Heuristic Search of Multiagent Influence Space

Stefan J. Witwicki
GAIPS / INESC-ID
Instituto Superior Técnico
Porto Salvo, Portugal
stefan.witwicki@ist.utl.pt

Frans A. Oliehoek
CSAIL, MIT / DKE, Maastricht University
Maastricht, The Netherlands
fao@csail.mit.edu

Leslie P. Kaelbling
CSAIL
MIT
Cambridge, MA 02139, USA
lpk@csail.mit.edu

ABSTRACT

Multiagent planning under uncertainty has seen important progress in recent years. Two techniques, in particular, have substantially advanced efficiency and scalability of planning. Multiagent heuristic search gains traction by pruning large portions of the joint policy space deemed suboptimal by heuristic bounds. Alternatively, 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 larger problems (for Dec-POMDPs and subclasses) than previously possible. In this paper, we take the logical albeit nontrivial next step of combining multiagent A* search and influence-based abstraction into a single algorithm. The mathematical foundation that we provide, such as partially-specified influence evaluation and admissible heuristic definition, enables an investigation into whether the two techniques bring complementary gains. Our empirical 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.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory, Performance

Keywords

Multiagent Planning Under Uncertainty, Heuristic Search, Multiagent A*, Influence-Based Abstraction, TD-POMDP.

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 [1, 2, 22, 23, 25]. 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.

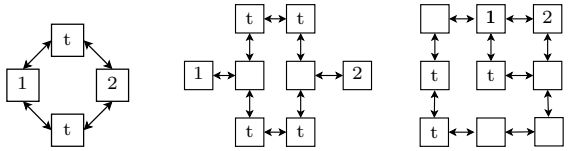
The problem of multiagent planning under uncertainty can be formalized as a decentralized partially observable Markov decision process (Dec-POMDP) [3]. However, its solution is provably intractable (NEXP-complete). As such, many methods either focus on finding approximate solutions without quality guarantees [10, 5, 13, 18, 22, 23], or providing optimal solutions for restricted subclasses. In particular, more efficient procedures have been developed for problems that exhibit *transition and observation independence* [2, 11, 11, 21] or *reward independence* [1]. Unfortunately, these subclasses are too restrictive for many interesting tasks, such as mobile agents collaborating in the search for a target.

The *transition-decoupled* POMDP (TD-POMDP) [25] has recently 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. The core idea is to exploit independence between agents by formalizing the *influence* they can exert on each other. This abstract representation of interaction-related behavior parameterizes a search space of joint influences, which is often significantly smaller than the joint policy space (cf. [24] chapter 4) and, in principle, cheaper to search. Nevertheless, like the policy space, the influence space can still grow exponentially in problem size.

The challenge that we address here is how to search the influence space efficiently. Whereas previous TD-POMDP solutions have focused on exhaustive influence-space search, in general Dec-POMDPs, A* policy-space search guided by heuristics, i.e., multiagent A* (MAA*), has been shown to be an extremely powerful method for reducing the computation required to find optimal solutions [14, 19, 20]. The main contribution of this paper is to show how the strengths of heuristic policy-space search can be transferred to influence-space search.

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,

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.



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

Figure 1: HOUSESEARCH environments. '1'/'2' marks search robot start positions. 't' marks possible static target locations.

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

After describing a motivating problem domain, we review the TD-POMDP model and influence-based policy abstraction, and explain how they can be exploited to find optimal solutions via optimal influence space search.

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 given a prior probability distribution over its location and a model of its movement. We assume that the target either remains stationary or moves 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. Such an environment can be represented by a graph, as illustrated in Fig. 1. At every time-step, each agent i can stay in its current node n or move to a neighboring node n'. The location of 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 found. When a robot is in the same node n as the target, there is a probability of detecting the target $p(detect_i|l_{target},l_i)$, an event which will be modeled by a state variable 'target found by agent i' (denoted f_i). When the target is detected, the agents receive a reward r_{detect}. Given the prior distribution and model of target behavior, the goal is to optimize the expected 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, and then we describe how a TD-POMDP extends this model to multiple agents.

A *factored partially observable Markov decision process* for a single agent (indexed i for consistency with multiagent notation later on) 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 that maximizes the expected sum of rewards over h time steps or stages. Such a policy maps from *beliefs*, probability distributions over states, to actions. While solving a POMDP is widely considered to be an intractable problem, in the last two decades many exact and approximate solution methods have been proposed (see, e.g., [7]).

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

Table 1: Different types of state factors that make up s_i.

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.[1] Moreover, the set of state factors can be divided into factors that occur only in one agent's local state space ('non-mutual' factors (NMFs)) and factors that are 'mutually modeled' by more than one agent (MMFs). A TD-POMDP imposes the restriction that each state factor can be *directly* affected by the action of at most one agent. That is, in the 2TBN, each factor can have an incoming edge from only 1 action variable. 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 leads to different parts of an agent's local state, as summarized in Table 1. Using the notation defined in this table, we will write the local state of an agent i as $s_i = \langle x_i^l, x_i^u, m_i^l, m_i^n, m_i^u \rangle = \langle x_i, m_i \rangle$. The joint reward function for the TD-POMDP is the summation of the individual reward functions for each agent's POMDP: $R(s,a) = \sum_i R_i(s_i,a_i)$, and we assume an initial joint state distribution $\boldsymbol{b}^0$. For a more formal introduction of the TD-POMDP framework, see [24, 25].

The 2TBN representation of the TD-POMDP's transition, observation, and reward model for HOUSESEARCH is illustrated in Fig. 2. Here, two search agent's local states overlap such that both model the target location and the factors f_1,f_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, in this domain, each agent's reward function R_i can also be factored as the sum of two components R_{detect} and R_{move}. The former models the rewards for de-

[1]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. One can alternatively consider the existence of a single transition model T defined over the joint action and the joint state.

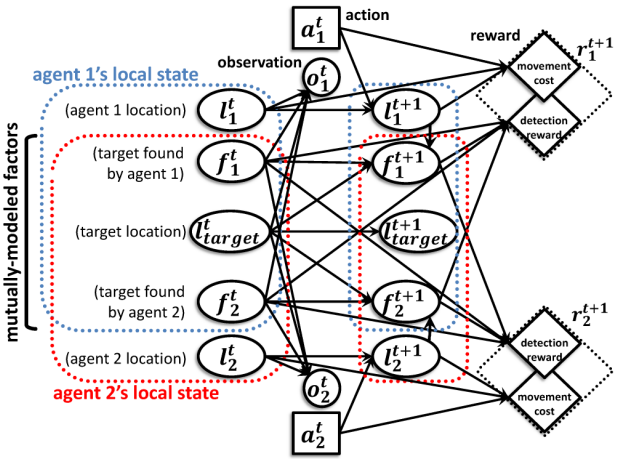


Figure 2: A TD-POMDP for HOUSESEARCH.

tection, 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 do the agents get 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 components only depend on the agents' non-mutual locations and local actions.

The TD-POMDP is a non-trival subclass of the factored Dec-POMDP [15], for which the NEXP-completeness result still holds [24]. This also means that single-agent POMDP solution methods do not directly apply. Intuitively, in a multiagent context, we are now searching for a joint policy $\pi = \langle \pi_1, \ldots, \pi_n \rangle$. Moreover, an agent can no longer base its policy on a simple belief over states, as this does not account for the beliefs and actions of its teammates.

2.3 Influences and Local Models

A well-known solution method for Dec-POMDPs, called JESP [10], 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 via dynamic programming. 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.

Since a TD-POMDP is a Dec-POMDP, JESP directly applies. However, because of the special structure a TD-POMDP imposes, we can account for this structure to compute the best response in a potentially more efficient way: rather than maintaining a JESP belief $b_i(s, \vec{o}_{-i})$, agent i can maintain a condensed belief $b_i(s_i^t, \vec{m}_i^{t-1})$ over just its own local state and the history of mutually modeled factors [25]. 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 $\vec{m}_i^t$.[2] That is, $\vec{m}_i^t$ d-separates the agent's observation history $\vec{o}_i^t$ from those of other agents $\vec{o}_{-i}^{t-1}$. For instance, given the DBN connectivity in Fig. 2, all information agent 2 has about l_1^t is inferred from the history of f_1^t and l_{target}^t.

[2]Note that m_i^t is contained in s_i^t such that we can write $b_i(s_i^t, \vec{m}_i^{t-1}) = b_i(x_i^t, \vec{m}_i^t)$.

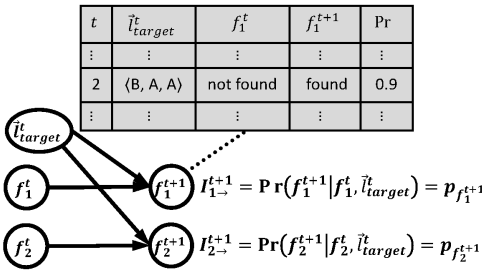


Figure 3: The Influence DBN for HOUSESEARCH.

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. 2 agent 1 only influences agent 2 via the f_1 factor. 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 $m_i^{n,0}, \ldots, m_i^{n,h-1}$ can be clustered together, since they will lead to the same best response of agent i.

To formalize this idea, 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 $t = 1, \ldots, h-1$.[3] 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 only necessary to use $\vec{m}_i$, the history of mutual features, as the parents [25]. I.e., the CPTs are specified as $p_{m_i^{n,t+1}}(v|\vec{m}_i^t)$. 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. 3 illustrates the dependencies of an influence point in a so-called *influence DBN*. For instance, the possi-

[3]For $t = 0$, the (non-conditional) distribution is specified by the initial state distribution b^0. The CPTs for subsequent stages may differ (from one another) because they summarize other agents' policies, which can depend on history.

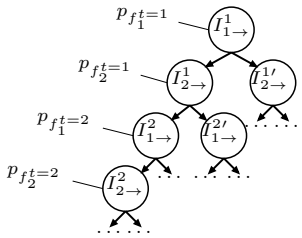


Figure 4: The influence search tree for HOUSESEARCH.

ble CPTs $p_{f_1^{t+1}}$ are conditioned on $\vec{l}^t_{target}$, the history of the target location, as well as 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 $\langle s_i^t, \vec{m}_i^{t-1}\rangle$ (or equivalently $\langle x_i^t, \vec{m}_i^t\rangle$), such that, as discussed above, a belief is of the form $b_i(s_i^t, \vec{m}_i^{t-1})$. 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 the constraint that its solution must be a policy that adheres to the probabilities dictated by the outgoing influence point (specified by I_i). 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 as a mixed integer linear program (MILP) [24, chapter 5].

2.4 Optimal Influence Search

The key property of these influences is that they can be used to compactly represent many of the other agents' policies. Rather than searching in the larger space of joint policies, we can search in the space of joint influence points and for each of them compute the agents' best responses to compute their value. 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 against $I_{\rightarrow i}$ 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) [25] does this by constructing a tree, as illustrated in Fig. 4. 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 in fact demonstrated that influence abstraction can lead to significant gains in performance, thereby establishing itself as a state-of-the-art method for computing optimal solutions for weakly-coupled agents [25].

3. HEURISTIC INFLUENCE SEARCH

The previous section explained how OIS can greatly improve over other methods by searching in the search of joint influences, which can be much smaller than the space of joint policies. However, the weakness of OIS is that it needs to search this space exhaustively. In contrast, for general Dec-POMDPs, heuristic search methods (in particular A*, see, e.g., [17]) have shown to be very effective [19]. 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

To 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 [14]. 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 [20] 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, 14] .

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

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 5. It shows that for a certain node $\check{I}$ in the search tree, the influences for the first number $(\overline{h})$ of stages

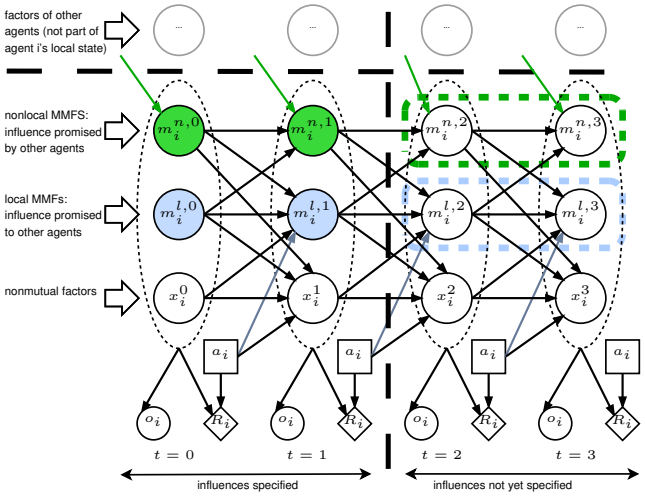


Figure 5: A partially specified joint influence point $\check{I}$ from the perspective of agent i. Dashed black elipses denote the agent's local state. The figure does not include unaffectable factors. Influences are specified for stage 0,1. Green (dark) nodes are specified incoming influences, blue (light) nodes are specified outgoing influences. The dashed boxes denote the unspecified incoming (green) and outgoing (blue) influences for stages 2,3.

are specified (up to but *not* including stage $\overline{h}$). For now we assume that *all* influences at stage $\overline{h}-1$ are specified, i.e., we assume that $\check{I}^{\overline{h}-1}$ is a fully specified influence slice. Figure 5, in which $\overline{h}=2$, shows that computation of $F_i(\check{I})$ depends on only a subset of state factors (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.

3.2.1 Step 1: The Unspecified-Influence Stages.

The goal here is to, for each IALM state, to compute a heuristic value $H_i^{\overline{h}}$, analogous to the term used in (3), that is an optimistic estimate of the value of that state over the remaining (unspecified-influence) stages. In particular, we use an approach similar to Q_{MDP}: 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 transition 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:

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

which upper bounds the value of the underlying restricted-scope MDP given *any* incoming influence point $I_{\to i}$:

$$V_{i,MDP}^{I_{\to i}}(x_i,\vec{m}_i) = \max_{a_i}\Big[R(s_i,a_i) + \sum_{x_i',m_i^{l\prime},m_i^{u\prime},m_i^{n\prime}} \Pr(x_i',m_i^{l\prime},m_i^{u\prime},m_i^{n\prime}|x_i,\vec{m}_i,a_i,I_{\to i}) V_{i,MDP}^{I_{\to i}}(x_i',\vec{m}_i')\Big]. \quad (6)$$

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 H_i^t can be computed via dynamic programming starting at the last stage $h-1$ and working back to stage $\overline{h}$.

3.2.2 Step 2: The Specified-Influence Stages.

Here we use $H_i^{\overline{h}}$ found in stage 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:

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

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 $F_i(\check{I}) \triangleq \overline{V}_i(\overline{I})$.

3.2.3 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 4 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 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 $\langle x_i^{l,\overline{h}-1}, x_i^{u,\overline{h}-1}, m_i^{l,\overline{h}-1}, m_i^{u,\overline{h}-1}\rangle$ the local state will always transition to the fully specified local state $\langle x_i^{l,\overline{h}-1}, x_i^{u,\overline{h}-1}, m_i^{l,\overline{h}-1}, m_i^{n,\overline{h}-1}, m_i^{u,\overline{h}-1}\rangle$ with the highest heuristic value.

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

The implication of Theorem 1, whose proof is given in the appendix, 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.

3.3 A Tighter Heuristic

While the heuristic of the previous section is admissible, it is not very tight, because the second maximization in (5)

corresponds to *always* assuming the most optimistic incoming influences. For instance, in the rectangle example, it will assume that the other agent finds the target in the second stage $t = 1$. However, from the possible locations of the target, we know that it will never be possible for the other agent to find the target at $t = 1$, it will take at least two steps. Next, we present a new heuristic that exploits this insight to yield a tighter upper bound to use during search.

Note that $V^{I_{\rightarrow i}}_{i,MDP}(x_i^t, \vec{m}_i^t)$ in (6) can be expanded to

$$\max_{a_i} \Big[R_i(s_i, a_i) + \sum_{\langle x_i, m_i^l, m_i^u \rangle^{t+1}} \Pr(\langle x_i^t, m_i^l, m_i^u \rangle^{t+1} | x_i^t, m_i^t, a_i) \sum_{m_i^{n,t+1}} \Pr(m_i^{n,t+1} | \vec{m}_i^t, I_{\rightarrow i}) V^{I_{\rightarrow i}}_{i,MDP}(x_i^{t+1}, \vec{m}_i^{t+1}) \Big] \quad (8)$$

due to the structure of the TD-POMDP. An important aspect in (8) is that $\Pr(m_i^{n,t+1} | \vec{m}_i^t, I_{\rightarrow i})$ exactly corresponds to one of the entries in $p_{m_i^{n,t+1}}$ that $I_{\rightarrow i}$ specifies.

Our first heuristic picks the heuristic best values for m_i^n, which we will denote $m_i^{n\star}$, and then assumes a optimistic influence $I^{\star}_{\rightarrow i}$ that prescribes $\Pr(m_i^{n\star} | \vec{m}_i^t, I^{\star}_{\rightarrow i}) = 1$. Here the idea is to use a more realistic (but still optimistic) influence $I^{*}_{\rightarrow i}$ by making use of an upper bound on $\Pr(m_i^{n\star} | \vec{m}_i^t, I_{\rightarrow i})$, which may be precomputed by examining the TD-POMDP CPT for factor m_i^n.

In particular, we compute the following upper bounds on $p_{m_i^{n,t}=v}$ the probability of each value v of a non-local factor $m_i^{n,t}$ as follows:

$$UB_{m_i^{n,t} | \vec{m}_i^{t-1}}(v) = \max_{\mathbf{v}_p \in PPP(m_i^{n,t} | \vec{m}_i^{t-1})} \Pr(v | \mathbf{v}_p), \quad (9)$$

where $\Pr(\cdot|\cdot)$ is specified by a CPT of the 2TBN, $\mathbf{v}_p$ denotes an instantiation of the parents of $m_i^{n,t}$ in the 2TBN, and where the positive probability parents, $PPP(m_i^{n,t} | \vec{m}_i^{t-1})$, is the set of such instantiations that 1) have positive probability of occuring, and 2) are consistent with $m_i^{n,t-1}$ (specified by $\vec{m}_i^{t-1}$).

We now use these bounds to define the more realistic influence $I^{*}_{\rightarrow i}$ and thus heuristic. In order to compute heuristic value $H(x_i^{t-1}, \vec{m}_i^{t-1})$, we first order the possible values of $m_i^{n,t}$ according to their heuristic next-stage value $H(x_i^t, \vec{m}_i^t)$. Next, we define $I^{*}_{\rightarrow i}$ to be such that it defines a CPT $p_{m_i^{n,t}}$ that gives maximal weight to the high-ranked values of $m_i^{n,t}$. We create this distribution as follows: first we select the highest ranked value v^* of $m_i^{n,t}$ and we assign it probability $UB_{m_i^{n,t} | \vec{m}_i^{t-1}}(v^*)$, then we select the next best ranked value v' and either assign it the remaining probability mass (if that is less then its upper bound) or we assign it its upper bound and continue with the next-best value, etc. The heuristic is now defined by substituting the thusly obtained distribution for $\Pr(m_i^{n,t+1} | \vec{m}_i^t, I_{\rightarrow i})$ in (8).

4. EXPERIMENTS

We now present a empirical evaluation of our heuristic influence-space search method. Our primary hypothesis is that exhaustive optimal influence-space search (OIS), which has been shown to solve a number of weakly-coupled transition-dependent problems more efficiently than policy space search methods, 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., [19]), we expect that the problems considered here 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 beyond what optimal Dec-POMDP solvers have demonstrated to 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 (we use the version where the agents can wait) [25].

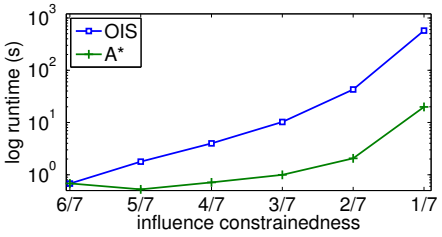


Figure 6: Runtimes on SATELLITEROVER.

For HOUSESEARCH, we experimented with different degrees of stochasticity. I.e., we considered problems ranging from deterministic observations and deterministic actions, labeled "d.o.d.a.", to stochastic observations (where the probability of observing no target when in the same room as the target is 0.25) and stochastic actions (where the probability that a move action will fail is 0.1), labeled "s.o.s.a". For all problems, the parameters were set to $c_{time} = -5$, $c_i = -1$ for each movement action, and $r_{detect} = 0$. Table 2 compares the runtimes of OIS with those of A* using the two variants of our restricted scope restricted horizon heuristic (where A*$_1$ corresponds to that described in Section 3.2 and A*$_2$ corresponds to that described in Section 3.3). As shown, using the simplest variant of our heuristic can lead to significant speed-ups 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* s.o.s.a). It also allows scaling up to larger time horizons than was previously possible. We also see that the heuristic A*$_2$ is indeed tighter, allowing for more pruning and hence faster solutions on almost all problems. 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.)

We also tested A*$_1$ on SATELLITEROVER, in which the lengths of task execution windows were systematically varied to affect the level of *influence constrainedness* [25]. 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. 6. 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

	Diamond (d.o.d.a)			*Diamond* (s.o.d.a)			*Diamond* (d.o.s.a)			*Diamond* (s.o.s.a)			*Rectangle* (s.o.d.a)			*Squares* (s.o.d.a)		
h	OIS	A*$_1$	A*$_2$	OIS	A*$_1$	A*$_2$	OIS	A*$_1$	A*$_2$	OIS	A*$_1$	A*$_2$	OIS	A*$_1$	A*$_2$	OIS	A*$_1$	A*$_2$
1	0.28	0.27	0.29	0.19	0.30	0.25	0.22	0.28	0.23	0.25	0.32	0.23	0.08	0.10	0.17	0.21	0.25	0.28
2	1.64	0.83	0.26	2.28	0.92	0.25	3.13	1.86	0.29	8.68	2.41	0.64	0.72	0.71	0.47	1.73	1.33	0.67
3	8.84	1.77	0.68	35.60	5.93	0.89	151.6	11.63	1.38	8,871	52.08	2.37	16.52	17.88	7.85	31.96	34.55	10.22
4	101.6	8.50	1.28	811.4	48.75	2.86		436.0	4.89		3,066	14.39	621.2	412.3	138.6	1,716	1,101	167.5
5	945.0	31.80	7.90		953.1	44.57			178.1			44.52			4,187			6,295

Table 2: Runtimes (in seconds), including heuristic computation (observed to be negligible), on variations of HOUSESEARCH.

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 probably find the target, in which case the agent need not look itself and incur the associated movement cost. To remedy this problem, we developed a simple specialized adaptation (A*-*imc*) that ignores the movement cost component of the factored reward in the first term of Equation 7. While this modification causes the heuristic to be less tight, it also takes away the bias against movement actions. Results for this modification are shown in Table 3. An interesting phenomenon occurs for *Rectangle* and *Squares* where for longer time horizons, runtimes are significantly decreased because the no-movement bias has been eliminated, but where for for shorter horizons we see slight increases in runtimes because here the optimal policy is actually to stay. Likewise, for *Diamond*, very little movement is required to find the target; in this case, the search also suffers from the fact that ignoring movement costs actually loosens the heuristic, causing more nodes to be expanded. All in all, this demonstrates that using specialized domain knowledge can significantly increase the effectiveness of A* influence space search.

	Diamond (s.o.d.a)		*Rectangle*		*Squares*	
h	A*$_2$	A*$_2$-*imc*	A*$_2$	A*$_2$-*imc*	A*$_2$	A*$_2$-*imc*
1	0.25	0.56	0.17	0.30	0.28	0.50
2	0.25	0.34	0.47	0.57	1.67	1.20
3	0.89	1.10	7.85	7.18	10.22	12.10
4	2.86	3.86	138.6	14.71	167.5	46.95
5	44.57	146.5	4,187	222.0	6,295	422.6

Table 3: Ignoring movement costs in heuristic calculation.

5. RELATED WORK

Having reviewed Dec-POMDP heuristic search [14, 19, 20] and TD-POMDP influence-space search [24, 25], which are most closely related to the work we have developed here, we now describe connections to other recent models and methods. For instance, the EDI-CR model [9] 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, instead of decoupling the problem into influence-abstracted local models.

The DPCL [22, 23] exploits 'coordination locales' for efficient computation of an agent's response policy by incorporating effects of other agents' policies into a compact local model, which is similar the TD-POMDP's IALM. However, in contrast to the optimal search methods that we develop, the DPCL has only ever afforded approximate solutions. Other methods have been developed for computing approximate solutions for general factored Dec-POMDPs, of which the TD-POMDP could be considered a specialized instance. Their more general factored structure has been exploited by using collaborative graphical Bayesian games in combination with non-serial dynamic programming [15] and approximate inference [12] in the finite-horizon case. In the infinite-horizon case finite state controllers and EM [8, 16] have been proposed. In contrast to the work presented here, these methods search in the policy space rather than the influence space.

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 a optimal decentralized probabilistic search problem, thereby showing the first application of 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 exhaustive 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. 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 [19]. 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

We thank the anonymous reviewers for their constructive

feedback. This work was supported, in part, by the Fundação para a Ciência e a Tecnologia (FCT) and the CMU-Portugal Program under project CMU-PT/SIA/0023/2009, by AFOSR MURI project #FA9550-09-1-0538, and by NWO CATCH project #640.005.003.

8. REFERENCES

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

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

[3] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Math. of OR*, 27(4):819–840, 2002.

[4] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *JAIR*, 11:1–94, 1999.

[5] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *AAMAS*, pages 136–143, 2004.

[6] M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *JAIR*, 13:33–94, 2000.

[7] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

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

[9] H. Mostafa and V. Lesser. Compact mathematical programs for DEC-MDPs with structured agent interactions. In *UAI*, pages 523–530, 2011.

[10] R. Nair, M. Tambe, M. Yokoo, D. V. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, 2003.

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

[12] F. A. Oliehoek. *Value-Based Planning for Teams of Agents in Stochastic Partially Observable Environments*. PhD thesis, University of Amsterdam, 2010.

[13] F. A. Oliehoek, J. F. Kooi, and N. Vlassis. The cross-entropy method for policy search in decentralized POMDPs. *Informatica*, 32:341–357, 2008.

[14] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *JAIR*, 32:289–353, 2008.

[15] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *AAMAS*, pages 517–524, 2008.

[16] J. Pajarinen and J. Peltonen. Efficient planning for factored infinite-horizon DEC-POMDPs. In *IJCAI*, pages 325–331, 2011.

[17] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2nd edition, 2003.

[18] S. Seuken and S. Zilberstein. Memory-bounded dynamic programming for DEC-POMDPs. In *IJCAI*, 2007.

[19] M. T. J. Spaan, F. A. Oliehoek, and C. Amato. Scaling up optimal heuristic search in Dec-POMDPs via incremental expansion. In *IJCAI*, pages 2027–2032, 2011.

[20] D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. In *UAI*, pages 576–583, 2005.

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

[22] P. Varakantham, J. young Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *ICAPS*, 2009.

[23] P. Velagapudi, P. Varakantham, P. Scerri, and K. Sycara. Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In *AAMAS*, 2011.

[24] S. J. Witwicki. *Abstracting Influences for Efficient Multiagent Coordination Under Uncertainty*. PhD thesis, University of Michigan, 2011.

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

APPENDIX

PROOF OF THEOREM 1. We need to show that

$$\forall_{I_i} \quad F_i(\check{I}) = \overline{V}_i(\overline{I}) \geq V_i(I^{*|\check{I}}) \tag{10}$$

We assume an arbitrary $I_{\rightarrow i}, I_{i\rightarrow}$ consistent with $\check{I}$. Since the first $\overline{h}-1$ stages are identical, (10) clearly holds if

$$\forall_{b_i,a_i} \qquad \overline{Q}_i^{\overline{h}-1}(b_i,a_i) \geq Q_i^{\overline{h}-1,I_{\rightarrow i},I_{i\rightarrow}}(b_i,a_i). \tag{11}$$

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

$$\overline{R}^{\overline{h}-1}(b_i,a_i) \geq R^{\overline{h}-1}(b_i,a_i) + \sum_{b'} P(b'|b_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(b'_i). \tag{12}$$

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

$$\forall_{x_i,\vec{m}_i} \quad \sum_{x'_i,m_i^{l\prime},m_i^{u\prime}} \Pr(x'_i,m_i^{l\prime},m_i^{u\prime}|s_i,a_i) \max_{m_i^{n\prime}} H_i^{\overline{h}}(x'_i,\vec{m}'_i) \geq \sum_{s'_i}\sum_{o_i} \Pr(s'_i,o_i|s_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x'_i,\vec{m}'_i,b'_i) \tag{13}$$

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 [6], we have

$$\begin{aligned} & \sum_{s'_i}\sum_{o_i} \Pr(s'_i,o_i|s_i,a_i) V_i^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x'_i,\vec{m}'_i,b'_i) \\ \leq & \sum_{s'_i}\sum_{o_i} \Pr(s'_i,o_i|s_i,a_i) V_{i,MDP}^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x'_i,\vec{m}'_i) \\ \leq & \sum_{s'_i} \Pr(s'_i|s_i,a_i) V_{i,MDP}^{\overline{h},I_{\rightarrow i},I_{i\rightarrow}}(x'_i,\vec{m}'_i) \\ \leq & \sum_{s'_i} \Pr(s'_i|s_i,a_i) V_{i,MDP}^{\overline{h},I_{\rightarrow i}}(x'_i,\vec{m}'_i) \end{aligned} \tag{14}$$

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

$$\leq \sum_{s'_i} \Pr(s'_i|s_i,a_i) H_i^{\overline{h}}(x'_i,\vec{m}'_i) \leq \sum_{x'_i,m_i^{l\prime},m_i^{u\prime}} \Pr(x'_i,m_i^{l\prime},m_i^{u\prime}|s_i,a_i) \max_{m_i^{n\prime}} H_i^{\overline{h}}(x'_i,\vec{m}'_i), \tag{15}$$

which concludes the proof. □