i'}\sum_{o_i} \Pr(s_i',o_i|s_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x_i',\vec{m}_i',b_i') \leq \sum_{s_i'}\sum_{o_i} \Pr(s_i',o_i|s_i,a_i) V_{i,MDP}^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x_i',\vec{m}_i')$$
$$\leq \sum_{s_i'} \Pr(s_i'|s_i,a_i) V_{i,MDP}^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x_i',\vec{m}_i') \leq \sum_{s_i'} \Pr(s_i'|s_i,a_i) V_{i,MDP}^{\overline{h},I_{\rightarrow i}}(x_i',\vec{m}_i')$$

The last term denotes the optimal value under only incoming influences, and the inequality holds because the set of policies available to agent i without restrictions due to promised outgoing influences is a strict superset of those when there are outgoing influences. Now, by (6) we directly get that the last quantity

$$\leq \sum_{s_i'} \Pr(s_i'|s_i,a_i) H_i^{\overline{h}}(x_i',\vec{m}_i') \leq \sum_{x_i',m_i^{l\prime},m_i^{u\prime}} \Pr(x_i',m_i^{l\prime},m_i^{u\prime}|s_i,a_i) \max_{m_i^{n\prime}} H_i^{\overline{h}}(x_i',\vec{m}_i'), \quad (12)$$

which concludes the proof. □

References

1. Becker, R., Zilberstein, S., Lesser, V.: Decentralized Markov decision processes with event-driven interactions. In: AAMAS. pp. 302–309 (2004)
2. Becker, R., Zilberstein, S., Lesser, V., Goldman, C.V.: Solving transition independent decentralized Markov decision processes. JAIR 22, 423–455 (2004)
3. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. Math. of OR 27(4), 819–840 (2002)
4. Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: Structural assumptions and computational leverage. JAIR 11, 1–94 (1999)
5. Emery-Montemerlo, R., Gordon, G., Schneider, J., Thrun, S.: Approximate solutions for partially observable stochastic games with common payoffs. In: AAMAS. pp. 136–143 (2004)
6. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. In: AAAI. pp. 709–715 (2004)

7. Hauskrecht, M.: Value-function approximations for partially observable Markov decision processes. JAIR 13, 33–94 (2000)
8. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence 101(1-2), 99–134 (1998)
9. Kumar, A., Zilberstein, S., Toussaint, M.: Scalable multiagent planning using probabilistic inference. In: IJCAI. pp. 2140–2146 (2011)
10. Mostafa, H., Lesser, V.: A compact mathematical formulation for problems with structured agent interactions. In: MSDM (AAMAS Workshop) (2011)
11. Nair, R., Tambe, M., Yokoo, M., Pynadath, D.V., Marsella, S.: Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In: IJCAI (2003)
12. Nair, R., Varakantham, P., Tambe, M., Yokoo, M.: Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In: AAAI. pp. 133–139 (2005)
13. Oliehoek, F.A.: Value-Based Planning for Teams of Agents in Stochastic Partially Observable Environments. Ph.D. thesis, University of Amsterdam (2010)
14. Oliehoek, F.A., Kooi, J.F., Vlassis, N.: The cross-entropy method for policy search in decentralized POMDPs. Informatica 32, 341–357 (2008)
15. Oliehoek, F.A., Spaan, M.T.J., Vlassis, N.: Optimal and approximate Q-value functions for decentralized POMDPs. JAIR 32, 289–353 (2008)
16. Oliehoek, F.A., Spaan, M.T.J., Whiteson, S., Vlassis, N.: Exploiting locality of interaction in factored Dec-POMDPs. In: AAMAS. pp. 517–524 (2008)
17. Pajarinen, J., Peltonen, J.: Efficient planning for factored infinite-horizon DEC-POMDPs. In: IJCAI. pp. 325–331 (2011)
18. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 2nd edn. (2003)
19. Seuken, S., Zilberstein, S.: Memory-bounded dynamic programming for DEC-POMDPs. In: IJCAI (2007)
20. Spaan, M.T.J., Oliehoek, F.A., Amato, C.: Scaling up optimal heuristic search in Dec-POMDPs via incremental expansion. In: IJCAI. pp. 2027–2032 (2011)
21. Szer, D., Charpillet, F., Zilberstein, S.: MAA*: A heuristic search algorithm for solving decentralized POMDPs. In: UAI. pp. 576–583 (2005)
22. Varakantham, P., young Kwak, J., Taylor, M., Marecki, J., Scerri, P., Tambe, M.: Exploiting coordination locales in distributed POMDPs via social model shaping. In: ICAPS (2009)
23. Varakantham, P., Marecki, J., Yabu, Y., Tambe, M., Yokoo, M.: Letting loose a SPIDER on a network of POMDPs: Generating quality guaranteed policies. In: AAMAS (2007)
24. Velagapudi, P., Varakantham, P., Scerri, P., Sycara, K.: Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In: AAMAS (2011)
25. Witwicki, S.J.: Abstracting Influences for Efficient Multiagent Coordination Under Uncertainty. Ph.D. thesis, University of Michigan (2011)
26. Witwicki, S.J., Durfee, E.H.: Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In: ICAPS (2010)